

Research Design

Declare, Diagnose, Redesign

Graeme Blair, Alexander Coppock, and Macartan Humphreys

2021-06-08

Contents

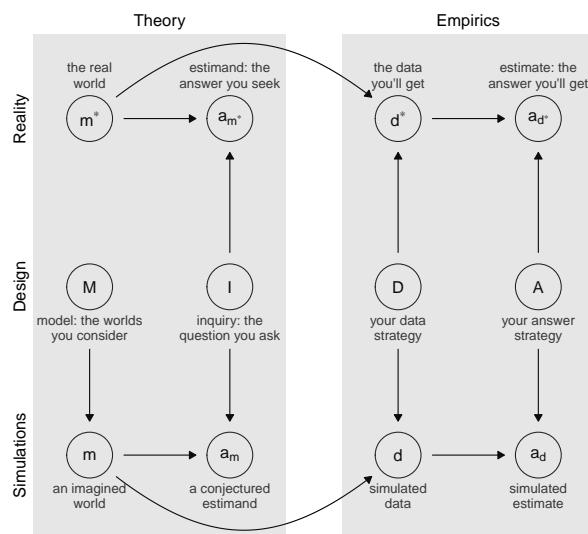
I	Introduction	1
1	Preamble	3
1.1	How to read this book	3
1.2	How to work this book	5
1.3	What this book will not do	5
2	What is a research design?	7
2.1	The four elements of research design	7
2.2	Declaration, diagnosis, redesign	14
2.3	Example: October surprise	18
2.4	Putting designs to use	21
3	Research design principles	25
4	Software primer	33
4.1	Installing R	33
4.2	Declaring research design elements	34
4.3	Building a design from design elements	41
4.4	Simulating a research design	43
4.5	Diagnosing a research design	44
4.6	Redesign	45
4.7	Library of designs	46
4.8	Complex declarations	46
	Further reading	47
II	Declaration, Diagnosis, Redesign	49
5	Declaration	51
5.1	Definition of research designs	51
5.2	Declaration in code	54
	Further reading	57

6 Specifying the model	59
6.1 Elements of models	60
6.2 Lexicon of common variable types	64
6.3 Substantive justifications for choices	67
6.4 Declaring models in code	69
7 Defining the inquiry	81
7.1 Elements	82
7.2 Examples of inquiries	87
7.3 How to choose among inquiries	90
7.4 Declaring inquiries in code	92
Further reading	94
8 Crafting a data strategy	95
8.1 Elements of data strategies	98
8.2 Seek M:I::D:A Parallelism	118
8.3 Robustness	119
8.4 Declaring data strategies in code	123
9 Choosing an answer strategy	127
9.1 Types of answer strategies	128
9.2 Uncertainty	137
9.3 Applying design principles	141
9.4 Declaring answer strategies in code	146
10 Diagnosis	151
10.1 Diagnostic statistics and diagnosands	152
10.2 Estimating diagnosands analytically	154
10.3 Estimating diagnosands via simulation	156
10.4 Types of diagnosands	161
10.5 Diagnosand completeness	163
10.6 Choosing diagnosands to explore design tradeoffs	163
10.7 Diagnosis under model uncertainty	166
10.8 Diagnosing a design in code	169
10.9 Summary	172
Further reading	173
11 Redesign	175
11.1 Power curve example	175
11.2 Redesign over multiple design parameters	176
11.3 Redesign over answer strategies	178
11.4 Redesign under model uncertainty	181
11.5 Redesigning in code	183
11.6 Summary	184
12 Design example	185
12.1 Declaration in words	185

12.2 Declaration in code	186
12.3 Diagnosis	188
12.4 Redesign	189
III Research Design Library	193
13 Research Design Library	195
14 Observational : descriptive	197
14.1 Simple random sampling	197
14.2 Cluster random sampling	201
14.3 Multi-level regression and poststratification	206
14.4 Index creation	210
15 Observational : causal	217
15.1 Process tracing	217
15.2 Selection-on-observables	222
15.3 Difference-in-differences	227
15.4 Instrumental variables	232
15.5 Regression discontinuity designs	238
16 Experimental : descriptive	245
16.1 Audit experiments	246
16.2 List experiments	250
16.3 Conjoint experiments	260
16.4 Behavioral games	266
17 Experimental : causal	275
17.1 Two-arm randomized experiments	276
17.2 Block-randomized experiments	283
17.3 Cluster-randomized experiments	293
17.4 Subgroup designs	296
17.5 Factorial experiments	300
17.6 Encouragement designs	305
17.7 Placebo-controlled experiments	311
17.8 Stepped-wedge experiments	316
17.9 Crossover experiments	320
17.10 Randomized saturation experiments	324
17.11 Experiments over networks	328
18 Complex designs	335
18.1 Mixed methods	335
18.2 Discovery using causal forests	341
18.3 Structural estimation	346
18.4 Meta-analysis	352
18.5 Multi-site studies	355

IV Research Design Lifecycle	363
19 Research Design Lifecycle	365
20 Brainstorming	367
21 Planning	371
21.1 Ethics	371
21.2 Approvals	373
21.3 Partners	374
21.4 Funding	376
21.5 Piloting	378
21.6 Criticism	381
21.7 Preanalysis Plan	382
22 Realization	393
22.1 Pivoting	393
22.2 Populated preanalysis plan	395
22.3 Reconciliation	396
22.4 Writing	399
22.5 Publication	401
23 Integration	405
23.1 Communicating	406
23.2 Decisionmaking	407
23.3 Archiving	409
23.4 Reanalysis	412
23.5 Replication	417
23.6 Meta-analysis	419
V Epilogue	423
24 Epilogue	425
VI References	429

Research Design: Declaration, Diagnosis, Redesign



Part I

Introduction

Chapter 1

Preamble

This book introduces a new way of thinking about research designs in the social sciences. Our hope is that this approach will make designing research studies easier – easier to produce strong research designs, but also easier to share designs and build on the designs of others.

The core idea is the *MIDA* framework, in which a research design is characterized by four elements: a model, an inquiry, a data strategy, and an answer strategy. We have to understand each of the four on their own and also how they interrelate. The design encodes your beliefs about the world, it describes your questions, and it lays out how you go about answering those questions, both in terms of what data you collect and how you analyze it. In strong designs, choices made in the model and inquiry are reflected in the data and answer strategies, and vice versa.

We think of designs as objects that can be interrogated. Each of the four design elements can be “declared” in computer code and – if done right – the information provided is enough to “diagnose” the quality of the design through computer simulation. Researchers can then select the best design for their purposes by “redesigning” over alternative, feasible designs.

This way of thinking pays dividends at multiple points in the research design lifecycle: brainstorming an idea, planning the design, implementing it, and integrating the results into the broader research literature. The declaration, diagnosis, and redesign process informs choices made from the beginning to the end of a research project.

1.1 How to read this book

We had multiple audiences in mind when writing this book. First, we’re thinking of the set of people who could benefit from a high-level introduction to these

ideas. If we only had 30 minutes with a person to try and get them to understand what our book is about, we would give them Part I. We're thinking of beginners, people who are new to the practice of research design and who are embarking on their first empirical projects. The *MIDA* framework introduced in Part I accommodates many different empirical approaches: qualitative and quantitative, descriptive and causal, observational and experimental. Beginners starting out in any of these traditions can use our framework to consider how the design elements in those approaches fit together. We're also thinking of researchers-in-training: graduate students in seminar courses where the main purpose is to read papers and discuss how well the empirics match the theory. These discussions can sometimes be a jumble of miscellaneous complaints, but our framework can focus attention on the most relevant concerns. What, exactly, is the inquiry? Is it the right one to be posing, and does the design do a good job of generating answers to it? We're also thinking of funders and decision-makers, who often wish to assess research not in terms of its results but its design. Our approach provides a way of defining the design and diagnosing its quality.

Part II is more involved. We provide the mathematical foundations of the *MIDA* framework. We walk through each component of a research design in detail, describe the finer points of design diagnosis, and explain how to carry out a redesign. Part II will resonate with several audiences of applied researchers both inside and outside of academia. We imagine it could be assigned early in a graduate course on research design in any of the social sciences. Data scientists and monitoring and evaluation professionals will find value in our framework for learning about research designs. Scholars will find value in declaring, diagnosing, and redesigning designs whether they are implementing randomized trials, multi-method archival studies, or calibrating structural theories with data.

In Part III, we apply the general framework to specific research designs. The result is a library of common designs. Many empirical research designs are included in the library, but not all. The set of entries covers a large portion of what we see in current empirical practice across social sciences, but it is not meant to be exhaustive. We don't expect that any readers will read straight through the design library, but will instead pick-and-choose depending on their interests.

We are thinking of three kinds of uses for entries in the design library. Collectively, the design entries serve to illustrate the fundamental principles of design. The entries clarify the variety of ways in which models, inquiries, data strategies, and answer strategies can be connected and show how high level principles operate in common ways across very different designs. The second use is pedagogical. The library entries provide hands-on illustrations of designs in action. A researcher interested in understanding the "regression discontinuity design," for example, can quickly see a complete implementation and learn under what conditions the standard design performs well or poorly. They can also compare the suitability of one type of design against another for a given problem. We emphasize that these descriptions of different designs provide entry points but

they are not exhaustive, so we refer the reader to the most up-to-date methodological treatments of the topic. The third use is as a starter kit to help readers get going on designs of their own. Each entry includes code for a basic design that can be fine-tuned to capture the specificities of particular research settings.

The last section of the book describes in detail how our framework can help at each step of the research process. Each of these sections should be readable for anyone who has read Part I. The entry on preanalysis plans, for example, can be assigned in an experiments course as guidance for students filing their first preanalysis plan. The entry on research ethics could be shared among coauthors at the start of a project. The entry on writing a research paper could be assigned to college seniors trying to finish their essays on time.

1.2 How to work this book

We will often describe research designs not just in words, but in computer code. If you want to work through the code and exercises, fantastic. This path requires investment in R, the tidyverse, and the DeclareDesign software package. Chapter 4 helps get you started. We think working through the code is very rewarding, but we understand that there is a learning curve. You could, of course, tackle the declaration, diagnosis, and redesign processes using bespoke simulations in any computer language you like,¹ but it is easier in DeclareDesign because the software guides you to articulate each of the four design elements.

If you want nothing to do with the code, you can skip all the code and exercises and just focus on the text. We have written the book so that understanding of the code is not required in order to understand research design concepts.

1.3 What this book will not do

This is a research design book, not a statistics textbook, nor a cookbook with recipes applicable to all situations. We will not derive estimators, we will provide no guarantees of the general optimality of designs, and we will present no mathematical proofs. Nor will we provide all the answers to all the practical questions you might have about your design.

What we do offer is a language to express research designs. We can help you learn that language so you can describe your own design in it. When you can declare your design in this language, then you can diagnose it, then improve it through redesign.

¹On our Web site, we provide examples in R, Python, Stata, and Excel.

6

1.3

Chapter 2

What is a research design?

At its heart, a research design is a procedure for generating empirical answers to theoretical questions. Research designs can be strong or weak. Assessing whether a design is strong requires having a clear sense of what the question is and knowing whether the answers a study is likely to deliver are reliable. This book offers a language for describing research designs and an algorithm for selecting among them. In other words, it provides a set of tools for weighing and describing the dozens of choices we make in our research activities that together determine whether we can provide useful answers to our questions.

We show that the same basic language can be used to describe research designs whether they target causal or descriptive questions, whether they are focused on theory testing or inductive learning, and whether they use quantitative, qualitative, or mixed methods. We can select a strong design by applying a simple algorithm: declare-diagnose-redesign. Once a design is declared in simple enough language that a computer can understand it, its properties can be diagnosed through simulation. We can then engage in redesign, or comparing across a range of neighboring designs. The same language we use to talk to the computer can be used to talk to others. Reviewers, advisors, students, funders, journalists, and the public need to know four basic things to understand your design.

2.1 The four elements of research design

Empirical research designs share in common that they all have an inquiry I , a data strategy D , and an answer strategy A . Less obviously perhaps, these three elements presuppose a model M of how the world works. The four together, which we refer to as *MIDA*, represent both a conceptual framework for your inquiries and a description of the choices you will make as a researcher to intervene in and learn about the world.

Figure 2.1 shows how these four elements of a design relate to one another, how they relate to real world quantities, and how they relate to simulated quantities. We will unpack this figure in the remainder of this chapter and highlight especially the important parallelisms, between actual processes and simulated processes and between the theoretical (M, I) and the empirical (D, A) halves of a design.

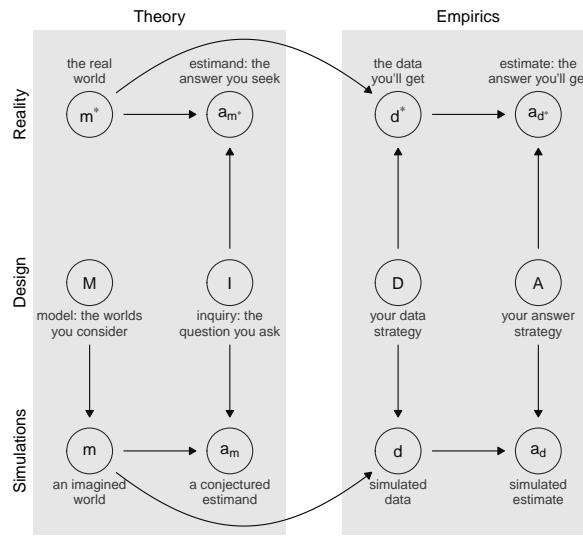


Figure 2.1: The *MIDA* framework. An arrow between two points means that the point at the end of an arrow depends in some way on the one at the start of the arrow. For instance 'the answer you'll get' depends on the data you'll get and the answer strategy you specify.

2.1.1 Model

The set of models in M comprises varied speculations about what causes what and how. It includes guesses about how important variables are generated, how things are correlated, and the sequences of events.

The M in *MIDA* does not necessarily represent your beliefs about how the world works. Instead, it describes a set of possible worlds in enough detail that you can assess how your design would perform if the real world worked like those in M . For this reason we sometimes refer to M as a set of “reference” models. Assessment of the quality of a design is carried out with reference to the models of the world that you provide in M . In other contexts, you might see M described as as the “data generating process.” We would prefer to describe M as the “event generating process” to honor the fact that data are produced or gathered via a

data strategy – and the resulting data are measurements taken of the events generated by the true causal model of the world.

Defining the model can feel like an odd exercise. Since researchers presumably want to learn about the world, declaring a model in advance may seem to beg the question. The discomfort we feel when writing down a model is real, because we simply don't know exactly how the real world works. We nevertheless have to declare models, because designing research requires us to imagine how the design would perform under the many possible ways the world might work. In practice, declaring models about which we are uncertain is already familiar to any researcher who has calculated the statistical power of a design along a range of effect sizes.

2.1.1.1 What's in a model?

The model has two responsibilities. First, the model provides a setting within which a question can be answered. The inquiry I should be answerable *under the model*. If the inquiry is the average difference between two potential outcomes, those two potential outcomes should be described in the model. Second, the model governs what data will be produced by any given data strategy D . The data that will be produced by a data strategy D should be foreseeable under the model. If the data strategy includes random sampling of units from a population and measurement of an outcome, the model should describe the outcome variable all units in that population.

These responsibilities in turn determine what needs to be in the model. In general, the model defines a set of units that we wish to study. Often, this set of units is larger than the set of units that we will actually study empirically, but we can nevertheless define this larger set about which seek to make inferences. The units might be all of the citizens in Lagos, Nigeria, or every police beat in New Delhi. The set may be restricted to the mayors of cities in California or the catchment areas of schools in rural Poland. The model also includes information about the baseline characteristics of those units: how many of each kind of unit there are and how features of the units may be correlated. For descriptive and causal questions alike, models are *causal* models. Even if questions are fundamentally descriptive, we pose them in the context of causal models, because our causal theories have implications for the level of one variable, or the correlation between two others.

Causal models include a set of outcome variables that may be functions of baseline characteristics and the effects of treatments. These treatments might be delivered naturally by the world or may be set by researchers. The values that an outcome variable takes depending on the level of a treatment are called *potential outcomes*. In the simplest case of a binary treatment, the treated potential outcome arises when a unit is treated; the untreated potential outcome when it is not. The causal effect of a particular treatment is usually defined as the difference between the treated and untreated potential outcomes.

2.1.1.2 *M* is a set.

On Figure 2.1, we describe M as the “the worlds you’ll consider.” The reason for this is that we are uncertain about how the world works. We don’t know the “right” model. For this reason we have to think through how our design will play out under different possible models including ones we think likely and ones we think less likely. For instance, the correlation between two variables might be large and positive, but it could just as well be zero. We might believe that, conditional on some background variables, a treatment has been as-if randomly assigned by the world — but we might be wrong about that too. In the figure we use m to denote the “right” model, or the actual, unknown, event generating process. We do not have access to m , but our hope is that m is sufficiently well-represented in M so that we can reasonably imagine what will happen when our design is applied in the real world.

How can you construct a sufficiently varied model of the world? For this difficult piece of theoretical work, you can draw on existing data, such as baseline surveys, or on new information gathered from pilot studies. Reducing uncertainty over the set of possible models is a core purpose of theoretical reflection, literature review, meta-analysis, and formative research. If there are important known features about your context it generally makes sense to include them in M .

Examples of models

1. Contact theory: When two members of different groups come into contact under specific conditions, they learn more about each other, which reduces prejudice, which in turn reduces discrimination.
2. Prisoner’s dilemma. When facing a collective problem, each of two people will choose non-cooperative actions independent of what the other will do.
3. Health intervention with externalities. When individuals receive deworming medication, school attendance rates increase for them and for their neighbors, leading to improved labor market outcomes in the long run.

2.1.2 Inquiry

The inquiry is a question stated in terms of the model. For example, the inquiry might be the average causal effect of one variable on another, the descriptive distribution of a third variable, or a prediction about the value of a variable in the future. We refer to “the” inquiry when talking about the main research question. But our theories are rich, so we may seek to learn about many inquiries in a single research study.

Many people use the word “estimand” to refer to an inquiry, and we do too when talking about research informally. When we are formally describing research designs, however, we distinguish between inquiries and estimands and

Figure 2.1 shows why. The inquiry I is the function that operates on the events generated by the world m . The estimand is the value of that function: a_m . In other words, we use “inquiry” to refer to the question and “estimand” to refer to the answer to the question.

Inquiries are defined with respect to units, conditions, and outcomes: they are summaries of outcomes of units in or across conditions. Inquiries may be causal, as in the sample average treatment effect (SATE). The SATE is the average difference in the outcome variable across the treatment condition and the control condition among units in a sample. Inquiries may also be descriptive, as in a population average of an outcome. While it may seem that descriptive inquiries do not involve conditions, they always do, since description of outcomes must take place under a particular set of circumstances, often set by the world and not the researcher.

Figure 2.1 shows that when I is applied to a model m , it produces an answer a^m . This set of relationships forces discipline on both M and I : I needs to be able to return an answer using information available from M and in turn M needs to provide enough information so that I can do its job.

We might think of the model and inquiry as forming the theoretical half of a research design. Together, they describe notional processes and quantities that we don’t observe directly. The data strategy and the answer strategy form the empirical half of the design, mirroring the theoretical half.

Examples of inquiries

1. What proportion of voters live with limited exposure to voters from another party in their neighborhood?
2. Does gaining political office make divorce more likely?
3. What types of people will benefit most from a vaccine?

2.1.3 Data strategy

The data strategy is the full set of procedures we use to gather information from the world. The three basic elements of data strategies parallel the three features of inquiries: units are sampled, conditions are assigned, and variables are measured.

All data strategies involve sampling in the sense that no empirical strategy is comprehensive: some units are sampled into the study and some units aren’t. Seemingly comprehensive research designs like a population census have a sampling strategy in that they don’t sample respondents in different years or different countries.

Assignment procedures describe how researchers generate variation in the world. If you ask some subjects one question, but other subjects a different question, you’ve generated variation on the basis of an assignment procedure.

We think of assignment procedures most often when they are randomized, as in a randomized experiment. Yet other kinds of research designs draw on assignment procedures that are not randomized, as in a pre-post design.

Measurement procedures are the ways in which researchers reduce the complex and multidimensional social world into a parsimonious set of data. These data need not be quantitative data in the sense of being numbers or values on a pre-defined scale; qualitative data are data too. Measurement is the vexing but necessary reduction of reality to a few choice representations. Measured values always contain measurement error, because this reduction is hard.

Figure 2.1 shows how the data strategy is applied to both the imagined worlds in M and to the real world. In practice, the application of D to the real world (m) might look quite different to the application of D to the worlds you imagine in M . We represent it in this way, however, to encourage you to make the representation of your data strategy as realistic as possible. Include in it not just the idealized elements, but also the challenges you might encounter in an uncooperative world such as nonresponse and noncompliance.

Examples of data strategies

Sampling procedures.

1. Random digit dial sampling of 500 voters in the Netherlands
2. Respondent-driven sampling of people who are HIV positive, starting from a sample of HIV-positive individuals
3. “Mall intercept” convenience sampling of men and women present at the mall on a Saturday

Treatment assignment procedures.

4. Random assignment of free legal assistance intervention for detainees held in illegal pretrial detention
5. Nature’s assignment of the sex of a child

Measurement procedures.

6. Voting behavior gathered from survey responses
7. Administrative data indicating voter registration
8. Measurement of stress using Cortisol readings

2.1.4 Answer strategy

The answer strategy is how we summarize the data produced by the data strategy. Just like the inquiry summarizes a part of the model, the answer strategy summarizes a part of the data. We can’t just “let the data speak” because complex, multidimensional datasets don’t speak for themselves — they need to be

summarized and explained. Answer strategies are the procedures we follow to do so.

Answer strategies are functions that take in data and return answers. For some research designs, this is a literal function like the R function `lm_robust` that estimates an ordinary least squares (OLS) regression with robust standard errors. For some research designs, the function is embodied by the researchers themselves when they read documents and summarize their meanings in a case study.

The answer strategy is more than the choice of an estimator. It includes the full set of procedures that begins with cleaning the dataset and ends with answers in words, tables, and graphs. These activities include data cleaning, data transformation, estimation, plotting, and interpretation. Not only do we define our choice of OLS as the estimator, we also specify that we will focus attention on a particular coefficient estimate, assess uncertainty using a 95% confidence interval, and construct a coefficient plot to visualize the inference. The answer strategy also includes all of the if-then procedures that researchers implicitly or explicitly take depending on initial results and features of the data. For example, in a stepwise regression procedure, the answer strategy is not the final regression that results from iterative model selection, but that whole procedure itself.

D and A impose a discipline on each other in the same way as we saw with M and I . Just like the model needs to provide the events that are summarized by the inquiry, the data strategy needs to provide the data that are summarized by the answer strategy. Declaring each of these parts in detail reveals the dependencies across the design elements.

A and I also enjoy a tight connection stemming from the more general parallelism between (M, I) and (D, A) . We elaborate the principle of parallel answer strategies in the next chapter and in Section 9.3.2. For now though we highlight that the nature of the question you ask can determine whether the answer strategy can does or does not require inference. If the question requires inference, then the design requires an inferential strategy.

Table @??tab:questiontypes02 shows three types of questions—descriptive, causal, general—and examples in which these questions do or do not require inference. Some descriptive questions can in principle be addressed without inference from the measurement to the thing being measured (though we grant some philosophers would doubt even this). But descriptive questions typically do require an inferential strategy that gives us confidence that our measures align with the quantities we care about. Causal questions *always* require inference. This is what is meant by the “fundamental problem of causal inference”: you cannot see causal effects, you *have* to infer them. Last, some questions are about general claims. In theory, a general claim could be answered without inference through exhaustive measurement. For instance, “Are all swans white” could be answered without inference if we find one nonwhite swan. But

general claims generally do require inference, including claims whose scope extends into the future. This type of inference is sometimes called “inductive inference” (Fisher, 1935), but we’ll refer to it as “generalization inference.”. Finally, answering some questions requires facing all three types of inferential challenge at once, for instance general claims about the causal effects of a treatment on latent outcomes.

Table 2.1: Kinds of inquiries

Inquiry type	Answerable without inference	Requires inference	Sample inferential strategy	Challenge
Descriptive inquiry	The winner got 7 votes	The president was angry	Seek observable implications	Descriptive inference
Causal inquiry	No examples possible	Human activity caused global warming	Random assignment, instrumental variables, controlled comparisons	Causal inference
General inquiry	All current British MPs are men	First past the post systems usually have two parties	Sample from a population, sample from history, make theory dependent claim	Generalization inference

Figure 2.1 shows how the same answer strategy A gets applied both to the expected data d and also to the data that you will ultimately gather d . We know that in practice, however, the A applied to the real data differs somewhat from the A applied to the data we plan for via simulation. Designs sometimes drift in response to data, but too much drift and the inferences we draw can become misleading. The *MIDA* framework encourages you to think through what the real data will actually look like, and adjust A accordingly *before* data strategies are implemented.

Examples of answer strategies

1. Multilevel modeling and poststratification
2. Bayesian process tracing
3. Difference-in-means estimation

2.2 Declaration, diagnosis, redesign

With the core elements of a design described, we are ready to work through the process of declaration, diagnosis, and redesign.

2.2.1 Declaration

Declaring a design entails figuring out which parts of your design belong in M , I , D , and A . The declaration process can be a challenge because mapping your ideas and excitement about your project into *MIDA* is not always straightforward, but it is rewarding. When you can express your research design in terms of these four components, you are newly able to think about its properties.

Designs can be declared in words, but declarations often become much more specific when carried out in code. You can declare a design in any statistical programming language: Stata, R, Python, Julia, SPSS, SAS, Mathematica, or many others. Design declaration is even possible – though somewhat awkward – in Excel. We wrote the companion software, *DeclareDesign*, in R because of the availability of other useful tools in R and because it is free, open-source, and high-quality. We have designed the book so that you can read it even if you do not use R, but you will have to translate the code into your own language of choice. On our Web site, we have pointers for how you might declare designs in Stata, Python, and Excel. In addition, we link to a “Design wizard” that lets you declare and diagnose variations of standard designs via a point-and-click web interface. Chapter 4 provides an introduction to *DeclareDesign* in R.

2.2.2 Diagnosis

Once you’ve declared your design, you can diagnose it. Design diagnosis is the process of simulating your research design in order to understand the range of ways the study could turn out. Each run of the design comes out differently because different units are sampled, or the randomization allocated different units to treatment, or outcomes were measured with different error. We let computers do the simulations for us because imagining the full set of possibilities is – to put it mildly – cognitively demanding.

Diagnosis is the process of assessing the properties of designs, and represents an opportunity to write down what would make the study a success. For a long time, researchers have classified studies as successful or not based on statistical significance. If significant, the study “worked”; if not, it is a failed “null.” Accordingly, statistical power (the probability of a statistically significant result) has been the most front-of-mind design property when researchers plan studies. As we learn more about the pathologies of relying on statistical significance, we learn that features beyond power are more important. For example, the “credibility revolution” throughout the social sciences has trained a laser-like focus on the bias that may result from omitted or “lurking” variables.

Design diagnosis relies on two new concepts: diagnostic statistics and diagnosands.

A “diagnostic statistic” is a summary statistic generated from a single simulation of a design. For example, the statistic e refers to the difference between the estimate and the estimand. The statistic s refers to whether the estimate was

deemed statistically significant at the 0.05 level.

A “diagnosand” is a summary of the distribution of a diagnostic statistic across many simulations of the design. The bias diagnosand is defined as the average value of the e statistic and the power diagnosand is defined as the average value of the s statistic. Other diagnosands include quantities like root-mean-squared-error (RMSE), Type I and Type II error rates, whether any subjects were harmed, and average cost. We describe these diagnosands in much more detail in Chapter 10.

One especially important diagnosand is the “success rate,” which is the average value of the “success” diagnostic statistic. As the researcher, you get to decide what would make your study a success. What matters most in your research scenario? Is it statistical significance? If so, optimize your design with respect to power. Is what matters most whether the answer has the correct sign or not? Then diagnose how frequently your answer strategy yields an answer with the same sign as your inquiry. Diagnosis involves articulating what would make your study a success and then figuring out, through simulation, how often you obtain that success. Success is often a multidimensional aggregation of diagnosands, such as the joint achievement of high statistical power, manageable costs, and low ethical harms.

We diagnose studies over the range of possibilities in the model, since we want to learn the value of diagnosands under many possible scenarios. A clear example of this is the power diagnosand over many possible values of the true effect size. For each effect size that we entertain in the model, we can calculate statistical power. The minimum detectable effect size is a summary of this power curve, usually defined as the smallest effect size at which the design reaches 80% statistical power. This idea extends well beyond statistical power. Whatever the set of important diagnosands, we want to ensure that our design performs well across all model possibilities.

Computer simulation is not the only way to do design diagnosis. Designs can be declared in writing or mathematical notation and then diagnosed using analytic formulas. Enormous theoretical progress in the study of research design has been made with this approach. Methodologists across the social sciences have described diagnosands such as bias, power, and root-mean-squared-error for large classes of designs. Not only can this work provide closed-form solutions for many diagnosands, it can also yield insights about the pitfalls to watch out for when constructing similar designs. That said, pen-and-paper diagnosis is challenging for the majority of social science research designs, first because many designs as actually implemented have idiosyncratic features that are hard to incorporate and second because the analytic formulas for many diagnosands have not yet been worked out by statisticians. For this reason, we usually depend on simulation.

Even when using simulation, design diagnosis doesn’t solve every problem and like any tool, can be misused. We outline two main concerns. The first is the

worry that the diagnoses are plain wrong. Given that design declaration includes conjectures about the world, it is possible to choose inputs such that a design passes any diagnostic test set for it. For instance, a simulation-based claim of unbiasedness that incorporates all features of a design is still only good with respect to the precise conditions of the simulation. In contrast, analytic results, when available, may extend over general classes of designs. Still worse, simulation parameters might be chosen opportunistically. A power analysis may be useless if implausible parameters are chosen to raise power artificially. While our framework may encourage more principled declarations, it does not guarantee good practice. As ever, garbage-in, garbage-out. The second concern is the risk that research may be evaluated on the basis of a narrow or inappropriate set of diagnosands. Statistical power is often invoked as a key design feature, but well-powered studies that are biased are of little theoretical use. The importance of particular diagnosands can depend on the values of others in complex ways, so researchers should take care to evaluate their studies along many dimensions.

2.2.3 Redesign

Once your design has been declared, and you have diagnosed it with respect to the most important diagnosands, the last step is redesign.

Redesign entails fine-tuning features of the data and answer strategies to understand how they change your diagnosands. Most diagnosands depend on features of the data strategy. We can redesign the study by varying the sample size to determine how big it needs to be to achieve a target diagnosand: 90% power, say, or an RMSE of 0.02. We could also vary an aspect of the answer strategy, for example, the choice of covariates used to adjust a regression model. Sometimes the changes to the data and answer strategies interact. For example, if we want to use covariates that increase the precision of the estimates in the answer strategy, we have to collect that information as a part of the data strategy. The redesign question now becomes, is it better to collect pretreatment information from all subjects or is the money better spent on increasing the total number of subjects and only measuring posttreatment?

The redesign process is mainly about optimizing research designs given ethical, logistical, and financial constraints. If diagnosands such as total harm to subjects, total researcher hours, or total project cost exceed acceptable levels, the design is not feasible. We want to choose the best design we can among the feasible set. If the designs remaining in the feasible set are underpowered, biased, or are otherwise scientifically inadequate, the project may need to be abandoned.

In our experience, it's during the redesign process that designs become *simpler*. We learn that our experiment has too many arms or that the expected level of heterogeneity is too small to be detected by our design. We learn that in our theoretical excitement, we've built a design with too many bells and too many

whistles. Some of the complexity needs to be cut, or the whole design will be a muddle. The upshot of many redesign sessions is that our designs pose fewer questions, but obtain better answers.

2.3 Example: October surprise

Political pollsters forecast the outcomes of elections by taking samples of eligible voters and asking which candidate they will vote for. A key data strategy choice pollsters have to make is how best to spend their limited budget. They would like to forecast the outcome of the election far in advance, so one possibility is to draw one large sample a few weeks before the the election, ask voters which candidate they prefer, then forecast the winner of the election on the basis of which candidate most subjects prefer.

But it's possible that public preferences over the candidates change in the run-up to the election. American national elections are held in November, so a late-breaking scandal or event that shakes up the race is called an "October surprise." Even a large poll conducted a few weeks ahead of the election might miss the October surprise, so pollster might want to consider an alternative data strategy: spreading their limited resources over three smaller polls over the last few weeks of the election instead one large poll.

In this example, we show how the declare, diagnose, redesign algorithm can help us think through the designs that will help us maximize one main diagnosis: the frequency of making the correct election prediction.

2.3.1 A "steady race" design

Here we declare a "steady race" design.

Suppose we believe the race is steady. We include this belief in the model by stipulating candidate A is favored by 51 percent of the population and candidate B is favored by 49%. We think the state of the race will stay steady through until the election, so the probability a respondent at time 1, time 2, and time 3 prefers candidate A remains constant at 51 percent.

The inquiry is the final vote share in the actual election at time 4. We imagine that the actual result will be centered on 51 percent, sometimes a little higher, sometimes a little lower, with a standard deviation of 1 percentage point. We imagine that this variation is not due to changes in the preferences of the electorate, but instead idiosyncratic features of election day, like the weather or voting machine problems. Because the actual result is a random draw from this distribution, candidate A wins more often than candidate B, but not always.

In the data strategy, we conduct a poll at time 1, measuring preferences as they stood a few weeks before the election. The answer strategy takes the mean of the observed data using a neat regression trick: regressing the outcome on a constant ($Y_{obs} \sim 1$) returns the sample mean.

Table 2.2: Diagnosis of the steady race design

Correct Call Rate
0.68

We diagnose this design with respect to the “Correct Call Rate.” Under this model, using this data and answer strategy, how frequently will we make the right decision? Since the sample consists of only 1,000 people, estimating which side of 50% we’re on is no easy task. Under these conditions, we make the right call about 68% of the time.

```
diagnosands <-
  declare_diagnosands(
    correct_call_rate = mean((estimate > 0.5) == (estimand > 0.5))
  )

diagnosis <-
  diagnose_design(design = steady_race_design,
                  diagnosands = diagnosands)
```

2.3.2 An “October surprise” design

But what about under other circumstances? Let’s imagine an alternative model: the October surprise. Now opinion is shifting beneath our feet. Candidate A is losing ground fast. At time 1, the fraction preferring A was 51%, but at time 2 it’s 50%, at time 3 it’s 49%. What we care about – the inquiry – is the fraction preferring candidate A at time 4, the time of the election, when opinions about Candidate A will have arrived at 48%. Now if we were to run the election over and over, Candidate B would win much more often than Candidate A.

If we’re concerned that opinion is shifting, we might want to spread our 1,000 subjects over all three periods in the run up to the election to track the shift. This alternative data strategy randomly samples a third of units at time 1, a third at time 2, and a third at time 3. In the answer strategy, we run an ordinary least squares regression of the outcome on time in order to estimate average opinion trends over time. We use that slope to predict average opinion at time 4, the time the election.

Now we diagnose the October surprise design with respect the same “Correct Call Rate.” The diagnosis shows that with this new design, the correct call rate is still 68%.

Table 2.3: Diagnosis of the October surprise design

Correct Call Rate
0.68

```
diagnosis <-
  diagnose_design(design = october_surprise_design,
  diagnosands = diagnosands)
```

2.3.3 Choosing between empirical designs

If the steady race theory is right and we do a single poll, we make the right call about 68% of the time. And if the October surprise theory is right, and we do three polls, we also make the right call about 68% of the time. We don't know which theory is right, so which empirical strategy should we implement – one poll or three?

To answer this question, we engage in redesign. We redesign theoretical beliefs over empirical strategies, considering what happens when we apply the single poll design to the October surprise theory, or the three poll design to the tight race theory. Redesign can be accomplished by putting all four designs into `diagnose_design`:

```
diagnosis <-
  diagnose_design(
    design_1 = steady_race + single_poll,
    design_2 = steady_race + three_polls,
    design_3 = october_surprise + single_poll,
    design_4 = october_surprise + three_polls,
    diagnosands = diagnosands
  )
```

Figure 2.2 shows that each empirical design does best when it matches the corresponding theoretical model of the race, which is to be expected. But which empirical design does better under the *other* theory? If the October surprise hits, the first empirical approach is terrible. It only makes the right call 25% of the time, because it relies on preferences measured before the big scandal changes political attitudes. How does the three poll design fare under the steady race theory? It's not as strong as the one poll design in this setting, but it does still make the correct call 56% of the time.

If we are *uncertain* about whether there will be an October surprise (and we should be; surprising things happen), we have to weigh the upsides and downsides of each approach. If we think both theoretical models are equally likely, then we should conduct three polls, since that strategy does well when there is a surprise, and moderately well when there isn't.

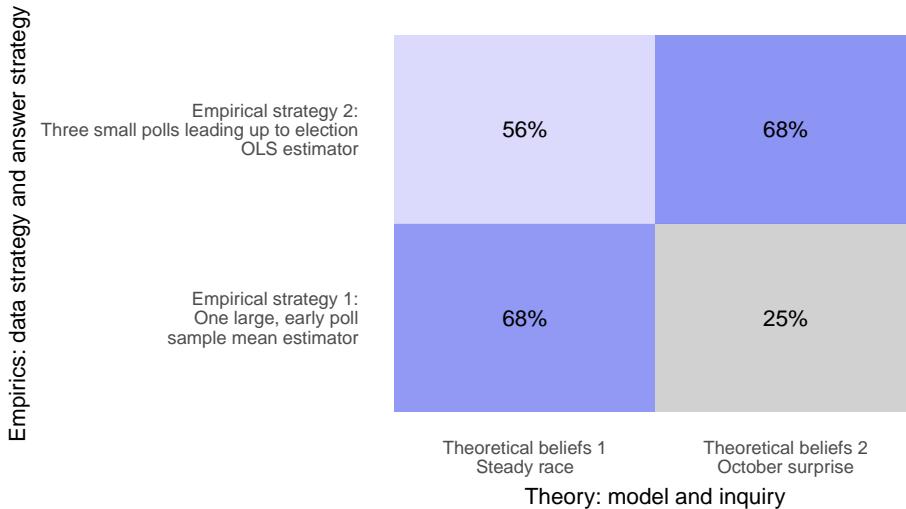


Figure 2.2: Correct call rates of four designs

This example illustrates the value of entertaining many theoretical models when declaring research designs. Here we found that neither empirical strategy beats the other across both theoretical settings, but that one did well under more circumstances. We could entertain many more models than these. We could vary how close the race is, for example, how many candidates are running, or the slope of the trend. And we could of course imagine much more complex data and answer strategies that would outperform either of these two under a much wider range of circumstances than we considered. Elaborations like these will be explored in detail in Part II, then applied across common designs in Part III.

2.4 Putting designs to use

The two pillars of our approach are the language for describing research designs (*MIDA*) and the algorithm for selecting high-quality designs (declare, diagnose, redesign). Together, these two ideas can shape research design decisions throughout the lifecycle of a project. The full set of implications is drawn out in Part IV but we emphasize the most important ones here.

Broadly speaking, the lifecycle of an empirical research project has four phases: brainstorming, planning, realization, and integration. Planning, realization,

and integration describe what happens before, during, and after the implementation of a research design. The inclusion of a pre-phase (brainstorming) reflects the idea that research doesn't just begin with "planning," it has to begin with some spark of inspiration.

The inspiration for a good research project can come from many sources: frustration with an article you're reading, a golden opportunity with a potential research partner, a conversation with a colleague (or adversary!). The spark of an idea might be some bit of a model, perhaps an inquiry in particular, maybe a portion of a data strategy, or just an itch to apply a new answer strategy you learned about. Wherever that kernel of an idea starts, the purpose of brainstorming is to develop each element (*M*, *I*, *D*, and *A*) such that the design becomes a coherent whole, *MIDA*.

After an idea has been fleshed out sufficiently, it's time to start planning. Planning entails some or all of the following steps, depending on the design: conducting an ethical review, seeking IRB approval, gathering criticism from colleagues and mentors, running pilot studies, and preparing preanalysis documents. The design as encapsulated by *MIDA* will go through many iterations and refinements during this period. Planning is the time when frequent re-application of the declare, diagnose, redesign algorithm will pay the highest dividends. How should you investigate the ethics of a study? By casting the ethical costs and benefits as diagnosands. How should you respond to criticism, constructive or not? By re-interpreting the feedback in terms of *M*, *I*, *D*, and *A*. How can you convince funders and partners that your research project is worth investing in? By credibly communicating your study's diagnosands: its statistical power, its unbiasedness, its high chance of success, however the partner or funder defines it. What belongs in a pre-analysis plan? You guessed it – a specification of the model, inquiry, data strategy, and answer strategy.

Realization is the phase of research in which all those plans are executed. You implement your data strategy in order to gather information from the world. Once that's done, you follow your answer strategy in order to finally generate answers to the inquiry. Of course, that's only if things go exactly according to plan, which has never happened once in our own careers. Survey questions don't work as we imagine, partner organizations may lose interest in your study, subjects move or become otherwise unreachable. A critic or a reviewer may insist you change your answer strategy, or may think a different inquiry altogether is theoretically appropriate. You may yourself change how you think of the design as you embark on writing up the research project. It is inevitable that some features of *MIDA* will change during the realization phase. Some design changes have very bad properties, like sifting through the data ex-post, finding a statistically significant result, then back-fitting a new *M* and a new *I* to match the new *A*. Indeed, if we declare and diagnose this actual answer strategy (sifting through data ex-post), we can show through design diagnosis that it is badly biased for any of the inquiries it could end up choosing. Other changes made along the way may help the design quite a bit. If the planned design did

not include covariate adjustment, but a friendly critic suggests adjusting for the pre-treatment measure of the outcome, the “standard error” diagnosis might drop nicely. The point here is that design changes during the implementation process, whether necessitated by unforeseen logistical constraints or required by the review process, can be understood using in terms of M , I , D , and A by reconciling the planned design with the design as implemented.

A happy realization phase concludes with the publication of results. But the research design lifecycle is not finished: the study and its results must be integrated into the broader community of scientists, decisionmakers, and the public. Studies should be archived, along with design information, to prepare for reanalysis. Future scholars may well want to reanalyze your data in order to learn more than is represented in the published article or book. Good reanalysis of study data requires a full understanding of the design as implemented, so archiving design information along with code and data is critical. Not only may your design be reanalyzed, it may also be replicated with fresh data. Ensuring that replication studies answer the same theoretical questions as original studies requires explicit design information without which replicators and original study authors may simply talk past one another. Indeed, as your study is integrated into the scientific literature and beyond, you should anticipate disagreement over your claims. Resolving disputes is very difficult if parties do not share a common understanding of the research design. Finally, you should anticipate that your results will be formally synthesized with others’ work via meta-analysis. Meta-analysts need design information in order to be sure they aren’t inappropriately mixing together studies that ask different questions or answer them too poorly to be of use.

24

What is a research design?

2.4

Chapter 3

Research design principles

The declare, diagnose, redesign framework suggests a set of eleven principles that can guide the design process. Not all principles are equally important in all cases but we think all are worth giving consideration when developing and assessing a design. This section offers succinct discussions of the eleven principles. We will discuss the implications of these principles for specific design choices throughout the book, which is just to say that everything we mean to communicate with these principles may not be immediately obvious on a first read.

Design principles

1. Design early
2. Design often
3. Entertain many models
4. Select answerable inquiries
5. Include strategies for descriptive, causal, and inductive inference
6. Declare data and answer strategies as functions
7. Seek M:I::D:A parallelism
8. Specify diagnosands as a function of research goals
9. Diagnose to break designs
10. Diagnose whole designs
11. Design to share

Principle 3.1. Design early

Designing an empirical project entails declaring, diagnosing, and redesigning the components of a research design: its model, inquiry, data strategy, and answer strategy. The design phase yields the biggest gains when we design early. By frontloading design decisions, we can learn about the properties of a design while there is still time to improve them. Once data strategies are implemented – units sampled, treatments assigned, and outcomes measured – there's no going

back. While applying the answer strategy to the revealed dataset, you might well wish you'd gathered data differently, or asked different questions. Post-hoc, we always wish our previous selves had planned ahead.

The deeper reason than regret for designing early is that the declaration, diagnosis, and redesign process inevitably changes designs, almost always for the better. Revealing how each of the four design elements are interconnected yields improvements to each. If the answer strategy and inquiry are mismatched, the designer faces a choice to change one or the other. If the units sampled in the data strategy are theoretically inappropriate, alternative participants might be selected. Models may reveal assumptions that require defense through additional data collection. Better inquiries, with greater theoretical leverage over the model, may be identified. Inquiries that cannot be answered may be replaced.

Designs are fine-tuned through redesign, which entails diagnosing across the feasible combinations of design parameters and selecting from the best-performing combinations. Redesign usually focuses on envisioning changes to the data strategy: alternative sampling procedures, assignment probabilities, or measurement techniques. Redesign can also consider changes to the answer strategy such as variations to the estimation or inferential procedures. These choices are almost always better made before any data are collected or analyzed.¹

A common objection to planning ahead is that inevitably, plans change. Empirical researchers encounter empirical problems: missing data, archival documents that cannot be traced, noncompliance with treatment assignments, evidence of spillovers, and difficulties recontacting subjects. Insofar as these are predictable problems, it can be useful to think of them as *parts* of your design not *deviations* from your design. Answer strategies can be developed that anticipate these problems, and account for them, including if-then plans for handling each potential problem. More fundamentally, anticipated failures themselves can be included in your model so that you can diagnose the properties of different strategies, in advance, given risks of different kinds.

Principle 3.2. Design often

Designing early does not mean being inflexible. In practice, unforeseen circumstances may change the set of feasible data and answer strategies. Implementation failures due to nonresponse, noncompliance, spillovers, inability to link datasets, funding contractions or logistical errors are common ways the set of feasible designs might contract. The set of feasible designs might expand if new data sources are discovered, additional funding is secured, or if you learn about a new piece of software. Whether the set expands or contracts, we should once again declare, diagnose, and redesign given the new realities.

¹See Principle 3.6 for why this is not in tension with the common desire to let data help make choices like model specification.

“Designing often” usually happens in the middle of implementation. The output of that process is usually a modification to the data strategy and any compensating changes to the answer strategy. Sometimes, the unexpected event necessitates a change to the inquiry itself. We need to diagnose over the new feasible set of designs in order to make a new best choice.

The principle that we should design often also extends beyond the implementation. A colleague may suggest an alternative answer strategy; whether or not the suggestion is a good one is a design question that is often best settled through explicit declaration and diagnosis of the proposed alternative. A critic may charge that the model is theoretically ill-specified. Assessing the consequences of this contention requires us to diagnose the alternative designs, holding the inquiry, data strategy, and answer strategy constant while varying the models. Designing often means engaging in declaration, diagnosis, and redesign all throughout the research design lifecycle.

Principle 3.3. Entertain many models

When we design a research study, we have in mind a model of how the world works. But really your model is not just one model, it's a family of possibilities. We think a set of variables are related, but we are uncertain in what ways and how closely related they are. Our family of possibilities includes plausible ranges for the parameters about which we are uncertain. The principle that we should entertain many models suggests that designers should expand the set of possible models they have in mind when deciding how to conduct research. We want to entertain many models because we want to be sure that the true model – how the work really works – is represented in the set we consider.

One way to entertain many models is to explicitly consider threats to inference. Randomized experiments can generate unbiased estimates of average causal effects under some models, but not others. Experiments can be biased for a population average causal effect if the sample is not representative, if the assignment affects outcomes via paths that are not mediated by the treatment, if the outcomes of one unit depend on the treatment status of others, and in many other settings besides. Threats to inference like these represent possible models in the family of possibilities that we entertain.

When we entertain many models, we learn the circumstances in which our designs perform well and poorly. Regression-like approaches for observational causal inference work well when the key “selection-on-observables” assumption holds and less well otherwise. “Doubly-robust” estimation is so named because it performs well when we correctly guess the outcome model, the selection model, or both, but poorly when we are wrong about both. “Design-based,” “nonparametric,” or “agnostic” approaches to inference enjoy the property that they often work well under a larger class of models than “model-based” approaches, since model-based approaches rely on the assumption that the stipulated model is the correct one.

The core idea is that your design gets stronger if it continues to perform well in

many circumstances. Entertaining many models leads us to choose these more robust empirical designs.

Principle 3.4. Select answerable inquiries

This principle has two components.

First, you should *have* an inquiry. Oddly, it is possible to carry out data analysis — for instance, running a regression of Y on X — and getting something that looks like an answer without specifying any question in particular.

Second, your inquiry should be answerable. That's trickier than it sounds. We can think of being answerable in theory and in practice.

An inquiry is answerable “in theory” if you can write down a model such that, if that model were the true model, and you knew the features of the model, you could answer the question. Simple sounding questions like “Did Germany cause the Second World War?” or “did New Zealand do well against Covid-19 because Prime Minister Jacinda Ardern was a woman?” can turn out to be difficult to ask and answer. This, we think, can give a hint to when a question is poorly posed. We must be able to describe *some* world such that the inquiry has a precise answer. In our framework, an inquiry is answerable in theory if for some m and I , $I(m) = a_m$ exists.

An inquiry is answerable “in practice” if the inquiry could be answered with data generated by a feasible data strategy, even if difficult-to-execute. That is, an inquiry is answerable in practice if for some D and A , $A(D() = d) = a_d$ is an estimate of a_m .

Selecting answerable inquiries means choosing inquiries that are answerable both in theory and in practice. Some inquiries that are answerable in theory are not answerable in practice, except under knife-edge subset of models.

Principle 3.5. Include strategies for descriptive, causal, and inductive inference

Empirical research designs can face three inferential challenges. Research designs that seek to draw causal inferences encounter the fundamental problem of causal inference that we can observe a unit in its treated state or in its untreated state, but not both. Designs that seek to measure latent variables using measured variables face a challenge of descriptive inference that measurements are different from the concepts they measure. Designs that seek to draw inferences about non-study units from study units face the challenge of general inference that non-study units might be different from study units.

These three inferential challenges can all be thought of as missing data problems. Some information we would like to have is out of reach. For causal inference, we would like to observe counterfactual outcomes, but they are not observable. We can observe $Y_i(\text{treated} = 1)$ or $Y_i(\text{treated} = 0)$ but not both. For descriptive inferences, we would like to observe latent values, but we have to content ourselves with measurements. We can observe Y_i , but not Y_i . For generalization, we would like to observe nonstudy units, but by definition, they

are not available for observation. We can observe $Y_i(\text{sampled} = 1)$ but we can never observe $Y_j(\text{sampled} = 0)$.

Confronting these problems requires strong research designs with inferential strategies targeted to the inferential challenges you face. Many such strategies exist. For instance, even though we can't observe the counterfactual outcome for any particular unit, we can design studies to yield good estimates of *average* treated and untreated outcomes, from which we can construct estimates of average causal effects. Even though we don't observe latent variables directly, we can sometimes triangulate their values through the aggregation of multiple measures or detailed understanding of the measurement process. Even though we don't observe any nonstudy units in particular, we can employ sampling designs that license stronger inferences about the average outcomes of some nonstudy units. For all of these strategies there are analytic results we can turn to to justify the strategies and in all cases we can demonstrate that our design is working well at least in the model set we consider.

Principle 3.6. Declare data and answer strategies as functions

The data strategy is the function that, when applied to the real world, produces the realized dataset. The answer strategy is the function that, when applied to the realized data set, yields the empirical answer to the inquiry.

The distinction between the data strategy versus the realized data is important. The realized data — what we will ultimately download onto our computer — represent just one draw of the data that can be generated by a data strategy. Under different randomizations, different units would be treated or sampled into the study, each resulting in a different realized dataset. Even with nonrandomized data strategies, we can think of the observed data as being one draw from an underlying event generating process that could have been different.

Similarly, we have to distinguish between the answer strategy and the answer that is produced. If the realized data were different, the empirical answer produced would be different. Thinking of the answer strategy as a function underlines how the empirical answer is just an estimate of the truth, not the truth itself. In general, we can't know if a_d equals a_m exactly.

Critically, the data and answer functions should be able to provide outputs for a wide variety of inputs: they should have a wide domain. In the case of the data strategy we should be able to envision what data will be produced for a variety of worlds, including worlds we have not imagined. In the case of the answer strategy we should be able to envision what answer we will get for different types of data that we might find, including data we have not imagined. Often setting up functions in this way requires the functions to operate as *procedures* that are responsive to inputs.

Adaptive random assignment schemes are an example of a data strategy as a procedure. In each round more effective treatments are assigned to more and more units — a change to the assignment probabilities each round. The decision

whether to include a control in a regression might depend on a the basis of how well the regression specification fits the data. A three-step procedure — fit regressions, assess fit, report coefficient from best-fitting specification — is what should then be declared as hte answer strategy.

The advantage of specifying flexible functions that can handle diverse inputs is that diagnosis takes account of the full procedure and not just the final result. A diagnosis of a design in which you include only the final specification that got used will be wrong because it does not capture properly the distribution if what would have happened under different circumstances.

Principle 3.7. Seek $M:I:D:A$ parallelism

The model and inquiry form the theoretical half of a research design and the data and answer strategies form the empirical half. Designs in which the relationship of M to I is parallel to the relationship of D to A are often strong, precisely because of the tight correspondence across the theoretical and empirical halves. Some intuition for this idea can be read off our formalism that the theoretical answer can be written $I(m) = a_m$ and the empirical answer as $A(D = d) = a_d$. In words, if the data strategy produces data that are “like” the events generated by a model, and if A is like I , the estimate will be like the estimand. When the theoretical and empirical halves of the design are parallel, then we can write the design as an analogy: M is to I as D is to A .

This idea is a version of the “plug-in principle”: under many data strategies, we can “plug-in” the inquiry for the answer strategy. For example if we are interested in estimating the population mean, wedraw a sample from the population and estimate it using the sample mean estimator.

Parallelism could break down if the data strategy does not produce data like the model produces events. When data strategies introduce distortions, we can make compensating changes to the answer strategy. We restore parallelism by seeking an A' such that $A(D = d) \approx A'(D = d)$. This idea underpins the maxim “analyze as you randomize” (Fisher, 1937): estimators should account for differential probabilities of assignment and other known features of the randomization procedure. Even outside randomized studies, parallelism is served when answer strategies respect known features of the data strategies.

Principle 3.8. Specify diagnosands as a function of research goals

When designing research, we should give careful thought to our diagnosands, the criteria by which we evaluate the qualities of candidate designs. Too often, researchers focus on a narrow set of diagnosands, and often consider them in isolation. Is the estimator unbiased? Do I have statistical power? The evaluation of a design nearly always requires balancing multiple criteria: scientific precision, logistical constraints, policy goals, as well ethical considerations. Each of these goals can be specified as a function of an implementation of the design. The cost is a function that translates the number of units and the amount of time it took to collect and analyze data into a total cost. Scientific goals may

be represented in a number of ways, such as the root mean-squared error or statistical power or most directly the amount of learning between before and after the study was conducted. Ethical goals may also be translated into functions. An ethical diagnosand might be the number of minutes of time taken from participants of the study or whether any participants were harmed.

A diagnosis of designs across multiple criteria provides us with a multidimensional value statement of each design. We then can select the best feasible design among them. Specifying diagnosands intentionally forces us to provide a weighting scheme between possibly competing ethical, logistical, and scientific values. This weighting scheme is important to understand explicitly because finding the best design in this high-dimensional space is difficult and separately evaluating designs on each dimension and then weighting across dimensions can help.

Principle 3.9. Diagnose to break designs

A corollary principle to “entertain many models” is that we should know the models under which our design performs well and those under which it performs poorly. We want to diagnose over many models to find where designs break.

Our design might assume for instance that one variable is not affected by another variable and the validity of our answer might depend on the extent to which this is true. A design that contains a set of models that include violations of this assumption can be used to assess the extent to which the assumption matters, how bad a violation has to be to produce misleading results of consequence, and what types of assumptions are critical for inference and which ones are not. In short, we want to diagnose over a model set that includes the worlds for which our design works and the worlds in which we run into problems.

All designs break under some models, so the fact that a design ever breaks is no criticism. As research designers, we just want to know which models pose problems and which do not.

Principle 3.10. Diagnose whole designs

When we diagnose, we evaluate designs in their entirety. Too often, researchers evaluate parts of their designs in isolation: is this a good question? Is this a good estimator? What's the best way to sample? Design diagnosis requires knowing how each part of the design fits together. If you say, “my design has 80% power,” we want know, “power for what?” The power diagnosand could refer to an average effect estimator or to a subgroup effect estimator. If we ask, “What's your research design?” and you respond “It's a regression discontinuity design,” we've learned what class your answer strategy might be, but we don't have enough information to decide whether it's a strong design until we learn about the model, inquiry, data strategy, and other parts of the answer strategy.

In practice we do this by declaring the entire design, and asking how it performs, from start to finish, with respect to specified diagnosands. This pro-

cess requires a sufficiently complete design declaration. Indeed, the ability to run through a design to the point where a diagnosis can be undertaken (“diagnosand-completeness”) is a good indicator of an adequately-declared design.

Principle 3.11. Design to share

The declaration, diagnosis, and redesign process can improve the quality of the research designs you implement. This same process can also help you communicate your work, justify your decisions, and contribute to the scientific enterprise. Formalizing design declaration makes this sharing easier. By coding up a design as an object that can be run, diagnosed, and redesigned, you help other researchers see, understand, and question the logic of your research.

We urge you to keep this sharing function in mind as you write code, explore alternatives, and optimize over designs. An answer strategy that is hard-coded to capture your final decisions might break when researchers try to modify parts. Alternatively, designs can be created specifically to make it easier to explore neighboring designs, let others see why you chose the design you chose, and give them a leg up in their own work. In our ideal world, when you create a design, you contribute it to a design library so others can check it out and build on your good work.

Chapter 4

Software primer

This chapter serves as a brief introduction to the `DeclareDesign` package for the R programming language. `DeclareDesign` is a software implementation of every step of the declare-diagnose-redesign process. While you can declare, diagnose, and redesign using nearly any programming language, `DeclareDesign` is structured to make it easy to mix-and-match design elements while handling the tedious simulation bookkeeping behind the scenes.

4.1 Installing R

You can download R for free from CRAN. We also recommend the free program RStudio, which provides a friendly interface to R. Both R and RStudio are available on Windows, Mac, and Linux.

Once you have R and RStudio installed, open up RStudio and install `DeclareDesign` and its related packages. These include three packages that enable specific steps in the research process: `fabricatr` for simulating social science data, `randomizr` for random sampling and random assignment, and `estimatr` for design-based estimators. You can also install `DesignLibrary`, which gets standard designs up and running in one line. To install them all, copy the following code into your R console:

```
install.packages(c(  
  "DeclareDesign",  
  "fabricatr",  
  "randomizr",  
  "estimatr",
```

Table 4.1: Example data

ID	age	sex	party	precinct
001	66	M	REP	9104
002	54	F	DEM	8029
003	18	M	GRN	8383
004	42	F	DEM	2048
005	27	M	REP	5210

```
"DesignLibrary"
))
```

We also recommend that you install and get to know the `tidyverse` set of packages for data analysis, which we will use throughout the book:

```
install.packages("tidyverse")
```

For introductions to R and the `tidyverse` we especially recommend the free resource R for Data Science.

4.2 Declaring research design elements

Designs are declared through a concatenation of design elements. Almost all elements take a dataset as an input and return a dataset as their output. We will imagine an input dataset of 100 voters in Los Angeles. The research project involves randomly assigning voters to receive (or not) a knock on their door from a canvasser. Our data look like this:

The data strategy is a function that takes this dataset, implements a random assignment, adds it to the dataset, and then returns the resulting dataset.

You could write your own function to do that, but you can also use one of the `declare_*` functions in `DeclareDesign`. Each one of these functions is a kind of “function factory”: it takes a set of parameters about your research design as inputs, and returns a function as its output.

Here is an example of a `declare_assignment` element:

Table 4.2: Data output following implementation of an assignment step.

ID	age	sex	party	precinct	Z
001	66	M	REP	9104	1
002	54	F	DEM	8029	1
003	18	M	GRN	8383	0
004	42	F	DEM	2048	1
005	27	M	REP	5210	0

```
simple_random_assignment <-
  declare_assignment(Z = simple_ra(N = N, prob = 0.6))
```

The big idea here is that the object we created, `simple_random_assignment`, is not a particular assignment. Instead, it is a function that conducts assignment when called (see Principle 3.6). You can run the function on data:

```
simple_random_assignment(voter_file)
```

We want to emphasize that most steps are “dataset-in, dataset-out” functions. The `simple_random_assignment` function took the `voter_file` dataset and returned a dataset with assignment information appended.

Every step in a research design can be declared using one of the `declare_*` functions. Table 4.3 collects these according to the four elements of a research design. Below, we walk through common uses of each of these declaration functions.

Table 4.3: Declaration functions in DeclareDesign

Design component	Function	Description
Model	<code>declare_model()</code>	background variables and potential outcomes
Inquiry	<code>declare_inquiry()</code>	research questions
Data strategy	<code>declare_sampling()</code> <code>declare_assignment()</code> <code>declare_measurement()</code>	sampling procedures assignment procedures measurement procedures
Answer strategy	<code>declare_estimator()</code> <code>declare_test()</code>	estimation procedures testing procedures

Table 4.4: Draw from a fixed population

ID	age	sex	party	precinct
001	66	M	REP	9104
002	54	F	DEM	8029
003	18	M	GRN	8383
004	42	F	DEM	2048
005	27	M	REP	5210

4.2.1 Model

The model defines the number of units under study, their background characteristics, the latent outcomes we want to measure, and their potential outcomes. We can define the model in several ways. In some cases, you may start a design with data on the units you wish to study. When that happens, we may not need to simulate all parts of the model. We can start declaring the model with existing data.

```
declare_model(data = voter_file)
```

We typically need to simulate part or all of the model. Even when we have background data, we don't have access to the latent outcomes or potential outcomes that are needed to define many kinds of causal and descriptive inquiries (see Principle 3.5).

We can use the data simulation functions from the `fabricatr` package to simulate when we do not have complete data on the units under study. For instance, we can declare a model that generates a dataset with 100 units and a random variable U:

```
declare_model(N = 100, U = rnorm(N))
```

When we run this model function, we will get a different 100-unit dataset each time, as shown in Table 4.5.

Defining potential outcomes is as easy as a single expression per potential outcome. Potential outcomes may depend on background characteristics or other potential outcomes.

Table 4.5: Five draws from the model

Draw 1		Draw 2		Draw 3		Draw 4		Draw 5	
ID	U								
001	0.377	001	1.369	001	1.556	001	-2.530	001	1.459
002	-1.310	002	-0.058	002	-1.327	002	0.243	002	0.409
003	0.078	003	0.449	003	-0.430	003	-1.596	003	-0.692
004	-0.795	004	1.077	004	0.814	004	0.076	004	0.037
005	1.766	005	0.186	005	0.035	005	1.590	005	-0.619

Table 4.6: Adding potential outcomes to the model

ID	U	Y_Z_0	Y_Z_1
001	0.521	0.521	0.771
002	1.990	1.990	2.240
003	-0.952	-0.952	-0.702
004	0.626	0.626	0.876
005	0.733	0.733	0.983

```
declare_model(
  N = 100,
  U = rnorm(N),
  Y_Z_0 = U,
  Y_Z_1 = Y_Z_0 + 0.25
)
```

We also provide an alternative interface for defining potential outcomes that uses R's formula syntax with the `potential_outcomes` function. The formula syntax lets you specify "regression-like" outcome equations. One downside is that it mildly obscures how the names of the eventual potential outcomes columns are named. We build the names using the outcome name (here `Y` on the left-hand side of the formula) and the name of the assignment variable from the variable name in the `conditions` argument (here `Z`). We also defined the two values `Z` takes on, 0 and 1 — so the two potential outcomes columns will be named `Y_Z_0` and `Y_Z_1`.

```
declare_model(
```

```
N = 100,
U = rnorm(N),
potential_outcomes(Y ~ 0.25 * Z + U, conditions = list(Z = c(0, 1)))
)
```

Either way of creating potential outcomes works; one may be easier or harder to code up in a given research design setting.

4.2.2 Inquiry

To define the inquiry, we declare a summary function of the events generated by the model. We can declare the “population average treatment effect” inquiry as the average difference in the two variables created by the model above.

```
declare_inquiry(PATE = mean(Y_Z_1 - Y_Z_0))
```

4.2.3 Data strategy

The data strategy constitutes one or more steps representing interventions the researcher makes in the world from sampling to treatment assignment to measurement.

4.2.3.1 Sampling

The sampling step typically involves drawing a random sample of units and then filtering to the sampled units, dropping the unsampled units. You can use the `complete_rs` function from the `randomizr` package to conduct the sampling. See Section 8.1.1 for an overview of the many kinds of sampling that are possible with `randomizr`. The second step is accomplished with the `filter` argument, which by default retains units for which `S == 1`. Here, we draw a complete random sample of 50 units from the population:

```
declare_sampling(S = complete_rs(N, n = 50), filter = S == 1)
```

When we draw data from our simple design at this point, it will have fewer rows. It will have shrunk from 100 units in the population to a data frame of 50 sampled units.

Table 4.7: Sampled data

	ID	U	Y_Z_0	Y_Z_1	S
1	001	-1.664	-1.664	-1.414	1
2	002	-1.355	-1.355	-1.105	1
3	003	1.025	1.025	1.275	1
6	006	-0.885	-0.885	-0.635	1
9	009	-0.660	-0.660	-0.410	1

Table 4.8: Sampled data with assignment indicator

ID	U	Y_Z_0	Y_Z_1	S	Z	Y
001	0.669	0.669	0.919	1	0	0.669
003	-0.251	-0.251	-0.001	1	1	-0.001
004	-0.724	-0.724	-0.474	1	0	-0.724
006	0.399	0.399	0.649	1	1	0.649
010	0.619	0.619	0.869	1	0	0.619

4.2.3.2 Treatment assignment

In experimental studies, units are assigned to one of two or more treatment conditions. The `randomizr` package provides functions for randomly assigning units. Here we use complete random assignment with probability 0.5:

```
declare_assignment(Z = complete_ra(N, prob = 0.5))
```

After treatments are assigned, some potential outcomes are revealed. Treated units reveal their treated potential outcomes and untreated units reveal their untreated potential outcomes. In most declarations, you need a measurement step to reveal measured outcomes. The `reveal_outcomes` function performs this “switching” operation, so called because the function “switches” which potential outcome is revealed depending on the value of the random assignment.

```
declare_measurement(Y = reveal_outcomes(Y ~ Z))
```

4.2.3.3 Measurement

Measurement is how we translate latent events into observed data. For example, we might imagine that the normally distributed outcome variable Y is a latent

Table 4.9: Sampled data with an explicitly measured outcome

ID	U	Y_Z_0	Y_Z_1	S	Z	Y	Y_binary
005	1.978	1.978	2.228	1	1	2.228	1
017	-0.603	-0.603	-0.353	1	1	-0.353	1
018	-1.296	-1.296	-1.046	1	0	-1.296	0
020	0.213	0.213	0.463	1	0	0.213	1
024	0.797	0.797	1.047	1	1	1.047	1

outcome that will be translated into a binary outcome when measured by the researcher:

```
declare_measurement(Y_binary = rbinom(N, 1, prob = pnorm(Y)))
```

4.2.4 Answer strategy

We declare answer strategy steps using the `declare_estimator` function, which plays nicely with the many statistical modeling functions available in R, such as `lm`, `glm`, or the `ictreg` function from the `list` package, among hundreds of others. Throughout the book, we will be using many estimators from `estimatr` (like `lm_robust` and `difference_in_means`) because they are fast and calculate robust standard errors easily.

Estimators are associated with inquiries. Here, we target the population average treatment effect with the difference-in-means estimator.

```
declare_estimator(
  Y ~ Z, model = difference_in_means, inquiry = "PATE"
)
```

The output from a modeling function is a complicated model fit object that contains large amounts of information. We typically only want a few summary pieces of information out of these objects, like the coefficient estimates, standard errors, and confidence intervals. We use model summary functions passed to the `model_summary` argument of `declare_estimator` to do so. Model summary functions take model fits as inputs and return answers as data frames.

The default model summary function is `tidy`:

```
declare_estimator(
  Y ~ Z, model = lm_robust, model_summary = tidy
)
```

4.3 Building a design from design elements

We now declare all the individual design elements in one go.

```
model <-
  declare_model(
    N = 100,
    U = rnorm(N),
    potential_outcomes(Y ~ 0.25 * Z + U)
  )

inquiry <-
  declare_inquiry(PATE = mean(Y_Z_1 - Y_Z_0))

sampling <-
  declare_sampling(S = complete_rs(N, n = 50))

assignment <-
  declare_assignment(Z = complete_ra(N, prob = 0.5))

measurement <-
  declare_measurement(Y = reveal_outcomes(Y ~ Z))

answer_strategy <-
  declare_estimator(
    Y ~ Z, model = difference_in_means, inquiry = "PATE"
  )
```

To construct a research design object that we can operate on — diagnose it, redesign it, draw data from it, etc. — we add them together with the `+` operator.

```
design <-
  model +
  inquiry +
```

```
sampling + assignment + measurement +
answer_strategy
```

We usually declare designs more compactly, concatenating steps directly with `+`. Declaration 4.1 shows the format of most declarations throughout the book.

Declaration 4.1. Two-arm randomized experiment

```
design <-
  declare_model(N = 100, U = rnorm(N),
               potential_outcomes(Y ~ 0.25 * Z + U)) +
  declare_inquiry(PATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_sampling(S = complete_rs(N, n = 50)) +
  declare_assignment(Z = complete_ra(N, prob = 0.5)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(
    Y ~ Z, model = difference_in_means, inquiry = "PATE"
  )
```

Order matters in declaring designs. We can think of the order of the declaration as the temporal order in which steps take place. Below, since the inquiry comes before sampling and assignment, the inquiry is a *population* inquiry, the population average treatment effect.

```
model +
  declare_inquiry(PATE = mean(Y_Z_1 - Y_Z_0)) +
  sampling +
  assignment +
  measurement +
  answer_strategy
```

We could instead define our inquiry as a *sample* average treatment effect by putting the inquiry after sampling:

```
model +
  sampling +
  declare_inquiry(SATE = mean(Y_Z_1 - Y_Z_0)) +
```

Table 4.10: Simulated data draw

ID	U	Y_Z_0	Y_Z_1	S	Z	Y
001	0.214	0.214	0.464	1	1	0.464
004	1.140	1.140	1.390	1	0	1.140
011	-0.136	-0.136	0.114	1	1	0.114
012	-0.987	-0.987	-0.737	1	0	-0.987
014	-0.795	-0.795	-0.545	1	0	-0.795

```
assignment +
measurement +
answer_strategy
```

4.4 Simulating a research design

Diagnosing a research design — learning about its properties — requires first simulating by running the design over and over. We need to simulate the event generating process, calculate the values of the inquiries, then draw simulated data and calculate the resulting estimates. To draw simulated data, we use `draw_data`:

```
draw_data(design)
```

`draw_data` runs all of the “data steps” in a design, which are both from the model and from the data strategy (sampling, assignment, and measurement).

To simulate the estimands from a single run of the design, we use `draw_estimands`. This function runs two operations at once: it draws the events, and calculates the estimands at the point defined by the design.

```
draw_estimands(design)
```

Similarly, we can draw the estimates from a single run with `draw_estimates`, which simulates data and, at the appropriate moment, calculates estimates.

Table 4.11: Estimands calculated from simulated data.

inquiry	estimand
PATE	0.25

Table 4.12: Estimates calculated from simulated data.

term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome	inquiry
Z	0.132	0.273	0.482	0.632	-0.418	0.681	47.837	Y	PATE

```
draw_estimates(design)
```

To simulate whole designs, we use the `simulate_design` function to draw data and calculate estimands and estimates many times in a row (500 times by default).

```
simulate_design(design)
```

4.5 Diagnosing a research design

Using the simulations data frame, we can calculate diagnosands like bias, root mean-squared-error, and power for each estimator-inquiry pair. In DeclareDesign, we do this in two steps. First, we declare diagnosands, which are functions that summarize the building blocks of diagnosands, diagnostic statistics. The software includes many pre-coded diagnosands (see Section 10), though

Table 4.13: Simulations data frame.

sim_ID	estimand	estimate	std.error	statistic	p.value	conf.low	conf.high	df
1	0.25	0.248	0.285	0.872	0.388	-0.324	0.821	47.918
2	0.25	-0.182	0.308	-0.592	0.557	-0.804	0.439	42.897
3	0.25	0.410	0.294	1.396	0.170	-0.181	1.002	46.011
4	0.25	0.172	0.234	0.732	0.468	-0.300	0.643	47.882
5	0.25	0.710	0.321	2.210	0.032	0.063	1.357	45.143

Table 4.14: Design diagnosis.

Bias	RMSE	Power
-0.00	0.28	0.14
(0.01)	(0.01)	(0.01)

you can write your own like this:

```
study_diagnosands <-
  declare_diagnosands(
    bias = mean(estimate - estimand),
    rmse = sqrt(mean((estimate - estimand) ^ 2)),
    power = mean(p.value <= 0.05)
  )
```

Diagnosands are summaries of the simulations data frame. The bias diagnosand first calculates the difference between estimate and estimand, and then takes the average.

Next, we apply your diagnosand declaration to the simulations data frame with the `diagnose_design` function:

```
diagnose_design(simulation_df, diagnosands = study_diagnosands)
```

We can also do this in a single step by sending a design object directly to `diagnose_design`. The function will first run the simulations, then calculate the diagnosands.

```
diagnose_design(design, diagnosands = study_diagnosands)
```

4.6 Redesign

We redesign to learn how the diagnosands change as design features change. We can do this using `redesign` over a range of sample sizes, resulting in a list of designs.

```
designs <- redesign(design, N = c(100, 200, 300, 400, 500))
```

Our simulation and diagnosis tools can operate directly on a list of designs:

```
diagnose_design(designs)
```

4.7 Library of designs

In our `DesignLibrary` package, we have created a set of common designs as designers (functions that create designs from just a few parameters), so you can get started quickly.

```
library(DesignLibrary)

block_cluster_design <- block_cluster_two_arm_designer(N = 1000, N_blocks = 10)
```

4.8 Complex declarations

We have illustrated the simple way to use `DeclareDesign` declarations thus far, and throughout the book the majority of declarations rely on this method. However, you can also escape the standard way of doing things at any step. Each design element has a “handler” that works behind the scenes on the bookkeeping. You can switch to your own function or to a function from another package at any time. In addition, you can skip our diagnosis tools by simply operating on the simulations data frame itself.

For example, we may want to declare an inquiry for many subgroups of the units in your population. A custom function relying on a `dplyr` pipeline to group data by city and calculate the by-city ATEs could be used like this:

```
declare_inquiry(
  handler = function(data) {
    # start with data
    data %>%
      # split the dataset by city
```

```
group_by(city) %>%
  # estimate city-level ATEs and return as a data.frame
  summarize(city_ATE = mean(Y_Z_1 - Y_Z_0), .groups = "drop")
}
```

Further reading

This primer is an introduction to DeclareDesign, but it only begins to scratch the surface of what we can do with the software. At the end of each section in Part II, we illustrate how to tackle interesting problems that come up during declaration, diagnosis, and redesign.

In the meantime, we recommend the following external resources for learning more.

- [DeclareDesign.org](#)
- [randomizr cheatsheet](#)
- [estimatr cheatsheet](#)
- [DeclareDesign cheatsheet](#)
- [fabricatr vignette](#)
- [R for Data Science](#)
- [RStudio R primers](#)
- [Computational social science bootcamp](#)

48

Software primer

4.8

Part II

Declaration, Diagnosis, Redesign

Chapter 5

Declaration

In Chapter 2, we gave a high-level overview of our framework for describing research designs in terms of their models, inquiries, data strategies, and answer strategies, our process for diagnosing their properties, and a general purpose approach for improving them to better fit research tasks. Now in this chapter, we place our approach on a firmer formal footing. To do so, we employ elements from Pearl's (2009) approach to causal modeling, which provides a syntax for mapping design inputs to design outputs. We also use the potential outcomes framework as presented, for example, in Imbens and Rubin (2015), which many social scientists use to clarify their inferential targets.

Describing a research design as a DAG helps us to see the fundamental symmetries across the theoretical (M and I) and empirical (D and A) halves of a research design. A recurring theme of our book is that research designs tend to be stronger when the relationship of M to I is mirrored by the relationship of D to A ; the aim of this chapter is to make this somewhat abstract claim more concrete.

5.1 Definition of research designs

Research design are defined by four elements: a model M , an inquiry I , a data strategy D , and an answer strategy A . Describing a research design entails “declaring” each of these four elements.

M is a set of possible models of how the world works. Following Pearl's definition of a probabilistic causal model, a model in M contains three core elements. The first is a specification of the variables X about which research is being conducted. This includes endogenous and exogenous variables (V and U respectively) and the ranges of these variables. In the formal literature, this is sometimes called the *signature* of a model (Halpern, 2000). The second element (F) is a specification of how each endogenous variable depends on other

variables. These can be considered functional relations or, as in Imbens and Rubin (2015), potential outcomes because they describe what *would* happen under different possible conditions. The third and final element is a probability distribution over exogenous variables, written as $P(U)$. Sometimes it is useful to think of the draws from U as implying distinct models of their own, in which case we might think of M as a family of models and a particular model m as an element of M that fully specifies what would happen under all conditions. We avoid the phrase “data generating process” to refer to m (since data are generated by the data strategy) and instead use the phrase “event generating process.”

The **inquiry** I is a summary of the variables X , perhaps given interventions on some variables. An inquiry might be the average value of an outcome Y : $E[Y] = (y \in \Pr(Y = y))$, or the average value of the outcome conditional on the value of a treatment Z : $E[Y|Z = 1] = (y \in \Pr(Y = y|Z = 1))$. Using Pearl’s notation we can distinguish between descriptive inquiries and causal inquiries. Causal inquiries are those that summarize distributions that would arise under interventions, as indicated by the `do()` operator, e.g., $\Pr(Y|\text{do}(Z = 1))$. Descriptive inquiries summarize distributions that arise without intervention, such as $\Pr(Y|Z = 1)$. This is the difference between the average outcome if you “set” Z to 1 compared to the average outcome when Z so happens to be 1. The difference, to use an example of the form found in Pearl (2009), between the probability that it is raining when you make people put up umbrellas (low) versus the probability it’s raining when people have umbrellas up (high).

We let a_m denote the answer to I under the model. Conditional on the model, a_m is the value of the estimand, the quantity that the researcher wants to learn about, or would want to learn about if the world were like the model. The connection of a_m to the model is given by: $a_m = I(m)$.

As the saying goes, models are wrong but some may perhaps be useful. We denote the *true* causal process as m : the process that generates events in the real world. The *right* answer, then, is $a_m = I(m)$. The answer under a reference model a_m may be close or far from the true value a_m , which is to say it could be wrong. If the model m is far from m , then of course a_m need not be correct. Moreover a_m might even be undefined, since inquiries can only be stated in terms of theoretical models. If the theoretical model is wrong enough—for instance conditioning on events that do not in fact arise—then the inquiry might be nonsensical when applied to the real world. For example, “what is the ideological slant of a speech that is not given” is an inquiry that is undefined.

A **data** strategy D generates data d . Data d arises under model M with probability $P_M(d|D)$. The data strategy includes sampling, assignment, and measurement strategies. Nearly all data strategies sample and measure, but not all assign treatments. Whether or not the data strategy includes assignment is the defining distinction between experimental and observational studies. When applied in the real world, the data strategy operates on m to produce the realized data: $D(m) = d$. When we simulate research designs, the data strategy

operates on a simulated model draw m to produce fabricated data: $D(m) = d$.

Finally, the **answer** strategy A generates answers using data. When applied to realized data, the answer strategy returns the empirical answer: $A(d) = a_d$. When applied to simulated data, it returns a simulated answer: $A(d) = a_m$

Table 5.1 provides a concise description of each element of a research design and relates them to some common terms. We flag here that the term estimand has a slightly different meaning in our framework than elsewhere. We say that an estimand a_m is the value of an inquiry I , whereas in some traditions “estimand” can refer to the inquiry I or to an intermediate parameter that happens to be targeted by an estimator.

Table 5.1: Elements of research design.

Notation	Description	Related terms
M	a stipulated collection of causal models	
m	a single model in M , represented by events	a hypothetical data generating process
m	the true model	true data generating process
I	the inquiry	estimand; quantity of interest
$a_m = I(m)$	the answer under the model; an estimand	
$a_m = I(m)$	the true answer; the estimand	estimand; quantity of interest
D	the data strategy	
$d = D(m)$	fabricated data; simulated data	
$d = D(m)$	realized data	
A	the answer strategy	data analysis; estimator
$a_d = A(d)$	a simulated answer; an estimate	the observed estimate
$a_d = A(d)$	the empirical answer; the estimate	the observed estimate

The full set of causal relationships between M, I, D, A , with respect to m and m , a_m and a_m , d and d , and a_d and a_m can be seen in the schematic representation of a research design given in Figure 5.1. The figure illustrates how a research design involves a correspondence between $I(m) = a_m$ and $A(d) = a_d$. The theoretical half of a research design produces an answer to the inquiry *in theory*. The empirical half of a research design produces an *empirical estimate* of the answer to the inquiry. Neither answer is necessarily close to the truth a_m , of course. And, as shown in the figure, the truth is not directly accessible either to us in theory or in empirics. Our gamble in empirical research, however, is

that our theoretical models are close enough to the truth: that the truth is like the set of models we imagine. If the models in M do not contain m or are too different from the truth, then the research design process – ex post – could cause researchers to select poor designs.

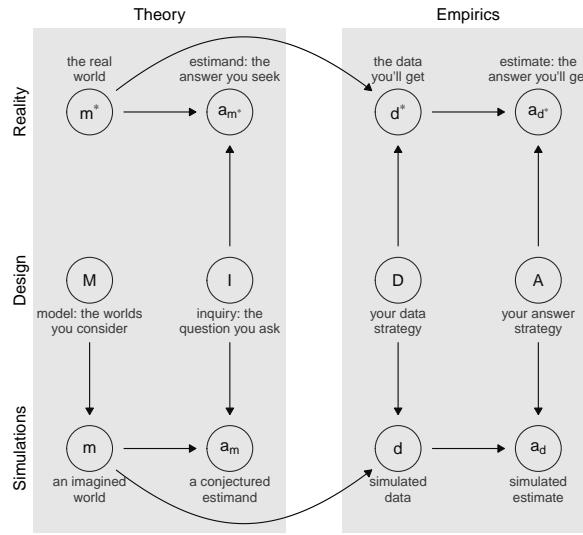


Figure 5.1: *MIDA* as a DAG

Figure 5.1. also reveals a striking analogy between the M, I relationship and the D, A relationship. The answer we aim for is obtained by applying I to a draw from M . But the answer we have access to is obtained by applying A to a draw from D . And our hope, usually, is that these two answers are quite similar. In some cases, this suggests that the function A should be “like” the function I . For instance, if we are interested in the mean of a population and we have access to a random sample, the data available to us from D is like the ideal data we would have if we could observe the nodes and edges in M directly. This mirroring across the two halves of a research design is the root of Principle 3.7: Seek $M:I:D:A$ parallelism.

Finally, in Figure 5.1 no arrows go into M, I, D , or A , since they are not caused by any of the other nodes in the DAG. We could have included a node for the research designer, who deliberately sets the details of M, I, D , and A , but we omit it for clarity.

5.2 Declaration in code

Table 5.2 illustrates these different quantities. We stipulate a set of event generating processes, M , in which Y depends on X . We define a question, I : what is the average value of Y when $X = 1$? We calculate what the right answer to our inquiry would be under one of our stipulated event generating processes, $I(w)$. We also imagine we could describe how the world in fact is, m , and calculate what the right answer would be in that case, a_m . We then apply the data strategy to produce d and use an answer strategy A and use it to calculate what answer we would get given d .

For each of these steps we show DeclareDesign code in the first column. In the second column, we show the simulated m , m , and d datasets, along with the values of a_m , a_m , and a_d for one run of the simulation.

Table 5.2: Elements of research design in code.

Description	Draw			
	m: draw from M (N = 1000)			
		ID	U	X
		0001	-1.397	0
		0002	0.523	1
		0003	0.142	1
		0004	-0.847	0
		0005	-0.412	0
		0006	-1.465	0
M <- declare_model(N = 1000, .seed = 12345, .inquiry_label = "Ybar") m <- M() U = rnorm(N), X = rbinom(N, 1, prob = pnorm(U)), Y = rbinom(N, 1, prob = pnorm(U + X)) a_m <- I(m)	a^m: answer under the model			
		inquiry_label	estimand	
		Ybar	0.615	
I <- declare_inquiry(Ybar ~ U + X)				
a_m <- I(m)				

Description	Draw			
		ID	U	X
w: draw from W (N = 1000)				
		0001	-0.109	0
		0002	1.204	0
		0003	0.712	1
		0004	1.649	1
		0005	0.745	0
		0006	-2.238	0
mstar <- fabricate(N = 1000,				
U = rnorm(N),				
X = rbinom(N, 1, prob = pnorm(U)),				
Y = rbinom(N, 1, prob = pnorm(U))				
	a^w: true answer			
		inquiry_label		estimator
		Ybar		0.48
a_mstar <- I(mstar)	d: D() applied to w: (N = 200)			
ID		U	X	Y
0003		0.712	1	1
0005		0.745	0	0
0009		1.073	1	1
0031		1.451	1	1
0041		-0.624	1	0
0046		0.367	0	1
D <- declare_sampling(
S = simple_rs(N, prob = 0.1))				
dstar <- D(mstar)				

Description	Draw					
		inquiry_label	estimate	std.error	conf.low	conf.high
	a^d: empirical answer					
A <- declare_estimator(
Y ~ 1, model = lm_robust						
subset = X == 1, inquiry = "Ybar")						
a_dstar <- A(dstar)						

Further reading

- Imbens and Rubin (2015) on potential outcomes
- Halpern (2000) on causal models

58

Declaration

5.2

Chapter 6

Specifying the model

Models are theoretical abstractions we use to make sense of the world and organize our understanding of it. They play many critical roles in research design. First and foremost, models describe the units, conditions, and outcomes that define inquiries. Without well-specified models, we cannot ask well-specified questions. Second, models provide a framework to evaluate the sampling, assignment, and measurement procedures that form the data strategy. Models encode our beliefs about the kinds of information that will result when we conduct empirical observations. Third, provide a framework to evaluate answer strategies: what variables should we condition on, what variables should we **not** condition on, how flexible or rigid should our estimation procedure be? Whenever we rely on assumptions in the model – for example, normality of errors, conditional independencies, or latent scores – we are betting that the real causal model m has these properties.

We need to imagine models in order to declare and diagnose research designs. This need often generates discomfort among students and researchers who are new to thinking about research design this way. In order to compute the root mean squared error, bias, or statistical power of a design, we need to write down **more than we know for sure** in the model. We have to describe joint distributions of covariates, treatments, and outcomes, which entails making guesses about the very means, covariances, and effect sizes (among many other things) that the empirical research design is supposed to measure. “What do you mean, write down the potential outcomes – that’s what I’m trying to learn!”

The discomfort arises because we do not know the true causal model of the world – what we referred to as m in Figure 5.1. We are uncertain about which of the many plausible models of the world we entertain is the correct one. The M in *MIDA* refers to these possible models, which we call “reference models.” M is a set of reference models, with the typical element m , though in some cases the M may just be a single model. We have no particular theoretical commitment to

reference models. Their role is to provide a stipulation of how the world works, which allows us to answer some questions about our research design. If the reference model were true, what would the value of the inquiry be? Would my estimator be unbiased? How many units would I need to achieve an RMSE of 0.5? Critically, whether a design is good or bad depends on the reference model. A data and analysis strategy might fare very well under one model of the world but poorly under another. Thus to get to the point where we can assess a design we need to make the family of reference models explicit. This chapter is about how to go about this difficult task.

6.1 Elements of models

Models are characterized by three elements: the signature, the functional relationships, and a probability distribution over exogenous variables. We'll describe each in turn.

6.1.1 Signature

The signature of the model describes the variables in the model and their ranges. The signature comprises two basic kinds of variables: exogenous variables and endogenous variables. Exogenous means “generated from without” and endogenous means “generated from within.” Stated more plainly, exogenous variables are not caused by other variables in the model because they are randomly assigned by nature or by human intervention. Endogenous variables result as a consequence of exogenous variables; they are causally downstream from exogenous variables.

What kinds of variables are exogenous? Typically, we think of explicitly randomly assigned variables as exogenous: the treatment assignment variable in a randomized experiment is exogenous. We'll often use the variable letter Z to refer to assignments that were explicitly randomized. We also often characterize the set of unobserved causes of observed variables as exogenous. We summarize the set of unobserved causes of an observed variable with the letter U . These unobserved causes are exogenous in the sense that, whatever the causes of U may be, they do not cause other endogenous variables in a model.

What kinds of variables are endogenous? Everything else: covariates, mediators, moderators, and outcome variables. We'll often use the letter X when describing covariates or moderators, the letter M when describing mediators, and the letter Y when describing outcome variables. Each of these kinds of variables is the downstream consequence of exogenous variables, whether those exogenous variables are observed or not.

Critically the signature of a model is *itself a part of the design*. We get to choose what are the variables of interest and even their scales. We do not, however, get to decide the functional relations between variables – those are set according to m .

6.1.2 Functional relations

The second element of the model is the set of functions that produce endogenous variables. The output of these functions are always endogenous variables and the inputs can be either exogenous variables or other endogenous variables. We embrace two different, but ultimately compatible, ways of thinking about these functional relationships: structural causal models and the potential outcomes model.

The structural causal model account of causality is often associated with directed acyclic graphs (DAGs). Each node on a graph is a variable and the edges that connect them represent possible causal effects. An arrow from a “parent” node to a “child” node indicates that the value of the parent sometimes influences the outcome of the child. More formally: the parent’s value is an argument in a functional equation determining the child’s outcome. DAGs emphasize a mechanistic notion of causality. When the exposure variable changes, the outcome variable changes as a result, possibly in different ways for different units.

DAGs represent *nonparametric* structural causal models. This means that they don’t show *how* variables are related, just *that* they are related. This is no criticism of DAGs — they just don’t encode all of our causal beliefs about a system. We illustrate these ideas using a simple DAG to describe a model with an abstract research design in which we will collect information about N units. We will assign a treatment Z at random, and collect an outcome Y . We know there are other determinants of the outcome beyond Z , but we don’t know much about them. All we’ll say about those other determinants U is that they are causally related to Y , but not to Z , since Z will be randomly assigned by us.

This nonparametric structural causal model can be written like this:

$$Y = f_Y(Z, U)$$

Here, the outcome Y is related to Z and U by some function f_Y , but the details of what the function f_Y is – whether Z has a positive or negative effect on Y , for example – are left unstated in this nonparametric model. The DAG in Figure 6.1 encodes this model in graphical form. We use a blue circle around the treatment assignment to reflect the idea that Z is randomly assigned as part of the data strategy.

To assess many properties of a research design we will often need to make the leap from nonparametric models to *parametric* structural causal models. We need to enumerate beliefs about effect sizes, correlations between variables, intra-class correlations (ICCs), specific functional forms, and so forth. Since any particular choice for these parameters could be close or far from the truth, we will typically consider a range of plausible values for each model parameter.

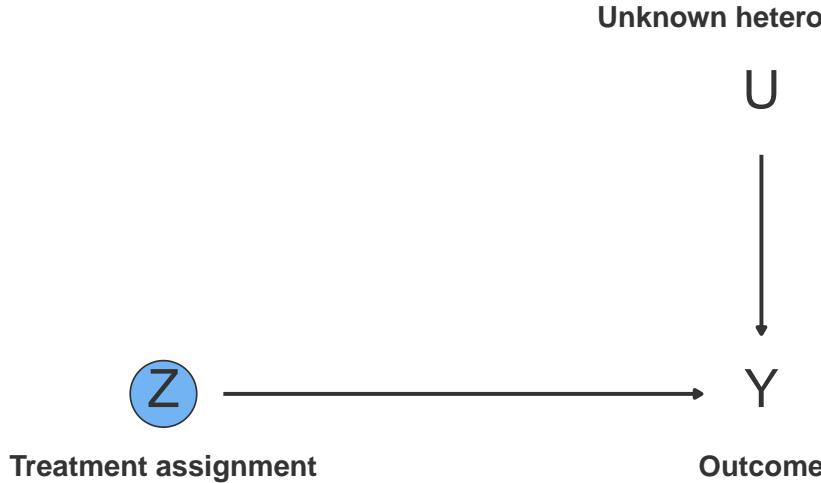


Figure 6.1: Simple DAG. U is unobserved.

One possible parametric model is given by the following:

$$Y = 0.5 \times Z + U$$

Here, we have specified the details of the function that relates Z and U to Y . In particular, it is a linear function in which Y is equal to the unobserved disturbance U in the control condition, but is 0.5 higher in the treatment condition. We could also consider a more complicated parametric model in which the relationship between Z and Y depends on an interaction with the unobservables in U :

$$Y = 0.25 \times Z - 0.05 \times Z \times U + U$$

Both of these parameterizations are consistent with the DAG in Figure 6.1, which underlines the powerful simplicity of DAGs but also their theoretical sparsity. The two parametric models are theoretically quite different from one another. In the first, the effects of the treatment Z are positive and the same for all units; in the second, the effects are negative and quite different from unit to unit, depending on the value of U . If both of these reference models are plausible, we'll want to include them both in M , to ensure that our design is robust to both possibilities. This is a small instance of Principle 3.3 to entertain many models – we want to consider a wide range of plausible parameterizations since we are ignorant of the true causal model (m).

By contrast with structural causal models, the **potential outcomes** formalization emphasizes a counterfactual notion of causality. $Y_i(Z = 0)$ is the outcome for unit i that would occur were the causal variable Z were set to zero and $Y_i(Z = 1)$ is the outcome that would occur if Z were set to one. The difference between them defines the effect of the treatment on the outcome for unit i . Since at most only one potential outcome can ever be revealed, at least one of the two potential outcomes is necessarily counterfactual. Usually, the potential outcomes notation $Y_i(Z)$ reports how outcomes depend on one feature, Z , ignoring all other determinants of outcomes. This is not to say that these don't matter—they do—they are just not the focus. In a sense, they are contained in the subscript i since the units carry with them all relevant features other than Z . We can generalize to settings where we want to consider more than one cause, in which case we use expressions of the form $Y_i(Z = 0, X = 0)$ or $Y_i(Z = 0, X = 1)$.

Under the first structural model we consider, the potential outcomes (for Y) might be written, for $i \in \{1, 2, \dots, n\}$ as:

$$\begin{aligned} Y_i(0) &= u_i \\ Y_i(1) &= 0.5 + u_i \end{aligned}$$

The potential outcomes under the second model would be written:

$$\begin{aligned} Y_i(0) &= u_i \\ Y_i(1) &= 0.25 + 0.95 u_i \end{aligned}$$

Despite what you may have inferred from the sometimes heated disagreements between scholars who prefer one formalization to the other, structural causal models and potential outcomes are compatible systems for thinking about causality. Potential outcome distributions can also be described using Pearl's `do()` operator: $Pr(Y|do(Z = 1))$ is the probability distribution of the treated potential outcome. We could use only the language of structural causal models or we could use only the language of potential outcomes, since a theorem in one is theorem in the other (Pearl, 2009, p.243) We choose to use both languages because they are useful for expressing different facets of research design. We use structural causal models to describe the web of causal interrelations in a concise way (writing out the potential outcomes for every relationship in the model is tedious). We use potential outcomes when the inquiry involves comparisons across conditions and to make fine distinctions between inquiries that apply to different sets of units.

6.1.3 Probability distribution over exogenous variables

The final element of a model is a description of the probability distribution of exogenous variables. For example, we might describe the distribution of the treatment assignment as a Bernoulli distribution with $p = 0.1$ to describe “coin flip” random assignment with a 10% chance of a unit being assigned to treatment. We might stipulate that the unobserved disturbances U are normally distributed with a mean of 1 and a standard deviation of 2. The distributions of the exogenous variables then ramify through to the distributions of the endogenous variables through the functional relations.

How we draw from these probability distributions has consequences for what we think of as the set of units about whom we are drawing inferences. There are three distinct ways of thinking about this problem.

The first, and possibly most common way of thinking is the “finite population” setting. Here we enumerate a fixed population of units about whom we seek to draw inferences. We sample from this finite population in the data strategy in such a way that we can use the sample to draw inferences about the population. The probability distribution over the exogenous variables simply enumerates the values that these variables take on in the population. Any randomness in the design is generated by the sampling and assignment procedures, not in the values of the exogenous variables.

A second, and closely related, framework is the finite *sample* setting. Here the population *is* the sample, and we don’t contemplate any extrapolation from the sample to the population. Finite sample inference is common in research designs that involve random assignment of treatments. The only source of randomness in the finite sample setting is the random assignment itself.

A third class is the superpopulation setting, in which we imagine that any particular population is just a draw from an infinite population. In this case, we can conceive of the randomness in the design as being fundamental – every unit is a random draw from the superpopulation.

6.2 Lexicon of common variable types

Any particular causal model will be a complex web of exogenous and endogenous variables woven together via a set of functional relationships. Despite the heterogeneity across models, we can describe the roles variables play in a research design with reference to the roles they play in structural causal models. There are seven:

1. **Outcomes:** the variable whose level or responses we want to understand, generally referred to as Y , as in Figure 6.2. Variously described as “dependent variables,” “endogenous variables,” “left-hand side variables,” or “response variables.”

2. **Treatments:** the main variable or variables that affect outcome variables under study. We will use D most often to refer to the main causal variable of interest in a particular study.
3. **Moderators.** Variables that condition the effects of treatment variables on outcomes. See for example X_2 in Figure 6.2. The figure indicates that X_2 is a cause of Y but does not explicitly indicate that there is an interaction between D and X_2 . One account for this is that as a general matter if two variables cause an outcome it would be surprising if they did *not* interact in some way.
4. **Mediators.** Variables “along the path” from treatment variables to outcomes. M is an example of a mediator in this figure. Mediators are often studied to assess “how” or “why” D causes Y .
5. **Confounders.** Variables that introduce a noncausal correlation between D and Y . In the figure, X_1 is a confounder because it causes both S and Y and could introduce a correlation between them even if D did not cause Y .
6. **Instruments.** An instrumental variable is an exogenous variable that affects a treatment variable and can help us figure out the relationship between the treatment variable and the outcome. Such variables are studied instrumentally and not for their own sake. We give a much more detailed treatment of these variables in Section 15.4. We reserve the letter Z for instruments. Random assignments are instruments in the sense that the assignment is the instrument and the actual treatment received is the treatment variable.
7. **Colliders.** Colliders are variables that are caused by two other variables. Colliders can be important because conditioning on a collider introduces a noncausal relationship between the causes of the collider. In figure 6.2, K is a collider that can create a noncausal correlation between D and Y (via U) if conditioned upon.

These labels reflect the researcher’s interest as much as their position in a model. Another researcher examining the same graph might, for instance, label M as their treatment variable or K as their outcome of interest.

6.2.1 What variables are needed?

Our models of the world can be more or less complex, or at least articulated at higher or lower levels of generality. How specific and detailed we need to be in our specification of possible models depends on the other features of the research design: the inquiry, the data strategy, and the answer strategy. At a minimum, we need to describe the variables required for each of these research design elements.

Inquiry: In order to reason about whether the model is sufficient to define the inquiry, we need to define the units, conditions, and outcomes used to construct our inquiry. If the inquiry is an average causal effect among a subgroup, we

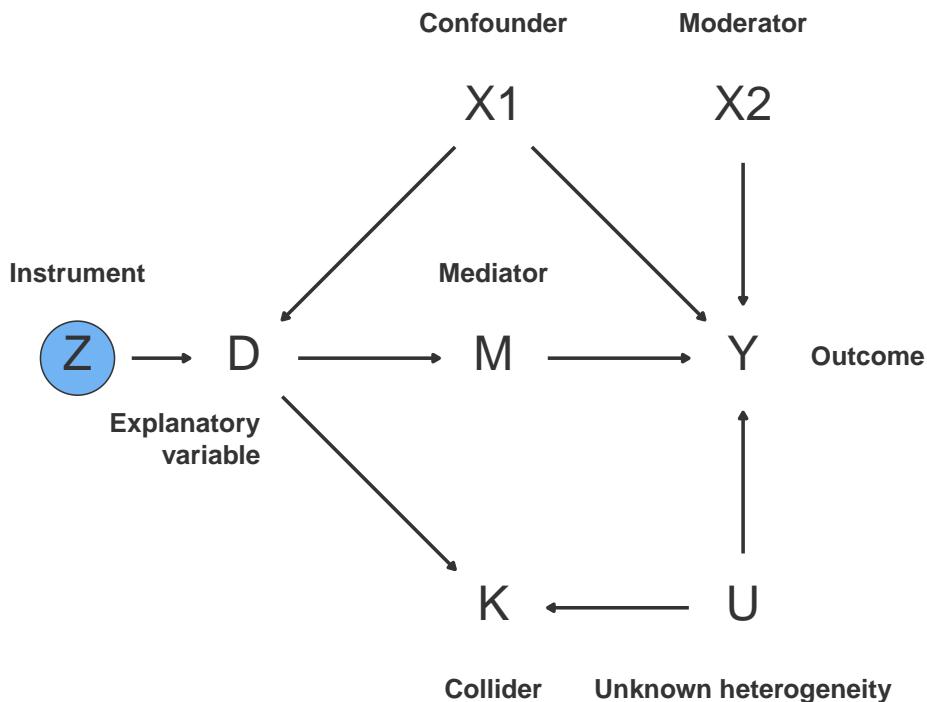


Figure 6.2: A DAG with an explanatory variable of interest (D), an outcome of interest (Y), a mediator (M), a confounder (X1), a moderator (X2), an instrument (Z), and a collider (K).

need to specify the relevant potential outcomes and the covariate that describes the subset.

Data strategy: Sampling procedures often involve stratification or clustering, so in the model, we need to define the variables that will be used to stratify and cluster. Similarly, treatment assignment might be blocked or clustered; the variables that are used to construct blocks or clusters must be defined in the model. Finally, all of the variables that will be measured should also be defined in the model. When we measure latent variables imperfectly, the model describes the latent trait and how measured responses may deviate from it.

Answer strategy: Any measured variable that will be used in the answer strategy should be included in the model. This clearly includes the observed outcomes and treatments, but also the covariates that are used to address confounding or to increase precision.

The variables required by the inquiry, data strategy, and answer strategy are necessary components of the model, but they are not always sufficient. For example, we might be worried about an unobserved confounder. Such a confounder would not be obviously included in any of the other research design

elements but is clearly important to include in the model. Ultimately, we need to specify all variables that are required for diagnosand completeness (see Section 10.5), which is achieved when research designs are described in sufficient detail that diagnosands can be calculated.

6.3 Substantive justifications for choices

To this point, we have described formal considerations but we have not described substantive considerations for including particular variables or stipulating particular relations between them. The justification for your choice of reference model will depend on the purpose of your design. Broadly we distinguish two desiderata: reality tracking and agnosticism.

6.3.1 Reality tracking

In stipulating reference models we have incentives to focus on models that we think reasonably track reality (m) as well as possible. Why waste resources stipulating processes that do not characterize those that you expect to be in operation?

The justification for reality tracking typically comes from two places: reading the past literature and qualitative research. Past theoretical work can guide the set of variables that are relevant and how they relate to one another. Past empirical work can provide further insight into the distributions and dependencies across variables. However, when past research is thin on a topic, there is no substitute for insights gained through qualitative data collection: focus groups and interviews with key informants who know aspects of the model that are hidden from the researcher, archival investigations to understand a causal process, or immersive participant observation to see with your own eyes how social actors behave. Fenno (1978) calls this “soaking and poking.” This mode of model development is separate from the qualitative research designs that provide an answer to an inquiry deductively. We examine those throughout the book. Instead, qualitative insights such as this, which Lieberman (2005) labels “model-building” case studies, do not aim to answer a question but rather yield a new theoretical model. Quantitative research is often seen as distinct from qualitative research, but the model building phase in both is itself qualitative.

The next step — selecting statistical distributions and their parameters to describe exogenous variables and the functional forms of endogenous variables — is often more uncomfortable. We do not know the magnitude of the effect of an intervention or the correlation between two outcomes before we do the research, that’s why we are conducting the study. However, we are not fully in the dark in most cases and can make educated guesses about most parameters.

We can conduct meta-analyses of past relevant studies on the same topic to identify the range of plausible effect sizes, intraclass correlations, correlations between variables, and other model parameters. Conducting such a meta-analysis

might be as simple as collecting the three papers that measured similar outcomes in the past and calculating the intraclass correlations across the three. How informative past studies are for your research setting may depend on the similarity of units, treatments, and outcomes across contexts. Except in the case of pure replication studies, we are typically studying a (possibly new) treatment in a new setting, with new participants, or with new outcomes, so there will not be perfect overlap. However, the variation in effects across contexts and these other dimensions will help structure the range of our guesses specified in the model.

When there are past studies that are especially close to our own, we may want our model to match the observed empirical distribution from that past study as closely as possible. To do so, we can resample or bootstrap from the past data in order to simulate realistic data. Where there are no past studies that are sufficiently similar in some dimensions, we can collect new data through pilot studies (see Section 21.5) or baseline surveys to serve a similar purpose.

Since it excludes cases we deem improbable, a focus on reality tracking models seems to contradict Principle 3.3: Entertain many models. However, by focusing on reality-tracking models, we aim to contain the smallest set of plausible models that contain the true one. In practice of course we might never include m . For instance we might contemplate a set of worlds in which an effect lies between 0 and 1 yet not include the true value of 2. This is not necessarily a cause for concern. The lessons learned from a diagnosis do not depend on the realized world m being among the set of possible draws of M , the relevant question is only whether the kinds of inferences one might draw given stipulated reference models would also hold reasonably well for the true data generating process. For instance, if your aim is to assess whether an analysis strategy generates an unbiased estimate of a treatment effect you may go to pains to make sure that that you model treatment assignment carefully but modeling the size of a treatment effect correctly may not be important. The idea is that what you learn from the models that you do study is *sufficient* for inferences about a broader class of models within which the true data generating process might lie.

6.3.2 Agnosticism

For some purposes, the reference model might be developed not to track reality, as you see it, but rather to reflect assumptions in a scholarly debate. For instance, the purpose might be to question whether a given conclusion is valid *under the assumptions maintained by some scholarly community*. Indeed it is possible that a reference model is used specifically because the researcher thinks it is inaccurate, allowing them to show that even if they are wrong about some assumptions about the world in M , their analysis will produce useful answers.

In a directed acyclic graph, every arrow indicates a possible relation between a cause and an outcome. The big assumptions in these models, however, are not seen in the arrows but the absence of arrows: every missing arrow represents a

claim that an outcome is not affected by a possible cause. Analysis strategies often depend upon such assumptions. Even when arrows are included, functional relations might presuppose particular features important for inference. For instance, a researcher using instrumental variables analysis (see Section 15.4) will generally assume that Z causes Y through D but not through other paths. This “excludability” assumption is about absent arrows. The same analysis might also assume that Z never affects D negatively. That “monotonicity” assumption is about functional forms. An agnostic reference model might loosen these assumptions to allow for possible violations of the excludability or monotonicity assumptions.

If we are agnostic, it is because we don’t know whether the truth is in the set of models we consider – so we entertain a wider set than the we might think plausible. We suggest three guides for choosing these ranges: the logical minimum and maximum bounds of a parameter, a meta-analytic summary of past studies, or best- and worst-case bounds, based on the substantive interpretations of previous work. A design that performs well in terms of power and bias under many such ranges might be labeled “robust to multiple models.”

A separate goal is assessing the performance of a research design under different models implied by alternative theories. A good design will provide probative evidence about which model is correct no matter the underlying state. A poor design might only affirm one model when it is true but fail to provide support for an alternative when *it* is true.

An important example is assessing the performance of a research design under a “null model” where the true effect size is zero. A good research design should report with a high probability that there is insufficient evidence to reject a null effect. That same research design, under an alternative model with a large effect size, should with a high probability return evidence rejecting the null hypothesis of zero effect. The example makes clear that in order to understand whether the research design is strong, we need to understand how it performs under the models implied by alternative theoretical understandings of the world.

6.4 Declaring models in code

In this section, we describe how to declare models in practice in the `DeclareDesign` code language. We start with declarations of units and the hierarchical structures that contain them, then move on to declarations of the characteristics of units. An important feature of models is the set of potential outcomes associated with each unit, so we spend some time describing a few approaches for thinking about them. This section is meant as a reference guide so it covers common settings and a few uncommon ones as well.

6.4.1 Units

The model is first defined by the units under study. If there are 1,000 people who live in a city that you wish to learn about, but you don't know anything else about them, you can declare:

```
M <- declare_model(N = 1000)
```

Units often sit within multiple, sometimes overlapping geographic and social hierarchies. Households live on blocks that make up neighborhoods. Workers have jobs at firms and also often are represented by unions that sometimes represent workers in multiple firms (a nonnested hierarchy). These hierarchies are important to declare in the model as they often form the basis for why units are similar and different. Within `declare_model`, we define hierarchy using `add_level`, which adds a new level of hierarchy. This model declaration creates 100 households of varying size, then creates the appropriate number of individuals within households.

```
M <- declare_model(
  households = add_level(
    N = 100,
    N_members = sample(c(1, 2, 3, 4), N,
                       prob = c(0.2, 0.3, 0.25, 0.25), replace = TRUE)
  ),
  individuals = add_level(
    N = N_members,
    age = sample(18:90, N, replace = TRUE)
  )
)
```

Panel data have a different structure. For example, in a country-year panel dataset, we observe every country in every year. To create data with this structure, we first declare a country-level dataset, then a years-level dataset, then we join them. The join is accomplished in `cross_levels` call, which defines the variables we join by with `by = join(countries, years)`. In `cross_levels`, we also create the observation-level outcome variable, which is a function in this case of a country shock, a year shock, an observation shock, and a time trend.

```
M <- declare_model(
  countries = add_level(
```

```

N = 196,
country_shock = rnorm(N)
),
years = add_level(
  N = 100,
  time_trend = 1:N,
  year_shock = runif(N, 1, 10),
  nest = FALSE
),
observation = cross_levels(
  by = join(countries, years),
  observation_shock = rnorm(N),
  Y = 0.01 * time_trend + country_shock + year_shock + observation_shock
)
)

```

6.4.2 Unit characteristics

We can describe the characteristics of units in two ways: we can use existing data or we can simulate.

Here is an example of the simulation approach. We imagine 100 units with a characteristic X that is uniformly distributed between 0 and 100.

```

M <-
declare_model(
  N = 100,
  X = runif(N, min = 0, max = 100)
)

```

You can use any of the enormous number of data simulation functions available in R for this purpose. Here we gather six functions we tend to use in our own declarations, but they are by no means exhaustive. Each function has arguments that govern exactly how the data are created; we chose arbitrary values here to show how they work. Figure 6.3 shows what these six look like for a 1,000 unit model.

```

M <-
declare_model(
  N = 1000,

```

```
X1 = rnorm(N, mean = 5, sd = 2),
X2 = runif(N, min = 0, max = 5),
X3 = rbinom(N, size = 1, prob = 0.5),
X4 = rbinom(N, size = 5, prob = 0.5),
X5 = rlnorm(N, meanlog = 0, sdlog = 1),
X6 = sample(c(1, 2, 3, 4, 5), N, replace = TRUE)
)
```

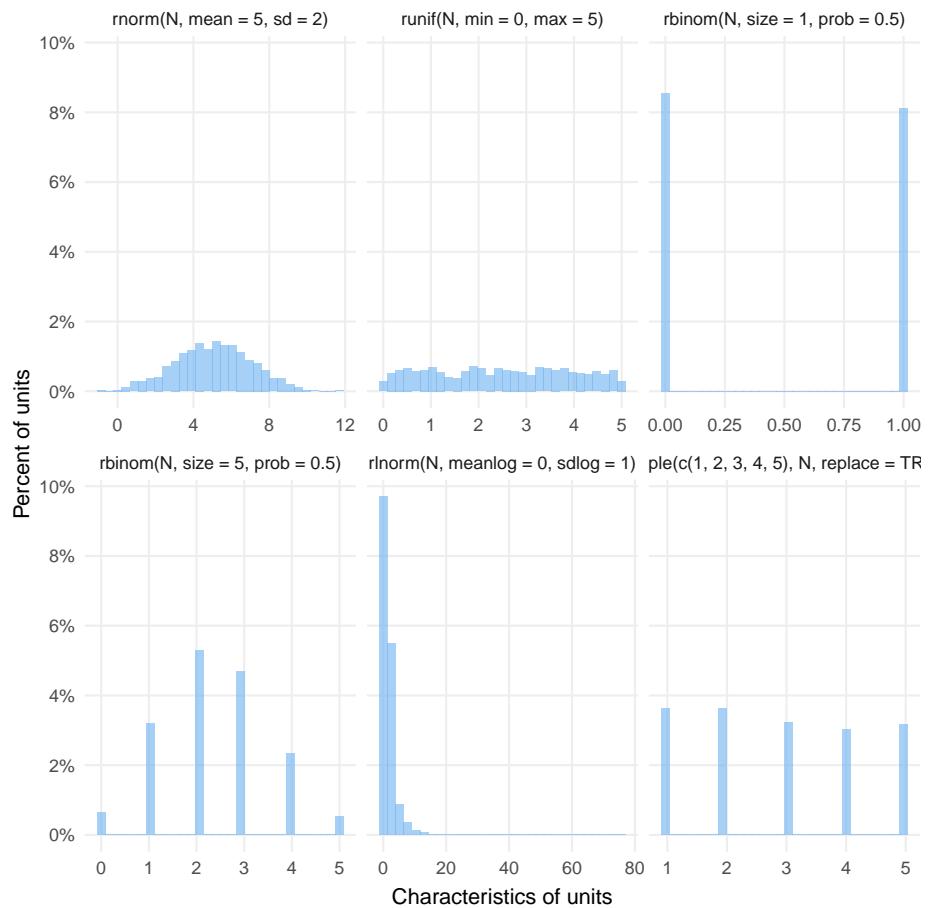


Figure 6.3: Six kinds of characteristics of units

Binary variables are very important in social science, but they can be particularly tricky to make, so we'll spend a little more time on them. In a common way of thinking, binary variables are translations from a latent, continuous variable into a observed binary outcomes.

We can draw binary outcomes in one of three ways. First, we can simulate a binary outcomes using the `rbinom` function as we have already seen. The function `rbinom(N, size = 1, prob = 0.5)` flips 1 coin for each of N subjects with a constant latent probability of success across units.

```
M1 <-
  declare_model(
    N = 1000,
    Y = rbinom(N, 1, prob = 0.5)
  )
```

If you believe the latent probability varies across units, you might want to set up latent variable first before the call to `rbinom`. A major reason to do it this way is to build in correlation between the binary outcome Y and some other variable, like Y2 in this model declaration.

```
M2 <-
  declare_model(
    N = 1000,
    latent = runif(N, min = 0, max = 1),
    Y = rbinom(N, 1, prob = latent),
    Y2 = latent + rnorm(N)
  )
```

A third way to create binary variable skips the call to `rbinom` altogether and creates a binary variable by assessing whether the latent variable exceeds some threshold, here 0.75. A major reason to do this is when we want to control the sources of randomness. The latent variable is random because of the call to the `runif` function. If we pass the resulting latent probabilities to `rbinom` as in M2, then we add a second layer of randomness, the coin flips. If one layer is enough, then M3 might be a more appropriate model declaration.

```
M3 <-
  declare_model(
    N = 1000,
    latent = runif(N, min = 0, max = 1),
    Y = if_else(latent > 0.75, 1, 0)
  )
```

6.4.2.1 Building in correlations between simulated variables

Most social science variables are interrelated. The correlations between variables may affect the quality of a research design in many ways. If we control for a variable in a regression to improve power, how much power we gain depends on the correlation between that variable and the outcome.

Here we walk through two main ways to create correlated variables. The first way is simply to make one variable a function of another:

```
M1 <-
  declare_model(
    N = 1000,
    X1 = rnorm(N),
    X2 = X1 + rnorm(N)
  )
```

The second way draws on an explicit draw from a *multivariate* distribution. The `mvrnorm` function in the MASS package generates draws from the multivariate normal distribution. We have to give it two means (`mu`) and a variance-covariance matrix (`Sigma`). The `draw_multivariate` function is a neat wrapper that makes these functions (that return more than one column of data!) play nicely with `declare_model`.

```
M2 <-
  declare_model(
    draw_multivariate(c(X1, X2) ~ MASS::mvrnorm(
      N, mu = c(0, 0),
      Sigma = matrix(c(1, 0.3, 0.3, 1), nrow = 2)))
  )
```

6.4.2.2 Building in correlations within clusters

A second important form of correlation is correlation within clusters, described by the intraclass correlation coefficient (ICC). When ICC is low, the within-cluster differences are similar across clusters. When it is high, clusters are more homogeneous within themselves and more heterogeneous across themselves. For more on clustered designs for surveys and for experiments, see Section 14.2 and Section 17.3.

We can introduce ICC using the `draw_normal_icc` function:

```
M <-
  declare_model(households = add_level(N = 1000),
                individuals = add_level(
                  N = 4,
                  X = draw_normal_icc(
                    mean = 0,
                    clusters = households,
                    ICC = 0.65
                  )
                ))
```

6.4.2.3 Unknown heterogeneity

Most declarations include unobserved variables, or unknown heterogeneity. These Us represent the lurking variables that confound our inferences and the variation in outcomes not correlated with observed values. In virtually every declaration in the book, we include a U term to represent these unobserved values.s

But now that we are at the point of declaration, of actually writing down a design in code, we face problems immediately: how much heterogeneity do we introduce, where, and of what kind?

M1 shows how to make rather benign unknown heterogeneity. U is normally distributed and only affects Y. The observed binary variable X is independent of U and affects Y in its own way.

```
M1 <- declare_model(
  N = 100,
  U = rnorm(N),
  X = rbinom(N, size = 1, prob = 0.5),
  Y = 0.1 * D + U
)
```

M2 is more worrisome. U now affects X as well, by affecting the probability of X equaling one. See Section 15.2 for designs that face this sort of problem.

```
M2 <- declare_model(
  N = 100,
  U = rnorm(N),
```

```
X = rbinom(N, size = 1, prob = pnorm(U)),
Y = 0.1 * D + U
)
```

A further question for U is how much it should vary. This question is tougher to answer. U is an important source of variation in outcomes, so it needs to be calibrated in a way that the simulated data look like the data you expect to generate or encounter in the world. So our best advice here is to follow Principle 3.3: entertain many types of unknown heterogeneity and figure out which ones seem reasonable.

6.4.3 Potential outcomes

We declare “potential” outcomes when describing counterfactual quantities.

The most straightforward way to declare potential outcomes is to make one variable per potential outcome. Here the two potential outcomes come from coin flip distributions with different success probabilities.

```
M2 <-
declare_model(N = 100,
              Y_Z_0 = rbinom(N, size = 1, prob = 0.5),
              Y_Z_1 = rbinom(N, size = 1, prob = 0.6)
)
```

The `potential_outcomes` function can do the same thing, but using R’s “formula” syntax, allowing us to write down potential outcomes in a “regression-like” way.

```
M <-
declare_model(N = 100,
              potential_outcomes(Y ~ rbinom(N, size = 1, prob = 0.1 * Z + 0.5))
)
```

By default, `potential` imagines you are making two potential outcomes with respect to a treatment variable Z that can take on two values, 0 and 1.

But we can vary those to multiple treatment conditions:

Table 6.1: One draw of two potential outcomes

ID	Y_Z_0	Y_Z_1
001	0	0
002	1	0
003	0	0
004	1	1
005	0	0

Table 6.2: One draw of three potential outcomes

ID	Y_Z_0	Y_Z_1	Y_Z_2
001	0	0	0
002	0	0	1
003	0	0	0
004	0	0	0
005	0	0	0

```
M <-
  declare_model(
    N = 100,
    potential_outcomes(
      Y ~ rbinom(N, 1, prob = 0.1 * (Z == 1) + 0.2 * (Z == 2)),
      conditions = list(Z = c(0, 1, 2))
    )
  )
```

Or to multiple treatment factors (see Section 17.5 on factorial experiments):

```
M <-
  declare_model(
    N = 100,
    potential_outcomes(
      Y ~ rbinom(N, 1, prob = 0.1 * Z1 + 0.2 * Z2 + 0.1 * Z1 * Z2),
      conditions = list(Z1 = c(0, 1), Z2 = c(0, 1))
    )
  )
```

Table 6.3: One draw of four potential outcomes

ID	Y_Z1_0_Z2_0	Y_Z1_1_Z2_0	Y_Z1_0_Z2_1	Y_Z1_1_Z2_1
001	0	0	0	1
002	0	0	0	1
003	0	0	0	1
004	0	0	0	1
005	0	0	0	1

6.4.3.1 Effect sizes

We often want to consider a range of plausible effect sizes, for example when estimating the minimum detectable effect of a design. A strategy we commonly use is to sample the treatment effect in the model declaration itself, and then draw the potential outcomes using that single number. When we diagnose many times, then we will get many different treatment effects (here, τ), which we can then summarize in a diagnosis. Section 10.8 describes how to create a plot of the power across values of τ .

```
M <-
  declare_model(
    N = 100,
    tau = runif(1, min = 0, max = 1),
    U = rnorm(N),
    potential_outcomes(Y ~ tau * Z + U)
  )
```

Where do our expectations about effect sizes, and the minimal plausible effect size, come from? We may conduct meta-analysis of past studies when there are more than one sufficiently relevant estimates of the same effect, or a systematic review or literature review when they are less comparable but we may want to find a range (see Section 18.4 for a discussion of meta-analysis). We may be tempted to conduct a pilot study to estimate the effect size. We need to be careful when doing so what to infer and how much to update from small pilot studies, as we discuss in Section 21.5, but we can often shrink our uncertainty about them. In the absence of pilot studies or past studies to draw on, we need to make educated guesses and understand under what true effect sizes our design will perform well and when it will not following Principle 3.10 (Diagnose to break designs).

6.4.3.2 Effect heterogeneity

Sometimes, the inquiry centers on treatment effect heterogeneity by subgroups. This heterogeneity has to be present in the model in order for the simulation to pick it up. Here we declare effect heterogeneity according to a binary covariate X . This example really shows off the utility of the formula syntax in the `potential_outcomes` function. We can write in our expectations about heterogeneity as if they were regression coefficients. Here, the “interaction term” is equal to 0.1.

```
M <-  
  declare_model(  
    N = 100,  
    U = rnorm(N),  
    X = rbinom(N, 1, prob = 0.5),  
    potential_outcomes(Y ~ 0.3 * Z + 0.2*X + 0.1*Z*X + U)  
)
```

6.4.3.3 Correlation between potential outcomes

Treated and untreated potential outcomes are typically highly correlated. When there is no treatment effect heterogeneity, there is often perfect correlation between the two; the only difference is the shift due to the treatment effect. Rarely are potential outcomes *negatively* correlated, but it can occur. The sign and magnitude of the correlation especially affects the standard errors of estimates for causal effects (for more discussion of the correlation of potential outcomes in experiments and also standard error estimators, see Section 17.1.1).

We described some complexities of generating binary variables above. They transfer over to the generation of correlated potential outcomes in special ways. In the declarations below, M1 generates uncorrelated potential outcomes, because the draws from `rbinom` are independent of one another. In M2, we still use `rbinom`, but with a transformation of the normally-distributed variable into a probability via `pnorm`. This allows the potential outcomes to be correlated because they are both influenced by the same latent variable. Finally, in M3, we generate highly correlated potential outcomes because we peel off the layer of randomness introduced by `rbinom`.

```
M1 <-  
  declare_model(  
    N = 100,  
    potential_outcomes(Y ~ rbinom(N, 1, prob = 0.2))  
)
```

```
M2 <-
  declare_model(
    N = 100,
    latent = rnorm(N),
    potential_outcomes(Y ~ rbinom(N, 1, prob = pnorm(latent + 0.2 * Z)))
  )

M3 <-
  declare_model(
    N = 100,
    latent = rnorm(N),
    potential_outcomes(Y ~ if_else(latent + 0.2 * Z > 0.5, 1, 0))
  )
```

6.4.4 Summary

If this section left you spinning from the array of choices we have to make in declaring a model, in some ways that was our goal. Inside every power calculator and bespoke design simulation code are an array of assumptions. Some crucially determine design quality. Others are unimportant. The salve to the dizziness is Principle 3.3: entertain many models. Where you are uncertain, explore whether both options produce the same diagnosands. The goal is for your data and answer strategy to hold up to many models, and to find out whether it does, you often need to build the many options into your model.

Chapter 7

Defining the inquiry

An inquiry is a question you ask of a model. If we stipulate a reference model, m , then our inquiry is a summary of m . Suppose in some reference model that X possibly affects Y . Using the framework provided in Pearl and Mackenzie (2018), one inquiry might be descriptive, or associational: what is the average level of Y when $X = 1$? A second might be about the effects of interventions: what is the average treatment effect of X on Y ? A third is about counterfactuals: for what share of units would Y have been different if X were different? If a theory involves more variables, many more questions open up, for instance regarding how the effect of one variable passes through, or is modified by, another.

When designing research, you should have your inquiries front of mind. Amazingly, many research projects do not: sometimes researchers start implementing data and answer strategies without any particular goal in mind, then end up discovering a question to answer in the process of generating estimates. To be clear, we are all for the kind of model-building research that helps us understand models well enough to even state a worthwhile inquiry. We also think that it's possible to learn unexpected things in the course of doing research. But it's essentially not possible to proactively design research to answer a question well unless we have an inquiry to target.

Formally, an inquiry is a summary function I that operates on an instance of a model m M . When we summarize the model with the inquiry, we obtain an “answer under the model.” We formalized this as $I(m) = a_m$. You can think of the difference between I and a_m as the difference between a question and its answer. I is the question we ask about the model and a_m is the answer.

In this book when we talk about inquiries, we will usually be referring to single-number summaries of models. Some common inquiries are descriptive, such as the means, conditional means, correlations, partial correlations, quantiles, and truth statements about variables in the model. Others are causal, such as the

average difference in one variable when a second variable is set to two different values. We can think of a single-number inquiry as the atom of a research question.

While most inquiries are “atomic” in this way, some inquiries are more complex than a single-number summary. For example, the best linear predictor of Y given X is a two-number summary: it is the pair of numbers (the slope and intercept) that minimizes the total squared distance between the line and each value of Y . No need to stop at two-number summaries though. We could imagine the best quadratic predictor of Y given X (three-number summary), and so on. We could have an inquiry that is the full conditional expectation function of Y given X , no matter how wiggly, nonlinear, and nuanced the shape of that function. It could in principle be a 1,000 number summary of the model, or much more.

The inquiry could be constituted by a series of interrelated questions about the model. For instance, a researcher might articulate a handful of important questions about the model that all have to come out a certain way or the model itself should be rejected. These complex inquiries are made up of a series of atomic inquiries. We’re interested in the sub-inquiries only insofar as they help us understand the real inquiry – is this model of the world a good one or not.

7.1 Elements

Every inquiry operates on the events generated by the model. We can think of the events as the “data”-set that describes the units, treatment conditions, and outcome variables over which inquiries can be defined. This definition is closely connected to the common UTOS (units, treatments, outcomes, and settings) framework (Shadish, Cook and Campbell, 2002). The units are the set of units within the model that the inquiry refers to, either all or a subset. The treatment conditions represents the set chosen for study. A descriptive inquiry is a summary of a single condition (reality), whereas a causal inquiry is a summary of multiple conditions. The outcomes are the set of nodes in the model that the inquiry concerns. Finally, the inquiry operates on the model events via a summary function. For example, the “population average” inquiry summarizes the outcome for all units in the population with the mean function. We discuss each element of inquiries in turn.

7.1.1 Units

The units of an inquiry defined by the set of people, places, or things that we are interested in studying. A study’s units might be the counties in Alabama, the set of students enrolled in Los Angeles Unified School System this March, countries in the world with mean income under \$100 per day, or the houses in the Westlands neighborhood in Nairobi, Kenya. In both descriptive and causal inquiries, units may be all of the units or a subset of them, defined for example

as those selected by a sampling procedure or those with a specific characteristic. In causal inquiries, the units may be those who *are* treated, who *are not* treated, or those who comply with treatments, again either in the entire population or a subset.

The reason we need to define the units of an inquiry is inquiry values may differ across units. If the units that are included in the sample live in easier-to-reach areas and people who live in easier-to-reach areas are wealthier than others, the sample average will differ from the population average — and from the average among those in hard-to-reach places.

The choice of which set of units to focus on is a theoretical one. To whom does the theoretical expectation apply? As a general matter, seeking insights that apply across many individuals is the goal of many social scientists. We are not typically interested in the effect of a treatment or the average outcome in a random sample of 100 units because we care about those units in particular, but because we wish to understand the treatment effect or outcomes in a broader population. Our theories often have so-called scope conditions, which define the types of units for which our theory is operative. A mechanism might operate only for coethnics of a country's president, small-to-medium towns, blue collar workers, or the mothers of daughters. The units of an inquiry should be defined by these theoretical expectations, not by what inquiries our data and answer strategies can target easily.

This distinction often arises in debates over instrumental variables designs, which target local average treatment effects (LATEs), meaning the average treatment among a subset. The effect these designs estimate is the average treatment effect among those units who are “compliers”. Compliers are the subset of units who take treatment if assigned and don't take treatment if not assigned. The effect among compliers may or may not be like the effect among the whole sample or the population from which the sample was drawn. The debate between Deaton (2010) and Imbens (2010) centers precisely on which inquiry is the appropriate one, the LATE among compliers or the ATE in the whole sample. In many settings, the LATE may be the only inquiry we can reliably estimate, so the question becomes – is the LATE an inquiry with theoretical relevance?

If the inquiry is defined with respect to the units sampled by the data strategy, then we do not have to engage in generalization inference – we learn directly about the sample from the sample. But if the inquiry is defined population-level, then we need to generalize from the sample to the population. We also need to engage in generalization inference when we want to generalize study results to *other* populations that we did not explicitly sample from. Whether an inquiry requires generalization inference depends on the data strategy in this way. If the data strategy samples the units that define the inquiry, we do not need to generalize beyond the study. If the data strategy explicitly samples from a well-defined population, we can generalize from sample to population using canonical sampling theory. But if we want to generalize to an inquiry

defined over some other set of units (for example, Brazilian citizens ten years in the future), we need to engage in generalization inference (See Egami and Hartman (2021)).

7.1.2 Outcomes

Every inquiry is also defined by what outcomes are considered for each of the units. The choice of outcome is again a theoretical one: what outcomes are to be described, or with which outcomes do we want to measure the effects of treatment? An inquiry might be about a single outcome or multiple outcomes. The average belief that climate change is real would be a single-outcome inquiry, and the difference between that belief and support for government rebates for purchasing electric vehicles a multiple-outcome inquiry.

In some cases, an inquiry will be about a latent outcome that we cannot directly measure, such as preferences, attitudes, or emotions. We can construct data strategies that elicit these latent outcomes using observable measures from asking or observing individuals, but we cannot directly measure them. Even though these constructs may be difficult or impossible to measure well, it is preferable to define the inquiry in terms of the true latent outcome of interest so we can later evaluate how well we do.

7.1.3 Treatment conditions

The final element of an inquiry is the treatment conditions under consideration and, in the case of more than one, compared.

Descriptive inquiries are defined with respect to one single treatment condition. That treatment condition is usually the “unmanipulated” condition in which the researcher exposes units to no additional causal agents. Here the goal is not to learn about the summaries of the distributions of outcomes as we observe them. Table 7.1 (top panel) enumerates some common descriptive estimands. These estimands have in common that you do not need any counterfactual quantities in order to define them. The covariance (similarly, the correlation) between X and Y enters as a descriptive estimand, so too does the line of best fit for Y given X .

In Table 7.1, we enumerate several common types of descriptive inquiries, listing the units, treatment conditions, and outcomes that define them. We also provide R code snippets for each.

Table 7.1: Examples of descriptive inquiries and their three elements: units, treatment conditions, and outcomes.

Inquiry	Units	Treatment condi- tions	Outcomes
Average value of variable Y in a finite population	Units in the population	Unmanipulated	<code>mean(Y)</code>
Average value of variable Y in a sample	Sampled units	Unmanipulated	<code>mean(Y[S == 1])</code>
Conditional average value of Y given $X = 1$	Units for whom $X = 1$	Unmanipulated	<code>mean(Y[X == 1])</code>
The variance of Y	Units in the population	Unmanipulated	<code>pop.var(Y)</code>
The covariance of X and Y	Units in the population	Unmanipulated	<code>pop.cov(X, Y)</code>
The best linear predictor of Y given X	Units in the population	Unmanipulated	<code>cov(Y, X) / var(X)</code>
Conditional expectation function of Y given X	Units in the population	Unmanipulated	<code>cef(Y, X)</code>

Causal inquiries involve a comparison of at least two possible treatment conditions. For example, an inquiry might be the causal effect of X on Y for a single unit. In order to infer that causal effect, we would need to know the value of Y in two worlds: one world in which X is set to 1 and one in which X is set to 0. Table 7.2 (middle panel) enumerates some common causal estimands. These estimands vary in the population they refer to. For instance, some are questions about samples (SATEs) and others about populations (PATEs). Inquiries can also be defined for units of a particular covariate class (CATEs). Finally, they may be summaries of more than one potential outcome. For instance, the interaction effect is defined here at the individual level as the effect of one treatment on the effect of another treatment.

Table 7.2: Inquiries and their three elements: units, treatment conditions, and outcomes.

Inquiry	Units	Treatment conditions	Outcomes
Average treatment effect in a finite population (PATE)	Units in the population	$D = 0, D = 1$	$Y \cdot \text{mean}(Y_{D_1} - Y_{D_0})$
Conditional average treatment effect (CATE) for $X = 1$	Units for whom $X = 1$	$D = 0, D = 1$	$\text{mean}(Y_{D_1}[X == 1] - Y_{D_0}[X == 1])$
Complier average causal effect (CACE)	Complier units	$D = 0, D = 1$	$Y \cdot \text{mean}(Y_{D_1}[D_{Z_1} > D_{Z_0}] - Y_{D_0}[D_{Z_1} > D_{Z_0}])$
Causal interactions of D_1 and D_2	Units in the population	$D1 = 1, D1 = 0, D2 = 1, D2 = 0$	$Y \cdot \text{mean}((Y_{D1_1}D2_1 - Y_{D1_0}D2_1) - (Y_{D1_1}D2_0 - Y_{D1_0}D2_0))$

Generations of students have been told to excise words that connote causality from their empirical writing. “Affects” becomes “is associated with” and “impacts” becomes “moves with.” Being careful about causal language is of course very important (it’s really true that correlation does not imply causation!). But this change in language is not usually accompanied by a change in inquiry. Many times we are faced with drawing causal inferences from less than ideal data – but the deficiencies of the data strategy should not lead us too far away from our inferential targets. If the inquiry is a causal inquiry, then the move from “causes” to “is correlated with” might be a good description of the actual data analysis, but it doesn’t move us closer to providing an answer to the inquiry.

7.1.4 Summary functions

With the units, treatments, and outcomes specified, the last element of the inquiry is the summary function that is applied to them. For a great many inquiries, this function is the `mean` function: the ATE, the CATE, the LATE, the SATE, the population mean – these are all averages. These and other inquiries are decomposable in the sense that you can think of an average effect for a large group as being the average of a set of average effects of smaller groups.

However, not all inquiries are of this form. For example, the line of best fit is defined as the covariance of X and Y divided by the variance of X. This inquiry is a complex summary of all the units in the model.

The inquiry that the regression discontinuity design shoots at is also non-decomposable. In the RDD model (see Section 15.5), we imagine units with $Y_i(1)$, $Y_i(0)$. Each i also has a value on a “running variable”, X_i , and units receive treatment if and only if $X_i > 0$. In this case the “effect at the point of discontinuity” might be written:

$$E_{i|X_i=0}(Y_i(Z=1) - Y_i(Z=0))$$

Curiously, however, there may be no units for whom X_i equals exactly zero (a candidate who wins exactly 50% of the vote happens, but it is rare), so we cannot easily think of the inquiry as being a summary of individual potential outcomes. Instead, we construct a conditional expectation function for both potential outcome functions with respect to X and evaluate the difference between these when $X = 0$. Though not an average of individual effects, this difference is nevertheless a summary of the potential outcomes.

7.2 Examples of inquiries

The largest division in the typology of inquiries is between descriptive and causal inquiries. It is for this reason that Part III, the design library, is organized into descriptive and causal chapters, separated by whether the data strategy involves assignment. In this section, we describe other important ways inquiries vary and how to think about declaring them.

7.2.1 Data-dependent inquiries

The inquiries we have introduced thus far depend on variables in the model, but not on features of the data and answer strategies. However, common inquiries do depend on realizations of the research design.

The first type depends on realizations of the data d : inquiries about units within a sample depend on which units enter the sample; inquiries about treated units depend on which are treated. For example, the average treatment effect on the treated (ATT) is a data-dependent inquiry in the sense that it is the average effect of treatment among the particular set of units that happened to be randomly assigned to treatment. The value of that *particular* ATT doesn’t change depending on the data strategy, of course, but *which* ATT we end up estimating depends on the realization of the data strategy. Table 7.3 describes three data dependent inquiries

Table 7.3: Data-dependent inquiries.

Inquiry	Units	Treatment conditions	Outcome	Code
Average treatment effect in a sample (SATE)	Sampled units	$D = 0, D = 1$	Y	<code>mean(Y_D_1[S == 1] - Y_D_0[S == 1])</code>
Average treatment effect on the treated (ATT)	Treated units	$D = 0, D = 1$	Y	<code>mean(Y_D_1[D == 1] - Y_D_0[D == 1])</code>
Average treatment effect on the untreated (ATU)	Untreated units	$D = 0, D = 1$	Y	<code>mean(Y_D_1[D == 0] - Y_D_0[D == 0])</code>

7.2.2 Causal attribution inquiries

A causal attribution is a different kind of data-dependent inquiry. Whereas a causal effect inquiry focuses on the change in an outcome that would be induced by a change in the causal variable, irrespective of the values that the outcome takes in the realized data. By contrast, causal attribution inquiries focus on probabilities that condition on realized outcomes, such as, the “probability of the absence of the outcome in the hypothetical absence of the treatment ($Y_i(0) = 0$) given the actual presence of both ($D_i = Y_i = 1$)” (Yamamoto, 2012, pp.240-241). Goertz and Mahoney (2012) refers to causal attribution inquiries as cause-of-effects questions because they start with an outcome (an effect) and seek to validate a hypothesis about its cause.

The dependence of these inquiries on actual outcomes makes them harder (though not impossible!) to answer with the tools of quantitative science, though they are often of central interest to scientific and policy agendas and have occupied a large number of qualitative studies. Questions like “Was economic crisis necessary for democratization in the Southern Cone of Latin America?” or ‘Were high levels of foreign investment in combination with soft authoritarianism and export-oriented policies sufficient for the economic miracles in South Korea and Taiwan?’ are examples of such inquiries (Goertz and Mahoney, 2012). Though they bear a resemblance and *are* related to causal effects inquiries that focus on observed subsets (such as the average treatment effect on the treated, or ATT)¹ it is important not to confuse the two kinds of inquiries.

¹Specifically, as Yamamoto (2012) points out, the causal attribution estimand for binary variables can be written $\Pr(Y_i(0) = 0 | D_i = Y_i = 1)$, while the average treatment effect among those successfully treated can be written $E[Y_i(1) | Y_i(0) = D_i = Y_i = 1]$. Given binary outcomes and the additive property of expectations, the ATE among those successfully treated can be written $\Pr(Y_i(1) | D_i = Y_i = 1) \Pr(Y_i(0) | D_i = Y_i = 1)$. The causal attribution inquiry can be written as one minus the second term of the ATE among the successfully treated.

While it is increasingly common to explicitly formalize causal effect inquiries, it is less common to formalize causal attribution inquiries. Doing so, however, can be important to provide the specificity required to diagnose a design on a computer. Pearl (1999) provides formal definitions for these inquiries using the language of causal necessity and sufficiency, depicted in the table below. To put these inquiries in the context of the democratic peace hypothesis, for example, in a given country dyad-year, $Y_i = 1$ and $D_i = 1$ could represent “Peace” and “Both democracies” and $Y_i = 0$ and $D_i = 0$ could represent “War” and “Not both democracies.” Then $\Pr(Y_i(D_i = 0) = 0 | D_i = Y_i = 1)$ asks, among peaceful, fully democratic dyads, what is the proportion that would have had wars were they not both democracies—that is, in what proportion of dyad-years was democracy a necessary cause of peace? Similarly, $\Pr(Y_i(D_i = 1) = 1 | D_i = Y_i = 0)$ asks, among dyads that had a war and at least one non-democracy in a given year, what is the proportion that would have experienced peace if both countries were democracies—in other words, in what proportion of cases would democracy have been sufficient to cause peace? Yamamoto (2012) extends on this account to focus on causal attribution inquiries that focus on important subsets, such as compliers.

Like all designs, those with causal attribution inquiries can be declared, simulated, and diagnosed on a computer. Something to consider, however, is that the model may produce datasets in which the effect does not occur, and so questions defined over units for whom it occurred are undefined. One way to avoid this is to construct a model such that the event occurs for at least one unit with probability one.

Table 7.4: Causal attribution inquiries.

Inquiry	Units	Treatment condi- tions	Outcome code
Probability D necessary for Y	Units for whom $D = 1$ and $Y = 1$	$D = 0$	$Y \& \text{mean}(Y_{_D_0}[D == 1 \& Y == 1] == 0)$
Probability D sufficient for Y	Units for whom $D = 0$ and $Y = 0$	$D = 1$	$Y \& \text{mean}(Y_{_D_1}[D == 0 \& Y == 0] == 1)$
Complier probability D necessary for Y	Units for whom $D = 1$ and $Y = 1$, $D = 0$ and $Y = 0$ who are compliers	$D = 0, Z = 1, D = 1, Z = 0$	$Y \& \text{mean}(Y_{_D_0}[D == 1 \& Y == 1 \& D_{_Z_1} == 1 \& D_{_Z_0} == 0] == 0)$

7.2.3 Complex counterfactual inquiries

Thus far, the causal inquiries we have considered have involved comparisons of the counterfactual values an outcome could take, depending on the value of

one or more treatment variables. These inquiries are mind-bending in that we have to imagine two counterfactual states at the same time. Complex counterfactual inquiries require more mind bending still.

An example is the “controlled direct effect.” Suppose our model contains a treatment Z , a mediator M , and outcome Y . The controlled direct effect of the treatment is defined as:

$$\text{CDE} = Y(Z = 1, M = 1) - Y(Z = 0, M = 1)$$

So far so good. but suppose now we stipulate that at least for some units $M = 1$ only when $Z = 1$, but it equals 0 when $Z = 0$ In order to imagine the CDE, we have to hold in our minds the complex counterfactual: what is the level of Y , when Z equals 1, but M is at the value it would take if Z equalled one.

7.2.4 Inquiries with continuous causal variables

We have mainly considered causal inquiries that compare across discrete treatment conditions Treatment versus control, or one of many arms in a multi-arm trial.

But sometimes, we can imagine a continuous treatment space. For example, we could think of the effects of any level of salary from 5 dollars an hour to 500 dollars an hour on workplace satisfaction. We could “discretize” these continuous treatment in bins, in which case we are back to defining inquiries as we have for multi-arm trials with discrete treatment conditions. Alternative, we could describe the estimand as the average of the slopes from many lines of best fit. For each subject, we describe the line of best fit of the outcome with respect to the treatment. Our inquiry is then the average of the resulting slopes. This inquiry is decomposable in the sense that it is the average of many slopes, but it differs from the other causal inquiries in that it is not a direct contrast between a pair of conditions, it is a description of difference across a continuum of conditions.

7.3 How to choose among inquiries

It’s hard to know where to start when choosing an inquiry. We want to pick one that is interesting in its own right or one that would facilitate a real-world decision. We want to pick research questions that we can learn the answer to someday, possibly with a lot of effort. Unfeasible research questions should be abandoned as soon as possible, but of course, that’s hard to do. The trouble is that it’s hard to know what research questions are feasible before you start looking into it, and it’s hard to quit research projects once you learn they are unfeasible. Among feasible research questions, we want to select ones that we are likely to obtain the most informative answers, in terms of moving our priors the most.

Sometimes, people advise students to follow a “theory-first” route to picking a research question. Read the literature, find an unsolved puzzle, then start choosing among the methodological approaches that might answer the problem. Others eschew the theory-first approach: “How on earth are you going to happen to land upon an unsolved – and yet somehow solvable – puzzle just by reading!?” These advice-givers emphasize a method-first route. Master the technical data-gathering and analysis procedures first, then set off to find opportunities to apply them. The theory-first people then say: “how would you know an interesting theoretical question if it smacked you in the face!?”

Iteration between the two is typically necessary. In order to select inquiries, empirical researchers have to be concerned about the entire research design. We have to learn a lot about how to select data and answer strategies in ways that map on to inquiries about models. So empiricists have to learn both about models and inquiries (theory) as well as about data strategies and answer strategies (empirics).

The first criterion is the subjective importance of a question. The object of the importance may be a scientist, considering the value of building a theoretical understanding of the world; to a policymaker, deciding how to collect and allocate resources in a government; a private firm, who is making decisions about how to invest their own resources to maximize profit; or another individual or organization. The scientific enterprise is designed around the idea that importance is in the eye of the beholder and is not some objective quantity. This is for two reasons. First, the scientific or practical importance of a discovery may not be understood until decades later, when other pieces of the causal model are put together or the world faces new problems. Moreover, “importance” differs for different segments of society, and scientists must be able to study questions not judged important by groups in power in order to discover new ways to solve problems faced by the left-out groups.

A second important criterion flows from Principle 3.4: Select answerable inquiries. How could an inquiry *not* be answerable? The main way is if we can’t find a feasible data or answer strategy. When for ethical, legal, logistic, or financial constraints, we simply can’t conduct the study, the inquiry is not answerable.

There are subtler ways in which an inquiry might not be answerable. For example, it might be undefined. Inquiries are undefined when I returns $I(m) = a_m = \text{NA}$. Sometimes audit studies consider the effect of treatment on responding to an email and on the tone of the email. However, in conditions where the email is never sent, it has no tone. As a result, we can’t learn about the average effect of treatment on tone, we can only learn about the effect in a subgroup: those units who always respond to email, regardless of condition. This new inquiry is defined, but hard to estimate (see Coppock (2019))

An inquiry is also not answerable if it is not “identified.” Identification means: a question is at least partly answerable if there are at least two different sets of

data you might observe that would lead you to make two different inferences. In the best case, one might imagine that you have lots of data and each possible data pattern you see is consistent with only one possible answer. You might then say that your model, or inquiry, is identified. Failing that you might imagine that different data patterns at least let you rule out some answers even though you can't be sure of the right answer. In this case we have "partial identification." Some inquiries might not even be partially identifiable. For instance if we have a model that says an outcome Y is defined by the equation $Y = (a + b)X$, no amount of data can tell us the exact values of a and b . Indeed without limits on the values of a and b (such as $a \neq 0$), no amount of data can even narrow down the ranges of a and b . The basic problem is that for any value of a we can choose a b that keeps the sum of $a + b$ constant. In this setting, even though there is an answer to our inquiry (a) in theory it is not one we can ever answer in practice. Many other types of inquiries, such as mediation inquiries, are not identifiable. There are some circumstances in which we can provide a partial answer to the inquiry, such as learning a range of values within which the parameter lives. At a minimum, we urge you to pose inquiries that are at least partially answerable with possible data.

7.4 Declaring inquiries in code

An inquiry is a summary function of events generated by a model. When we declare inquiries in code, we declare this summary function. Here, we declare a causal inquiry, the `mean` of the differences in two potential outcomes described in the model:

```
M <- declare_model(N = 100, U = rnorm(N), potential_outcomes(Y ~ Z + U))
I <- declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))
```

Descriptive inquiries can be declared in a similar way: they are just functions of outcomes rather than potential outcomes.

```
M <- declare_model(N = 100, Y = rnorm(N))
I <- declare_inquiry(mean_Y = mean(Y))
```

7.4.1 Inquiries among subsets of units

We often want to learn about an inquiry defined among a subgroup of units. For example, if we are interested in the conditional average treatment effect (CATE) among units with $X = 1$, we can use the subset argument.

```
M <- declare_model(
  N = 100,
  U = rnorm(N),
  X = rbinom(N, 1, prob = 0.5),
  potential_outcomes(Y ~ 0.3 * Z + 0.2*X + 0.1*Z*X + U))
I <- declare_inquiry(CATE = mean(Y_Z_1 - Y_Z_0), subset = X == 1)
```

Equivalently, we could use R's [] syntax for subsetting:

```
I <- declare_inquiry(CATE = mean(Y_Z_1[X == 1] - Y_Z_0[X == 1]))
```

7.4.2 Inquiries with continuous potential outcomes

“Non-decomposable” inquiries are not as simple as an average over the units in the model. A common example arises with continuous potential outcomes. The regression discontinuity design described in Section 15.5 has an inquiry that is defined by two continuous functions of the running variable. The control function is a polynomial function representing the potential outcome under control and the treatment function is a different polynomial representing treated potential outcomes. The inquiry is the difference in the two functions evaluated at the cut-off point on the running variable. We declare it as follows:

```
cutoff <- 0.5
control <- function(X) {
  as.vector(poly(X, 4, raw = TRUE) %*% c(.7, -.8, .5, 1))}
treatment <- function(X) {
  as.vector(poly(X, 4, raw = TRUE) %*% c(0, -1.5, .5, .8)) + .15}

I <- declare_inquiry(LATE = treatment(cutoff) - control(cutoff))
```

7.4.3 Multiple inquiries

In some designs, we are interested in the value of an inquiry for many units or for many types of units.

We can enumerate them one-by-one, to describe the average treatment effect, two conditional average treatment effects, and the difference between them.

```
I <- declare_inquiry(
  ATE = CATE_X1 = mean(Y_Z_1[X == 1] - Y_Z_0[X == 1]),
  CATE_X0 = mean(Y_Z_1[X == 0] - Y_Z_0[X == 0]),
  CATE_X1 = mean(Y_Z_1[X == 1] - Y_Z_0[X == 1]),
  Difference_in_CATEs = CATE_X1 - CATE_X0)
```

In the multilevel regression and poststratification (MRP) design in Section 14.3, we want to know what the average of a survey question is in each state.

We declare an inquiry at the county level below. We rely on `group_by` and `summarize` from `dplyr` to write a function `MRP_inquiry` that uses a pipeline to group the data into counties and take the average. Now, our design targets an inquiry for each state.

```
M <-
declare_model(
  counties = add_level(N = 5, county_quality_mean = rnorm(N)),
  schools = add_level(N = 5, school_quality = rnorm(N, mean = county_quality_mean))
)

MRP_inquiry <-
function(data) {
  data %>%
    group_by(counties) %>%
    summarize(mean_school_quality = mean(school_quality),
             .groups = "drop")
}

I <- declare_inquiry(handler = MRP_inquiry)
```

We discuss further in 9.4 how to link inquiries to answer strategies, including the case of multiple inquiries.

Further reading

- Goertz and Mahoney (2012) on differences across inquiries in qualitative and quantitative research.
- Dawid (2000) on cause-of-effects questions.
- Yamamoto (2012) on causal attribution.
- Zhang and Rubin (2003) on “truncation-by-death”

Chapter 8

Crafting a data strategy

In order to collect information about the world, researchers must deploy a data strategy. Depending on the design, the data strategy could include decisions about any or all of the following: sampling, assignment, and measurement. Sampling is the procedure for selecting which units will be measured; assignment is the procedure for allocating treatments to sampled units; and measurement is the procedure for turning information about the sampled units into data. These three procedures parallel the three elements of an inquiry: the units, treatment conditions, and outcomes.

We think about data strategies in response to Principle 3.5: Confront the challenges of descriptive, causal, and generalization inference.

Sampling choices are used to justify generalization inference: we want to make general claims which often implies inferences about units *not* sampled. For this reason, we need to pay special attention to the procedure by which units are selected into the sample. We might use a random sampling procedure in order to generate a design-based justification for generalizing from samples to populations. Nonrandom sampling procedures are also possible: convenience sampling, respondent-driven sampling, and snowball sampling are examples of data strategies that do not include an explicitly randomized component.

Assignment choices are used to justify causal inferences: we want to make inferences about the conditions to which units were *not* assigned. For this reason, experimental design is focused on the assignment of treatments. Should the treatment be randomized? How many treatment conditions should there be? Should we use a simple coin flip to decide who receives treatment, or should we use a more complicated strategy like blocking?

Measurement choices are used to justify descriptive inferences: we want to make inferences about latent values *not* observed on the basis of measured values. The tools we use to measure are a critical part of the data strategy. For

many social scientific studies, a main way we collect information is through surveys. A huge methodological literature on survey administration has been developed to help guide questionnaire development. Bad survey questions yield distorted or noisy responses due to large measurement error. A biased question systematically misses the true latent target it is designed to measure, in which case the question has low validity. A question is high variance if (hypothetically) you would obtain different answers each time you asked, in which case the question has low reliability. The concerns about validity and reliability do not disappear once we move out of the survey environment. For example, the information that shows up in an administrative database is itself the result of many human decisions, each of which has the possibility of increasing or decreasing the distance between the measurement and the latent measurement target.

Strong research design can help address these three inferential challenges, but we can never be sure that our sample generalizes, or that we know what would have happened in a counterfactual state of the world, or what the true latent value of the outcome is (or if it even exists). Researchers have to choose good sampling, assignment, and measurement techniques that, when combined and applied to the world, will produce analysis-ready information.

More formally, the data strategy D is a set of procedures that result in a dataset d . It is important to keep these two concepts straight. If you apply data strategy D to the world m , it produces a dataset d . We say d is “the” result of D , since when we apply the data strategy to the world, we only do so once and we obtain the data we obtain. But when we are crafting a data strategy, we have to think about the many datasets that the data strategy *could have* produced under all the models in M , since we don’t know which one m is. Some of the datasets might be really excellent. For example, in good datasets, we achieve good covariate balance across the treatment and control groups. Or we might draw a sample whose distribution of observable characteristics looks really similar to the population. But some of the datasets might be worse: because of the vagaries of randomization, the particular realizations of the random assignment or random sampling might be more or less balanced. We do not have to settle for data strategies that might produce weak datasets – we are in control of the procedures we choose. We want to choose a data strategy D that is likely to result in a high-quality dataset d .

In Figure 8.1, we illustrate the data strategy and its three elements: sampling, treatment assignment, and measurement. The three elements of data strategies are highlighted by blue boxes to emphasize that they are in the control of the researcher. No arrows go into these nodes; they are set by the researcher. In each case, the strategy selected by the researcher affects an endogenous variable related to sampling, treatment assignment, and measurement. The sampling procedure causes changes in R , a variable which represents whether participants provide outcome data, for example responding to survey questions. R is not in control of the researchers, which is why it is not highlighted in blue. It

is affected by S , the sampling procedure, but also by the idiosyncratic choices of participants who have higher and lower interest and ability to respond and participate in the study. These idiosyncratic features and their causal effect on whether participants respond is reflected in the arrow between U and R . Similarly, the endogenous variable D represents whether participants receive the treatment. D is affected by the treatment assignment procedure in the data strategy (Z), which is controlled by the researcher, but also potentially by unobserved idiosyncratic features of individuals U . Some data strategies, e.g., random assignment in this case, will block arrows between U and D . However, even random assignment may not fully block this path, because noncompliance may lead to divergences between assigned treatments Z and received treatments D . The final researcher node is Q , the measurement procedure. Q affects Y , the observed outcome, measured by the researcher. Y is also affected by a latent variable Y^* , which cannot be directly observed. The measurement procedure provides an imperfect measurement of that latent variable, which is (potentially) affected by treatment D and unobserved heterogeneity U . In the robustness section at the end of the chapter, we explore further variations in this DAG that incorporate threats to inference from noncompliance, attrition, excludability violations, and interference.

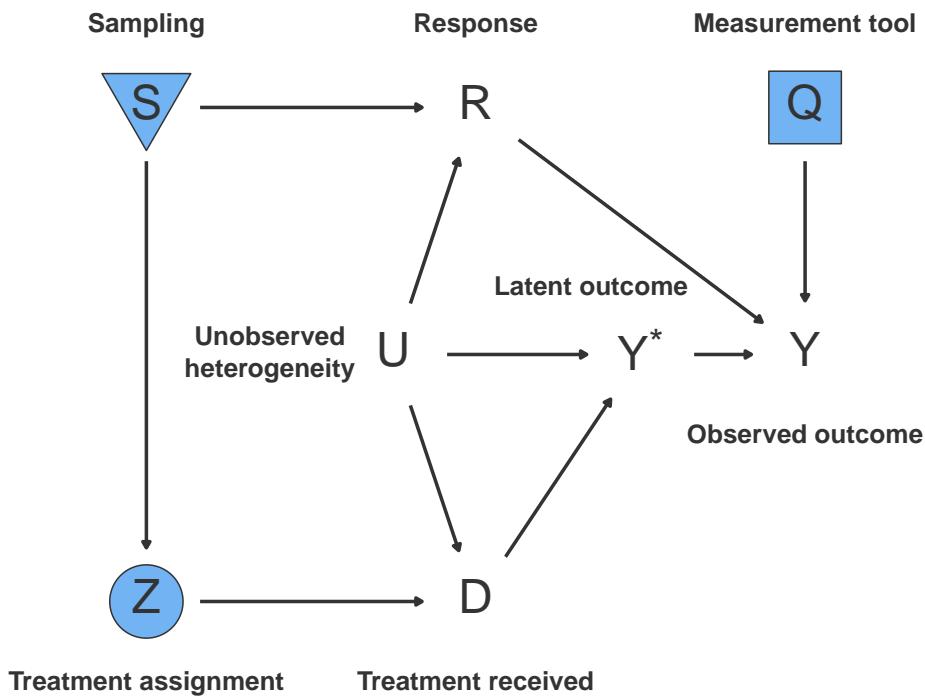


Figure 8.1: DAG illustrating three elements of a data strategy: sampling, assignment, and measurement.

8.1 Elements of data strategies

Every inquiry is defined by a set of outcomes measured within a set of one or more treatment conditions for a set of units, as well as a function that summarizes those outcomes. The three elements of the data strategy parallel the first three elements of inquiries: we sample units, assign treatment conditions, and measure outcomes.

8.1.1 Sampling

Sampling is the process by which units are selected from the population to be studied. The starting point for every sampling strategy should be to consider the units defined in the inquiry. In some cases, all the units in the population are included in the study, but in others, we consider only a subset.

Why would we ever be content to study a sample and not the full population? For infinite populations, we have no choice. For finite populations the first and best explanation is cost: it's expensive and time-consuming to conduct a full census of the population. Even well-funded research projects face this problem, since money and effort spent answering one question could also be spent answering a second question. A second reason to sample is the diminishing marginal returns of additional data collection. Increasing the number of sampled units from 1,000 to 2,000 will greatly increase the precision of our estimates. Moving from 100,000 to 101,000 will improve things too, but the scale of the improvement is much smaller. Finally, it may simply not be possible to sample some units. Units in the distant past or distant future, for example, are not available to be sampled, even if they are in the set of units that define the inquiry.

Some sampling procedures involve randomization while others do not. Whether a sampling procedure is randomized or not has large implications for the answer strategy. Randomized designs support “design-based inference,” which refers to the idea that we rely on known features of the sampling process when producing population-level estimates – much more about this in the next chapter on answer strategies. When randomization breaks down (e.g., if the design encounters attrition) or if nonrandomized designs are used, then we have to fall back on model-based inference to generalize from the sample to the population. Model-based inference relies on researcher beliefs about the nature of the uncontrolled sampling process in order to make inferences about the population. When possible, design-based inference has the advantage of letting us ground inferences in known rather than assumed features of the data generation process. That said, when randomly sampled individuals fail to respond or when we seek to make inferences about *new* populations, we oftentimes fall back to model-based inference.

8.1.1.1 Randomized sampling designs

Owing to the natural appeal of design-based inference, we start off with randomized designs before proceeding to nonrandomized designs. Randomized sampling designs typically begin with a list of all units in a population, then choose a subset to sample using a random process. These random processes can be simple (every unit has an equal probability of inclusion) or complex (first we select regions at random, then villages at random within selected regions, then households within selected villages, then individuals within selected households).

Table 8.1 collects all of these kinds of random sampling together and offers an example of functions in the `randomizr` package you can use to conduct these kinds of sampling. The most basic form is simple random sampling. Under simple random sampling, all units in the population have the same probability p of being included in the sample. It is sometimes called coin flip random sampling because it is as though for each unit, we flip a weighted coin that has probability p of landing heads-up. While quite straightforward, a drawback of simple random sampling is that we can't be sure of the number of sampled units in advance. On average, we'll sample $N \cdot p$ units, sometimes slightly more units will be sampled and sometimes fewer.

Table 8.1: Kinds of random sampling

Design	Description	Randomizr function
Simple random sampling	“Coin flip” or Bernoulli random sampling. All units have the same inclusion probability p	<code>simple_rs(N = 100, p = 0.25)</code>
Complete random sampling	Exactly n of N units are sampled, and all units have the same inclusion probability n/N	<code>simple_rs(N = 100, p = 0.25)</code>
Stratified random sampling	Complete random sampling within pre-defined strata. Units within the same strata have the same inclusion probability n_s / N_s	<code>strata_rs(strata = regions)</code>
Cluster random sampling	Whole groups of units are brought into the sample together.	<code>cluster_ra(clusters = households)</code>
Stratified cluster sampling	Cluster random sampling within strata	<code>strata_and_cluster_rs(strata = regions, clusters</code>
Multi-stage random sampling	First clusters, then units within clusters	<code>cluster_ra(clusters = villages)</code> <code>strata_ra(strata = villages)</code>

Complete random sampling addresses this problem. Under complete random sampling, exactly n of N units are sampled. Each unit still has an inclusion probability of $p = n/N$, but in contrast to simple random sampling, we are guaranteed that the final sample will be of size n .¹ Complete random sampling represents an improvement over simple random sampling because it rules out samples in which more or fewer than $N p$ units are sampled. One circumstance in which we might nevertheless go with simple random sampling is when the size of the population is not known in advance, sampling choices may have to be made “on the fly.”

Complete random sampling solves the problem of fixing the total number of sampled units, but it doesn’t address the problem that the total number of units with particular characteristics will not be fixed. Imagine a population with N_y young people and N_o old people. If we sample exactly n from the population $N_y + N_o$, the number of sampled young people (n_y) and sampled old people (n_o) will bounce around from sample to sample. We can solve this problem by conducting complete random sampling *within* each group of units. This procedure goes by the name stratified random sampling, since the sampling is conducted separately within the strata of units.² In our example, our strata were formed by a dichotomous grouping of people into “young” and “old” categories, but in general, the sampling strata can be formed by any information we have about units before they are sampled. Stratification offers at least three major benefits. First, we defend against sampling surprisingly too few units in some stratum by “bad luck.” Second stratification tends to produce lower variance estimates of most inquiries. Finally, stratification allows researchers to “oversample” subgroups of particular interest.

Stratified sampling should not be confused with cluster sampling. Stratified sampling means that a fixed number of units from a particular group are drawn into the sample. Cluster sampling means that units from a particular group are brought into the sample *together*. For example, if we cluster sample households, we interview all individuals living in a sampled household. Clustering introduces dependence in the sampling procedure – if one member of the household is sampled, the other members are also always sampled. Relative to a complete random sample of the same size, cluster samples tend to produce higher variance estimates. Just as the individual sampling designs, cluster sampling comes in simple, complete, and stratified varieties with parallel logics and motivations.

Lastly, we turn to multi-stage random sampling, in which we conduct random sampling at multiple levels of a hierarchically-structured population. For example, we might first sample regions, then villages within regions, then households within villages, then individuals within households. Each of those sampling

¹To convince yourself of the difference between simple and complete random sampling, run `table(simple_rs(N = 100, prob = 0.5))` a few times and compare the results with `table(complete_rs(N = 100, n = 50))`

²To convince yourself of the difference between complete and stratified sampling, run `age <- rep(c("Y", "O"), 50); table(age, complete_rs(N = 100, n = 50))` a few times and compare the results with `table(age, strata_rs(strata = age))`

steps might be stratified or clustered depending on the researcher's goals. The purpose of a multi-stage approach is typically to balance the logistical difficulties of visiting many geographic areas with the relative ease of collecting additional data once you have arrived.

Figure 8.2 gives a graphical interpretation of each of these kinds of random sampling. Here, we imagine a population of 64 units with two levels of hierarchy. For concreteness, we can imagine that the units are individuals nested within 16 households of four people each and the 16 households are nested within four villages of four people each. Starting at the top left, we have simple random sampling at the individual level. The inclusion probability was set to 0.5, so on average, we ought to sample 32 people, but in this particular draw, we actually sampled only 29. Complete random sampling (top center), fixes this problem, so exactly 32 people are sampled – but these 32 are unevenly spread across the four villages. This is addressed with stratified sampling. In the top right, we sample exactly 8 people at random from each village of 16 total people.

Moving down to the middle row of the figure, we have three approaches to clustered random sampling. Under simple random sampling at the cluster level, each cluster has the same probability p of inclusion in the sample, so on average we will sample eight clusters. This time, we only sampled seven. This problem can again be fixed with complete random sampling (center facet), but again we have an uneven distribution across villages. Stratified cluster sampling ensures that exactly two households from each village are sampled.

The bottom row of the figure illustrates some approaches to multistage sampling. In the bottom left panel, we conduct a simple random sample of individuals in each sampled cluster. In the bottom center, we draw a complete random sample of individuals in each sampled household. And in the bottom right, we stratify on an individual level characteristic – we always draw one individual from each row of the household. “Row” could refer to the age of the household members. This doubly-stratified multistage random sampling procedure ensures that we sample two households from each village and within those households, one older member and one younger member.

8.1.1.2 Nonrandomized sampling designs

Because nonrandomized sampling procedures are defined by what they don't do – they don't use randomization – a hugely varied set of procedures could be described this way. We'll consider just a few common ones, since the idiosyncrasies of each approach are hard to systematize.

Convenience sampling refers to the practice of gathering units from the population in an inexpensive way. Convenience sampling is a good choice when generalizing to an explicit population is not a main goal of the design, for example when a sample average treatment effect is a theoretically-important inquiry. For many decades, social science undergraduates were the most abundant data source available to academics and many important theoretical claims have been

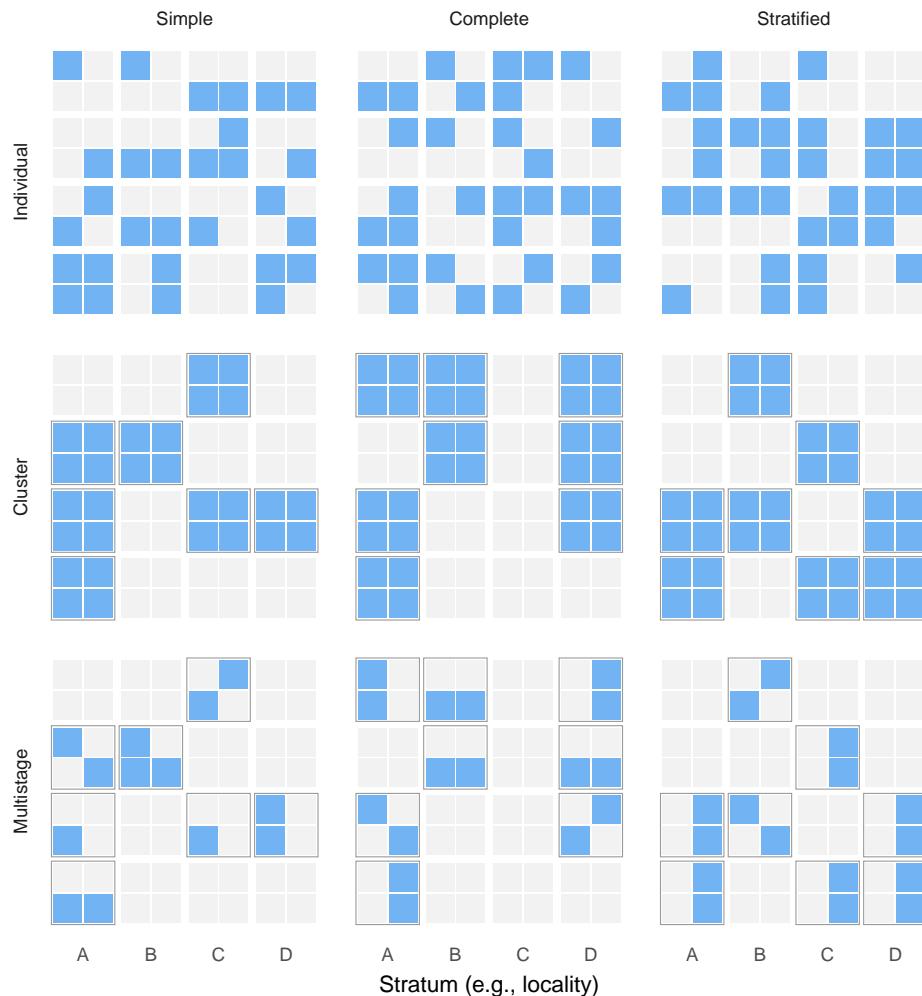


Figure 8.2: Nine kinds of random sampling

established on the basis of experiments conducted with such samples. In recent years, however, online convenience samples like Mechanical Turk, Prolific, or Lucid have mostly supplanted undergraduates as the convenience sample of choice. Convenience sampling may lead to badly biased estimates of population quantities. For example, cable news shows often conduct viewer polls that should not be taken at all seriously. While such polls might promote viewer loyalty (and so might be worth doing from the cable executives' perspective) they do not provide credible evidence about what the population at large thinks or believes.

Many types of qualitative and quantitative research involve convenience sam-

pling. Archival research often involves a convenience sample of documents on a certain topic that exist in an archive. The question of how these documents differ from those that would be in a different archive, or how the documents available in archives differ from those that do not ever make it into the archive importantly shapes what we can learn from them. With the decline of telephone survey response rates, researchers can no longer rely on random digit dialing to obtain a representative sample of people in many countries, and instead must rely on convenience samples from the internet or panels who agree to have their phone numbers in a list. Sometimes, reweighting techniques in the answer strategy can, in some cases, help recover estimates for the population as a whole if sampling if a credible model of the unknown sampling process can be agreed upon.

Next, we consider purposive sampling. Purposive is a catch-all term for rule-based sampling strategies that do not involve random draws but also are not purely based on convenience and cost. A common example is quota sampling. Sampling purely based on convenience often means we will end up with many units of one type but very few of another type. Quota sampling addresses the problem by continuing to search for subjects until target counts (quotas) of each kind of subject are found. Loosely speaking, quota sampling is to convenience sampling as stratified random sampling is to complete random sampling: it fixes the problem that not enough (or too many) subjects of particular types are sampled by employing specific quotas. Importantly, however, we have no guarantee that the sampled units *within* a type are representative of that type overall. Quota samples remain within-stratum convenience samples.

A second common form of purposive sampling is respondent-driven sampling (RDS), which is used to sample from hard-to-reach populations such as HIV-positive needle users. RDS methods often begin with a convenience sample and then systematically obtain contacts for other units who share the same characteristic in order to build a large sample.

Each of these three nonrandom sampling procedures – convenience, quota, and respondent-driven – is illustrated in Figure 8.3. Imagining that village A is easier to reach, we could obtain a convenience sample by contacting everyone we can reach in village A before moving on to village B. This process doesn't yield good coverage across villages and for that, we can turn to quota sampling. Under this quota sampling scheme, we talk to the five people who are easiest to reach in each of the four villages. Finally, if we conduct a respondent-driven sample, we select one seed unit in each village, and that person recruits their four closest friends (who may or may not reside in the same village).

8.1.1.3 Sampling designs for qualitative research

Another term for sampling is case selection. In case study research, whether qualitative or quantitative, the way we select the (typically small) set of cases is of great importance, and considerable attention has been paid to developing

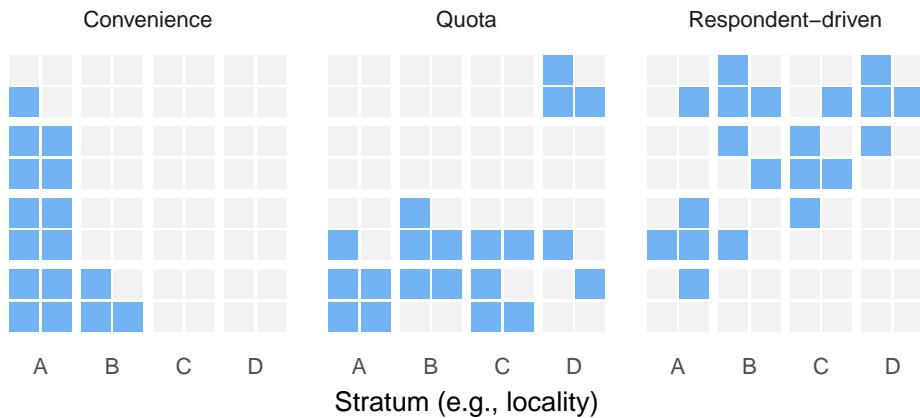


Figure 8.3: Three forms of non-random sampling.

case selection methods.

Advice for selecting cases rages widely with many seeming disagreements across scholars (see for instance the symposium in Collier et al. (2008)). We describe the major strategies used below and highlight some of the goals and assumptions motivating them. The most general advice however, is that there are likely situations and rationales justifying any of these strategies. But whether one or other strategy is right for the problem you face mostly likely depends on the three other components of your design: what your model set is, what your inquiry is, and what your answer strategy is. Conversely, it is very difficult to assess whether one approach is more appropriate than another without knowing about these other parts of a design because is hard to tell whether a case will be useful without knowing what you plan to do with it. In short, the case selection decision is one that is usefully made, and justified, by diagnosis.

Geddes (2003) warned that “the cases you choose affect the answers you get.” This warning emphasizes the importance of case selection. If we select cases in order to arrive at a particular answer, then the research design doesn’t provide good evidence in favor of the answer.

Non-purposive selection. Fearon and Laitin (2008) argue that the best approach is to select randomly. The argument for this approach depends on the purpose and details of the design. If the goal is to use case studies to check the quality of data used in large n analysis, or to explore the sets of pathways that might link a cause to an outcome (or that link a non cause to a non outcome) then random selection has the virtue of generating a representative set of cases and guards against cherry picking. It is not hard to imagine however cases in which measurement concerns are different for $Y = 1$ and $Y = 0$ cases. One might be confident in coding for a subject recorded as having contracted Covid-19, but less certain about the coding that a subject has not.

Positive selection. Goertz (2008) argues that one should select multiple cases for which a positive outcome (e.g., a revolution) is unambiguously observed. One should also seek diversity in possible causes. A similar reasoning underpins the two principles. The goal is to have many opportunities as possible to observe possibly distinct paths leading to an outcome. This approach presupposes an interest in figuring out what the causes of a positive outcome are across cases and an ability to figure out the causal factors within a case. Thus Goertz presupposes that one can assess the counterfactual values of outcomes within a case. Given these goals and capabilities, Goertz argues that cases in which $X = 0$ and $Y = 0$ are not very useful for figuring out if $X = 1$ causes $Y = 1$. You can imagine counter arguments. We might for instance believe that the effect of X on Y runs through a positive effect of X on M and a positive effect of M on Y . But if looking at an $X = 0, Y = 0$ case we find that, awkwardly, $M = 1$, the evidence casts doubt on the causal importance of X in the $X = Y = 1$ cases. Ultimately, whether this advice is correct in any given instance is a question for diagnosis insofar it depends on the model, the inquiry, and the answer strategy.

Other purposive strategies. Lieberman (2005) proposes using the predicted values from a regression model—often referred to as the “regression line”—from an initial quantitative analysis in order to select cases for in-depth analysis. Exactly how to select however depends on the inquiry and answer strategy. When the inquiry is focused on uncovering the same causal relationship sought in the quantitative analysis, Lieberman (2005) suggests selecting cases that are relatively well-predicted and that maximize variation on the causal variable. He points to Martin (1992) and Swank (2002) as examples of designs employing this strategy. However, Lieberman (2005) advocates a different case selection strategy when the goal is to expand upon the theory initially tested in the quantitative analysis. In that instance, he recommends choosing cases lying far from the regression line, which are not well-predicted and may therefore lead to insights about what alternative mechanisms were left out of the initial regression.

Seawright and Gerring (2008) use the regression line analogy to describe seven different sampling strategies tailored to suit different inquiries.³

These include “typical cases” which are representative of the cross-case relationship and can be chosen in order to explore and validate mediating mechanisms. If the researcher’s model implies union membership increases welfare spending in democracies through its effects on negotiations with the government, for example, then the researcher might look for evidence of such processes in the cases well-predicted by the theory. Diverse cases maximize variation on both X and Y , while extreme cases are located at a maximal distance from other cases on just one dimension—in our example, the researcher chooses the two cases with the highest degree of union strength. While diverse and extreme cases might lie on the regression line, deviant cases are defined by their distance from it. Such cases call for new explanations to account for outcomes. Influential cases are

³See Gerring and Cojocaru (2016) and Plümper, Troeger and Neumayer (2010) for still larger lists.

those whose exclusion would most noticeably change the imaginary regression line (i.e., those with the highest leverage in a regression).

Two more approaches, correspond to “methods of difference” and “methods of similarity” (Mill (1884)). The method of difference approach selects a set of cases that are similar in a set of pretreatment variables but nevertheless differ in Y . This gives an opportunity to search for a cause other than those held constant that could explain the variation. The method of similarity approach selects a set of cases that have similar outcomes and discounts causes that vary across these cases and focuses on potential causes that do not. As we highlight below these methods make sense for identifying possible causes within cases rather than for assessing the effect of a putative cause that has been identified in advance.

Herron and Quinn (2016) used Monte Carlo simulations to study how well these strategies perform for the specific question of providing leverage on average causal effects. The inquiry is the average treatment effect in the population, and the answer strategy involves, perhaps optimistically, perfectly observing the selected cases’ causal types. With these simplifying assumptions, they uncover a clear hierarchy and set of prescriptions: extreme and deviant case selection fare much worse than the other methods in terms of the three diagnosands considered (root mean square error, variance, and bias of the mean of the posterior distribution). By contrast, influential case selection outperforms the other strategies, followed closely by diverse and simple random sampling. As the authors acknowledge, however, this hierarchy might look very different if the inquiry aimed at a different, exploratory quantity (such as discovering the number of causal types that exist).

Other advice focuses less on the values of X and Y and more about the scope for learning within the case. Humphreys and Jacobs (2015) provide simulations where they incorporate a process tracing inferential procedure and highlight the importance of “probative value” for case selection. The point is that there is rarely a case selection strategy that fits all problems equally well—the best strategy is the one that optimizes a particular diagnosand given stipulations about the inquiry, the model, and the answer strategy. If you can justify those stipulations and the importance of the diagnosand, then defending the choice of sampling strategy is straightforward.

Finally Levy (2008) clarifies the logics behind “most likely” and “least likely” case selection strategies – what are sometimes called “crucial case” designs. The idea here is that we may have beliefs over the heterogeneity of causal effects over cases but uncertainty about the level. If we learn that a causal effect is indeed in operation in a least likely case, we update on our beliefs about it operating in other cases. This is “Sinatra inference” (Levy, 2008): “if I can make it here I’ll make it anywhere.” Conversely the most likely case is based on the idea that if I *can’t* make it here then I *can’t* make it anywhere! The logic presupposes an answer strategy that figures out within case effects and a model that yields a structured distribution over effects.

A case selection strategy that isn't one. Last we note that an approach to case selection sometimes associated with John Stuart Mill (1884) can confuse a data strategy for an answer strategy. Mill elaborated two principles of inference (“methods”). The method of difference involves examining cases that have divergent outcomes but otherwise look very similar. If one characteristic covaries with the outcome, it becomes a candidate for the cause. For example, Skocpol (1979) compares historical periods in France, Russia, the United Kingdom, and Germany that look very similar in many regards. The first two, however, had social revolutions, while the second two did not. The presence of agrarian institutions that provided a degree of political autonomy to the peasants in France and Russia and their absence in the UK and Germany then becomes a possible clue to understanding the underlying causal structure of social revolutions. By contrast, the method of agreement involves examining cases that share the same outcome but diverge on other characteristics. Any characteristics that are *common* to the cases then become candidates for causal attribution. These “methods” are inferential rules given characteristics of cases.

But these methods are dangerous guides to case selection, because they defy Geddes’ warning. We should not select on both X and Y if we are trying to learn based on the covariation of X and Y . If we *select* two cases because they differ on the outcome but on all but one (observable) characteristic and then apply the method of difference to conclude that the different factor made the difference, then we have effectively *selected* the answer. More generally, if the information used to make an inference is already available prior to data gathering, then there is noting to be gained from the data gathering.⁴ Following Principle 3.10 to diagnose whole designs will point to the errors of the strategy.

8.1.1.4 Choosing among sampling designs

The choice of sampling strategy depends on features of the model and the inquiry, and different sampling strategies can be compared in terms of power and RMSE in design diagnosis. The model defines the population of units we want to make inferences about, and the sampling frame of the sampling strategy should match that as much as possible. The model also points us to important subgroups that we may wish to stratify on, depending on the variability within those subgroups. Whether we select convenience, random, or purposive sampling depends on our budget and logistical constraints as well as the efficiency (power or RMSE) of the design. If there is little bias from convenience sampling, we will often want to select it for cost reasons. If we cannot obtain a convenience sample that has the right composition, we may choose a purposive method that ensures we do. The choice between simple and stratified sampling comes down to the inquiry and to a diagnosis of the RMSE. When the inquiry involves a comparison of subgroups, we will often select stratified sampling. In

⁴This problem does not arise if cases are selected to be similar on background features other than X when Y is unknown — in this case the there is learning about effects from later observation of Y .

either, a diagnosis of alternative designs in terms of power or RMSE will guide selection.

8.1.2 Treatment assignment

In many studies, researchers intervene in the world to **set** the level of the causal variable of interest. The procedures used to assign units to treatment are tightly analogous to the procedures explored in the previous section on sampling. Like sampling, assignment procedures fall into two classes, randomized and nonrandomized.

8.1.2.1 Two arm trials

The analogy between sampling and assignment runs deep. All of the sampling designs discussed in the previous section have directly equivalent assignment designs. Simple random sampling is analogous to Bernoulli random assignment, stratified random sampling is analogous to blocked random assignment and so on. Many of the same design tradeoffs hold as well: just like cluster sampling generates higher variance estimates than individual sampling, clustered assignment generates higher variance estimates than individual assignment. While we usually think of randomized assignment designs only, nonrandomized designs in which the researcher applies treatments also occur. For example, researchers sometimes treat a convenience sample, then search out a different convenience sample to serve as a control group. Within-subject designs in which subjects are measured, then treated, then measured again are a second example of a nonrandomized application of treatment.

The analogy between sampling and assignment runs so deep because, in a sense, assignment **is** sampling. Instead of sampling units in or out of the study, we sample from alternative possible worlds. The treatment group represents a sample from the alternative world in which all units are treated and the control group represents a sample from the alternative world in which all units are untreated.⁵ We can reencounter the fundamental problem of causal inference through this lens – if a unit is sampled from one possible world, it can't be sampled from any other possible world. Table 8.2 collects together common forms of random assignment.

⁵Strictly speaking, this claim only holds under a noninterference assumption; if the usual noninterference assumption is incorrect, we have to redefine potential outcomes in order to recover “stability.” Assignment strategies sample from possible worlds of stable potential outcomes that we imagine in M .

Table 8.2: Kinds of random assignment

Design	Description	Randomizr function
Simple random assignment	“Coin flip” or Bernoulli random assignment. All units have the same probability of assignment	<code>simple_ra(N = 100, prob = 0.25)</code>
Complete random assignment	Exactly m of N units are assigned to treatment, and all units have the same probability of assignment m/N	<code>complete_ra(N = 100, m = 40)</code>
Block random assignment	Complete random assignment within pre-defined blocks. Units within the same block have the same probability of assignment m_b / N_b	<code>block_ra(blocks = regions)</code>
Cluster random assignment	Whole groups of units are assigned to the same treatment condition.	<code>cluster_ra(clusters = households)</code>
Block-and-cluster assignment	Cluster random assignment within blocks of clusters	<code>block_and_cluster_ra(blocks = regions, clusters = villages)</code>
Saturation random assignment	First clusters are assigned to a saturation level, then units within clusters are assigned to treatment conditions according to the saturation level	<code>saturation = cluster_ra(clusters = villages, conditions = c(0, 0.25, 0.5, 0.75))</code> <code>block_ra(blocks = villages, prob_unit = saturation)</code>

Figure 8.4 visualizes nine kinds of random assignment, arranged according to whether the assignment procedure is simple, complete, or blocked and according to whether the assignment procedure is carried out at the individual, cluster, or saturation level. In the top left facet, we have simple (or Bernoulli) random assignment, in which all units have a 50% probability of treatment, but the total number of treated units can bounce around from assignment to assignment. In the top center, this problem is fixed: under complete random assignment, exactly m of N units are assigned to treatment and the Nm are assigned to control. While complete random assignment fixes the number of units treated at exactly m , the number of units that are treated within any particular group of units (defined by a pre-treatment covariate) could vary. Under block random assignment, we conduct complete random assignment within each block separately, so we directly control the number treated within each block. Moving from simple to complete random assignment tends to decrease sampling variability a bit, by ruling out highly unbalanced allocations. Moving from complete to blocked can help more, so long as the blocking variable is correlated with the outcome.

Blocking rules out assignments in which too many or too few units in a particular subgroup are treated. To build intuition for why the correlation of the blocking variable with the outcome is important, consider forming blocks at random. None of the assignments under complete random assignment would be ruled out, so the sampling distributions under the two assignment procedures would be equivalent.

The second row of Figure 8.4 shows clustered designs in which all units within a cluster receive the same treatment assignment. Clustered designs are common for household-level, school-level, or village-level designs, where it would be impractical or infeasible to conduct individual level assignment. When units within the same cluster are more alike than units in different clusters (as in most cases), clustering increases sampling variability relative to individual level assignment. Just like in individual level designs, moving from simple to complete or from complete to blocked tends to result in lower sampling variability.

The final row of Figure 8.4 shows a series of designs that are analogous to the multi-stage sampling designs shown in Figure 8.2 – but their purpose is subtly different in spirit. Multi-stage sampling designs are employed to reduce costs – first clusters are sampled but not all units within a cluster are sampled. A saturation randomization design (sometimes called a “partial population design”) uses a similar procedure to both contain and learn about spillover effects. Some clusters are chosen for treatment, but some units *within* those clusters are not treated. Units that are untreated in treated clusters can be compared with units that are untreated in untreated clusters in order to suss out intra-cluster spillover effects (Sinclair, McConnell and Green, 2012). The figure shows how the saturation design comes in simple, complete, and blocked varieties.

8.1.2.2 Multiarm and factorial trials

Thus far we have considered assignment strategies that allocate subjects to just two conditions: either treatment or control. All generalize quite nicely to multiarm trials. Trials that have three, four, or many more arms can of course be simple, complete, blocked, clustered, or feature variable saturation. Figure 8.5 shows blocked versions of a three-arm trial, a factorial trial, and a four-arm trial.

In the three-arm trial on the left, subjects can be assigned to a control condition or one of two treatments. This design enables three comparisons: a comparison of each treatment to the control condition, but also a comparison of the two treatment conditions to each other. In the four-arm trial on the right, subjects can be assigned to a control condition or one of three treatments. This design supports six comparisons: each of the treatments to control, and all three of the pairwise comparisons across treatments.

The two-by-two factorial design in the center panel shares similarities with both the three-arm and the four-arm trials. Like the three-arm, it considers two treatments T1 and T2, but it also includes a fourth condition in which both



Figure 8.4: Nine kinds of random assignment. In the first row individuals are the sampling units, in the second row clusters are sampled, in the third clusters are sampled and then individuals within these clusters are sampled. In the first column units are sampled independently, in the second units are sampled to hit targets, in the third units are sampled to hit targets within strata.

treatments are applied. Factorial designs can be analyzed like a four-arm trial, but the structure of the design also enables further analyses. In particular, the factorial structure allows researchers to investigate whether the effects of one treatment depend on the level of the other treatment.

Three-arm	Factorial	Four-arm
C T2 C T1 T1 T2 C C	C T2 T2 T1 C T2 T2 T1	C T2 T2 T1 T2 T2 T1 T2
T1 T2 T1 T2 T1 T2 T1 T2	T1 T2 T1 C T1 T2 C T1	T3 T1 C T3 C T1 C T1
T2 T1 T1 T1 T1 T2 T2 T1	T1 C C T1 T2 T1 T2 T1 T2	T1 C C T1 T3 T1 T1 T2
C T1 T2 C C T1 C T1	T1 T2 T1 T2 C T2 T1 T2	T3 T2 T3 T2 C T2 T3 C
T1 C T2 C C T1 C T1	T1 T2 C T1 T2 T2 C C T1	T1 T3 C T3 T2 C C T1
C T2 C T1 T2 C C T2	T2 C T2 T1 T1 T2 T1 T2	T2 C T2 T1 T3 T1 T3 T2
T2 T1 T1 T2 T2 C T2 T2	T1 T2 T1 C T1 T2 T2 C T1	T3 T1 C T1 T3 T2 C T1
C T1 T1 C T1 C T1 C	T2 C T2 T1 T2 C T1 T1 T2	T2 C T2 T3 C T1 T3 T2

Figure 8.5: Multi-arm random assignment

8.1.2.3 Over-time designs

Treatment conditions can also be randomized over multiple time periods, with each unit receiving different treatment conditions in different periods. By focusing on variation in outcomes *within units* rather than across them, these designs can be more efficient than designs that compare across units. Often there is more variation across units than within the same units over time. However, there can be a tradeoff in the form of increased bias. Within-unit comparisons must rely on strong stability assumptions such as “no carry-over effects” of the treatment condition assigned in the preceding period. If which condition the unit is assigned to affects outcomes in later periods, we cannot isolate the effect of treatment just by considering the treatment it was assigned this period, we need to know the entire treatment *history*.

A stepped-wedge random assignment procedure involves assigning a subset of units to treatment in the first period, a subset of those who were not treated in the first in the second period, and so on. In the final period, all units are treated. In this design, once you are treated in a period you are treated in all subsequent periods. For example, once you receive information in a treatment about how to vote, you already have that information in later periods. In Figure 8.6, we illustrate a three-period step-wedge design, in which one third of units are assigned in the first period, a second third are treated in the second period, and the remainder in the third and final period. In such a design, we can make two comparisons: the treatment versus control contrast in each period, and the within-units over-time contrast before and after treatment. By combining these two comparisons, we have a more efficient estimate of the average treatment effect than if we had randomly assigned one half of units to treatment and the other half to control in a single period. However, we must invoke a no carry-over assumption that in the second and third period potential outcomes are only a function of the current treatment status not whether (or not) the unit was treated earlier.

Crossover designs are a second common over-time random assignment proce-

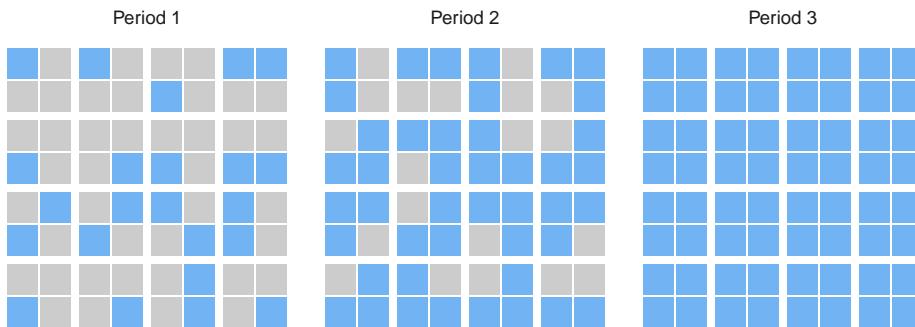


Figure 8.6: Step-wedge random assignment.

dure, in which units are first assigned one condition and then, in a second period, the opposite condition. Such a design is appropriate when units, once treated, do not retain their treatment over time. Crossover designs must also rely on an assumption of no carry-over. If this assumption is valid, the design is highly efficient: instead of having half treated and half control in a single period, all units receive treatment in one period and control in the other so we can make comparisons within each period across units with different conditions *and* for all units over time before and after treatment. . Whether the crucial no carry-over assumption holds is fundamentally not testable: it is an excludability assumption about the unobservable potential outcomes. The assumption may be bolstered by “washout period” between measurement waves, like buffer rows between crops in agricultural experiments.

8.1.2.4 Data-adaptive assignment strategies

We usually think of data strategies as static: a survey asks a fixed set of questions, a randomization protocol has a fixed probability of assignment, sampling designs are designed to yield a fixed number of subjects. But they can also be dynamic. For example, the GRE standardized test many graduate students take is data-adaptive: if you answer the easy questions right, they skip you to harder ones. This process uses fewer questions to figure out test-takers’ scores, saving everyone the laborious effort of taking and grading long examinations (see 8.1.3.4 for more on data-adaptive measurement).

Data-adaptive designs are also used when the space of possible treatments to choose from is large. We could conduct a static multi-arm trial to evaluate all of them, but experiments with too many conditions tend to have low precision because the sample is spread too thinly across conditions. The usual response to this cost problem is to turn to theory to consider which treatments are most likely to work and test those options only.

“Response-adaptive” designs are an alternative that may be appropriate in these settings. The subject pool is split into sequential “batches” subjects. The first

batch does the experiment, then the second, and so on. The probabilities of assignment to each condition (or arm) starts out equal, but we tweak them between batches. We assign a higher fraction of the second batch to conditions that performed well in the first batch. This process continues until the sample pool is exhausted. Many algorithms for deciding how to update between batches are available, but the most common (Thompson sampling) estimates the probability that each arm is the best arm, then randomly allocates subjects to arms using these probabilities. See Offer-Westort, Coppock and Green (2021) for a recent introduction to this algorithm and elaborations.

Figure @*(fig:adaptive)* shows one draw of an adaptive experimental design. We assign the 100 subjects in batch 1 to 10 conditions with equal probability. Quickly, the best arm is identified (with a true average binary outcome of 0.6) and more subjects are allocated to it. The figure illustrates how even with a total sample of just 1,000, we can obtain a very good estimate of the average outcome in the best-performing treatment arm, even without know *ex ante* which of the 10 arms was best.

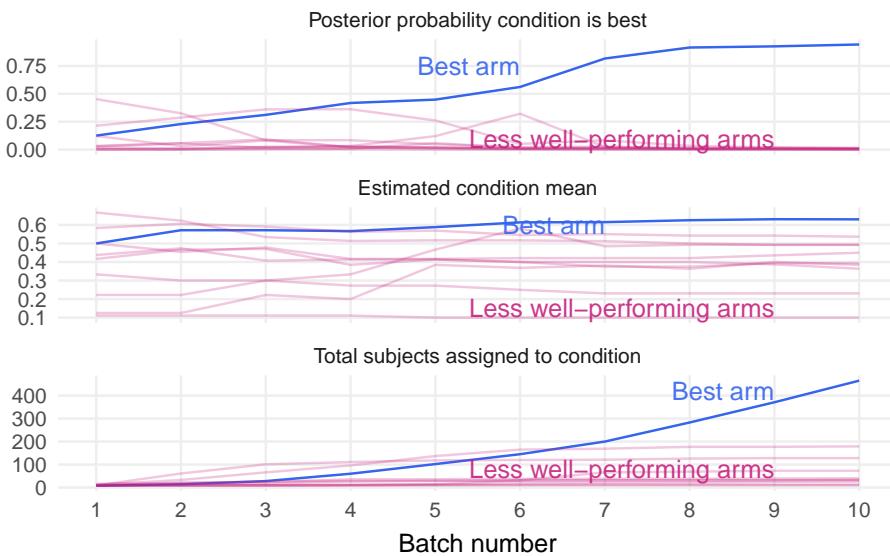


Figure 8.7: One possible path taken by the adaptive treatment assignment algorithm. From 10 arms, the best arm was quickly identified and more subjects were assigned to it.

Evaluating data-adaptive designs is complex. A core consideration when diagnosing data-adaptive designs is imagining all the ways the algorithm could have turned, as in Principle 3.10: Diagnose holistically and in Principle 3.6: Declare data and answer strategies as functions.

8.1.2.5 Non-randomized assignment

Strong causal inferences can be drawn from treatment allocation strategies that do not involve random assignment. We outline four such strategies below, with their costs and benefits.

A commonly considered strategy is alternating assignment, in which every other participant who arrives is assigned to treatment. The procedure would be identical to block random assignment — blocked on time of treatment — if participants arrived in a randomized order. It is appealing for this similarity, but it is often impossible to demonstrate that order was randomized. In fact, participants who work at different times of day may arrive at different times, and many other correlations between individual characteristics and order may arise. But the real problem comes when there are correlations between those characteristics and the order within each couple of participants. For example, if treatment status is correlated with who goes through the door first, there could be a very strong correlation between individual characteristics and treatment condition. A simple fix for this would be to block units into pairs, two by two, and randomize within each pair, rather than alternating. That procedure would be block randomization but have similar logistical advantages to the alternating design.

When participants can be assigned a score that represents need, desire, or eligibility for a treatment, with higher score representing higher likelihood of treatment, a common design is to set a cutoff score above which all units are treated and below which none are. With such a cutoff, units very near the cut-off may be very similar to each other, so a regression discontinuity design can be used to estimate the treatment effect by predicting the outcome under control (just below the cutoff) and the outcome under treatment (just above the cutoff). In such a design, the assignment of treatment is deterministic and has no random component.

A range of strategies aim to improve upon random assignment by identifying assignments that are optimal in some sense. Bayesian optimal assignment strategies identify individually-optimal assignments from a set of multiple treatments, based on past data from experiments and individual characteristics that predict treatment effectiveness. Diagnosing the properties of these so-called optimal designs is crucial, because though a treatment assignment may be optimal in terms of the likelihood that each individual receives the treatment most effective for them, the design may be inefficient due to highly variable assignment propensities and even some units with zero probability of receiving one of the treatments. Such choices may be appropriate, but can in a diagnosis researchers can directly tradeoff design criteria like efficiency with the average expected effectiveness of the treatment assigned to units.

8.1.3 Measurement

Measurement is the part of the data strategy in which variables are collected about the population of units to enable sampling, variables are collected about the sample before treatment assignment including those used in treatment assignment, and outcomes are collected after treatment assignment. All variables used in the answer strategy are collected in measurement, aside from the treatment assignment variable and assignment and sample inclusion probabilities.

Challenge of description inference arise when we want make claims about the values of variables that we do not measure. In some cases we are interested in “latent variables”, that cannot be directly measured, such as fear, support for a political candidate, or economic well-being. Instead, we use a measurement technology to imperfectly observe them, which we represent as the function Q that yields the observed outcome Y^{obs} : $Q(Y) = Y^{obs}$. Our measurement strategy is a set of functions Q for each variable we measure.

We can evaluate Each function Q : bias, or the difference between the observed and latent outcome, $Y^{obs} - Y$, which is given the special label *measurement validity*; and *measurement reliability*, which is the variance across multiple outcomes for a given individual, $V(Y_1^{obs}, Y_2^{obs}, Y_3^{obs})$. In addition, we may be concerned about the cost of each measurement, either in terms of money or time. In survey research, the costs of adding an additional survey question often come in money to pay enumerators, the opportunity cost of time for participants, and also the validity of responses if participants suffice and answer items randomly during a survey that is too long.

Selecting among measurement modes, data collectors, time periods, frequency, and the number of measurements reduces to tradeoffs between their validity and reliability. Learning which measurement tools are valid and reliable is ultimately guesswork, though it can be informed guesswork. We cannot measure the true Y_i , so we cannot truly “validate” any measurement technique (Principle 3.5). Often studies present themselves as validation studies by comparing a proposed measure to a “ground truth,” measured from administrative data or a second technique to reduce measurement error. However, neither measurement is known to be exactly Y_i , so ultimately these studies are comparisons of multiple techniques each with their own advantages and disadvantages. This does not make these studies useless, but rather points out that they should be used in service of argument in favor of some concept-measure pairs over others.

8.1.3.1 Selecting a single measure

Researchers select several characteristics of Q : who collects the measures, the mode of measurement, how often and when measures are taken, how many different observed measures of Y are collected, how they are summarized into a single measure. These design characteristics may affect validity, reliability, cost, or all three.

Data may be collected by researchers themselves, by participants, or by third parties. In some forms of qualitative research such as participant-observation and interview-based research, the researcher may be the primary data collector. In survey research, the interviewer is typically a hired agent of the researcher, and in many cases, multiple interviewers are hired. These interviewers may ask questions differently, leading to less reliable answers and in some cases validity problems when they ask questions in a way that leads to biased measures of Y . Participants are often asked to collect data on themselves, either through self-administered surveys, journaling, or taking measurements of themselves using thermometers or scales. A primary concern with self-reports is validity: do respondents report their measurements truthfully. A parallel concern is raised when participants do not collect their own data but are made aware of the fact that they are being measured by others. Finally, data may be collected by agents of government or other organizations, yielding so-called administrative data. The difference between administrative data and other forms of data is only in the identity of the data collector.

Most of the variety in measurement strategies is how those data collectors obtain their data. Humans can code data by observation through the five senses of sight, hearing, touch, smell, and taste, and by asking other humans for self-reports about themselves in surveys. Measurement instruments can also be used to record waves of light (e.g., photos), sound (e.g., audio and seismic recordings), electromagnetism (e.g., EKGs and x-rays), and combinations of more than one (e.g., video); characteristics of the atmosphere (e.g., temperature and pressure), the water (e.g., salinity and pollution), and the soil (e.g., mercury pollution); and human and animal health (e.g., blood tests). Considerable recent progress has been made in taking advantage of all of these measurement modes due to increasing computing power and machine learning techniques that can code streams of raw data from photos, videos, and these other sources and translate them into usable data. The translation of raw data into coded data that can be used for analysis is part of Q in the measurement strategy.

When data are collected can also affect validity and reliability. The inquiry should guide when data is collected in relation to other events such as an election or the holiday period or the time after a treatment is delivered to research participants. The inquiry defines whether the effect of interest is a month after treatment or in the case of long-term effects a year or more.

8.1.3.2 Multiple measures

We measure Y_i imperfectly with any single measure. In many cases, we have access to multiple imperfect measures of the same Y_i . When possible, collecting all of these different measures and averaging them to construct a single index measure will yield efficiency improvements. The average measure can borrow the different strengths of the different measures. When the tools produce answers that are highly correlated, taking multiple measures is unlikely to be worth the cost because the same information is simply duplicated, but when

the correlation is low, it will be worth taking multiple measurements and averaging to improve efficiency. Pilot studies may be usefully tasked with measuring the correlation between items. Index measures are distinct from Y_i outcomes that have multiple dimensions and must be measured with multiple items, one per dimension. In these cases, we have a single measure of Y_i just constructed in a more complex way.

8.1.3.3 Over-time measurement

Data need not be collected at a single time period. The model encodes beliefs about the autocorrelation (correlation over time) of outcomes, and this can help guide whether to collect multiple measurements or just one. If data are expected to be highly variable (low autocorrelation), then taking multiple measurements and averaging them may provide efficiency gains.

When outcomes exhibit high autocorrelation, there will be large precision gains from collecting a baseline measure before a treatment in an experiment. When outcomes exhibit lower autocorrelation, baseline measurements may not be worth the cost.

8.1.3.4 Data-adaptive measurement

Just as we can use data-adaptive methods to hone in on the most effective treatments (Section 8.1.2.4), we can use adaptive measurement techniques to hone in on the most useful measures. Adaptive inventory techniques enable deploying long batteries of survey items, for example, but enumerating the shortest set of items to any given respondent that results in a definitive measurement of Y_i . In the same way as many modern standardized tests condition the choice of survey items on students past answers in order to hone in quickly on the correct test score, adaptive inventories ask questions that will be maximally informative. The logic is the same as that of using multiple different measures for the same construct: the lower the correlation, or in other words the more new information, between two items the more informative they are. Adaptive inventories select a set of items to enumerate that provide the most uncorrelated information. See Montgomery and Rossiter (2020) for an up-to-date treatment of the adaptive measurement possibilities for constructs measured by long survey batteries.

8.2 Seek M:I:D:A Parallelism

We will discuss Principle 3.7 in much greater detail in Section 9.3.2, but we anticipate a few points here.

In the data strategy, we sample, assign units to treatment conditions, and measure outcomes to target as closely as we can the three analogous elements of the model. In the answer strategy, we take that data and plug in the observed

data in the inquiry's summary function in place of the unobserved data, which is the idea of the "plug-in principle." When the data strategy introduces distortions in the sampling, treatment assignment, or measurement from the units, conditions, and outcomes of the model, we need to adjust the answer strategy to compensate, which is the idea behind "analyze as you randomize."

The data strategy's contribution to parallelism is fidelity to the units, outcomes, and treatment conditions that define the inquiry. The units in the realized data should be representative of units defined in the inquiry. Representativeness might be rooted in random sampling in the data strategy or an assumption of ignorability in the answer strategy. The measured outcomes used in the answer strategy should be valid, reliable measures of the latent outcomes defined in the inquiry. The units in each treatment condition should be representative of the corresponding set of potential outcomes in the model. Again, claims about representativeness might be grounded in the data strategy (e.g., random assignment) or in the model (e.g., selection on observables).

8.3 Robustness

Principle 3.3 encourages us to "entertain many models," considering plausible variations of the set of variables, their probability distributions, and the relationships between them. The payoff of doing so comes in selecting the data and answer strategies, in particular choosing D and A such that they are good designs under a wide array of plausible models.

In this section, we begin the discussion of how to select empirical strategies that are robust to multiple models by focusing on the data strategy. We identify four core threats to data strategies: noncompliance (failure to treat), attrition (failure to be included in the sample or provide measures), and excludability violations (causal effects of random sampling, random assignment, or measurement on the latent outcome). If serious, these threats may necessitate a changes to the inquiry, the answer strategy, or the data strategy itself.

Below, we adapt Figure 8.1 presented in the chapter's introduction to introduce each of these threats and discuss each threat in turn.

8.3.1 Noncompliance

The first type of threat during implementation is noncompliance, which occurs when the assignment variable Z imperfectly manipulates the treatment variable D . When noncompliance is not a problem, $D_i = Z_i$, but in design that encounter noncompliance $D_i \neq Z_i$. One-sided noncompliance occurs when some treated units fail to be treated (and receive the control condition instead). Two-sided noncompliance occurs when some units assigned to treatment do not take treatment and some units assigned to control do take treatment. Noncompliance hampers experimental studies but also affects observational designs for

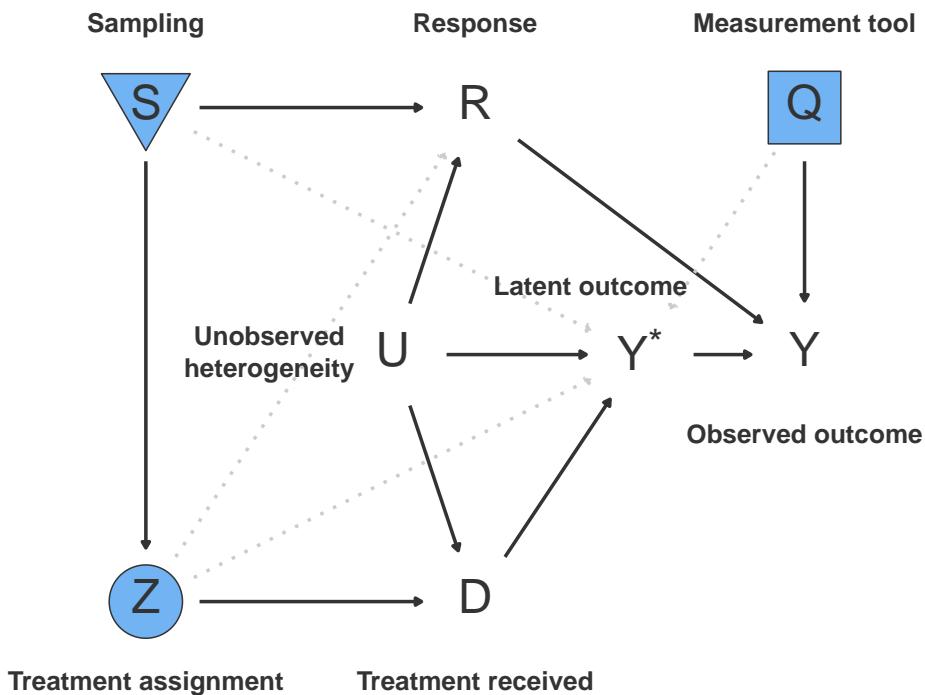


Figure 8.8: DAG with exclusion restrictions.

causal inference in which nature or a non-random administrative process affects treatment such as a threshold cut-off, but only imperfectly.

In the presence of noncompliance, a change in inquiry is sometimes unavoidable. The average difference between those assigned to treatment and those assigned to control no longer targets the average treatment effect, but instead only the effect of *assignment* to treatment. We instead call this inquiry the intent-to-treat effect, and we can estimate it well by comparing the groups as assigned.

Answer strategies that compare those who *received* treatment to those that did not are prone to bias because unobserved heterogeneity now jointly affects D with Z . The randomized experiment is broken for the ATE because the treatment group no longer randomly samples from the untreated potential outcomes.

Instead, we might have to which a complier average treatment may be obtained using instrumental variables estimation, which implies switching to a “local” inquiry among a subset of units that comply with treatment (take it when offered). This effect may differ from the average treatment effect if the kinds of participants who comply with treatments differ systematically from other types of participants. Estimating the complier average treatment effect requires the addition of assumptions on top of those for randomized experiments, including

the ignorability of treatment assignment and, in the case of two-sided noncompliance, a monotonicity assumption that rules out defiers.

In the case of randomized experiments, spending budget and time to carefully design the treatment delivery protocols to avoid noncompliance will help avoid or minimize the threat from noncompliance. A parallel set of decisions faces the designer of an observational study with noncompliance in treatments. Instrumental variables designs imply there is noncompliance and the inquiry is the complier average treatment effect (in some cases, the intent-to-treat effect is also of interest). Researchers who adopt regression discontinuity designs also focus on a local effect among units near the threshold, and in the case of the fuzzy regression discontinuity design with noncompliance must switch to a complier local average treatment effect.

See Section 17.6 for a discussion of noncompliance in experiments and Section 15.4 for related discussion of “noncompliance” in observational studies.

8.3.2 Attrition

Attrition occurs when we do not have outcome measures for all sampled units. Two types of missing data may result: when a single measure is missing, commonly known as item nonresponse; and when all measures are missing for a participant, known as survey nonresponse. Though these terms were coined by survey researchers, the problems are just them same nonsurvey measurement strategies, like missing administrative data, for example.

Whether attrition is a problem depends on whether response (R) is causally affected by variables other than random sampling. If it is not, we say the missingness completely at random, just as if we had simply added one more random sampling step to the design. Outside of explicit sampling designs, missingness completely at random is rare, though possible, perhaps due to idiosyncratic administrative procedures or computer error. If attrition is completely at random, there is no effect of any variable on R , and there is a loss of sample size but no added threat of bias.

If missingness is affected by other variables – some units are more likely to respond because of unobserved background characteristics such as being at home when the survey taker calls – then inferences may be biased. Attrition is doubly difficult in experiments, because if treatment affects not just how a unit responds, but whether it responds, then treatment-control comparisons on the basis of observed data may be biased.

A bounding approach like the one described in Section 9.1.4 is a design-based answer strategy to drawing inferences despite missingness. Section 14.1.1 describes a design-based data strategy for avoiding the problem in the first place. Model-based approaches involve reweighting the data, much according to the strategy described in Section 17.4.

8.3.3 Excludability

Excludability means that when we define potential outcomes, we can exclude a variable from our definition of the potential outcome. When we define the treated potential outcome for the latent outcome as $Y_i(D_i = 0)$, we invoke (at least) three important excludability assumptions: no effect of sampling S , no effect of treatment assignment Z (except through treatment D !), and no effect of measurement Q on the latent outcome Y_i . If we do not invoke these assumptions, we must define the potential outcome function as $Y_i(D_i, S_i, Z_i, Q_i)$. When we do invoke the assumptions, we can write plain $Y_i(D_i)$. The three assumptions are represented as gray dotted lines in Figure 8.8. These are strong assumptions that are often not met in practice.

The first excludability assumption is that there is no causal effect of sampling S on latent outcome Y_i . This assumption could be violated if the fact of being *included in the sample* changes your attitudes. For example, if the very act of being asked to be in a focus group makes you reflect on your political beliefs and there change them, the sampling excludability assumption may be violated.

The second excludability assumption is that there is no causal effect of assignment Z on outcome Y – except through the treatment D . This assumption is constantly under threat! In observational studies “instrumental variables” design, excludability is the assumption of no alternative channels through which the instrument affects outcomes except the treatment variable. In the language of economics, no alternative channels through which the exogenous variable affects the outcome, except through the endogenous variable. In the entertainingly titled “Rain, Rain, Go Away: 176 potential exclusion-restriction violations for studies using weather as an instrumental variable,” Mellon (2021) discusses how random variation in whether it rains has been misused to study the effects of other treatments.

Equally worrying is the excludability of measurement assumption, that Q does not affect Y . Hawthorne effects, in which the fact of being measured changes outcomes, are an example of a violation of this excludability assumption. If outcomes depend on whether subjects know they are being measured or do not, then we cannot exclude the effect of measurement from our effect estimates.

A final excludability assumption is an addendum to the second: Z must have no effect on Q . How and whether we measure outcomes should not depend on whether a unit is assigned to treatment. This excludability assumption is commonly referred to as the requirement that measurement be parallel across treatment conditions. If we measure outcomes using a face-to-face survey in the treatment group and a mail-back survey in control, then we cannot separate (exclude!) the effect of measurement from the effect of treatment.

8.3.4 Interference

We have four endogenous outcomes in the DAG of a research design above: R , whether a participant responds to data collection; D , whether a respondent receives treatment; Y , the latent outcome; and Y , the observed outcome. Setting aside attrition and noncompliance for the moment, R is a function only of sampling; D of treatment assignment; Y of D ; and Y of measurement strategy Q .

Interference occurs when these endogenous variables depend not only on whether and how *they* are sampled, assigned to treatment, and measured, but whether and how *other units* are sampled, assigned to treatment, and measured. We usually assume, for example, that $Y_i(Z_i) = Y_i(Z_i, Z_i)$. In other words, Y_i the outcome for unit i , is a function of its own treatment assignment status Z_i not those of other units (Z_i).

We often think of interference when considering how treatments spill from treated to untreated units. But interference can also be induced by sampling: potential outcomes might depend on whether other units are included in the sample. Or by measurement: Measurement interference occurs when Y_i depends on whether and how *other* units (or outcomes) are measured. For example, asking about one attitude might affect how subjects respond to a second question.

We discuss the some complications of interference in experiments in Sections 17.10 and 17.11.

8.4 Declaring data strategies in code

The three data strategy functions, `declare_sampling`, `declare_assignment`, and `declare_measurement` share most features in common. All three all add variables to the running data frame. `declare_sampling` is special in that it has a `filter` argument that determines which (if any) of the units should be dropped from the data and which should be retained as the sample. `declare_assignment`, and `declare_measurement` work in the exact same way as one another. The reason we separate them is to insist on the features of the data strategy, not for a deep programming reason.

8.4.1 Sampling

Declaring a sampling procedure involves constructing a variable indicating whether a unit is sampled or not and then filtering to sampled units. By default, you should create a variable `S` and `declare_sampling` will filter to sampled units by selecting those for which `S == 1`. You can rename your sampling variable or create more than to develop multistage sampling procedures, you just may need to alter the `filter` argument to reflect your changed procedure.

```
D <- declare_sampling(S = complete_rs(N = 100, n = 10))
```

For a multistage sample of districts then villages then households, we start out with all the data and sample at each stage then combine the three sampling indicators to form the final indicator S.

```
D <-
declare_sampling(
  # sample 20 districts
  S_districts = cluster_rs(clusters = districts, n = 20),
  # within each district, sample 50 villages
  S_villages = strata_and_cluster_rs(
    strata = districts,
    clusters = villages,
    strata_n = 10
  ),
  # within each village select 25 households
  S_households = strata_and_cluster_rs(
    strata = villages,
    clusters = households,
    strata_n = 25
  ),
  S = S_districts == 1 & S_villages == 1 & S_households == 1
  filter = S == 1
)
```

You could also perform each of these steps in separate calls, and the data will be filtered appropriately step-to-step.

```
D <-
declare_sampling(S = cluster_rs(clusters = districts, n = 20)) +
declare_sampling(S = strata_and_cluster_rs(
  strata = districts,
  clusters = villages,
  strata_n = 10
)) +
declare_sampling(S = strata_and_cluster_rs(
  strata = villages,
  clusters = households,
```

```
    strata_n = 25
)}
```

For many sampling designs, the probabilities of inclusion in the sample cause distortions in parallelism in the data strategy. To conform to Principle 3.7, we often need to adjust in the answer strategy for these distortions by reweighting the data according to the inverse of the inclusion probabilities, to reverse the distortion. For common sampling designs in `randomizr`, we provide a built-in function for calculating these. If you roll your own sampling function, you will need to calculate them yourself. Here we show how to include probabilities from a stratified sampling design.

```
M <-
  declare_model(N = 100,
                X = rbinom(N, 1, prob = 0.5))

D <-
  declare_sampling(
    S = strata_rs(strata = X, strata_prob = c(0.2, 0.5)),
    S_inclusion_probability =
      strata_rs_probabilities(strata = X,
                                strata_prob = c(0.2, 0.5))
  )
```

8.4.2 Treatment assignment

The declaration of treatment assignment procedures works similarly to sampling, but we don't drop any units. Treatment assignment probabilities often come into play just like in sampling in order to restore parallelism. You can use `randomizr` to calculate them for many common designs in a similar fashion, except that for treatment assignment in order to know with what probability you were assigned to the condition you are in we have to know what condition you are in. To obtain condition assignment probabilities we can declare:

```
D <-
  declare_assignment(
    Z = complete_ra(N, m = 50),
    Z_condition_probability =
      obtain_condition_probabilities(assignment = Z, m = 50)
  )
```

8.4.3 Measurement

Measurement procedures can be declared with `declare_measurement`. A common use is to generate an observed measurement from a latent value:

```
M <- declare_model(N = 100, latent = runif(N))
D <- declare_measurement(observed = rbinom(N, 1, prob = latent))
```

8.4.3.1 Revealing potential outcomes

The most common use of `declare_measurement` in this book, however, is for the “revelation” of potential outcomes according to treatment assignments. We build potential outcomes in `declare_model`, randomly assign in `declare_assignment`, then reveal outcomes in `declare_measurement`. We use the `reveal_outcomes` function to pick out the right potential outcome to reveal for each unit.

```
M <-
  declare_model(N = 100,
    potential_outcomes(Y ~ rbinom(
      N, size = 1, prob = 0.1 * Z + 0.5
    )))
D <-
  declare_assignment(Z = complete_ra(N, m = 50)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z))
```

8.4.3.2 Index creation

Many designs use multiple measures of the same outcome, which are then combined into an index. For example, here’s a design with three measures of `Y` that we will combine using factor analysis.

```
library(psych)

D <- declare_measurement(
  index = fa(
    r = cbind(Y_1, Y_2, Y_3),
    nfactors = 1,
    rotate = "varimax"
```

```
)$scores  
)
```

128

Crafting a data strategy

8.4

Chapter 9

Choosing an answer strategy

Your answer strategy is your plan for what you will do with the information gathered from the world in order to generate an answer to the inquiry. Qualitative and quantitative methods courses overflow with advice about which answer strategies to choose. Under what conditions should you use ordinary least squares, when should you use logit? When is a machine learning algorithm the appropriate choice and when would a comparative case study be more informative? When is *no* answer strategy worth pursuing because of the fundamental limitations of the data strategy?

Evaluation of an answer strategy depends on your ultimate goals: what is the answer to be used for? A perfect answer is generally elusive in empirical research and so in practice we often need to select among strategies that might all be appropriate, but which come with different strengths and weaknesses. For instance some might suffer less from bias while others might be more precise. In other words, which answer strategy is best depends on what diagnosands you care about.

This chapter first describes four types of answer strategies: point estimators, tests, Bayesian posteriors, and interval estimation. Second, we describe how to provide estimates of uncertainty along with answers. Uncertainty estimates can often be thought of as empirical estimates of a diagnosand like the true standard error, or the probability of obtaining an estimate larger than a specified value under a null model. Third, we revisit several design principles that guide how to choose an answer strategy. Principle 3.6 emphasizes that answers strategies as *procedures*, since they describe what analyses will be undertaken under different contingencies. Principle 3.7 says we should “seek *M:I:D:A* parallelism.” We want to choose answer strategies that strengthen the analogy between the empirical and theoretical halves of the research. How best to choose *A* in order to strengthen the analysis depends, of course, on the three elements of research design: the model, the inquiry, and the data strategy. Principle 3.10 is a reminder

to diagnose holistically: we can't choose answer strategies in isolation from the other design elements because the influence of the choice of estimator on diagnosands is different, depending on the other design elements.

9.1 Types of answer strategies

9.1.1 Point estimation

The most familiar class of answer strategies are point-estimators that produce estimates of scalar parameters. The sample mean of an outcome, the difference-in-means estimate, the coefficient on a variable in a logistic regression, and the estimated number of topics in a text corpus are all examples of point estimators.

To illustrate point estimation in general, we'll try to estimate the average age of the citizens of a small village in Italy. Our model is straightforward – the citizens of the small village all have ages – and the inquiry is the average of them. In our data strategy, we randomly sample three citizens whose ages are then measured via survey to be 5, 15, and 25. Our answer strategy is the sample mean estimator, so our estimate of the population average age is a point estimate of 15.

Is this a good answer? It is almost certainly wrong in the sense that the population average age in the small village is probably not fifteen (Italy's population is aging!), but we don't know how wrong because, of course, we don't actually know the value of the inquiry under study. We have instead to evaluate the properties of the procedure. Under a random sampling design – even an egregiously stingy random sampling design that only selects three citizens! – we can justify the approach on the basis of the “bias” diagnosand. The average of all the answers you would get if you repeated the data strategy (random sampling) and the answer strategy (taking the sample average) over and over would correspond to the correct answer.

Though the bias diagnosand is zero, the variance diagnosand is not. Over many repeated draws, the estimates bounce around dramatically. The design in Declaration 9.1 can be used to generate a view of what answers we might get for a particular distribution when we choose just three subjects for our sample. We use a linear regression of the age variable on a constant to estimate the sample mean. Using OLS in this way is a neat trick for estimating sample means along with various uncertainty statistics, discuss in more detail in the next section.

Declaration 9.1. Italian village design

```
design <-  
  declare_model(N = 100, age = sample(0:80, size = N, replace = TRUE)) +  
  declare_inquiry(mean_age = mean(age)) +
```

```
declare_sampling(S = complete_rs(N = N, n = 3)) +
  declare_estimator(age ~ 1, model = lm_robust)
```

```
simulations <- simulate_design(design, sims = sims)
```

Figure 9.1 shows that we are right on average but wrong a lot. The average estimate lies right on top of the true value of the estimand (40), but the estimates range enormously widely, from close to zero to close to 80 in some draws. The answer strategy – the sample mean estimator – is just fine, the problem here lies in the data strategy that generates tiny samples.

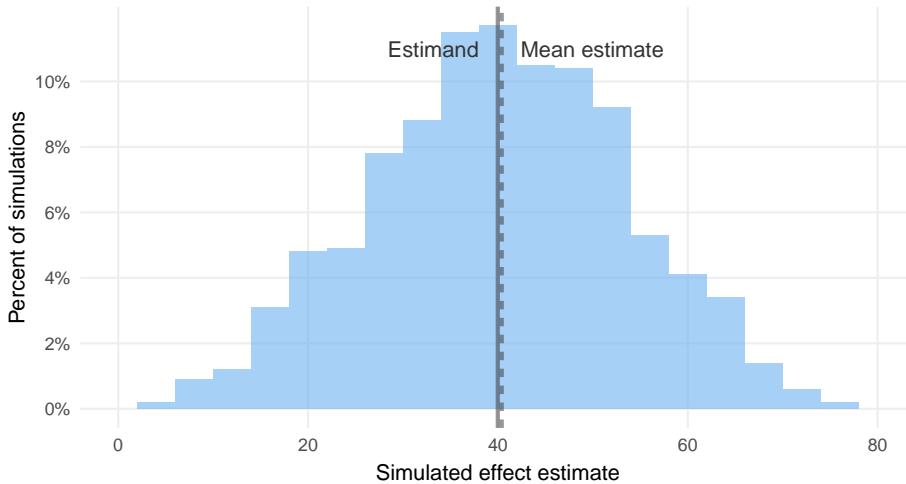


Figure 9.1: Distribution of the point estimator answer strategy

9.1.2 Tests

Tests are an elemental kind of answer strategy. Tests yield binary yes/no answers to a binary yes/no inquiry. In some qualitative traditions, hoop tests, straw-in-the-wind tests, smoking-gun tests, and doubly-decisive tests are common. These tests are procedures for making analysis decisions in a structured way. Similarly, many forms of quantitative tests have been developed. Sign tests assess whether a test statistic is positive, negative, or zero. Null hypothesis significance tests assess whether a parameter is different than a null value, such as zero. Equivalence tests assess whether a parameter falls within a range,

rather than comparing to a fixed value. Many procedures for conducting tests are also available, with different assumptions about the null hypothesis, the distributions of variables, and the data strategy.

A typical null hypothesis test proceeds by imagining a null model M_0 and imagining the sampling distribution of the empirical answer a_{d_0} under a hypothetical design M_0IDA . That sampling distribution enumerates all the ways the design could have come out if the null model M_0 were the correct one. For a null hypothesis test, we *entertain* the null model and consider its implications. We ask, under M_0 how frequently would we obtain an answer as large or larger than the empirical answer a_d (or other test statistic)? That frequency is known as a *p-value*.¹ The last step of the test is to turn the *p*-value into a binary significance choice. The typical threshold in the social sciences is 0.05: hypothesis test with *p*-values less than 0.05 indicate statistical significance. This threshold is arbitrary, reflecting the inertia of the scientific community much more than some a priori scientific standard. The appropriate threshold value for statistical significance is a matter of furious debate, with some authors calling for the threshold to be lowered to 0.005 to guard against false positives (Benjamin et al., 2018).

We'll illustrate the idea of a hypothesis test in general with the Italian village example. Here, we test against the hypothesis that the average age is 20. If we have strong evidence against this hypothesis, we will reject it. If we have weak evidence against the hypothesis, we will fail to reject it. For instance, we might reject 10 and 70 but fail to reject 35 and 45.

Declaration 9.2. Italian village design continued

```
design <-  
  design +  
  declare_test(age ~ 1,  
              linear_hypothesis = "(Intercept) = 20",  
              model = lh_robust, label = "test")
```

Here's one run of that design. The output can be confusing. By default, most statistical software tests against the null hypothesis that the true parameter value is zero – so the *p.value* in the first row refers to that null hypothesis test. The second row is the test against the hypothesis that the mean is equal to twenty. The “estimate” in the second row is the difference of the observed estimate from 20. The *p.value* in the second row is the one we care about when testing against the null hypothesis that the average age is 20.

Figure 9.2 shows how frequently we reject the null that the average age is 20.

¹The *p*-value can be thought of as a diagnosand of the M_0IDA design. If M_0 were true, what fraction of simulations would generate answers as big as some value?

Table 9.1: Estimates from the test of the mean age equalling 20 from Italian villages example

estimator	estimate	p.value
estimator	37	0.05
test	17	0.19

When estimate is close to 20, we rarely reject the null, but when the estimate is far from 20, we are more likely to reject. Again, this simulation comes from a design with a weak data strategy of sampling only three citizens at time. We need to see estimates breaking 60 before the testing answer strategy reliably rejects this (false) null hypothesis.

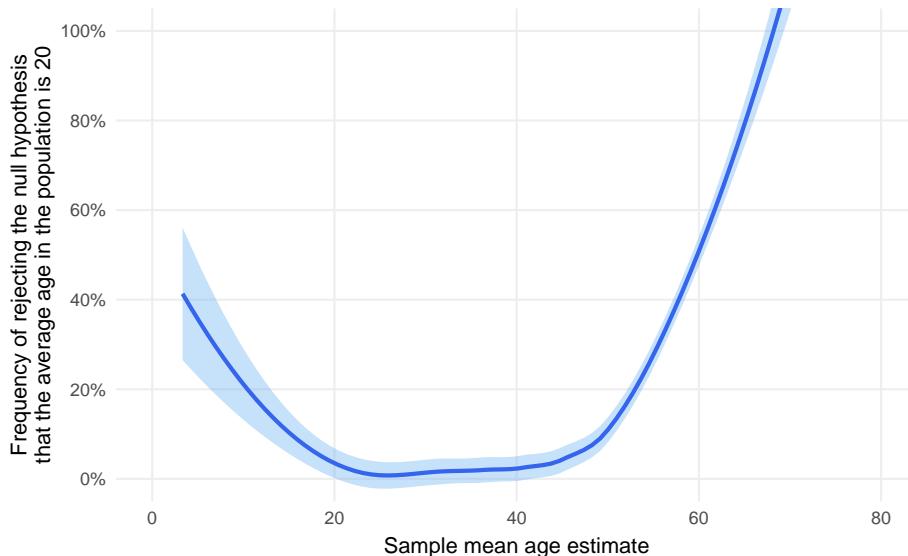


Figure 9.2: Distribution of the test answer strategy

9.1.2.1 Randomization inference

Randomization inference describes a large class of testing procedures that merit special attention. Randomization inference tests leverage known features of the randomization procedure to implement tests of many kinds (see Gerber and Green (2012), chapter 3, for an introduction to randomization inference). In a common case, a randomization inference test proceeds by stipulating a null model under which the counterfactual outcomes of each unit are exactly equal to the observed outcomes, the so-called “sharp null hypothesis of no effect.” Under this null hypothesis, the treated and untreated potential outcomes are

Table 9.2: Results from @clingingsmith2009estimating study

term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome
success	0.47	0.16	2.92	0	0.16	0.79	954.3	views

exactly equal for each unit, reflecting a model in which the treatment has exactly zero effect for each unit.

As described above, a *p*-value is an answer to the question: what is the probability the null model would generate estimates as large or larger in absolute value than the observed estimate? We can answer this question by diagnosing the design under the sharp null model. We take the example of exercise 3.6 from Gerber and Green (2012), which prompts students to evaluate the sharp null hypothesis of no effect in the context of Clingingsmith, Khwaja and Kremer (2009)'s study of the effect of being randomly assigned to go on Hajj on tolerance of foreigners.

Here is the observed estimate, which is positive, indicating that our best guess is that going on Hajj increases tolerance.

```
clingingsmith_eta <- read_csv("data/ clingingsmith_et_al_2009.csv")
observed_estimate <-
  difference_in_means(views ~ success, data = clingingsmith_eta)
observed_estimate
```

Here we declare the null model and add it to the data and answer strategies:

Declaration 9.3. Randomization inference under the sharp null

```
null_model <-
  declare_model(data = clingingsmith_eta,
                potential_outcomes(Y ~ 0 * Z + views))

null_design <-
  null_model +
  declare_assignment(Z = complete_ra(N = N, m = sum(success))) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z)
```

Table 9.3: Diagnosands of randomization inference example

design	estimator	term	p.value	se(p.value)	n_sims
null_design	estimator	Z	0	0	1000

We diagnose the null design with respect to the p.value diagnosand: what fraction of simulations under the null model exceed the observed estimate? We find that the p.value is low, suggesting that the observed estimate is unlikely to have been generated under the null model.

```

p.value <-
  declare_diagnosands(
    p.value = mean(abs(estimate) >= abs(observed_estimate$coefficients))
  )

nhst <-
  diagnose_design(null_design, diagnosands = p.value, sims = 1000)

get_diagnosands(nhst)

```

9.1.3 Bayesian formalizations

Bayesian answer strategies sometimes target the same inquiries as classical approaches, but rather than seeking a point estimate, they try to generate rational beliefs over possible values of the estimand. Rather than trying to provide a single best guess for the average age in a village, a Bayesian answer would try to figure out how likely different answers are given the data. To do so they need to know how likely different age distributions are *before* seeing the data—the priors—and the likelihood of different types of data for each possible age distribution. A Bayesian would likely not be very impressed by the 15 answer given by the point estimator in Section 9.1.1 because, prior to see any samples, they would likely expect that the answer had to be bigger than this. Bayesians would chalk the answer “15” down to an unusual draw.

A Bayesian answer strategy might look like this. Here we need to modify the default model summary option to deal with the fact that the estimates are returned in log form by default.

The Bayesian answer strategy specifies a prior distribution (here distributed normal centered on 50 to reflect a prior that Italian villages skew older) as well as a log normal distribution for ages. Here we retain the (median) posterior estimates for average age alongside a standard error based on the posterior variance. In the model_summary argument we ask the tidier to exponentiate the

coefficient estimate and standard error before returning them.

Declaration 9.4.

```
library(rstanarm)
library(broom.mixed)

design <-
  declare_model(N = 100, age = sample(0:80, size = N, replace = TRUE)) +
  declare_inquiry(mean_age = mean(age)) +
  declare_sampling(S = complete_rs(N = N, n = 3)) +
  declare_estimator(
    age ~ 1,
    model = stan_glm,
    family = gaussian(link = "log"),
    prior_intercept = normal(50, 5),
    model_summary = ~tidy(.., exponentiate = TRUE),
    inquiry = "mean_age"
  )
```

We can then simulate this design in the same way and examine the distribution of estimates we might get.

```
simulations <- simulate_design(design, sims = sims)
```

What we see in Figure 9.3 is that using the same (poor) data strategy as before, a Bayesian answer strategy gets us a somewhat tighter distribution on our answer, but exhibits greater bias: the average estimate is higher than the estimand. We might accept higher bias for lower variance if overall, the root-mean-squared error is lower for the Bayesian approach. See Section 10.6 for a further discussion of RMSE. The main difference between the Bayesian and classical approaches is the handling of prior beliefs, which carry a lot of weight in the Bayesian estimation but no weight in the classical approach.

Bayesian approaches are also used by qualitative researchers drawing case level inferences from causal process observations. Recent developments in qualitative methods have sought to take Bayes' rule “from metaphor to analytic tool” (Bennett, 2015). This approach characterizes qualitative inference as one in which prior beliefs about the world can be specified numerically and then are updated on the basis of evidence observed. At a minimum, writing down such an answer strategy on a computer requires specifying beliefs, expressed as probabilities, about the likelihood of seeing certain kinds of evidence under different

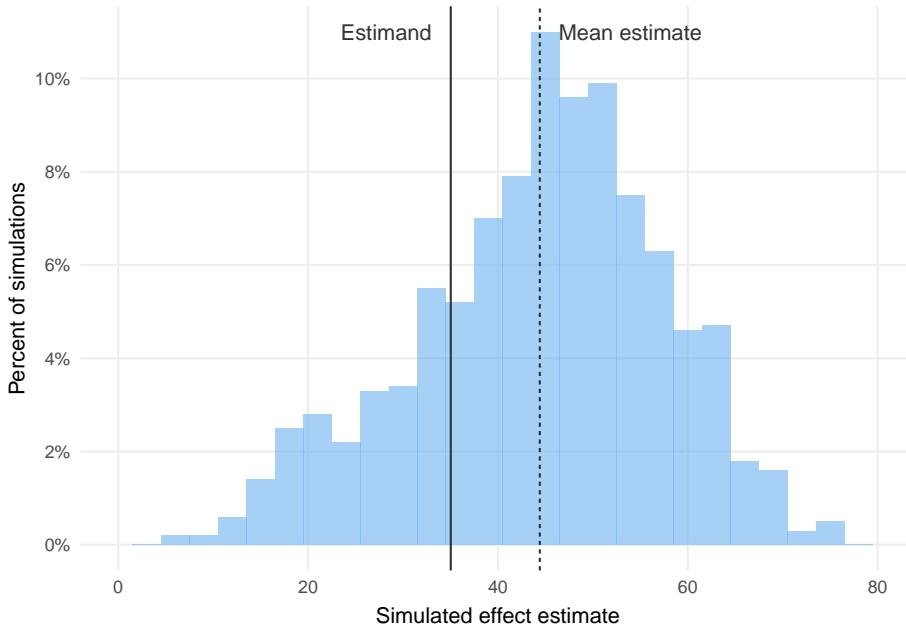


Figure 9.3: Sampling distribution of the Bayesian answer strategy

hypotheses. We provide an example of such a strategy the design library Section 15.1. Herron and Quinn (2016) provide one approach to formalizing a qualitative answer strategy that focuses on understanding an average treatment effect. Humphreys and Jacobs (2015) provide an approach that can be used to formalize answer strategies targeting both causal effect and causal attribution inquiries, while Fairfield and Charman (2017) formalize a Bayesian approach that approaches causal attribution as a problem of attaching a posterior probability to competing alternative hypotheses. Abell and Engel (2019) suggest the use of “supra-Bayesian” methods to aggregate multiple participant-provided narratives in ethnographic studies targeting causal attribution estimands.

9.1.4 Interval estimation

In many circumstances, the details of the data strategy alone are insufficient to pin down a parameter value exactly, in which case the parameter is not “point-identified,” which means we can’t generate a point estimate without further assumptions. The standard approach is to simply make those further assumptions and move on to reporting point estimates. Under an agnostic approach – we don’t know if those assumptions are right because they aren’t grounded in the data strategy – we can turn to interval estimation instead.

One way to handle settings in which parameters are not point-identified is to generate “extreme value bounds.” These bounds report the best and worst

Table 9.4: Extreme value bounds estimate

Lower bound	Upper bound
0.45	107.15

Table 9.5: Tigher extreme value bounds estimate with more data

Lower bound	Upper bound
39.6	50.6

possibilities according to the logical extrema of the outcome variable.

We illustrate interval estimation back in our Italian village where we have learned the ages of three of the 100 citizens. Suppose we *did not know* whether the data strategy used random sampling, so we can't rely on the guarantee that, under random sampling, the sample mean is unbiased for the population mean. Now we have reason about best and worst case scenarios. Let's agree that the youngest a person can be is zero and the oldest is 110. Starting with an estimate of 15 among three citizens, we can generate lower and upper bound estimates for the average age of the entire 100-person village like this:

```
lower_bound <- (3 * 15 + 97 * 0)/100
upper_bound <- (3 * 15 + 97 * 110)/100
c(lower_bound, upper_bound)
```

This procedure generates enormously wide bounds – we already knew before we started that the average age had to be somewhere between 0.45 and 107.15 years. But consider if we had data on 90 of the 100 citizens and among those 90, the average is 44. Now when we generate the bounds, they are still wide but not ridiculously so – the bounds put the average age somewhere between 40 and 50.

```
lower_bound <- (90 * 44 + 10 * 0)/100
upper_bound <- (90 * 44 + 10 * 110)/100
c(lower_bound, upper_bound)
```

Extreme value bounds and variations on the idea can be applied when experiments encounter missingness or when we want to estimate effects among subgroups that only reveal themselves in some but not all treatment conditions (see

Aronow, Baron and Pinson (2019) or Coppock (2019) for examples). The extreme value bound approach can also be used in qualitative settings in which we can impute some but not all of the missing potential outcomes using qualitative information; the bounds reflect our uncertainty about those missing values (Coppock and Kaur, 2021).

9.2 Uncertainty

Answers should usually be accompanied by uncertainty estimates. We want to communicate to others not just what our answer is, but also how certain we are of it. Our uncertainty about our answers stems from the properties of a design: when designs are stronger, our uncertainty is smaller. One reason to communicate the uncertainty associated with your estimates is to communicate the strength of your design.

For point and interval estimators, uncertainty is often expressed as a standard error estimate or confidence interval. Many approaches to standard error estimation are available. Indeed, just like point estimators for inquiries, we have point estimators for standard errors. You might choose classical standard errors or cluster-robust standard errors; you might bootstrap your standard errors or use the jackknife. Similarly, many approaches to confidence interval construction are available. Most often, confidence intervals are built from variance estimates under an appeal to sampling theory. Alternatively, a confidence interval can be formed by “inverting the test,” i.e. finding the range of null hypotheses we fail to reject. Whether any particular approach to uncertainty estimation is appropriate in a context will depend on the full set of design parameters and we encourage you to diagnose your uncertainty estimates as well.

Here is the output of the answer strategy from Declaration 9.1, applied to the realized data set, rounded to the nearest whole number. We see the sample mean estimate of 15, the standard error estimate of 6, and the confidence interval from -10 to 40. These numbers communicate that our answer is 15 – but also that we *know* that number is shaky. We’re uncertain because the tool we used to answer the inquiry is high variance: it could bounce around a lot depending on which three people we happened to sample.

```
three_italian_citizens <- fabricate(N = 3, age = c(5, 15, 25))
answer_strategy <- declare_estimator(age ~ 1)
answer_strategy(three_italian_citizens)
```

For tests, uncertainty is expressed by describing the properties of a procedure in terms of error rates. A test is an answer strategy that returns a binary answer to a binary estimand. The result of a test is an error if the empirical answer a_d does not equal the truth a_m . Conventionally, a Type 1 error occurs when $a_d = 1$

Table 9.6: One draw of the answer strategy

estimate	std.error	statistic	p.value	conf.low	conf.high
15	5.77	2.6	0.12	-9.84	39.84

but $a_m = 0$ and a Type 2 error occurs when $a_d = 0$ but $a_m = 1$. A perfect test (i.e., a test about about which we are fully certain) has Type 1 error rate of 0% and a Type 2 error rate of 0% as well. A test about which we are less certain might return $a_d = 1$ 40% of the time when $a_m = 0$ (a Type I error rate of 40%) and might return $a_d = 1$ 90% of the time when $a_m = 1$ (a Type II error rate of 10%).

The test reported by the `answer_strategy` function is obscured by its presentation, but because this sort of presentation is so common in social science, it's worth showing raw. The test implicit in the output is a null hypothesis significance test against the null hypothesis that the average age in this Italian village is equal to exactly zero. The test returns "yes" if we reject the null and "no" if we fail to reject it.² If we use the standard significance threshold of $= 0.05$ we fail to reject the null model because the `p.value` reported in the table is 0.12.

Our uncertainty about the decision we made in the hypothesis test to fail to reject is *not* represented by the information in the table. Importantly, the `p.value` does *not* represent the probability that the null model is correct. The `p.value` is the probability that with our data and answer strategy, draws from the null model would lead to estimates of 15 or larger. According to our calculations, draws from the null model will do so 12 percent of the time. We use this probability along the way to making a decision about whether to reject the null model, but amazingly, a `p.value` does *not* describe our certainty about the significance test!

What does characterize our uncertainty about a significance test? The Type I and type II error rates of the test. The Type I error rate is controlled by the significance threshold. A Type I error occurs if we reject the null model when it is true. If we use $= 0.05$ and the test is correctly accounts for all design elements, then a Type I error should only happen 5% of the time. Type II error rates are harder to learn about. In our case, we failed to reject the null model. To characterize our uncertainty about the test, we also want to calculate the probability that a design like this one would generate Type II errors. To do so, we have to imagine what it means for the null model to be *false*, since they can be false in many ways. One approach is to imagine how the test would perform on under a series of non-null models.

²It's a silly test, but silly tests like these are reported by default in many statistical software languages and in many scientific papers to boot. It's a silly test because we always knew the average age was not zero!

Figure 9.4 describes the Type II error rate over a range of non-null models. If the true population mean is around 25 or lower, we fail to reject the null 75% of the time or more. With this comically small sample size, even if the true mean were 75, we would still fail to reject 20% of the time. We are rightly uncertain about this test – it may have a low enough Type I error rate (set as $\alpha = 0.05$), but the Type II errors are way too big.

Declaration 9.5.

```
design <-  
  declare_model(N = 100,  
                age = round(rnorm(N, mean = true_mean, sd = 23))) +  
  declare_inquiry(mean_age = mean(age)) +  
  declare_sampling(S = complete_rs(N = N, n = 3)) +  
  declare_estimator(age ~ 1, model = lm_robust)
```

```
designs <- redesign(design, true_mean = seq(0, 100, length.out = 10))
```

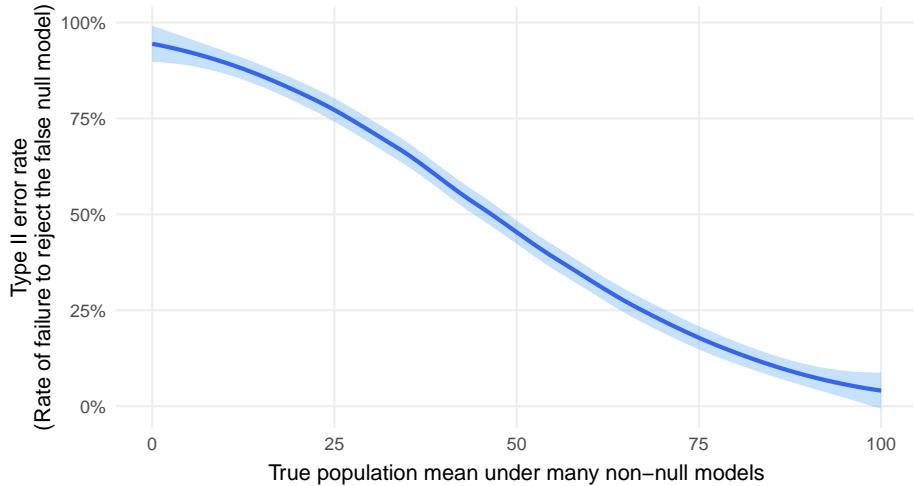


Figure 9.4: Type II error rates of Italian village design

9.2.1 Uncertainty estimates as diagnosands

In this section, we encourage you to think of uncertainty statistics as empirical estimates of diagnosands describing the quality of your design under some set

of models.

Suppose we report that the estimate is equal to 15 and carries with it a standard error of 6. Here we are communicating that we now think our design has the property that, if we were to run it over and over, the standard deviation of our estimates would be 6. That is, if we were to declare a design over a large class of models that all have the same outcome variance and combined it with I , D , and A , and diagnosed, the standard deviation of the estimates under all those models would be 6.

Likewise, estimated p -values can also be thought of as estimates of a peculiar diagnosand: the probability of obtaining a test statistic of a particular value, under a maintained null model. That is, if we were to write down a null model m_0 under which the estimand were in fact zero, then combined it with I , D , and A , we could diagnose that design to learn the fraction of estimates that are as large or larger than 15 – a p -value.

Thinking about p -values as diagnosands can be seen most directly in the randomization inference approach to hypothesis testing. Randomization inference often considers “sharp” null hypothesis in which we impute the missing potential outcomes with exactly their observed values, then simulate all the ways the design could come out holding I , D , and A constant. The p -value is the fraction of simulations in which the null model generates estimates that are more extreme than the observed estimate. See Section 9.1.2.1 for a worked example of this approach.

Even frequentist confidence intervals, with their notoriously confusing interpretation (a 95% confidence interval is an interval that has the goal of covering the estimand 95% of the time) can be viewed as estimates of the 2.5th and 97.5th quantiles of the sampling distribution under the model that the estimand equals the estimate.

The payoff from thinking about uncertainty statistics as estimates of diagnosands is that uncertainty statistics can be **poor** estimators of diagnosands. For example, social scientists criticize one another’s choice of standard error – classical or robust, clustered or not, bootstrapped, jackknifed, and on and on. The reason for this debate is that when the procedures we follow to form uncertainty statistics are inappropriate for the design setting, we can be falsely confident in our answers.

Figure 9.5 illustrates what’s at stake in the age old contest between classical standard errors and robust standard errors. Here we are in the context of a two arm trial under three settings: the control potential outcomes are higher variance, the groups have the same variances, or the treated outcomes have higher variance. The calculation for classical standard errors pools the data from both groups when estimating a variance, thereby assuming “homoskedasticity,” a Greek work for having the “same spread.” This estimation choice leads to poor performance. Depending on the fraction of the sample that receive treatment, the estimate of the sampling distribution can be upwardly biased

(conservative) or downwardly biased (anti-conservative or falsely confident). We're usually worried about standard error estimates being too small because anti-conservatism is probably worse for scientific communication than conservatism.

Robust standard errors are “heteroskedasticity-robust,” which means that the calculation does not pool the data from the two groups. Samii and Aronow (2012) show that a particular variant of robust standard errors (HC2) is exactly equal to the Neyman variance estimator described in Section 17.1.1, which is why HC2 robust standard errors are default in the `lm_robust` function. The bottom row of Figure 9.5 shows that the robust standard error estimator hews closely to the true value of the diagnosand in all of the design settings considered. We prefer robust SEs because they do a better job of estimating the standard deviation diagnosand in more settings.

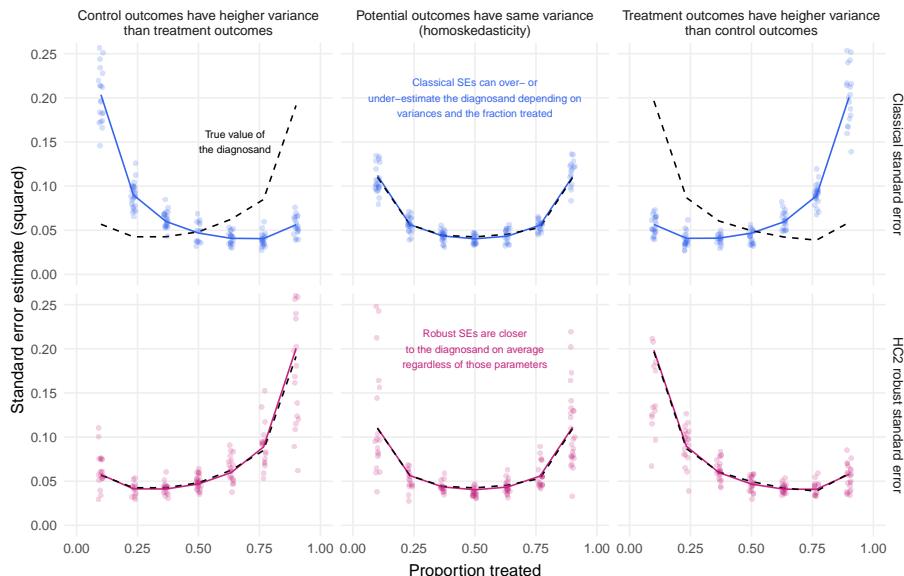


Figure 9.5: Why robust standard errors are preferred to classical standard errors

9.3 Applying design principles

Now that we have discussed all four research design elements in detail, we can return to some of the design principles laid out in Chapter 3.

9.3.1 Declare data and answer strategies as functions

Both the data and answer strategies are functions. What we mean by this is that they respond to their inputs. If the events generated by the world had been

different, the data produced by the data strategy would be different too. If the data produced by the data strategy had been different, the answers rendered by the answer strategy would be different too. These design elements are procedures and we want to understand the properties of those procedures over many possible ways the world could be. We can't do that unless we declare data and answer as functions.

When declaring answer strategies as functions, we have to think about more than just the single estimation function that ends up in the final paper. To see this, consider an estimator that is selected through an exploratory procedure in which multiple estimators are compared on the basis of fit statistics. The answer strategy is not this final estimator – it is this entire multi-step if-then procedure.

The reason to declare the procedure rather than the final estimator is that the diagnosis of the design may differ. The procedure may be more powerful, if for example we assessed multiple sets of covariate controls and selected the specification with the lowest standard error of the estimate. But that procedure would also exhibit poor coverage, since the confidence interval produced by the final estimator does not account for these multiple bites at the apple.

Answer strategies can become multi-stage procedures in unexpected ways. For example, sometimes a planned-on maximum likelihood estimator won't converge when executed on the realized data. In these cases, analysts switch estimators (or sometimes inquiries!). The full set of steps — a decision tree, depending on what is estimable — is the answer strategy we want to declare and compare to alternative decision trees.

This principle extends to settings in which analysts run diagnostic tests, like falsification or placebo tests. If we learn from a sensitivity test that a mediation estimate is very sensitive to unobserved confounding, we might choose not to present it at all. By this logic, the answer strategy includes the sensitivity test, the decision made on the basis of the test, and the resulting distribution of mediation estimates, some of which are undefined.

Writing down the full set of if-then choices you might make in the answer strategy depending on revealed data is hard to do. It's hard to do because it's difficult we often imagine answer strategies if things go well but spend less imagination on elaborating what might happen if things go wrong. When things do go wrong — missing data, noncompliance, suspension of the data collection — answer strategies will change. One way to guard against over-correcting to the revealed data is to write down a *standard operating procedures* document that systematize these procedures in advance (Green and Lin, 2016).

9.3.2 Seek M:I:D:A parallelism

The model and the inquiry form the empirical half of the design, and the data and answer strategies make up the empirical half. Research designs that have

parallel theoretical and empirical halves tend to be strong (though not all strong designs need be parallel in this way). This principle is motivated by the intersection of two ideas from statistics: the “plug-in principle” and “analyze as you randomize.”

The plug-in principle refers to the idea that sometimes, the answer strategy function and the inquiry function are very similar in form. The estimand, $I(m) = a_m$, can often be estimated by choosing an A that is very similar to I and then “plugging-in” the realized data d that result from the data strategy for the unobserved data m , i.e. $A(d) = a_d$.

More formally, Aronow and Miller (2019) describe a plug-in estimator as:

For i.i.d. random variables X_1, X_2, \dots, X_n with common CDF F , the plug-in estimator of $= T(F)$ is: $= T(F)$.

For example, suppose that our inquiry is the average treatment effect among the N units in the population.

$$I(m) = \frac{1}{N} \sum_{i=1}^N [Y_i(1) - Y_i(0)] = \frac{1}{N} \sum_{i=1}^N Y_i(1) - \frac{1}{N} \sum_{i=1}^N Y_i(0) = \text{ATE}$$

We can develop a plug-in ATE estimator by replacing the population means — $\frac{1}{N} \sum_{i=1}^N Y_i(1)$ and $\frac{1}{N} \sum_{i=1}^N Y_i(0)$ — with sample analogues:

$$A(d) = \frac{1}{m} \sum_{i=1}^m Y_i - \frac{1}{N-m} \sum_{i=m+1}^N Y_i,$$

where units 1 through m reveal their treated potential outcomes and the remainder reveal their untreated potential outcomes.

Following plug-in principle only yields good answer strategies under some circumstances. Those circumstances are determined by the data strategy. In order to seek M:I:D:A parallelism, we need data strategies that sample units, assign treatment conditions, and measure outcomes such that the revealed data can indeed be “plugged in” to the inquiry function. Whether this plug-in ATE estimator is a good answer strategy depends of features of the data strategy. It’s a good estimator when units are assigned to treatment with equal probabilities, but it’s a bad estimator if the probabilities differ.

When the data strategy introduces distortions like differential probabilities of assignment, the answer strategy function should *not* equal the inquiry function: we can no longer just plug in the observed data. In order to seek parallelism, we should adjust for those distortions, reversing them to reestablish parallelism.

This idea can be summarized as “analyze as you randomize,” a dictum attributed to R.A. Fisher. We use known features of the data strategy to adjust the answer strategy. We can undo the distortion introduced by differential probabilities of assignment by weighting units by the inverse of the probability of being in the condition that they are in. If we use an inverse-probability weighted (IPW) estimator, we restore parallelism because even though A no longer equals I , the relationship of D to A once again parallels the relationship of M to I .

Declaration 9.6 illustrates this idea. We declare the theoretical half of the design as MI then consider the intersection of two data strategies with two answer strategies. D1 has constant probabilities of assignment and D2 has differential probabilities of assignment. A1 is the plug-in estimator and A2 is the IPW estimator with the inverse probability weights generated by the D2 randomization protocol.

Declaration 9.6. Seeking parallelism design

```

MI <-
  declare_model(
    N = 100,
    X = rbinom(N, size = 1, 0.5),
    U = rnorm(N),
    potential_outcomes(Y ~ 0.5 * Z + 0.5 * X + 0.5 * X * Z + U)
  ) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))

D1 <- declare_assignment(Z = complete_ra(N = N)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z))
D2 <- declare_assignment(Z = block_ra(blocks = X, block_prob = c(0.1, 0.8))) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z))

A1 <- declare_estimator(Y ~ Z, label = "Unweighted")
A2 <-
  declare_step(
    handler = fabricate,
    ipw = 1 / obtain_condition_probabilities(
      assignment = Z,
      blocks = X,
      block_prob = c(0.1, 0.8)
    )
  ) +
  declare_estimator(Y ~ Z, weights = ipw, label = "Weighted")

designs <- list(MI + D1 + A1,
                 MI + D1 + A2,
                 MI + D2 + A1,
                 MI + D2 + A2)

```

We diagnose the bias of all four design. Figure 9.6 shows that when the answer strategy and the data strategy match ($D1 + A1$ and $D2 + A2$), we have no bias.

When they do not match ($D_1 + A_2$ and $D_2 + A_1$), we do. In this case, seeking parallelism in the choice of answer strategy improves the design. Of course, an alternative answer strategy we might call A_3 that implements the weights corresponding to whatever the data strategy says they should be unbiased under both D_1 and D_2 .

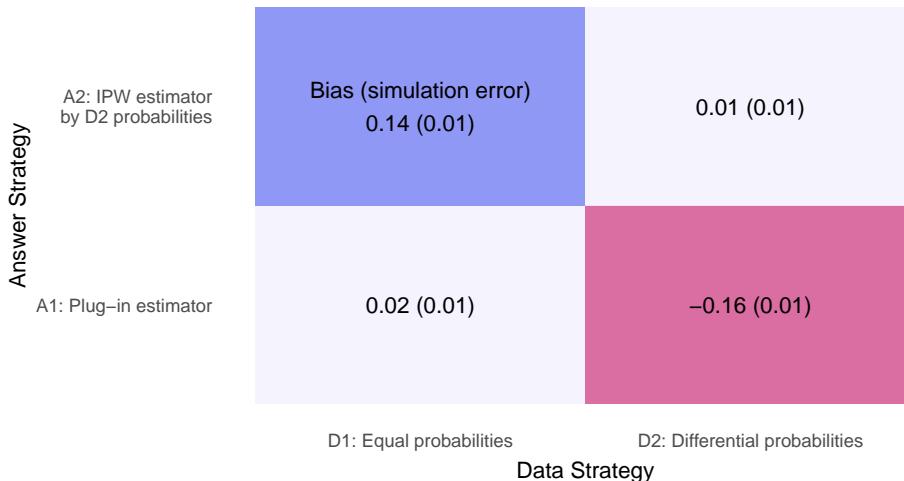


Figure 9.6: When data and answer strategies are mismatched, we obtain bias.

This principle applies most clearly to the bias diagnosand, but it applies to others as well. For example, Abadie et al. (2017) recommend that answer strategies should include clustered standard errors at the level of sampling or assignment, whichever is higher. The data strategies that include clustering introduce a dependence among units that was not present in the model; clustered standard errors account for this dependence. If we did not do so, our estimated standard error would be a bad estimate of the “standard deviation” diagnosand.

9.3.3 Entertain many models

One of the main reasons to “entertain many models” is that we want to choose answer strategies that are “robust” to models. By robust, we mean that the answer strategy should produce good answers under a wide range of models. Selecting answer strategies that are robust to multiple models ensures that we not only get good answers when our model is spot on — which is rare! — but under many possible circumstances.

Understanding whether the choices over answer strategies — logit or probit or OLS — depend on the model being a particular way is crucial to making a choice. For example, many people have been taught that whenever the outcome variable is binary, OLS is inappropriate they must use a binary choice model like logit instead. When the inquiry is the probabilities of success for each unit and

we use covariates to model them, how much better logit performs at estimating probabilities depends on the model. When probabilities are all close to 0.5, the two answer strategies both perform well. When the probabilities spread out from 0.5, OLS is less robust and logit beats it. In the same breath, however, we can consider these same two estimators in the context of a randomized experiment with a binary outcome. Here, OLS is just as strong as logit, no matter the distribution of the potential outcomes. In this setting, when we entertain many models, we find that both estimators are robust.

Entertaining many models has something in common with robustness checks: both share the motivation that we have fundamental uncertainty about the true model. A robustness check is an *alternative* answer strategy that changes some model assumption that the main answer strategy depends on. Presenting three estimates of the same parameter under different answer strategies (logit, probit, and OLS) and making a joint decision based on the set of estimates about whether the main analysis is “robust” is a procedure for assessing “model dependence.” But robustness checks are just answer strategies themselves, and we should declare them and diagnose them to understand whether they are good answer strategies. We want to understand the *properties* of the robustness check, e.g., under what models and how frequently does it correctly describe the main answer strategy as “robust.”

9.4 Declaring answer strategies in code

An answer strategy is a function that provides answers to an inquiry. Declaring one in code involves selecting that function and linking the answer or answers it returns to one or more inquiries.

The functions we declare in `DeclareDesign` for answer strategies differ from those for the other elements of research design to reflect the two-step nature of many answer strategies. Often, first a statistical model (e.g., a linear regression) is fit to data, and then summaries of that model fit (e.g., the coefficient on a variable X , its standard error, t -statistic, p-value, and confidence interval) are combined to form an answer and its associated measures of uncertainty.

9.4.1 Statistical modeling functions

In `DeclareDesign`, we call these two steps `model` and `model_summary`. (This usage is a little confusing, since this “model” refers to the “statistical modeling function” and not the research design element, but so be it.) The `model` argument in `declare_estimator` can take almost any modelling function in R (e.g., `lm` for linear regression) and `model_summary` consists of a summary function that calculates statistics from the fit such as `tidy` or `glance`. You can write your own `model` or `model_summary`. When your answer strategy does not fit this two-step structure, you can as with all `declare_` functions write your own handler.

Table 9.7: The estimate from one draw of the linear regression design

estimator	term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome	inquiry
lm_no_controls	Z	0.02	0.18	0.13	0.89	-0.34	0.39	98	Y	ATE

We break down each part of a standard answer strategy declaration using the example of a linear regression of the effect of a variable Z on an outcome Y in Declaration @ref(def:answerstrategyincode}. The first argument in our `declare_estimator` step defines the model we will use, here `lm_robust` which is our function in the `estimatr` package for running linear regressions with robust standard errors. The second is the main argument for `lm_robust`, the formula for the regression specification, in this case Y on Z with no controls.

Declaration 9.7. Linear regression design

```
design <-
  declare_model(
    N = 100, U = rnorm(N), potential_outcomes(Y ~ 0.2 * Z + U)
  ) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(model = lm_robust,
    formula = Y ~ Z,
    model_summary = tidy,
    term = "Z",
    inquiry = "ATE",
    label = "lm_no_controls")
```

```
draw_estimates(design)
```

9.4.2 Tidying statistical modelling function output

Let's unpack the `model_summary` argument. In this case we sent the `tidy` function from the `broom` package (the default). Understanding what `tidy` does opens a window into the way we match estimates and estimands. The `tidy` function takes many model fit objects and returns a data frame in which rows represent estimates and columns represent statistics about that estimate. The columns typically include the estimate itself (`estimate`), an estimated standard

Table 9.8: The model fit statistics from one draw of the linear regression design

r.squared	adj.r.squared	statistic	p.value	df.residual	nobs	se_type
0.01	0	1.05	0.31	98	100	HC2

error (`std.error`), a test statistic of some kind reported by the model function such as a *t*-statistic or *Z*-statistic (`statistic`), a p-value based on the test statistic (`p.value`), a confidence interval (`conf.low`, `conf.high`), and the degrees of freedom of the test statistic if relevant (`df`).

A key column in the output of `tidy` is `term`, which represents which coefficient (term) is being described in that row. The term column uniquely identifies the row. We will often need to use the term column in conjunction with the name of the estimator to link estimates to estimands when there are more than one. If in the regression we pull out two coefficients (e.g., for treatment indicator 1 and for treatment indicator 2), we need to be able to link those to separate inquiries representing the true effect of treatment 1 and the true effect of treatment 2. `Term` is our tool for doing so. The default is for `term` to pick the first coefficient that is not the intercept, so for the regression $Y \sim Z$ there will be an intercept and then the coefficient on Z which is what will be picked.

The `inquiry` argument defines which inquiry or inquiries the estimates will be linked to. In this case, we link to a single inquiry, the ATE. You can also declare an estimator that shoots at multiple inquiries: `declare_estimator(Y ~ Z, model = lm_robust, term = "Z", inquiry = c("ATE", "ATT"))` - useful for learning how well an estimator does for different targets. When we run the answer strategy on data, we get two additional pieces of information tacked on to the model summary data frame: the name of the estimator, which comes from the `label` argument, and the inquiry. The unit of analysis of a diagnosis is the inquiry-estimator pair, so if you link an estimator to multiple inquiries, then there will be a row for each inquiry.

Tidy is not the only commonly-used model summary function. `glance` will provide model fit statistics such as the r-squared:

```
A <- declare_estimator(Y ~ Z,
                      model = lm_robust,
                      model_summary = glance)
A(draw_data(design))
```

When neither `tidy` nor `glance` works well for your answer strategy, you can write your own model summary function. Below, we slowly build up a `tidy` function for the `lm` model. (One is already built-in to the `broom` package, but

we do so here to illustrate how you can write your own for a function that does not already have one.) Before you start to write your own summary function, check whether one exists on the *Broom* Web site.

There are three sets for a tidy function:

1. Pull out statistics from the model fit object. You can extract out any statistics and transform them in any relevant way.
2. Return a data frame (or `tibble`).
3. Name your estimates in a common format that works across all tidy functions. The estimate column should be called “estimate”, the standard error column “std.error”, etc., as described earlier. However, if you want to add statistics from the model fit that you will diagnose, you can and you can name them whatever you want.

```
tidy_lm <- function(fit) {
  # calculate estimates by grabbing the coefficients from the model fit
  estimate <- coef(lm)

  # get the names of the coefficients (e.g., "(Intercept)", "Z")
  # we will call these "term" to represent regression terms
  term <- names(estimate)

  # calculate the standard error by grabbing the variance-covariance
  # matrix, then pulling the diagonal elements of it and taking the
  # square root to transform from variances to standard errors
  std.error <- sqrt(diag(vcov(lm)))

  # return a tibble with term, estimate, and std.error
  tibble(term = term, estimate = unlist(estimate), std.error = std.error)
}

declare_estimator(
  Y ~ Z
  model = lm,
  model_summary = tidy_lm
)
```

In other cases, you may want to build on functions that interoperate with the `broom` functions to do specialized summary tasks like calculating marginal effects or predicted effects. The `margins` function from the `margins` package calculates marginal effects and the `predictions` package from the `predictions` package are especially useful and work well with the `tidy` workflow. To calculate marginal effects, run `margins` and then `tidy` as your model summary:

```

tidy_margins <- function(x) {
  tidy(margins(x, data = x$data), conf.int = TRUE)
}

declare_estimator(
  Y ~ Z + X,
  model = glm,
  family = binomial("logit"),
  model_summary = tidy_margins,
  term = "Z"
)

```

9.4.3 Custom answer strategies

If your answer strategy does not use a `model` function, you'll need to provide a function that takes data as an input and returns a data frame with the estimate. Set the handler to be `label_estimator(your_function_name)` to take advantage of DeclareDesign's mechanism for matching inquiries to estimators. When you use `label_estimator`, you can provide an inquiry, and DeclareDesign will keep track of which estimates match each inquiry. (It simply adds a column to your tidy estimates data frame for the name of the estimator and the inquiry.) For example, to calculate the mean of an outcome, you could write your own estimator in this way:

```

my_estimator <- function(data) {
  data.frame(estimate = mean(data$Y))
}
declare_estimator(handler = label_estimator(my_estimator),
                  label = "mean",
                  inquiry = "Y_bar")

```

Often you may want to construct a test as part of your answer strategy that does not target an inquiry. Our `declare_test` function works just like `declare_estimator` except you need not include an inquiry. The `label_test` infrastructure works just like `label_estimator` for custom test functions.

Chapter 10

Diagnosis

Research design diagnosis is the process of evaluating the properties of a research design. We invent the term “diagnosand” to refer to those properties of a research design we would like to diagnose. Many diagnosands are familiar. Power is the probability of obtaining a statistically significant result. Bias is the average deviation of estimates from the estimand. Other diagnosands are more exotic, like the Type-S error rate, which is the probability the estimate has the incorrect sign, conditional on being statistically significant (Gelman and Carlin, 2014). This chapter focuses mainly on the use of Monte Carlo computer simulations to estimate diagnosands, though we touch on analytic design diagnosis as well.

Research designs are strong when the empirical answer a_d generated by the design is close to the true answer a_m . We can never know a_m with certainty – some combination of the fundamental problems of generalization, causal inference, and descriptive inference conspire to hide the truth from us. This problem means we have to assess the performance of our designs, not with respect to the real world, but instead with respect to our unverified and unverifiable beliefs about the world. In other words, we assess the properties of research designs by comparing the simulated answer a_d to the answers under the model a_m , over many possible realizations of the design.

Figure 10.1 below is similar to Figure 5.1, which we described when defining the components of a research design. To recapitulate the main points of that discussion: The theoretical half of an actual research design defines an inquiry I in terms of a model M . The true answer to that question is the inquiry applied to the real world $I(w) = a_m$. The empirical half of a research design applies the data strategy to the real world to generate a dataset ($D(m) = d$), then applies the answer strategy to the realized dataset to generate an answer: ($A(d) = a_d$). All these stars reflect that fact that *reality* plays an important role in empirical research designs. We commit to the notion that there are *real* answers to our

theoretical questions. Our empirical answers are tethered to reality because the data strategy generates information that depends on the real world. A fundamental premise of empirical research is that there is in fact a real truth to learn.

When we simulate and diagnose designs, however, this tether to reality is snipped, and we find ourselves in the bottom half of Figure 10.1. When we simulate, the theoretical half of the design entertains many possibilities in the model M , which we label m_1, m_2, \dots, m_k . The answers under each of m_1, m_2, \dots, m_k models are labeled a_{m_k} . Each of the k possibilities generates a different answer to the inquiry: $a_{m_1}, a_{m_2}, \dots, a_{m_k}$. Importantly, the simulated research design does not have access to the true answer a_m , only answers under the k considered models.

We pause to address one confusing aspect of our notation. The set of models m_1, m_2, \dots, m_k could refer to a set of k separate theoretical perspectives. Or it could refer to m_1, m_2, \dots, m_k draws from the same basic model, but the values of the exogenous variables (following the specific meaning described in Section 6.1.1) are slightly different. In other words, our notations doesn't draw a deep distinction between large differences between theoretical models and small ones.

When we simulate, the empirical half of the design is similarly dissociated from reality. We apply the data strategy D to each of the model draws m_k to produce simulated datasets index as d_k . These fabricated datasets may be similar to or different from the true dataset d that would result if the design were realized (the more similar the better). We apply the answer strategy A to each simulated dataset d_k in order to produce simulated answers a_{d_k} .

A comparison of the top row to the bottom three rows of Figure 10.1 shows how actual research designs differ from simulated research designs. Actual research designs are influenced by reality – we learn about the real world by conducting empirical research. We don't learn about the real world from simulated research designs. Instead, we learn about research designs themselves. We learn how the research design would behave if the real world m were like the model draw m_1 – or like m_{100} . We can only evaluate designs under the possibilities we consider in M . If the world m is not in that set of possibilities, we won't have evaluated the design under the most important setting, i.e., what will happen when we actually apply the design in reality. We want to follow the research design Principle 3.3 to “entertain many models” or in order words, to consider such a wide range of possibilities that some a_{m_k} will be close enough to a_m .

10.1 Diagnostic statistics and diagnosands

Diagnosands are summaries of the distributions of diagnostic statistics. We'll start by defining diagnostic statistics, then move on to describing how to generate the distribution of diagnostic statistics, then how to summarize those distri-

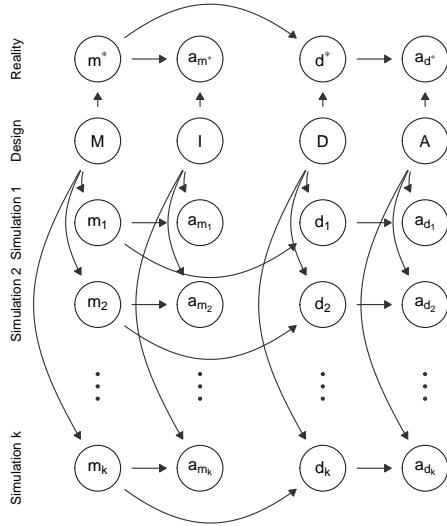


Figure 10.1: The *MIDA* framework

butions in order to estimate diagnosands.

A diagnostic statistic a_k is itself a function of a_{m_k} and a_{d_k} : $a_k = g(a_{m_k}, a_{d_k})$. We elide here two important notational annoyances. For the purpose of the following discussion of diagnostic statistics and diagnosands, a_{d_k} can refer to a estimate like an difference-in-means estimate or an associated uncertainty statistic like a standard error or a p -value. Importantly, diagnostic statistics can be functions that depend on a_{m_k} only, a_{d_k} only, or both together.

Each diagnostic statistic is the result of a different function g . For example, the “error” diagnostic statistic is produced by this function: $g(a_{m_k}, a_{d_k}) = a_{d_k} - a_{m_k}$. The “significant” diagnostic statistic is $g(a_{d_k}) = I[p(a_{d_k})]$, where $p()$ is a function of the empirical answer that returns a p -value, I is a significance cutoff, and I is an indicator function that returns 1 when the p -value is below the cutoff and 0 when it is above. A diagnostic statistic is something that can be calculated on the basis of a single run of the design.

Typically, diagnostic statistics will be different from simulation to simulation. In other words, a_1 will be different from a_2 , a_2 will be different from a_3 , and so on. These differences arise partially from the variation in M : m_1 is different from m_2 , m_2 is different from m_3 , and so on. Differences can also arise from the explicitly random procedures in D : sampling, assignment, and measurement can all include stochastic elements that will ramify through to the diagnostic statistics. As a result of these sources of variation, a diagnostic statistic is a random variable.

A diagnosand is a summary of the random variable $f()$, written $f()$, where $f()$ is a statistical functional that summarizes the random variable. A statistical functional is a function of a random variable that is not itself a random variable. For example, the expectation function $E[X]$ summarizes the random variable X with its expectation, the mean, while the variance function summarizes the expectation of the squared deviation of a random variable from its mean.

Let's back up a moment to work through two concrete examples of common diagnosands: bias and power (see Section 10.4 for a more exhaustive list).

Consider the diagnosand "bias" in the context of a two-arm randomized trial where the inquiry is the average treatment effect, the data strategy entails complete random assignment, and the answer strategy is the difference-in-means. Bias is the average difference between the estimand and the estimate. Under a single realization m_k of the model M , the value of the ATE will be a particular number, which we call a_{m_k} . We simulate a random assignment and measurement of observed outcomes, then apply the difference-in-means estimator. The diagnostic statistic is the error $a_{d_k} - a_{m_k}$; this error is a random variable because each m_k differs slightly. The bias diagnosand is the expectation of this random variable is $E[a_{d_k} - a_{m_k}]$, where the expectation is taken over the randomization distribution implied by M and D (or distinct regions of the randomization distribution).

Now consider the diagnosand "power." Like bias, statistical power is an expectation, this time of the "significant" diagnostic statistic $I(p(a_{d_k}) < 0.05)$. Power describes how frequently the answer strategy will return a statistically significant result. Some textbooks define statistical power as one minus the Type II error rate, where a Type II error is the failure to reject the null hypothesis, given that the null hypothesis is false. This definition is accurate, but hard to understand. The phrase "given that the null hypothesis is false" refers to model possibilities (a_{m_k} 's) in which the null hypothesis does not hold. Our definition of power is instead, "the probability of getting a statistically significant result, conditional on a set of beliefs about the model."

10.2 Estimating diagnosands analytically

Diagnosis can be done with analytic, pencil-and-paper methods. In an analytic design diagnosis, we typically derive a formula that returns the value of a diagnosand under a stipulated set of beliefs about the model, inquiry, data strategy, and answer strategy. For example, research design textbooks often contain analytic design diagnoses for statistical power. Gerber and Green (2012) write:

"To illustrate a power analysis, consider a completely randomized experiment where $N > 2$ of N units are selected into a binary treatment. The researcher must now make assumptions about the distributions of outcomes for treatment and for control units. In this example, the researcher assumes that the control group has a nor-

mally distributed outcome with mean c , the treatment group has a normally distributed outcome with mean t , and both group's outcomes have a standard deviation s . The researcher must also choose α , the desired level of statistical significance (typically 0.05). Under this scenario, there exists a simple asymptotic approximation for the power of the experiment (assuming that the significance test is two-tailed):

$$= \left(\frac{|t - c| N}{2} \right)^{-1} \left(1 - \frac{\alpha}{2} \right)$$

where β is the statistical power of the experiment, $\Phi(\cdot)$ is the normal cumulative distribution function (CDF), and $\Phi^{-1}(\cdot)$ is the inverse of the normal CDF."

This power formula makes **detailed** assumptions about M , I , D , and A . Under M , it assumes that both potential outcomes are normally distributed with group specific means and a common variance. Under I , it assumes the average treatment effect. Under D , it assumes a particular randomization strategy (simple random assignment). Under A , it assumes a particular hypothesis testing approach (equal variance t -test with $N - 2$ degrees of freedom). Whether this set of assumptions is "close enough" will depend on the research setting.

Analytic design diagnosis can be hugely useful, since they cover a large families of designs that meet the scope criteria. For example, the "standard error" diagnosand $E[(a_{d_k} - E[a_{d_k}])^2]$ of a standard two-arm trial has been worked out by statisticians to be $\frac{1}{n_1} \left\{ \frac{mV(Y_i(0))}{nm} + \frac{(Nm)V(Y_i(1))}{m} + 2Cov(Y_i(0), Y_i(1)) \right\}$ (see Section 17.1.1 for details). This standard error is accurate for any completely randomized design with stable potential outcomes and a difference-in-means estimator. Many, if not most, advances in our understanding of the properties of research design come from analytic design diagnosis. For example, Middleton (2008) and Imai, King and Nall (2009) show that cluster randomized trials with heterogeneous cluster sizes are not unbiased for the ATE, which leads to the design recommendation that clustered trials should block on cluster size. These lessons apply broadly, more broadly perhaps than the lessons learned about a specific design in a specific Monte Carlo simulation.

That said, scholars conducting analytic design diagnosis have only worked out a few diagnosands for a limited set of designs. Since designs are so heterogeneous and can vary on so many dimensions, computer simulation is often the only feasible way to diagnose. We learn a lot from analytic design diagnosis – what are the important parameters to consider, what are the important inferential problems – but they often cannot provide direct answers to practical questions like, how many subjects do I need for my conjoint experiment? For that reason, we turn to design diagnosis via Monte Carlo simulation.

Table 10.1: By-hand design diagnosis with a for-loop

power
0.15

10.3 Estimating diagnosands via simulation

Research design diagnosis by simulation occurs in two steps. First we simulate research designs repeatedly, collecting diagnostic statistics from each run of the simulation. Second, we summarize the distribution of the diagnostic statistics in order to estimate the diagnosands.

Monte Carlo simulations of research designs can be written in any programming language. To illustrate the most common way of simulating a design – a for loop – we’ll write a concise simulation in base R code. We conduct the simulation 500 times, each time, collecting the p -value associated with a regression estimate of the average effect of an outcome on a treatment.

Loops like this one can be implemented in any language and they remain a good tool for design diagnosis. We think, however, writing simulations in this way obscures what parts of the design refer to the model, the inquiry, the data strategy, or the answer strategy. We might imagine Z and Y are generated by a data strategy that randomly assigns Z , then measures Y (though without a language for linking code to design steps, it’s a bit unclear). The answer strategy involves running a regression of Y on Z , then conducting a hypothesis test against the null hypothesis that the coefficient on Z is 0. The model and inquiry are left entirely implicit. The inquiry might be the ATE, but it might also be a question of whether the ATE is equal to zero. The model might include only two potential outcomes for each subject, or it might have more, we don’t know.

```
sims <- 500
p.values <- rep(NA, sims)

for(i in 1:sims){
  Z <- rbinom(100, 1, 0.5)
  U <- rnorm(100)
  Y <- 0.2 * Z + U
  p.values[i] <- summary(lm(Y ~ Z))$coefficients[2, 4]
}

power <- mean(p.values <= 0.05)
power
```

Table 10.2: Design diagnosis using DeclareDesign

power	se(power)	n_sims
0.158	0.0154331	500

For this reason, we advocate for explicit description of all four research design components. Explicit design declaration can *also* occur in any programming language, but DeclareDesign is purpose-built for this task. We begin with the explicit declaration of the same two-arm trial as above. We have 100 subjects with a constant response to treatment of 0.2 units. Our inquiry is the average difference between the treated and untreated potential outcomes – the ATE. We assign treatment using complete random assignment and estimate treatment effects using difference-in-means.

Declaration 10.1.

```
design <-
  declare_model(
    N = 100,
    U = rnorm(N),
    potential_outcomes(Y ~ 0.2 * Z + U)
  ) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, inquiry = "ATE")
```

We can now diagnose the declared design. For comparison with the loop, we calculate just one diagnosand: statistical power. Both approaches return the same answer (within simulation error) of 16% statistical power.

```
diagnosands <- declare_diagnosands(power = mean(p.value <= 0.05))

diagnosis <-
  diagnose_design(design,
                  diagnosands = diagnosands,
                  sims = 500)
diagnosis
```

Table 10.3: One simulation draw

estimand	estimate	std.error	df	p.value	conf.low	conf.high
0.2	0.448	0.171	98	0.01	0.108	0.789

10.3.1 Breaking down diagnosis

In this section, we break down the diagnosis process from start to finish. We build up from a single simulation run of the design to the distribution of simulations, to summaries of that distribution.

We can run this simulation a single time with `run_design`:

```
run_design(design)
```

Figure 10.2 shows the information we might obtain from a single run of the simulation. The filled point is the estimate a_{d_k} . The open triangle is the estimand a_{m_k} . The bell-shaped curve is our normal-approximation based estimate of the sampling distribution. The standard deviation of this estimated distribution is our estimated standard error, which expresses our uncertainty. The confidence interval around the estimate is another expression of our uncertainty: We’re not sure where a_{d_k} is, but if things are going according to plan, confidence intervals constructed this way will bracket a_{m_k} 95% of the time.

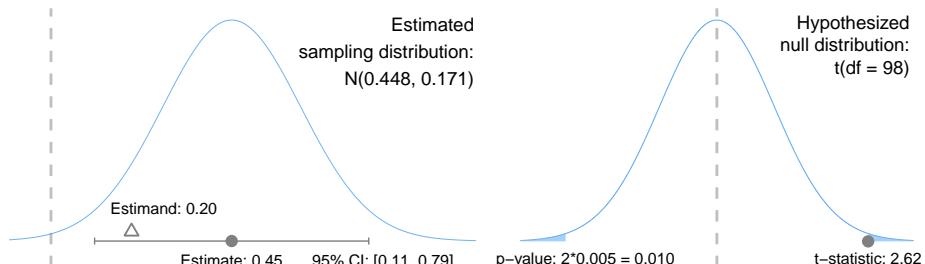


Figure 10.2: Visualization of one draw from a design diagnosis.

From this single draw, we can’t yet estimate diagnosands, but we can calculate diagnostic statistics. The estimate was higher than the estimand in this draw, so the error is $0.45 - 0.20 = 0.25$. Likewise, the squared error is $(0.45 - 0.20)^2 = 0.0625$. The p -value is 0.01, which is well below the threshold of 0.05, so “statistical significance” diagnostic statistic is equal to TRUE. The confidence interval stretches from 0.11 to 0.79, and since the value of the estimand (0.20) is between those bounds, the “covers” diagnostic statistic is equal to TRUE as well.

To calculate the distributions of the diagnostic statistics, we have to simulate designs not just once, but many times over. The bias diagnosand is the average error over many runs of the simulation. The statistical power diagnosand is the fraction of runs in which the estimate is significant. The coverage diagnosand is the frequency with which the confidence interval covers the estimand.

Figure 10.3 visualizes 10 runs of the simulation. We can see that some of the draws produce statistically significant estimates (the shaded areas are small and the confidence intervals don't overlap zero), but not all. We get a sense of the *true* standard error by seeing how the point estimates bounce around. We get a feel for the difference between the estimates of the standard error and true standard error. Design diagnosis is the process of learning about many ways the study might come out, not just the one way that it will.

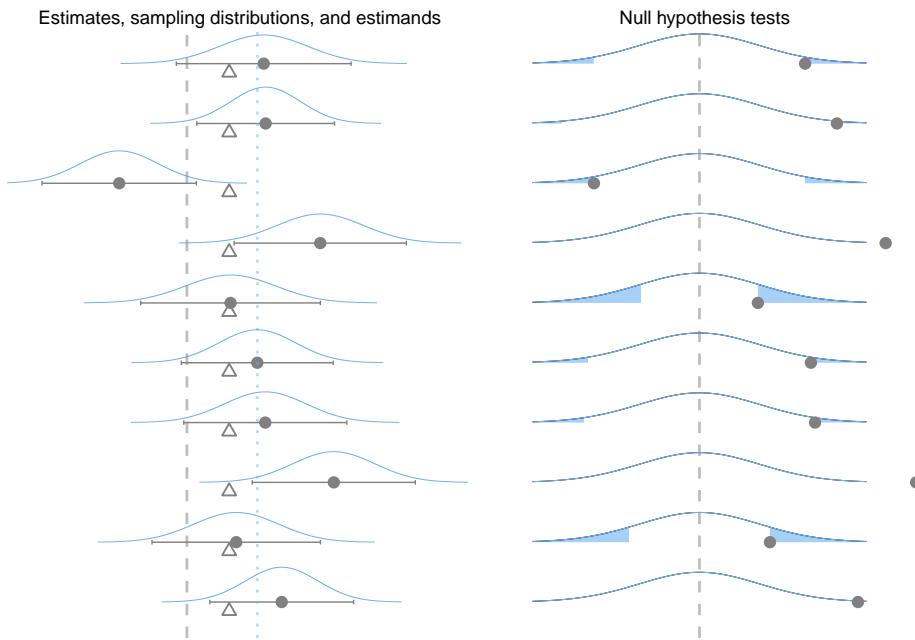


Figure 10.3: Visualization of ten draws from a design diagnosis.

This line of code does it all in one. We simulate the design 1000 times to calculate the diagnostic statistics, then we summarize them in terms of bias, the true standard error, RMSE, power, and coverage.

```
diagnosis <-
  diagnose_design(
    design, sims = 1000,
    diagnosands = declare_diagnosands()
```

Table 10.4: Diagnosand estimates with bootstrapped standard errors

bias	true se	power	coverage
0.00 (0.01)	0.20 (0.00)	0.17 (0.01)	0.94 (0.01)

```
bias = mean(estimate - estimand),  
true_se = sd(estimate),  
power = mean(p.value <= 0.05),  
coverage = mean(estimand <= conf.high & estimand >= conf.low)  
)  
)
```

10.3.2 Simulation error

We use Monte Carlo simulation to estimate diagnosands. We only get estimates – not the exact value – of diagnosands under a particular model because of simulation error. Simulation error declines as we conduct more simulations. When we conduct many thousands of simulations, we’re relatively certain of the value of the diagnosand under the model. If we conduct just tens of simulations, we are much more uncertain.

We can characterize our uncertainty attending to the diagnosand estimate by calculating a standard error. If the Monte Carlo standard error is large relative to the estimated diagnosand, then we need to increase the number of simulations. Unlike empirical settings where additional data collection can be very expensive, in order to get more observations of diagnostic statistics, we can just increase the number of simulations. Computation time isn't free – but it's cheap.

The standard error of diagnosand estimates can be calculated using standard formulas. If the diagnosand can be written as a population mean, and the simulations are fully independent, then we can estimate the standard error as $\frac{\text{sd}()}{\sqrt{k}}$, where k is the number of simulations. The power diagnosand summarizes a binary diagnostic statistic (is the estimate significant or not). Binary variables exhibit maximum variability when the probability of a 1 is 0.5. So with 1,000 independent simulations, the standard error for the mean of a binary diagnostic statistic is $\frac{0.5 \cdot 0.5}{\sqrt{1000}} = 0.016$. This level of simulation uncertainty is often acceptable (the 95% confidence interval is approximately $4 * 0.016 = 6.4$ percentage points wide), but if it isn't, you can always increase the number of simulations.

Since some diagnosands are more complex than a population mean (i.e., we can't characterize the estimation error with simple formulas), so `DeclareDesign` does nonparametric bootstrapping by default whenever the

`diagnose_design()` function is called.

10.4 Types of diagnosands

As described above, a diagnostic statistic is any summary function of a_m and a_d , and a diagnosand is any summary of the distribution of diagnostic statistics. As a result, there are a great many diagnosands researchers may consider. In Table 10.5, we introduce a nonexhaustive set of diagnostic statistics, and in Table 10.6 a nonexhaustive set of diagnosands.

Table 10.5: Diagnostic statistics.

Diagnostic statistic	Definition
Estimate	a_{d_k}
Estimand under the model	a_{m_k}
p -value	$p(a_{d_k})$
p -value is no greater than	$I(p^-)$
Confidence interval	CI_1
Confidence interval covers the estimand under the model	$\text{covers}_{a_m} I\{a_m \in CI_1\}$
Estimated standard error	(A)
Cost	cost
Proportion of subjects harmed	$\Pr(\text{harm}) = \frac{1}{n} \sum_i \text{harm}_i$

Table 10.6: Diagnosands.

Diagnosand	Description	Definition
Average estimate		$E(a_d)$
Average estimand		$E(a_m)$
Power	Probability of rejecting null hypothesis of no effect	$E(I(p^-))$
Bias	Expected difference between estimate and estimand	$E(a_d - a_m)$
Variance of the estimates		$V(a_d)$
True standard error		$V(a_d)$

Diagnosand	Description	Definition
Average estimated standard error		(A)
Root mean-squared-error (RMSE)		$E(a_d \ a_m)$
Coverage	Probability confidence interval overlaps estimand	$\Pr(\text{covers}_{a_m})$
Biaseliminated coverage	Probability confidence interval overlaps average estimate (Morris, White and Crowther (2019))	$\Pr(\text{covers}_{a_d})$
Type-S error rate	Probability estimate has an incorrect sign, if statistically significant (Gelman and Carlin, 2014)	$\Pr(\text{sgn}(a_d) \neq \text{sgn}(a_m) \mid p)$
Exaggeration ratio	Expected ratio of absolute value of estimate to estimand, if statistically significant (Gelman and Carlin, 2014)	$E(a_d / a_m \mid p)$
Type I error	Rejecting the null hypothesis when it is true	$\Pr(p \mid a_m = a^0)$
Type II error	Failure to reject the null hypothesis when it is false	$\Pr(p \mid a_m = a^0)$
Sampling bias	Expected difference between population average treatment effect and sample average treatment effect (Imai, King and Stuart, 2008)	$E(a_{m_{\text{sample}}} - a_{m_{\text{population}}})$
Expected maximum cost	Maximum cost across possible realizations of the study	max cost
Bayesian learning	Difference between prior and posterior guess of the value of the estimand	$a_{m_{\text{post}}} - a_{m_{\text{pre}}}$
Value for money	Probability that a decision based on estimated effect yields net benefits	
Success	Qualitative assessment of the success of a study	
Minimum detectable effect (MDE)	Smallest effect size for which the power of the design is nominal (e.g., powered at 80%)	$\text{argmin}_{a_m} \Pr(p \mid a_m) = 0.8$
Robustness	Joint probability of rejecting the null hypothesis across multiple tests	

Diagnosand	Description	Definition
Maximum proportion of subjects harmed		$\max \Pr(\text{harm})$

10.5 Diagnosand completeness

A design declaration is “diagnosand-complete” when the declaration contains enough information to calculate the diagnosand. The “bias” diagnosand requires that all four of M , I , D , and A are specified in sufficient detail because we need to be able to compare the answer under the model (a_m) to the simulated empirical answer (a_d). The “statistical power” diagnosand often does not require the inquiry I to be specified, since we can mechanistically conduct null hypothesis significance tests without reference to estimands. Neither the “bias” nor “statistical power” diagnosands require any details of confidence interval construction, but without those details, the declaration is not diagnosand-complete for “coverage.”

Every design we have ever declared is diagnosand-complete for some diagnosands but not for others. Attaining diagnosand-completeness for each of the diagnosands in Table 10.6 in a single design declaration would be challenging and cumbersome, though of course not impossible. Total diagnosand-completeness is not even a goal for design declaration. We want to be diagnosand-complete for the set of diagnosands that matter most in a particular research setting, which is the topic of the next section.

10.6 Choosing diagnosands to explore design tradeoffs

Principle 3.8 says we should consider all important diagnosands – but not all diagnosands are important for every study. For example, in a descriptive study whose goal is to estimate the fraction of people in France who are left-handed, statistical power is irrelevant. A hypothesis test against the null hypothesis that zero percent of the people in France are left-handed is preposterous. We know for sure that the fraction is not zero, we just don’t know its precise value. A much more important diagnosand for this study would be RMSE (root-mean-squared-error), which is a measure of how well-estimated the estimand is that incorporates both bias and variance.

Often, we need to look at several diagnosands in order to understand what might be going wrong. This If your design exhibits “undercoverage” (e.g., a procedure for constructing “95%” confidence interval only covers the estimand 50% of the time), that might be because your standard errors are too small

or because your point estimator is biased, or some combination of the two. In really perverse instances, you might have a biased point estimator which, thanks to overly-wide confidence intervals, just happens to cover 95% of the time.

Many research design decisions involve trading off bias and variance. In trade-off settings, we may need to accept higher variance in order to decrease bias. Likewise, we may need to accept a bit of bias in order to achieve lower variance. The tradeoff is captured by mean-squared error, which is the average squared distance between a_d and a_m . Of course, we would ideally like to have as low a mean-squared error as possible. We would like to achieve low variance and low bias simultaneously.

To illustrate, consider the following three designs as represented by three targets. The inquiry is the bullseye of the target. The data and answer strategies combine to generate a process by which arrows are shot towards the target. On the left, we have a very bad archer: even though the estimates are unbiased in the sense that they hit the bullseye “on average”, very few of the arrows are on target. In the middle, we have an excellent shot: they are both on target and low variance. On the right, we have an archer who is very consistent (low variance) but biased. The mean squared error is highest on the left and lowest in the middle.

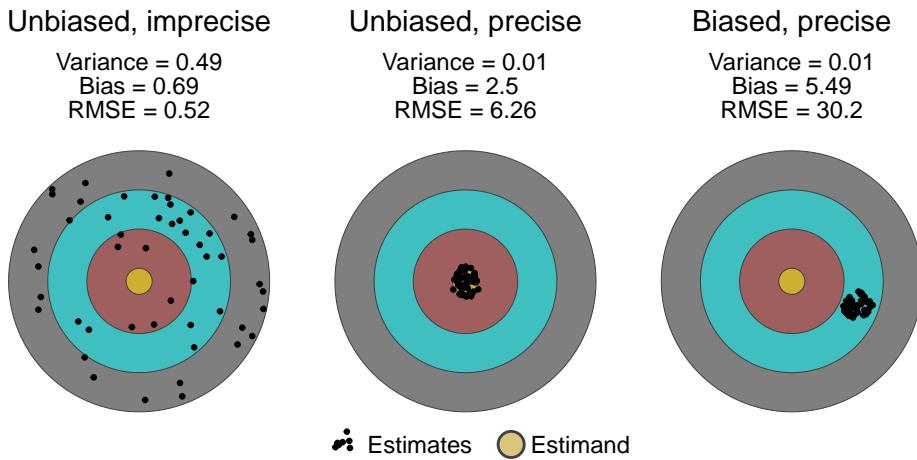


Figure 10.4: Visualization of the bias and variance of three ‘estimators’ of the bullseye

The archery metaphor is common in research design textbooks because it effectively conveys the difference between variance and bias, but it does elide an important point. It really matters **which target** your archer is shooting at. Figure 10.5 shows a bizarre double-target representing two inquiries. The empirical strategy is unbiased and precise for the left inquiry, but it is clearly biased for the right inquiry. When we are describing the properties of an answer strategy, we have to be clear about which inquiry it is associated with.

Diagnosands depend on the estimand

Variance = 0.007 (left), 0.007 (right)

Bias = 0.08 (left), 0.24 (right)

RMSE = 0.01 (left), 0.06 (right)

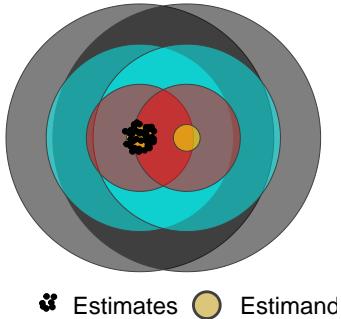


Figure 10.5: The bias, variance, and RMSE of an answer strategy depend on the inquiry.

MSE is an exactly equal weighting of variance and bias (squared). Yet many other weightings of these two diagnosands are possible, and different researchers will vary in their weightings.

In evaluating a research design diagnosis, what understand the weighting of all relevant diagnosands. We can think of this as our research design utility function. Our utility function describes how important it is to study big questions, to shift beliefs in a research field, to overturn established findings, to obtain unbiased answers, and to get the sign of the inquiry right. Your utility function evaluated for a given design will yield a utility and these can be compared across empirical designs (we call this process of comparison redesign, described in detail in the next section).

We often consider the diagnosand power on its own. This diagnosand is the probability of getting a statistically significant result, which of course depends on many things about your design including, crucially, the unknown magnitude of the parameter to be estimated. You can think of statistical power as the probability of a success, where success is defined as getting significant results. The conventional power target is 80% power. One could imagine redefining statistical power as “null risk,” or the probability of obtaining a null result. In these terms, the conventional power target entails a 20% null risk, or a one in five chance of “failure.” Those odds aren’t great, so we recommend designing studies with lower null risk. But considering power alone is also misleading: no researcher wants to design a study that is 80% powered but returns highly biased estimates. Another way of saying this is that researchers always care about both power and bias. How much they care about each feature determines the weight of power and bias in their utility function.

Diagnosands need not be about hypothesis testing or even statistical analysis of the data at all. We often trade off how much we learn from a research design with its cost in terms of money and our time. We have financial and time budgets that provide hard constraints to our designs, but we also at the margin many researchers wish to select cheaper (or shorter) designs in order to carry out more studies or finish their degree sooner. Time and cost are also diagnostic statistics! We may wish to explore the maximum cost of a study or the maximum amount of time it would take.

Ethical considerations also often enter the process of assessing research designs, if implicitly. We can explicitly incorporate them into our utility function by valuing minimizing harm and maximizing the degree of informed consent requested of subjects. When collecting, researchers often believe that they face a tradeoff between informing subjects about the subject of the data collection (an ethical consideration, or a requirement of the IRB) on the one hand and the bias that comes from Hawthorne or demand effects. We can incorporate these considerations in a research design diagnosis by specifying diagnostic statistics related to the amount of disclosure about the purposes of research or the number of subjects harmed in the research.

10.7 Diagnosis under model uncertainty

We are always uncertain about the model in M (Principle 3.3). If we were certain of M (or there was no real dispute about it), there would be no need to conduct new empirical research. Research design diagnosis can incorporate this uncertainty by evaluating the performance of the design under alternative models. For example, if we are unsure of the exact value of the intra-class correlation (ICC), we can simulate the design under a range of plausible ICC values. If we are unsure of the true average treatment effect, we can diagnose the power of the study over a range of plausible effect sizes. Uncertainty over model inputs like the means, variances, and covariances in data that will eventually be collected is a major reason to simulate under a range of plausible values.

We illustrate diagnosis under model uncertainty with the declaration below. Here we have a 200 unit two-arm trial in which we explicitly describe our uncertainty over the value of the true average treatment effect. In the `potential_outcomes` call, we have $Y \sim \text{runif}(1, 0, 0.5) * Z + U$ which indicates that the treatment effect in each run of the simulation is one draw from uniform distribution between 0.0 and 0.5.

Declaration 10.2.

```
design <-
  declare_model(
```

```

N = 200, U = rnorm(N),
# this runif(1, 0, 0.5) generates 1 random ATE between 0 and 0.5
potential_outcomes(Y ~ runif(1, 0, 0.5) * Z + U)) +
declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
declare_assignment(Z = complete_ra(N, prob = 0.5)) +
declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
declare_estimator(Y ~ Z, inquiry = "ATE")

```

Figure 10.6 shows how the statistical power of this design varies over the set of model possibilities. We plot the possible effect sizes we entertain in the model on the horizontal axis and the statistical power on the vertical axis. We also plot a loess curve flexibly smooths over the full distribution of effect sizes. We see that at low levels of the effect size, statistical power is quite poor. With only 200 subjects, we couldn't achieve 80% statistical power unless the true effect size were approximately 0.45 standard deviations. The effect size at which a design achieves 80% power is often referred to as the minimum detectable effect size (MDE). This exercise shows how the MDE diagnosand is a summary of a design that admits uncertainty over the effect size. We return to this example in Section 11.4, when we redesign this study to learn how the MDE changes at different sample sizes.

What is “the” power of this design? Under one view, the true power of the design is the whichever value for power is associated with the effect size. But under an agnostic view, the ex ante “power” of the design is a weighted average of all these power values, weighted by the researcher’s prior beliefs over the distribution of possible effect sizes.

10.7.1 Adjudicating between competing models

Principle 3.3 says we should entertain many models. The principle extends to *competing* models. Imagine that you believe M_1 is true but that your scholarly rival believes M_2 . Suppose that under M_1 , the treatment affects Y_1 but not Y_2 . Your rival posits the reverse: Under M_2 : the treatment affects Y_2 but not Y_1 . In the spirit of scientific progress, the both of you engage in an “adversarial collaboration.” You design a study together. The design you choose should, first, demonstrate M_1 is true if it is true and, second, demonstrate M_2 is true if it is true. In order to come to an agreement about the properties of the design, you will need to simulate the design under both models.

Declaration 10.3.

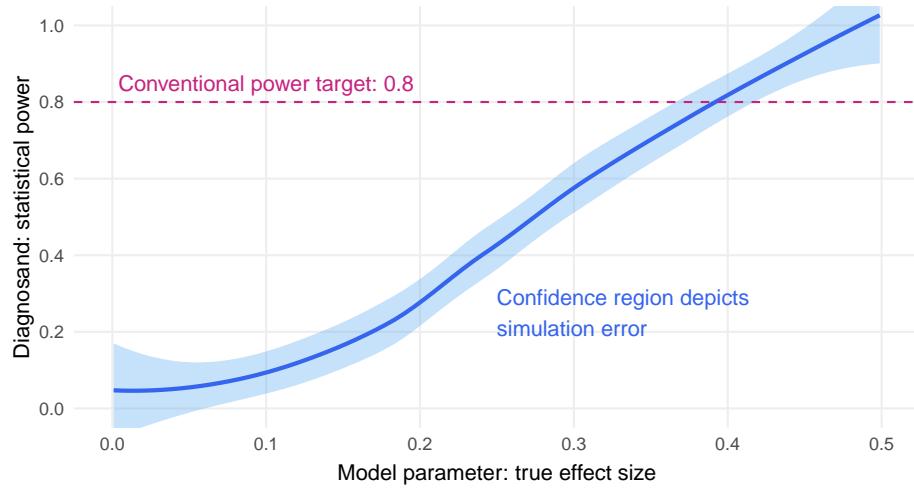


Figure 10.6: Diagnosing an experiment over uncertainty about the true effect size

```

M1 <-
  declare_model(
    N = 200,
    U = rnorm(N),
    potential_outcomes(Y1 ~ 0.2 * Z + U),
    potential_outcomes(Y2 ~ 0.0 * Z + U)
  )

M2 <-
  declare_model(
    N = 200,
    U = rnorm(N),
    potential_outcomes(Y1 ~ 0.0 * Z + U),
    potential_outcomes(Y2 ~ 0.2 * Z + U)
  )

IDA <-
  declare_inquiry(ATE1 = mean(Y1_Z_1 - Y1_Z_0),
                 ATE2 = mean(Y2_Z_1 - Y2_Z_0)) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y1 = reveal_outcomes(Y1 ~ Z),
                      Y2 = reveal_outcomes(Y2 ~ Z)) +
  declare_estimator(Y1 ~ Z, inquiry = "ATE1", label = "DIM1") +

```

Table 10.7: Design diagnosis under two alternative theories

design	Only theory 1 supported	Only theory 2 supported	Both supported	Neither supported
design1	0.26	0.030	0.028	0.682
design2	0.03	0.262	0.028	0.680

```
declare_estimator(Y2 ~ Z, inquiry = "ATE2", label = "DIM2")

design1 <- M1 + IDA
design2 <- M2 + IDA
```

We simulate the design, then count up how frequently each perspective receives support. If we define support by statistical significant (other metrics are of course possible), then you are correct if the effect of treatment on Y_1 is significant but the effect on Y_2 is nonsignificant. If the reverse pattern is obtained, your rival can claim victory. Two kinds of split decisions are possible: neither estimate is significant, or both are. By simulation, we can estimate the rates of these four possibilities, both under your beliefs and your rivals.

The diagnosis shows that the study is responsive to the truth. When theory 1 is correct, the design is more likely to yield empirical evidence in favor of it; the reverse holds when theory 2 is correct. That said, the major concern facing this adversarial collaboration is that the study is too small to resolve the dispute. About two-thirds of the time – regardless of who is right! – neither theory receives support. This problem can be ameliorated either by elaborating more tests of each theoretical perspective or by increasing the size of the study.

10.8 Diagnosing a design in code

Once a design is declared in code, diagnosing it is usually the easy part. `diagnose_design` handles all the details and bookkeeping for you. In this section, we outline how to simulate the design and learn about the properties of it by hand. We don't provide built-in plotting features in `DeclareDesign` because every design diagnosis is a little bit different. But plotting simulations is a great way to get acquainted with the design and to explore design variations.

We declare a two-arm randomized experiment as an example to simulate, and simulate the design 100 times.

Declaration 10.4.

```

effect_size <- 0.1
design <-
  declare_model(
    N = 100,
    U = rnorm(N),
    X = rnorm(N),
    potential_outcomes(Y ~ effect_size * Z + X + U)
  ) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, inquiry = "ATE", label = "unadjusted") +
  declare_estimator(Y ~ Z + X, inquiry = "ATE", label = "adjusted")

```

```
simulations_df <- simulate_design(design, sims = 100)
```

You could diagnose your design using dplyr tools on your own. We provide a basic pipeline that includes all of the default diagnosands below. You can modify them, add or subtract, and group your data more flexibly with the simulations data in hand.

```

simulations_df %>%
  group_by(design, inquiry, estimator, term) %>%
  summarize(
    bias = mean(estimate - estimand),
    rmse = sqrt(mean((estimate - estimand)^2)),
    power = mean(p.value < 0.05),
    coverage = mean(estimand <= conf.high & estimand >= conf.low),
    mean_estimate = mean(estimate),
    sd_estimate = sd(estimate),
    mean_se = mean(std.error),
    type_s_rate =
      mean((sign(estimate) != sign(estimand))[p.value < 0.05]),
    mean_estimand = mean(estimand),
    .groups = "drop"
  )

```

Many diagnosands are summaries of the sampling distribution of the estimates.

To get a deeper sense of what estimates look like, we can plot them using a histogram. In this design, we have two estimators, so we create two facets one for each estimator's sampling distribution using `facet_wrap`. We display the sampling distribution and overlay that with the value of the estimand in each case (which is the same!) so that you can get a sense for whether the sampling distribution is centered on the estimand or not — meaning it is biased. The width of the sampling distribution indicates the precision of the estimates.

```
# first create summary for vertical lines
summary_df <-
  simulations_df %>%
  group_by(estimator) %>%
  summarize(estimand = mean(estimand))

# then plot simulations
ggplot(simulations_df) +
  geom_histogram(aes(estimate),
                 bins = 40, fill = "lightblue") +
  geom_vline(data = summary_df,
             aes(xintercept = estimand),
             lty = "dashed", color = "red") +
  annotate("text", y = 80, x = 0, label = "Estimand",
           color = "red", hjust = 1) +
  facet_wrap(~ estimator) +
  labs(x = "Estimate", y = "Count of simulations")
```

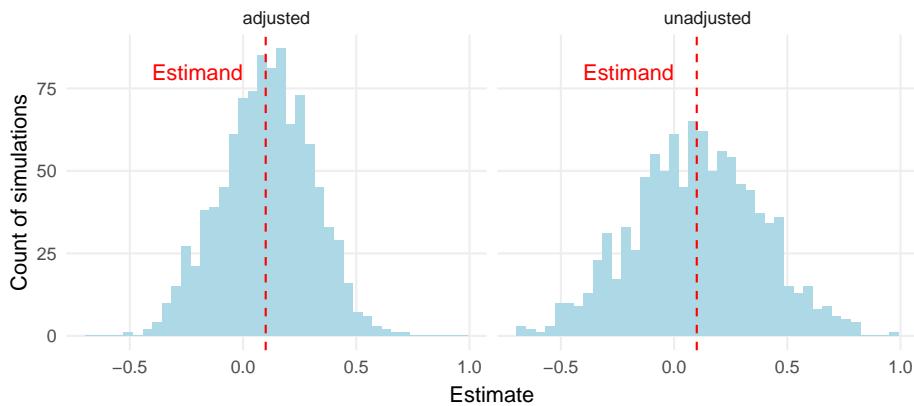


Figure 10.7: Example visualization of a diagnosis

Diagnosing over model uncertainty is a crucial part of diagnosis. We want to understand when our design performs well and when it does not. A classical

example of this in wide practice is the power curve. In a power curve, we display the power of a design (the probability of achieving statistical significance) along different possible effect sizes.

```
design <-
  declare_model(
    N = 200,
    U = rnorm(N),
    potential_outcomes(Y ~ runif(1, 0.0, 0.5) * Z + U)
  ) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, inquiry = "ATE")

simulations_df <-
  simulate_designs(design, sims = 500) %>%
  mutate(significant = if_else(p.value <= 0.05, 1, 0))

ggplot(simulations_df) +
  stat_smooth(aes(estimand, significant), method = 'loess', color = "blue", fill = "lightblue") +
  geom_hline(yintercept = 0.8, color = "red", linetype = "dashed") +
  annotate("text", x = 0, y = 0.85, label = "Conventional power threshold = 0.8", hjust = 0) +
  scale_y_continuous(breaks = seq(0, 1, 0.2)) +
  coord_cartesian(ylim = c(0, 1)) +
  theme(legend.position = "none") +
  labs(x = "Model parameter: true effect size",
       y = "Diagnosand: statistical power")
```

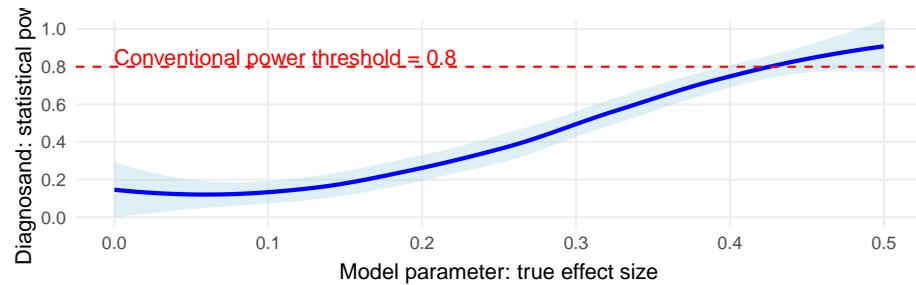


Figure 10.8: Example visualization of a diagnosis

10.9 Summary

Diagnosis is the process of estimating the properties of research designs under uncertain beliefs about the world. We simulate under many alternative designs because we want to choose designs that are strong under a large range of model possibilities. Each run of a simulate generates a diagnostic statistic. We learn about the distribution of the diagnositc statistic by running the simulation repeatedly. Diagnosands are summaries of the distribution of diagnostic statistics. Which diagnosands are most important will vary from study to study, and depend on the relative weight you place on one diagnosand versus another. Analytic design diagnosis is possible and can be quite powerful – we nevertheless recommend full simulation of research designs in order to learn about a range of diagnosands.

Further reading

- Gelman and Carlin (2014) on Type M and Type S errors
- Herron and Quinn (2016) on case selection and sampling bias
- Baumgartner and Thiem (2017) and Rohlfing (2018) on diagnosands in qualitative research
- Rubin (1984) on diagnosands in Bayesian research
- Gelman, Hill and Vehtari (2020) Chapter 5 describes fake data simulation, Chapter 7 describes how to use fake data simulation to check model fitting procedures, and Chapter 16 describes simulations of whole designs in much the same way as we describe design diagnosis.

176

Diagnosis

10.9

Chapter 11

Redesign

Redesign is the process of choosing the single empirical design you will implement from a very large family of possible designs. To make this choice, you systematically vary aspects of the data and answer strategies to understand their impact on the most important diagnosands. Redesign entails diagnosing many possible empirical designs over the range of plausible theoretical models, and comparing them.

A sample size calculation is the prototypical example of a redesign. Holding the model, inquiry, and answer strategy constant, we vary the “sample size” feature of the data strategy in order to understand how a diagnosand like the width of the confidence interval changes as we change N .

Not surprisingly, most designs get stronger as we allocate more resources to them. The expected width of a confidence interval could always be tighter, if only we had more subjects. Standard errors could always be smaller, if only we took more pre-treatment measurements. At some point, though, the gains are not worth the increased costs, so we settle for an affordable design that meets our scientific goals well enough. (Of course, if the largest affordable design has poor properties, no version of the study is worth implementing). The knowledge-expense tradeoff is a problem that every empirical study faces. The purpose of redesign is to explore this and other tradeoffs in a systematic way.

11.1 Power curve example

A power curve is a common tool for redesign. We want to learn the power of a test at many sample sizes, either so we can learn the price of precision, or so we can learn what sample size is required for a minimum level of statistical power.

We start with a minimal design declaration: we draw samples of size N and

measure a single binary outcome Y , then conduct a test against the null hypothesis that the true proportion of successes is equal to 0.5.

Declaration 11.1. Power curve design

```
design <-  
  declare_model(N = N) +  
  declare_measurement(Y = rbinom(n = N, size = 1, prob = 0.55)) +  
  declare_test(  
    handler =  
      function(data) {  
        test <- prop.test(x = table(data$Y), p = 0.5)  
        tidy(test)  
      }  
  )
```

To construct a power curve, we redesign over values of N that vary from 100 to 1000.

```
diagnosis <-  
  design %>%  
  redesign(N = seq(100, 1000, 100)) %>%  
  diagnose_designs()
```

Redesigns are often easiest to understand graphically, as in Figure 11.1. At each sample size, we learn the associated level of statistical power. We might then choose the least expensive design (sample size 800) that meets a minimum power standard (0.8).

11.2 Redesign over multiple design parameters

Sometimes, we have a fixed budget (in terms of financial resources, creative effort, or time), so the redesign question isn't about how much to spend, but how to spend it across competing demands. For example, we might want to find the sample size N and the fraction of units to be treated prob that minimize a design's error subject to a fixed budget. Data collection costs \$2 per unit and treatment costs \$20 per treated unit. We need to choose how many subjects to sample and how many to treat. We might rather add an extra 11 units to the control units (additional cost $\$2 * 11 = \22) than add one extra unit to the treatment group (additional cost $\$2 + \$20 = \$22$).

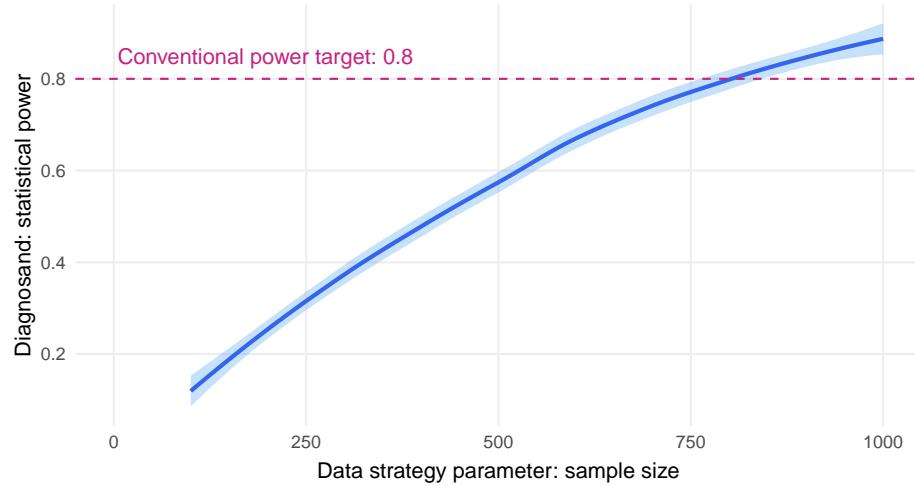


Figure 11.1: Redesigning a sample survey with respect to power

We solve the optimization problem:

$$\begin{aligned} \operatorname{argmin}_{N, N_t} \quad & E_M(L(a^d - a^m | D_{N,m})) \\ \text{s.t.} \quad & 5N + 20m \leq 5000 \end{aligned}$$

where L is a loss function, increasing in the difference between a^d and a^m .

We can explore this optimization with bare-bones declaration of a two-arm trial that depends on two separate data strategy parameters, N and prob :

Declaration 11.2. Bare-bones two arm trial

```
design <-  
  declare_model(N = N, U = rnorm(N),  
                potential_outcomes(Y ~ 0.2 * Z + U)) +  
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +  
  declare_assignment(Z = complete_ra(N = N, prob = prob)) +  
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +  
  declare_estimator(Y ~ Z, inquiry = "ATE")
```

We redesign, varying those two parameters over reasonable ranges: 100 to 1000 subjects, with probabilities of assignment from 0.1 to 0.5. The redesign function smartly generates designs with all combinations of the two parameters. We

want to consider the consequences of these data strategy choices for two diagnosands: cost and a very common loss function: mean squared error.

```
diagnosands <-
  declare_diagnosands(cost = unique(N * 2 + prob * N * 20),
                       rmse = sqrt(mean((estimate - estimand) ^ 2)))

diagnosis <-
  design %>%
  redesign(N = seq(100, 1000, 50),
           prob = seq(0.1, 0.5, 0.2)) %>%
  diagnose_designs(diagnosands = diagnosands)
```

The diagnosis is represented in Figure 11.2. The top panel shows the cost of empirical designs, at three probabilities of assignment over many sample sizes. The bottom panel shows the RMSE of each design. According to this diagnosis, the best combination that can be achieved for less than \$5,000 is $N = 600$ with $\text{prob} = 0.3$. This conclusion is in mild tension with common the design advice that under many circumstances, balanced designs are preferable (see Section 17.1.1 in the design library for an in-depth discussion of this point). Here, untreated subjects are so much less expensive than treated subjects, we want to tilt the design towards having a larger control group. How far to tilt depends on model beliefs as well as the cost structure of the study.

11.3 Redesign over answer strategies

Redesign can also take place over possible answer strategies. An inquiry like the average treatment effect could be estimated using many different estimators: difference-in-means, logistic regression, covariate-adjusted ordinary least squares, the stratified estimator, doubly robust regression, targeted maximum likelihood regression, regression trees – the list of possibilities is long. Redesign is an opportunity to explore how many alternative analysis approaches work.

A key tradeoff in the choice of answer strategy is the bias-variance tradeoff. Some answer strategies exhibit higher bias but lower variance while others have lower bias but higher variance. Choosing which side of the bias-variance trade-off to take is complicated and the process for choosing among alternatives must be motivated by the scientific goals at hand.

A common heuristic for trading off bias and variance is the mean squared error (MSE) diagnosand. Mean squared error is equal to the square of bias plus variance, which is to say MSE weighs bias and variance equally. Typically, researchers choose among alternative answer strategies by minimizing MSE. If in your scientific context, bias is more important than variance, you might choose

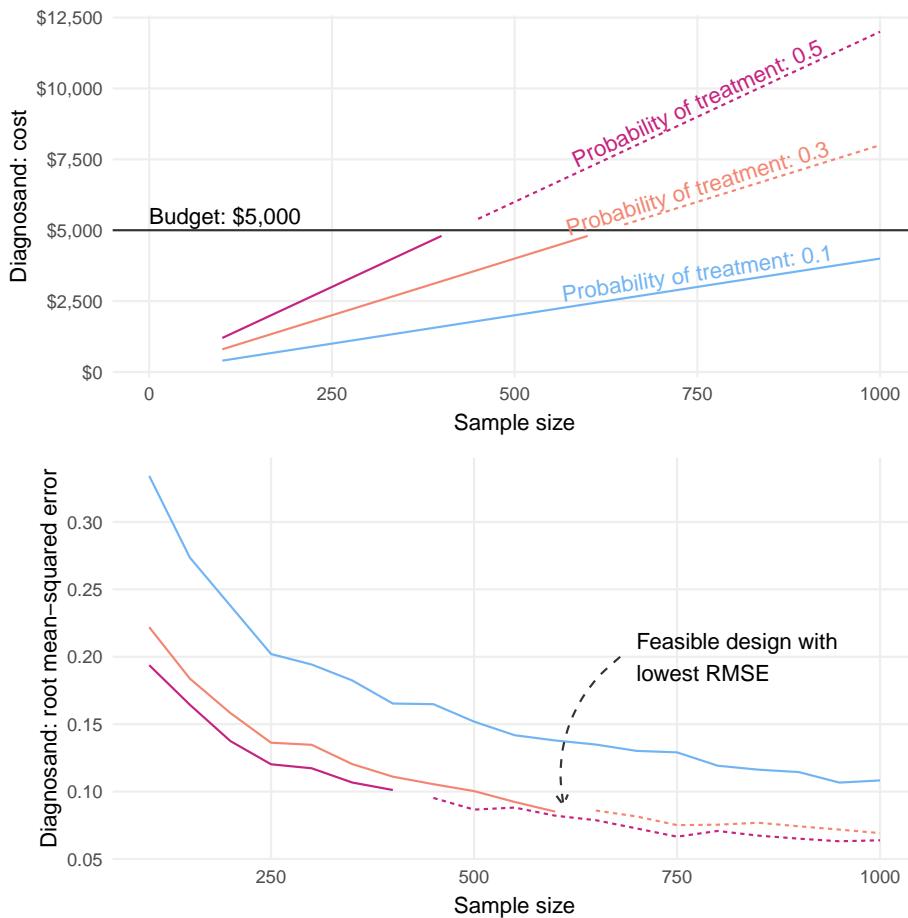


Figure 11.2: Redesigning a sample survey with respect to power

a an answer strategy that accepts slightly more variance in exchange for a decrease in bias.

To illustrate the bias-variance tradeoff, we consider a setting in which the goal is to estimate the conditional expectation of some outcome variable Y with respect to a covariate X . The design declaration below depends on three user-defined functions (`dip`, `cef_inquiry`, and `cef_estimator`) that we have hidden so as not to clog the narrative flow of the section.

Declaration 11.3. Conditional expectation function design

```
design <-
```

```

declare_model(
  N = 100,
  X = runif(N, 0, 3)) +
declare_inquiry(handler = cef_inquiry) +
declare_measurement(Y = dip(X) + rnorm(N, 0, .5)) +
declare_estimator(handler = cef_estimator)

```

Figure 11.3 shows one draw of this design – the predictions of the CEF made by nine regressions of increasing flexibility. A polynomial of order 1 is just a straight line, a polynomial of order 2 is a quadratic, order 3 is a cubic etc. Aronow and Miller (2019) show (Theorem 4.3.3) that even nonlinear CEFs can be approximated to up to an arbitrary level of precision by increasing the order of the polynomial regression used to estimate it, given enough data. The figure provides some intuition for why. As the order of the polynomial increases, the line becomes more flexible and can accommodate unexpected twists and turns in the CEF.

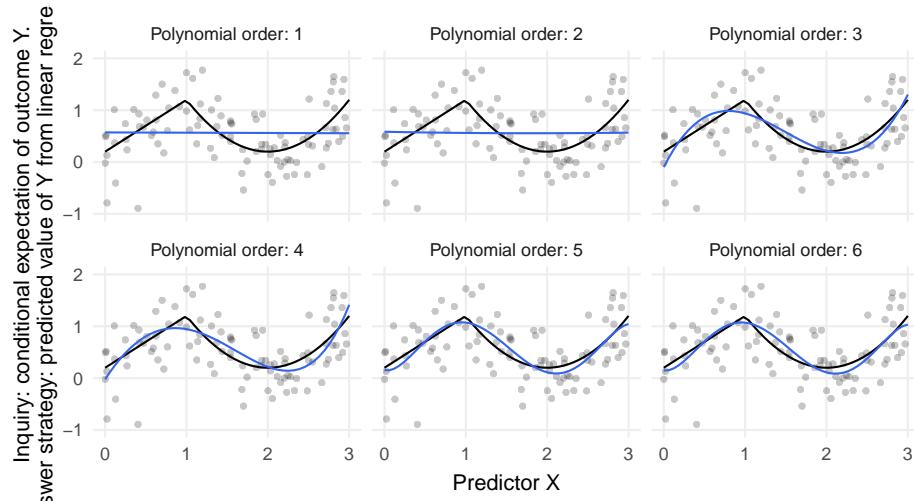


Figure 11.3: Estimating a CEF with polynomials of increasing order

Increasing the order of the polynomial decreases bias, but this decrease comes at the cost of variance. Figure 11.4 shows how, when the order increases, bias goes down while variance goes up. Mean squared error is one way to trade these two diagnosands off one another. Here, MSE is minimized with a polynomial of order 3. If we were to care much more about bias than variance, perhaps we would choose a polynomial of even higher order.

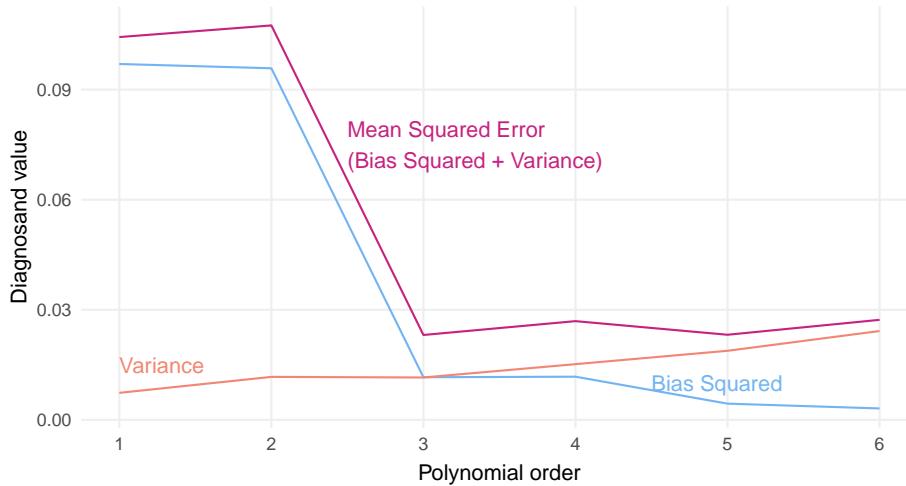


Figure 11.4: Redesigning a sample survey with respect to power

11.4 Redesign under model uncertainty

When we diagnose studies, we do so over the many theoretical we entertain in the model. Through diagnosis, we learn how the values of the diagnosands change depending on model parameters. When we redesign, we explore a *range* of empirical designs over the set of model possibilities. Redesign might indicate that one design is optimal under one set of assumptions, but that a different design would be preferred if a different set holds.

We illustrate this idea with an analysis of the minimum detectable effect (MDE) and how it changes at different sample sizes. The MDE diagnosand is complex. Whereas most diagnosands can be calculated with respect to a single possible model in M , the MDE is defined over a range of possible models. It is obtained by calculating the statistical power of the design over a range of possible effect sizes (holding the empirical design constant), then reporting the effect size that is associated with (typically) 80% statistical power.

MDEs can be a useful heuristic for thinking about the multiplicity of possibilities in the model. If the minimum detectable effect of a study is enormous – a one standard deviation effect, say – then we don't have to think much harder about our beliefs about the true effect size. Whatever our priors over the true effect size are, they are probably smaller than 1.0 SDs, so we can immediately conclude that the design is too small.

The declaration below contains uncertainty over the true effect size. This uncertainty is encoded in the `runif(n = 1, min = 0, max = 0.5)` command, which corresponds to our uncertainty over the ATE. It could be as small as 0.0 SDs or as large as 0.5 SDs, and we are equally uncertain about all the

values in between. We redesign over three values of N : 100, 500, and 1000, then simulate each design. Each run of simulation features a different true ATE somewhere between 0.0 and 0.5.

Declaration 11.4. Uncertainty over effect size design

```
N <- 100
design <-
  declare_population(N = N, U = rnorm(N),
    # this runif(n = 1, min = 0, max = 0.5) generates 1 random ATE between 0 and 0.5
    potential_outcomes(Y ~ runif(n = 1, min = 0, max = 0.5) * Z + U))
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_assignment(Z = complete_ra(N, prob = 0.5)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, inquiry = "ATE")
```

```
designs <- redesign(design, N = c(100, 500, 1000))
simulations <- simulate_designs(designs, sims = 500)
```

Figure 11.5 summarizes the simulations by smoothing over effect sizes: the loess curves describes the fraction of simulations that are significant at each effect size.¹ The MDEs for each sample size can be read off the plot by examining the intersection of each curve with the dotted line at 80% statistical power. At $N = 1000$, the MDE is approximately 0.175 SDs. At $N = 500$, the MDE is larger, at approximately 0.225 SDs. If the design only includes 100 units, the MDE is some value higher than 0.5 SDs. We could of course expand the range of effect sizes considered in the diagnosis, but if effect sizes above 0.5 SDs are theoretically unlikely, we don't even need to – we'll need a design larger than 100 units in any case.

This diagnosis and redesign shows how our decisions about the data strategy depend on beliefs in the model. If we think the true effect size is likely to be 0.225 SDs, then a design with 500 subjects is a reasonable choice, but if it is smaller than that, we'll want a larger study. Small differences in effect size have large consequences for design. Researchers who arrive at a plot like Figure 11.5 through redesign should be inspired to sharpen up their prior beliefs about the true effect size, either through literature review, meta-analysis of past studies, or through piloting (See section 21.5).

¹DeclareDesign tip: this procedure is more computationally efficient than an alternative, which would be to conduct simulations of each design at specific effect sizes across the plausible range.

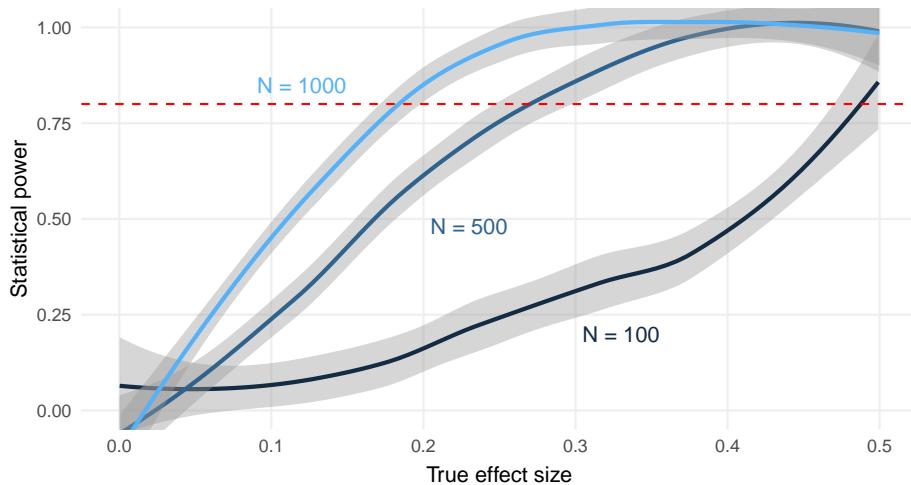


Figure 11.5: Redesigning an experiment over uncertainty about the true effect size

11.5 Redesigning in code

This chapter already displays the main computational approaches to redesign. The basic principle is that we need to create a list of designs which then get passed to `simulate_designs` or `diagnose_designs`. (The plural versions of these functions are identical to their singular counterparts, we just provide both to allow the code to speak for itself a little more easily).

You can make lists of designs to redesign across directly with `list`:

```
designs <- list(design1, design2)
```

More often, you'll vary designs over a parameter with `redesign`. Here, we're imagining we've already declared a `design` that has an `N` parameter that we allow to have 3 values.

```
designs <- redesign(design, N = c(100, 200, 300))
```

Whichever way you use to create `designs`, you can then diagnose all of the designs in the list with:

```
designs <- diagnose_designs(designs)
```

11.6 Summary

This section has explored some of the ways that you can use the redesign process to learn about design tradeoffs. Most often, the tradeoff is some measure of design quality like power against cost. We want to trade quality off against cost until we find a good enough study for the budget. Sometimes the tradeoff is across design parameters – should I sample more clusters or should I sample more people within clusters, holding costs constant? Sometimes the tradeoff is across diagnosands – more flexible answer strategies may exhibit lower bias but higher variance. Minimizing RMSE is weighs bias and variance equally, but other weightings are possible. Tradeoffs across diagnosands are implicit in many design decisions, but design diagnosis and redesign can help make those tradeoffs explicit.

Chapter 12

Design example

We illustrate the declare-diagnose-redesign framework with a study of political motivations among office-seekers in Pakistan. Gulzar and Khan (2021) conducted an experiment that estimated the effects of two alternative incentives for becoming a politician: helping the community or generating personal benefits. The researchers randomly assigned eligible citizens to receive different encouragements to stand for office and measured the rates of running for office, the types of people who chose to run, and the congruence of elected politicians' policy positions with those of the general population.

This design is moderately complex. We have multiple inquiries and a layered data strategy that has important implications for the answer strategy. The reason we selected it is that we want to show how you would actually apply the declare-diagnose-redesign framework in a real world setting with all its complexities.

12.1 Declaration in words

The model describes the units under study: citizens who are eligible to run for office in villages in the study region. The model also includes citizens' individual characteristics and their potential outcomes depending on which encouragement they receive. The model set includes four theories of political motivation: politicians respond only to encouragements that focus on themselves, only to encouragements that focus on others, to neither, or to both. Among theories that include room for both motivations, some claim that personal motivations are more powerful than community-minded motivations, while others claim the reverse. The potential outcomes are defined in terms of subjects' underlying ("latent") probability of running for office, which is tightly related to the binary choice to run or not to run.

The two inquiries for this study are the average treatment effects of each encouragement, defined as the average difference in potential outcomes between receiving and not receiving each encouragement to run. The authors consider a third inquiry — the difference between these two average treatment effects — but we'll leave that complication to the side for the moment. It's worth highlighting what the inquiry is not: the inquiry is not why do actual politicians run for office or what are the features of the job that attract candidates. The answers to the average treatment effect inquiries may shed light onto those questions, but not directly.

The data strategy for this study includes sampling, treatment assignment, and measurement. The sampling step takes place in two stages. First, the researchers sample 192 villages and then they sample 48 citizens who are eligible to stand for election from each village. In the assignment step, the authors allocate participants to a personal benefits encouragement, a prosocial encouragement, or no encouragement (control). All eligible citizen in a village are assigned to the same treatment condition, which is to say that this experiment used cluster random assignment. Lastly, the data strategy measures the decision to run for office by checking whether a participant's name appears on the official candidate lists released by the Election Commission of Pakistan. In contrast to the latent probability outcome in the model, the outcome variable as measured by the data strategy is binary.

The answer strategy is an ordinary least squares regression of outcome variable on the treatment variable, with standard errors clustered at the village level. The clustering of the errors reflects the clustered assignment of treatments. This mirroring is an example of how choices in the answer strategy should reflect choices in the data strategy.

12.2 Declaration in code

We are now ready to declare the Gulzar and Khan (2021) study in code.

Declaration 12.1. Gulzar and Khan (2021) design

With `declare_model`, we describe a hierarchical structure with 660 villages, each of which is home to many citizens who are eligible to run for elected office. Each citizen harbors three potential outcomes. `Y_Z_neutral` is the citizen's latent probability of standing for election if treated with a neutral appeal, `Y_Z_personal` is the probability if treated with an appeal that emphasizes the personal returns to office, and `Y_Z_social` is the probability if treated with an appeal that underlines the benefits to the community. Our simplified model assumes a constant treatment effect of about 3 percentage points for the personal appeal and 4 percentage points for the social appeal.¹

¹Readers may wonder where 3 and 4 percentage points are declared in the code. We use the `pnorm` function to define a latent variable representing the potential to run for office. The measured our binary outcome `Y_observed` springs from this latent outcome `Y_latent`. The effect sizes of

```

model <-
  declare_model(
    villages = add_level(N = 660, U_village = rnorm(N, sd = 0.1)),
    citizens = add_level(
      N = 100,
      U_citizen = rnorm(N),
      potential_outcomes(
        Y ~ pnorm(
          U_citizen + U_village +
          0.10 * (Z == "personal") +
          0.15 * (Z == "social")),
        conditions = list(Z = c("neutral", "personal", "social"))
      )
    )
  )
)

```

We have two inquiries, representing the average treatment effects in the population for the personal and social appeals compared to the neutral appeal, defined as the average differences in potential outcomes:

```

inquiry <- declare_inquiry(
  ATE_personal = mean(Y_Z_personal - Y_Z_neutral),
  ATE_social = mean(Y_Z_social - Y_Z_neutral)
)

```

The data strategy consists of four steps: sampling of villages, sampling of citizens, treatment assignment, and outcome measurement. In sampling, we sample 192 villages and 48 of the eligible citizens from each village. In assignment, we cluster assign 25% of the villages to the neutral condition, 37.5% to the personal appeal, and 37.5% to the social appeal. The measurement step maps the “revealed,” but still latent, probability of running to the observed choice to run or not.

```

n_villages <- 192
citizens_per_village <- 48

data_strategy <-

```

0.10 and 0.15 in the latent outcome translate in this setting to 3 and 4 percentage point effects on a binary scale. See Section 6.4.2 for further discussion.

```

declare_sampling(
  S_village = cluster_rs(clusters = villages, n = n_villages),
  filter = S_village == 1) +
declare_sampling(
  S_citizen = strata_rs(strata = villages, n = citizens_per_village),
  filter = S_citizen == 1) +
declare_assignment(
  Z = cluster_ra(
    clusters = villages,
    conditions = c("neutral", "personal", "social"),
    prob_each = c(0.250, 0.375, 0.375))) +
declare_measurement(
  Y_latent = reveal_outcomes(Y ~ Z),
  Y_observed = rbinom(N, 1, prob = Y_latent)
)

```

The answer strategy consists of an ordinary least squares regression (as implemented by `lm_robust`) of the outcome on the treatments. The standard errors are clustered at the village level in order to account for the clustering in the assignment procedure. The regression will return three coefficients: an intercept and two treatment effect estimates. We ensure that the estimators are mapped to the relevant inquiries by explicitly linking them.

```

answer_strategy <-
  declare_estimator(Y_observed ~ Z, term = c("Zpersonal", "Zsocial"),
                    clusters = villages,
                    model = lm_robust,
                    inquiry = c("ATE_personal", "ATE_social"))

```

When we concatenate all four elements with the `+` operator, we get a design:

```
design <- model + inquiry + data_strategy + answer_strategy
```

12.3 Diagnosis

To diagnose the design, we first define a set of diagnosands: bias, statistical power, the root mean-squared error, and total cost. The total cost calculation is in an arbitrary unit and reflects an assumption that sampling an additional vil-

Table 12.1: Diagnosis of the simplified Gulzar-Khan design.

inquiry	term	bias	rmse	power
ATE_personal	Zpersonal	0	0.014	0.511
ATE_social	Zsocial	0	0.014	0.847

large incurs a cost that is ten times larger than the cost of sampling an additional subject within a village.

```
diagnosands <- declare_diagnosands(
  bias = mean(estimate - estimand),
  rmse = sqrt(mean((estimate - estimand)^2)),
  power = mean(p.value <= 0.05),
  cost = mean(10 * n_villages + 1 * n_villages * citizens_per_village)
)
```

We then diagnose the design by simulating the design over and over, then calculating the diagnosands based on simulations data.

```
diagnosis <- diagnose_design(design, diagnosands = diagnosands, sims = sims, bootstrap_sims = b_sims)
```

The diagnosis reveals that the design is unbiased for both inquiries. The power of the design for the social treatment is above the standard 80% threshold but it is not for the personal treatment. This gives us a sense of what effect sizes the design is powered for, since the only difference in the design between these two inquiries is the assumed effect size in the model.

12.4 Redesign

Two of the most important design decisions in this study are the number of sampled villages and the number of sampled citizens per village. Due to the large fixed costs of traveling to each village, an additional sampled village is more expensive than an additional sampled citizen. In order to best allocate constrained study resources, we need understand the gains from changes to the data strategy along each margin. Here, we redesign the study across possible combinations of numbers of villages and citizens per village:

```

designs <- redesign(design,
                     n_villages = c(192, 500),
                     citizens_per_village = c(25, 50, 75, 100))

diagnosis <- diagnose_designs(designs, diagnosands = diagnosands)

```

In Figure 12.1, we illustrate the results of our redesign exercise across all four diagnosands. The number of citizens per village is plotted on the horizontal axis and the value of the diagnosand is shown on the vertical axis. The plot is faceted by diagnosand and each line represents a different possible number of villages. We focus here on the social treatment only.

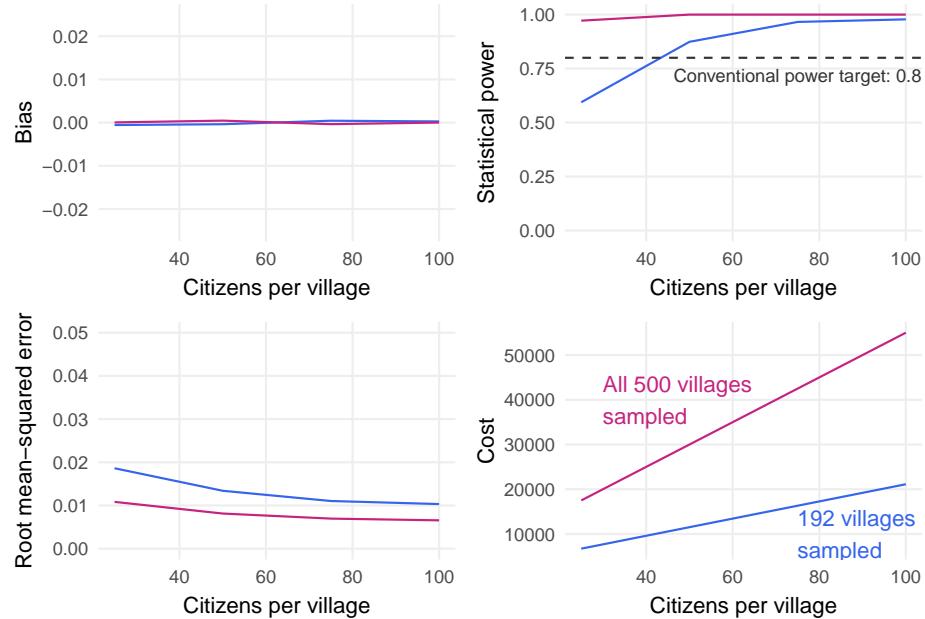


Figure 12.1: Redesign of Gulzar and Khan (2020)

What we see is that bias is invariant to these choices. The study is unbiased regardless of the number of villages and the number of citizens interviewed per village. However, our other three diagnosands do change. Power is increasing in the number of citizens per village, and is always higher with more villages. We might reject designs with 192 villages with only 25 citizens per village, because they fall below the 80% power threshold (in fact, the number chosen by the researchers, 48, is just over the threshold, suggesting they chose the most cost-effective design in terms of power). Root mean-squared error, a measure capturing both bias and efficiency of the design, is improving (decreasing) in

the number of citizens per village and the number of villages. Cost is, of course, increasing in both sample size parameters. We can use the cost parameters to make decisions about what sample sizes to choose accounting both for scientific diagnosands of the design (i.e., power) and cost at the same time.

194

Design example

12.4

Part III

Research Design Library

Chapter 13

Research Design Library

This section of the book enumerates a series of common social science research designs. Each entry will include description of the design in terms of M and also a declaration of the design in code. We'll often diagnose designs over the range of values of some design parameters in order to point out especially interesting or unusual features of the design.

Our goal in this section is not to provide a comprehensive accounting of all empirical research designs. It's also not to describe any of the particular designs in exhaustive detail, because we are quite sure that in order for these designs to be useful for any practical purpose, they will need to be modified. The entries in the design library are not recipes that will automatically produce high-quality research. Instead, we hope that the entries provide inspiration for how to tailor a particular class of designs to your own research setting.

The design library is also a corpus of design elements and code that can be mixed-and-matched to fit your particular research setting. We do not have a stepped-wedge experimental design with blocking, but you can create one if that is your design by combining elements from the stepped-wedge design and the block-randomized experimental design.

We've split up designs by inquiry and by data strategy. Inquiries can be descriptive or causal and data strategies can be observational or experimental. This gives rise to four categories of research design: observational descriptive, experimental descriptive, observational causal, and experimental causal. We dedicate chapters to each of these four as well as a chapter for "complex" designs – designs that involve multiple stages, multiple estimands, or inferences from multiple distinct projects.

Table 13.1: Research design types with examples

	Data strategy: Observational	Data strategy: Experimental
Inquiry: Descriptive	Sample survey or case study	List experiment or participant observation
Inquiry: Causal	Regression discontinuity design or process tracing	Randomized controlled trial

Chapter 14

Observational : descriptive

An observational design for descriptive inference has an inquiry like a population mean, covariance, or distribution as the main research goal. In an observational research design, the data strategy includes sampling and measurement components, but no treatments are allocated. Put differently, in an observational design for descriptive inference, researchers seek to measure and summarize the world, but not to change it. This class of research design encompasses a huge portion of research activity – most surveys fall into this class, as do large-scale data collections of economic and sociopolitical indicators. Other examples of observational designs for descriptive inference include classic case studies focused on “thick description” and many text analysis projects.

14.1 Simple random sampling

We declare a design in which a researcher takes a simple random sample of a population and uses an instrument to measure a latent quantity. The inquiry is the population average of the measured quantity. We show how to declare this design and an approach to incorporate concerns about non random non response in the design.

Descriptive inquiries like the population mean of one variable or the population correlation between two variables are defined with respect to a well-defined group of units – that group of N (written with an upper-case N) units is the population about which we want to draw inferences.

One approach to studying a population is to conduct a census in which we record data on all N units. A census has the clear advantage of being comprehensive, but it usually comes at an overwhelming and prohibitive cost.

To avoid those costs, we collect data from only n units (written with a lower-case n), where the sample size n is smaller than the population size N . When

the n units we happen to sample are chosen at random, we can obtain unbiased estimates of many descriptive inquiries.¹

Imagine we seek to estimate the average political ideology of adult residents of the small town of Portola, California (population 2,100). Under our model M , the latent ideology Y is drawn from a standard normal distribution.

The data strategy D has two components: a survey question Q and a sampling procedure S . The survey question asks subjects to place themselves on a left-right scale that varies from 1 (most liberal) to 7 (most conservative). We approximate this measurement procedure with a function that “cuts” the latent ideology into 7 separate groups. This survey question will introduce measurement error insofar as it does not distinguish among units with different latent ideologies despite placing themselves at the same place on the seven-point scale. Our main hope for this measurement procedure is that all of the people who give themselves higher scores are indeed more conservative than those who give themselves lower scores. The sampling procedure is “complete” random sampling. We draw a sample of exactly $n = 100$, where every member of the population has an equal probability of inclusion in the sample, $\frac{n}{N}$.

The model and data strategy are represented by the DAG in Figure 14.1. The DAG shows that the observed outcome Y is a function of the latent score Y and the survey question Q . The observed outcome Y is only measured for sampled units, i.e., units that have $S = 1$. This simple diagram represents important design assumptions. First, no arrow leads from Y to S . If such an arrow were present, then units with higher or lower latent ideologies would be more likely to be sampled. Second, an arrow does lead from Y to Y , indicating that we assume the measured variable does indeed respond to the latent variable. Finally, the diagram includes an explicit role for the survey question, which helps us to consider how alternative wordings of Q might change the observed variable Y .

Our inquiry I is the population mean of the *measured* variable Y : $\frac{1}{N} \sum_i Y_i = Y$, rather than the mean of the latent variable Y . In this sense, our inquiry is “data-strategy dependent”, since we are interested in the average value of what we *would* measure for any member of the population were we to sample them.

Our answer strategy is the sample mean estimator: $\bar{Y} = \frac{1}{n} \sum_i Y_i$, implemented here as an ordinary least squares regression to facilitate the easy calculation of auxiliary statistics like the standard error of the estimate and the 95% confidence interval.

We incorporate these design features into Declaration 18.5. The `portola` object is a fixed population of 2100 units with a latent ideology `Y_star`. The declaration of the measurement strategy comes before the declaration of the inquiry,

¹But not all. Hedayat, Cheng and Pajda-De La O (2019) prove that that unbiased estimators of the population minimum, maximum, and even median do not exist for any sampling procedure except a census. The intuition behind this result is easiest to see for a maximum: unless the random sample happens to contain the unit with the highest value of the outcome, the estimate will necessarily fall below the maximum.

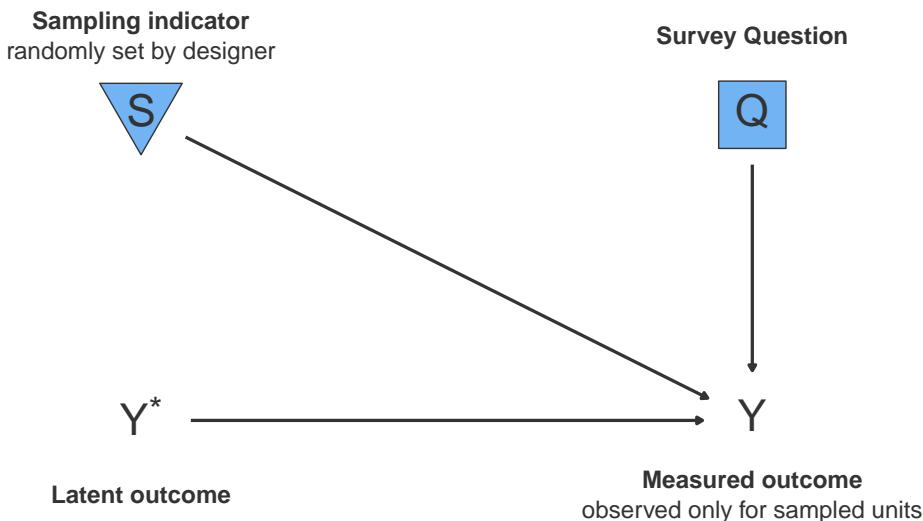


Figure 14.1: DAG for the simple random sampling design

showing how the inquiry is data strategy dependent.

Declaration 14.1.

```
set.seed(343)
portola <-
  fabricate(
    N = 2100,
    Y_star = rnorm(N)
  )

design <-
  declare_model(data = portola) +
  declare_measurement(Y = as.numeric(cut(Y_star, 7))) +
  declare_inquiry(Y_bar = mean(Y)) +
  declare_sampling(S = complete_rs(N, n = 100)) +
  declare_estimator(Y ~ 1, inquiry = "Y_bar")
```

Two main diagnosands for the simple random sampling design are bias and rmse. We want to know if we get the right answer on average and we want to know, on average, how far off from the truth we are.

Table 14.1: Complete random sampling design diagnosis

Bias	RMSE
-0.01 (0.01)	0.11 (0.00)

```
diagnosands <- declare_diagnosands(
  bias = mean(estimate - estimand),
  rmse = sqrt(mean((estimate - estimand) ^ 2))
)
diagnosis <- diagnose_design(design, diagnosands = diagnosands)
```

The diagnosis in table 14.1 indicates that under complete random sampling, the sample mean estimator of the population mean is unbiased and that the root mean squared error is manageable at 0.11.

14.1.1 What can go wrong

The most serious threat to descriptive inference in a randomized sampling design is nonresponse. Missingness due to nonresponse can lead to bias if missingness is correlated with outcomes. Sometimes this bias is referred to as “selection bias” in the sense that some units select out of responding when sampled.

Depending on what information is available about the missing units, various answer strategy fix-ups are available to analysts. For example, if we have demographic or other covariate information about the missing units, we can search for similar-seeming units in the observed data, then impute their outcomes for the missing outcomes. This approach depends on the strong assumption that units with the same covariate profile have the same average outcome, regardless of whether they go missing. The imputation process is often done on the basis of a regression model; multiple imputation methods attempt to incorporate the additional uncertainty that accompanies the modeling technique (see [decent introduction to MI]).

Avoiding – or dynamically responding to – missingness in the data strategy is usually preferable to the addition of modeling assumptions in the answer strategy. Avoiding missigness often means adding extra effort and expense: monetary incentives for participation, multiple rounds of attempted contact, and a variety of modes of contact (phone, mail, email, direct message, text, canvass). The best way to allocate extra effort will vary from context to context, as will the payoff from doing so. Our recommendation is to reason about the plausible response rates that would result from different levels of effort, then to consider

how to optimize the bias-effort tradeoff. Sometimes, achieving zero bias would be far too costly, so we would be willing to tolerate some bias because effort is too expensive.

Declaration 14.2 builds in a dependence between the latent outcome Y and the probability of responding to the survey. That probability also responds to researcher effort. The diagnosis shows how effort translates into higher response rates and lower bias:

Declaration 14.2. Survey nonresponse design

```
design <-
  declare_model(data = portola) +
  declare_measurement(Y = as.numeric(cut(Y_star, 7))) +
  declare_inquiry(Y_bar = mean(Y)) +
  declare_sampling(S = complete_rs(N, n = 100)) +
  declare_measurement(
    R = rbinom(n = N, size = 1, prob = pnorm(Y_star + effort)),
    Y = if_else(R == 1, Y, NA_real_)
  ) +
  declare_estimator(Y ~ 1, inquiry = "Y_bar") +
  declare_estimator(R ~ 1, label = "Response Rate")
```

```
designs <- redesign(design, effort = seq(0, 5, by = 0.5))
diagnosis <- diagnose_designs(designs, sims = 500)
```

14.2 Cluster random sampling

We declare a design in which a researcher takes a clustered random sample of a population and uses the design to ask whether they should invest in sampling more clusters or in more individuals per cluster. The design includes a budget constraint that affects how many can be sampled within clusters as a function of the number of clusters sampled.

Researchers often cannot randomly sample at the individual level because it may, among other reasons, be too costly or logically impractical. Instead, they may choose to randomly first sample clusters, then sample units within clusters. Clusters might be schools, localities, or households.

How does clustering change the research design relative to the individual-level design? First, we need to elaborate the model M to make the clustering hierar-

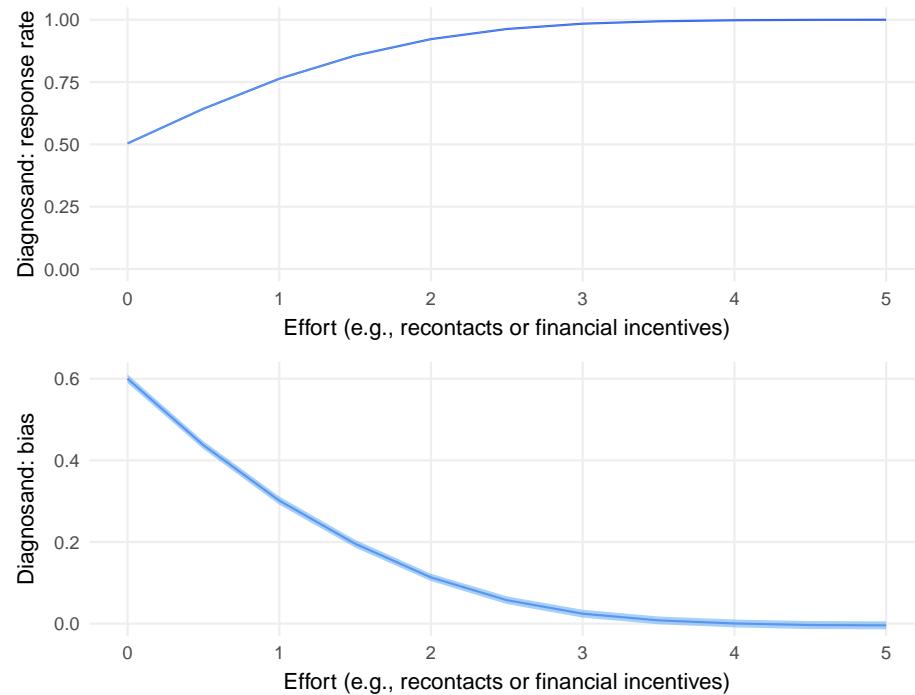


Figure 14.2: Redesigning the random sampling design over researcher effort

chy explicit. In the declaration below, we imagine our research site is two states in Nigeria. Localities are nested within states and individuals are nested within localities. Second, we want to respect this hierarchy when thinking about the distribution of outcomes. Individuals living in the same locality are likely to share ideological view points, either through the explicit transmission of political views or because of common exposure to political influence. The “intra-cluster correlation” or ICC is an extremely important statistic for the design of cluster-sampled studies. It describes what fraction of the total variation in the outcome can be attributed to the across-cluster differences in average outcomes.

In the declaration below, the latent outcome Y_{star} describes subject's latent political ideology. This latent outcome is a function of a locality shock and an individual shock. The variances of these two shocks are determined by the ICC parameter. If ICC were equal to 1, the variance across localities would be equal to 1, and all individuals within a locality would have exactly the same ideology. If ICC were equal to zero, then the variation in ideology would take place entirely at the individual level.

Declaration 14.3.

```

set.seed(343)
ICC <- 0.4

two_nigerian_states <-
  fabricate(
    state = add_level(N = 2,
                      state_name = c("taraba", "kwara"),
                      state_mean = c(-0.2, 0.2)),
    locality = add_level(
      N = 500,
      locality_shock = rnorm(N, state_mean, sqrt(ICC)))
  ),
  individual = add_level(
    N = 100,
    individual_shock = rnorm(N, sd = sqrt(1 - ICC)),
    Y_star = locality_shock + individual_shock
  )
)

```

Many different cluster sampling designs are possible, but a very standard choice is a two-stage design in which first, some but not all clusters are sampled, and second, some but not all units within a cluster are sampled. The sampling at either stage may be stratified by covariates at the appropriate level. The first stage can be stratified by cluster-level covariates and the second stage can be stratified by individual-level covariates in order to improve precision. In this declaration, we form cluster-level strata by state.

The two stage random-sampling design raises an important tradeoff: Should we invest in sampling more clusters or in more individuals per cluster? Typically, adding the marginal cluster is more expensive than adding the marginal individual. We formalize the tradeoff with a “budget function” that returns the largest individual level inclusion probability that is budget-compatible with a given cluster sampling probability:

```

budget_function <-
  function(cluster_prob){
  budget = 20000
  cluster_cost = 20
  individual_cost = 2
  n_clusters = 1000
}

```

```

n_individuals_per_cluster = 100

total_cluster_cost <-
  cluster_prob * n_clusters * cluster_cost

remaining_funds <- budget - total_cluster_cost

sampleable_individuals <-
  cluster_prob * n_clusters * n_individuals_per_cluster

individual_prob =
  (remaining_funds/individual_cost)/sampleable_individuals

pmin(individual_prob, 1)
}

```

We use the output of this function to determine the probability of an individual being sampled, conditional on their cluster being sampled.

Lastly, the answer strategy must also respects the data strategy by clustering standard errors at the highest level at which units are sampled, which in this case is the locality.

Declaration 14.4.

```

design <-
  declare_model(data = two_nigerian_states) +
  declare_measurement(Y = as.numeric(cut(Y_star, 7))) +
  declare_inquiry(Y_bar = mean(Y)) +
  declare_sampling(
    S_cluster = strata_and_cluster_rs(
      strata = state,
      clusters = locality,
      prob = cluster_prob
    ),
    filter = S_cluster == 1
  ) +
  declare_sampling(
    S_individual =
      strata_rs(strata = locality,
                prob = budget_function(cluster_prob)),
    filter = S_individual == 1
  )

```

```
declare_estimator(Y ~ 1,
                  clusters = locality,
                  inquiry = "Y_bar")
```

We redesign Declaration 14.4 over various levels of the cluster-level probability of sampling, which in turn sets the probability of sampling at the individual level. Figure 14.3 shows that for a good while, adding additional clusters yields precision gains. At some point, however, the cost to sample size within cluster is too large, and we start seeing precision loss around a probability of 0.75 of a cluster being sampled. The precise combination of design parameters that minimize the standard deviation of the sampling distribution will depend on nearly every aspect of the design declaration, but the most important are the total budget, the relative costs of clusters and individuals, and the ICC. When the ICC is zero, we should invest in few clusters and many individuals. When the ICC is one, we should invest in many clusters and few individuals.

```
designs <- redesign(design, cluster_prob = seq(0.1, 0.9, 0.1))
diagnosis <- diagnose_design(designs, sims = 500)
```

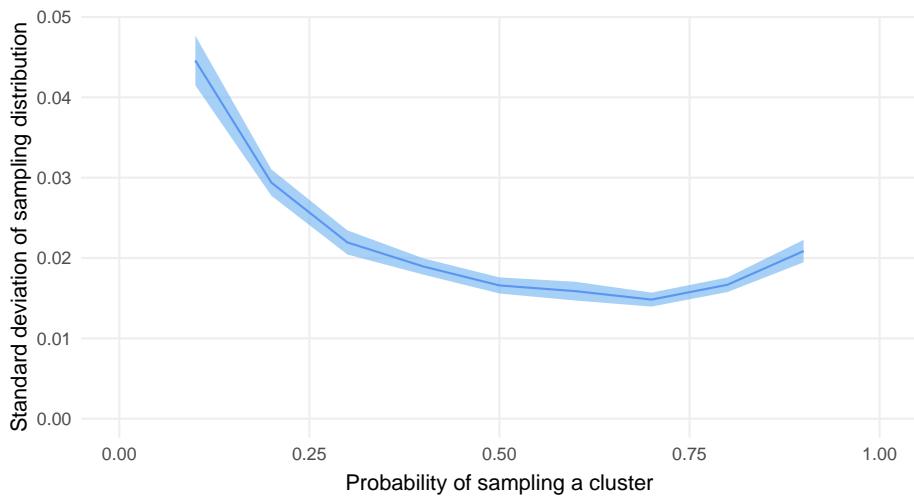


Figure 14.3: Trading off the number of clusters and the number of individuals per cluster

14.3 Multi-level regression and poststratification

We declare a design in which researchers reweight the responses of different units in a sample in order to better estimate a population level quantity. Reweighting depends on how much units are thought to “represent” other unsampled units and requires making decisions about how much units of different types should be pooled together. Design performance of a partially pooled model is compared against designs that involve no pooling and full pooling.

Multi-level regression and postratification (MRP) is a technique used primarily for “small area estimation.” In the prototypical setting, we conduct a nationally-representative survey of public opinion, but our goal is to generate estimates of the average level of public opinion for many subnational units. In the United States context, these “small area” units are often the 50 states. The main problem is that in a national poll of 2,000 Americans, we might only have 4 respondents from small states like Alaska, Wyoming, or Vermont, but more than 100 from large states like California, New York, or Texas. Accordingly, it is harder to estimate opinion in small states than in large states. The key insight of an MRP design is that we can “borrow strength” across states and kinds of people in order to improve state level estimates.

In an MRP design, the answer strategy includes two steps: a multi-level regression step and a poststratification step. The regression model generates estimates of the average opinion for classes of people within each state. The precise flavor of regression model can vary from application to application. In the simple example below, we use a generalized linear mixed effect model with an individual-level covariate and random effects by state, but regression models of astounding complexity are sometimes used to model important nuances in how opinions covary with individual and state-level characteristics.

The regression model generates estimates of the average opinion for classes of people within each state – the post-stratification step reweights these estimates to *known* proportions of each class of person within each state. The knowledge of these proportions has to come from outside the survey. The US census is the usual source of these population proportions, though any reliable source of this information is suitable as well.

We begin with a dataset of the fifty states that describes what fraction of people in each state has graduated high school. This code block also establishes the true `state_means` that will be our inquiry.

```
states <-  
  as_tibble(state.x77) %>%  
  transmute(
```

```

state = rownames(state.x77),
prop_of_US = Population / sum(Population),
prob_HS = `HS Grad` / 100,
state_shock = rnorm(n = n(), sd = 0.5),
state_mean = prob_HS * pnorm(0.2 + state_shock) + (1 - prob_HS) * pnorm(state_shock)
)

```

Here we declare the design. In the model, we draw a nationally-representative sample of size 2,000, respecting the fraction of people within each state with a high school degrees. The post-stratification weights are built from the that fraction as well. The binary public opinion variable `policy_support` is a function of the high school covariate, an individual-level shock, and a state-level shock. The inquiry is the mean policy support at the state level.

Declaration 14.5.

```

design <-
declare_model(
  data = states[sample(1:50,
                      size = 2000,
                      replace = TRUE,
                      prob = states$prop_of_US),],
  HS = rbinom(n = N, size = 1, prob = prob_HS),
  PS_weight =
    case_when(HS == 0 ~ (1 - prob_HS),
              HS == 1 ~ prob_HS),
  individual_shock = rnorm(n = N, sd = 0.5),
  policy_support = rbinom(N, 1, prob = pnorm(0.2 * HS + individual_shock + state_shock))
) +
declare_inquiry(
  handler = function(data) {
    states %>% transmute(state, inquiry = "mean_policy_support", estimand = state_mean)
  }
)

```

The tricky part of this design is the two-step answer strategy. The first step is handled by the multilevel regression function `glmer`. The second step is handled by the `ps_helper` function that obtains predictions from the model, then reweights them according to the post-stratification weights.

```
ps_helper <- function(model_fit, data) {
  prediction(
    model_fit,
    data = data,
    allow.new.levels = TRUE,
    type = "response"
  ) %>%
  group_by(state) %>%
  summarize(estimate = weighted.mean(fitted, PS_weight))
}
```

```
design <-
  design +
  declare_estimator(handler = label_estimator(function(data) {
    model_fit <- glmer(
      formula = policy_support ~ HS + (1 | state),
      data = data,
      family = binomial(link = "logit")
    )
    ps_helper(model_fit, data = data)
  )),
  label = "Partial pooling",
  inquiry = "mean_policy_support")
```

Figure 14.4 shows one draw from this design, plotting the MRP estimates against the true level of opinion.

14.3.1 Redesign over answer strategies

The strengths of the MRP design are best appreciated by contrasting MRP's partial pooling approach to two alternatives: no pooling and full pooling. Under no pooling, we estimate each state mean separately with a national adjustment for the individual-level high school covariate. Under full pooling, we only adjust for high school and ignore state information altogether. Here we add both estimators to the design and diagnose.

```
design <-
  design +
```

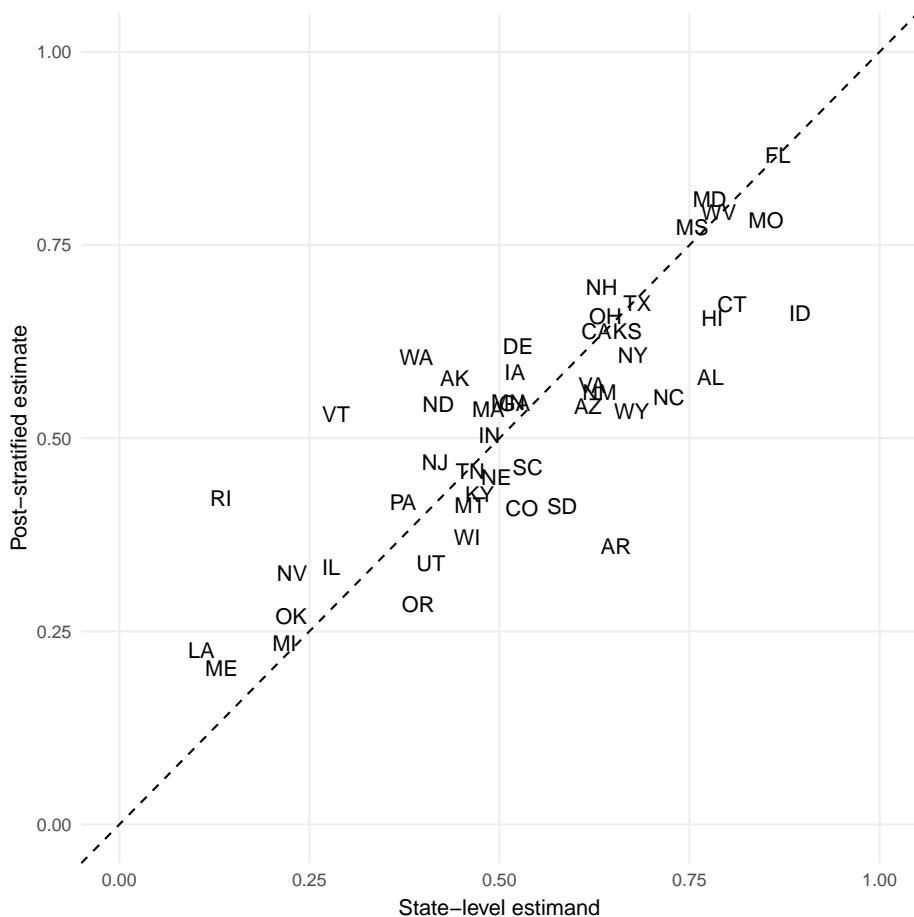


Figure 14.4: Estimates of state-level option plotted against their true levels

```
declare_estimator(
  handler = label_estimator(function(data) {
    model_fit <- lm_robust(
      formula = policy_support ~ HS + state,
      data = data
    )
    ps_helper(model_fit, data = data)
  }),
  label = "No pooling",
  inquiry = "mean_policy_support") +
```

```

declare_estimator(
  handler = label_estimator(function(data) {
    model_fit <- lm_robust(
      formula = policy_support ~ HS,
      data = data
    )
    ps_helper(model_fit, data = data)
  }),
  label = "Full pooling",
  inquiry = "mean_policy_support")

```

Figure 14.5 compares the three estimators. The first column of facets shows one draw of the estimates against the estimands. The main thing to notice here is that the full pooling estimate is more or less a flat line – regardless of the estimand, the estimates are just above 50%. Relative to partial pooling, the no pooling estimates are further spread around the 45 degrees line, with small states bouncing around the most.

On the right side of the figure, we see the bias, RMSE, and standard deviation diagnosands for each inquiry under all three answer strategies. Under no pooling, bias is very low, but the RMSE and standard deviation for small states is very high. Under full pooling, the standard deviation is very low, but bias is very positive for states with low support and very negative for states with high support. The resulting RMSE has a funny “V” shape – we only do well for states that happen to have opinion that is very close to the national average.

Partial pooling represents a Goldilocks compromise between full and no pooling. Yes, we have some positive bias for low-opinion states and negative bias for high opinion states, but variance has been brought under control. As a result, the RMSE for both small and large states is small.

14.4 Index creation

We declare a design in which researchers take multiple measures and combine them to learn about a latent, unobservable quantity. We use diagnosis to show that it is possible to generate interpretable conditional estimates of this quantity and assess bias even though the metric of the unobservable quantity is unknown. Diagnosis highlights however how subtle differences in scale construction generate different types of bias.

Models often specify a latent variable (Y_{star}) that we can't observe directly. The measurement procedures we use to produce observed variables (Y_1 , Y_2 , Y_3) are not perfect: observed values are related latent variables but may be on their own scales. A common strategy for addressing measurement error is

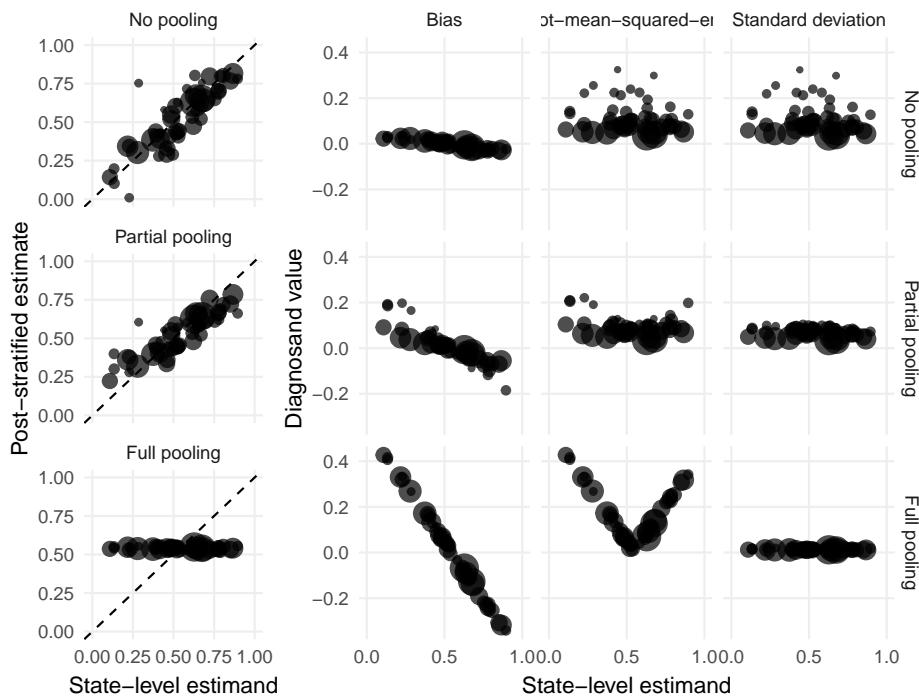


Figure 14.5: Comparison of three answer strategies

to combine multiple measures of the same latent variable into an index. The basic intuition for this procedure is that when we combine multiple measures, the errors attending to each measure will tend to cancel one another out. When this happens, the index itself has lower measurement error than any of the constituent parts.

The first difficult feature of such problems is that we do not have access to the *scale* on which Y_{star} is measured and so it may seem like a hopeless exercise to try to assess whether we have got good or bad estimates of Y_{star} when we combine the measured data.

One way around this is to normalize the scale of both the latent variable and the measured variable so that they have a mean of 0 and unit standard deviation. But in that case, we are guaranteed that our estimate of the mean of the normalized variable will be unbiased! We will certainly estimate a mean of 0! That may be but as we show in the declaration below, if your model is correct this approach may still be useful for calculating other quantities — such as conditional means—that you don't just get right by construction.

A second challenge is deciding which measurements to combine into the index. We clearly only want to create indices using measurements of the same latent variable, but it can hard to be sure which latent variable, exactly, is being mea-

sured. Just relying on whether the measurements are positively correlated or not is not sufficient, because measurements can be correlated even if they are not measurements of the same underlying variable. Ultimately we have to rely on theory to make these decisions, as uncomfortable as that may make us.

Lastly, we have to choose a procedure to combine the multiple measures into a single index. Many such procedures exist. Here we'll consider the most common approach, which is simply to take the average of the three raw measures.

In Declaration 14.6, our inquiry is the average level of the latent variable among units whose binary covariate X is equal to one.

In the declaration below Y_{star} has a normal distribution but it is not centered on zero. The measured variables Y_1 , Y_2 , Y_3 are also normally distributed but each has its own scale; they are all related to Y_{star} , though some more strongly than others. We construct two indices. One (Y_{av}) is constructed by first scaling each of these measured variables, then averaging and then scaling again. This is akin to the approach used in Kling, Liebman and Katz (2007). The second also weights the components but this time using weights generated by 'principal components analysis' which, intuitively, seeks to find a weighting that minimizes the distance to the measured variables.

Declaration 14.6.

```
design <-
  declare_model(
    N = 500,
    X = rep(0:1, N / 2),
    Y_star = 1 + X + 2 * rnorm(N)
  ) +
  declare_inquiry(Y_bar_X1 = mean(scale(Y_star)[X == 1])) +
  declare_measurement(
    Y_1 = 3 + 0.1 * Y_star + rnorm(N, sd = 5),
    Y_2 = 2 + 1.0 * Y_star + rnorm(N, sd = 2),
    Y_3 = 1 + 0.5 * Y_star + rnorm(N, sd = 1),
    Y_av = ((scale(Y_1) + scale(Y_2) + scale(Y_3))),
    Y_av_adj = (
      # rescaling according to the X = 0 group
      (Y_1 - mean(Y_1[X == 0])) / sd(Y_1[X == 0]) +
      (Y_2 - mean(Y_2[X == 0])) / sd(Y_2[X == 0]) +
      (Y_3 - mean(Y_3[X == 0])) / sd(Y_3[X == 0])
    ) / 3,
    Y_av_scaled = scale((scale(Y_1) + scale(Y_2) + scale(Y_3))),
    Y_fa = princomp(~ Y_1 + Y_2 + Y_3, cor = TRUE)$scores[, 1]
  ) +
  declare_estimator(
```

```

Y_av ~ 1,
model = lm_robust,
inquiry = "Y_bar_X1",
subset = X == 1,
label = "Average"
) +
declare_estimator(
    Y_av_scaled ~ 1,
    model = lm_robust,
    inquiry = "Y_bar_X1",
    subset = X == 1,
    label = "Average (rescaled)"
) +
declare_estimator(
    Y_av_adj ~ 1,
    model = lm_robust,
    inquiry = "Y_bar_X1",
    subset = X == 1,
    label = "Average (adjusted)"
) +
declare_estimator(
    Y_fa ~ 1,
    model = lm_robust,
    inquiry = "Y_bar_X1",
    subset = X == 1,
    label = "Principal Components"
)

```

In 14.6 we show that the correlation between all the indices and the underlying quantity is quite good, even though the strength of the correlations for some of the components is weak. Trickier however is being sure we have an interpretable *scale*.

```
diagnosis <- diagnose_design(design)
```

Figure 14.7 shows the distribution of estimates from different approaches to generating indices. A few features stand out from the distribution of estimates.

First, the principal component version appears to do poorly, but there is a simple reason for this: the method does not presuppose knowledge of the *direction* of the scale of the latent variable and can come up with index that reverse

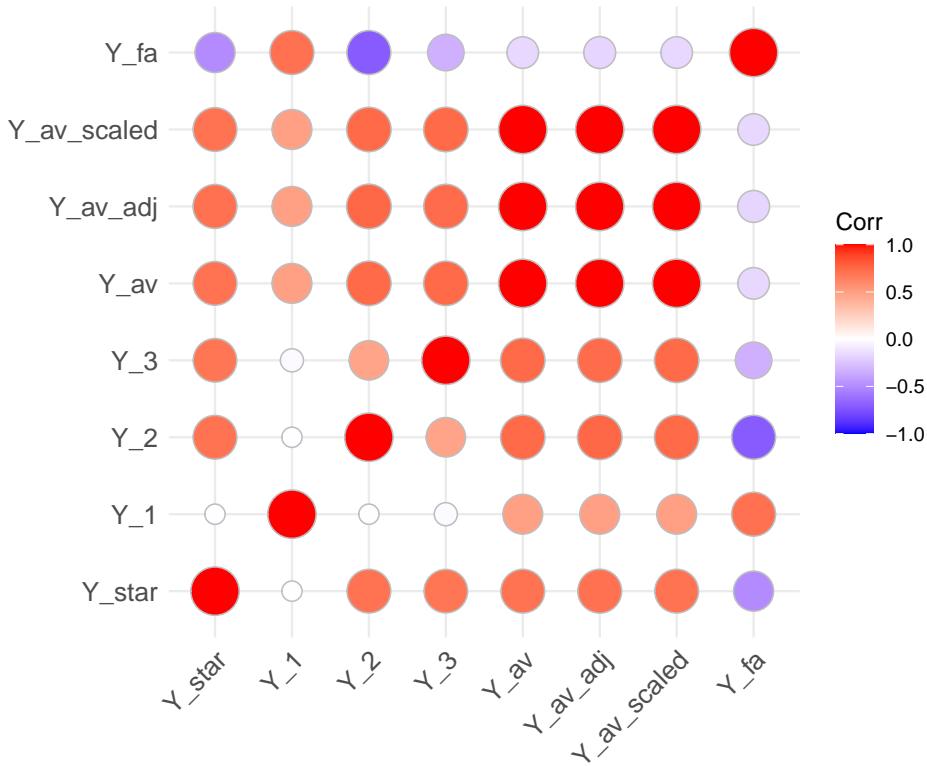


Figure 14.6: Correlations between the latent variable, measured components, and indices.

the actual scale of interest. Avoiding this requires an interpretative step after the principal components analysis is implemented (more subtly the averaging approach also has an interpretative step *before* averaging, when components are introduced with a metric that presupposes a positive correlation with the underlying quantity). Even accounting for the different sign patterns however, the estimates are too small.

The simple averaging of the normalized scales, by contrast has generated estimates that are too large on average. Two rescaling approach fare somewhat differently. An approach that simply rescales again after rescaling the component parts produces bias in the opposite direction. A third approach in which the component parts are rescaled in terms of the average and standard deviations observed in the women's group does well however. The key insight here is that the total variation in the latent variable combines the variation within groups and between them: we want to measure outcomes on a scale benchmarked to the within group variation and so have to take out the between group variation when rescaling.

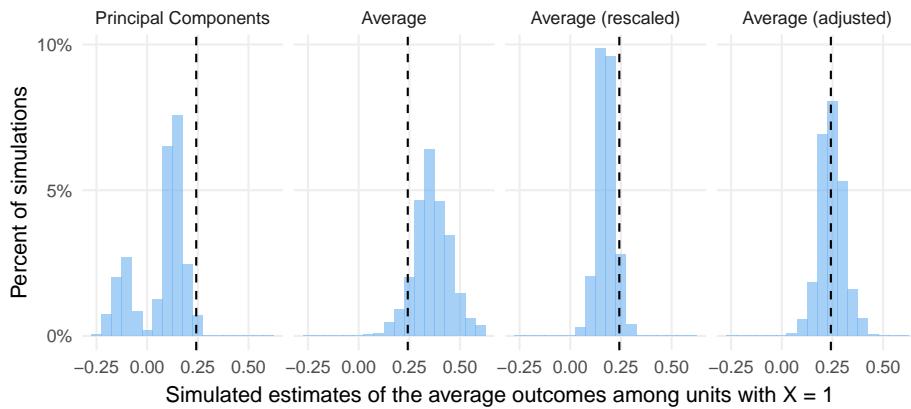


Figure 14.7: Distribution of estimates from different approaches to generating indices

Overall we see from the diagnosis that we *can* do quite well here in recovering the conditional mean of the standardized latent variable. Though we see risks here. We declared a design under optimistic conditions in which we knew the relevant components and these were all related to the latent variable in a simple way. Even in this case we had difficulty getting the answer right.

14.4.1 Exercises

1. Modify the design so that only two rather than three observed components are used. How is the diagnosis affected? Does it matter which indices you use?
2. How is estimation affected if you have the wrong model linking measures to the latent variables? Try a design in which the measures are non linear in the latent variable.

218

Observational : descriptive

14.4

Chapter 15

Observational : causal

In an observational design, researchers do not themselves set the conditions to which units are exposed; the natural processes of the world are responsive for the observed treatment conditions. Observational causal inference is the art of inferring what the effects of the observed treatments were, or what the effects of treatments not applied by the work would have been had things been different.

In some accounts, “observational causal inference” is an oxymoron, but this is wrong. An “observational experiment” is an oxymoron, but observational causal inference is possible and necessary – just sometimes unflinchingly difficult. It’s hard because observational causal inference depends on the world generating just the right circumstances. Process tracing requires observation of just the right clues to the causal mystery. Section on observables requires measurement of a set of variables that collectively close all the back door paths from treatment to outcome. A difference-in-difference design requires a certain kind of overtime stability – that untreated potential outcomes move in parallel. Instrumental variables needs nature to randomly assign something we can measure. Regression discontinuity designs requires a cutoff that we can observe and understand.

These are the big five observational causal research designs as we see it. Many additional innovative designs have been developed to try to estimate causal quantities with observational data. These generally seek clever innovations in A in order to have as few assumptions on M as possible (Principle 3.3: Entertain many models). We refer readers to Angrist and Pischke (2008) and Dunning (2012) for excellent overviews of the theory behind many of these methods.

15.1 Process tracing

We declare a qualitative design in which researchers seek to learn about the effect of a cause on a single unit. The diagnosis helps evaluate the gains from

different within-case data gathering strategies.

In qualitative research we are often interested in learning about the causal effect for a single unit. For example, for a country unit that underwent sanctions (an “event”), we want to know the causal effect of the sanctions on government behavior. To do so, we need to know what would have happened if the event did not happen, the *counterfactual outcome* that did occur as opposed to the factual outcome that did not. Due to the fundamental problem of causal inference, we cannot observe what would have happened if that counterfactual case had happened. We have to guess—or infer—what would have happened (Principle 3.5). Social scientists have developed a large array of tools for guessing missing counterfactual outcomes — what would have happened in the counterfactual case, if the event had not happened.¹

A common inquiry in this setting is whether an outcome was *due* to a cause. This “causal attribution” inquiry or “Cause of Effect” inquiry can be written $\text{CoE} = 1 | Y(0)|X = 1 \& Y = 1$. For a unit with $X = 1$ and $Y = 1$, $\text{CoE} = 1$ if $Y(0) = 0$. This “Cause of Effect” inquiry is different from the treatment effect of the treated case (TET). The TET is the difference between the treated potential outcome and the control potential outcome in the posttreatment period for the treated unit of interest. The TET is defined regardless of whether the unit’s outcome equals 1 or 0, but the “cause of effect” inquiry is asked only for units whose treated outcome is equal to one.

“Process tracing” is a prominent strategy for assessing causes of effects (Bennett and Checkel, 2015; Fairfield and Charman, 2017). Here, following for example Humphreys and Jacobs (2017), we think of process tracing as a procedure in which researchers provide a theory, in the form of a causal model, that is rich enough to characterize the probability of observing ancillary data (“Causal process observations” (Brady, 2004)) given underlying causal relations. If equipped with priors, such a model in turn lets one use Bayes’ rule to form posteriors over causal relations when these observations are observed.

For intuition, say we are interested in whether a policy caused a change in economic outcomes. We expect that for the policy to matter it at least had to be implemented and so if we find out that it was not implemented we infer that it did not matter. We make theory dependent inferences that are reasonable *insofar as* the theory is reasonable. In this example, if there are plausible channels through which a policy might have mattered even if not implemented, then our conclusion would not be warranted.

To illustrate design choices for a process tracing study we consider a setting in which we have observed $X = 1$ and $Y = 1$ and we are interested in figuring out whether $Y = 1$ because $X = 1$. More specifically we imagine a model with two ancillary variables, M and W . We posit that X causes Y via M —with negative

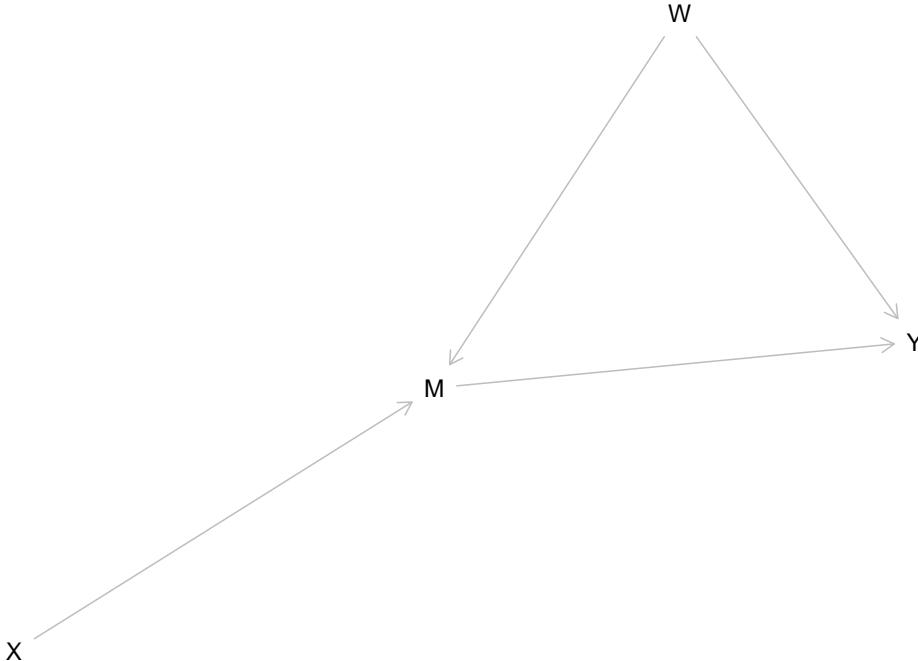
¹We are also interested in the opposite case sometimes: we have a unit that did not experience an event, and we want to know the causal effect of *not* having it. In this case, we need to guess what would have happened if the event did happen. The same tools apply in reverse.

effects of X on M and of M and Y ruled out. And we posit that W is a cause of both M and Y , specifically we posit that if $W = 1$ then X causes M and M causes Y for sure. Under this model M and W each serve as “clues” for the causal effect of X on Y .

Using the language popularized by Van Evera (1997), M provides a “hoop” test: if you look for data on M and find that $M = 0$ then you infer that X did not cause Y . If on the other hand you examine W and find that $W = 1$ then you have a “smoking gun” test: You infer that X did indeed cause Y . However, if you find both $M = 0$ and $W = 1$, then you know your model is wrong.

The model can be described using the `CausalQueries` package thus:

```
model <- make_model("X -> M -> Y <- W -> M") %>%
  set_restrictions("(M[X=1] < M[X=0]) | (M[X=1, W=1] == M[X=0, W=1])") %>%
  set_restrictions("(Y[M=1] < Y[M=0]) | (Y[M=1, W=1] == Y[M=0, W=1])") %>%
plot(model)
```



This model definition describes the DAG but also specifies a set of restrictions on causal relations. By default flat priors are then placed over all other possible causal relations, though of course other prior beliefs could also be specified.

Table 15.1: Beliefs for a case with $X = 1, Y = 1$. The first row gives the prior belief that $X = 1$ caused $Y = 1$. The second row gives the expectation that $M=1$. The last rows give posterior beliefs that $X=1$ caused $Y=1$ after M is observed, depending on what is found.

Query	Given	mean
Prob(CoE=1)	$Y==1 \& X==1$	0.71
Prob(M=1)	$Y==1 \& X==1$	0.93
Prob(CoE=1 M=0)	$Y==1 \& X==1 \& M==0$	0.00
Prob(CoE=1 M=1)	$Y==1 \& X==1 \& M==1$	0.77

We now have all we need to assess what inferences we might make given different sorts of observations using `CausalQueries::query_model`. This function lets you ask any causal question of a model conditional on observed (or counterfactual) conditions. We can use it to calculate beliefs, likely data, and conditional inferences thus:

```
queries <-
  CausalQueries::query_model(
    model,
    query = list('Prob(CoE=1)' = "Y[X=1] > Y[X=0]",
                 'Prob(M=1)' = "M==1",
                 'Prob(CoE=1 | M=0)' = "Y[X=1] > Y[X=0]",
                 'Prob(CoE=1 | M=1)' = "Y[X=1] > Y[X=0]"),
    given = list("Y==1 & X==1",
                 "Y==1 & X==1",
                 "Y==1 & X==1 & M==0",
                 "Y==1 & X==1 & M==1"),
    using = "parameters")
```

From the model specification, we can calculate directly the probability of observing $M = 0$ or $M = 1$ (or $W = 0$ or $W = 1$) and what we would infer in each case. Table 15.1 shows an example of these values. From a table like this we can calculate directly the expected posterior variance associated with a process tracing data strategy and a Bayesian answer strategy. For instance, here our prior on CoE is 0.71 which implies a variance of 0.2. If we gather data on M however our posterior variance will be either 0.18 (with probability 0.93) or 0.

Thus we see that we can already imagine how our a design in which we seek data on M only will perform in expectation in terms of reducing our uncertainty. No simulation is required. Even still, we think it useful to fold these quantities

Table 15.2: Inference upon observation of $*M*$ or W for cases in which $X = 1$ and $Y = 1$. We see that expected posterior variance (equivalently, expected mean squared error) falls modestly when $*M*$ is observed but substantially when W is observed. The gains from observing W are modest if $*M*$ is already observed.

Inquiry	Estimator	N Sims	Bias	Mse	Mean Posterior Var	Mean Estimate	Mean Estimand
CoE	Prior	1000	-0.01 (0.01)	0.20 (0.01)	0.20 (0.00)	0.71 (0.00)	0.72 (0.01)
CoE	M only	1000	-0.01 (0.01)	0.16 (0.01)	0.17 (0.00)	0.72 (0.01)	0.72 (0.01)
CoE	W only	1000	-0.00 (0.01)	0.06 (0.00)	0.06 (0.00)	0.72 (0.01)	0.72 (0.01)
CoE	Both	1000	-0.00 (0.01)	0.05 (0.00)	0.05 (0.00)	0.72 (0.01)	0.72 (0.01)

into a design declaration so that users can access the data strategies and answer strategies in the same way as they would for any other problem.

Declaration 15.1 uses a custom data function that draws a value of the estimand (EoC) using the prior in the first row of Table 15.1 and then draws values of M , using values from the second row of Table 15.1 (and similarly for W). The estimation step also uses a custom function, which simply returns the posterior estimates—like those in the last rows of Table 15.1, but for both M and W . No data strategy is provided because we imagine estimation given all possible data strategies (observation of M , of W , and of both).

Declaration 15.1.

```
design <-
  declare_model(data = data_function()) +
  declare_inquiry(CoE = CoE) +
  declare_estimator(handler = my_estimator_function)
```

Given such a model, a case in which $X = Y = 1$, and limited resources, we now want to know whether we would be better gathering data on M or on W or both? The answers are given in Table 15.2. We see only modest declines in expected posterior variance from observation of the mediator M (consistent with the manual calculation above), but large declines from observing W .

Note here that the causal model used for M and I is the same model used for A

(Principle 3.7). This does not *have* to be the case however. If instead we used different models for these two parts we could assess how well or poorly our strategy performs when our model is wrong. In this case, and unlike the results in table 15.2, we would find that the expected posterior variance (where the expectation is taken with respect to the model in M but posterior taken with respect to the model in A) will not be the same as the expected mean squared error.

15.2 Selection-on-observables

We declare a design in which a researcher tries to address concerns about confounding by conditioning on observable third variables. In the design the researcher has to specify how a third variable creates a risk of confounding and how exactly they will take account of these variables to minimize bias.

When we want to learn about causal effects – but treatments are allocated by the world and not by the researcher – we are sometimes stuck. It's possible that a comparison of treated units to control units will generate biased inferences because of selection. If selection puts units with higher average potential outcomes into the treatment group, then a comparison with the control group will yield biased causal inferences.

Sometimes, however, we know enough about selection in order to condition on the variables that cause it. A selection-on-observables design stipulates a family of models M of the world that describe which variables are responsible for selection, then employs a data strategy that measures those variables, rendering them “observable”. In the answer strategy, we draw on substantive knowledge of the causal process to generate an “adjustment set”: a set of variables that predict selection into treatment, or in the language of DAGs, a set that when conditioned upon, closes all back door paths from the treatment to the outcome. We can condition in many ways, for example through regression adjustment, stratification, or matching.

The quality of causal inferences we draw comes down to whether our claims about selection into treatment are correct. If we've missed a cause (missed a back door path), then our inferences will be prone to bias.

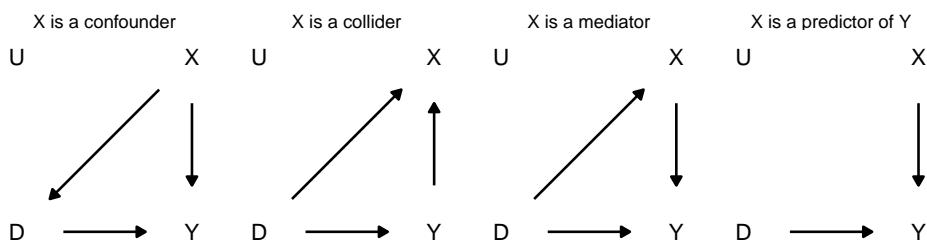


Figure 15.1: Three roles for an observable variable X.

However, it is not as simple as adjusting for any variable that will close back paths. We face problems if we fail to adjust for an observable X under some models — but we will also face problems if we *do* adjust for X under other models. In Figure 15.1 we illustrate four of the possible roles for an observable variable X : as a confounder of the relationship between D and Y ; as a collider, which is affected by both D and Y ; as a mediator of the relationship between D and Y ; and as a predictor of Y with no connection to D . We set aside in these DAGs the possible roles of an unobservable variable U that would introduce additional problems of confounding.

If X is a confounder, failing to adjust for it in studying the relationship between the treatment D and outcome Y will lead to confounding bias. We often think of fourth DAG as the alternative to this, where X is a predictor of Y but has no link to D . In this circumstance, we still want to adjust for X to seek efficiency improvements by soaking up additional variation in Y , but failing to do so will not introduce bias. If the true model is definitely one of these two, first and fourth DAGs, we should clearly choose to adjust for X to ensure we avoid confounding bias in case the first is correct (and we will do no worse if instead it is the fourth).

However, the middle two DAGs present problems if we *do* adjust for X . In the first, X is a collider: it is affected by both D and Y . Adjusting for X if this is the true model introduces collider bias, because we open a backdoor path between D and Y through X . We also introduce bias if we control for X if the mediator model (DAG 3) is the true model, wherein D affects X and Y and X affects Y (i.e., X is a mediator for the relationship between D and Y). But the reason here is different: controlling for a mediator adjusts away part of the true treatment effect.

In a selection-on-observables design, we must get many features of the model correct, not only about the factors that affect D . We must be sure of all the arrows into X , D , and Y and the order of the causal arrows. In some cases, in natural experiments where selection processes are not randomized by researchers but are known, these assumptions can be sustained. In others, we will be making heroic assumptions.

We declare a design to explore these considerations, with a model defining the relationship between a causal factor of interest D and outcome Y and an observable confounder X , the average treatment effect as the inquiry, a simple measurement strategy, and two estimators with and without adjustment for X . We use exact matching as our adjustment strategy.

Declaration 15.2.

```
exact_matching <-
  function(data) {
```

```

matched <- matchit(D ~ X, method = "exact", data = data)
match.data(matched)
}

design <-
declare_model(
  N = 100,
  U = rnorm(N),
  X = rbinom(N, 1, prob = 0.5),
  D = rbinom(N, 1, prob = 0.25 + 0.5 * X),
  Y_D_0 = 0.2 * X + U,
  Y_D_1 = Y_D_0 + 0.5
) +
declare_inquiry(ATE = mean(Y_D_1 - Y_D_0)) +
declare_step(handler = exact_matching) +
declare_measurement(Y = reveal_outcomes(Y ~ D)) +
declare_estimator(Y ~ D,
  weights = weights,
  model = difference_in_means,
  label = "adjusted") +
declare_estimator(Y ~ D,
  model = difference_in_means,
  label = "unadjusted")

```

We declare beliefs about the selection process and how D , Y , and X are related. The model needs to include potential outcomes for the main outcome of interest (Y) and a specification of the assignment of the key causal variable (here, D). Here, we have defined the assignment process as a function of an observable variable X . In fact, X is the only variable that affects selection into treatment: X is a binary variable (i.e., two groups), and the probability of treatment is 0.4 when $X = 0$ and 0.6 when $X = 1$. In addition, we define the potential outcomes for Y , which invoke confounding by X because it affects both D and Y . We only invoke one possible relationship between X , Y , and D , and so do not consider the possibilities of colliders or mediators.

In our model U is not a confounder in this model because it affects Y but not D ; this is a strong excludability assumption on which our causal inferences depend. The assumption is strong because ruling out all unobservable confounders is typically impossible. Most causal factors in nature are affected by many variables, only some of which we can imagine and measure. The problem with unobserved confounding is bias. In the design declared above, the unadjusted answer strategy suffers from unobserved confounding, because we do not ad-

just for X which predicts both treatment and the outcome. Depending on the inquiry and your diagnosands, a biased answer strategy may be good enough — or it may be all you can get. If the decision you face is whether to adopt a treatment, you may simply want to know if the treatment is at least 10% effective, even if you can't get a precise estimate of its effectiveness. With this goal in mind, a little bias that still allows you to correctly determine if the treatment effect is at least 0.1 may be okay.

Sensitivity analysis tackles this idea of *how much* bias is a problem. We can't adjust for unobservable confounders, but we can reason about how much confounding would have to be present to draw the wrong conclusions from our data. How much confounding would there have to be to decide incorrectly that a treatment is at least 10% effective is a question sensitivity analysis may be able to answer. An answer sensitivity analysis could provide is that you would have to have a variable with a 20% correlation with the outcome and a 5% correlation with treatment to make the wrong decision. The substantive or theoretical judgement then about whether that amount of confounding is likely, takes us back to the model and whether we think that is plausible. We explore one implementation of the idea of sensitivity analysis in the exercises.

In addition to the model, we define our inquiry as the average treatment effect of D on Y , a measurement strategy to collect observations on Y , and two possible estimators with and without adjustment of X . The first estimator, with adjustment, uses the weights from the exact matching estimator. The matching procedure adjusts for differences in the probability of selection into treatment according to X . The second, unadjusted, estimator does not control for X . We see in a diagnosis of the bias of the design for these two estimators that the first has no bias and the second substantial bias (about 12% of the average estimate).

We see in a diagnosis of the design that if we fail to adjust for the observable features of the selection process — that the probability of treatment depends on the value of the observable characteristic X — then we have biased answers. If we do control, we obtain unbiased answers. However, as highlighted earlier, these results depend on the believability of our model, which in this case involves observed confounding of the relationship between D and Y by X . If there was *unobservable* confounding from U , which we ruled out, or if X was a collider of mediator, there would be bias.

```
diagnosis <- diagnose_design(design, sims = sims, bootstrap_sims = b_sims)
```

Exercises

In this exercise, we'll investigate how to diagnose an observation research design that includes sensitivity as part of the answer strategy. Researchers use sensitivity analysis in settings when treatment effect estimates may be biased

Table 15.3: Bias and RMSE for two estimators from a selection-on-observables design

estimator	bias
adjusted	0.009
unadjusted	0.113

due to *unobserved* confounding. In such cases, we can't condition on the variables that would deconfound the treatment effect estimates, because we don't have access to them. In such cases, we turn to sensitivity analyses which essentially ask, "how bad would the unobserved confounding have to be" to overturn the substantive conclusions of the paper.

To diagnose sensitivity analysis, we'll need to make use of a helper function:

```
library(sensemakr)
sensitivity_helper <-
  function(model) {
    sensitivity <- sensemakr(model = model, treatment = "Z")
    sensitivity$sensitivity_stats[,c("estimate", "rv_qa")]
  }
```

Here's a design:

```
design <-
  declare_population(N = 1000,
                     U = rnorm(N, sd = 0.6),
                     X = correlate(rnorm, given = U, rho = 0.5),
                     Z = rbinom(N, 1, prob = pnorm(X)),
                     Y = 0.2 * Z + 0.5 * X + U) +
  declare_estimator(Y ~ Z + X, model = lm, model_summary = sensitivity_helper)
```

1. Inspect the `declare_population` call. What is the true average treatment effect on Z on Y?
2. Draw data from this design. Run a regression of Y on Z and X, using the `lm` function. Conduct a sensitivity analysis using the `sensemakr` function from the `sensemakr` package. Interpret the "Robustness Value, q = 1, alpha = 0.05" statistic. (Hint: run `summary` on the output from the `sensemakr` function)
3. Diagnose the design with respect to bias relative to the true average treat-

ment effect. In this design, is a regression of Y on Z , controlling for X unbiased for the true ATE? Why or why not?

4. Now diagnose the design with respect to the robustness value. How frequently do we obtain a robustness value less than 0.10? less than 0.15?
5. Recall that in this design, the treatment and outcome are *not* confounded by unobservables. In this light, interpret the diagnosis of the sensitivity analysis.

15.3 Difference-in-differences

We declare a differences in differences design in which the effect of a cause is assessed by comparing changes over time for a unit that gets treated to changes over time for a unit that does not get treated. The declaration and diagnosis helps clarify when effect heterogeneity threatens inference using this approach.

The difference-in-difference design does not rely on the assumption of a known assignment mechanism as in selection-on-observables. But it adds a new one instead: the parallel trends assumption. In order for the difference-in-difference design to work, the *change* between before and after treatment in the control potential outcomes must be equal (i.e., parallel). Because this assumption depends on the change in values of the unrealized (and thus unobservable) control potential outcome in the treated unit after treatment, it cannot be tested. A widely-used diagnostic is to look at the trends in outcomes between the treated and control unit *before* treatment; this is only an indirect test because parallel trends concerns the unobserved control trend of the actually treated unit.

The design has been famously used for analyses of two periods (before and after) and two groups (treated and untreated) such as a policy change in one state compared to another before and after the policy change. Today, it is most often used in many-period many-group settings with observational panel data. Here, the logic of the two-period two-group design is extended through analogy. Parallel trends between treated and control groups are assumed *on average* across treated groups and periods. Unfortunately, the analogy holds only under limited circumstances, a fact only recently discovered.

We declare a design for a 20-period, 20-unit design in which units are become treated at different times, a common setting in empirical social science often referred to as the staggered adoption design. The treatment effect of interest might be a state-level policy adopted in 20 states at some point within a 20-year period, so we draw on comparisons before and after policy adoption within states and across states that have and have not yet adopted treatment.

Declaration 15.3.

```

N_units <- 20
N_time_periods <- 20

design <-
  declare_population(
    units = add_level(
      N = N_units,
      U_unit = rnorm(N),
      D_unit = if_else(U_unit > median(U_unit), 1, 0),
      D_time = sample(1:N_time_periods, N, replace = TRUE)
    ),
    periods = add_level(
      N = N_time_periods,
      U_time = rnorm(N),
      nest = FALSE
    ),
    unit_period = cross_levels(
      by = join(units, periods),
      U = rnorm(N),
      potential_outcomes(
        Y ~ U + U_unit + U_time +
          D * (0.2 - 3 * (D_time - as.numeric(periods))),
        conditions = list(D = c(0, 1))
      ),
      D = if_else(D_unit == 1 & as.numeric(periods) >= D_time, 1, 0),
      D_lag = lag_by_group(D, groups = units, n = 1, order_by = periods)
    )
  ) +
  declare_inquiry(
    ATT_switchers = mean(Y_D_1 - Y_D_0),
    subset = D == 1 & D_lag == 0 & !is.na(D_lag)
  ) +
  declare_measurement(Y = reveal_outcomes(Y ~ D)) +
  declare_estimator(
    Y ~ D, fixed_effects = ~ units + periods,
    model = lm_robust,
    inquiry = "ATT_switchers",
    label = "2wfe"
  ) +
  declare_estimator(
    Y = "Y",
    G = "units",
    T = "periods",
  )

```

```

D = "D",
handler = label_estimator(did_multiplegt_tidy),
inquiry = "ATT_switchers",
label = "chaisemartin"
)

```

We define hierarchical data with time periods nested within groups, such that each of the 20 units have 20 time periods from 1 to 20. We assign units to be treated at some point in the period (D_{unit}), and confound treatment assignment with unobservable unit-specific features U_{unit} . (If there was no confounding, we would not need the parallel pretrends assumption. This would be guaranteed by virtue of the randomization.) In addition, we assign the timing of treatment is randomly assigned (D_{time}), such that for treated units treatment turns on the randomly assigned time. D then is jointly determined by whether the unit is treated and whether the current period is after the assigned D_{time} . We allow for unit-specific variation U_{unit} and time-specific variation U_{time} that affects the outcome as well as unit-period characteristics U . Potential outcomes are a function of these unit-, time-, and unit-time-specific characteristics, and a treatment effect of 0.2 that varies according to when units are treated (more on the importance of this treatment effect heterogeneity below).

The difference-in-difference design lends itself to the average treatment effect on the treated (ATT) inquiry. We are seeking counterfactual comparisons for each treated unit in untreated units, whether those that are never treated or those not yet treated. We leverage over time comparisons within units to isolate the causal effect of treatment, and net out time-varying characteristics by subtracting off the change in untreated units. Unfortunately, except under extremely limited circumstances — exactly homogenous treatment effects — we will not be able to recover unbiased estimates of the ATT. We can, however, under some circumstances and with some estimators, recover the ATT for a specific subgroup: units that just switched from untreated to treated. We declare the ATT for these “switchers” as the inquiry.

We measure the outcome Y , and define two answer strategies. First, we define the standard two-way fixed effects estimator with fixed effects by time and unit. The two-way fixed effects fits the empirical goal of difference-in-differences: the time fixed effects net out time-varying unit-invariant variables such as seasonality and time trends. The unit fixed effects net out unit-varying variables that are time-invariant like race or age-at-birth of individuals and longterm histories of violence of territories. Second, we define the newly-defined De Chaisemartin and d'Haultfoeuille (2020) estimator, which addresses bias in the staggered adoption difference-in-differences design. The De Chaisemartin and d'Haultfoeuille (2020) design addresses a problem in standard analyses

of the staggered adoption difference-in-difference design: the two-way fixed effects estimator relies on comparisons between units that are treated and units that are *not yet* treated. When treatment effects differ across units depending on when they are treated (as they do in the design here), then those comparisons will be biased: part of the treatment effect will be subtracted out of the estimate.

15.3.1 Treatment timing

The first important design parameter is the pattern of treatment timing across units. Three patterns are common: all units are treated at one common point in time, and typically stay treated once treated; the staggered design as in the declaration above, in which units are treated for the first time at different times and once treated stay treated; and on-off patterns in which units are treated at different times and once treated become untreated again.

We declare these three possibilities as three different treatment variables D:

```
declare_model(
  D_sametime =
    if_else(D_unit == 1 & as.numeric(periods) >= N_time_periods/2, 1, 0)

  D_staggered =
    if_else(D_unit == 1 & as.numeric(periods) >= D_time, 1, 0)

  D_one_period =
    if_else(D_unit == 1 & as.numeric(periods) == D_time, 1, 0)
)
```

The proportion of treated units that are treated at different points is also consequential, so in addition to specifying which of these general patterns holds we need to either get the specific timing pattern of treatment right or explore how the design fares under different possible patterns. For example, if half of units are treated within the first three periods and the other half in the last three periods of 20, the properties of the design may differ from a design in which the second half is treated in the fourth through sixth periods. In our staggered design above, we declared a uniform distribution of timing with `D_time = sample(1:N_time_periods, N, replace = TRUE)`. When you know details of timing in advance, bring in that data directly to the declaration instead.

15.3.2 Treatment effect heterogeneity

Whether treatment effects vary across units and across time affects whether two-way fixed effects designs can recover treatment effects. We started with homogenous treatment effects in the declaration above. In this case, as long as a

generalized form of parallel trends holds, the timing of treatment and other details of the potential outcomes will not affect whether you can recover unbiased estimates of the ATT for switchers.

When treatment effects differ depending on when units are treated, however, problems emerge for many treatment timing profiles. We declare homogenous treatment effects below, as well as two kinds of time-varying heterogeneous treatment effects where effects get lower for units treated later ($Y_{\text{later_lower}}$) and where effects get higher for units treated later ($Y_{\text{later_higher}}$).

```
declare_model(
    potential_outcomes(
        Y_homogenous ~ U + U_unit + U_time + D * 0.2,
        conditions = list(D = c(0, 1))
    ),
    potential_outcomes(
        Y_later_lower ~ U + U_unit + U_time +
            D * (0.2 + 0.5 * (D_time - as.numeric(periods))),
        conditions = list(D = c(0, 1))
    )
    potential_outcomes(
        Y_later_higher ~ U + U_unit + U_time +
            D * (0.2 - 0.5 * (D_time - as.numeric(periods))),
        conditions = list(D = c(0, 1))
    )
)
```

To follow the Principle 3.3 “entertain many models” for a difference-in-differences design, we assess the performance of the design under all three possible kinds of heterogeneity. With some treatment profiles and some answer strategies proposed in recent years, unbiased estimates will be possible, but not with other combinations. The timing of treatment, nature of heterogeneous effects, and answer strategies interact to determine the properties of the design.

15.3.3 Answer strategies

So many answer strategies have been proposed in recent years to address bias in the difference-in-differences design that we cannot summarize them in this short entry. Instead, we illustrate how assess the properties of just one estimator under a range of conditions. We point in further reading to references for the alternative answer strategies, both estimation procedures and diagnostic tools.

In the declaration, we define the units as in the first declaration and adopt the same staggered treatment allocation with uniform timing, add all three sets of potential outcomes declared above (homogenous, increasing over time, and

decreasing over time), and examine both the ATT and the ATT for switchers inquiries. We consider both the two-way fixed effects estimator and the De Chaisemartin and d'Haultfoeuille (2020) estimator.

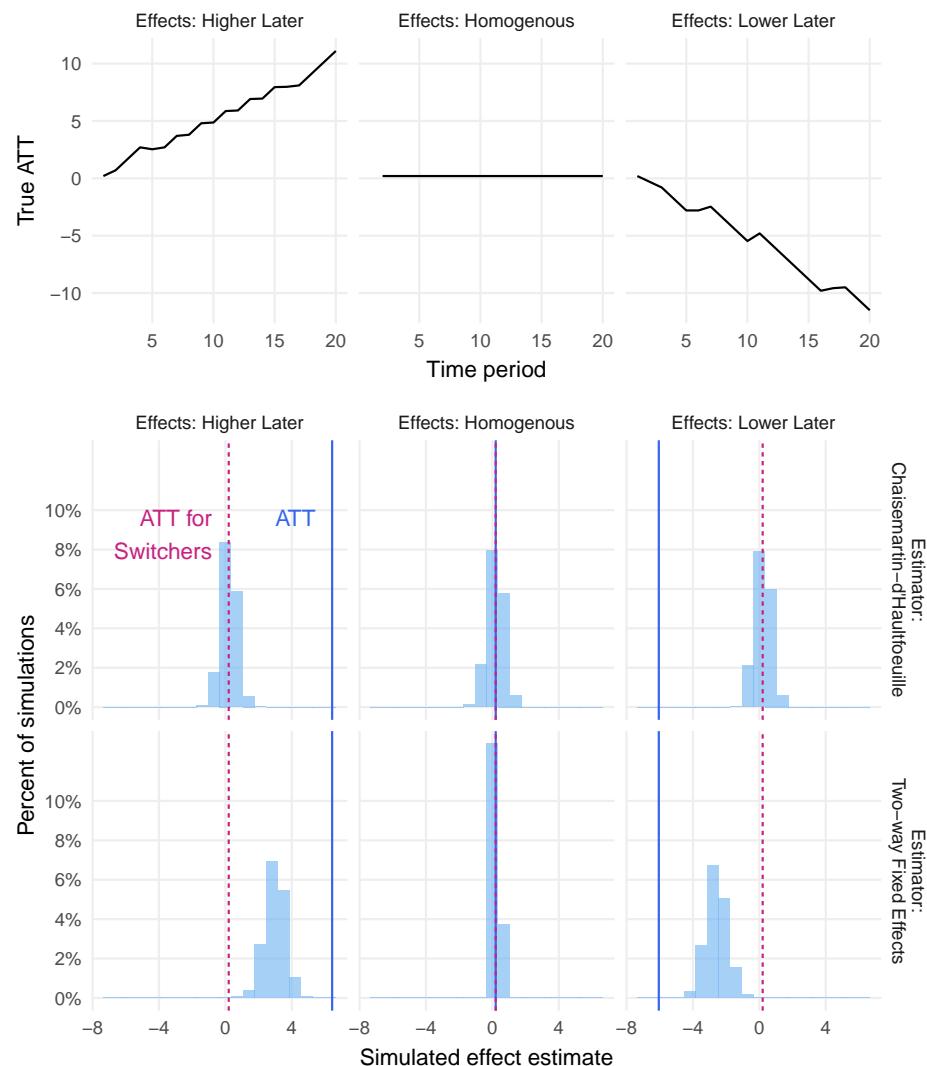


Figure 15.2: Diagnoses of two generalized difference-in-differences estimators under three types of potential outcomes.

15.4 Instrumental variables

We declare a design in which a researcher addresses confounding concerns by using an instrumental variable. Under the right conditions, the approach can generate unbiased estimates for treatment effects for units whose treatment status is sensitive to the instrument.

When we cannot credibly assert that we have controlled for all confounding variables in a selection-on-observables design (see Section 15.2) — blocked all back-door paths between a treatment D and an outcome Y — we might have to give up on our goal of drawing causal inferences.

But occasionally, the world yields an opportunity to sidestep unobserved confounding by generating as-if random variation in a variable that itself affects the treatment variable. We call the variable that is as-if randomly assigned by the world an “instrument.”

Instruments are special. They are variables that are randomly assigned by nature, the government, or other individuals or groups. Usually, genuine random assignments have to be crafted by researchers as part of a deliberate data strategy. Experimenters go to great lengths to randomly expose some units but not others to treatments. When the world provides bonafide random variation in a variable of interest, it’s a rare and valuable opportunity. By virtue of as-if random assignment, we can learn about the average causal effects of the instrument itself without any further consternation. Conditional on geography and season, weather conditions are as-if randomly assigned, so we can learn about the average effects of rainfall on many outcomes, like crop yields, voter turnout, and attendance at sporting events. Conditional on gender and birth year, draft numbers are randomly assigned by the government, so we can learn about the average effects of being drafted on educational attainment, future earnings, or public policy preferences.

We illustrate the logic of instrumental variables in the DAG below. An instrument Z affects an endogenous treatment D which subsequently affects an outcome Y . Many other common causes, summarized in terms of unknown heterogeneity U , affect both D and Y . It is because of this unmeasured confounding that we cannot learn about the effect of D on Y directly using standard tools.

We can naturally imagine an inquiry that represents the effect of the instrument (Z) on an outcome of interest such as the effect of the draft number on future earnings. In the terminology of instrumental variables, this inquiry is called the “reduced form” or the “intention-to-treat” (ITT) effect. If it’s really true that the instrument is randomly assigned by the world, estimation of the reduced form effect is straightforward: we estimate the causal effect of Z on Y using, for example, a difference-in-means estimator. Sometimes, we are also interested in the “first-stage” effect of the instrument on the treatment variable. Similarly, we can target this inquiry by studying the causal effect of Z on D , for example using the difference-in-means estimator again.

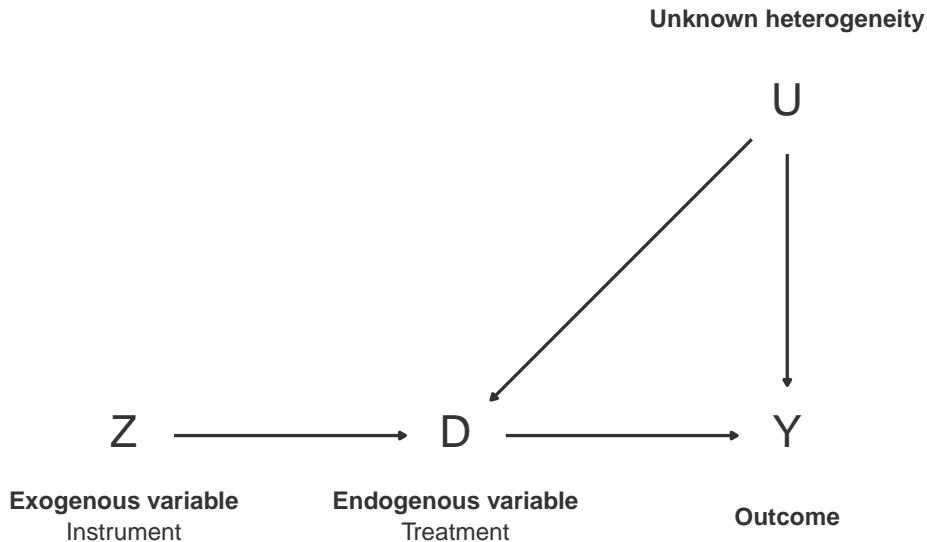


Figure 15.3: Directed acyclic graph (DAG) of an instrumental variables design.

The trouble comes when we want to leverage the random assignment of the instrument (Z) to learn about the average causal effects of the treatment variable (D) on the outcome (Y), where D is not randomly assigned but rather affected by Z and also other variables. To do so, we need to understand better how the instrument affects whether you are *treated* and then how both affect the outcome of interest. We need to understand “compliance” with the instrument.

We can define the compliance types of units in terms of the combinations of values of the potential outcomes of D in terms of values of instrument Z takes on. The combinations are made up of the value of treatment if unit i is assigned to control ($D_i(Z_i = 0)$) and the value if it is assigned to treatment ($D_i(Z_i = 1)$). For a binary instrument and binary treatment, there are four possible potential outcomes values and thus four types, enumerated in Table 15.4. These compliance types are often known as “principal strata” (Frangakis and Rubin, 2002).

Table 15.4: Compliance types

Type	$D_i(Z_i = 0)$	$D_i(Z_i = 1)$
Never-taker	0	0
Complier	0	1
Defier	1	0
Always-taker	1	1

Never-takers are those who never take the treatment, no matter what the value of the instrument. Compliers take exactly the value of the instrument they are

assigned. Defiers do exactly the opposite. When the instrument is take the treatment, they refuse it, but when assigned to not take it, they take treatment. Like their name suggests, always-takers take the treatment regardless of the instrument condition nature assigns them to.

With these types in mind, we can now define a new inquiry that we will be able to target with instrumental variables under a special set of assumptions: a “local” average treatment effect (LATE) among compliers. The concept of “local” reflects the idea that the effect applies only to the specific group of units whose treatment status changes as a result of the instrument.² The LATE is $E[Y_i(D_i = 1) - Y_i(D_i = 0)|D_i(Z_i = 1) > D_i(Z_i = 0)]$ – the average treatment effect among the group of people whose value of the treatment is higher as a result of the treatment. The LATE is different from the ATE because it does not average over those units whose value of the treatment does not depend on the instrument.

We can adopt the instrumental variables answer strategy — using two-stage least squares for example — to estimate the LATE. But to do so without bias, we need to invoke new assumptions on top of those for randomized experiments. In the case of a binary instrument and a binary treatment, the five assumptions are:

1. Exogeneity of the instrument: $Y_i(D_i = 1), Y_i(D_i = 0), D_i(Z_i = 1), D_i(Z_i = 0) \perp\!\!\!\perp Z_i | X_i$. Substantively, this assumption requires that (possibly conditional on observed covariates in X) the instrument is as-if randomly assigned, so it is jointly independent of the treated and untreated potential outcomes as well as the potential values the treatment variable D would take on depending on the values of the instrument Z . Exogeneity is usually justified on the basis of qualitative knowledge of why as-if random assignment is a reasonable assumption to make. The assumption can be bolstered — but not directly tested — with design checks like the balance on pre-treatment values of the covariates according to the levels of the instrument.
2. Excludability of the instrument: $Y_i(D_i(Z_i), Z_i) = Y_i(D_i(Z_i))$. We can “exclude” the instrument from the potential outcomes function $Y_i()$ — the only relevant argument is the value of the treatment variable. Substantively, this means that Z has exactly no effect on Y except by changing the value of D , or in other words a “total mediation assumption.” Under the exclusion restriction, the effect of the instrumental variable is wholly mediated by the treatment variable. The validity of the exclusion restriction cannot be demonstrated empirically and typically must be asserted on qualitative grounds. Since the reduced form of the instrument can be estimated for many different outcome variables, one piece of evidence that

²There are many local average treatment effects that are adopted as inquiries by empirical researchers. The instrumental variables is particularly common, so adopts the general term for its specific use. Often researchers refer specifically in the case of IV to the complier average causal effect.

can bolster the exclusion restriction is to show that the instrument does not affect other plausible causal precedents of the outcome variable. If it does affect other variables that might, in turn, affect the outcome, doubt may be cast on the exclusion restriction.

3. Non-interference: $Y(D_i(Z_i), D_i, Z_i) = Y(D_i(Z_i))$. Like any non-interference assumption, here we assert that for any particular unit, other units' values of the instrument or the treatment do not affect the outcome.
4. Monotonicity: $D_i(Z_i = 1) \geq D_i(Z_i = 0)$, i.e. This assumption states that the effect of the instrument on the treatment is either zero or is positive for all units. Monotonicity rules the defier complier type who would have $D = 1$ if $Z = 0$ but would have $D = 0$ if $Z = 1$. Monotonicity is usually quite plausible (it's tough to imagine a person who would serve in the military if not drafted but *would* serve if drafted!), but it's not possible to affirm empirically. An empirical test that demonstrates a positive effect of the instrument on the treatment for one group but a negative effect for a different group could, however, falsify the monotonicity assumption.
5. Non-zero effect of the instrument on the treatment. If the instrument does not affect the treatment, then it is useless for learning about the effects of the treatment on the outcome, simply because it generates no compliers. If there are no compliers, the LATE itself is undefined.

If all five of these assumptions are met, it can be shown that the inquiry $\text{LATE} = \frac{\text{ATE}_{ZY}}{\text{ATE}_{ZD}} = \frac{\text{Reduced Form}}{\text{First Stage}}$. It is the quotient of the causal effect of the instrument on the outcome divided by the causal effect of the instrument on the received treatment. This expression underlines the importance of assumption 5: if the instrument doesn't affect the treatment, the first stage is equal to zero and the ratio is undefined.

Going back to the DAG above, many of the five assumptions is represented in the DAG. The exogeneity of the instrument is represented by the exclusion of a causal effect of U on Z . The omission of an arrow from Z to Y directly invokes the exclusion restriction. Non-interference is typically not directly represented in DAGs. Monotonicity and the non zero effect of the instrument on the treatment is represented by the causal arrow from Z to D .

Moving to answer strategy, a plug-in estimator of the LATE is the difference-in-means of the outcome according to the instrument divided by the difference-in-means of the treatment according to the instrument. Equivalently, we can use two-stage least squares, which will yield the identical answer as the ratio of the difference-in-means estimates when no covariates are included in the regression.

With the four elements of the design in hand, we declare the model in a simple form below, invoking each of the five assumptions in doing so:

Declaration 15.4.

```

design <-
  declare_model(
    N = 100,
    U = rnorm(N),
    potential_outcomes(D ~ if_else(Z + U > 0, 1, 0),
                        conditions = list(Z = c(0, 1))),
    potential_outcomes(Y ~ 0.1 * D + 0.25 + U,
                        conditions = list(D = c(0, 1))),
    complier = D_Z_1 == 1 & D_Z_0 == 0
  ) +
  declare_inquiry(LATE = mean(Y_D_1 - Y_D_0), subset = complier == TRUE) +
  declare_assignment(Z = complete_ra(N, prob = 0.5)) +
  declare_measurement(D = reveal_outcomes(D, Z),
                      Y = reveal_outcomes(Y, D)) +
  declare_estimator(Y ~ D | Z, model = iv_robust, inquiry = "LATE")

```

The exclusion restriction is invoked in omitting a causal direct effect of Z in the Y potential outcomes. We invoke the monotonicity and non-zero effect of Z on D in the D potential outcomes, which have an effect of Z of 1. We invoke the non-interference assumption by excluding effects of the instrument values from other units in the definition of the D and Y potential outcomes. And we invoke the ignorability of Z by randomly assigning it in the assignment step.

Many instrumental variables designs involve historical data, such that most remaining design choices are in the answer strategy. Declaring and diagnosing the design can yield insights about how to construct standard errors and confidence intervals and the implications of analysis procedures in which data-dependent tests are run before fitting instrumental variables models — and only fit if the tests pass. But other instrumental variables papers involve prospective data collection, in which an instrument is identified and outcome data (and possibly instrument and treatment data) are collected anew. In such settings, the tools design diagnosis and redesign are useful just as in any prospective research design, to help select sampling and measurement procedures from the sample size to the number of outcomes to be collected. The key in this case is to build in the features of the instrument, the potential outcomes of treatment receipt, and the potential outcomes of the ultimate outcome of interest and to explore violations of the five assumptions in the model.

The instrumental variables setup is perfectly analogous to a randomized experiment with noncompliance (see Section 17.6). The instrument is equivalent to the random *assignment*. If some units do not comply with their assignment (some in the treatment group don't take treatment or some in the control group

do take treatment), then a comparison of groups according to the treatment variable will be biased by unobserved confounding. The best we can do under noncompliance is to redefine the estimand to be the complier average causal effect (equivalent to the LATE), then estimate it via two-stage least squares. The required excludability assumption is that the assignment to treatment can't affect the outcome except through the realized treatment variable, which may or may not hold in a given experimental setting.

15.5 Regression discontinuity designs

We declare a design in which the assignment of a treatment is determined by whether a unit exceeds an arbitrary threshold. We demonstrate through diagnosis the bias-variance tradeoff at the heart of the choice of answer strategies that target the local average treatment effect inquiry, defined right at the cutoff.

Regression discontinuity designs exploit substantive knowledge that treatment is assigned in a particular way: everyone above a threshold is assigned to treatment and everyone below it is not. Even though researchers do not control the assignment, substantive knowledge about the threshold serves as a basis for a strong causal identification claim.

Thistlewhite and Campbell introduced the regression discontinuity design in the 1960s to study the impact of scholarships on academic success. Their insight was that students with a test score just above a scholarship cutoff were plausibly comparable to students whose scores were just below the cutoff, so any differences in future academic success could be attributed to the scholarship itself.

Just like instrumental variables, regression discontinuity designs identify a *local* average treatment effect: in this case, the average effect of treatment *exactly at the cutoff*. The main trouble with the design is that there is vanishingly little data exactly at the cutoff, so any answer strategy needs to use data that is some distance away from the cutoff. The further away from the cutoff we move, the larger the threat of bias.

Regression discontinuity designs have four components: A running variable X , a cutoff, a treatment variable D , and an outcome Y . The cutoff determines which units are treated depending on the value of the running variable. The running variable might be the Democratic party's margin of victory at time $t - 1$; and the treatment, D , might be whether the Democratic party won the election in time $t - 1$. The outcome, Y , might be the Democratic vote margin at time t .

A major assumption required for regression discontinuity is that the conditional expectation functions — a function that defines the expected value of the outcome at every level of the running variable — for both treatment and control potential outcomes are continuous at the cutoff.³

³An alternative motivation for some designs that do not rely on continuity at the cutoff is “local

The regression discontinuity design is closely related to the selection-on-observables design in that exact knowledge of the assignment mechanism is needed. In Figure 15.4, we illustrate the DAG for the RD design, which includes a treatment with a known assignment mechanism (exactly that units with $X >$ cutoff are treated and those below are not). We highlight that the exogenous variable X , the running variable, is a common cause both of treatment and the outcome, and so must be adjusted for (dashed line) in order to identify the effect of treatment on the outcome and avoid confounding bias. However, confounding bias *remains* far from the cut-off, even after adjustment, because of the possibly different functional forms linking X to Y .

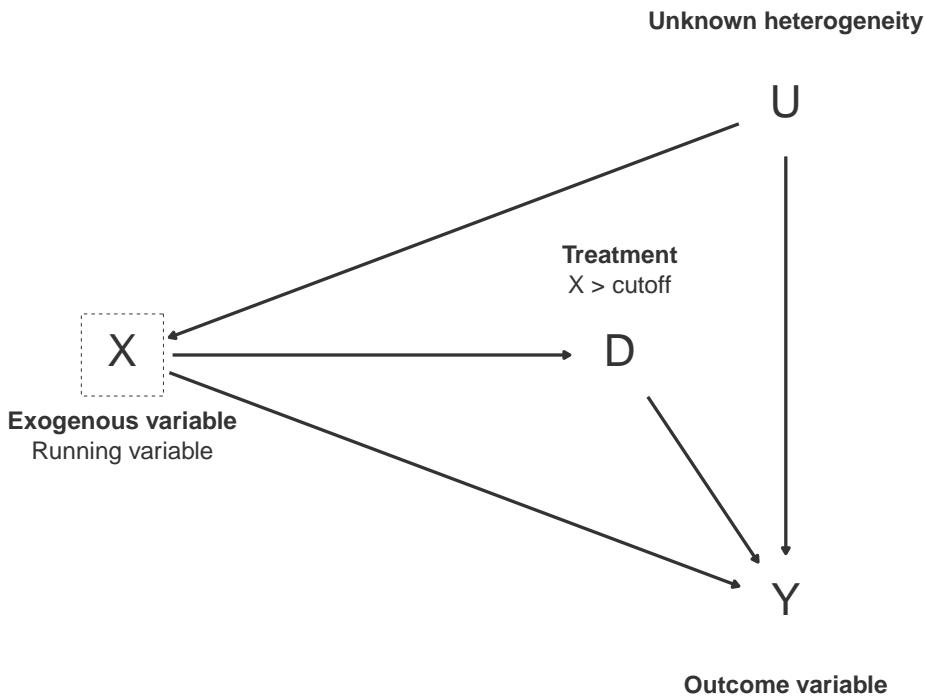


Figure 15.4: Directed acyclic graph (DAG) for the regression discontinuity design.

We illustrate major features of the model for the regression discontinuity using simulated data in Figure 15.5. The raw data is plotted as dots with untreated data (blue dots) to the left of the cutoff (vertical line) and treated data (red dots) to the right of the cutoff. The true conditional expectation functions for both the treated potential outcome and the control potential outcome, which are fourth-order polynomials, are displayed as colored lines, dashed where they are unobservable and solid where they are observable. The discontinuity at the cutoff, above zero, is visible from the raw data alone: the red dots just to the

randomization".

right of the cutoff have higher average values than the blue dots to the left of it. The true treatment effect at the cutoff is illustrated by the dark vertical line, which is the difference between the two polynomials from which the data are drawn. There is a positive treatment effect of about 0.15.

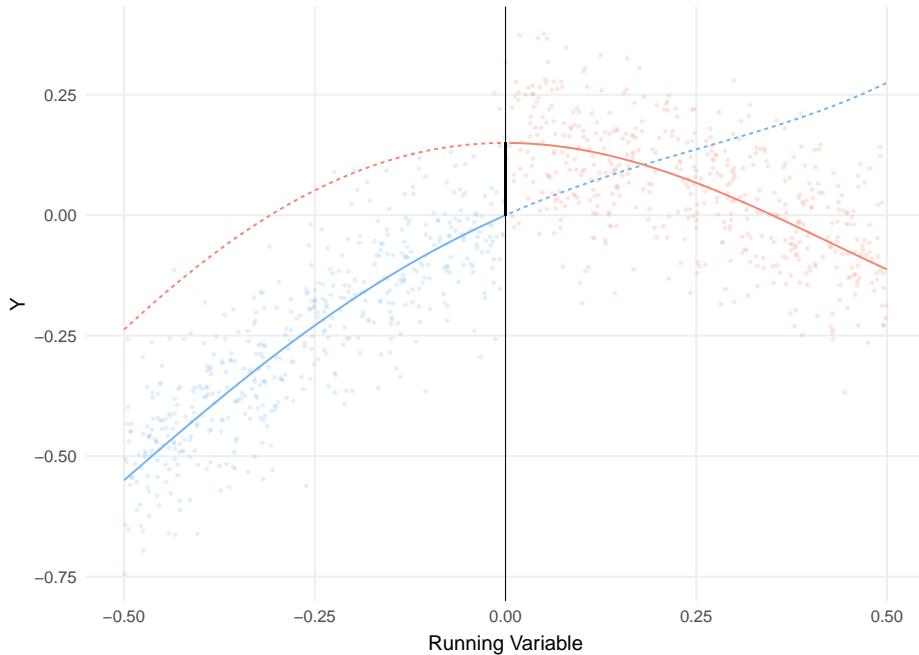


Figure 15.5: Data from a regression discontinuity design. On the x-axis is the running variable, which defines the cut off between control (here, below 0 on the running variable) and treatment (above 0). The outcome Y is on the y-axis. We illustrate smooth potential outcomes by showing the observed data curve (solid) and the counterfactual potential outcome (dashed).

The inquiry in a regression discontinuity design is the effect of the treatment exactly at the cut-off. Formally, it is the difference in the conditional expectation functions of the control and treatment potential outcomes when the running variable is exactly zero. The black vertical line in the plot shows this difference.

The data strategy is to collect the outcome data for all units within a interval (bandwidth) around the cut-off; there must be data above and below. The choice of bandwidth, which we explore further below, involves a tradeoff between bias and variance. The wider we make the bandwidth, the more data we have, and thus the more precision we may be able to achieve. However, the claim of causal identification comes from the comparability of units *just* above and *just* below the threshold. Units far from the cut-off are not likely to be comparable: at a minimum they differ in their values of the running variable and if that is correlated with other unobserved heterogeneity then they differ in

that too. Candidates who *just* win reelection because they get a vote share just over 50%, at 50.01% for example, are likely to be comparable to those who get just below, such as 49.99%, in terms of resources and favorability. Candidates who win in blowout elections with 65% of the vote are likely to have more resources and favorability from those who lose with 35% of the vote. Thus, the more units we compare further from the threshold, the more likely we are to induce bias in our estimates of the treatment effect. However, as we describe now, optimal bandwidth selectors tradeoff bias and variance in the selection of the bandwidth and other parameters, and so collecting data more expansively if inexpensive will still enable subsetting to an appropriate bandwidth at the analysis stage.

Regression discontinuity answer strategies approximate the treated and untreated conditional expectation functions to the left and right of the cutoff. The choice of answer strategy involves two key choices: the bandwidth of data around the threshold that is used for estimation, and the model used to fit that data. We declare the local polynomial regression discontinuity estimator with robust bias-corrected inference procedures described in Calonico, Cattaneo and Titiunik (2014), which fits a nonparametric model to the data and chooses an optimal bandwidth that minimizes bias. This robust estimator is now widely used in practice, because it navigates the bias-variance tradeoff in selecting bandwidths and statistical model specifications we describe below in a data-driven manner and removes the need for researchers to select these parameters themselves.

Declaration 15.5.

```
cutoff <- 0.5
control <- function(X) {
  as.vector(poly(X, 4, raw = TRUE) %*% c(.7, -.8, .5, 1))}
treatment <- function(X) {
  as.vector(poly(X, 4, raw = TRUE) %*% c(0, -1.5, .5, .8)) + .15}

design <-
  declare_model(
    N = 100,
    U = rnorm(N, 0, 0.1),
    X = runif(N, 0, 1) + U - cutoff,
    D = 1 * (X > 0),
    Y_D_0 = control(X) + U,
    Y_D_1 = treatment(X) + U
  ) +
  declare_inquiry(LATE = treatment(0.5) - control(0.5)) +
  declare_measurement(Y = reveal_outcomes(Y ~ D)) +
  declare_estimator(
```

```

Y, X, c = 0, vce = "hc2",
term = "Bias-Corrected",
handler = label_estimator(rdrobust_helper),
inquiry = "LATE",
label = "optimal"
)

```

To illustrate the tradeoffs in choosing bandwidths and statistical models more generally, we consider a broader class of models holding bandwidth fixed and then explore varying the bandwidth.

A simple statistical modeling strategy would be a fully-saturated model with a one-degree polynomial for the running variable interacted with treatment. Controlling for the running variable through the polynomial in addition to the treatment variable is a recognition that units far to the left or far to the right of the cutoff may have different potential outcomes (adjusting closes the back-door path through the running variable). We illustrate data from the regression discontinuity design in Figure 15.6 (top left panel) with a one degree polynomial (linear regression) interacted with treatment, such that there are different slopes and intercepts to the left and right of the cut-off. The next panels in the top of the figure illustrate the same data with three other choices of model fit: a second-, third-, and fourth-degree polynomial. The dark black interval indicates the estimated treatment effect; it is the vertical distance between the predicted value of the treatment polynomial at the cut-off and the predicted value for the control polynomial at the cut-off. We can see that each estimates a slightly different value for the treatment effect: smallest for the one-degree polynomial.

Intuitively, the more degrees in the polynomial, the better the fit with the data. If the data is in fact drawn from a model with a lower-degree polynomial, the higher-order statistical model fit reduces to a lower one (estimating zero for some parameters). However, choosing a higher-order polynomial is less efficient than a lower-degree polynomial. We can see this in the sampling distribution of the estimates from each polynomial fit across draws of the data in the lower panel of Figure 15.6. On the left, the one-degree polynomial model exhibits low variability in estimates; the fourth-degree polynomial estimates are much more variable. However, the center of the fourth-order polynomial is much closer to the estimand (vertical red dashed line) than is the one-degree polynomial, which is highly biased. In short, there is a bias-variance tradeoff in the selection of polynomial: you may improve model fit with higher-order polynomials, but this will introduce variability in the estimates.

The second central choice in a regression discontinuity estimator is the bandwidth: how far from the threshold will data be used to fit the model. The choice

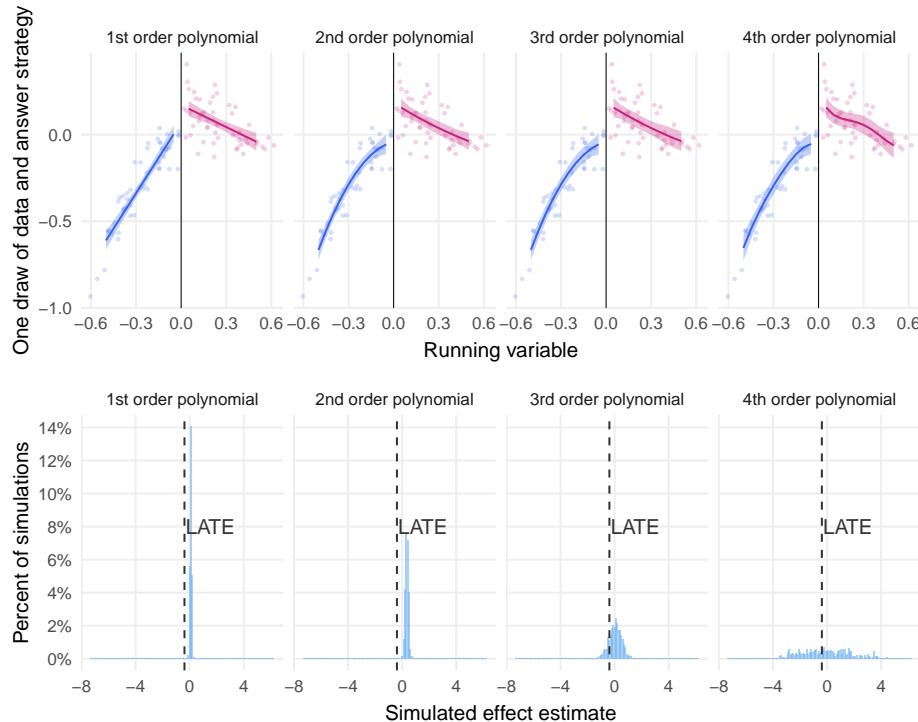


Figure 15.6: Polynomial degree choice.

also amounts to a bias-variance tradeoff: the wider the bandwidth and thus the more data used, the more efficient the estimator is; but the wider the bandwidth, the further from the point at which units are guaranteed to be similar due to the assumption that nothing differs at the limit just above and just below the threshold but the difference in treatment status. In other words, the further away from the threshold, the bigger the differences there are in the potential outcomes from those at the cut-off. With units with very different potential outcomes, the model fit may be biased.

We illustrate in Figure 15.7 this tradeoff in bandwidth selection for a one-degree polynomial (linear interaction). We see the bias-variance tradeoff directly: the bias increases the wider the bandwidth and the more we are relying on data further from the threshold whereas the variance (here, the standard deviation of the estimates) is decreasing the more data we add by widening the bandwidth.

Exercises

1. Gelman and Imbens (2017) point out that higher order polynomial regression specifications lead to extreme regression weights. One approach to obtaining better estimates is to select a bandwidth, h , around the cutoff,

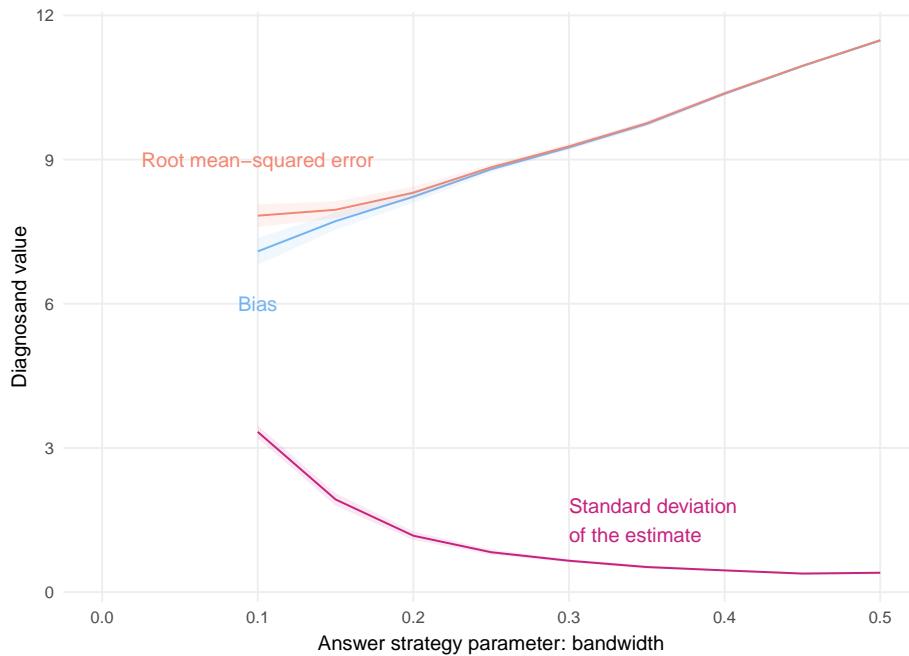


Figure 15.7: Bias-variance tradeoff in bandwidth selection for two regression discontinuity design estimators.

and run a linear regression. Declare a sampling procedure that subsets the data to a bandwidth around the threshold, as well as a first order linear regression specification, and analyze how the power, bias, RMSE, and coverage of the design vary as a function of the bandwidth.

2. The `rdrobust` estimator in the `rdrobust` package implements a local polynomial estimator that automatically selects a bandwidth for the RD analysis and bias-corrected confidence intervals. Declare another estimator using the `rdrobust` function and add it to the design. How does the coverage and bias of this estimator compare to the regression approaches declared above?
3. Reduce the number of polynomial terms of the `treatment()` and `control()` functions and assess how the bias of the design changes as the potential outcomes become increasingly linear as a function of the running variable.
4. Redefine the population function so that units with higher potential outcome are more likely to locate just above the cutoff than below it. Assess whether and how this affects the bias of the design.

Chapter 16

Experimental : descriptive

Why would we ever need to do an *experiment* to do descriptive inference?

Suppose we want to understand the causal model M of a violin. In particular, we have a descriptive inquiry I about the pitch of the highest string, the E string. We want to know if the E string is in tune. Call the latent pitch of the string Y . No matter how hard we listen to the string, we can't hear Y – it is latent. As part of a data strategy D , we could measure the pitch by P plucking it: $Y > Y < P$. This is descriptive research about the causal model M , because the DAG of the violin includes four string nodes which each cause pitch nodes; we'd like to know a descriptive fact about the pitch nodes (at what frequency do they vibrate?).

This question could be recast as a causal inquiry: the untreated potential outcome is the pitch of the unplucked string, as defined by the frequency of vibration. While strings are never *perfectly* still, we can call the untreated potential outcome $Y_i(0) = 0\text{hz}$. The treated potential outcome is the frequency when the string is plucked $Y_i(1) = 650\text{hz}$. The causal effect of plucking the string is $Y_i(1) - Y_i(0) = 650 - 0 = 650$.

Whether framed as a descriptive inquiry or a causal inquiry, we arrive at an answer of 650 hertz. Violinists reading this will know that that means the E string is flat and will need to be tuned up.

Likewise, we can use the assignment of units to conditions to learn about descriptive quantities. Audit experiments estimate the fraction of units that discriminate. List experiments estimate the prevalence rate of a sensitive item. Conjoint experiments measure (aggregations of) preferences over candidates. Experimental behavioral games measure trust. The fact of randomization does not render the inquiry causal any more than the lack of randomization renders the inquiries of observational causal research descriptive.

16.1 Audit experiments

Audit experiments are used to measure discrimination against one group in favor of another group. The design is used commonly in employment settings to measure whether job applications that are otherwise similar but come from candidates from different genders, races, or social backgrounds receive the same rate of job interview invitations. The same approach has been applied to a very wide range of settings, including education, housing, and requests to politicians.

The audit experiment design we'll explore in this chapter has the identical data and answer strategies as the two-arm trial for causal inference described in @ref{sec:p3twoarm}. The difference between an audit study and the typical randomized experiment lies in the model and inquiry. In a two arm trial, a common (causal) inquiry is the average difference between the treated and untreated potential outcomes, the ATE. In an audit experiment, by contrast, the inquiry is descriptive: the fraction of the sample that discriminates.

We can hear our colleagues objecting now – the inquiry in an audit study can of course be conceived of as causal! It's the average effect of signaling membership in one social group on a resume versus signaling membership in another. We agree, of course, that this interpretation is possible and technically correct. But when we think of the inquiry as descriptive, we can better understand how the audit experiment relies on substantive assumptions about how people who do and do not discriminate behave.

Consider White, Nathan and Faller (2015), which seeks to measure discrimination against Latinos by election officials by assessing whether election officials respond to emailed requests for information from putatively Latino or White voters. We imagine three types of election officials: those who would always respond to the request (regardless of the emailer's ethnicity), those who would never respond to the request (again regardless of the emailer's ethnicity), and officials who discriminate against Latinos. Here, discriminators are defined by their behavior: they would respond to the White voter but not to the Latino voter. These three types are given in Table 16.1.

Table 16.1: Audit experiment response types

Type	$Y_i(Z_i = \text{White})$	$Y_i(Z_i = \text{Latino})$
Always-responder	1	1
Anti-Latino Discriminator	1	0
Never-responder	0	0

Our descriptive inquiry is the fraction of the sample that discriminates: $E[\text{Type} = \text{AntiLatinoDiscriminator}]$. Under this behavioral assumption, $E[\text{Type} = \text{AntiLatinoDiscriminator}] = E[Y_i(Z_i = \text{White}) - Y_i(Z_i = \text{Latino})]$, which is why we can use a randomized experiment to measure this descriptive

quantity. In the data strategy, we randomly sample from the $Y_i(Z_i = \text{White})$'s and from the $Y_i(Z_i = \text{Latinos})$'s, then in the answer strategy, we take a difference-in-means, generating an estimate of the fraction of the sample that discriminates.

Some finer points about this behavioral assumption. First, we assume that Always-responders and Never-responders do not engage in discrimination. It could be that some Never-responder doesn't respond to the Latino voter out of racialized animus but doesn't respond to the White voter out of laziness. In this model, such an official would be not be classified as an Anti-Latino Discriminator. Second, we assume that there are no Anti-White discriminators. If there were, then the difference-in-means would not be unbiased for the fraction of Anti-Latino discriminators. Instead, it would be unbiased for “net” discrimination, i.e., how much more election officials discriminate against Latinos versus how much they discriminate against Whites. Anti-Latino discrimination and net discrimination are theoretically distinct inquiries, but the distinction is often elided in experimental audit studies.

Declaration 16.2 connects the behavioral assumption we make about subjects to the randomized experiment we used to infer the value of a descriptive quantity. Only Never-responders fail to respond to the White request while only Always-responders respond to the Latino request. The inquiry is the proportion of the sample that is an anti-Latino discriminator. The data strategy involves randomly assigning the putative ethnicity of the voter making the request and recording whether it was responded to. The answer strategy is compares average response rates by randomly assigned group.

Declaration 16.1.

```
types <- c("Always-Responder", "Anti-Latino Discriminator", "Never-Responder")
design <-
  declare_model(
    N = 1000,
    type = sample(size = N,
                  replace = TRUE,
                  x = types,
                  prob = c(0.30, 0.05, 0.65)),
    # Behavioral assumption represented here:
    Y_Z_white = if_else(type == "Never-Responder", 0, 1),
    Y_Z_latino = if_else(type == "Always-Responder", 1, 0)
  ) +
  declare_inquiry(anti_latino_discrimination = mean(type == "Anti-Latino Discriminator")) +
  declare_assignment(Z = complete_ra(N, conditions = c("latino", "white"))) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, inquiry = "anti_latino_discrimination")
```

16.1.1 Intervening to decrease discrimination

Butler and Crabtree (2017) prompt researchers to “move beyond measurement” in audit studies. Under the model assumptions in the design, audit experiments measure the level of discrimination, but of course they do not do anything to reduce them. To move beyond measurement, we intervene in the world to reduce discrimination in a treatment group but not in a control group, then measure the level of discrimination in both arms using the audit experiment technology.

This two-stage design can be seen clearly in the declaration below. The first half of the design is about causal inference: we want to learn about the effect of the intervention on discrimination. The second half of the design is about descriptive inference – **within each treatment arm**. We incorporate both stages of the design in the answer strategy, in which the coefficient on the interaction of the intervention indicator with the audit indicator is our estimator of the effect on discrimination.

Even at 5,000 subjects, the power to detect the effect of the intervention is quite poor, at approximately 15%. This low power stems from the small treatment effect (reducing discrimination by 50% from 5.0% to 2.5%) and from the noisy measurement strategy.

Declaration 16.2.

```
N = 5000

design <-
  # This part of the design is about causal inference
  declare_model(
    N = N,
    type_D_0 = sample(
      size = N,
      replace = TRUE,
      x = types,
      prob = c(0.30, 0.05, 0.65)
    ),
    type_tau_i = rbinom(N, 1, 0.5),
    type_D_1 = if_else(
      type_D_0 == "Anti-Latino Discriminator" &
        type_tau_i == 1,
      "Always-Responder",
      type_D_0
    )
  )
```

```

    )
) +
declare_inquiry(ATE = mean((type_D_1 == "Anti-Latino Discriminator") -
                           (type_D_0 == "Anti-Latino Discriminator"))
)) +
declare_assignment(D = complete_ra(N)) +
declare_measurement(type = reveal_outcomes(type ~ D)) +
# This part is about descriptive inference in each condition!
declare_model(
  Y_Z_white = if_else(type == "Never-Responder", 0, 1),
  Y_Z_latino = if_else(type == "Always-Responder", 1, 0)
) +
declare_assignment(
  Z = complete_ra(N, conditions = c("latino", "white"))) +
declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
declare_estimator(Y ~ Z * D, term = "Zwhite:D", inquiry = "ATE")

```

Further reading

- Coppock (2019) discusses how to avoid post-treatment bias when studying how the audit treatment affects the “quality” of responses, such as the tone of an email or the enthusiasm of the hiring call-back. Interestingly, this causal effect is not defined among the discriminators, because they never send emails to Latinos, so those emails never have a tone. The causal effect on tone is defined only among the “nicest” subjects who always respond, but estimating this effect without bias is tricky.
- Fang, Guess and Humphreys (2019) randomizes a New York City government intervention designed to stop housing discrimination, which was then measured by an audit study design.
- Statistical power is a major issue when using audit studies to measure the causal effect of interventions on discrimination. Blair, Coppock and Moor (2020) describe the analogous problem of low statistical power when using list experiments to measure the causal effect of interventions on sensitive traits

Exercises

1. Modify the descriptive design to allow for anti-White discrimination.
hint: please use this set of types: type = sample(size = N, replace = TRUE, x = c("Always-Responder", "Anti-Latino Discriminator",

"Never-Responder", "Anti-White Discriminator), prob = c(0.30, 0.05, 0.63, 0.02))

- a) What is the bias for the anti-Latino discrimination inquiry?
 - b) Include a net_discrimintation inquiry. What is the bias for that inquiry?
2. Modify the sample size of the “moving beyond measurement” design. How large does it have to be before the power for the interaction term reaches 80%?

16.2 List experiments

Sometimes, subjects might not tell the truth about certain attitudes or behaviors when asked directly. Responses may be affected by sensitivity bias, or the tendency of survey subjects to dissemble for fear of negative repercussions if some individual or group learns their true response (Blair, Coppock and Moor, 2020). In such cases, standard survey estimates based on direct questions will be biased. One class of solutions to this problem is to obscure individual responses, providing protection from social or legal pressures. When we obscure responses systematically through an experiment, we can often still identify average quantities of interest. One such design is the list experiment (introduced in Miller (1984)), which asks respondents for the count of the number of “yes” responses to a series of questions including the sensitive item, rather than for a yes or no answer on the sensitive item itself. List experiments give subjects cover by aggregating their answer to the sensitive item with responses to other questions.

For example, study religious discrimination among Americans regarding immigration policy. They worried that in asking people directly whether they were willing to grant citizenship to Muslims that prejudiced people would not be willing to admit their opposition to the policy. To mitigate this risk, the authors obtained estimates of preferences for allowing legal immigration for Muslims that were free of social desirability bias using a list experiment. Subjects in the control and treatment groups were asked: “Below you will read [three/four] things that sometimes people oppose or are against. After you read all [three/four], just tell us HOW MANY of them you OPPOSE. We don’t want to know which ones, just HOW MANY.”

Table 16.2: Creighton and Jamal (2015) list experiment conditions

Control	Treatment
The federal government increasing assistance to the poor.	The federal government increasing assistance to the poor.
Professional athletes making millions of dollars per year.	Professional athletes making millions of dollars per year.

Control	Treatment
Large corporations polluting the environment.	Large corporations polluting the environment. Granting citizenship to a legal immigrant who is Muslim.

The treatment group averaged 2.123 items while the control group averaged 1.904 items, for a difference-in-means estimate 0.219. Under the usual assumptions of randomized experiments, the difference-in-means is an unbiased estimator for the average treatment effect of *being asked* to respond to the treated list versus the control list. But our (descriptive) inquiry is the proportion of people who would grant citizenship to a legal immigrant who is Muslim.

For the difference-in-means to be an unbiased estimator for that inquiry, we invoke two additional assumptions (Imai, 2011):

- **No design effects.** The count of “yes” responses to control items must be the same whether a respondent is assigned to the treatment or control group. The assumption highlights that we need a good estimate of the average control item count from the control group (in the example, 1.843). We use that to net out the control item count from responses to the treated group (what is left is the sensitive item proportion).
- **No misreporting.** The respondent must report the truthful answer to the sensitive item in the treatment group, when granted the anonymity protection of the list experiment. This assumption relies on the fact that the sensitive item is aggregated among the control items and so identifying individual responses is, in most cases, not possible, and this cover is enough to change the respondent’s willingness to truthfully report. However, there are two circumstances in which the respondent is not provided any cover: if the respondent reports “zero” in the treatment group, they are exactly identified as not holding the sensitive trait; when they report the highest possible count in the treatment group, they are exactly identified as holding the trait. We describe the resulting biases below as floor and ceiling effects, respectively.

The no liars assumption of list experiments is evident in the DAG (Figure 16.1): sensitivity bias S is not a parent of the list experiment outcome Y^L by assumption (no liars). The no design effects assumption is not directly visible.

We declare a design for the list experiment to study the descriptive estimand of the prevalence. Our model includes subjects’ true support for granting citizenship to Muslims (Y_{star}) and whether or not they are “shy” (S). These two variables combine to determine how subjects will respond when asked directly about support for the policy. The potential outcomes model combines three

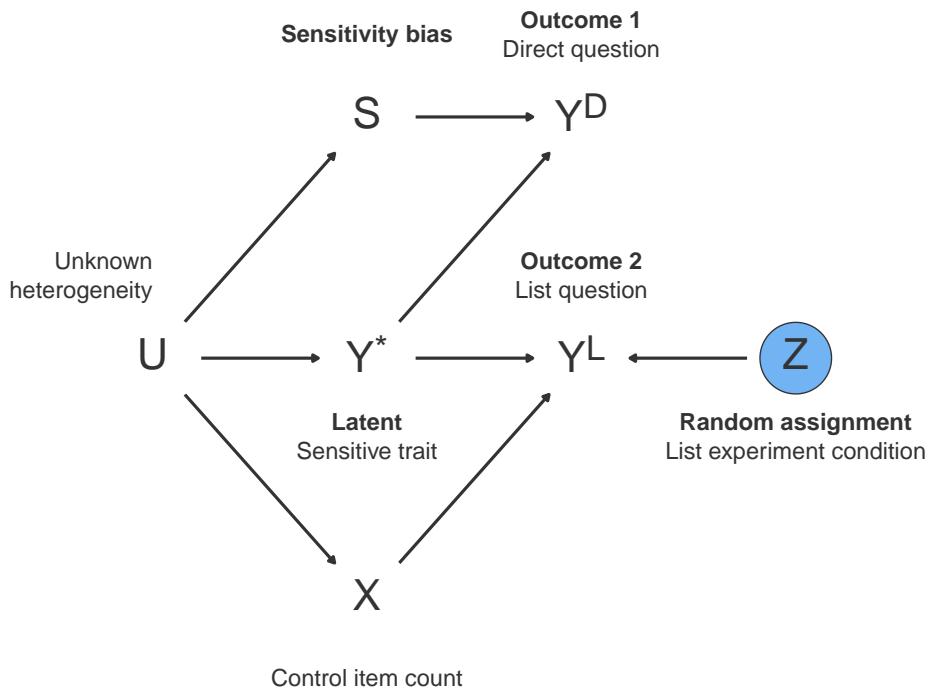


Figure 16.1: DAG for the list experiment.

types of information to determine how subjects will respond to the list experiment: their responses to the three nonsensitive control items (`control_count`), their true support for Trump (`Y_star`), and whether they are assigned to see the treatment or the control list (`Z`). Notice that our definition of the potential outcomes embeds the no liars and no design effects assumptions required for the list experiment design. Our estimand is the proportion of voters who support granting citizenship to Muslims. In the data strategy, we randomly assign 50% of our 100 subjects to treatment and the remainder to control. In the survey, we ask subjects the list experiment question (`Y_list`). Our answer strategy involves estimating the proportion who would grant citizenship to Muslims by calculating the difference-in-means in the list outcome between treatment and control.

Declaration 16.3.

```
design <-  
  declare_model(  
    N = 5000,  
    control_count = rbinom(N, size = 3, prob = 0.5),
```

Table 16.3: Diagnosis of a list experiment.

Bias	Mean CI width
-0.001	0.325

```

Y_star = rbinom(N, size = 1, prob = 0.3),
potential_outcomes(Y_list ~ Y_star * Z + control_count)
) +
declare_inquiry(prevalence_rate = mean(Y_star)) +
declare_sampling(S = complete_rs(N, n = 500)) +
declare_assignment(Z = complete_ra(N)) +
declare_measurement(Y_list = reveal_outcomes(Y_list ~ Z)) +
declare_estimator(Y_list ~ Z, model = difference_in_means,
inquiry = "prevalence_rate")

```

```

diagnosands <- declare_diagnosands(
  bias = mean(estimate - estimand),
  mean_CI_width = mean(abs(conf.high - conf.low))
)
diagnosis <- diagnose_design(design, sims = sims, diagnosands = diagnosands)

```

We see in the diagnosis that there is no bias, but the average width of the confidence interval is enormous: 32 percentage points.

16.2.1 Assumption violations

Recent work on list experiments emphasizes the possibility of violations of both the no liars and the no design effects assumptions. We can diagnose the properties of our design under plausible violations of each.

First, we consider violations of the no design effects assumption, which means that the control item count differs depending on whether a subject is assigned to treatment or control. Typically, this means that the inclusion of the sensitive item changes responses to the control items, because they are judged in relative terms or because the respondent became suspicious of the researcher's intentions due to the taboo of asking a sensitive question.

We declare a modified design below that defines two different potential control item counts depending on whether the respondent is in the treatment group

Table 16.4: Diagnosis of bias due to design effects

inquiry	bias
prevalence_rate	-0.75

(control_count_treat) or control group (control_count_control). The potential outcomes for the list outcome also change: control_count_treat is revealed in treatment and control_count_treat in control.

Declaration 16.4. List experiment “design effects” design

```
design_design_effects <-
  declare_model(
    N = 5000,
    U = rnorm(N),
    control_count_control = rbinom(N, size = 3, prob = 0.5),
    control_count_treat = rbinom(N, size = 3, prob = 0.25),
    Y_star = rbinom(N, size = 1, prob = 0.3),
    potential_outcomes(Y_list ~ (Y_star + control_count_treat) * Z + control_count_control)
  ) +
  declare_inquiry(prevalence_rate = mean(Y_star)) +
  declare_sampling(S = complete_rs(N, n = 500)) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y_list = reveal_outcomes(Y_list ~ Z)) +
  declare_estimator(Y_list ~ Z, inquiry = "prevalence_rate")
```

```
diagnose_design_effects <- diagnose_design(design_design_effects, sims = sims)
```

In the diagnosis, we see that there is substantial bias in estimates of the prevalence rate in the presence of design effects:

Second, a violation of no liars implies that respondents do not respond truthfully even when provided with the privacy protection of the list experiment. Two common circumstances researchers worry about are ceiling effects and floor effects. In ceiling effects, respondents respond with the maximum number of control items rather than with their truthful response of that plus one, to avoid being identified as holding the sensitive trait. The floor effects problem is the reverse, when respondents hide not holding the sensitive trait by responding one *more* than their truthful count in the treatment group.

Table 16.5: Diagnosis of bias due to ceiling effects

inquiry	bias
prevalence_rate	-0.037

Declaration 16.5. List experiment ceiling effects design

```
design_liars <-
  declare_model(
    N = 5000,
    U = rnorm(N),
    control_count = rbinom(N, size = 3, prob = 0.5),
    Y_star = rbinom(N, size = 1, prob = 0.3),
    potential_outcomes(
      Y_list ~
        if_else(control_count == 3 & Y_star == 1 & Z == 1,
               3,
               Y_star * Z + control_count)
    ) +
    declare_inquiry(prevalence_rate = mean(Y_star)) +
    declare_sampling(S = complete_rs(N, n = 500)) +
    declare_assignment(Z = complete_ra(N)) +
    declare_measurement(Y_list = reveal_outcomes(Y_list ~ Z)) +
    declare_estimator(Y_list ~ Z, inquiry = "prevalence_rate")
```

```
diagnose_liars <- diagnose_design(design_liars, sims = sims)
```

Again, we see in the presence of a violation of this assumption, no liars, that there is bias in our estimates of the prevalence rate.

Ceiling and floor effects are not the only ways in which the no liars assumption might be violated. Respondents, when noting the sensitive item among the list, might always respond zero (or the highest number) regardless of their control item count to hide their response. A declaration could be made for these other kinds of violations, also by changing the potential outcomes for Y_list.

16.2.2 Choosing design parameters

Researchers have control of three important design parameters that affect the inferential power of list experiments: the number of control items, the selection of control items, and sample size. Of course, researchers also must choose whether to adopt a list experiment, compared to a simpler direct question. We take up this question in the discussion of sample size.

How many control items. After sample size, an early choice list researchers must make is how many control items to select. Here we also face a tradeoff: the more control items, the more privacy protection for the respondent; but the more items the more variance and the less efficient our estimator of the proportion holding the sensitive item. We can quantify the amount of privacy protection provided as the average width of the confidence interval on the posterior prediction of the sensitive item given the observed count. The efficiency can be quantified as the RMSE.

Which control items. The choice of which set of control items to ask can be as or more important than the number. There are three aims with their selection: reduce bias from ceiling and floor effects, provide sufficient cover to respondents so the no liars assumption is met, and increase efficiency of the estimates. The first goal can be met by reducing the number of people whose latent control count is between one and $J - 1$, one above and one below the lowest and highest numbers possible in the treated group. Respondents in this band will not feel pressured to subtract (add) from their responses to hide that they (do not) hold the sensitive item. One solution to this is to add an item with high prevalence and an item with low prevalence. Though this would address problem one, it would violate problem two: items that are obviously high and low prevalence do nothing to add to privacy protection. The ideal control item count is one with low variance around the middle of the range of the count. To achieve this while providing sufficient cover, items that are inversely correlated can be added.

Sample size. The bias-variance tradeoff in the choice between list and direct questioning can be diagnosed by examining the root mean-squared error (a measure of the efficiency of the design) across two varying parameters: sample size and the amount of sensitivity. We declare a new design with varying `n` and varying `proportion_shy`, the proportion of Trump voters who withhold their truthful response when asked directly:

Declaration 16.6. Combined list experiment and direct question design

```
design <-  
  declare_model(  
    N = 5000,  
    U = rnorm(N),
```

```

control_count = rbinom(N, size = 3, prob = 0.5),
Y_star = rbinom(N, size = 1, prob = 0.3),
W = case_when(Y_star == 0 ~ 0L,
               Y_star == 1 ~ rbinom(N, size = 1, prob = proportion_shy)),
potential_outcomes(Y_list ~ Y_star * Z + control_count)
) +
declare_inquiry(prevalence_rate = mean(Y_star)) +
declare_sampling(S = complete_rs(N, n = n)) +
declare_assignment(Z = complete_ra(N)) +
declare_measurement(Y_list = reveal_outcomes(Y_list ~ Z),
                    Y_direct = Y_star - W) +
declare_estimator(Y_list ~ Z, inquiry = "prevalence_rate", label = "list") +
declare_estimator(Y_direct ~ 1, inquiry = "prevalence_rate", label = "direct")

designs <- redesign(design, proportion_shy = seq(from = 0, to = 0.3,
by = 0.1), n = seq(from = 500, to = 5000, by = 500))

```

```
diagnosis_tradeoff <- diagnose_design(designs, sims = sims, bootstrap_sims = b_sims)
```

Diagnosing this design, we see that at low levels of sensitivity and low sample sizes, the direct question is preferred on RMSE grounds. This is because though the direct question is biased for the proportion of Trump voters in the presence of any sensitivity bias (positive `proportion_shy`), it is much more efficient than the list experiment. When we have a large sample size, then we begin to prefer the list experiment for its low bias. At high levels of sensitivity, we prefer the list on RMSE grounds despite its inefficiency, because bias will be so large. Beyond the list experiment, this diagnosis illustrates that when comparing two possible designs we need to understand both the bias and the variance of the designs in order to select the best one in our setting. In other designs, it will not be the proportion who are shy but some other feature of the model or data and answer strategy that affect bias.

In the upper left, we see that when there is no sensitivity bias we always prefer the direct question due to the inefficiency of the list experiment. The red line is always below the blue. However, when we get to 0.1, there are sample sizes at which we prefer the direct question to the list: below 3000 subjects. However, above 3000 subjects the RMSE of the list experiment is better than the direct question. When we get to 25 percent of Trump supporters misreporting, we always prefer the list experiment in terms of RMSE. In other words, at such high levels of sensitivity bias we are always willing to tolerate the efficiency loss to get an unbiased estimate in this region.

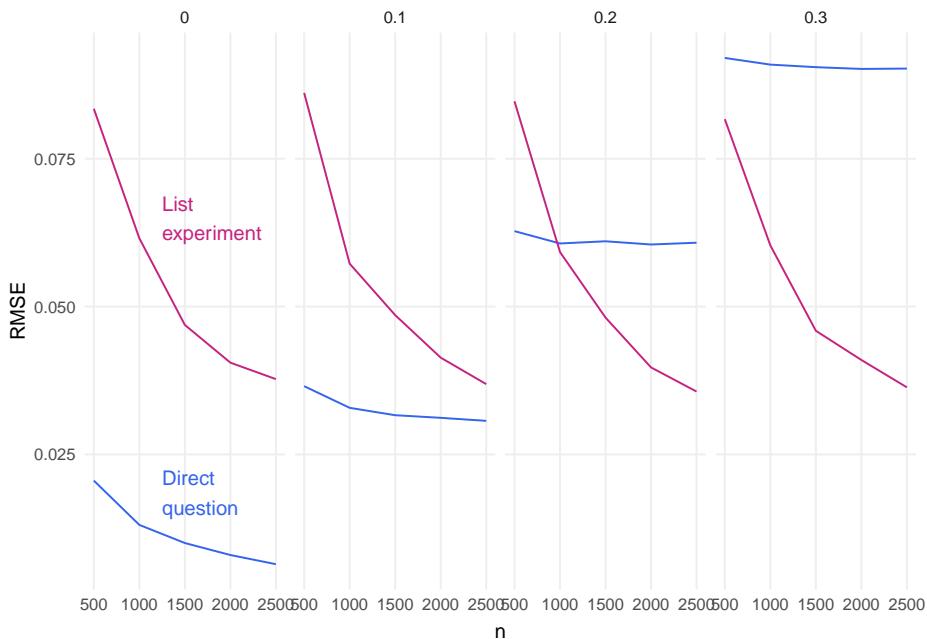


Figure 16.2: Redesign to illustrate tradeoffs in RMSE between list experiment and direct question

Exercises

1. The variance of the list experiment is given by this expression, where $V(Y_i(0))$ is the variance of the control item count and $\text{cov}(Y_i(0), D_i)$ is the covariance of the control item count with the sensitive trait.

$$\frac{1}{N-1} \{(1) + 4V(Y_i(0)) + 4\text{cov}(Y_i(0), D_i)\}$$

Our goal is to compare the direct question and list experiment designs with respect to the RMSE diagnosand. Recall that RMSE equals the square root of variance plus bias squared: $\text{RMSE} = \sqrt{\text{Variance} + \text{Bias}^2}$. Assume the following design parameters: $\sigma^2 = 0.10$, $\mu = 0.50$, $V(Y_i(0)) = 0.075$, $\text{cov}(Y_i(0), D_i) = 0.025$.

- a. What is the RMSE of the direct question when $N = 100$?
- b. What is the RMSE of the list experiment when $N = 100$?
- c. Make a figure with N on the horizontal axis and RMSE on the vertical axis. Plot the RMSE for both designs over a range of sample sizes from 100 to 2000. Hint: you'll need to write a function for each design that takes N as an input and returns RMSE. You can get started by filling out this starter function: `direct_rmse <- function(N){ # write_your_function_here}`

- d. How large does the sample size need to be before the list experiment is preferred to the direct question on RMSE grounds?
- e. Comment on how your answer to (d) would change if were equal to 0.2? What are the implications for the choice between list experiments and direct questions?
2. The list experiment is one of several experimental designs for answering descriptive inquiries about sensitive topics. Most can target the same inquiry: the proportion of subjects who hold a sensitive trait. The randomized response technique is another such design, itself with many variants. In a “forced response” randomized response design (see Blair, Imai and Zhou (2015)), respondents are asked to roll a dice, and depending on the dice result they either answer honestly or are “forced” to answer either “yes” or “no.” With a six-sided dice, respondents might be asked to answer “yes” if they roll a 6, “no” if they roll a “1”, and to answer the question truthfully if they roll any other number, two through five. Because the probability of rolling a 1 and 6 are known, we can back out the probability of answering the sensitive item from the observed data. Declaring this design necessitates changes in M (potential outcomes are a function of the dice roll); D (random assignment is the dice roll itself); and A (an estimator that is a function of the observed outcomes and the known probability of being forced into each response). We declare one below:

```
model_rr <-
  declare_model(
    N = 100,
    U = rnorm(N),
    X = rbinom(N, size = 3, prob = 0.5),
    Y_star = rbinom(N, size = 1, prob = 0.3),
    potential_outcomes(
      Y_rr ~
        case_when(
          dice == 1 ~ 0L,
          dice %in% 2:5 ~ Y_star,
          dice == 6 ~ 1L
        ),
        conditions = list(dice = 1:6)
    ) +
    declare_assignment(
      dice = complete_ra(N, prob_each = rep(1/6, 6),
                         conditions = 1:6)) +
    declare_measurement(Y_rr = reveal_outcomes(Y_rr ~ dice)) +
    declare_estimator(Y_rr ~ 1, handler = label_estimator(rr_forced_known),
                      label = "forced_known", inquiry = "proportion")
```

16.3 Conjoint experiments

Conjoint survey experiments have become hugely popular in political science and beyond for describing multidimensional preferences over profiles (Hainmueller, Hopkins and Yamamoto, 2014). Conjoint experiments come in two basic varieties: the single profile design and the forced-choice design. Throughout this chapter, we'll discuss these studies in the context of hypothetical candidate experiments, in which candidates are described in terms of a number of attributes each of which can take on levels. In the single profile design, subjects asked to rate one profile at a time using, for example, a 1 - 7 support scale. In a forced-choice conjoint experiment, subjects are shown two profiles at a time, then asked to make a binary choice between them. Forced choice conjoint experiments are especially useful for studying electoral behavior because they closely mirror the real-world behavior of choosing between two candidates at the ballot box. A similar logic applies to purchasing behavior when consumers have to choose one product over another. Occasionally, forced-choice conjoint experiments are applied even when no real-world analogue for the binary choice exists. For example, we rarely face a binary choice between two immigrants or between two refugees, so in those cases, perhaps rating profiles one at a time would be more appropriate.

We take the slightly unorthodox position that conjoint experiments target descriptive, rather than casual inquiries. The reason can be most easily seen in the single profile design case. For concreteness, imagine subjects are presented with one profile at a time that describes the age (young, middle-aged, old), gender (woman, man), and employment sector (public, private) of a candidate for office and are asked to rate their support for the candidate on a 1-7 Likert scale. This set of attributes and levels generates $2 * 3 * 2 = 12$ possible profiles. We *could* ask subjects to rate all 12, but we typically ask them instead to rate only random subset. This design can support many inquiries. First, our inquiry could be average preference for each the 12 profiles. It might also be the difference in the average preference across two profiles. The most common inquiry is the Average Marginal Component “Effect” or AMCE, which summarizes the average difference in preference between two levels of one attribute, averaging over all of the levels of the other attributes. The AMCE for gender, for example, considers the average difference in preference for women candidates versus men candidates among young candidates who work in the private sector, among middle-aged candidates who work in the public sector, and so on for all six combinations. The overall AMCE is a weighted average of all six of these average preference differences, where the weights are given the relative frequency of each type of candidates. We put the “Effect” in AMCE in scare quotes because we think of the AMCE as a descriptive quantity. We of course agree there is a sense in which the AMCE is a causal quantity, since it is the average effect on preferences of describing a hypothetical candidate as a man or a woman. Sure. But we can see this quantity as descriptive if we just imagine asking subjects about both candidates and describing the difference in their preferences. The

only reason we don't ask about all possible profiles is that typically, there are far too many to get through in a typical survey, so we ask subjects about a random subset.

Just like single-profile conjoints, forced-choice conjoints also target descriptive inquiries, but the inquiry is one step removed from raw preferences over profiles. Instead, we aim to describe the fraction of pairwise contests that a profile would win, averaging over all subjects in the experiment. That is, we aim to describe a profile's 'average win rate.' We can further describe the differences in the average win rate across profiles. For example, among young candidates who work in the private sector, what is the average difference in win rates for women versus men? Just as in the single profile case, the AMCE is a weighted average of these differences, weighted by the relative frequency of each type of candidate.

Here again, we *could* think of the AMCE as a causal effect, i.e., the average effect of describing a profile as a woman versus a man. But we can also imagine asking subjects to consider all $12 * 12 = 144$ possible pairwise contests, then using those binary choices to fully describe subjects' preferences over contests. A forced-choice conjoint asks subjects to rate just a random subset of those contests, since asking about all of them would be impractical.

One final wrinkle about the AMCE inquiries, in both the single-profiled and forced-choice cases. They are "data-strategy-dependent" inquiries in the sense that, implicitly AMCEs average over the distribution of the other profile attributes, and that distribution is controlled by the researcher.¹ The AMCE of gender for profiles that describe age and employment sector is different from the AMCE of gender for profiles that also include partisanship. Further, and more subtly, the AMCE of gender for profiles that are 75% public sector and 25% private sector is different from the AMCE of gender for profiles that are 50% public sector and 50% private sector, because those relative frequencies are part of the very definition of the inquiry. For contrast, consider a vignette-style hypothetical candidate experiment in which all or most of the other candidate features are fixed, save gender. In that design, we estimate an ATE of gender under only one set of conditions, but in the conjoint design, the AMCE averages over ATEs under many sets of conditions.

The data strategy for conjoints, then, requires making these four choices, in addition to the usual measurement, sampling, and assignment concerns:

1. Which attributes to include in the profiles
2. Which levels to include in each attribute (and in what proportion)
3. How many profiles subjects are asked to rate at a time
4. How many sets of profiles subjects are asked to rate in total

¹The AMCE need not be data-dependent. We could write down one distribution of profiles in the model to establish the AMCE inquiry, then randomly sample the profiles shown to respondents for a different distribution. This would be a headache, because the estimator would need to be reweighted to successfully target the AMCE inquiry. Better to bring the data strategy in line with the model in the first place.

The right set of attributes is governed by the “masking/satisficing” tradeoff (Bansak et al., 2019). If you don’t include an important attribute (like partisanship in a candidate choice experiment), you’re worried that subjects will partially infer partisanship from other attributes (like race or gender). If so, partisanship is “masked”, and the estimates for the effects of race or gender will be biased by these “omitted variables” (Kirkland and Coppock, 2018). But if you add too many attributes in order to avoid masking, you may induce “satisficing” among subjects, whereby they only take in a little bit of information, enough to make a “good enough” choice among the candidates.

The right set levels to include is a tricky choice. We want to include all of the most important levels, but every additional level harms statistical precision. If an attribute has three levels, it’s like you’re conducting a three-arm trial, so you’ll want to have enough subjects for each arm. The more levels, the lower the precision.

How many profiles to rate at the same time is also tricky. Our point of view is that this choice should be guided by the real-world analogue of the survey task. If we’re learning about binary choices between options in the real world, then the forced-choice, paired design makes good sense. If we’re learning about preferences over many possibilities, the single profile design may be more appropriate. That said, the paired design can yield precision gains over the single profile design in the sense that subjects rate two profiles at the same time, so we effectively generate twice as many observations for perhaps less than twice as much cognitive effort.

Finally, the right number of **choice tasks** usually depends on your survey budget. You can always add more conjoint tasks and the only cost is the opportunity cost of asking a different question of the survey that may serve some higher scientific purpose. If you’re worried that respondents will get bored with the task, you can always throw out profile pairs that come later in the survey. Bansak et al. (2019) suggest that you can ask many tasks without much loss of data quality.

We begin by establishing the attributes, levels, and their probability distributions, and creating a dataset of candidate types:

```
f1 = c("man", "woman")
f1_prob = c(0.5, 0.5)
f2 = c("young", "middleaged", "old")
f2_prob = c(0.25, 0.50, 0.25)
f3 = c("private", "public")
f3_prob = c(0.5, 0.5)

candidates_df <-
```

```

bind_cols(expand_grid(f1, f2, f3),
          expand_grid(f1_prob, f2_prob, f3_prob)) %>%
mutate(
  candidate = paste(f1, f2, f3, sep = "_"),
  woman = as.numeric(f1 == "woman"),
  middleaged = as.numeric(f2 == "middleaged"),
  old = as.numeric(f2 == "old"),
  public = as.numeric(f3 == "public"),
  prob = f1_prob * f2_prob * f3_prob
)

```

Here we describe the true preferences of the 1,000 individuals we will enroll in our conjoint experiment. We describe preferences using a regression-model-like approach. Subject evaluations of candidates are given by the following equation:

$$\begin{aligned}
Y = &_0 + \\
&_1 \text{woman} + \\
&_2 \text{middleaged} + \\
&_3 \text{old} + \\
&_4 \text{public} + \\
&_5 \text{woman middleaged} + \\
&_6 \text{woman old} + \\
&_7 \text{woman public} + \\
&_8 \text{public middleaged} + \\
&_9 \text{public old} \\
&_{10} \text{woman public middleaged} + \\
&_{11} \text{woman public old}
\end{aligned}$$

Every subject has a different value for each of $_0$ through $_{11}$. Here we imagine that $'s$ are larger and more variable for base terms than for the two-way interactions than for the three-way interactions as this structure of preferences appears to approximate how individuals evaluate candidates (Schwarz and Copock, 2020).

```

individuals_df <-
  fabricate(

```

```

N = 1000,
beta_woman = rnorm(N, mean = 0.1, sd = 1),
beta_middleaged = rnorm(N, mean = 0.1, sd = 1),
beta_old = rnorm(N, mean = -0.1, sd = 1),
beta_public = rnorm(N, mean = 0.1, sd = 1),
# two way interactions
beta_woman_middleaged = rnorm(N, mean = -0.05, sd = 0.25),
beta_woman_old = rnorm(N, mean = -0.05, sd = 0.25),
beta_woman_public = rnorm(N, mean = 0.05, sd = 0.25),
beta_public_middleaged = rnorm(N, mean = 0.05, sd = 0.25),
beta_public_old = rnorm(N, mean = 0.05, sd = 0.25),
# three-way interactions
beta_woman_public_middleaged = 0,
beta_woman_public_old = 0,
# Idiosyncratic error
U = rnorm(N, sd = 1)
)

```

Next, we join candidates and individuals, in order to calculate preferences over each of the 12 candidates for all 1000 subjects. We use this dataset to calculate the true values of the AMCEs. The `calculate_amces` function is hidden, but can be found in the accompanying full chapter script.

```

candidate_individuals_df <-
  left_join(candidates_df, individuals_df, by = character()) %>%
  mutate(
    evaluation =
      beta_woman * woman +
      beta_middleaged * middleaged +
      beta_old * old +
      beta_public * public +
      beta_woman_middleaged * woman * middleaged +
      beta_woman_old * woman * old +
      beta_woman_public * woman * public +
      beta_public_middleaged * public * middleaged +
      beta_public_old * public * old +
      beta_woman_public_middleaged * woman * public * middleaged +
      beta_woman_public_old * woman * public * old +
      U
  )

```

Table 16.6: AMCE Inquiries

inquiry	estimand
AMCE Woman	0.017
AMCE Old	-0.021
AMCE Middle-aged	0.020
AMCE Public	0.035

```
inquiries_df <- calculate_amces(candidate_individuals_df)
inquiries_df
```

We're almost ready to declare this full design. We have to reshape the data back to a wider format in which the rows are individuals and the columns are evaluations of candidates.

```
individuals_wide_df <-
  candidate_individuals_df %>%
  transmute(ID, candidate, evaluation) %>%
  pivot_wider(id_cols = ID,
              names_from = candidate,
              values_from = evaluation)
```

Declaration 16.7 shows the full design. We ask each of our 1000 subjects to evaluate four pairs of candidates. Which pairs of candidates candidates they see are determined by the `declare_assignment` function, which respects the relative frequencies of the candidates set above. The two measurement functions are mildly complicated to address the particularities of the data structure. The estimator is OLS with robust standard errors clustered at the respondent level.

Figure 16.3 shows the sampling distribution of the four AMCE estimators. They are all four unbiased, but with only 1000 subjects evaluating 4 pairs of candidates, the power for the smaller AMES is less than ideal.

Exercises

- Modify Declaration 16.7. How many pairs would the 1,000 subjects need to evaluate before power is above 0.80 for all four estimators? What concerns would you have about asking subjects to evaluate that many pairs?
- Modify Declaration 16.7. How many subjects would you need if every

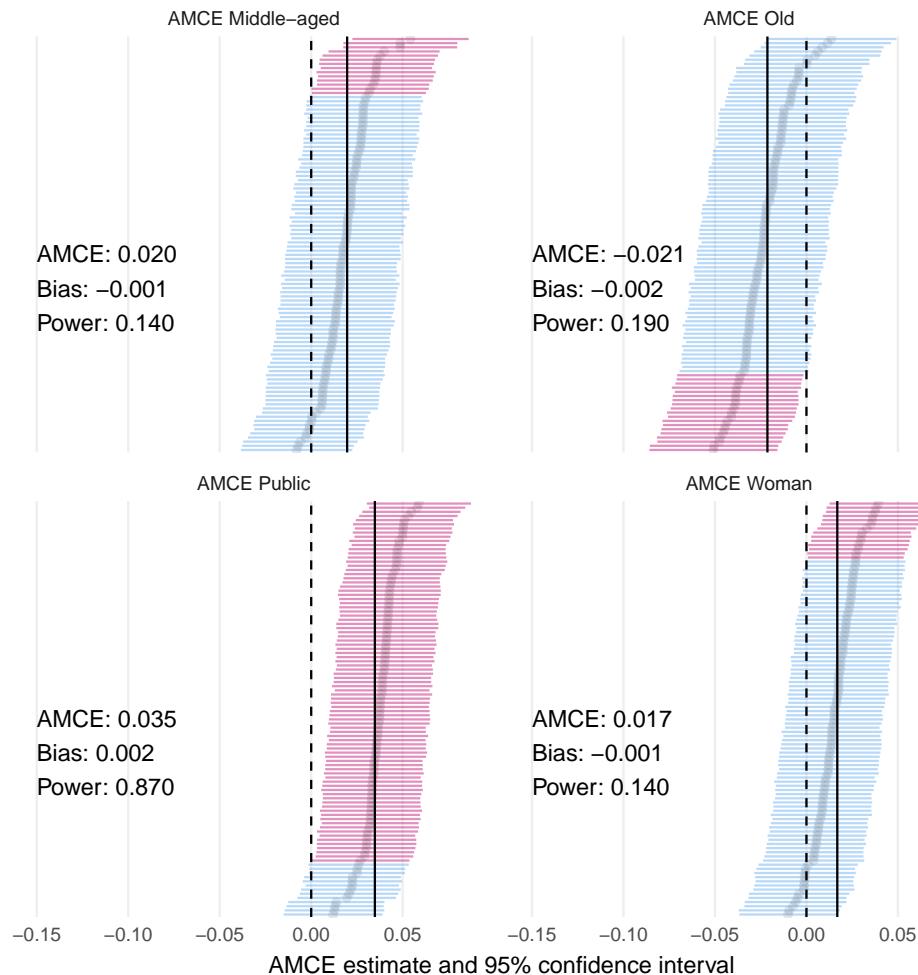


Figure 16.3: Sampling Distribution of four AMCE estimators

subject only evaluated 1 pair?

16.4 Behavioral games

Behavioral games are often used to study difficult-to-measure characteristics of subjects: risk attitudes, altruism, prejudice, trust. The approach involves using lab or other mechanisms to control contexts. A high level of control brings two distinct benefits. First, it can eliminate noise. One can get estimates under a particular well defined set of conditions rather than estimates generated from averaging over a range of conditions. Second, more subtly, it can prevent various forms of confounding. For instance outside the lab we might observe how

people act when they work on tasks with an outgroup member. But we only observe the responses among those that *do* work with out group members, not among those that do not. By setting things up in a controlled way you can see how people *would* react when put into particular situations.

The approach holds enormous value. But, as highlighted by Green and Tusicny (2012), it also introduces many subtle design choices. Many of these can be revealed through declaration and diagnosis. We illustrate using the “trust” game.

The trust game has been implemented hundreds of times to understand levels and correlates of social trust. Following the meta-analysis given in Johnson and Mislin (2011) we consider a game in which one player (Player 1, the “trustor”) gets the chance to invest some share of \$1. Whatever is sent is then doubled. A second player (Player 2, “the trustee”) can then decide what share of the doubled amount to keep for herself and what share to return to the trustor.

As described by Johnson and Mislin (2011), “trust” is commonly measured by the share given and “trustworthiness” is measured by the share returned. With the *MIDA* framework in mind, we will be more specific and define the inquiry independent of the measurement, defining “trust” as the share that *would* be invested by a trustor when confronted with a random trustee, whereas “trustworthiness” is the average share that *would* be returned over a range of possible investments.

To motivate M we will assume a very simple decision making model. We assume that each person seeks to maximize a weight average of logged payoffs. Define:

$$u_i = (1 - a_i) \log(i) + a_i \log(\bar{i})$$

where i, \bar{i} denotes the monetary payoffs to i, \bar{i} and a_i (“altruism”) captures the weight players place of the (logged) payoffs of other players.

Let x denote the amount sent by the trustor from endowment 1.

The trustee then maximizes:

$$u_2 = (1 - a_2) \log((1 - x)2x) + a_2 \log((1 - x) + 2x)$$

where denotes the share of $2x$ that the trustee returns. Maximizing with respect to x yields:

$$= a_2 + (1 - a_2) \frac{x - 1}{2x}$$

in the interior. Taking account of boundary constraints,² we have best response function:

² $a_2 + (1 - a_2) \frac{x - 1}{2x} \leq 0$ requires $x \geq \frac{1 - a_2}{1 + a_2}$

$$(x) = \max (0, a_2 + (1 - a_2) \frac{x - 1}{2x})$$

Note that the share sent back is increasing in the amount sent (because player 2 has greater incentive to compensate player 1 for her investment). If the full amount is sent then the share sent back is simply a_2 .

Given this, the trustor chooses x to maximize:

$$u_1 = (1 - a_1) \log(1 - x + (x)2x) + a_1 \log((1 - (x))2x)$$

In the interior this reduces to:

$$u_1 = (1 - a_1) \log((1 + x)a_2) + a_1 \log((1 - a_2)(1 + x))$$

with greatest returns at $x = 1$.

For ranges in which no investment will be returned, utility reduces to:

$$u_1 = (1 - a_1) \log(1 - x) + a_1 \log(2x)$$

which is maximized at: $x = a_1$.

The global maximum depends on which of these yields higher utility.

Figure 16.4 shows the returns to the trustor from different investments given own and other player other regarding preferences. We see that when other regarding preferences are weak for both players nothing is given and nothing is returned. When other regarding preferences are strong for player 1 she offers substantial amounts even when nothing is expected in return. When other regarding preferences are sufficiently strong for player 2, player 1 invests fully in anticipation of a return.

The predictions of this model are then used to define the inquiry and predict outcomes in the model declaration. The model part of the design includes information on underlying preferences and an indicator of arrival time in the lab, which we allow to be correlated with underlying preferences.

The inquiries for this design are the expected share offered to different types of trustees, the expected returns, averaged over possible offers, and the expected action by a trustee when the full amount is invested. The data strategy involves assigning players to pairs and orderings based on arrival time at the lab. The first half is assigned to the trustor role and matched with the second group that are assigned the trustee role. For the answer strategy, we simply measure average behavior across subjects.

The design declaration is below. Two features are worth highlighting. First the inquiries are defined using a set of hypothetical responses under the model

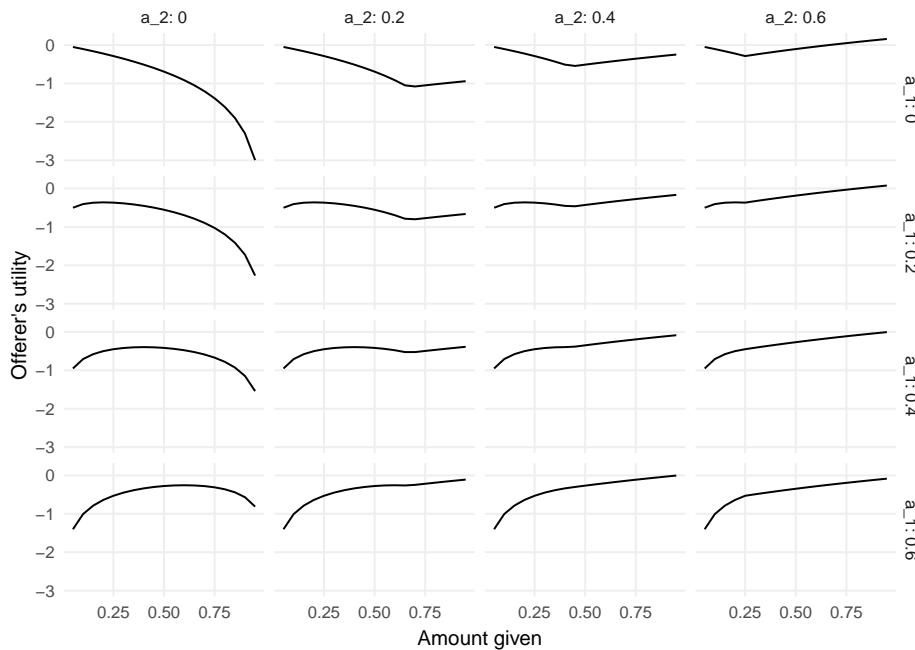


Figure 16.4: Illustration of a trust game

using a specified response function. Second the declaration involves a step where we shift from a “long” data frame with a row per subject to a “wide” data frame with a row per game.

Declaration 16.7.

```

returned <- function(x1, a_2 = 1 / 3) {
  ((2 * a_2 * x1 - (1 - a_2) * (1 - x1)) / (2 * x1)) * (x1 > (1 - a_2) / (1 + a_2))
}

invested <- function(a_1, a_2) {
  u_a = (1 - a_1) * log(1 - a_1) + a_1 * log(2 * a_1) # give a1
  u_b = (1 - a_1) * log(2 * a_2) + a_1 * log(2 * (1 - a_2)) # give 1
  ifelse(u_a > u_b, a_1, 1)
}

average_invested <- function(a_1) {
  mean(sapply(seq(0, 1, .01), invested, a_1 = a_1))
}

```

```

average_returned <- function(a_2) {
  mean(sapply(seq(0.01, 1, .01), returned, a_2 = a_2))
}

rho      <- 0.8
n_pairs <- 200

design <-
  declare_model(N = 2 * n_pairs,
                a = runif(N),
                arrival = rank(correlate(given = a, rho = rho, runif))) +
  declare_inquiries(
    mean_invested = mean(sapply(a, average_invested)),
    mean_returned = mean(sapply(a, average_returned)),
    return_from_1 = mean(returned(1, a))
  ) +
  declare_assignment(pair = (arrival - 1) %% n_pairs,
                     role = 1 + (arrival > n_pairs)) +
  declare_step(
    id_cols = pair,
    names_from = role,
    values_from = c(ID, a),
    handler = tidyverse::pivot_wider
  ) +
  declare_measurement(invested = invested(a_1, a_2),
                      returned = returned(invested, a_2)) +
  declare_estimator(invested ~ 1,
                    model = lm_robust,
                    inquiry = "mean_invested",
                    label = "mean_invested") +
  declare_estimator(returned ~ 1,
                    model = lm_robust,
                    inquiry = "mean_returned",
                    label = "mean_returned") +
  declare_estimator(
    returned ~ 1,
    model = lm_robust,
    subset = invested == 1,
    inquiry = "return_from_1",
    label = "return_from_1"
  )

```

Table 16.7: Sample data

pair	ID_1	ID_2	a_1	a_2	invested	returned
0	117	189	0.00	0.13	0.00	0.00
1	234	122	0.02	0.41	0.02	0.00
2	288	018	0.02	0.12	0.02	0.00
3	332	009	0.00	0.66	1.00	0.66
4	282	190	0.05	0.29	0.05	0.00
5	089	331	0.09	0.92	1.00	0.92

Table 16.8: Diagnosis of a simple trust game.

Inquiry	Mean Estimand	Mean Estimate	Bias
mean_invested	0.82 (0.00)	0.85 (0.00)	0.03 (0.00)
mean_returned	0.31 (0.00)	0.66 (0.00)	0.35 (0.00)
return_from_1	0.50 (0.00)	0.68 (0.00)	0.18 (0.00)

Data generated by this design might look like this:

Note we have a row for each game, we have the (unobserved) a_i, a_j parameters as well as actions by both players in the data. The diagnosis looks like this:

The diagnosis reveals that we do very poorly for all three inquiries. In fact, despite the simplicity of the design, we made a number of errors – partly due to over-zealous control.

The first problem is that we assigned subjects to roles based on the time of arrival at the lab rather than randomly. The consequence of this choice is that the set of people in each role is not representative of our sample (much less a population). That problem is fixed through random generation of pairs and random assignment of roles *within* pairs (Note that using the same pairs and randomly assigning roles would *not* be enough in this case since in principle the non random assignment of pairs could mean that players are playing with other players that are like or unlike them more than one would typically expect).

Here is the diagnosis when this fix is made:

```
design <-
```

Table 16.9: Diagnosis of a design with random matching and random assignment of roles.

Inquiry	Mean Estimand	Mean Estimate	Bias
mean_invested	0.82 (0.00)	0.82 (0.00)	0.00 (0.00)
mean_returned	0.31 (0.00)	0.47 (0.00)	0.16 (0.00)
return_from_1	0.50 (0.00)	0.51 (0.00)	0.01 (0.00)

```
design %>%
  replace_step(3, declare_assignment(pair = complete_ra(N=N, num_arms = n_pairs))) %>%
  replace_step(4, declare_assignment(role = 1 + block_ra(blocks = pair)))
```

We do better now for the first inquiry. But we still do poorly for the second and third inquiries. There are two distinct reasons for this.

1. Incorrect stage 2 distributions. Although we have assigned roles randomly, the choices confronting Player 2 are not random: they reflect the particular assignments generated by Player 1's choices. These player generated assignments are generally higher than those specified in the definition of the inquiry resulting in higher returns than would arise from random offers.
2. Self-selection in stage 2. This problem is more subtle: even if the distribution of offers confronting the trustees in the second stage were correct, we could still suffer from a problem that the trustees that are sent larger investments are sent those investments partly *because* trustors expects them to return a large share. The bias on the third inquiry — which conditions on the full amount being sent— suggests that this is indeed a concern in this design, though the magnitude of the bias is substantively small.

Fixing this calls for redesign and there are various possibilities. One approach is to confront the returners with a different investment than those generated by Player 1, for instance by confronting them with a random offer. A second approach is to redefine the trustworthiness inquiry to define trustworthiness in reference to the actual distribution of offers made in a population. That is, given the kinds of offers that get made, what share generally gets returned? The first approach involves deception and seems to cut against the spirit of the exercise, which is to observe actual behavior rather than responses to hypothetical situations. The second approach involves a change of question which requires

defending. Moreover it does not solve the concern regarding self selection.

A final approach is to limit the information that players have about each other. We assumed in this design that players had enough information on each other to figure out a_i . Say instead that information on players were coarsened — for instance so that players know only each other's gender and ethnicity. In this case we might have a small set of "types" for Player 1 and Player 2. Conditional on the type pair, the variation in offers is as-if random with respect to a Player 2's characteristics and one could assess the average response of each Player 2 type to each offer received from a Player 1 type.

16.4.1 Pointers

As you declare and diagnose behavioral games we suggest you keep a couple of questions in mind:

1. Is your inquiry defined with respect to a population or sample? The declaration above ignored recruitment of subjects into the lab and focused on sample specific quantities. But if you care about population quantities you can include sampling explicitly in your design declaration.
2. Does the game involve self selection into treatment? As above, in games with more than one stage it is easy to generate situations in which the choices a player faces depends on their characteristics. In such cases you may have to alter your inquiries or your data or answer strategies to get reliable answers.
3. Does your inquiry capture the concept you care about? Lab experiments are often motivated by a desire to learn about some underlying psychological quantity and not a marginal effect of an intervention. In the example above the models we used generate behavior that gets recorded as trusting and trustworthiness but without any features that many might think of as really characteristic of trust. The trustee returns only because they care about outcomes, not because they feel that the trustor is owed some amount on account of the trusting. Similarly, the trustor gives not because they think the trustee will fulfill their part of an implicit bargain but rather because they (a) care directly about the welfare of the trustee and (b) anticipate that the trustee cares about their welfare. These motivations might be what we mean by trust, but, we expect, they might fall short. We see this as a point for declaration. Declaration of the model makes it easier to see whether the inquiry you define (e.g. share returned) actually relates to the underlying quantity that you care about. If you are worried by this, an alternative approach is not to measure trust as a summary of behavior, as we did here, but as a parameter of an underlying model. See our treatment of structural models in Section 18.3 for an introduction to this kind of approach.
4. What are the implications of different control decisions. In the design

above we controlled many features: How many games each person played, the size of endowments, the amount by which investments were increased, and myriad more. As highlighted by Johnson and Mislin (2011) the findings from trust games can in fact be very sensitive to these details. Declarations that incorporate these controlled elements explicitly can help you figure out how they matter. For instance our game involved the investment of \$1 and we then defined trust as the share of \$1 invested. But say our players had \$2 to invest, would we measure trust by the number of cents invested or by the share of \$2 invested? Would the inquiry's value change as we alter these decisions? A declaration that let endowments vary would force us to answer these questions.

Chapter 17

Experimental : causal

An inquiry is causal if it involves a comparison of counterfactual states of the world and a data strategy is experimental if it involves explicit assignment of units to treatment conditions. Experimental designs for causal inference combine these two elements. The designs in this section aim to estimate causal effects and the procedure for doing so involves actively allocating treatments.

Many of experimental designs for causal inference in the social sciences take advantage of researcher control over the assignment of treatments to assign treatments *at random*. In the archetypal two-arm randomized trial, a group of N subjects are recruited, m of them are chosen at random to receive treatment and the remaining $N - m$ of them do not receive treatment and serve as controls. The inquiry is the average treatment effect, the answer strategy is the difference-in-means estimator. The strength of the design can be appreciated by analogy to random sampling. The m outcomes in the treatment group represent a random sample from the treated potential outcomes among all N subjects, so the sample mean in the treatment group is a good estimator of the true average treated potential outcome; an analogous claim holds for the control group.

The randomization of treatments to estimate average causal effects is a relatively recent human invention. While glimmers of the idea appeared earlier, it wasn't until at least the 1920s that explicit randomization appeared in agricultural science, medicine, education, and political science (Jamison, 2019). Only a few generations of scientists have had access to this tool. Sometimes critics of experiments will charge "you can't randomize [important causal variable]." There are of course practical constraints on what treatments researchers can control, be they ethical, financial, or otherwise. We think the main constraint is researcher creativity. The scientific history of randomized experiments is short – just because it hasn't been randomized yet doesn't mean it can't be. (By the same token, just because it *can* be randomized doesn't mean that it should be.)

Randomized experiments are rightly praised for their desirable inferential prop-

erties, but of course they can go wrong in many ways that designers of experiments should anticipate and minimize. These problems include problems in the data strategy (randomization implementation failures, excludability violations, noncompliance, attrition, and interference between units), problems in the answer strategy (conditioning on post-treatment variables, failure to account for clustering, p -hacking), and even problems in the inquiry (estimator-inquiry mismatches). Of course all these problems apply *a fortiori* to nonexperimental studies, but they are important to emphasize for experimental studies since they are often characterized as being “unbiased” without qualification.

The designs in this chapter proceed from the simplest experimental design – the two arm trial – up through very complex designs like the randomized saturation design. The chapter can profitably be read alongside Gerber and Green (2012).

17.1 Two-arm randomized experiments

We declare a canonical two arm trial, motivate key diagnosands for assessing the quality of a design, use diagnosis and redesign to explore the properties of two arm trials, and discuss key risks to inference. This entry includes code for a “designer” which lets you quickly design and redesign two arm trials.

All two-arm randomized trials have in common that subjects are randomly assigned to one of two conditions. Canonically, the two conditions include one treatment condition and one control condition. Some two-arm trials eschew the pure control condition in favor of a placebo control condition, or even a second treatment condition. The uniting feature of all these designs is that the model includes two and only two potential outcomes for each unit and that the data strategy randomly assigns which of these potential outcomes will be revealed.

A key choice in the design of two arm trials is the random assignment procedure. Will we use simple (coin flip, or Bernoulli) random assignment or will we use complete random assignment? Will the randomization be blocked or clustered? Will we “restrict” the randomization so that only randomizations that generate acceptable levels of balance on pre-treatment characteristic are permitted? We will explore the implications of some of these choices in the coming sections, but for the moment, the main point is that saying “treatments were assigned at random” is insufficient. We need to describe the randomization procedure in detail in order to know how to analyze the resulting experiment. See Section 8.1.2 for a description of many different random assignment procedures.

In this chapter, we’ll consider the canonical two arm-trial design described in Gerber and Green (2012). The canonical design conducts complete random assignment in a fixed population, then uses difference-in-means to estimate the average treatment effect. We’ll now unpack this shorthand into the components of M , I , D , and A .

The model specifies a fixed sample of N subjects. Here we aren't imagining that we are sampling from a larger population first. We have in mind a fixed set of units among whom we will conduct our experiment. That is, we are conducting "finite sample inference." Under the model, each unit is endowed with two latent potential outcomes: a treated potential outcome and an untreated potential outcome. The difference between them is the individual treatment effect. In the canonical design, we assume that potential outcomes are "stable," in the sense that all N units' potential outcomes are defined with respect to the same treatment and that units potential outcomes do not depend on the treatment status of other units. This assumption is often referred to as the "stable unit treatment value assumption," or SUTVA (Rubin, 1980).

The potential outcomes themselves have a correlation of ρ . If units with higher untreated potential outcomes also have higher treated potential outcomes, ρ will be positive. Developing intuitions about ρ is frustrated by the fundamental problem of causal inference. Since we can only ever observe a unit in its treated or untreated state (but not both), we can't directly observe the correlation in potential outcomes. In order to make a guess about ρ , we need to reason about treatment effect heterogeneity. If treatment effects are very similar from unit to unit, ρ will be close to 1. In the limiting case of exactly constant effects, ρ is equal to 1.

It is difficult (but not impossible) to imagine settings in which ρ is negative. So-called "Robin Hood" treatments generate negatively correlated potential outcomes, because they "take" from units with high untreated potential outcomes and "give" to units with low untreated outcomes. An example of a Robin Hood treatment might be a "surprising" partisan cue in the context of the American party system. Imagine that in the control condition, Democratic subjects tend to support a policy ($Y_i(0)$ is high) and Republicans tend to oppose it ($Y_i(0)$ is low). The treatment is a "surprise" endorsement of the policy by a Republican elite: treatment group Republicans will find themselves supporting the policy ($Y_i(1)$ is high) whereas treatment group Democrats will infer from the Republican endorsement that the policy must not be a good one ($Y_i(1)$ is low.) Treatments with extreme heterogeneity like this example could in principle cause negatively correlated potential outcomes.

Because the model specifies a fixed sample, the inquiries are also defined at the sample level. The most common inquiry for a two-arm trial is the sample average treatment effect, or SATE. It is equal to the average difference between the treated and untreated potential outcomes for the units in the sample: $E_{iN}[Y_i(1) - Y_i(0)]$. Two-arm trials can also support other inquiries like the SATE among a subgroup (called a conditional average treatment effect, or CATE), but we'll leave those inquiries to the side for the moment.

The data strategy uses complete random assignment in which exactly m of N units are assigned to treatment ($Z = 1$) and the remainder are assigned to control ($Z = 0$). We measure observed outcomes in such a way that we measure the treated potential outcome in the treatment group and untreated potential

outcomes in the control group: $Y = Y_i(1) \text{ if } Z + Y_i(0) \text{ if } (1 - Z)$. This expression is sometimes called the “switching equation” because of the way it “switches” which potential outcome is revealed by the treatment assignment. It also embeds the crucial assumption that indeed units reveal the potential outcomes they are assigned to. If the experiment encounters noncompliance, this assumption is violated. It’s also violated if “excludability” is violated, i.e., if something other than treatment moves with assignment to treatment. For example, if the treatment group is measured differently from the control group, excludability would be violated.

The answer strategy is the difference-in-means estimator with so-called Neyman standard errors:

$$DIM = \frac{\frac{1}{m} \sum_{i=1}^m Y_i - \frac{N}{m+1} \sum_{i=m+1}^{m+N} Y_i}{\sqrt{\frac{Var(Y_i|Z=1)}{m} + \frac{Var(Y_i|Z=0)}{N-m}}} \quad (17.1)$$

$$se(DIM) = \sqrt{\frac{Var(Y_i|Z=1)}{m}} \quad (17.2)$$

$$(17.3)$$

The estimated standard error can be used as an input for two other statistical procedures: null hypothesis significance testing via a t -test and the construction of a 95% confidence interval.

The DAG corresponding to a two-arm randomized trial is very simple. An outcome Y is affected by unknown factors U and a treatment Z . The measurement procedure Q affects Y in the sense that it measures a latent Y and records the measurement in a dataset. No arrows lead into Z because it is randomly assigned. No arrow leads from Z to Q , because we assume no excludability violations wherein the treatment changes how units are measured. This simple DAG confirms that the average causal effect of Z on Y is nonparametrically identified because no back-door paths lead from Z to Y .

17.1.1 Analytic design diagnosis

The statistical theory for the canonical two-arm design is very well explored, so analytic expressions for many diagnosands are available.

1. Bias of the difference-in-means estimator. Equation 2.14 in Gerber and Green (2012) demonstrates that regardless of the values (except in degenerate cases) of m , N , or π , the bias diagnosand is equal to zero. This is the “unbiasedness” property of many randomized experimental designs. On average, the difference-in-means estimates from the canonical design will equal the average treatment effect. As we’ll explore later in this chapter, not every experimental design yields unbiased estimates. Some (like blocked experiments with differential probabilities of assignments)

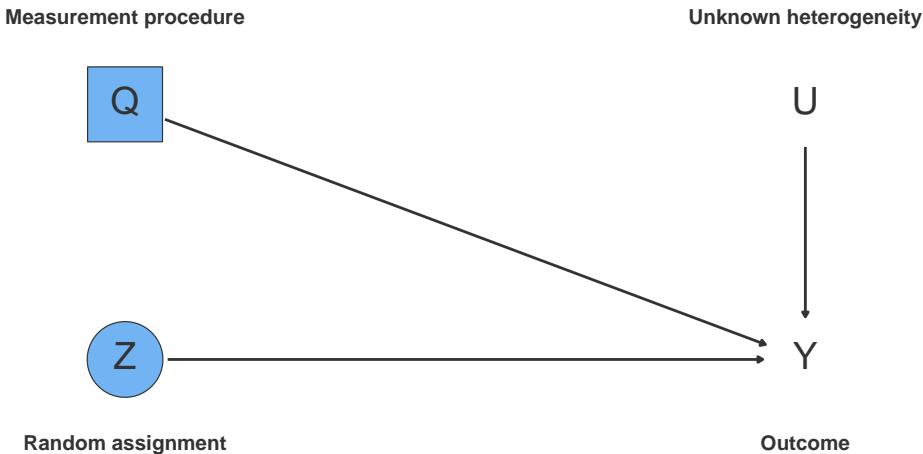


Figure 17.1: DAG of a two-arm randomized experiment

require fix-ups in the answer strategy and others (like clustered experiments with unequal cluster sizes) require fix-ups in the data strategy.

2. The true standard error of the difference-in-means estimator. Equation 3.4 in Gerber and Green (2012) provides an exact expression for the true standard error of the canonical two-arm trial.

$$SE(DIM) = \frac{1}{n-1} \left\{ \frac{mV(Y_i(0))}{n-m} + \frac{(N-m)V(Y_i(1))}{m} + 2Cov(Y_i(0), Y_i(1)) \right\}$$

This equation contains many design lessons. It shows how the standard error decreases as sample size (N) increases and as the variances of the potential outcomes decrease. It provides a justification for “balanced” designs that assign the same proportion of subjects to treatment and control. If the variances of $Y_i(0)$ and $Y_i(1)$ are equal, then a balanced split of subjects across conditions will yield the lowest standard error. If the variances of the potential outcomes are not equal, the expression suggests allocating more units to the condition with the higher variance.

3. Bias of the standard error estimator. Equation 3.4 is the *true* standard error. We also learn from analytic design diagnosis that the standard error estimator is upwardly biased, which is to say that it is conservative (see Section 9.2.1). The intuition for this bias is that we can't directly estimate the covariance term in Equation 3.4, so we bound the variance under a worst-case assumption.¹ The amount of bias in the standard error estimator depends on how wrong this worst case assumption is. When $\sigma^2_{Y_i(0)} = \sigma^2_{Y_i(1)}$, the bias goes to zero.

¹See Aronow, Green and Lee (2014) for an alternative bounding procedure.

4. Coverage. Since the standard errors are upwardly biased – they are “too big” – the statistics that are built on them will inherit this bias as well. The 95% confidence intervals will also be “too big,” so the coverage diagnosis will be above nominal, that is, 95% confidence intervals will cover the true parameter more frequently than 95% of the time.
5. Power. Our answer strategy involves conducting a statistical significance test against the null hypothesis that the average outcome in the control group is equal to the average outcome in the treatment group. This test is also built on the estimated standard error, so the upward bias in the standard error estimator will put downward pressure on statistical power. In Section 17.1.1, we reproduced the formula given in Gerber and Green (2012) for statistical power that makes two further restrictions on the canonical design: equally-sized treatment groups and equal variances in the potential outcomes.

Analytic design diagnosis is tremendously useful, for two reasons. First, we obtain guarantees for a large class of designs. Any experiment that fits into the canonical design will have these properties. Second, we learn from the analytic design diagnosis what the important design parameters are. In our model, we need to think about treatment effect heterogeneity in order to develop expectations about the variances and covariances of the potential outcomes. In our inquiry, we need to be thinking about specific average causal effects – the SATE, not the PATE or the CATE or the LATE. The data strategy in the canonical design is complete random assignment, so we need to think about how many units to assign to treatment (m) relative to control ($N - m$). The answer strategy is difference-in-means with Neyman standard errors – difference in means is unbiased for the ATE, but the Neyman standard error estimator is upwardly biased. This means our coverage will be conservative and we’ll take a small hit to statistical power.

17.1.2 Design diagnosis through simulation

Of course we can also declare this design and conduct design diagnosis using simulation. This process will confirm the analytic results, as well as provide estimates of diagnosands for which statisticians have not yet derived analytic expressions. This code produces a “designer” that allows us to easily vary the important components of the design.

```
eq_3.4_designer <-
  function(N, m, var_Y0, var_Y1, cov_Y0_Y1, mean_Y0, mean_Y1) {

  fixed_sample <-
    MASS::mvrnorm(
      n = N,
```

```

mu = c(mean_Y0, mean_Y1),
Sigma = matrix(c(var_Y0, cov_Y0_Y1, cov_Y0_Y1, var_Y1), nrow = 2),
  empirical = TRUE # this line makes the means and variances "exact" in the sample data
) %>%
  magrittr::set_colnames(c("Y_Z_0", "Y_Z_1"))

declare_model(data = fixed_sample) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_assignment(m = m) +
  declare_reveal(Y, Z) +
  declare_estimator(Y ~ Z, inquiry = "ATE")

}

```

This simulation investigates how much of the sample we should allocate to the treatment group if the treatment group variance is twice as large as the control group variance. The diagnosis confirms the bias is zero, the true standard errors are what Equation 3.4 predicts, coverage is above nominal, and that we are above the 80% power target for a middle range of m . We learn that power is maximized (and the true standard error is minimized) when we allocate 60 or 70 units (of 100 total) to treatment. We also learn from this that the gains from choosing the unbalanced design (relative to a 50/50 allocation) are very small. Even when the variance in the treatment group is twice as large as the variance in the control group, we don't lose much when sticking with the balanced design. Since we can never be sure of the relative variances of the treatment and control groups *ex ante*, this exercise provides further support for choosing balanced designs in many design settings.

```

designs <-
  expand_design(designer = eq_3.4_designer,
    N = 100,
    m = seq(10, 90, 10),
    var_Y0 = 1,
    var_Y1 = 2,
    cov_Y0_Y1 = 0.5,
    mean_Y0 = 1.0,
    mean_Y1 = 1.75)

dx <- diagnose_designs(designs, sims = 100, bootstrap_sims = FALSE)

```

Figure 17.2 illustrates how key diagnosands respond to treatment assignment propensities. We see that bias and coverage are unaffected while standard errors

are minimized, and so statistical power maximized, with middling assignment propensities.

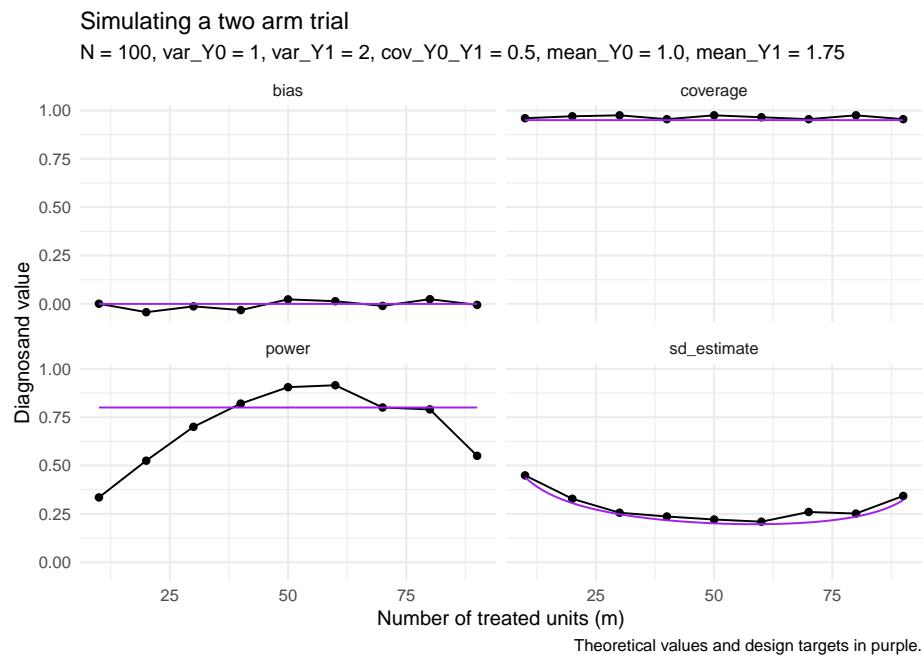


Figure 17.2: How diagnosis depends on the number of units assigned to treatment

17.1.3 What can go wrong

Even for the simplest two-arm trial design, many things can go wrong. Naturally, the sorts of problems can be described in terms of M , I , D , and A .

The most crucial assumption we made in the model is that there are exactly two potential outcomes for each unit. This is violated if there are “spillovers” between units, say between housemates. If a unit’s outcome depends on the treatment status of their housemate, we could imagine four potential outcomes for each unit: only unit i is treated, only unit i ’s housemate is treated, both are treated, or neither is treated. If there are indeed spillovers, but they are ignored, the very definition of the inquiry is malformed. If units don’t even have clearly defined “treated” and “untreated” potential outcomes because of spillovers from others, then we can’t define the ATE in the usual way. The solution to this problem is to *elaborate* the model to account for all the potential outcomes, then to redefine the inquiry with respect to *those* potential outcomes. We explore two experimental designs for learning about spillovers in Sections 17.10 and 17.11.

Other ways for a two arm-trial to go wrong concern the data strategy. If you *think* you are using complete random assignment but you in fact are not, bias may creep in. A nonrandom assignment procedure might be something like “first-come, first-served.” If you assign the “first” m units to treatment and the remainder to control, the assignment procedure is not randomized. Bias will occur if the potential outcomes of the first m are unlike the potential outcomes of the remaining $N - m$ units.

Sometimes researchers do successfully conduct random assignment, but the random assignment happened to produce treatment and control groups that are unlike each other in observable ways. The unbiasedness property applies to the whole procedure – over many hypothetical iterations of the experiment, the average estimate will be equal to the value of the inquiry. But any *particular* estimate can be close or far from the true value. A solution to this problem is to change the answer strategy to adjust estimates for covariates, though we would recommend adjusting for covariates regardless of whether the treatment and control groups appear imbalanced. We explore procedures for including covariates in the data strategy (blocking) and in the answer strategy (covariate adjustment) in Section 17.2.

Other data strategy problems include noncompliance and attrition. Noncompliance occurs when units’ treatment status differs from their treatment assignment. We describe two designs for addressing noncompliance in Sections 17.6 and 17.7. Attrition occurs when outcome data are missing. The attrition problem in experiments is exactly analogous to the attrition problem in descriptive studies (see section 14.1), since we no longer have random samples of each set of potential outcomes.

Further reading

- Chapters 2 and 3 of Gerber and Green (2012) cover the potential outcomes framework and features of sampling distribution of the difference-in-means estimator of the average treatment effect.
- Chapter 2 of Angrist and Pischke (2008) describes the two arm trial as the “experimental ideal” that observational studies are trying to emulate.
- Chapter 7 of Aronow and Miller (2019) provides a rigorous mathematical foundation for causal inference in the two-arm randomized experimental setting.

17.2 Block-randomized experiments

We declare a block randomized trial in which subjects are assigned to treatment and control conditions within groups. We use design diagnosis to assess the reductions in variance of estimation that can be achieved from block randomization, examine possible downsides of block randomization, and compare

strategies that randomize ex ante and that introduce controls ex post.

In a block-randomized experimental design, homogeneous sets of units are grouped together into blocks on the basis of covariates. The ideal blocking would group together units with identical potential outcomes, but since we don't have access to any outcome information at the moment of treatment assignment, let alone the full set of potential outcomes, we have to make do grouping together units on the basis of covariates we hope are strongly correlated with potential outcomes. The blocking will be more effective in terms of increasing precision, the more strongly the blocking variable predicts potential outcomes.

Blocks can be formed on the basis of the levels of a single discrete covariate. We might be able to do better by blocking on the intersection of the levels of two discrete covariates. We could coarsen a continuous variable in order to create strata. We might want to create matched quartets of units, partitioning the sample into sets of four units that are as similar as possible on many covariates. Methodologists have crafted many algorithms for creating blocks, each with their own tradeoffs in terms of computational speed and efficiency guarantees (BlockTools, SoftBlock, Gram-Schmidt). The main point is that there are many paths to the creation of blocks on the basis of covariates and which one is the best choice in any particular setting will depend on the availability of covariate information that is correlated with potential outcomes.

In this design, we block our assignment on a binary covariate X . We assign different fractions of each block to treatment to illustrate the notion that probabilities of assignment need not be constant across blocks, and if they aren't, we need to weight units by the inverse of the probability of assignment to the condition that they are in. In the answer strategy, adjust for blocks using the Lin (2013) regression adjustment estimator including IPW weights. A fuller motivation for the Lin estimator is presented at the end of this section.

Declaration 17.1.

```

    ipw = 1 / probs
) +
declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
declare_estimator(
  Y ~ Z,
  covariates = ~ X,
  model = lm_lin,
  weights = ipw,
  label = "Lin"
)

```

17.2.1 Why does blocking help?

Why does blocking increase the precision with which we estimate the ATE? One piece of intuition is that blocking rules out “poor” random assignments that exhibit imbalance on the blocking variable. If $N = 12$ and $m = 6$, complete random assignment allows $\text{choose}(12, 6) = 924$ possible permutations. If we form two blocks of size 6 and conduct block random assignment, then there are $\text{choose}(6, 3) * \text{choose}(6, 3) = 400$ remaining possible assignments. The assignments that are ruled out are those in which too many or too few units in a block are assigned to treatment, because blocking requires that exactly m_B units be treated in each block B . When potential outcomes are correlated the blocking variable, those “extreme” assignments produce estimates that are in the tails of the sampling distribution associated with complete random assignment.²

This intuition behind blocking is illustrated in Figure 17.3, which shows the sampling distribution of the difference-in-means estimator under *complete* random assignment. The histogram is shaded according to whether the particular random assignment is permissible under a procedure that blocks on the binary covariate X . The sampling distribution of the estimator among the set of assignments that are permissible under blocking is more tightly distributed around the true average treatment effect than the estimates associated with assignments that are not perfectly balanced. Here we can see the value of a blocking procedure – it *rules out by design* those assignments that are not perfectly balanced.

²One mistake sometimes made by new experimenters is to conduct *simple* random assignment within each block – none of the gains from blocking described here apply if simple random assignment is conducted in each block, because that procedure produces the identical randomization distribution as a simple random assignment procedure without any blocking (provided that the probability of assignment is the same in each block).

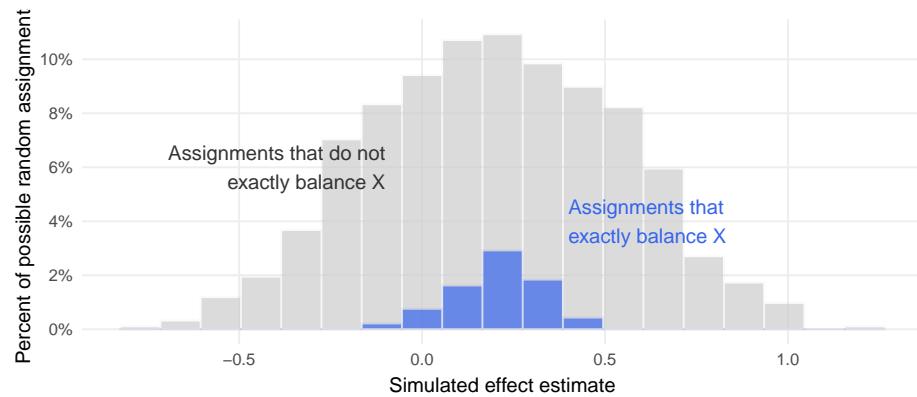


Figure 17.3: Sampling distribution under complete random assignment, by covariate balance

17.2.2 Can blocking ever hurt?

We showed above how blocking typically increases precision by ruling out some random assignments allowed under complete random assignment. When we form blocks of units whose potential outcomes are similar, then assignments that generate estimates that are far from the ATE are ruled out. A “bad blocking” occurs if the assignments that are ruled out are the ones that generate estimates that are close to the ATE. This possibility can only occur if, rather than forming blocks of units with similar potential outcomes, we unwittingly form blocks of units whose potential outcomes are very different. Doing so turns out to be rare and requires a convoluted blocking strategy, but it is possible.

Here is an example of design in which blocking hurts precision. We block on “couple,” but we imagine an “opposites attract” model of romance: the unit with the highest value of X is paired with the unit with the lowest value, the second highest with the second lowest, and so on. The diagnosis shows that in this odd case, the complete random assignment design has a tighter sampling distribution than the block random assignment design. Problems like this can be avoided by blocking together units who are similar on a prognostic covariate, not dissimilar.

Declaration 17.2.

```
MI <- declare_model(
  N = 100,
  X = sort(rnorm(N)),
  couple = c(1:(N / 2), (N / 2):1),
  U = rnorm(N, sd = 0.1),
```

Table 17.1: Bad blocking leads to precision loss

design	bias	Standard Error
design_blocked	-0.001	0.164
design_complete	0.000	0.142

```

potential_outcomes(Y ~ Z + X * Z + U)
) +
declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))

design_complete <- MI +
declare_assignment(Z = complete_ra(N)) +
declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
declare_estimator(Y ~ Z)

design_blocked <- MI +
declare_assignment(Z = block_ra(blocks = couple)) +
declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
declare_estimator(Y ~ Z)

```

```
simulations <- simulate_designs(design_complete, design_blocked)
```

```
simulations <- simulate_designs(design_complete, design_blocked)
```

17.2.3 Connection of blocking to covariate adjustment

Choosing block random assignment over complete random assignment is a way to incorporate covariate information into the data strategy D . We can also incorporate covariate information into the answer strategy A for the same purpose (increasing precision), by controlling for covariates or otherwise conditioning on them when estimating the average treatment effect. In observational settings like the one we explore in Section 15.2, conditioning on covariates is used to block back-door paths to address confounding. Here, confounding is no problem – the treatment is assigned at random by design, so we do not need to control for covariates in order to decrease bias. Instead, we control for covari-

ates in order to reduce sampling variability.

Figure 17.4 illustrates this point. The sampling distribution under difference in means is shown on the top line and the sampling distribution under ordinary least squares ($Y \sim Z + X$) is shown on the bottom line. Estimates that are on the extreme ends of the distribution under difference-in-means are pulled in more tightly to center on the ATE. One interesting wrinkle this graph reveals is that covariate adjustment does not tighten up the estimates for assignments that exactly balance X – they only help the assignments that are slightly imbalanced.

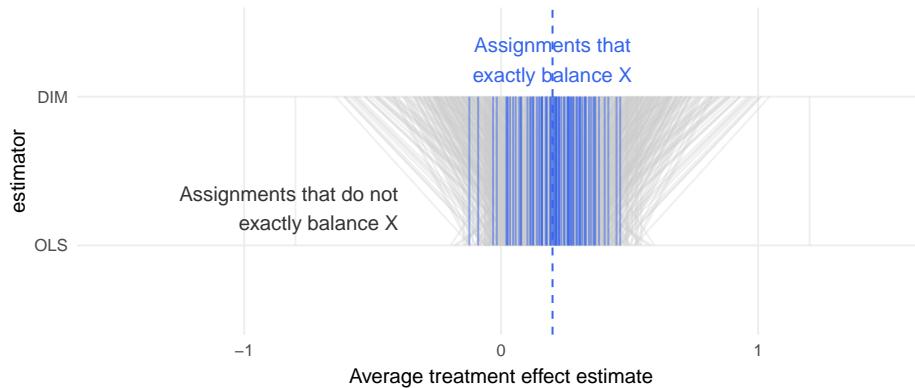


Figure 17.4: Impact of covariate adjustment on the sampling distribution

17.2.4 Simulation comparing blocking to covariate adjustment

Adjusting for pre-treatment covariates that are predictive of the outcome almost always increases precision; blocking on covariates that are predictive of the outcome almost always increases precision too. Another way of putting this idea is that covariate information can be incorporated in the answer strategy through covariate adjustment or in the data strategy through blocking and that in this way, the two procedures are approximately equivalent.

We'll now declare and diagnose four closely-related experimental designs. To begin, we describe a fixed population of 100 units with a binary covariate X and unobserved heterogeneity U . Potential outcomes are a function of the treatment Z and are correlated with X . Throughout this exercise, our inquiry is the ATE.

Declaration 17.3.

```
fixed_pop <-
  fabricate(
    N = 100,
```

```

X = rbinom(N, 1, 0.5),
U = rnorm(N),
potential_outcomes(Y ~ 0.2*Z + X + U)
)

MI <-
declare_model(data = fixed_pop) +
declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))

```

We have two answer strategies and two data strategies that we'll mix-and-match.

```

# Data strategies
complete_assignment <-
declare_assignment(Z = complete_ra(N = N)) +
declare_measurement(Y = reveal_outcomes(Y ~ Z))

blocked_assignment <-
declare_assignment(Z = block_ra(blocks = X)) +
declare_measurement(Y = reveal_outcomes(Y ~ Z))

# Answer strategies
unadjusted_estimator <- declare_estimator(Y ~ Z, inquiry = "ATE")
adjusted_estimator <- declare_estimator(Y ~ Z + X, model = lm_robust, inquiry = "ATE")

```

These combine to create four designs, which we then diagnose.

```

design_1 <- MI + complete_assignment + unadjusted_estimator
design_2 <- MI + blocked_assignment + unadjusted_estimator
design_3 <- MI + complete_assignment + adjusted_estimator
design_4 <- MI + blocked_assignment + adjusted_estimator

```

```
diagnose_designs(list(design_1, design_2, design_3, design_4))
```

The diagnosis shows that incorporating covariate information either in the data strategy or in the answer strategy yields similar gains to the true standard error. Relative to the canonical design (complete random assignment with difference-

Table 17.2: Comparison of block randomization to covariate adjustment

Data Strategy	Answer Strategy	True Standard Error	Average Estimated Standard Error
Complete Random Assignment	Difference-in-means	0.229	0.217
Block Random Assignment	Difference-in-means	0.168	0.218
Complete Random Assignment	Covariate Adjustment	0.190	0.181
Block Random Assignment	Covariate Adjustment	0.168	0.181

in-means), any of the alternatives represents an improvement. Blocking on X in the data strategy decreases sampling variability. Controlling for X in the answer strategy decreases sampling variability. Doing both – blocking on X and controlling for X – does not yield *additional* gains, but controlling for X is nevertheless appropriate when using a blocked design. The reason for this can be seen in the “average estimated standard error” diagnosand. If we block, but still use the difference-in-means estimator, the *estimated* standard errors do not decrease relative to complete random assignment. The usual Neyman variance estimator doesn’t “know” about the blocking. A number of fixes to this problem are available. You can, as we do in the simulation, control for the blocking variable in an OLS regression. Alternatively, you can use the “stratified” estimator that obtains block-level ATE estimates, then averages them together, weighting by block size. The stratified estimator has an associated standard error estimator – see Gerber and Green (2012) page 73-74. The stratified estimator is an instance of Principle 3.7: Seek M:I::D:A parallelism. Respecting the data strategy in the answer strategy (by adjusting for the blocking) brings down the estimated standard error as well.

17.2.5 Can controlling for covariates hurt precision?

Freedman (2008) critiques the practice of using OLS regression to adjust experimental data. While the difference-in-means estimator is unbiased for the average treatment effect, the covariate-adjusted OLS estimator exhibits a small sample bias (sometimes called “Freedman bias”) that diminishes quickly as sample sizes increase. More worrying is the critique that covariate adjustment can even hurt precision. Lin (2013) unpacks the circumstances under which this precision loss occurs and offers an alternative estimator that is guaranteed to be at least as precise as the unadjusted estimator. The trouble occurs when the correlation of covariates to outcomes is quite different in the treatment condition from in the control condition and when designs are strongly imbalanced in the sense of having large proportions of treated or untreated units. We refer the reader to this excellent and quite readable paper for details and the connection between covariate adjustment in randomized experiments and covariate adjustment in random sampling designs. In sum, the Lin estimator deals with the problem by performing covariate adjustment in each arm of the experiment separately, which is equivalent to the inclusion of a full set of treatment-by-

covariate interactions. In a clever bit of regression magic, Lin shows how first pre-processing the data by de-meaning the covariates renders the coefficient on the treatment regressor an estimate of the overall ATE. The `lm_lin` estimator in the `estimatr` package implements this pre-processing seamlessly.

Declaration 17.4 will help us to explore the precision of three estimators under a variety of circumstances. We want to understand the performance of the difference-in-means, OLS, and Lin estimators depending on how different the correlation between X and the outcome is by treatment arm, and depending on the fraction of units assigned to treatment.

Declaration 17.4.

```
prob = 0.5
control_slope = -1

design <-
  declare_model(N = 100,
    X = runif(N, 0, 1),
    U = rnorm(N, sd = 0.1),
    Y_Z_1 = 1*X + U,
    Y_Z_0 = control_slope*X + U
  ) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_assignment(Z = complete_ra(N = N, prob = prob)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, inquiry = "ATE", label = "DIM") +
  declare_estimator(Y ~ Z + X, model = lm_robust, inquiry = "ATE", label = "OLS") +
  declare_estimator(Y ~ Z, covariates = ~X, model = lm_lin, inquiry = "ATE", label = "Lin")
```

```
designs <- redesign(design,
  control_slope = seq(-1, 1, 0.5),
  prob = seq(0.1, 0.9, 0.1))

simulations <- simulate_designs(designs)
```

```
designs <- redesign(design,
  control_slope = seq(-1, 1, 0.5),
  prob = seq(0.1, 0.9, 0.1))
```

```
simulations <- simulate_designs(designs)
```

Figure 17.5 considers a range of designs under five possible models. The five models are described by the top row of facets. In all cases, the slope of the treated potential outcomes with respect to X is set to 1. All the way to the left, the slope with respect to the control potential outcomes is set to -1, and all the way to the right, is set to +1. The bottom row of facets shows the performance of three estimators along a range of treatment assignment probabilities.

When the control slope is -1, we can see Freedman's precision critique. The standard error of the OLS is *larger* than difference-in-means for many designs, though they coincide when the fraction treated is 50%. This problem persists in some form until the slope of the control potential outcome with respect to X gets close enough to the slope of the treated potential outcomes with respect to X .

All along this range, however, the Lin estimator dominates OLS and difference-in-means. Regardless of the fraction assigned to treatment and the model of potential outcomes, the Lin estimator achieves equal or better precision than either difference-in-means or OLS.

17.2.6 Summary

Covariate information can help experimenters increase the precision of their estimates. This precision gain is "free" if covariate information is easily available. If measuring covariates is costly, then experimenters face a tradeoff: should scarce resources be spent on increasing the sample size or on measuring covariate information? When covariates are especially predictive of outcomes, the measurement of covariates can be a good investment.

Covariate information can be included either in the data strategy as the basis for blocks or in the answer strategy as a control procedure. To a first approximation, we can achieve the same gains using either approach, though we take the view that, where possible, it would be preferable to incorporate covariates in the data strategy rather than the answer strategy, since inferences will be less dependent on the specifics of the estimator.

The incorporation of covariates almost always increases precision, but bad blocking or perverse control can cause decreases in precision. Bad blocks are easy to avoid if we block on covariates that are predictive of the outcome. Adverse consequences of regression adjustment can be easily sidestepped by adopting the Lin estimator as the default form of covariate adjustment.

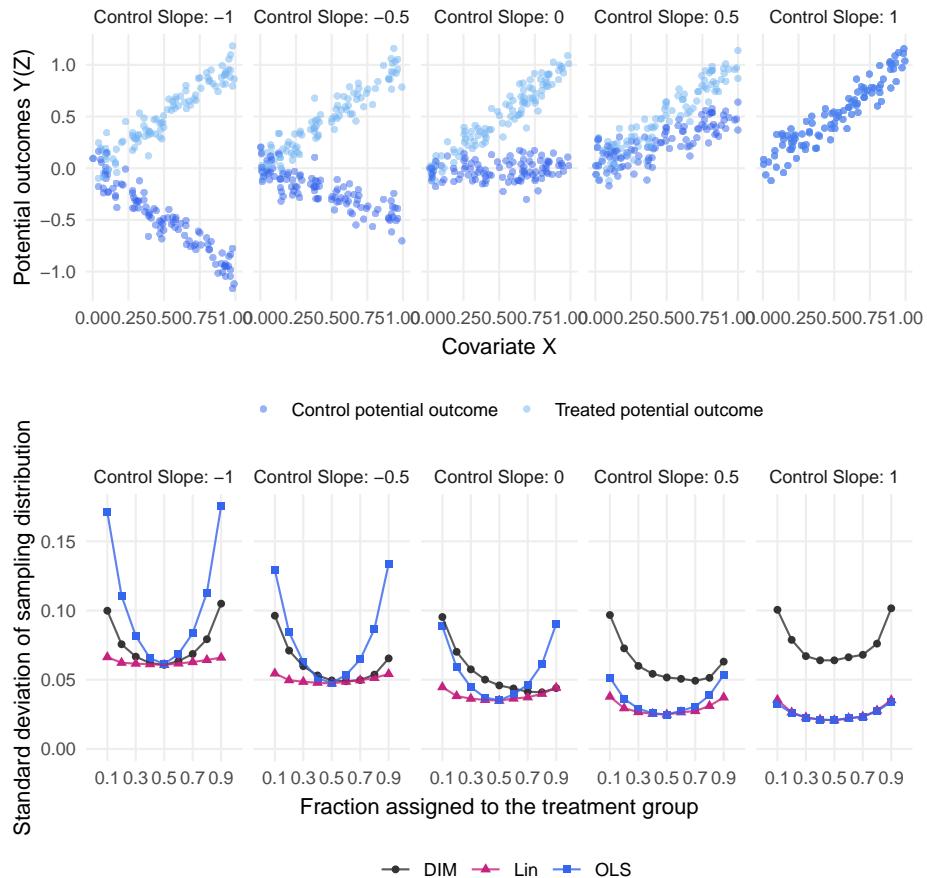


Figure 17.5: Performance of three estimators

17.3 Cluster-randomized experiments

We declare a cluster randomized trial in which subjects are assigned to treatment and control conditions in groups. We use design diagnosis to assess the reductions in variance of estimation that can be achieved from block randomization, examine possible downsides of block randomization, and compare strategies that randomize *ex ante* and that introduce controls *ex post*.

When whole groups of units are assigned to treatment conditions together, we say that the assignment procedure is clustered. A common example is an education experiment that is randomized at the classroom level. All students in a classroom are assigned to either treatment or control together; assignments do not vary within classroom. Clusters can be localities, like villages, precincts, or neighborhood. Clusters can be households if treatments are assigned at the household level.

Typically, cluster randomized trials exhibit higher variance than the equivalent individually-randomized trial. How much higher variance depends on a statistic that can be hard to think about, the intra-cluster correlation (ICC). The total variance can be decomposed into the variance of the cluster means $\sigma_{between}^2$ plus the individual variance of the cluster-demeaned outcome σ_{within}^2 . The ICC is a number between zero and one that describes the fraction of the total variance that is due to the between variance: $\frac{\sigma_{between}^2}{\sigma_{between}^2 + \sigma_{within}^2}$. If ICC equals one, then all units within a cluster express the same outcome, and all of the variation in outcomes is due to cluster-level differences. If ICC equals zero, then the cluster means are all identical, but the individuals vary within each cluster. When ICC is one, the effective sample size is equal to the number of clusters. When ICC is zero, the effective sample size is equal to the number of individuals. Since ICC is usually somewhere between these two values, we can see that clustering decreases the effective sample size from the number of individuals. The size of this decrease depends on how similar outcomes are within cluster compared to how similar outcomes are across clusters.

For these reasons clustered random assignment is not usually a desirable feature of a design, However sometimes it is useful or even *necessary* for logistical or ethical reasons for subjects to be assigned to together in groups.

To demonstrate the consequences of clustering, Declaration 17.5 shows a design in which both the untreated outcome Y_{Z_0} and the treatment effect τ_{i_t} exhibit intra-cluster correlation.

The inquiry is the average treatment effect over individuals which can be defined without reference to the clustered structure of the data.

The data strategy employs clustered random assignment. We highlight two features of the cluster assignment.

First we highlight that the clustered nature of the data does not itself *call* for clustered assignment. In principle one could assign at the individual level or subgroup level even if outcomes are correlated within groups.

Second, surprisingly, random assignment of clusters to conditions does not guarantee unbiasedness of outcomes when clusters are of unequal size. (Middleton, 2008; Imai, King and Nall, 2009). The bias stems from the possibility that potential outcomes could be correlated with cluster size. With uneven cluster sizes, the total number of units (the denominator in the mean estimation) in each group bounces around from assignment to assignment. Since the expectation of a ratio is not, in general, equal to the ratio of expectations, any dependence between cluster size and potential outcomes will cause bias. We can address this problem by blocking clusters into groups according to cluster size. If all clusters in a block are of the same size, then the overall size of the treatment group will remain stable from assignment to assignment. For this reason the design below uses clustered assignment blocked on cluster size.

Declaration 17.5.

```

ICC <- 0.9

design <-
  declare_model(
    cluster =
      add_level(
        N = 10,
        cluster_size = rep(seq(10, 50, 10), 2),
        cluster_shock =
          scale(cluster_size + rnorm(N, sd = 5)) * sqrt(ICC),
        cluster_tau = rnorm(N, sd = sqrt(ICC))
      ),
    individual =
      add_level(
        N = cluster_size,
        individual_shock = rnorm(N, sd = sqrt(1 - ICC)),
        individual_tau = rnorm(N, sd = sqrt(1 - ICC)),
        Y_Z_0 = cluster_shock + individual_shock,
        Y_Z_1 = Y_Z_0 + cluster_tau + individual_tau
      )
  ) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_assignment(
    Z = block_and_cluster_ra(clusters = cluster, blocks = cluster_size)
  ) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z,
    clusters = cluster,
    inquiry = "ATE")

designs <- redesign(design, ICC = seq(0.1, 0.9, by = 0.4))

```

```

diagnoses <- diagnose_designs(designs)

```

Figure 17.6 shows the sampling distribution of the difference-in-means estimator under cluster random assignment at five levels of intra-cluster correlation ranging from 0.1 to 0.9.

The top row of panels plots the treatment effect on the vertical axis and the untreated potential outcome on the horizontal axis. Clusters of units are circled.

At low levels of ICC, the circles all overlap, because the differences across clusters are smaller than the differences within cluster. At high levels of ICC, the differences across clusters are more pronounced than differences within cluster. The bottom row of panels shows that the sampling distribution of the difference-in-means estimator spreads out as the ICC increases. At low levels of ICC, the standard error is small; at high levels the standard error is high.

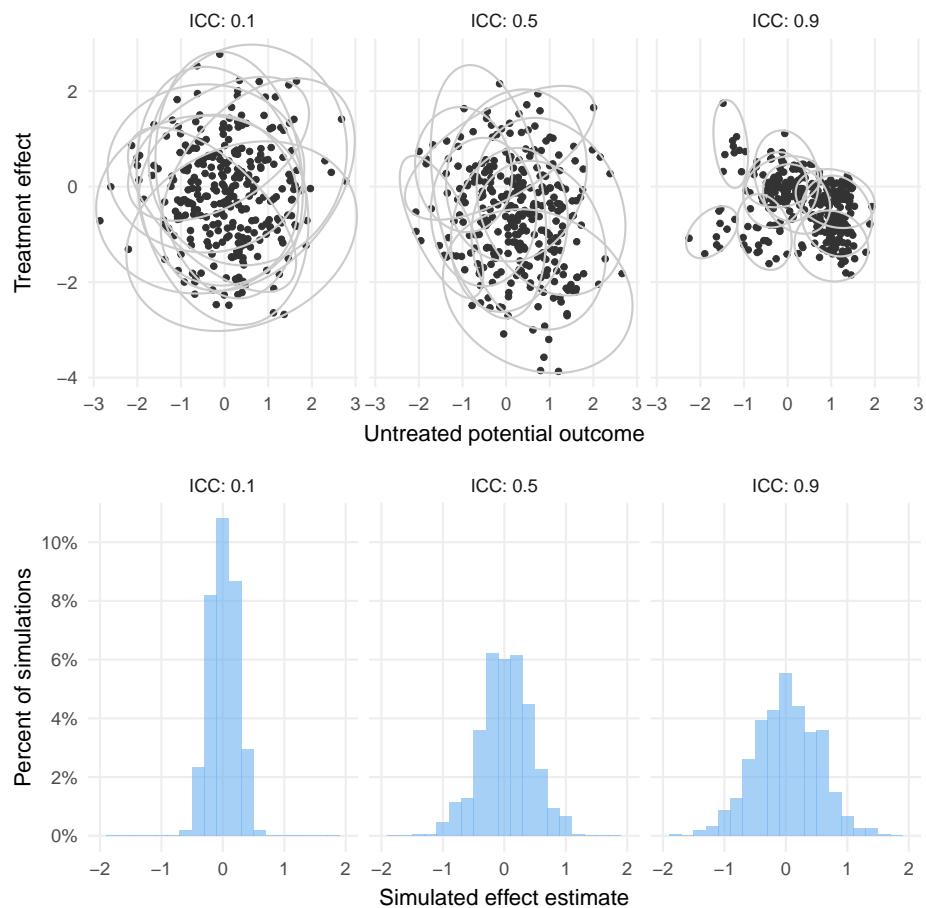


Figure 17.6: Sampling distribution under different ICCs

This diagnosis clarifies the *costs* of cluster assignment. These costs are greatest when there are few clusters and when units within clusters have similar potential outcomes. Diagnosis can be further used to compare these costs to advantages and assess the merits of variations in the design that seek to alter the number or size of clusters.

17.4 Subgroup designs

We declare and diagnose a design that is targeted at understanding the difference in treatment effects between subgroups. The design combines a sampling strategy that ensures reasonable numbers within each group of interest and a blocking assignment strategy to minimize variance.

Subgroup designs are experimental designs that have been tailored to a specific inquiry, the difference-in-CATEs. A CATE is a “conditional average treatment effect,” or the average treatment effect conditional on membership in some group. A difference-in-CATEs is just the difference between two CATEs.

For example, studies of political communication often have the *difference* in response to a party cue by subject partisanship as the main inquiry, since Republican subjects tend to respond positively to a Republican party cue, whereas Democratic subjects tend to respond negatively.

Subgroup designs share much in common with factorial designs, discussed in detail in Section 17.5. The main source of commonality is the answer strategy for the difference-in-CATEs inquiry. In subgroup designs and factorial designs, the usual approach is to inspect the interaction term from an OLS regression. The two designs differ because in the subgroup design, the difference-in-CATEs is a descriptive difference. We don’t randomly assign partisanship, so we can’t attribute the difference in response to treatment to partisanship, which could just be marker for the true causes of the difference in response. In the factorial design, we randomize the levels of all treatments, so the differences-in-CATEs carry with them a causal interpretation.

Since we don’t randomly assign membership in subgroups, how can we optimize the design to target the difference-in-CATEs? Our main data strategy choice comes in sampling. We need to obtain sufficient numbers of both groups in order to generate sharp enough estimates of each CATE, the better to estimate their difference. For example, at the time of this writing, many sources of convenience samples (Mechanical Turk, Lucid, Prolific, and many others) appear to underrepresent Republicans, so researchers sometimes need to make special efforts to increase the their numbers in the eventual sample.

Declaration 17.6 describes a fixed population of 10,000 units, among whom people with $X = 1$ are relatively rare (only 20%). In the `potential_outcomes` call, we build in both baseline differences in the outcome, and also oppositely signed responses to treatment. Those with $X = 0$ have a CATE of 0.1 and those with $X = 1$ have a CATE of $0.1 - 0.2 = -0.1$. The true difference-in-CATEs is therefore 20 percentage points.

If we were to draw a sample of 1000 at random, we would expect to yield only 200 people with $X = 1$. Here we improve upon that through stratified sampling. We deliberately sampling 500 units with $X = 1$ and 500 with $X = 0$, then block randomly assigned the treatment by X .

Declaration 17.6.

```

fixed_pop <-
  fabricate(N = 10000,
            X = rbinom(N, 1, 0.2),
            potential_outcomes(
              Y ~ rbinom(N, 1,
                          prob = 0.7 + 0.1 * Z - 0.4 * X - 0.2 * Z * X))
            )

total_n <- 1000
n_x1 <- 500
# Note: n_x2 = total_n - n_x1

design <-
  declare_population(data = fixed_pop) +
  declare_inquiry(
    CATE_X1 = mean(Y_Z_1[X == 1] - Y_Z_0[X == 1]),
    CATE_X0 = mean(Y_Z_1[X == 0] - Y_Z_0[X == 0]),
    diff_in_CATEs = CATE_X1 - CATE_X0
  ) +
  declare_sampling(
    S = strata_rs(strata = X, strata_n = c(total_n - n_x1, n_x1))
  ) +
  declare_assignment(Z = block_ra(blocks = X)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z + X + Z * X,
                   term = "Z:X",
                   inquiry = "diff_in_CATEs")

```

To show the benefits of stratified sampling for experiments, we redesign over many values of under- and over-sampling units with $X = 1$, holding the total sample size fixed at 1000. The top panel Figure 17.7 shows the distribution difference-in-CATE estimates at each size of the $X = 1$ group. When very small or very large fractions of the total sample have $X = 1$, the variance of the estimator is much larger than when the two groups the same size.

The bottom panel of the figure shows how three diagnosands change over the oversampling design parameter. Bias is never a problem – even small subgroups will generate unbiased difference-in-CATE estimates. As suggested by the top panel, the standard error is minimized in the middle and is largest at the extremes. Likewise, statistical power is maximized in the middle, but drops off surprisingly quickly as we move away from evenly balanced recruitment.

```
designs <- redesign(design, n_x1 = seq(20, 980, by = 96))
simulations <- simulate_designs(designs)
```

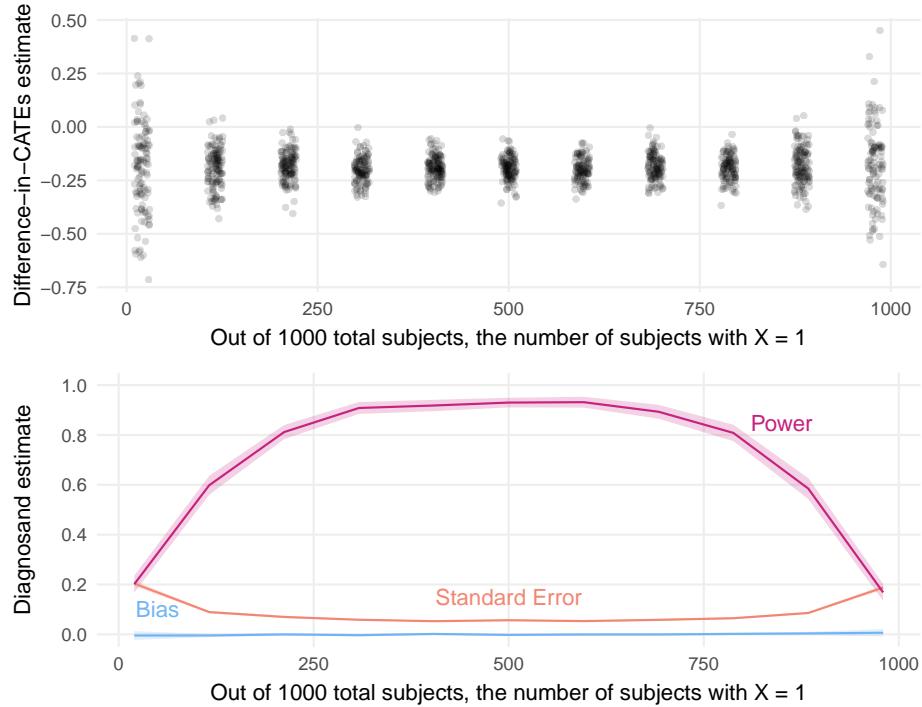


Figure 17.7: Design performance given different sampling strategies.

Exercises

- Suppose it costs \$2 to recruit an $X = 1$ and \$1 to recruit an $X = 0$ and we have a budget of \$1000. Modify the redesign to find the power-maximizing financially feasible design.

Further reading

- Druckman and Kam (2011) makes the point that a main difficulty when using convenience samples (in their case, student samples) is a lack of variation on crucial moderating variables.

17.5 Factorial experiments

We declare and diagnose a canonical factorial design in which two different treatments are crossed. The design allows for unbiased estimation of a wide range of estimands including conditional effects and interaction effects. We highlight the difficulty of achieving statistical power for interaction terms and the risks of treating a difference between a significant conditional effect and a nonsignificant effect as itself significant.

In factorial experiments, researchers randomly assign the level of not just one treatment, but multiple treatments. The prototypical factorial design is a “two-by-two” factorial design in which factor 1 has two levels and factor 2 has two levels as well. Similarly, a “three-by-three” factorial design has two factors, each of which has three levels. We can entertain any number of factors with any number of levels. For example, a “two-by-three-by-two” factorial design has three factors, two of which have two levels and one of which has three levels. Conjoint experiments are (Section 16.3) are highly factorial, often including six or more factors with two or more levels each.

Factorial designs can help researchers answer many inquiries, so it is crucial to design factorials with a particular set in mind. Let’s consider the two-by-two case, which is complicated enough. Let’s call the first factor Z_1 and the second factor Z_2 , each of which can take on the values of zero or one. Considering only average effects, this design can support seven separate inquiries:

1. the average treatment effect (ATE) of Z_1 ,
2. the ATE of Z_2 ,
3. the conditional average treatment effect (CATE) of Z_1 given Z_2 is 0,
4. the CATE of Z_1 given Z_2 is 1
5. the CATE of Z_2 given Z_1 is 0
6. the CATE of Z_2 given Z_1 is 1
7. The difference-in-CATEs of Z_1 given Z_2 is 1 and of Z_1 given Z_2 is 0
8. Which is numerically equivalent to the difference-in-CATEs of Z_2 given Z_1 is 1 and of Z_2 given Z_1 is 0

The reason we distinguish between the ATE of Z_1 versus the CATEs of Z_1 depending on the level of Z_2 is that the two factors may “interact.” When factors interact, the effects of Z_1 are heterogeneous in the sense that they are different depending on Z_2 . We often care about the difference-in-CATEs inquiry because of theoretical settings in which the effects of one treatment are supposed to depend on the level of another treatment.

However, if we are not so interested in the difference-in-CATEs, then factorial experiments have another good justification – we can learn about the ATEs of each treatment for half price, in the sense that we apply treatments to the same subject pool using the same measurement strategy. Conjoint experiments are a kind of factorial design (discussed in Section 16.3) often target average treatments effects that average over the levels of the other factors.

Here we declare a factorial design with two treatments and a normally distributed outcome variable. We imagine that the CATE of Z1 given Z2 is zero is equal to 0.2 standard units, the CATE of Z2 given Z1 is zero is equal to 0.1, and the interaction of the two is 0.1 as well.

Declaration 17.7. 2x2 Factorial design

```
CATE_Z1_Z2_0 <- 0.2
CATE_Z2_Z1_0 <- 0.1
interaction <- 0.1
N <- 1000

design <-
  declare_model(
    N = N,
    U = rnorm(N),
    potential_outcomes(Y ~ CATE_Z1_Z2_0 * Z1 +
                        CATE_Z2_Z1_0 * Z2 +
                        interaction * Z1 * Z2 + U,
                        conditions = list(Z1 = c(0, 1),
                                           Z2 = c(0, 1)))) +
  declare_inquiry(
    CATE_Z1_Z2_0 = mean(Y_Z1_1_Z2_0 - Y_Z1_0_Z2_0),
    CATE_Z1_Z2_1 = mean(Y_Z1_1_Z2_1 - Y_Z1_0_Z2_1),
    ATE_Z1 = 0.5 * CATE_Z1_Z2_0 + 0.5 * CATE_Z1_Z2_1,

    CATE_Z2_Z1_0 = mean(Y_Z1_0_Z2_1 - Y_Z1_0_Z2_0),
    CATE_Z2_Z1_1 = mean(Y_Z1_1_Z2_1 - Y_Z1_1_Z2_0),
    ATE_Z2 = 0.5 * CATE_Z2_Z1_0 + 0.5 * CATE_Z2_Z1_1,

    diff_in_CATEs_Z1 = CATE_Z1_Z2_1 - CATE_Z1_Z2_0,
    #equivalently
    diff_in_CATEs_Z2 = CATE_Z2_Z1_1 - CATE_Z2_Z1_0
  ) +
  declare_assignment(Z1 = complete_ra(N),
                     Z2 = block_ra(Z1)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z1 + Z2)) +
  declare_estimator(Y ~ Z1, subset = (Z2 == 0),
                   inquiry = "CATE_Z1_Z2_0", label = 1) +
  declare_estimator(Y ~ Z1, subset = (Z2 == 1),
                   inquiry = "CATE_Z1_Z2_1", label = 2) +
  declare_estimator(Y ~ Z2, subset = (Z1 == 0),
                   inquiry = "CATE_Z2_Z1_0", label = 3) +
  declare_estimator(Y ~ Z2, subset = (Z1 == 1),
```

```

inquiry = "CATE_Z2_Z1_1", label = 4) +
declare_estimator(Y ~ Z1 + Z2, term = c("Z1", "Z2"),
                   inquiry = c("ATE_Z1", "ATE_Z2"), label = 5) +
declare_estimator(Y ~ Z1 + Z2 + Z1*Z2, term = "Z1:Z2",
                   inquiry = c("diff_in_CATEs_Z1", "diff_in_CATEs_Z2"),
                   label = 6)

```

We now redesign this factorial over many sample sizes, considering the statistical power for each of the inquiries. Figure 17.8 shows that depending on the inquiry, the statistical power of this design can vary dramatically. The average treatment effect of Z_1 is relatively large at 0.25 standard units, so power is above the 80% threshold at all the sample sizes we consider. The ATE of Z_2 is smaller, at 0.15 standard units, so power is lower, but not dramatically so. Both ATEs use all N data points, so power is manageable for the average effects. The conditional average effects generally fare worse, mainly because each is estimated on only half the sample. The power for the 0.1 standard unit difference-in-CATEs is abysmal at all sample sizes considered here.

```

designs <- redesign(design, N = seq(500, 3000, 500))
simulations <- simulate_design(designs, sims = 100)

```

17.5.1 Avoiding misleading inferences

The very poor power for the difference-in-CATEs sometimes leads researchers to rely on a different answer strategy for considering whether the effects of Z_1 depend on the level of Z_2 . Sometimes, researchers will consider the statistical significance of each of Z_1 's CATEs separately, then conclude the CATEs are “different” if the effect is significant for one CATE but not the other. This is bad practice.

Here we diagnose over the true values of the Z_1 ATE, setting the true interaction term to zero. Our diagnosis question will be, how frequently do we conclude the two CATEs are different, using two different strategies. The first is the usual approach, i.e., we consider the statistical significant of the interaction term. The second considers whether one, but not the other, of the two CATE estimates is significant.

```

designs <- redesign(

```

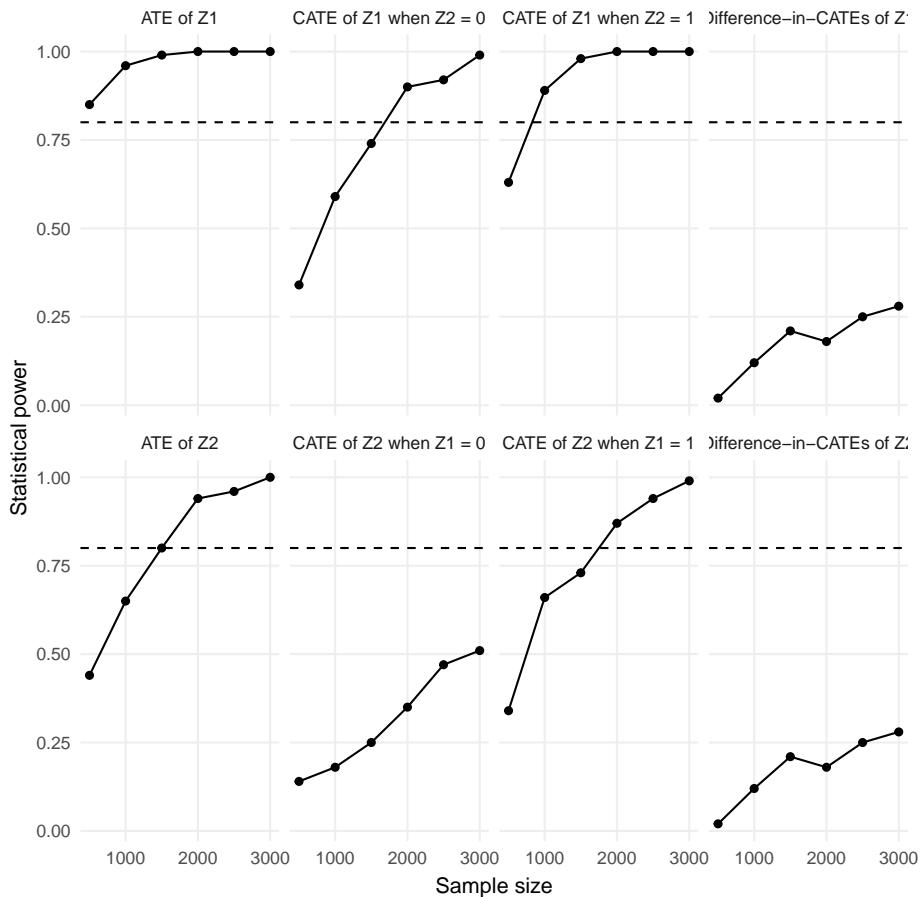


Figure 17.8: Power for factorial inquiries

```

design,
CATE_Z1_Z2_0 = seq(0, 0.5, 0.05),
CATE_Z2_Z1_0 = 0.2,
interaction = 0
)

simulations <- simulate_design(designs, sims = 500)

```

Figure 17.9 shows that the error rate when we consider the statistical significance of the interaction term is nominal. Only 5% of the time do we falsely reject the null that the difference-in-CATEs is zero. But when we claim “treatment effect heterogeneity!” when one CATE is significant but not the other, we make

egregious errors. When the true (constant) average effect of Z1 approaches 0.2, we falsely conclude that there heterogeneity nearly 50% of the time!

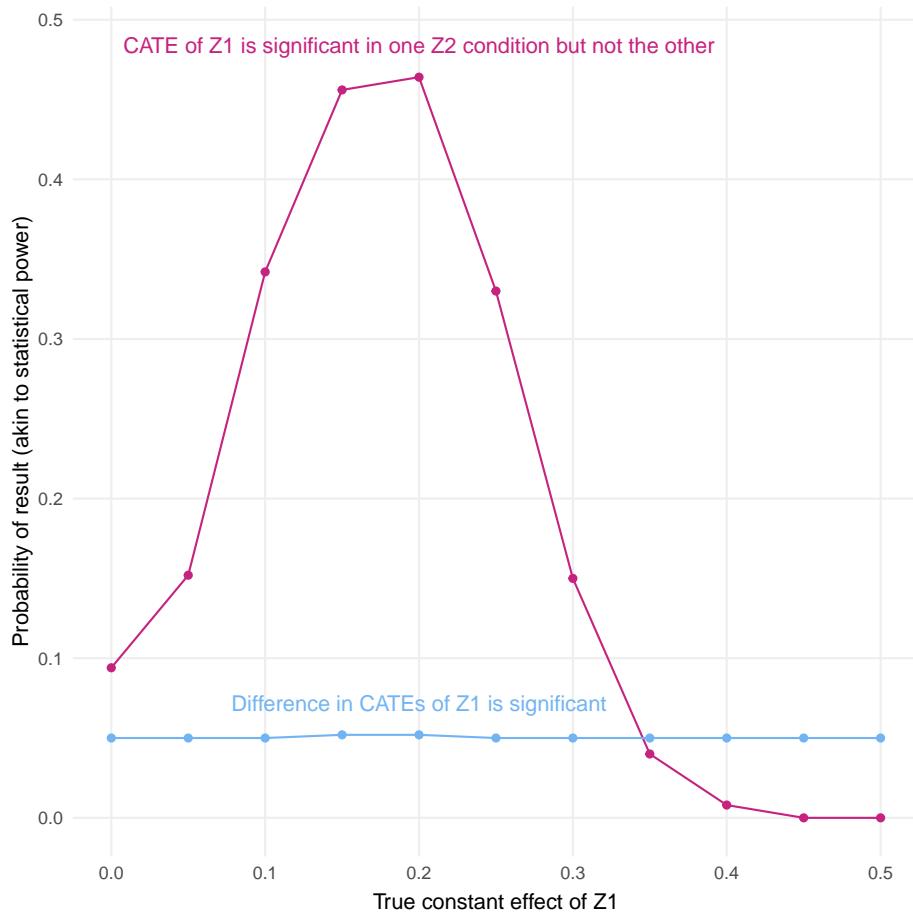


Figure 17.9: False conclusions of heterogeneity

Exercises

1. With 1000 subjects, how big does the interaction term need to be to achieve 80% power?
2. Holding the interaction at 0.1, how many subjects do we need for the interaction term to achieve 80% power?

17.6 Encouragement designs

In many experimental settings, we can't we can't *require* units we assign to take treatment to actually take treatment. Nor can we require units assigned to the control group *not* to take treatment. Instead, we have to content ourselves with "encouraging" units assigned to the treatment group to take treatment and "encouraging" units assigned to the control group not to.

Encouragements are often only partially successful. Some units assigned to treatment refuse treatment and some units assigned to control find a way to obtain treatment after all. In these settings, we say that experiments encounter "noncompliance." This section will describe the most common approach to the design and analysis of encouragement trials, and will point out potential pitfalls along the way.

Any time a data strategy entails contacting subjects in order to deliver a treatment like a bundle of information or some good, noncompliance is a potential problem. Emails go undelivered, unopened, and unread. Letters get lost in the mail. Phone calls are screened, text messages get blocked, direct messages on social media are ignored. People don't come to the door when you knock, either because they aren't home or they don't trust strangers. Noncompliance can affect noninformational treatments as well: goods may be difficult to deliver to remote locations, subjects may refuse to participate in assigned experimental activities, or research staff might simply fail to respect the realized treatment schedule out of laziness or incompetence.

Experimenters who anticipate noncompliance should make compensating adjustments to their research designs (relative to the canonical two arm design). These adjustments ripple through M , I , D , and A .

17.6.1 Changes to the model

The biggest change to M is developing beliefs about compliance types, also called "principal strata" (Frangakis and Rubin, 2002). In a two-arm trial, subjects can be one of four compliance types, depending on how their treatment status responds to their treatment assignment. The four types are described in Table 17.3. $D_i(Z = 0)$ is a potential outcome – it is the treatment status that unit i would express if assigned to control. Likewise, $D_i(Z = 1)$ is the treatment status that unit i would express if assigned to treatment. These potential outcomes can take each take on a value of 0 or 1, so their intersection allows for four types. For Always-takers, D_i is equal to 1 regardless of the value of Z – they always take treatment. Never-takers are the opposite – D_i is equal to 0 regardless of the value of Z . For Always-takers and Never-takers, assignment to treatment *does not change* whether they take treatment.

Compliers are units that take treatment if assigned to treatment and do not take treatment if assigned to control. Their name "compliers" connotes that something about their disposition as subjects makes them "compliant" or otherwise

docile, but this connotation is misleading. Compliance types are generated by the confluence of subject behavior and data strategy choices. Whether or not a subject answers the door when the canvasser comes calling is a function many things, including whether the subject is at home and whether they open the door to canvassers. Data strategies that attempt to deliver treatment in the evenings and on weekends might generate more (or different) compliers than those that attempt treatment during working hours.

Table 17.3: Compliance types

Compliance Type	$D_i(Z_i = 0)$	$D_i(Z_i = 1)$
Never-taker	0	0
Complier	0	1
Defier	1	0
Always-taker	1	1

The last compliance type to describe are defiers. These strange birds refuse treatment when assigned to treatment, but find a way to obtain treatment when assigned to control. Whether or not “defiers” exist turns out to be a consequential assumption that must be made in the model. We have good reason to believe that defiers are rare – assignment to treatment almost always has a positive average effect on treatment take-up.

A unit’s compliance type often not possible to observe directly. Subjects assigned to the control group who take treatment ($D_i(0) = 1$) could be defiers or always-takers. Subjects assigned to the treatment group who do not take treatment ($D_i(1) = 0$) could be defiers or never-takers. Our inability to be sure of compliance types is another facet of the fundamental problem of causal inference. Even though a subject’s compliance type (with respect to a given design) is a stable trait, it is defined by how the subject would act in multiple counterfactual worlds. We can’t tell what type a unit is because we would need to see whether they take treatment when assigned to treatment and also when assigned to control.

17.6.2 Changes to the inquiry

The inclusion of noncompliance and compliance types to the model also necessitate changes to the inquiry. Always-takers and Never-takers present a real problem for causal inference. Even with the power to randomly assign, we can’t change what treatments these units take. As a result, we don’t get to learn about the effects of treatment among these groups. Even if our inquiry were the average effect of treatment among the never-takers, the experiment (as designed) would not be able to generate empirical estimates of it.³ Our inquiry has to fall

³We write “as designed” because compliance types are defined with respect to a particular design. If it were possible to induce the never-takers to take treatment (i.e., under a different data strategy,

back to the average effects among those units that whose treatment status we can successfully encourage to change – the compliers.

This inquiry is called the complier average causal effect (the CACE). It is defined as $E[Y_i(1) - Y_i(0)|d_i(1) > d_i(0)]$. Just like the average treatment effect, it refers to an average over individual causal effects, but this average is taken over a specific subset of units, the compliers. Compliers are the only units for whom $d_i(1) > d_i(0)$, because for compliers, $d_i(1) = 1$ and $d_i(0) = 0$. When assignments and treatments are binary, the CACE is mathematically identical to the local average treatment effect (LATE) described in Chapter @ref(sec: p3iv). Whether we write CACE or LATE sometimes depends on academic discipline, with LATE being more common among economists. An advantage of “CACE” over “LATE” is that it is specific about which units the effect is “local” to – it is local to the compliers.

When experiments encounter noncompliance, the CACE is *usually* the most important inquiry for theory, since it refers to the average effect of the causal variable, at least for a subset of the units in the study. However, two other common inquiries are important to address here as well.

The first is the intention-to-treat (ITT) inquiry, which is defined as $E[Y_i(D_i(Z = 1), Z = 1) - Y_i(D_i(Z = 0), Z = 0)]$. The encouragement itself Z has a total effect on Y that is mediated in whole or in part by the treatment status. Sometimes the ITT is the policy-relevant inquiry, since it describes what would happen if a policy maker implemented the policy in the same way as the experiment, *inclusive* of noncompliance. Consider an encouragement design to study the effectiveness of a tax webinar on tax compliance. Even if the webinar is very effective among people willing to watch it (the CACE is large), the main trouble faced by the policy maker will be getting people to sit through the webinar. The ITT describes the average effect of *inviting* people to the webinar, which could be quite small if very few people are willing to join.

The second additional inquiry is the compliance rate, sometimes referred to as the ITT_D . It describes the average effect of assignment on treatment, and is written $E[(D_i(Z = 1) - D_i(Z = 0))]$. A small bit of algebra shows that the ITT_D is equal to the fraction of the sample that are compliers minus the fraction that are defiers.

These three inquiries are tightly related. Under five very important assumptions (described below), we can write:

$$\text{CACE} = \frac{\text{ITT}}{\text{ITT}_D}$$

A derivation of this relationship is given in Section 15.4 on instrumental variables. The five assumptions described in that section are identical to the assump-

these units might be compliers), this inquiry would not *necessarily* be out of reach.

tions required here. In an experimental setting, “exogeneity of the instrument” is guaranteed by features of the data strategy. Since we use random assignment, we know for sure that the “instrument” (the encouragement) is exogenous. Excludability of the instrument refers to the idea that the effect of the encouragement on the outcome is fully mediated by the treatment. This assumption could be violated if the mere act of encouragement changes outcomes. Stated differently, if never-takers or always-takers reveal *different* potential outcomes in treatment and control ($Y_i(D_i(Z = 1), Z = 1) \neq Y_i(D_i(Z = 0), Z = 0)$), it must be because encouragement itself changes outcomes. Non-interference in this setting means that units’ treatment status and outcomes do not depend on the assignment or treatment status of other units. In an experimental context, the assumption of monotonicity rules out the existence of defiers. This assumption is often made plausible by features of the data strategy (perhaps it is impossible for those who are not assigned to treatment to obtain treatment) or features of the model (“defiant” responses to encouragement are behaviorally unlikely). The final assumption – nonzero effect of the instrument on the treatment – can also be assured by features of the data strategy. In order to learn about the effects of treatment, data strategies must successfully encourage at least some units to take treatment.

17.6.3 Changes to the data strategy

When experimenters expect that noncompliance will be a problem, they should take steps to mitigate that problem in the data strategy. Sometimes doing so just means trying harder: investigating the patterns of noncompliance, attempting to deliver treatment on multiple occasions, or offering subjects incentives for participation. “Trying harder” is about turning more subjects into compliers by choosing a data strategy that encounters less noncompliance.

A second important change to the data strategy is the explicit measurement of treatment status as distinct from treatment assignment. For some designs, measuring treatment status is easy. We just record which units were treated and which were untreated. But in some settings, measuring compliance is trickier. For example, if treatments are emailed, we might never know if subjects read the email. Perhaps our email service will track read receipts, in which case one facet of this measurement problem is solved. We won’t know, however, how many subjects read the subject line – and if the subject line contains any treatment information, then even subjects who don’t click on the email may be “partially” treated. Our main advice is to measure compliance in the most conservative way: if treatment emails bounce altogether, then subjects are not treated.

In multi-arm trials or with continuous rather than binary instruments, noncompliance becomes a more complex problem to define and address through the data strategy and answer strategy. We must define complier types according to all of the (potentially-infinite) possible treatment conditions. For multiarm trials, the complier types for the first treatment may not be the same for the

second treatment; in other words, units will comply at different rates to different treatments. Apparent differences in complier average treatment effects and intent-to-treat effects, as a result, may reflect not differences in treatment effects but different rates of compliance.

17.6.4 Changes to the answer strategy

Estimation of the CACE is not as straightforward subsetting the analysis to compliers. A plug-in estimator of the CACE with good properties takes the ratio of the ITT estimate to the ITT_d estimate. Since the ITT_d must be a number between zero and one, this estimator “inflates” the ITT by the compliance rate. Another way of thinking about this is that the ITT is deflated by all the never-takers and always-takers, among whom the ITT is by construction 0, so instead of “inflating”, we are “re-inflating” the ITT to the level of the CACE. Two-stage least squares in which we instrument the treatment with the random assignment is a numerically equivalent procedure when treatment and assignments are binary. Two-stage least squares has the further advantage of being able to seamlessly incorporate covariate information to increase precision.

Two alternative answer strategies are biased and should be avoided. An “as-treated” analysis ignores the encouragement Z and instead compares units by their revealed treatment status D . This procedure is prone to bias because those who come to be treated may differ systematically from those who do not. The “per protocol” analysis is similarly biased. It drops any unit that fails to comply with its assignment, but those who take treatment in the treatment group (compliers and always-takers) may differ systematically from those who do not take treatment in the control group (compliers and never-takers). Both the “as-treated” and “per-protocol” answer strategies suffer from a special case of post-treatment bias, wherein conditioning on a post-assignment variable (treatment status) essentially de-randomizes the study.

Declaration 17.8 elaborates the model to include the four compliance types, setting the share of defiers to zero to match the assumption of monotonicity. It imagines that the potential outcomes of the outcomes Y with respect to the treatment D are different for each compliance type, reflecting the idea that compliance type could be correlated with potential outcomes. The declaration also links compliance type to the potential outcomes of the treatment D with respect to the randomized encouragement Z . We then move on to declaring two inquiries (the CACE and the ATE) and three answer strategies (two-stage least squares, as-treated analysis, and per-protocol analysis).

Declaration 17.8.

```
design <-  
  declare_model(
```

```

N = 100,
type =
  rep(c("Always-Taker", "Never-Taker", "Complier", "Defier"),
       c(0.2, 0.2, 0.6, 0.0)*N),
U = rnorm(N),
# potential outcomes of Y with respect to D
potential_outcomes(
  Y ~ case_when(
    type == "Always-Taker" ~ -0.25 - 0.50 * D + U,
    type == "Never-Taker" ~ 0.75 - 0.25 * D + U,
    type == "Complier" ~ 0.25 + 0.50 * D + U,
    type == "Defier" ~ -0.25 - 0.50 * D + U
  ),
  conditions = list(D = c(0, 1))
),
# potential outcomes of D with respect to Z
potential_outcomes(
  D ~ case_when(
    Z == 1 & type %in% c("Always-Taker", "Complier") ~ 1,
    Z == 1 & type %in% c("Never-Taker", "Defier") ~ 0,
    Z == 0 & type %in% c("Never-Taker", "Complier") ~ 0,
    Z == 0 & type %in% c("Always-Taker", "Defier") ~ 1
  ),
  conditions = list(Z = c(0, 1))
)
) +
declare_inquiry(
  ATE = mean(Y_D_1 - Y_D_0),
  CACE = mean(Y_D_1[type == "Complier"] - Y_D_0[type == "Complier"])) +
declare_assignment(Z = conduct_ra(N = N)) +
declare_measurement(D = reveal_outcomes(D ~ Z),
                     Y = reveal_outcomes(Y ~ D)) +
declare_estimator(
  Y ~ D | Z,
  model = iv_robust,
  inquiry = c("ATE", "CACE"),
  label = "Two stage least squares"
) +
declare_estimator(
  Y ~ D,
  model = lm_robust,
  inquiry = c("ATE", "CACE"),
  label = "As treated"
)

```

```

) +
declare_estimator(
  Y ~ D,
  model = lm_robust,
  inquiry = c("ATE", "CACE"),
  subset = D == Z,
  label = "Per protocol"
)

```

Figure 17.10 represents the encouragement design as a DAG. No arrows lead into Z , because the treatment was randomly assigned. The compliance type C , the assignment Z , and unobserved heterogeneity U conspire to set the level of D . The outcome Y is affected by the treatment D of course, but also by compliance type C and unobserved heterogeneity U . The required exclusion restriction that Z only affect Y through D is represented by the lack of an arrow from Z to Y . The deficiencies of the as-treated and per-protocol analysis strategies can be learned from the DAG as well. D is a collider, so conditioning on it will open up back-door paths between Z , C , and U , leading to bias of unknown direction and magnitude.

The design diagnosis shows the sampling distribution of the three answer strategies and compares it to two potential inquiries: the complier average causal effect and the average treatment effect. Our preferred method, two-stage least squares, is *biased* for the ATE. Because we can't learn about the effects of treatment among never-takers or always-takers, any estimate of the true ATE will be necessarily be prone to bias, except in the happy circumstance that never-takers and always-takers happen to be just like compliers.

Two-stage least squares does a much better job of estimating the complier average causal effect. Even though the sampling distribution is wider than those for the per-protocol and as-treated analysis, it is at least centered on a well-defined inquiry. By contrast, the other two answer strategies are biased for either target.

```
diagnosis <- diagnose_design(design, sims = sims, bootstrap_sims = b_sims)
```

17.7 Placebo-controlled experiments

In common usage, the notion of a placebo is a treatment that carries with it everything about the bonafide treatment – except the active ingredient. We're used to thinking about placebos in terms of the “placebo effect” in medical trials.

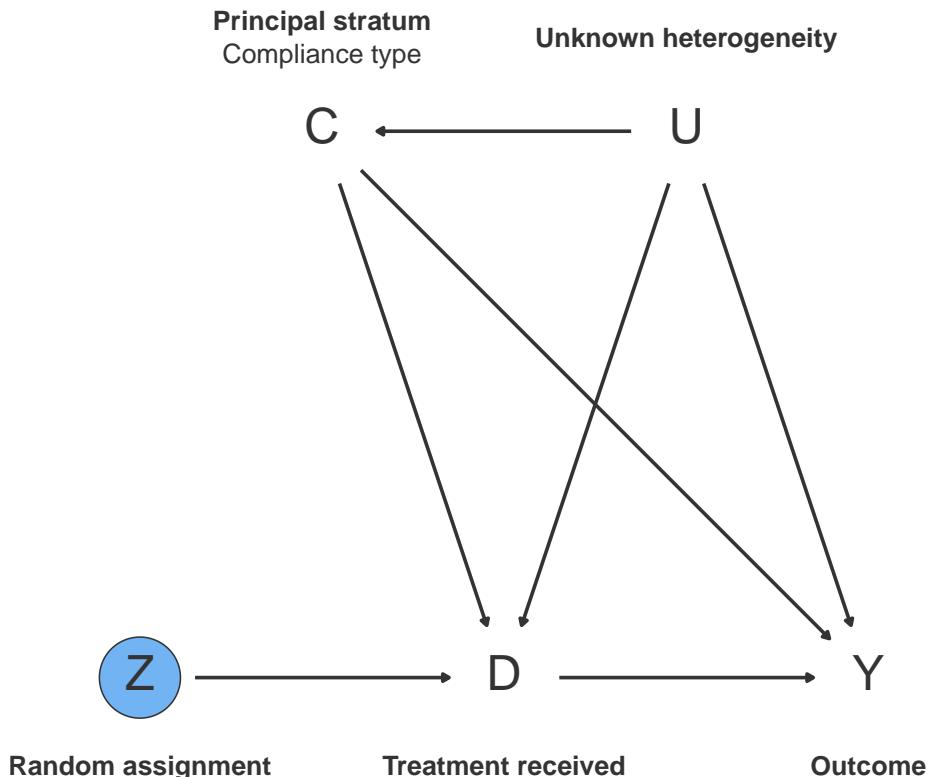


Figure 17.10: DAG of the encouragement design

Some portion of the total effect of the actual treatment is due the mere act of getting any treatment, so the administration of placebo treatments can difference this portion off. Placebo controlled designs abound in social sciences too (see Porter and Velez (2021)) for similar purposes. Media treatments often work through a bundle of priming effects and new information; a placebo treatment might include only the prime but not the information. The main use of placebos is to difference off the many small excludability violations involved in bundled treatments the better to understand the main causal variable of interest.

In this chapter, we study the use of placebos for a different purpose: to combat the negative design consequences of noncompliance in experiments. As described in the previous chapter, a challenge for experiments that encounter noncompliance is that we do not know for sure who the compliers are. Compliers are units that would take treatment if assigned to treatment, but would not do so if assigned to control. Compliers are different from always-takers and never-takers in that assignment to treatment actually changes which potential outcome they reveal.

In the placebo-controlled design, we attempt to deliver a real treatment to the

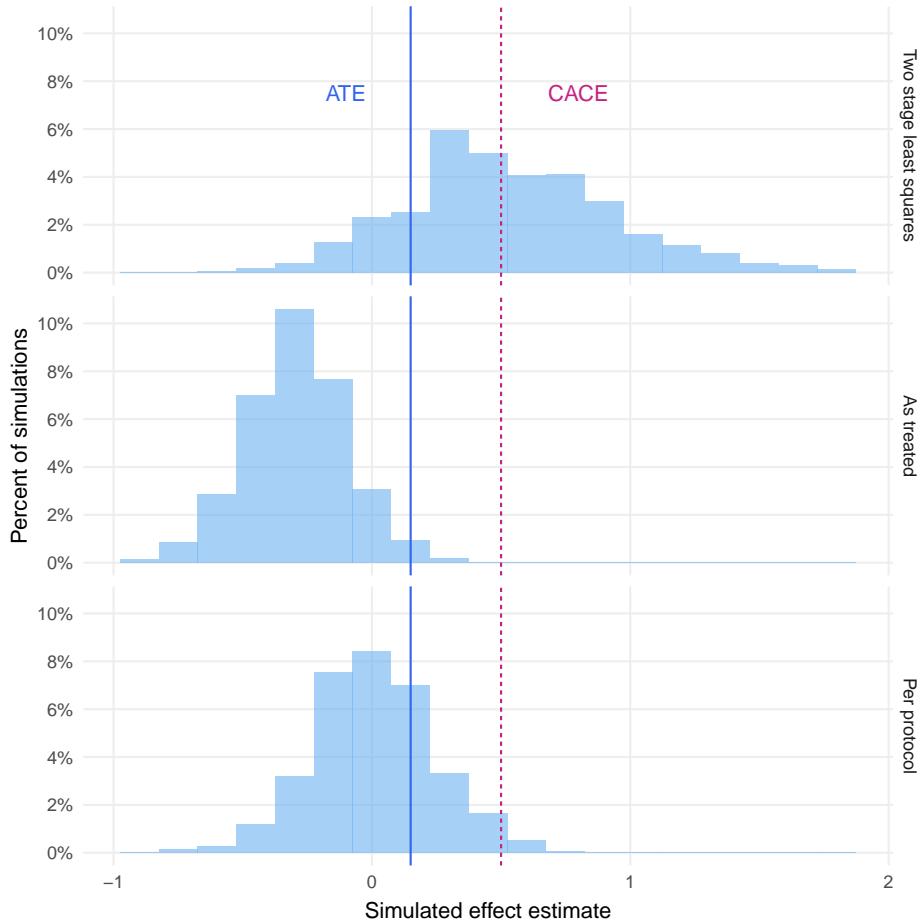


Figure 17.11: Sampling distribution of three estimators

treatment group and a placebo treatment to the placebo group, then we conduct our analysis among those units that accept either treatment. This design solves two problems at once. First, it lets us answer a *descriptive* question: “Who are the compliers?” Second, it lets answer a *causal* causal question: “What is the average effect of treatment among compliers?”

Employing a placebo control can seem like an odd design choice – you go to all the effort of contacting a unit but at the very moment you get in touch, you deliver a placebo message instead of the treatment message. It turns out that despite this apparent waste, the placebo-controlled design can often lead to more precise estimates than the standard encouragement design. Whether it does or not depends in large part on the underlying compliance rate.

Declaration 17.9 actually includes two separate designs. Here we'll directly

compare the standard encouragement design to the placebo controlled design. They have identical theoretical halves, so we'll just declare those once, before declaring the specifics of the empirical strategies for each design.

Declaration 17.9.

```
compliance_rate <- 0.2

MI <-
  declare_model(
    N = 400,
    type = sample(x = c("Never-Taker", "Complier"),
                  size = N,
                  prob = c(1 - compliance_rate, compliance_rate),
                  replace = TRUE),
    U = rnorm(N),
    # potential outcomes of Y with respect to D
    potential_outcomes(
      Y ~ case_when(
        type == "Never-Taker" ~ 0.75 - 0.25 * D + U,
        type == "Complier" ~ 0.25 + 0.50 * D + U
      ),
      conditions = list(D = c(0, 1))
    ),
    # potential outcomes of D with respect to Z
    potential_outcomes(
      D ~ if_else(Z == 1 & type == "Complier", 1, 0),
      conditions = list(Z = c(0, 1))
    )
  ) +
  declare_inquiry(CACE = mean(Y_D_1[type == "Complier"] - Y_D_0[type == "Complier"]))
)
```

Here again are the data and answer strategies for the encouragement design (simplified from the previous chapter to focus on the one-sided compliance case). We conducted a random assignment among all units, then reveal treatment status and outcomes according to the potential outcomes declared in the model. The two-stage least squares estimator operates on all N units to generate estimates of the CACE.

```
encouragement_design <-
  MI +
  declare_assignment(Z = conduct_ra(N = N)) +
```

```

declare_measurement(D = reveal_outcomes(D ~ Z),
                    Y = reveal_outcomes(Y ~ D)) +
  declare_estimator(
    Y ~ D | Z,
    model = iv_robust,
    inquiry = "CACE",
    label = "2SLS among all units"
  )

```

By contrast, here are the data and answer strategies for the placebo-controlled design. In a typical canvassing experiment setting, the expensive part is sending canvassing teams to each household, regardless of whether a treatment or a placebo message is delivered when the door opens. So in order to keep things “fair” across the placebo controlled and encouragement designs, we’re going to hold fixed the number of treatment attempts – the sampling step subsets to the same $N/2$ that we will attempt. Then among that subset, we conduct a random assignment to treatment or placebo. When we attempt to deliver the placebo or the treatment, we will either succeed or fail, which gives us a direct measure of whether a unit is a complier. This measurement is represented in the `declare_measurement` step where an observable `X` now corresponds to compliance type. We conduct our estimation directly conditioning on the subset of the sample we have measured to be compliers.

```

placebo_controlled_design <-
  MI +
  declare_sampling(S = complete_rs(N)) +
  declare_assignment(Z = conduct_ra(N = N)) +
  declare_measurement(X = if_else(type == "Complier", 1, 0),
                      D = reveal_outcomes(D ~ Z),
                      Y = reveal_outcomes(Y ~ D)) +
  declare_estimator(
    Y ~ Z,
    subset = X == 1,
    model = lm_robust,
    inquiry = "CACE",
    label = "OLS among compliers"
  )

```

We diagnose both the encouragement design and the placebo-controlled design over a range of possible levels of noncompliance, focusing on the standard de-

viation of the estimates (the standard error) as our main diagnosand. Figure 17.12 shows the results of the diagnosis. At high levels of compliance, the standard encouragement design actually outperforms the placebo-controlled design. But when compliance is low, the placebo controlled design is preferred. Which is preferable in any particular scenario will depend on the compliance rate as well as other design features like the total number of attempts and the fraction treated.

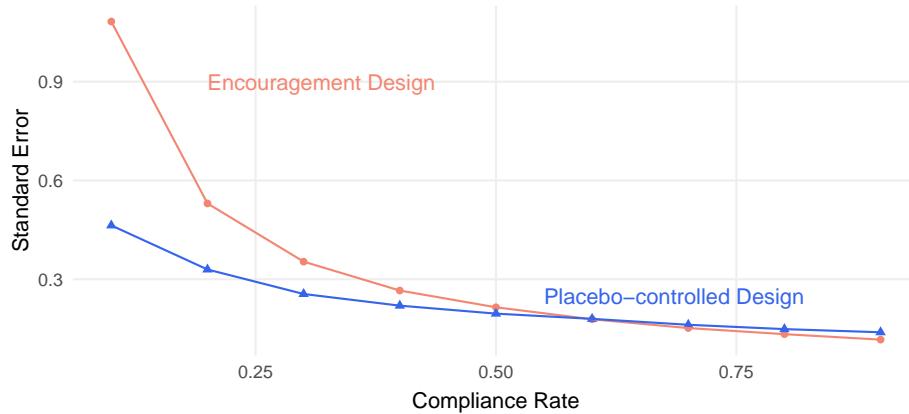


Figure 17.12: Comparison of the placebo controlled design to a standard encouragement design

Further reading

- See Wilke, Green and Cooper (2020) for a placebo-controlled design that incorporates features of the crossover design described in Section 17.9.

17.8 Stepped-wedge experiments

We often face an ethical dilemma in allocating treatments to some units but not others, since we would rather not withhold treatment from anyone. However, practical constraints often make it impossible to allocate treatments to everyone at the same time. In these circumstances, a stepped-wedge experiment can help. Under a stepped-wedge design, we follow an allocation rule that randomly assigns a portion of units to treatment in each of one or more periods, and then in a final period, everyone is allocated treatment. We conduct posttreatment measurement after each period except for the last one. Figure 17.13 illustrates the allocation procedure. A common design is allocating one third to treatment in the first period, an additional third in the second period, and the remaining third in the final third period.

Declaration 17.10.

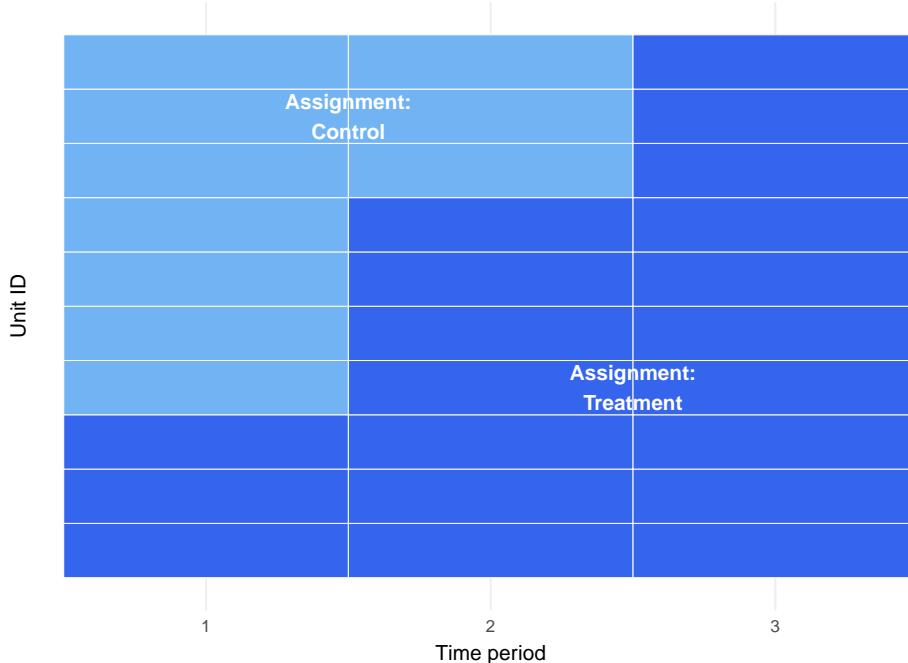


Figure 17.13: Illustration of random assignment in a stepped-wedge design.

Our model describes unit-specific effects, time-specific effects, and time trends in the potential outcomes. In the data strategy, we allocate treatment for 100 units in three time periods, following the 1/3, 2/3, 3/3 allocation rule.

We assign treatment by randomly assigning the wave each unit will receive treatment. We use cluster assignment at the unit level because the data is at the unit-period level. We then transform this treatment variable into a unit-period treatment indicator, if the time period is at or after the treatment wave.

Our inquiry is the average treatment effect among time periods *before* the last period. In the stepped-wedge design, we don't obtain information about the control potential outcome in the final period. Our answer strategy also only uses the data from the first two periods (in reality, we probably would not collect outcome data after the last period for this reason). We fit a two-way fixed effects regression model by periods and units with standard errors clustered at the unit level.

We show in the difference-in-differences design entry in Section 15.3 that under a very similar model, the two-way fixed effects estimator is biased for the average treatment effect on the treated in the presence of treatment effect by time interactions. The differences between the designs are twofold. Here, we

randomize treatment, rather than using observational data with confounded treatment assignment, so we do not need to make the parallel trends assumption. Our diagnosis below will show no bias in estimating the average treatment effect with the two-way fixed effects estimator.

Declaration 17.11.

```
design <-
  declare_model(
    units = add_level(
      N = 100,
      U_unit = rnorm(N)
    ),
    periods = add_level(
      N = 3,
      time = 1:max(periods),
      U_time = rnorm(N),
      nest = FALSE
    ),
    unit_period = cross_levels(
      by = join(units, periods),
      U = rnorm(N),
      potential_outcomes(
        Y ~ scale(U_unit + U_time + time + U) + effect_size * Z
      )
    )
  ) +
  declare_assignment(
    wave = cluster_ra(clusters = units, conditions = 1:max(periods)),
    Z = if_else(time >= wave, 1, 0)
  ) +
  declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0), subset = time < max(time)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z, fixed_effects = ~ periods + units,
                    clusters = units,
                    subset = time < max(time),
                    inquiry = "ATE", label = "TWFE")
```

17.8.1 When to use a stepped wedge experiment

Compared to the equivalent two-arm randomized experiment, a stepped-wedge experiment involves the same number of units, but more treatment (all versus half) and more measurement (all units are measured at least twice). The

decision of whether to adopt the stepped-wedge design, then, rides on your budget, the relative costs of measurement and treatment, ethical and logistical constraints such as the imperative to treat all units, and your beliefs about effect sizes relative to measurement noise in your outcomes.

We compare the stepped-wedge design to a two-arm randomized experiment with varying sample sizes to assess these tradeoffs. In particular, we examine a study with the same number of units, which would be the relevant comparison if the main constraint is you cannot increase the number of units in the study. The second comparison is a two-arm experiment with double the number of units, which would be the right comparison if you can increase the number of units but have a fixed budget for measurement and treatment allocation. We summarize each design in terms of the number of study units, the number that are treated, and the number of unit measurements taken.

Table 17.4: Design parameters in the comparison between stepped-wedge and two-arm experimental designs.

Design	N	m treated	n measurements
Stepped-wedge	100	100	200
Two-arm v1	100	50	100
Two-arm v2	200	100	200

We declare a comparable two-arm experimental design below, with the wrinkle being that the estimand is slightly different by necessity. In the stepped-wedge design, we target the average treatment effect averaging over all periods up to the penultimate one, because there is no information about the control group from the last period. In a single period design, by its nature, we cannot average over time. We would obtain a biased answer if we targeted an out-of-sample time period. The average treatment effect we target is the current-period ATE for the period that is chosen. We cannot extrapolate beyond that if treatment effects vary over time. If you expect time heterogeneity in effects, you may *not* want to use a stepped-wedge design but instead design a new experiment that efficiently targets the conditional average treatment effects within each period. Then you could describe both the average effect and how effects vary over time.

Declaration 17.12.

```
design_single_period <-
  declare_model(
    N = n_units,
    U_unit = rnorm(N),
    U = rnorm(N),
```

```

    effect_size = effect_size,
    potential_outcomes(Y ~ scale(U_unit + U) + effect_size * Z)
) +
declare_assignment(Z = complete_ra(N, m = n_units / 2)) +
declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0)) +
declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
declare_estimator(Y ~ Z, inquiry = "ATE", label = "TWFE")

```

We plot power curves for the three comparison designs in Figure 17.14. The top line (blue dashed) is the 200-unit study, which is preferred in terms of power, and by a considerable margin. That design involves the same amount of measurement and treatment as the stepped-wedge so may be the same cost. Ethical constraints such as you must treat all units or logistical constraints such as there are only 100 eligible units to study would be the only reason under these beliefs about the model to adopt the stepped wedge design. However, the stepped-wedge design here strictly dominates a two-arm experiment with only 100 units. In that design, there is less measurement (half) and fewer units are treated (half). However, it delivers much less power. The two-arm may in some cases be logically less complicated.

17.9 Crossover experiments

In empirical research, we can only ever access one potential outcome for any given unit, because only one can be revealed at a time. As a result, we cannot estimate the treatment effect for any individual, which would require knowing both the treatment potential outcome and the untreated potential outcome. We only get one. Experiments typically address this fundamental problem of causal inference by randomly assigning *different* units to reveal *different* potential outcomes (see Principal 3.5).

An appealing possibility is to do the same thing, but *within* units. If we could assign a unit to reveal its treated potential outcome in one period and its untreated potential outcome in the next period, we could subtract the two realized outcomes and obtain that unit's individual treatment effect. With that effect in hand for all units, we could efficiently explore how treatment effects vary across units, as well as the average effect.

The crossover design is founded on that possibility. In the cross-over design, we block randomize units to receive treatment and control over two periods (the blocks are the units). Each unit either receives treatment first then control or control first then treatment. We collect endline outcome data after the first period and after the second period. The treatment that is randomized in the de-

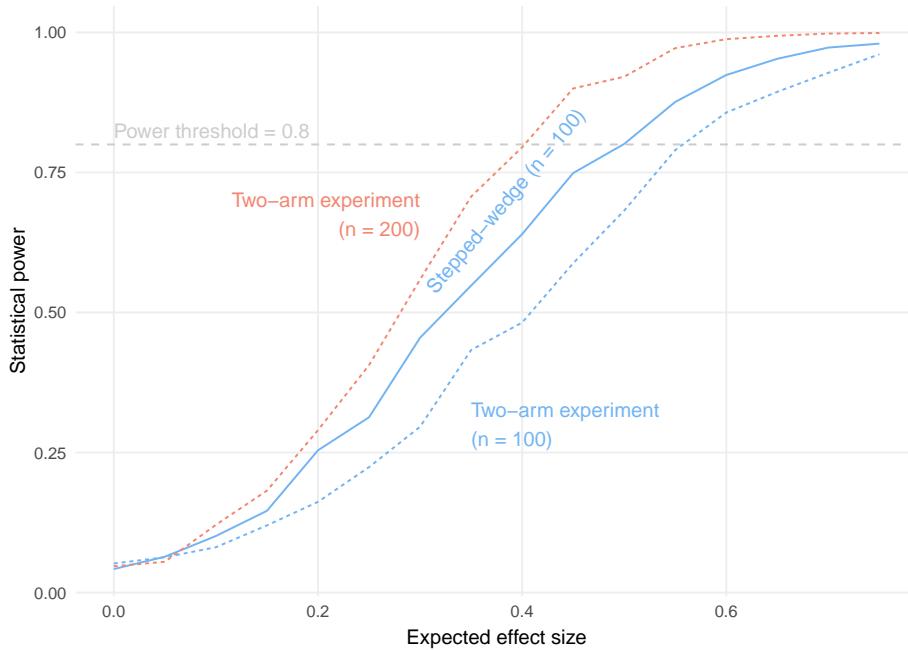


Figure 17.14: Power analysis of three designs: stepped wedge with 100 units and 1/3-1/3-1/3 allocation, two-arm experiment with 100 units, and two-arm experiment with 200 units.

sign must be one in which once you receive the treatment, you can be untreated again in future periods – it is not sticky. A treatment that could be randomized is providing a time-delimited voucher in one period that you cannot use in the next period. There still could be *effects* of treatment in the first period (“carry-over” effects), but in the second period you are untreated. An example of a treatment that could not be randomized in this design would be a fixed asset like a plough; once you receive it, you have it, so in the second period you still have the plough and are in this sense still treated. For such sticky treatments, a stepped-wedge design might be more appropriate (Section 17.8).

In order to declare a cross-over design, we need to redefine potential outcomes, because of the possibility of carry-over effects. We allow potential outcomes to not only be a function of whether they are treated now, in the current period, but whether they were treated in the past period. In particular, we consider three potential outcomes: what outcome a unit would have if untreated in the past period and untreated now, if untreated in the past but treated now, and treated in the past but untreated now. We will not be able to reveal the outcome if you were treated in the past and treated now, the fourth of the two-by-two of current and past treatment statuses and we don’t need it for the inquiry, so we don’t define that one in the declaration. We define our potential outcomes such

that there is a 0.2 treatment effect, and there is the possibility of a carryover effect to the outcomes for units that were treated in the preceding period. In the redesign, we explore no carryover effects (`carryover = 0`) up through large carry-over effects representing the treatment effect remaining the same into the next period for treated units (`carryover = 1`).

We make several other changes for this design from a standard two-arm experiment. The potential outcome redefinition means that we need to redefine our average treatment effect inquiry too: which two outcomes are we differencing? We choose to focus on the two that involve an untreated period beforehand, and so the average difference is between if you are treated now and untreated now. We measure outcomes that are a function of both current and past treatment. Our answer strategy is an OLS regression with fixed effects for units and standard errors clustered on unit. We explore in the exercises why clustering is needed.

Declaration 17.13.

```
design <-
  declare_model(
    units = add_level(
      N = 100,
      U_unit = rnorm(N, sd = 5)
    ),
    periods = add_level(
      N = 2,
      time = 1:2,
      U_time = rnorm(N),
      nest = FALSE
    ),
    unit_periods = cross_levels(
      by = join(units, periods),
      U = rnorm(N),
      Y_Z_0_Zlag_0 = U_time + U_unit + U,
      Y_Z_1_Zlag_0 = Y_Z_0_Zlag_0 + 0.2,
      Y_Z_0_Zlag_1 = Y_Z_0_Zlag_0 + 0.2 * carryover
    )
  ) +
  declare_inquiry(
    ATE_untreated_before = mean(Y_Z_1_Zlag_0 - Y_Z_0_Zlag_0)
  ) +
  declare_assignment(
    Z = block_ra(blocks = units, prob = 0.5),
    Zlag = if_else(time == 2 & Z == 0, 1, 0)
  ) +
```

```

declare_measurement(Y = reveal_outcomes(Y ~ Z + Zlag)) +
declare_estimator(
  Y ~ Z,
  cluster = units,
  fixed_effects = ~units,
  model = lm_robust,
  inquiry = "ATE_untreated_before"
)

```

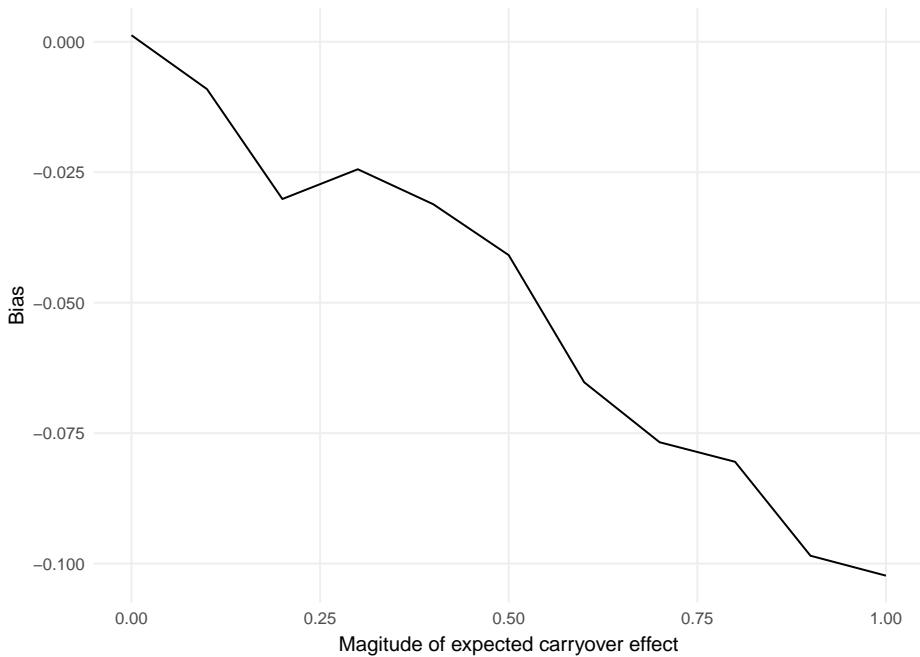


Figure 17.15: Bias in estimates of the average treatment effect if untreated before as a function of the magnitude of carryover effects from none (far left) to carryover effects the same size as the average treatment effect if untreated.

In Figure 17.15, we show the results of the redesign exercise. When there are no carryover effects, the design is unbiased for the average treatment effect if untreated before. However, as carryover effects increase, bias increases. The bias does not reach the magnitude of the estimand, in fact it is about half. The reason is that half the data are assigned to the control first then treated sequence. For these data, there is no bias because in the first period they reveal the untreated then untreated potential outcome and in the second period they reveal

the untreated then treated potential outcome. Thus, they have no risk of carry-over effects. These data are half the data, so the bias comes from the other half that move from treated to untreated.

Further reading

- See Wilke, Green and Cooper (2020) for an elaborated crossover design that incorporates features of the placebo-controlled design described in Section 17.7.

17.10 Randomized saturation experiments

We study most treatments at an isolated, atomized, individualistic level. We define potential outcomes with respect to a unit's own treatment status, ignoring the treatment status of all other units in the study. Accordingly, our inquiries tend to be averages of individual-level causal effects and our data strategies tend to assign treatments at the individual level as well. All of the experimental designs we have considered to this point have been of this flavor.

However, when the potential outcome that a unit reveals depends on the treatment status of other units, then we have to make adjustments to every part of the design. We have to redefine the model M to specify what potential outcomes are possible. Under a no-spillover model, we might only have the treated and untreated potential outcome $Y_i(1)$ and $Y_i(0)$. But under spillover models, we have to expand the set of possibilities. For, we might imagine that unit i 's potential outcomes can be written as a function of their own treatment status and that of their housemate, unit j : $Y(Z_i, Z_j)$. We have to redefine our inquiry I with respect to those reimagined potential outcomes. The average treatment effect is typically defined as $E[Y_i(1) - Y_i(0)]$, but if $Y_i(1)$ and $Y_i(0)$ no longer exist, we need to choose a new inquiry, like the average direct effect of treatment when unit j is not treated: $E[Y_i(1, 0) - Y_i(0, 0)]$. We have to alter our data strategy D so that the randomization procedure produces healthy samples of all of the potential outcomes involved in the inquiries, and we have to amend our answer strategy A to account for the important features of the randomization protocol.

We divide up our investigation of experimental designs to learn about spillovers into two sets. This chapter addresses randomized saturation designs, which are appropriate when we can exploit a hierarchical clustering of subjects into groups within which spillover can occur but across which spillover can't occur. The next chapter addresses experiments over networks, which are appropriate when spillover occurs over geographic, temporal, or social networks.

The randomized saturation design (sometimes called the partial population design, as in Baird et al. (2018)) is purpose-built for scenarios in we have good reason to imagine that a units potential outcomes depend on the fraction of treated units within the same cluster. For example, we might want to consider the fraction of people within a neighborhood assigned to receive a vaccine:

a person's health outcomes could easily depend on whether two-thirds or one-third of neighbors have been treated.

In the model, we now have to define potential outcomes with respect to both the individual level treatment and also the saturation level. We can imagine a variety of different kinds of potential outcomes functions. Consider the vaccine example, imagining a 100% effective vaccine against infection. Directly treated individuals never contract the illness, but the probability of infection for untreated units depends on the fraction who are treated nearby. If the treatment is a persuasive message to vote for a particular candidate, which might imagine that direct treatment is ineffective when only a few people around you hear the message, but becomes much more effective when many people hear the message at the same time. The main challenge in developing intuitions about complex interactions like this is articulating the discrete potential outcomes that each subject could express, then reasoning about the plausible values for each potential outcome.

The randomized saturation design is a factorial design of sorts, and like any factorial design can support a wide range of inquiries. We can describe the average effect of direct treatment at low saturation, at high saturation, the average of the two, or the difference between the two. Similarly, we could describe the average effect of high versus low saturation among the untreated, among the treated, the average of the two, or the difference between the two. In some settings, all eight of these inquiries might be appropriate to report, in others just a subset.

The design employs a two-stage data strategy. First, pre-defined clusters of units are randomly assigned to treatment saturation levels, for example 25% or 75%. Then, in each clusters, individual units are assigned to treatment or control with probabilities determined by their clusters' saturation level. The main answer strategy complication is that now there are two levels of randomization that must be respected. The saturation of treatment varies at the cluster level, so whenever we are estimating saturation effects, we have to cluster standard errors at the level saturation was assigned. The direct treatments are assigned at the individual level, so we do not need to cluster.

Declaration 17.14 describes 50 groups of 20 individuals each. We imagine one source of unobserved variation at the group level (the `group_shock`) and another at the individual level (the `individual_shock`). We built potential outcomes in which the individual and saturation treatment assignments each have additive (non-interacting) effects, though more complex potential outcomes functions are of course possible. We choose two inquiries in particular: the conditional average effect of saturation among the untreated and the conditional average effect of treatment when saturation is low.

We can learn about the effects of the dosage of indirect treatment by comparing units with the same individual treatment status across the levels of dosage. For example, we could compare untreated units across the 25% or 75% satura-

tion clusters. We can also learn about the direct effects of treatment at either saturation level, e.g., the effect of treatment when saturation is low. We use difference-in-means estimators of both inquiries, subsetted and clustered appropriately.

Declaration 17.14.

```
design <-
  declare_model(
    group = add_level(N = 50, group_shock = rnorm(N)),
    individual = add_level(
      N = 20,
      individual_shock = rnorm(N),
      potential_outcomes(
        Y ~ 0.2 * Z + 0.1 * (S == "low") + 0.5 * (S == "high") +
          group_shock + individual_shock,
        conditions = list(Z = c(0, 1),
                           S = c("low", "high"))
      )
    )
  ) +
  declare_inquiry(
    CATE_S_Z_0 = mean(Y_Z_0_S_high - Y_Z_0_S_low),
    CATE_Z_S_low = mean(Y_Z_1_S_low - Y_Z_0_S_low)
  ) +
  declare_assignment(
    S = cluster_ra(clusters = group,
                   conditions = c("low", "high")),
    Z = block_ra(blocks = group,
                  prob_unit = if_else(S == "low", 0.25, 0.75))
  ) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z + S)) +
  declare_estimator(
    Y ~ S,
    model = difference_in_means,
    subset = Z == 0,
    term = "Shigh",
    clusters = group,
    inquiry = "CATE_S_Z_0",
    label = "Effect of high saturation among untreated"
  ) +
  declare_estimator(
    Y ~ Z,
    model = difference_in_means,
```

```

subset = S == "low",
blocks = group,
inquiry = "CATE_Z_S_low",
label = "Effect of treatment at low saturation"
)

```

```
simulations <- simulate_design(design)
```

The diagnosis plot in Figure 17.16 shows the sampling distribution of the two estimators with the value of the relevant inquiry overlaid. Both estimators are unbiased for their targets, but the thing to notice from this plot is that the estimator of the saturation inquiry is far more variable than the estimator of the direct treatment inquiry. Saturation is by its nature a group-level treatment, so must be assigned at a group level. The clustered nature of the assignment to saturation level brings extra uncertainty. When designing randomized saturation experiments, researchers should be aware that we typically have much better precision for individually-randomized treatments than cluster-randomized treatments, and should plan accordingly.

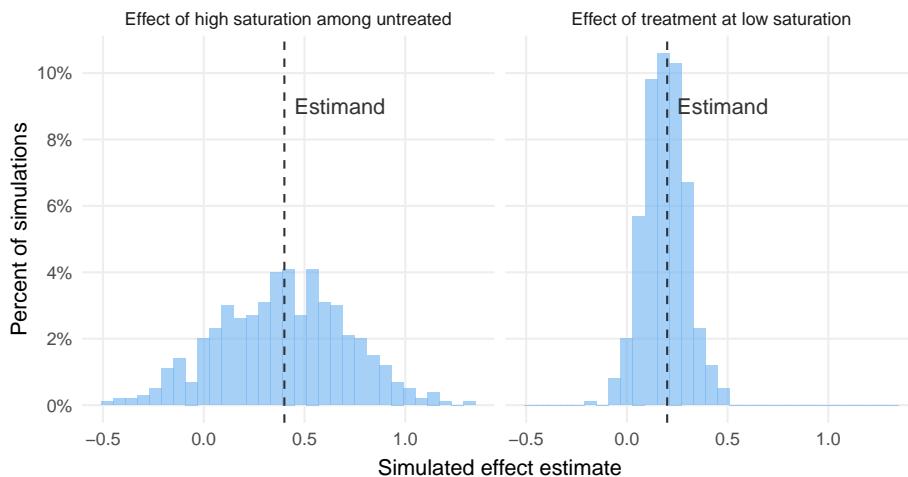


Figure 17.16: Sampling distribution of two estimators

Exercises

1. Diagnose the declared design with respect to statistical power. What is the statistical power for each inquiry?
2. Add an inquiry for the average effect of treatment, averaging over both high and low saturation. Add an associated estimator. What are the power, bias, and standard error diagnosands for this inquiry - estimator pair?
3. We chose saturations of 25% and 75%. Write a short paragraph describing under what circumstances it would be preferable to choose saturations of 10% and 90%.

17.11 Experiments over networks

When experimental subjects are embedded in a network, units' outcomes may depend on the treatment statuses of nearby units. In other words, treatments map spill over across the network. For example, in a geographic network, vote margin in one precinct may depend on outdoor advertisements in neighboring precincts. In a social network, information delivered to a treated subject might be shared with friends or followers. In a temporal network, treatments in the past might affect outcomes in the future.

This chapter describes the special challenges associated with experiments over networks. In the previous chapter, we discussed randomized saturation designs, which are appropriate when we can describe a hierarchy of units embedded in clusters, within which spillovers can occur but across which spillovers cannot occur. In other words, the randomized saturation design is appropriate when the network is composed of many disconnected network components (the clusters). But most networks are not disconnected; all or almost all units connected in a vast web. This chapter describes how we need to modify the model M , inquiry I , data strategy D , and answer strategy A to learn from experiments over networks.

In the model, our main challenge is to define how far apart (in social, geographic, or temporal space) units have to be in order for unit i 's potential outcomes not to depend on unit j . We might say units within 5km matter but units further away do not. We might say that units within two friendship links matter but more distal connections do not. We might allow the treatment statuses of three, two, or one periods ago to impact present outcomes differently from one another. For example, we might stipulate that each unit has only four potential outcomes that depend on whether a unit is directly treated or indirectly treated by virtue of being adjacent to a directly treated unit.

Table 17.5: Example treatment conditions for an experiment over a network

Condition	Potential outcomes
Pure control	$Y_i(\text{direct} = 0, \text{indirect} = 0)$
Direct only	$Y_i(\text{direct} = 1, \text{indirect} = 0)$
Indirect only	$Y_i(\text{direct} = 0, \text{indirect} = 1)$
Direct and indirect	$Y_i(\text{direct} = 1, \text{indirect} = 1)$

With potential outcomes defined, we can define inquiries. With four potential outcomes, there are six pairwise contrasts that we could contemplate. For example, the direct effect in the absence of indirect treatment is defined as $E[Y_i(\text{direct} = 1, \text{indirect} = 0) - Y_i(\text{direct} = 0, \text{indirect} = 0)]$ and the direct effect in the presence of indirect treatment is $E[Y_i(\text{direct} = 1, \text{indirect} = 1) - Y_i(\text{direct} = 0, \text{indirect} = 1)]$. We could similarly define indirect effects as $E[Y_i(\text{direct} = 0, \text{indirect} = 1) - Y_i(\text{direct} = 0, \text{indirect} = 0)]$ or $E[Y_i(\text{direct} = 1, \text{indirect} = 1) - Y_i(\text{direct} = 1, \text{indirect} = 0)]$. We may be interested in how direct and indirect treatments interact, which would require taking the difference between the two direct effect inquiries or taking the difference between the two indirect effect inquiries. Which inquiry is most appropriate will depend on the theoretical setting.

The data strategy for an experiment over networks still involves random assignment. Typically, however, experimenters are in direct control of the direct treatment application only, and the resulting indirect exposures result from the natural channels through which spillover occur. The mapping from a direct treatment vector to the assumed set of conditions is described by Aronow and Samii (2017) as an “exposure mapping.” The exposures mapping defines how the randomized treatment results in the exposures that reveal each potential outcome. The probabilities of assignment to each of the four conditions are importantly *not constant* across units, for the main reason that units with more neighbors are more likely to receive indirect treatment. Furthermore, exposures are dependent across units: if one unit is directly treated, then all adjacent units must be indirectly treated.

Under Principle 3.7: Seek M:I::D:A parallelism, we need to adjust our answer strategy to compensate for the differential probabilities generated by this complex data strategy. As usual, we need to weight units by the inverse of the probability of assignment to the condition that they in. In the networked setting we have to further account for dependence in treatment assignment probabilities. This dependence tends to increase sampling variability. For intuition, consider how clustering (an extreme form of across-unit dependence in treatment conditions) similarly tends to increase sampling variability. Aronow and Samii (2017) propose Hajek- and Horvitz-Thompson-style point and variance estimators that account for these complex joint probabilities of assignment, which are

themselves estimated by simulating the exposures that would result from many thousands of possible random assignments.

17.11.1 Example

Here we declare a experiment design to estimate the effects of lawn signs. The units are the lowest level at which we can observe vote margin, the voting precinct. In our model, we define four potential outcomes. Precincts can be both directly and indirectly treated, only directly treated, only indirectly treated, or neither. Indirect treatment occurs when a neighboring precinct is treated. This model could support many possible inquiries, but here we will focus on three: the direct effect of treatment when the precinct is not indirectly treated, the effect of indirect treatment when the precinct is not directly treated, and the total effect of direct and indirect treatment versus pure control. The data strategy will involve randomly assigning some units to direct treatment, which will in turn cause other units to be indirectly treated. We will need to learn via simulation the probabilities of assignments to conditions that this procedure produces. We'll make use of two answer strategies: the Horvitz-Thompson and Hajek estimators proposed by Aronow and Samii (2017), along with their associated variance estimators, as implemented in the `interference` package.

Some features of this design must be described outside the design-as-declared, for the main reason that some of the matrices required by the design (the geographic adjacency matrix, the permutation matrix, and the probability matrices) have non-tidy data structures that must be handled outside of `DeclareDesign`.

First, we load the Fairfax County, Virginia voting precincts shapefile, removing one singleton voting precinct, and we plot the precincts in Figure 17.17.

Next, we obtain the adjacency matrix.

```
adj_matrix <-
  fairfax %>%
  as("Spatial") %>%
  poly2nb(queen = TRUE) %>%
  nb2mat(style = "B", zero.policy = TRUE)
```

The last bit of preparation we need to do is to create a permutation matrix of possible random assignments, from which probabilities of assignment to each condition can be calculated:

```
ra_declaration <- declare_ra(N = 238, prob = 0.1)

permutatation_matrix <-
```

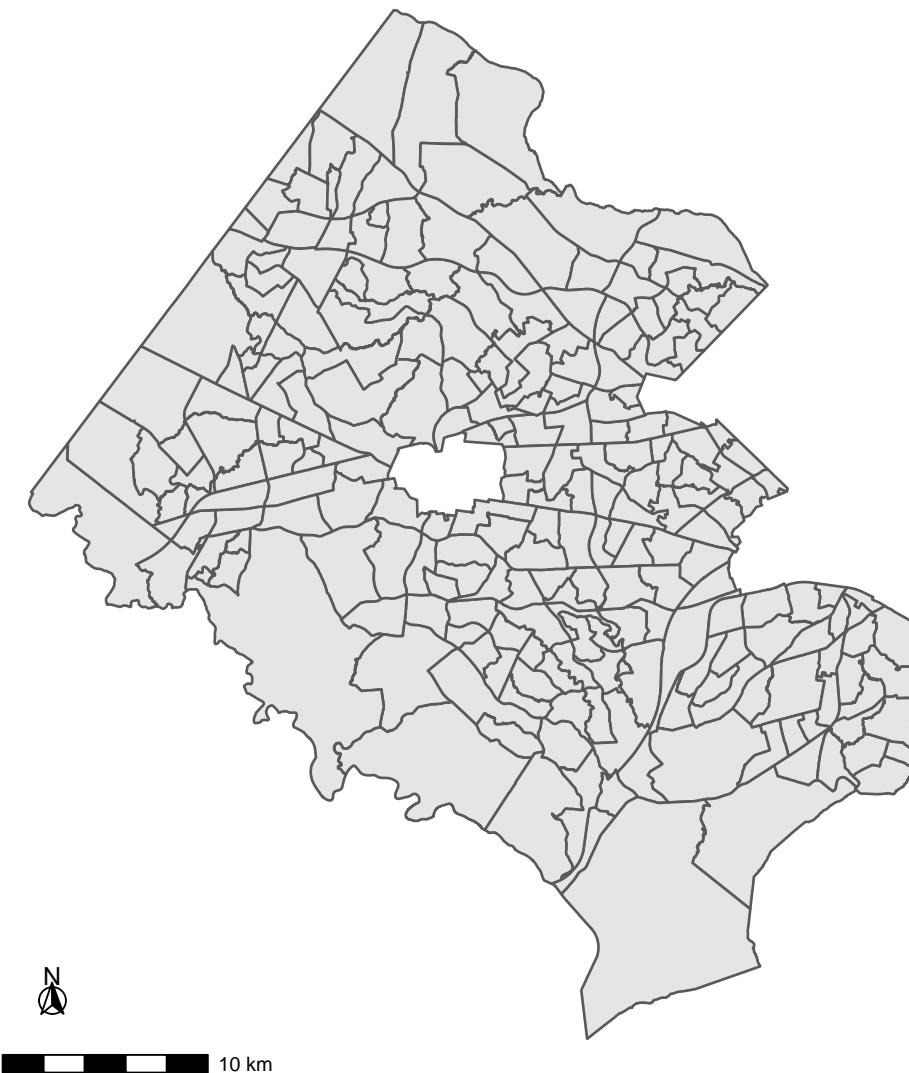


Figure 17.17: Voting Precincts in Fairfax County, Virginia

```
ra_declaration %>%
  obtain_permutation_matrix(maximum_permutations = 10000) %>%
  t()
```

Now we're ready to declare the full design. We've hidden the `get_exposure_AS` and `estimator_AS` helper functions to reduce the amount of code to read

through. The `declare_model` call builds in a dependence of potential outcomes on the length of each precinct's perimeter to reflect the idea that outcomes are correlated with geography in some way. The `declare_inquiry` call describes the three inquiries in terms of potential outcomes. The `declare_assignment` call first conducts a random assignment according to the procedure described by `declare_ra` above, then obtains the exposures that the assignment generates. Finally, all the relevant information is fed into the Aronow and Samii estimation functions via `estimator_AS`.

Declaration 17.15.

```
design <-
  declare_model(
    data = select(fairfax, -geometry),
    Y_0_0 = pnorm(scale(SHAPE_LEN), sd = 3),
    Y_1_0 = Y_0_0 + 0.02,
    Y_0_1 = Y_0_0 + 0.01,
    Y_1_1 = Y_0_0 + 0.03
  ) +
  declare_inquiry(
    total_ATE = mean(Y_1_1 - Y_0_0),
    direct_ATE = mean(Y_1_0 - Y_0_0),
    indirect_ATE = mean(Y_0_1 - Y_0_0)
  ) +
  declare_assignment(
    Z = conduct_ra(ra_declaration),
    exposure = get_exposure_AS(make_exposure_map_AS(adj_matrix, Z, hop = 1))
  ) +
  declare_measurement(
    Y = case_when(
      exposure == "dir_ind1" ~ Y_1_1,
      exposure == "isol_dir" ~ Y_1_0,
      exposure == "ind1" ~ Y_0_1,
      exposure == "no" ~ Y_0_0
    )
  ) +
  declare_estimator(handler = estimator_AS,
                    permutation_matrix = permutatation_matrix,
                    adj_matrix = adj_matrix)
```

The maps in Figure 17.18 show how this procedure generates differential probabilities of assignment to each exposure condition. Units that are in denser areas of the county are more likely to be in the Indirect Exposure Only and Direct

and Indirect Exposure conditions than those in less dense areas.

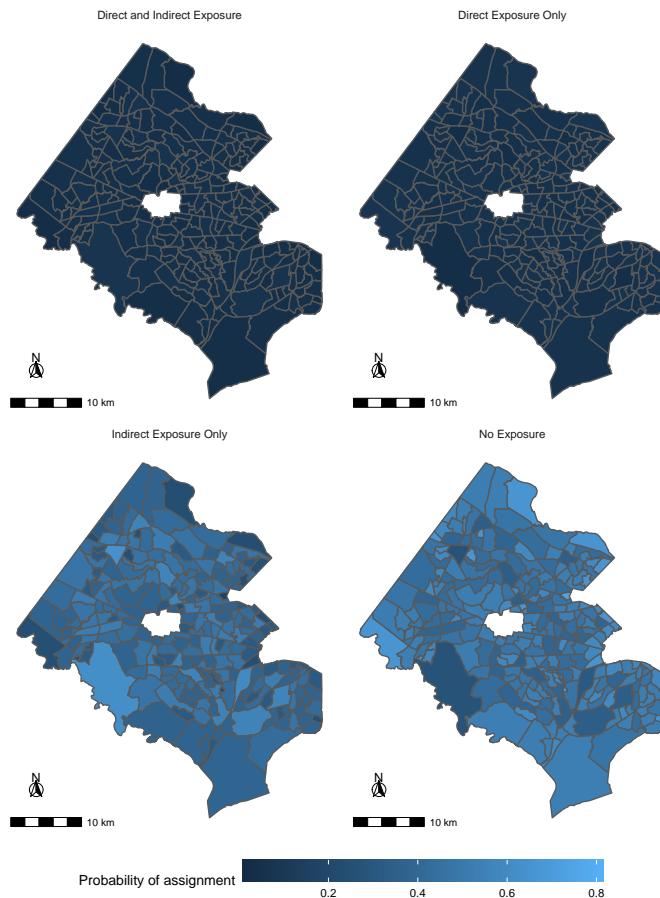


Figure 17.18: Probabilities of assignment to condition

Figure 17.19 compares the performance of the Hajek and Horvitz-Thompson estimators. Both are approximately unbiased for their targets, but the Horvitz-Thompson estimator is much higher variance, suggesting that in many design settings, researchers will want to opt for the Hajek estimator.

```
simulations <- simulate_design(design)
```

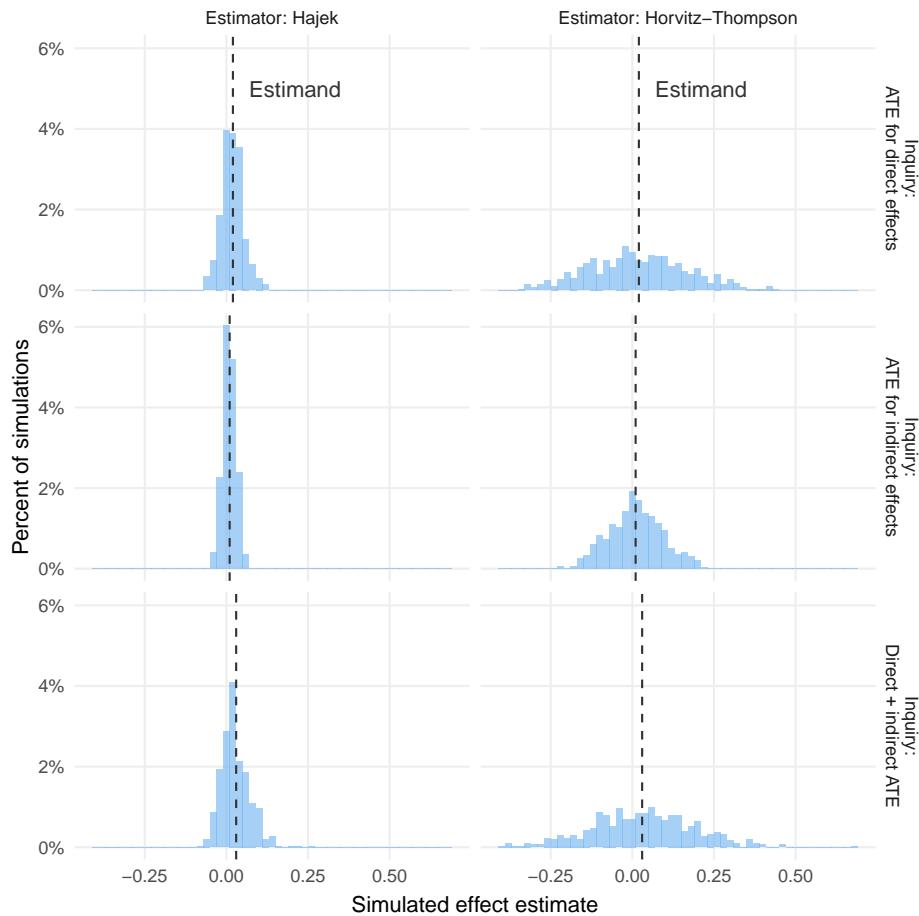


Figure 17.19: Sampling distribution of two estimators for three inquiries. The vertical lines refer to the true values of the inquiries.

Exercises

1. Increase the fraction directly treated to 0.3, and describe how the bias and standard deviation of the sampling distributions for all six estimators change.

Chapter 18

Complex designs

In the designs we presented thus far, the aim was generally to learn about the level of some variable or some particular causal effect. In most cases, a single set of observations was collected and an answer strategy was applied directly to this data to answer a causal or descriptive inquiry.

Very many published studies have this actual form – most draw in complex ways on a series of research designs, each targeted to a different inquiry, that when brought together answer the deeper theoretical question at the heart of the research. Research studies are complex in this way.

But studies can also be “complex” in multiple ways. For instance, although we have assumed researchers start with well defined inquiries, some studies focus on first figuring out what question to ask and then proceed to ask and answer it. The study engages in model building, then reports the results of a research design targeted at questions posed by the new model.

Some studies seek not to learn about levels and effects but search explicitly for a model of a phenomenon, asking for instance “what causes Y ?” or “what model, in some class, best accounts for observed data”? These studies have complex inquiries. Other studies have complex data and answer strategies, for instance, mixing qualitative and quantitative inference strategies or gathering together findings from multiple sub-studies in order to arrive at an overall conclusion.

18.1 Mixed methods

We declare a design in which a researcher combines qualitative and quantitative data to draw integrated inferences about causal effects. We imagine that a researcher has access to experimental data on X and Y from many cases but can also invest in gathering “process” data on a mediator M from a small number of cases. Overall inferences are made based on patterns in both sets of data.

Diagnosis helps researchers figure out the expected gains from process data and help determine which kinds of cases are most useful for maximizing expected learning.

In section 15.1 we described a qualitative strategy for making case level inferences from combining observations of within-case information with a background causal model. In section 17.1 we described standard approaches for making population level inferences from experimental data. While these approaches appear to use very different types of data, ask different questions, and employ different inferential strategies, it is possible to combine the two to form integrated inferences. This is one way of doing “mixed methods” analysis (Humphreys and Jacobs, 2017).

We imagine a setting in which a binary treatment X is randomized. Outcomes for Y , also binary, are recorded. We imagine that researchers can also gather data on a presumed mediator, M , for the effect of X on Y . Our model involves a few assumptions. First we imagine that any effect of X on Y must go through M . Second we imagine that X does not negatively affect M and M does not negatively affect Y . For instance we might imagine that providing a drug might cure a disease but only if the drug is actually consumed. We will assume that X is randomized but we do not assume “sequential ignorability”, and in particular do not assume that M is as if random given X . Thus the kind of assumptions we impose here are similar to those invoked when doing estimation using instrumental variables (see section 15.4).

We set up a causal model using the `CausalQueries` package on binary nodes and impose an assumption that X does not negatively affect M and M does not negatively affect Y . This model (summarized by the DAG $X \rightarrow M \rightarrow Y$) can be parameterized by a Bernoulli distribution giving the probability that $X = 1$, a Categorical distribution over the three ways that M can respond to X (M responds positively to X , and $M = 1$ or $M = 0$ regardless), and a Categorical distribution over the three ways that Y can respond to M (Y responds positively to X , and $Y = 1$ or $Y = 0$ regardless) *given* each of the three ways that M responds to X : this is equivalent to a joint distribution over types for M and types for Y , allowing, for instance, for the possibility that M affecting Y is more likely for cases for which X affects M .

We assume that we know that X is assigned with 50% probability, but otherwise provide a “flat” Dirichlet prior probability distribution over these parameter sets by stipulating $\alpha = 1$ for each parameter set. This prior is used twice in the design: first the prior characterizes the set M and a draw m is a draw from the induced prior. Second the prior is also used in the answer step as part of Bayesian updating. We emphasize however that the same distribution does *not* have to be used for these two purposes: we can assess the performance of a design using a distribution on M that is quite different to that employed by the researcher when generating answers.

Table 18.1: Parameters correspond to causal types for M and Y , with V_{ab} meaning that V takes value a when V 's parent is 0 and b when V 's parent is 1. There are 12 parameters for M and Y in all and 8 degrees of freedom.

Parameter set	Parameter	Prior (α)
M	λ^M_{00}	1
M	λ^M_{01}	1
M	λ^M_{11}	1
$Y M_{00}$	$\lambda^Y_{00 00}$	1
$Y M_{00}$	$\lambda^Y_{01 00}$	1
$Y M_{00}$	$\lambda^Y_{11 00}$	1
$Y M_{01}$	$\lambda^Y_{00 01}$	1
$Y M_{01}$	$\lambda^Y_{01 01}$	1
$Y M_{01}$	$\lambda^Y_{11 01}$	1
$Y M_{11}$	$\lambda^Y_{00 11}$	1
$Y M_{11}$	$\lambda^Y_{01 11}$	1
$Y M_{11}$	$\lambda^Y_{11 11}$	1

```
library(CausalQueries)
causal_model <- make_model("X -> M -> Y; M <-> Y") %>%
  set_priors(node = "X", alpha = c(100, 100)) %>%
  set_restrictions(labels = list(M = "10", Y = "10"))
```

Table 18.1 shows parameters corresponding to the ways that M can respond to X and Y to M given M 's type. Each “run” of the model involves one draw of the parameters from each parameter set.

I: We focus on two inquiries: the average treatment effect and an “attribution” estimand: the probability that $X = 1$ caused $Y = 1$ in a case with $X = 1$ and $Y = 1$. The estimands are calculated by querying the causal model given the parameter vector that has been drawn.

D: The quantitative part of the data strategy involves selecting a set of units, randomly assigning X and measuring the outcome Y . The qualitative part involves selecting a set of cases for which we get to observe M . Recall in section 8.1 we described case selection strategies to learn about the effects of X and Y in which researchers first assessed global relations between X and Y and then selected cases with particularly known values of X and Y for further investigation. That discussion supposed that the quantitative data structure was fixed. In contrast, the design here lets us simultaneously examine the gains from large n decisions and different case selection strategies. In practice we will examine strategies that look into 20 case studies and compare a strategy that selects from

the regression line ($X = Y = 0$ and $X = Y = 1$ cases) against a strategy that selects positive cases only ($X = Y = 1$ cases). One wrinkle in the design is that we specify the within case analysis strategy before we know the X, Y pattern. This gives rise to the problem that we might want to examine 20 cases on the regression line (for instance) but only have 5 to choose from. To address this in a way consistent with Principle 3.6 we specify that we will gather data on as many cases as possible given the observed data.

```
# Quantitative data
data_handler = function(data, n = n)
  CausalQueries::make_data(causal_model,
    parameters = data$parameters,
    n = n)

# Case selection data targets within X, Y data combinations
strategy <- c("00" = 10, "01" = 0, "10" = 0, "11" = 10)
strategy_names <- names(strategy)

# strat_n flexible to take account of the possibility of no data in some strata
strata_n <- function(strategy, strata)
  sapply(1:4, function(i) min(strategy[i], sum(strata == strategy_names[i])))[strategy]
```

A: The answer strategy involves updating the causal model given observed data. Bayesian updating is done using a `stan` function from the `CausalQueries` package using a multinomial likelihood implied by the causal model. The updated model is then queried using the same function as used to generate the estimand, illustrating the application of principle 3.7 (seek parallelism) in Bayesian analysis.

```
estimation_handler = function(data)
  causal_model %>% update_model(data = data) %>%
  query_model(query = "Y[X=1] - Y[X=0]",
    using = "posteriors",
    given = c(TRUE, "X==1 & Y==1")) %>%
  rename(estimate = mean) %>%
  select(estimate, sd) %>%
  mutate(Inquiry = c("ATE", "POC"))
```

The full design is then declared as follows.

```

n <- 50

design <-

declare_model(data.frame(parameters = CausalQueries::get_parameters(causal_model, param_type = "mean"),
                        parameters = parameters, using = "parameters")$mean) +
  declare_inquiry(ATE = CausalQueries::query_model(causal_model, "Y[X=1] - Y[X=0]", given = "X==1",
                                                    parameters = parameters, using = "parameters")$mean) +
  declare_inquiry(POC = CausalQueries::query_model(causal_model, "Y[X=1] - Y[X=0]", given = "X==1",
                                                    parameters = parameters, using = "parameters")$mean) +
  declare_measurement(handler = data_handler, n = n) +
  declare_measurement(
    strata = paste0(X,Y),
    M_observed = strata_rs(strata = paste0(X,Y), strata_n = strata_n(strategy, strata)),
    M = ifelse(M_observed==1, M, NA)) +
  declare_estimator(handler = label_estimator(estimation_handler), inquiry = c("ATE", "POC"))

```

We extract as our key diagnosand the expected posterior variance over runs. This quantity tells us how uncertain you expect to be after implementing your study. If this expected posterior variance were 0 it means that you would expect to be certain of whatever answer you get no matter what data you get. If the expected posterior variance is the same as the prior variance then you do not expect to learn anything, regardless of what you see. The expected posterior variance also has a nice interpretation as a measure of the expected squared error of the posterior mean, (with expectations are taken over the prior).

```

mixed_diagnosands <-
  declare_diagnosands(mean_estimate = mean(estimate),
                      sd_estimate = sd(estimate),
                      bias = mean(estimate - estimand),
                      posterior_variance = mean(sd^2))

```

```

diagnosis <- design %>%
  redesign(N = c(50, 100, 150),
           strategy = list(c(0,0,0,0), c(10,0,0,10), c(0,0,0,20))) %>%

```

```
diagnose_design(sims = 2, diagnosands = mixed_diagnosands)
```

Diagnoses for a range of quantitative and qualitative data strategies are shown in Figure 18.1 for both estimands. The figure shows first that there is learning from case studies for the average treatment effect inquiry. Learning is greatest when cases are drawn on the regression line, but there is also some learning even when there is no variation in X and Y in the cases. For both inquiries however we see that we reduce variance by about the same amount by collecting data on M within 20 cases as we do from gathering data on X and Y for an additional 20 cases. With similar marginal gains, whether marginal resources should be allocated to going “wide” or going “deep” likely depends on the relative costs of these strategies.

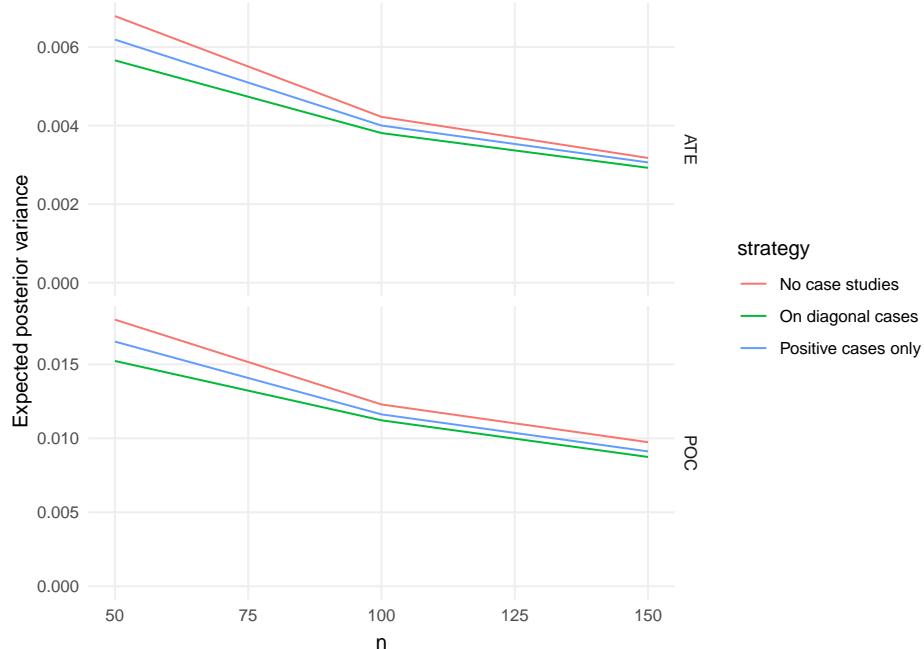


Figure 18.1: Expected posterior variance

As with the Bayesian analysis in Section 15.1, the design declared here uses the same model and priors for the reference model set (M) and for the answer strategy. This is not required however. Instead you can assess the performance of the answer strategy with respect to reference model sets that differ in at least three ways from that used in the answer strategy: with respect to the assumed causal structure, with respect to monotonicity assumptions, and with respect to the prior distribution over parameters.

18.2 Discovery using causal forests

We declare a design in which a researcher examines a large set of continuous covariates to assess (i) which covariate best explains heterogeneity in the effects of a treatment and (ii) the effect for the subjects for whom the treatment effect is weakest or strongest. The design declaration clarifies the inquiries in a discovery analysis and can be used to guide design decisions regarding how best to split data into training and testing sets.

In most designs we have discussed, researchers have a clear idea what they are looking for when they begin the research. How big is some population? What is the size of some effect? But some research involves questions that are much more open in nature. We focus here on discovery research that has two types of open question. The first inquiry poses an open question of the form “what matters?” (rather than than the more common closed question of the form “what is the effect of this on that?”). The second inquiry poses a question about a not yet identified group—who are the people for whom effects are especially strong or weak and what is the effect for those people?

We imagine a setting in which a researcher has access to a large group of covariates X and has conducted an experiment to assess the effects of Z on Y . The researcher is interested in the *heterogeneity* of effects of Z as a function of variables in X and in particular asks:

- Which covariate best “explains” variation in the effect of Z ?
- What combinations of covariates describe individuals for whom effects are particularly weak?

M: We first need to describe a background model, or set of models. For this we imagine a complex function linking Z to Y in which the effect of Z depends non linearly on some but not all variables in X . As always, this model can and should be altered to help understand whether diagnosis depends on the particular type of structure assumed.

```
covariates <- paste0("X.", 1:10)

f_Y <- function(z, X.1, X.2, X.3, X.4, u) {
  z * X.1 + z * X.2 ^ 2 + z * exp(X.3) + z * X.3 * X.4 + u
}
```

I: For the first inquiry we need to be specific about what we mean by “best explains.” We will imagine asking which X produces the lowest conditional variance $E_x(Y(1)|Y(0)|X = x)$. Specifically we will partition each covariate into deciles and take the average variance in treatment effect across each decile. We will call this the `best_predictor` inquiry and calculate it using a fixed effects model applied to potential outcomes data.

The following handler function calculates this estimand.

```
best_predictor <- function(data) {
  data.frame(
    inquiry = "best_predictor",
    estimand = lapply(covariates, function(j) {
      lm_robust(tau ~ cut(data[[j]], 20), data = data)$r.squared
    }) %>%
      unlist %>% which.max
  )
}
```

There is a simple and a more complex understanding of the second inquiry. The simple understanding is that we are interested in the average effect among the units whose effects are in the bottom 20% (say) of all effects. We will call this the `worst_effects` inquiry. This is a natural notion of the worst affected. But it is a very difficult quantity to estimate.

Another approach is to examine realized data and do our best to identify a set of units (say of size 20%) that we think will have weak effects, and with this set identified we will return to the model and ask what is the average effect for this set. We will call this the `weak_effects` inquiry, to acknowledge that the effects for this group may not be the worst effects.

We assume data strategy is the same as a two simple two arm trial (Section 17.1).

The answer strategy employs a “causal forests” algorithm. The approach randomly partitions data into a training and testing group. Within the training group it repeatedly generates “trees” by repartitioning the covariates (generating “branches”) to identify subgroups (“leafs”) with similar treatment effects. At each step, partitioning is implemented to yield estimated minimum variance in treatment effects and unit level predictions of treatment effects are generated by combining estimates of effects for units over different trees. See Wager and Athey (2018) for full details of the approach. Our estimate of the `best_predictor` is based on the variable that is most frequently partitioned to reduce variance. We have two estimates of the weak effects, one based on the training set and one based on the test set. We will assess the performance of these against both the `weak_effects` inquiry and the `worse_effects` inquiry.

We note that the “most common variable” indicator is not *designed* to capture the variable that induces the greatest reduction in heterogeneity. Indeed it is not very clear which inquiry corresponds to this measure, and, perhaps for this reason Wager and Athey (2018) do not emphasize this quantity. Including it here however allows us to illustrate the ability of diagnosis to assess the performance of an estimator for a task for which it was not designed.

The estimates are generated by a causal forest handler which makes use of the `grf` package. In addition to estimating these quantities we note that the handler calculates, `weak_effects`, our post estimation *estimand*.

```
causal_forest_handler <- function(data, ...) {

  X <- as.matrix(data %>% select(all_of(covariates)))
  train <- data$train

  cf <- causal_forest(X = X[train, ], Y = data$Y[train], W = data$Z[train], ...)

  # Prep and return data
  data$pred <- NA
  data$pred[train] <- predict(cf, estimate.variance=FALSE)$predictions
  data$pred[!train] <- predict(cf, newdata=X[!train,], estimate.variance=FALSE)$predictions

  data %>%
    mutate(var_imp = variable_importance(cf) %>% which.max,
          low_test = (!train & (pred < quantile(pred[!train], .2))),
          high_test = (!train & (pred > quantile(pred[!train], .8))),
          low_all = pred < quantile(pred, .2))
}

take_1 <-
  function(data) {
    data %>%
      slice(1) %>%
      mutate(estimate = var_imp) %>%
      select(estimate)
}
```

Declaration 18.1.

```
N <- 1000

design <-
  declare_model(
    N = N,
    X = matrix(rnorm(10*N), N),
    u = rnorm(N),
    Z = sample(0:1, N, replace = TRUE)) +
  declare_measurement(handler = fabricate,
```

```

Y_1 = f_Y(1, X.1, X.2, X.3, X.4, u),
Y_0 = f_Y(0, X.1, X.2, X.3, X.4, u),
tau = Y_1 - Y_0,
Y = f_Y(Z, X.1, X.2, X.3, X.4, u),
train = simple_rs(N==1) +
declare_inquiry(handler = best_predictor, label = "custom") +
declare_step(handler = causal_forest_handler) +
declare_inquiry(
  worst_effects = mean(tau[tau <= quantile(tau, .2)]),
  weak_effects = mean(tau[low_test]),
  weak_all = mean(tau[low_all]),
  strong_effects = mean(tau[high_test])) +
declare_estimator(Y ~ Z, model = lm_robust, subset = low_test,
                  inquiry = c("weak_effects", "worst_effects"), label = "lm_weak") +
declare_estimator(Y ~ Z, model = lm_robust, subset = low_all,
                  inquiry = "weak_all", label = "lm_weak_all") +
declare_estimator(Y ~ Z, model = lm_robust, subset = high_test,
                  inquiry = "strong_effects", label = "lm_strong") +
declare_estimator(handler = label_estimator(take_1),
                  inquiry = "best_predictor", label = "cf")

```

Declaration 18.1 is relatively straightforward though we point out that the causal forests estimation is introduced as a general step and not specifically as an estimation step. The reason for this is that the procedure produces predictions for each observation and so the output is naturally added to the primary data frame. Estimates are then defined using this output.

Before turning to diagnosis we can check the performance of the causal forest model by comparing the predicted effects for each unit generated by the model, with the actual unit level treatment effects generated by *M*. Figure 18.2 illustrates.

We see that we do reasonably well but also that the *range* of the predictions is narrower than the range of the true effects, which will mean that the average effects in the groups with the best or worst predicted effects will generally not be the same as the effects for the groups with the best and worst actual effects.

For the diagnosis we need to take account of the fact that the answers to one of the inquiries (“Which X accounts for the most variation in effects?”) should be treated as categorical. For this, rather than replying on the average estimate and average estimand we report the modal estimate and estimand; and rather than relying on bias we calculate the probability that we get the correct answer.

We see that we do very well in identifying the most powerful predictor of hetero-

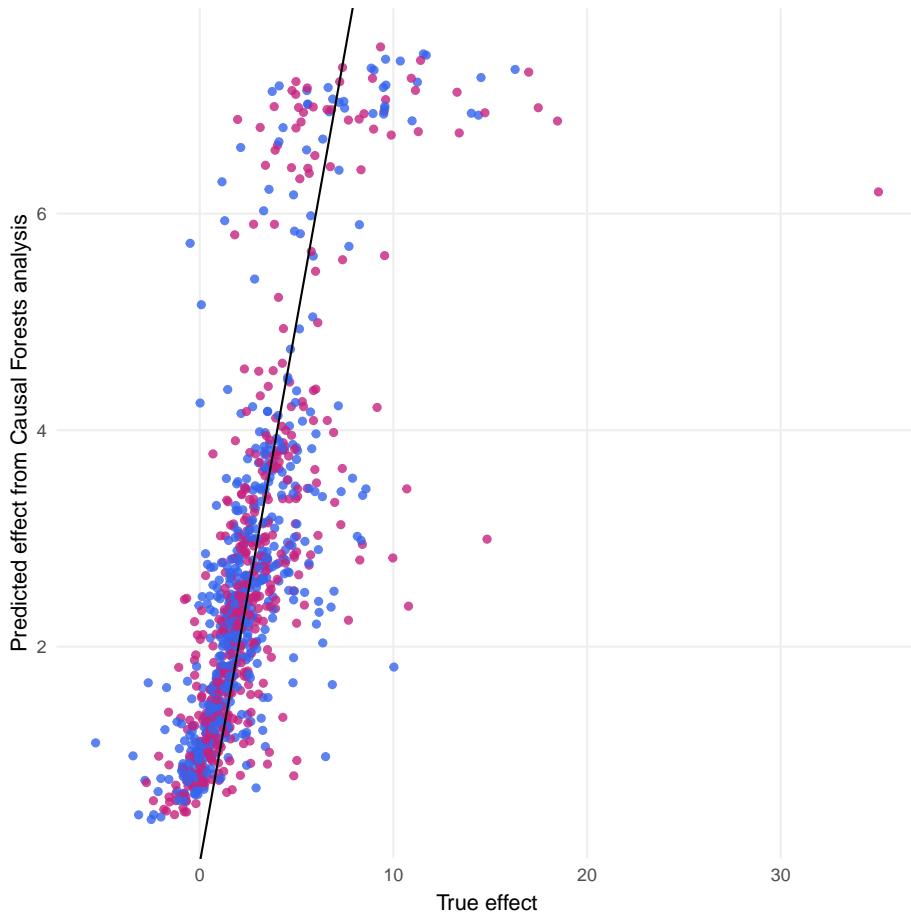


Figure 18.2: One draw from the causal forests design

genuity, correct nearly all of the time. (We are never “correct” for the continuous estimands, but we would never expect to be.) Our in-sample (or full sample) estimate of effects for the weak group is biased. Our out-of-sample subgroup estimate estimates the average effects for the “weak” and “strong” groups without bias however. Finally we see that the out-of-sample estimate for the weak group does not estimate the effects for the “worst” group. This highlights the fact that the procedure can get unbiased estimates for *a* group that does poorly but not *the* group that does most poorly.

Overall the approach fares very well and through diagnosis we get clarity over which quantities are well estimated.

Modifications of this design can let you assess how sensitive performance is to the type of model stipulated and what are the best divisions between treatment

Table 18.2: Casual forests design diagnosis

Inquiry	Estimator	Correct	Bias	Mean Estimate	Modal Estimate	Mean Estimand
best_predictor	cf	0.99 (0.00)	-0.01 (0.00)	2.99 (0.00)	3.00 (0.00)	3.00 (0.00)
strong_effects	lm_strong	0.00 (0.00)	0.00 (0.02)	6.01 (0.03)	5.90 (0.12)	6.01 (0.02)
weak_all	lm_weak_all	0.00 (0.00)	0.03 (0.01)	0.07 (0.01)	0.10 (0.07)	0.04 (0.01)
weak_effects	lm_weak	0.00 (0.00)	0.01 (0.01)	0.04 (0.01)	-0.00 (0.04)	0.03 (0.01)
worst_effects	lm_weak	0.00 (0.00)	0.41 (0.01)	0.04 (0.01)	-0.00 (0.04)	-0.37 (0.00)

and training sets for any stipulated model.

18.3 Structural estimation

We declare a design in which a researcher wants to use experimental data to estimate the parameters of a game theoretic model. The promise of the approach is that, if the underlying theory is correct, parameter estimation allows the researcher to make interesting external inferences to different types of treatment effect in other settings. The risk is that if the model is wrong these inferences may be invalid. Diagnosis helps assess the gains and risks of the approach.

Structural estimation is used in situations where researchers have a general model in mind for how processes work and their goal is to fit the parameters of the model. With the fitted model they might then estimate levels of unobserved variables, treatment effects, or other quantities. They might even extrapolate to estimate counterfactual quantities, such as the effects of interventions that have not been implemented (Reiss and Wolak, 2007).

We illustrate using an example of a model of bargaining between pairs of players, drawing on an example use in Wilke and Humphreys (2020).

Wilke and Humphreys (2020) imagine a bargaining game with payments from customer i to a taxi driver given by:

$$i = i(z_i y + (1 - z_i)(1 - y)) + (1 - i)$$

Here $y = \sum_{j=2}^n (1-j)^{j-1}$ is the equilibrium offer made by the first mover as predicted by the Rubinstein (1982): alternating offers bargaining model with n possible bargaining rounds given discount factor γ . The customer's payoff depends on

whether she goes first ($z_i = 1$) or second ($z_i = 0$). Non-rational customers ($i = 0$) do not engage in bargaining but successfully invoke a norm, insisting on giving the driver some share of their endowment, irrespective of whether they go first or second. We let q denote the probability that $= 1$.

To allow for a disturbance we assume that *measured* payments are a draw from a Beta distribution with expectation given by the expected payment and variance parameterized by .

We imagine that Z is randomly assigned and we have access to data on payments, . We will also assume we know price . Our goal however is not simply to measure the effect of Z on but to estimate the model parameters, $, q, \alpha, \beta$, which themselves can be used to estimate this effect and other counterfactual quantities (if we assume the model is true).

M. In this declaration we will assume that the data is indeed produced by a process similar to that assumed at the estimation stage.

I. Our inquiries will include parameters k, d , and q corresponding to , and q in the model (we will treat as known). In addition we will be interested in the effect of Z on payments in a two period game and in the game of indefinite duration.

D. First mover position, Z , is randomly assigned. Payments are measured with some error (in this model however we cannot distinguish between measurement error and decision making error). We imagine that we have access to data from games with $n = 2$ and games with $n =$ in order to compare the performance of estimators in both conditions.

A. The analysis is implemented using maximum likelihood to identify which parameter values are most consistent with the data (which collection of parameter values produce the observed data with greatest likelihood). In this declaration the model employed in *A* is the same as that employed in *M*. We will report analysis results from both differences in means and structural estimation generated both from using the $n = 2$ data (`DIM_two`, `Struc_two`) and from using $n =$ data (`DIM_inf`, `Struc_inf`)

The most complex part of the design is the specification of the estimator, shown next:

```
structural_estimator <- function(data, pi, y, chi = 3/4){

  # Define negative log likelihood as a function of k, d and q
  LL <- function(k, d, q) {
    m <- with(data, y(Z, d))
    R <- q * dbeta(data[pi][[1]], k * chi, k * (1 - chi)) +
      (1 - q) * dbeta(data[pi][[1]], k * m, k * (1 - m))
  }
}
```

```

    - sum(log(R))
}

# Estimation
M <- mle2(
  LL,
  method = "L-BFGS-B",
  start = list(k = 2, d = 0.50, q = 0.50),
  lower = c(k = 1, d = 0.01, q = 0.01),
  upper = c(k = 1000, d = 0.99, q = 0.99)
)

# Format output from estimation
out <- data.frame(coef(summary(M)), outcome = pi)

names(out) <- c("estimate", "std.error", "statistic", "p.value", "outcome")

# Use estimates of q and delta to predict average treatment effects (ATEs)
# Predicted ATE for n=2
out[4, 1] <- (1 - out["q", "estimate"]) * (2 * out["d", "estimate"] - 1)

# Predicted ATE for n=infinity
out[5, 1] <- (1 - out["q", "estimate"]) * (2 * out["d", "estimate"] /
                                             (1 + out["d", "estimate"]) - 1)

out
}

```

The design makes use of this estimator to estimate parameter values as well as treatment effects. It is accompanied by a simpler difference-in-means estimator of treatment effects.

An attraction of structural estimation is that, with a fitted model, one can generate estimates of effects of treatments that have not been implemented. In this case, the same parameters that describe the equilibrium outcomes in a two round game are sufficient to describe outcomes in the infinitely repeated game. So if you understand the effects of a treatment in one case you understand it in the other. At least if the model is correct. In Declaration 18.2, we generate such estimates for the effects of unimplemented treatments.

Declaration 18.2.

```

d = 0.8      # True delta (unknown)
k = 6        # Parameter to governance variance (unknown)
q = 0.5      # Share of behavioral types in the population (unknown)
chi = 0.75   # Price paid by norm following ("behavioral" customers) (known)

design <-

declare_model(


# Define the population: indicator for behavioral type (norm = 1)
N = 500, norm = rbinom(N, 1, q),


# Define mean potential outcomes for n = 2
potential_outcomes(
  pi_two ~ norm*chi + (1-norm)*(Z*d + (1-Z)*(1-d))
),


# Define mean potential outcomes for n = infinity
potential_outcomes(
  pi_inf ~ norm*chi + (1-norm)*(Z*d/(1+d) + (1-Z)*(1-d/(1+d)))
)
) +


declare_inquiry(ATE_two = mean(pi_two_Z_1 - pi_two_Z_0), # ATE n = 2
                ATE_inf = mean(pi_inf_Z_1 - pi_inf_Z_0), # ATE n = infinity
                k = k,                                # kappa
                d = d,                                # delta
                q = q) +                                # q


declare_assignment(Z = complete_ra(N)) +


declare_measurement(


pi_two = reveal_outcomes(pi_two ~ Z),
pi_inf = reveal_outcomes(pi_inf ~ Z),


# Get draws from beta distribution given means for n = 2 and n = infinity
pi_two_obs = rbeta(N, pi_two*k, (1-pi_two)*k),
pi_inf_obs = rbeta(N, pi_inf*k, (1-pi_inf)*k)
) +


# Difference-in-means for n = 2
)

```

```

declare_estimator(pi_two_obs ~ Z, inquiry = "ATE_two", label = "DIM_two") +
  # Difference-in-means for n = infinity
declare_estimator(pi_inf_obs ~ Z, inquiry = "ATE_inf", label = "DIM_inf") +
  # MLE for n = 2
declare_estimator(handler = tidy_estimator(structural_estimator),
  pi = "pi_two_obs",
  y = function(Z, d) Z * d + (1 - Z) * (1 - d),
  inquiry = c("k", "d", "q", "ATE_two", "ATE_inf"),
  label = "Struc_two") +
  # MLE for n = infinity
declare_estimator(handler = tidy_estimator(structural_estimator),
  pi = "pi_inf_obs",
  y = function(Z, d) Z*d/(1+d) + (1-Z)*(1-d/(1+d)),
  inquiry = c("k", "d", "q", "ATE_two", "ATE_inf"),
  label = "Struc_inf")

```

Now for the diagnosis.

We see here that we do a good job in recovering parameter values and we also recover treatment effects. When using two period data the estimate for the ATE_two is as good when estimated using the structural and design based approaches. In the case with data from $n =$ games however the estimate from the structural model is less precise, though unbiased. In contrast we have no estimate using design based methods for this inquiry. Finally, , we see, is better estimated using data from the 2 period games; the estimate for is biased though the bias is small.

The basic structure used here can be used for a wide range of structural model: write down your theory and specify the implied likelihood function—which reports the probability of observing different types of data given model parameters. This might require adding noise to your model so that all data that is seen in practice *can* be seen in theory. Then identify the parameters that have the greatest likelihood of producing the data that you see. The same fundamental approach can be used for estimation via Bayesian methods or methods of moments. As suggested here, the payoffs from this approach can be great. The risks are large too however. Insofar as inferences are model dependent, model misspecification can lead to faulty conclusions. When doing structural estimation, apply Principle 3.9 liberally.

18.4 Meta-analysis

We declare a design for meta-analytic study in which a researcher seeks to combine findings from a collection of existing studies to form an overall conclusion. Declaration helps clarify the estimand in such studies. Diagnosis highlights risks associated with a common estimator used in meta-analysis, the fixed effect estimator.

In a meta-analysis, inquiries are summaries of the findings of past research studies. In most designs in this book, an observation in the data we analyze represent people, places, businesses, or governments. In a meta-analysis, an observation represents one estimate from a past empirical study.

Meta-analyses can tell us about the empirical consensus on a particular inquiry, often represented as the average value of an inquiry across studies. The mean value of the inquiry might not be especially interesting, however, if the true effect size varies across studies. True effects might vary for two reasons: true effect heterogeneity across studies of different units or contexts with the same research design, and differences in the treatments, outcomes, and bias induced by variations in research design. In fact, it is rare for at least small differences in research design to creep in, making the case of effect homogeneity uncommon. As a result, a second common inquiry in meta-analysis is how much true effects vary across studies, often represented by the variance in effect sizes.

Not every estimate from past literature is created equal. Some come from large studies with credible research designs; other small ones that introduce bias. We can use two features of past studies to guide how to treat their estimates. The research design — its *MIDA* — can tell us about how much bias may be induced in the design. We may want to exclude studies entirely with high bias. The estimated standard error reported in the past study also provides a clue to how informative the study will be. An estimate that comes from an unbiased design with a very low standard error — with a lot of precision — is more informative than one with a high standard error. Most meta-analysis techniques directly incorporate the estimated standard error by weighting by weights that are in proportion to the amount of precision (inverse of the estimated standard error) from the study.

We can also learn what we *don't* know from a meta-analysis. Analysis may reveal that the confidence interval on our estimates of the average value of the inquiry are very wide. Or we may find that there are few or no studies that can deliver unbiased results. The model describes where our sample of studies comes from and the statistics we collect from them. Data collection for a meta-analysis involves searching for relevant studies, filtering for eligibility on topical and research design grounds, extracting estimates and uncertainty measures, and in some cases postprocessing to standard estimates across studies. We declare a design with 100 studies with a mean true effect size () of 0.2 and estimated standard errors between 0.025 and 0.6.

We consider two common true models of the studies, which are commonly known (confusingly!) as random effects and fixed effect. In the fixed effect model—perhaps better called the “common effects” model so as to avoid confusion with the fixed effect model—we start with the true effect and estimated effects are drawn from a normal distribution centered on with standard deviation set to the estimated standard error extracted from the study. For each of our 100 studies, the true effect in this setup is assumed to be and we obtain noisy estimates of that true effect from studies with higher and lower precision depending on features of those individual studies such as sample size. We summarize the precision of past studies using the estimated standard error. In the random effects model, we relax the assumption that there is a single true effect for all studies, and that instead effects may vary due to contextual or design differences in the study. It may be that there are heterogeneous effects across studies due to different political institutions or economic conditions, or that the studies used slightly different measurement tools and so target slightly different inquiries. Variation in true effects across studies is represented in the model by drawing effect estimates from that same normal distribution, but centered on a true study effect which itself is a draw from a normal distribution centered on . Thus, there is still an average of true effects across studies of , but the true effects vary across contexts.

The two most common inquiries for a meta-analysis are the mean true effect across studies, often labeled , and the variance of true effects, τ^2 . When true effects do not vary across studies, then $\tau^2 = 0$ and all true studies effects are equal to . However, this is not always the case. Thus, whether the mean is a good single summary of the past evidence depends on the value of τ^2 , suggesting we should always estimate it and assess both statistics. A common goal in meta-analysis is to try to assess what the effect would be for a new study being contemplated or for a context in which a policy studied in past literature is being implemented. In both cases, we want to predict from past evidence what the effect for this current setting will be. To do so, we need to know how much true effects vary and what their average is. We also may want to more formally predict using characteristics that predict effect heterogeneity. A simple example of this prediction would be to estimate and τ^2 , observe the mean is around 0.2 but there is substantial variance of 0.2 and so find a past study with similar characteristics to the new study in terms of political institutions, economic conditions, and measurement tools.

We estimate the average true effect and the variance in true effects τ^2 using a frequentist random effects statistical model. Importantly, the random effects statistical model does not *impose* variance in true effects but rather estimates τ^2 and allows the effects to vary if it is estimated to be above zero. When the variance is estimated to be zero, then the random effects statistical model reduces to the fixed effect statistical model in which study estimates are drawn from a common distribution across sites.

Declaration 18.3.

```

design <-
  declare_model(
    N = 100,
    site = 1:N,
    mu = 0.2,
    tau = case_when(model == "random-effects" ~ 1,
                    model == "fixed-effects" ~ 0),
    std.error = pmax(0.1, abs(rnorm(N, mean = 0.8, sd = 0.5))),
    eta = rnorm(N),
    theta = mu + tau * eta, # note when tau = 0, theta = mu
    estimate = rnorm(N, mean = theta, sd = std.error)
  ) +
  declare_inquiry(mu = first(mu), tau_sq = first(tau^2)) +
  declare_estimator(
    yi = estimate, sei = std.error, method = "REML",
    model = rma_estimator, model_summary = rma_mu_tau,
    term = c("mu", "tau_sq"), inquiry = c("mu", "tau_sq"),
    label = "random-effects")

```

```

designs <- redesign(design, model = c("random-effects", "fixed-effects"))

simulations <- simulate_design(designs, sims = sims)

```

To illustrate the properties of the fixed effect estimator in some settings, we add it as a second estimator:

```

declare_estimator(
  yi = estimate, sei = std.error, method = "FE",
  model = rma_estimator, model_summary = rma_mu_tau,
  term = c("mu", "tau_sq"), inquiry = c("mu", "tau_sq"),
  label = "fixed-effects")

```

We explore the bias, efficiency (root mean-squared error), and coverage of the two estimators under each model both for the mean effects inquiry and for the variance of effects inquiry². The random effects model, across the possible models of how effects are realized, performs best. Whether the model is fixed effect or random effects it estimates both parameters without bias. Coverage is approximately nominal, but the standard errors are slightly biased for² when

Table 18.4: Bias, RMSE, and coverage from the random effects and fixed effect estimators under the models assumed by the estimators.

estimator	inquiry	model	bias	rmse	coverage
fixed-effects	mu	fixed-effects	0.00	0.03	0.95
fixed-effects	mu	random-effects	0.00	0.26	0.19
fixed-effects	tau_sq	fixed-effects	0.00	0.00	
fixed-effects	tau_sq	random-effects	-1.00	1.00	
random-effects	mu	fixed-effects	0.00	0.03	0.96
random-effects	mu	random-effects	0.00	0.13	0.95
random-effects	tau_sq	fixed-effects	0.00	0.01	0.98
random-effects	tau_sq	random-effects	0.02	0.22	0.95

the fixed effect model is the right one (i.e., when $\tau^2 = 0$). However, the bias in a conservative direction (confidence intervals are wider than they should be), which may be preferable. The fixed effect estimator has two problems. When the random effects assumption is correct, then the estimator gets the variance in estimates wrong, because it assumes it is zero. When it is not zero, there is bias. This leads to a second problem: the estimator confuses variability in true study effects for variability in *estimation* of the study effects, so the standard errors it produces are highly biased (coverage is about 0.2). Therefore, if we are even a little unsure of the true distribution, we should not trust the uncertainty estimates from the fixed effect model. And, of course, we know that there may be variation in true effects that we assumed away by adopting this estimator.

18.5 Multi-site studies

We declare a design for a coordinated research project in which multiple research teams combine data gathering and analysis strategies to address a common inquiry. Diagnosis clarifies the benefits and risks of coordination versus tailoring of treatments to contexts.

Nearly all social science is produced atomistically: individual scientists identify a question and a research strategy for answering it, apply the research strategy, and try to publish the results of that one study. Increasingly, there is a realization that, though this promotes discovery, it may not be the most efficient way to accumulate general insights. One reason is that scientists are opportunistic in the contexts and units they study. If inquiry values differ in the contexts and units that scientists choose to study from the population of contexts and units of general interest then we may not be learning general insights. One response to this problem of generalizability has been to promote replication of past studies in new contexts or with new populations. Another has been to develop methods for extrapolating from single studies, relying on variation within studies in unit

characteristics and effect heterogeneity. A third, which we explore here, is multi-site studies in which by design a study is conducted in multiple contexts in order to produce general insights by averaging effects across contexts and exploring how effects vary by context.

Working in more than one site is expensive. Fixed implementation and data collection costs may be duplicated in each site, and typically outcome measurement and other implementation details must be coordinated across sites. To justify conducting a study in multiple sites, at a minimum it should be clear what inferential gains are to be had relative to a study of the same size in a single site. Concentrating resources in the single site is likely to yield lower costs for the same total sample size.

The scientific value from studying multiple sites that could outweigh cost considerations come from the design's ability to improve our understanding of the generalizability of estimates. If we find a treatment has an effect of around 0.1 in five contexts, we have learned more about how effective that treatment generally is than if we had only learned in the first site the treatment effect is 0.1. In that single site case, we might wonder if the effect was larger or smaller or nonexistent in the other four sites.

How much we can learn about the generalizability of effects depends on how many sites we study, how we *select* the sites, our beliefs about effect heterogeneity, and the level of coordination across studies. The coordination level is important because its absence we might not be learning about the generalizability of our answers to a single inquiry but rather about one answer to many distinct inquiries.

We declare a multi-site study with five sites and 500 subjects per site. We define in our model 100 possible sites we could work in in order to be able to assess the generalizability of the answers our design yields. The model reflects the fact that there is a true study effect (unknown of course) for each site between 0 and 0.2 (note the 0.1 bump in the potential outcomes below).

We explicitly define sampling of sites — here complete random sampling — to call attention to the fact that how you sample sites shapes whether answers are generalizable. If the researchers select only feasible contexts to work in, we can learn generalizable insights from the design — but likely only about feasible contexts. We then sample individuals at random within sites. Treatment assignment is blocked at the site level, which is the same as complete random assignment within each site.

Estimation proceeds in two steps: first we calculate site-level effects, just as we do for a meta-analysis, then we meta-analyze those site-level effects. We use here a random effects model (see meta-analysis design library entry for a discussion of alternative modeling strategies), reflecting our expectation that true site effects differ.

Declaration 18.4.

```

estimate_study_effects <- function(formula, data) {
  data %>%
    group_by(sites) %>%
    do(tidy(lm_robust(formula, data = .))) %>%
    filter(term == "Z") %>%
    ungroup
}

n_study_sites <- 5
n_subjects_per_site <- 500

design <-
  declare_model(
    sites = add_level(
      N = 100,
      study_effect = seq(from = -0.1, to = 0.1, length.out = N)
    ),
    subjects = add_level(
      N = n_subjects_per_site,
      U = rnorm(N),
      potential_outcomes(Y ~ Z * (0.1 + study_effect) + U)
    )
  ) +
  declare_inquiry(PATE = mean(Y_Z_1 - Y_Z_0),
                 tau_sq = ) +
  declare_sampling(S = cluster_rs(clusters = sites, n = n_study_sites)) +
  declare_sampling(S = strata_rs(strata = sites, n = n_subjects_per_site)) +
  declare_assignment(Z = block_ra(blocks = sites, prob = 0.5)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_step(Y ~ Z, handler = estimate_study_effects) +
  declare_estimator(yi = estimate, sei = std.error, method = "REML",
                    model = rma_estimator, model_summary = rma_mu_tau, inquiry = "PATE",
                    label = "random-effects")

```

18.5.1 Coordinating treatments

Without coordination of research designs across sites in a multi-site study, researchers risk answering different questions in different sites. If the goal is to understand a common inquiry and how values of the inquiry vary across contexts, then each site must be providing answers to (close to) the same inquiry. To do so, researchers coordinate measurement strategies so as to collect the

same outcome measures and coordinate the details of treatment procedures to ensure they are studying the effects of the same intervention.¹

To fix ideas we might imagine an information campaign in which the common treatment uses radio dissemination of messages, but in some sites a lower tech strategy might be more effective (e.g. community meetings) and in others a higher tech strategy (e.g. Facebook ads) might be.

The disadvantage of coordinating is that researchers end up answering a *different* question than they started out with. In the worst case they end up answering one common question across all sites that is interesting in none. We explore these tradeoffs by declaring a modified multi-site design in which different treatments have different effects in different sites. There is a treatment that is expected to perform best on average, which researchers are interested in assessing. However qualitative insight and local ownership can enable researchers to select questions that are responsive to local conditions. We might imagine for instance that contextual research identifies a treatment that is most likely to be effective for a given context. This is the version that would be selected absent coordination (this type of self selection is often association with the “Roy model”). However it could also be that local selection results in selecting the *least* effective treatment: for example if local politicians has a hand in determining which anti-corruption intervention to implement.

We consider different possible designs that vary in terms of the *level* of coordination and the type of self selection if coordination fails. With no coordination, the selected treatment may be more or less effective than the researcher selected treatment; with full coordination there is no self selection and so suboptimal treatments might be imposed in different settings.

In this setting we can define multiple distinct estimands such as effect of the treatment that is chosen by the researcher, the effect of the best or worst treatment for each site, the average effect across all treatments, across sites, or the effect of the treatment that *would* be chosen via self selection in the absence of any coordination.

Our data strategy takes account of how self-selection plays out if coordination is imperfect (either accidentally or deliberately). In the design below a single variable Z indicates treatment (randomly assigned) but the outcome observed, Y , depends on which version of treatment is employed.

Declaration 18.5.

```
coordination <- 1
prob_select_pos <- .2
```

¹They may also wish to coordinate on details like sampling to ensure the same types of subjects are enrolled at each site and on the consent and enrollment procedures so as to avoid introducing selection or demand effects differentially across sites.

```

n_subjects_per_site <- 2500

design <-
  declare_model(
    sites = add_level(
      N = 5,
      tau_1 = rnorm(N, .1),
      tau_2 = rnorm(N),
      tau_3 = rnorm(N),
      compliant = runif(N) < coordination,
      selects_pos = simple_rs(N, prob_select_pos)),
    subjects = add_level(
      N = n_subjects_per_site,
      tau_mean = (tau_1 + tau_2 + tau_3)/3,
      tau_max = pmax(tau_1, tau_2, tau_3),
      tau_min = pmin(tau_1, tau_2, tau_3),
      tau_selected = tau_max*selects_pos + tau_min*(1-selects_pos),
      U = rnorm(N))) +
  declare_inquiry(
    PATE_1 = mean(tau_1),
    PATE_average = mean(tau_mean),
    PATE_best = mean(tau_max),
    PATE_worst = mean(tau_min),
    PATE_selected = mean(tau_selected)
  ) +
  declare_assignment(Z = block_ra(blocks = sites)) +
  declare_measurement(
    Y = Z*(compliant*tau_1 + (!compliant)*tau_selected) + U
  ) +
  declare_step(Y ~ Z, handler = estimate_study_effects) +
  declare_estimator(
    yi = estimate,
    sei = std.error,
    method = "REML",
    model = rma_estimator,
    model_summary = rma_mu_tau,
    inquiry = c(
      "PATE_1",
      "PATE_average",
      "PATE_best",
      "PATE_worst",
      "PATE_selected"
    ),
  )

```

```
    label = "random-effects"
)
```

```
designs <- redesign(design, coordination = (0:4)/4, prob_select_pos = (0:3)/3)
diagnoses <- diagnose_design(designs, sims = sims, bootstrap_sims = 100)
```

Figure 18.3 shows the distribution of estimands as a function of the level of coordination. The estimand for the (unknown) “best” treatment and the researcher controlled treatment do not vary with coordination. However the value of the self selected treatment does in the obvious way: if units self select into treatments with low or negative effects then the researcher controlled treatment trumps the self selected treatment. The opposite is true if units self select into treatments with more positive effects; in those case weaker control can put a focus on an estimand with stronger effects.

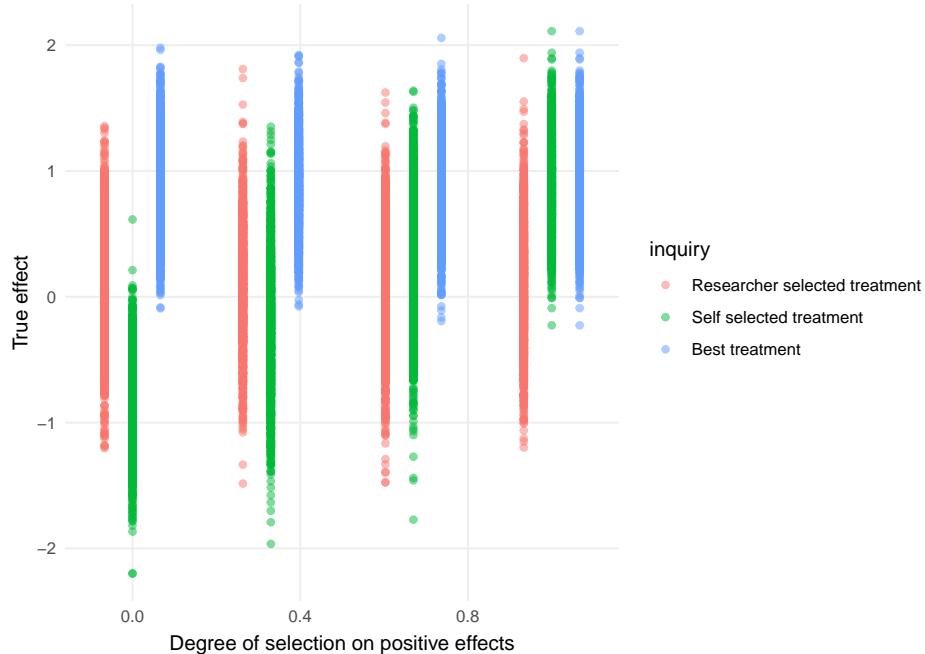


Figure 18.3: Estimands.

Figure 18.4 describes the bias associated with different strategies given imperfect coordination.

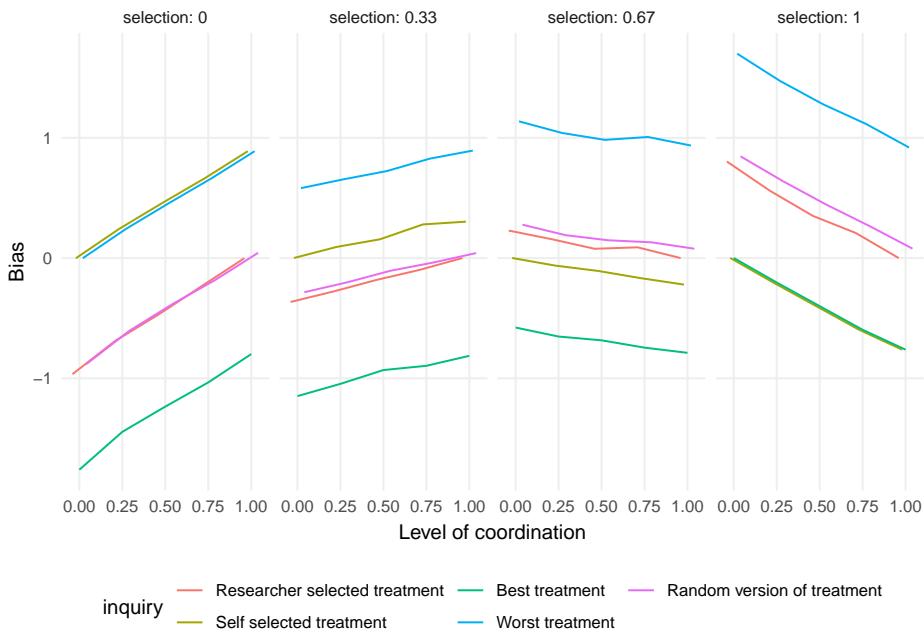


Figure 18.4: Bias for the for different estimands as a function of coordination and the probability of self selection into more treatments with more positive effects.

The take home message is that if you can ensure coordination you can get unbiased estimates for the treatment you select. More importantly, if you cannot ensure coordination then you can still get unbiased estimates but for the self-selected treatment. For moderate levels of coordination you do not get unbiased estimates for any of these estimands. You can neither claim to have estimated effects for a particular treatment or for whatever treatment a site would self select into.

364

Complex designs

18.5

Part IV

Research Design Lifecycle

Chapter 19

Research Design Lifecycle

A research design begins with a spark of inspiration or opportunity, develops through a planning phase, comes to fruition in the stages of realization, and adds to knowledge and influences decisions as it is integrated into collective scientific understanding. At each step in the research design lifecycle, your specification of M , I , D , and A shapes your choices and how others will learn from your work.

This part of the book works through stages of the research design lifecycle. While it is presented linearly, the steps are all intertwined by their shared connection to the research design. We describe in each entry how we can use the declaration, diagnosis, and redesign framework to make progress at each step. Not every research project will feature each and every stage. For example, prospective research designs like experiments and surveys often include pilot studies to learn about important unknown features of M before implementing the full studies. Retrospective studies, like textual analyses of speeches delivered to parliament, might not.

We divide the research design lifecycle into four broad categories: brainstorming, planning, realization, and integration. Brainstorming is the process of going from the kernel of a research idea to a tractable design. Planning includes all of the activities you undertake before data collection starts: conducting ethical reviews, obtaining approvals, organizing partnerships, securing funding, running pilots, gathering criticism, and filing analysis plans. Realization begins with the execution of the data strategy as planned, and continues through the inevitable changes that come with analytic challenges and scientific surprises. We trace realization through implementation, pivoting in response to unexpected developments, writing up results, reconciling planned and implemented designs, and responding to peer reviewers. In the final phase of a research project, the results are integrated into the scientific literature. Integration includes how the study will inform theories and decisions and also how the study will later be

reanalyzed, replicated, and meta-analyzed.

Chapter 20

Brainstorming

A research design starts from an idea, a kernel of a project. People arrive at their good ideas from every direction through idiosyncratic processes. Some people are inspired by their reading of academic literature. Others are sparked by a conversation with colleagues. Still others have their research questions thrust upon them by exigent circumstances. Ideas come from primary observation of social processes, reading secondary accounts by other authors, and our own past thoughts and lived experiences.

Whatever the process is that ignites the imagination, we can characterize what part or parts of the research design the idea is about. Ideas can be a piece of a model, a vague inquiry, a scrap of a data strategy, or an exciting answer strategy. An idea in the form of a model includes one or more nodes and one or more edges between them: for example, a treatment, a mediator, and an outcome. An inquiry is perhaps the most common kernel of a research project. It might be “does D cause Y.” A data strategy might be the discovery of a discontinuity in some administrative rule. A research project that starts with an answer strategy might be a new measurement technology that enables answering previously unanswerable – or unasked – questions.

The goal of brainstorming is to build a complete design from this kernel, regardless of where in the four elements M , I , D , or A the idea starts out. Sometimes a “theory-first” approach to developing a research project is unhelpfully contrasted with a “methods-first” approach. Better might be to describe the “theory-first” approach as starting with M or I and an “opportunity-first” approach as starting with D or A . We should seek to provide credible answers to important questions, and often progress in providing credible answers will start with new data and answer strategies. That said, unimportant questions aren’t worth the research investment, no matter the credibility of the answers – worse still are unreliable answers to the biggest questions in social science.

Brainstorming sessions with colleagues and mentors can help take a kernel of an

idea to a set of feasible research designs. What do participants in the brainstorming session need to know about your idea so they can effectively help identify possible designs? We suggest using the following “problem statement.¹” You need not have all the parts of the design worked out – that’s why you are having a brainstorming session! Systematizing what you do know is helpful. The more information you can provide, even if it is the range of possibilities rather than an exact specification.

Brainstorming document

Instructions: fill out any part you can!

Model

- What is the population of units of interest?
- How many units are in the population?
- What are the important variables that describe each unit?
- Can you represent your theory as a DAG?
- What parts of the theory are you more or less confident of?
- How would an alternative model describe a similar process?

Inquiry

- What main question about the theoretical model will the design address?
- Why is this question important for scholarship, the public, or decision-makers?
- Are there auxiliary inquiries that could be used to check model assumptions?
- For which units is the inquiry defined?

Data strategy

- How will you select cases or sample units from the population defined in the model?
- If randomizing treatments:
 - How many conditions will there be?
 - At what level will you randomize?
 - How many units can be assigned to each condition?
 - Which procedure will you use to randomize?
- If not randomizing treatments:
 - How do units come to receive different treatments?
 - Are there known processes that approximate random assignment
- Regardless of how units come to be treated:
 - How will you address the possibility of noncompliance?
 - How will you address the possibility of spillovers?
 - If there were no financial, logistical, or ethical constraints, what is the ideal experiment you would run?

¹The problem statement idea was developed by Graeme Blair, Darin Christensen, Erin Hartman, and Chad Hazlett for a research design seminar at UCLA.

- How will you measure outcomes?
 - What survey instrument or measurement tool will you use?
 - When will you measure outcomes? How many times?
 - How will you minimize measurement error and attrition?

Answer strategy

- How will you use the data that results from your proposed data strategy to produce answers to your inquiry?
 - What subset of the data will you use analyze?
 - What contrasts will you make to draw comparisons?
 - Which outcomes will you analyze?
 - What estimator will you use?
 - How will you estimate uncertainty in your estimates?

Chapter 21

Planning

We list “design early” first among our research design principles (Principle 3.1) because it suffuses our point of view about how to construct strong research designs. “Measure twice, cut once.” Research projects are very long journeys: going from the kernel of an idea to a published paper typically takes multiple years. Once the data strategy has been implemented, you’re stuck with it, so mindful planning beforehand is important.

The planning process changes designs. We work out designs that meet ethical as well as scientific standards, accommodate the needs of research partners, and operate within financial and logistical constraints. When we are insufficiently sure of key inputs to design declarations, we can run pilots, but we need to be careful about how we incorporate what we learn from them. Finally, when we write up a declaration or a PAP with a declaration, this can be a useful moment to get feedback from our peers to improve the design. We discuss each of these steps in this chapter.

21.1 Ethics

As researchers, we have ethical obligations beyond the requirements of national laws and the regulations of institutional review boards.

For a long time thinking about research ethics have been guided by the ideas in the Belmont report, that emphasize beneficence, respect for persons, and autonomy. Recently, more attention has been given to principles that extend beyond care for human subjects to include considerations for the well-being of collaborators and partners and the broader social impact of research. Social scientific professional associations have developed principles and guidelines to help think through these issues. Key references include:

- American Political Science Association ethics guidelines

- American Sociological Association Code of Ethics
- American Psychological Association Ethical Principles of Psychologists and Code of Conduct

The considerations at play vary across context and methods. For example, Teele (2021) describes ethical considerations in field experimentation, Humphreys (2015) focuses on development settings, Slough (2020) considers the ethics of field experimentation in the context of elections, and Wood (2006) and Baron and Young (2020) consider ethical challenges specific to field research in conflict settings.

However a common meta-principle underlying many of these contributions is the injunction to give prominent consideration to ethical issues: reflect on ethical dimensions of your work *ex ante* and report on ethical implications *ex post*. Lyall (2020) specifically connects ethical reflection to *ex ante* design considerations.

We encourage you to engage with ethical considerations in this way, early in the research design lifecycle. Some design declarations and diagnoses elide ethical considerations. For instance, a declaration that is diagnosand-complete for statistical power may tell you little about the level of care and respect accorded to subjects. Many declarations are diagnosand-complete for bias, but obtaining an unbiased treatment effect estimate is not always the highest goal.

Ethical diagnosands can be directly incorporated into the declare-diagnose-redesign framework. Diagnosands could include the total cost to participants, how many participants were harmed, the average level of informed consent measured by a survey about comprehension of study goals, or the risks of adverse events. More complex ethical diagnosands may be possible as well: Slough (2020) provides a formal analysis of the “aggregate electoral impact” diagnosand for experiments that take place in the context of elections. We consider two specific ethical diagnosands here, costs and potential harms, though many others may apply in particular research scenarios.

Costs. A common concern is that measurement imposes a cost on subjects, if only by wasting their time. Subjects’ time is a valuable resource they often donate willingly to the scientific enterprise by participating in a survey or other measurement. Although subjects’ generosity is sometimes repaid with financial compensation, in many scenarios direct payments are not feasible. Regardless of whether subjects are paid, the costs to subjects should be top of mind when designing the study.

Potential harms. Different realizations of the data from the same data strategy may differ in their ethical status. Ex-post, a study may not have ended up harming subjects, but ex-ante, there may have been a risk of harm (Baron and Young, 2020). The project’s ethical status depends on judgments about *potential* harms and *potential* participants: not only what did happen, but what could have happened. The potential harm diagnosand might be formalized as the maximum harm that could eventuate under any realization of the data strategy.

Researchers could then follow a minimax redesign procedure to find the design that minimizes this maximum potential harm.

When the design is diagnosed, we can characterize the ethical status of possible realizations of the design as well as the ethicality of the *distribution* of these realizations. Is the probability of harm minimal “enough”? Is the degree of informed consent sufficient? Given that these characteristics vary across designs and across realizations of the same design, writing down concretely both the measure of the ethical status and the ethical threshold can help structure thinking. These diagnoses and the considerations that inspire them can be shared in funding proposals, preanalysis plans, or other report. Articulating them in a design may help clarify whether proper account was taken of risks *ex ante*, or, more usefully, remind researchers to be sure to take account of them.

Often, once an ethical threshold is met, we select among feasible designs based on research design criteria such as statistical power and bias. This approach has appeal since we should only implement designs that meet the relevant research community’s ethical standards. However, dichotomizing designs into “ethical” and “unethical” is a difficult task in general. Instead, we should continue to assess ethical considerations alongside the quality of the research design. Even among ethical designs, we still face tradeoffs between how much time is asked of subjects and the risk of harm. We should select designs that appropriately weight these considerations against other desiderata and be able to articulate and justify the of weighting used. When obtaining a credible answer would come at too high an ethical cost, the study may need to be scrapped altogether.

21.2 Approvals

When researchers sit at universities in the United States, research must be approved by the university’s institutional review board (IRB) under the federal regulation known as the “Common Rule.” Similar research review bodies exist at universities worldwide and at many independent research organizations and think tanks. Though these boards are commonly thought to judge research ethics, in fact, they mainly exist to protect their institution from liability for research gone awry (King and Sands, 2015). Accordingly, a researcher’s obligation to consider their study’s ethics is neither constrained nor checked by IRBs. Instead, a set of idiosyncratic rules and practices specific to each institution are checked (Schrag, 2010). The researcher, as a result, remains responsible for their own ethical decision about whether or not to move forward with the research. That said, the IRB process is not necessarily without benefit. In some cases, useful discussions can be had with IRB board members about study decisions, and the approval itself may protect the researcher from some kinds of liability.

Laws and regulations at the country, state or province, or municipality level may also govern research on human subjects besides the IRB. Many countries require

human subjects approval, especially for health research, in addition to the approvals researchers must seek from their home institutions. These approvals serve a similar purpose to the home institution IRB, but by virtue of their authority coming from the context in which the research is conducted rather than from far away bureaucrats, they may serve to more directly protect human subjects.

Though these bodies' goals differ from the broader ethical aims social scientists hold, design diagnosis may also be useful here. Many IRBs ask researchers to describe tradeoffs between the costs and benefits to research subjects. In some cases, researchers are asked to defend research design choices that provide benefits to science, but where the only direct effects on participants are costs with no immediate benefits. Defining the costs and benefits to participants in terms of their time and money and the compensation provided by researchers, if any, can both simplify communication with IRBs and provide tools for researchers to more easily clarify these tradeoffs for themselves. The expected benefit and expected cost can be diagnosed across possible realizations of the design. The design diagnosis can highlight tradeoffs between the value to participants and the scientific value in the form of standard diagnostics. Rather than argue in the abstract about these quantities, they can be simulated and described formally through declaration and diagnosis.

21.3 Partners

Partnering with third-party organizations in research entails cooperating to intervene in the world or to measure outcomes. Researchers seek to produce (and publish) scientific knowledge; they work with political parties, government agencies, nonprofit organizations, and businesses to learn more than they could if they worked independently. These groups work with researchers to learn about how to achieve their own organizational goals. Governments may want to expand access to healthcare, corporations to improve their ad targeting, and nonprofits to demonstrate program impact to funding organizations.

In the best-case scenario, the goals of the researchers and partner organizations are aligned. When the scientific question to be answered is the same as the practical question the organization cares about, the gains from cooperation are clear. The research team gains access to the organization's financial and logistical capacity to act in the world, and the partner organization gains access to the researchers' scientific expertise. Finding the right research partner almost always amounts to finding an organization with a common – or at least not conflicting – goal. Selecting a research design amenable to both parties requires understanding each partners' private goals. Research design declaration and diagnosis can help with this problem by formalizing tradeoffs between the two sets of goals.

One frequent divergence between partner and researcher goals is that partner

organizations often want to learn, but they care most about their primary mission. This dynamic is sometimes referred to as the “learning versus doing” tradeoff. (In business settings, this tradeoff goes by names like “learning versus earning” or “exploration versus exploitation”). An aid organization cares about delivering their program to as many people as possible. Learning whether the program has the intended effects on the outcomes of interest is obviously also important, but resources spent on evaluation are resources *not* spent on program delivery.

Research design diagnosis can help navigate the learning versus doing tradeoff. One instance of the tradeoff is that the proportion of units that receive a treatment represents the rate of “doing,” but this rate also affects the amount of learning. In the extreme, if all units are treated, we can’t measure the effect of the treatment. The tradeoff here is represented in Figure 21.1, which shows the study’s power versus the proportion treated (top facet) and the partner’s utility (bottom facet). The researchers have a power cutoff at the standard 80% threshold. The partner also has a strict cutoff: they need to treat at least 2/3 of the sample to fulfill a donor requirement.

Researchers might simply ignore the proportion treated and select the design with the highest power in the absence of partners. With a partner organization, the researcher might use this graph in conversation with the partner to jointly select the design that has the highest power that has a sufficiently high proportion treated to meet the partner’s needs. This is represented in the “zone of agreement” in gray: in this region, the design has at least 80% power and at least two-thirds of the sample are treated. Deciding within this region involves a tradeoff between power (which is decreasing in the proportion treated here) and the partner’s utility (which is increasing in proportion treated). The diagnosis surfaces the zone of the agreement and clarifies the choice between designs in that region.¹

Choosing the proportion treated is one example of integrating partner constraints into research designs. A second common problem is that there are a set of units that must be treated or that must not be treated for ethical or political reasons (e.g., the home district of a government partner must receive the treatment). If these constraints are discovered after treatment assignment, they lead to noncompliance, which may substantially complicate the analysis of the experiment and even prevent providing an answer to the original inquiry. Gerber and Green (2012) recommend, before randomizing treatment, exploring possible treatment assignments with the partner organization and using this exercise to elicit the set of units that must or cannot be treated. King et al. (2007) describe a “politically-robust” design, which uses pair-matched block randomization. In this design, when any unit is dropped due to political constraints, the whole pair is dropped from the study.²

¹Unfortunately, some partnerships simply will not work out if the zone of agreement is empty.

²This procedure is prone to bias for the average treatment effect among the “political feasible” units if within some pairs, one unit is treatable but the other is not.

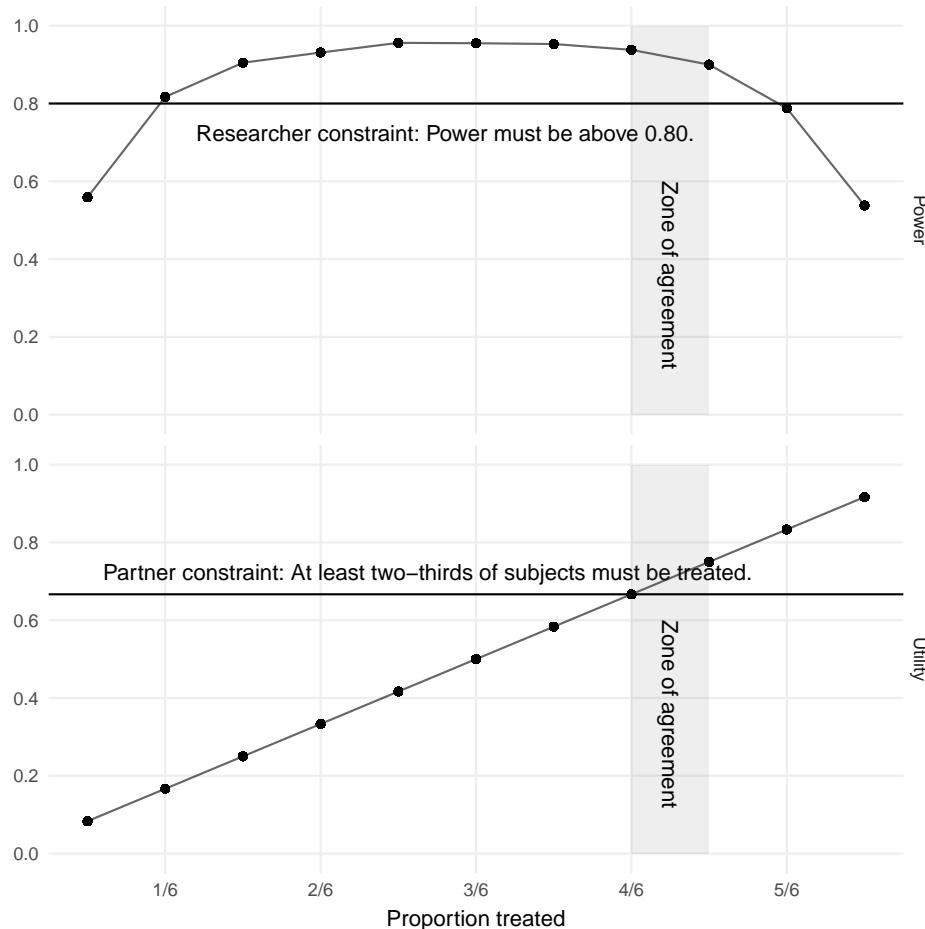


Figure 21.1: Navigating research partnerships.

A major benefit of working with partners is their deep knowledge of the substantive area. For this reason, we recommend involving them in the design declaration and diagnosis process. How can we develop intuitions about the means, variances, and covariances of the variables to be measured? Ask your partner for their best guesses, which may be far more educated than your own. For experimental studies, solicit your partner's beliefs about the magnitude of the treatment effect on each outcome variable, subgroup by subgroup if possible. Engaging partners in the declaration process improves design – and it very quickly sharpens the discussion of key design details. Sharing your design diagnoses and mock analyses *before* the study is launched can help to build a consensus around the study's goals.

21.4 Funding

Higher quality designs usually come with higher costs. Collecting original data is more expensive than analyzing existing data, but collecting new data may be more or less costly depending on the ease of contacting subjects or conducting measurements. As a result, including cost diagnosands in research design diagnosis can directly aid data strategy decision-making. These diagnosands may usefully include both average cost and maximum cost. Researchers may make different decisions about cost: in some cases, the researcher will select the “best” design in terms of research design quality subject to a budget constraint. Others will choose the cheapest among similar quality designs to save money for future research. Diagnosis can help identify each set and decide among them.

To relax the budget constraint, researchers apply for funding. Funding applications have to communicate important features of the proposed research design. Funders want to know why the study would be useful, important, or interesting to scholars, the public, or policymakers. They also want to ensure that the research design provides credible answers to the question and that the research team is capable of executing the design. Since it’s their money on the line, funders also care that the design provides good value-for-money.

Researchers and funders have an information problem. Applicants wish to obtain as large a grant as possible for their design but have difficulty credibly communicating the quality of their design given the subjectivity of the exercise. On the flip side, funders wish to get the most value-for-money in the set of proposals they decide to fund and have difficulty assessing the quality of proposed research. Design declaration and diagnosis provide a partial solution to the information problem. A common language for communicating the proposed design and its properties can communicate the value of the research under design assumptions that can be understood and interrogated by funders.

Funding applications should include a declaration and diagnosis of the proposed design. In addition to common diagnosands such as bias and efficiency, two special diagnosands may be valuable: cost and value-for-money. The cost can be included for each design variant as a function of design features such as sample size, the number of treated units, and the duration of survey interviews. Simulating the design across possible realizations of each variant explains how costs vary with choices the researcher makes. Value-for-money is a diagnosand that is a function of cost and the amount learned from the design.

In some cases, funders request applicants to provide multiple options and multiple price points or make clear how a design could be altered so that it could be funded at a lower level. Redesigning over differing sample sizes communicates how the researcher conceptualizes these options and provides the funder with an understanding of tradeoffs between the amount of learning and cost in these design variants. Applicants could use the redesign process to justify the high cost of their request directly in terms of the amount learned.

Ex-ante power analyses are required by an increasing number of funders. Current practice, however, illustrates the crux of the misaligned incentives between applicants and funders. Power calculators online have difficult-to-interrogate assumptions built in and cannot accommodate the specifics of many common designs (Blair et al., 2020). As a result, existing power analyses can demonstrate that almost any design is “sufficiently powered” by changing expected effect sizes and variances. Design declaration is a partial solution to this problem. By clarifying the assumptions of the design in code, applicants can more clearly link the assumptions of the power analysis to the specifics of the design setting.

Finally, design declarations can also help funders compare applications on standard scales: root mean-squared-error, bias, and power. They also want to weigh considerations like importance and fit. Moving design considerations onto a common scale takes some of the guesswork out of the process and reduces reliance on researcher claims about properties.

21.5 Piloting

Designing a research study always entails relying on a set of beliefs, what we’ve referred to as the set of possible models in M . Choices like how many subjects to sample, which covariates to measure, which treatments to allocate, and depend on beliefs about treatment effects, the correlations of the covariates with the outcome, and the variance of the outcome.

We may have reasonably educated guesses about these parameters from past studies or theory. Our understanding of the nodes and edges in the causal graph of M , expected effect sizes, the distribution of outcomes, feasible randomization schemes, and many other features are directly selected from past research or chosen based on a literature review of past studies.

Even so, we remain uncertain about these values. One reason for the uncertainty is that our research context and inquiries often differ subtly from previous work. Even when replicating an existing study as closely as possible, difficult-to-intuit features of the research setting may have serious consequences for the design. Moreover, our uncertainty about a design parameter is often the very reason for conducting a study. We run experiments *because* we are uncertain about the average treatment effect. If we knew the ATE for sure, there would be no need to run the study. Frustratingly, we always have to design using parameters whose values we are unsure of.

The main goal of pilot studies is to reduce this uncertainty over the possible models in M so that the main study can be designed, taking into account design parameters closer to the true values. Pilots take many forms: focus groups to learn how to ask survey questions, small-scale tests of measurement tools, even miniature versions of the main study on a smaller scale. We want to learn things like the distribution of outcomes, how covariates and outcomes might be corre-

lated, or how feasible the assignment, sampling, and measurement strategies are.

Almost by definition, pilot studies are inferentially weaker than main studies. We turn to them in response to constraints on our time, money, and capacity. If we were not constrained, we would run a first full-size study, learn what is wrong with our design, then run a corrected full-size study. Since running multiple full studies is too expensive or otherwise infeasible, we run either smaller mini-studies or test out only a subset of the elements of our planned design. Accordingly, the diagnosands of a pilot design will not measure up to those of the main design. Pilots have much lower statistical power and may suffer from higher measurement error and less generalizability. Accordingly, the goal of pilot studies should not be to obtain a preliminary answer to the main inquiry, but instead to learn the information that will make the main study a success.

Like main studies, pilot studies can be declared and diagnosed – but importantly, the diagnosands for main and pilot studies need not be the same. Statistical power for an average treatment effect may be an essential diagnosand for the main study, but owing to their small size, power for pilot studies will typically be abysmal. Pilot studies should be diagnosed with respect to the decisions they imply for the main study.

Figure 21.2 shows the relationship between effect size and the sample size required to achieve 80% statistical power for a two-arm trial using simple random assignment. Uncertainty about the true effect size has enormous design consequences. If the effect size is 0.17, we need about 1,100 subjects to achieve 80% power. If it's 0.1, we need 3200.

Suppose we have prior beliefs about the effect size that can be summarized as a normal distribution centered at 0.3 with a standard deviation of 0.1, as in the bottom panel of Figure 21.2. We could choose a design that corresponds to this best guess, the average of our prior belief distribution. If the true effect size is 0.3, then a study with 350 subjects will have 80% power.

However, redesigning the study to optimize for the “best guess” is risky because the true effect could be much smaller than 0.3. Suppose we adopt the redesign heuristic of powering the study for an effect size at the 10th percentile of our prior belief distribution, which works out here to be an effect size of 0.17. Following this rule, we would select a design with 1100 subjects.

Now suppose the true effect size is, in actuality, only 0.1, so we would need to sample 3200 subjects for 80% power. The power of our chosen 1100-subject design is a mere 38%. Here we see the consequences of having incorrect prior beliefs: our ex-ante guess of the effect size was too optimistic. Even taking what we thought of as a conservative choice – the 10th percentile redesign heuristic – we ended up with too small a study.

A pilot study can help researchers update their priors about important design parameters. If we do a small scale pilot with 100 subjects, we'll get a noisy but

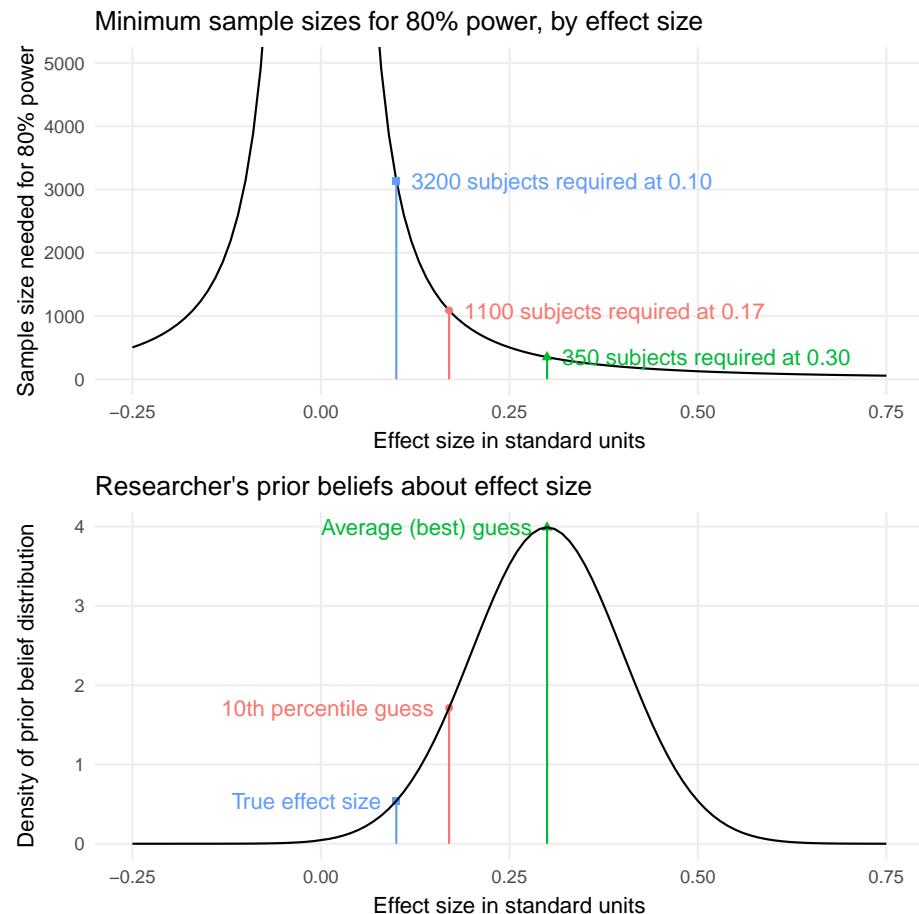


Figure 21.2: Minimum required sample sizes and uncertainty over effect size

unbiased estimate of the true effect size. We can update prior beliefs by taking a precision weighted average of our priors and the estimate from the pilot, where the weights are the inverse of the variance of each guess. Our posterior beliefs will be closer to the truth, and our posterior uncertainty will be smaller. If we then follow the heuristic of powering the 10th percentile of our (now posterior) beliefs about effect size, we will have come closer to correctly powering our study. Figure 21.3 shows how large the studies would be, depending on how the pilot study came out if we were to follow the 10th percentile decision rule. On average, the pilot leads us to design the main study with 1800 subjects, sometimes more and sometimes less.

This exercise reveals that a pilot study can be quite valuable. Without a pilot study, we would chose to sample 1100 subjects, but since the true effect size is only 0.1 (not our best guess of 0.3), the experiment would be underpowered.

The pilot study helps us correct our diffuse and incorrect prior beliefs. However, since the pilot is small, we don't update our priors all the way to the truth. We still end up with a main study that is on average too small (1800), with a corresponding power of 56%. That said, a 56% chance of finding a statistically significant result is better than a 38% chance.

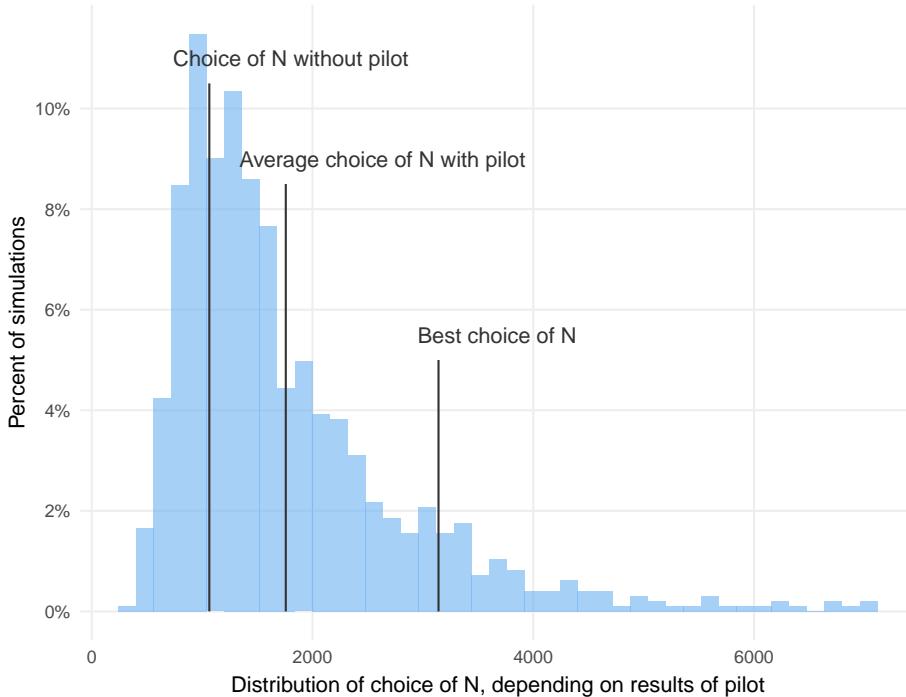


Figure 21.3: Distribution of post-pilot sample size choices

In summary, pilots are most useful when we are uncertain – or outright wrong – about important design parameters. This uncertainty can often be shrunk by quite a bit without running pilot studies by meta-analyzing past empirical studies. Some things are hard to learn by reading others' work; pilot studies are especially useful tools for learning about those things.

21.6 Criticism

A vital part of the research design process is gathering criticism and feedback from others. Timing is delicate here. Asking for comments on an underdeveloped project can sometimes lead to brainstorming sessions about what research questions one might look into. Such unstructured sessions can be quite useful but essentially restarts the research design lifecycle from the beginning. Sharing work only after a full draft has been produced is worse since the data strategy

will have already yielded the realized data. The investigators may have become attached to favored answer strategies and interpretations. While critics can always suggest changes to *I* and *A* post-data collection, an almost-finished project is fundamentally constrained by the data strategy as it was implemented.

The best moments to seek advice come before registering preanalysis plans or, if not writing a PAP, before implementing major data strategy elements. The point is not to seek advice exclusively on sampling, assignment, or measurement procedures; the important thing is that there's still time to modify those design elements (Principle 3.1). Feedback about the design as a whole can inform changes to the data strategy before it is set in stone.

Feedback will come in many forms. Sometimes the comments are directly about diagnosands. The critic may think the design has too many arms and won't be well-powered for many inquiries. Or they may be concerned about bias due to excludability violations or selection issues. These comments are especially useful because they can easily be incorporated in design diagnosis and redesign exercises.

Other comments are harder to pin down. A fruitful exercise in such cases is to understand how the criticism fits in to *M*, *I*, *D*, and *A*. Comments like, "I'm concerned about external validity here" might seem to be about the data strategy. If the units were not randomly sampled from some well-specified population, we can't generalize from the sample to the population. But if the inquiry is not actually a population quantity, then this inability to use sample data to estimate a population quantity is irrelevant. The question then becomes whether knowing the answer to your sample inquiry helps make theoretical progress or whether we need to generalize – to switch the inquiry to the population quantity to make headway. Critics will not usually be specific about how their criticism relates to each element of design, so it is up to the criticism-seeker to understand the implications for design.

Sometimes we seek feedback from smart people, but they do not immediately understand the design setting. If the critic hasn't absorbed or taken into account important features of the design, their recommendations and amendments may be off-base. For this reason, it's important to communicate the design features – the model, inquiry, data strategy, and answer strategy – at a high enough level of detail that the critic is up to speed before passing judgment.

21.7 Preanalysis Plan

In many research communities, it is becoming standard practice to publicly register a pre-analysis plan (PAP) before implementing some or all of the data strategy. PAPs serve many functions, but most importantly, they clarify which design choices were made before data collection and which were made after. Sometimes – perhaps every time! – we conduct a research study, aspects of *M*, *I*, *D*, and *A* shift along the way. A concern is that they shift in ways that

invalidate the apparent conclusions of the study. For example, “*p*-hacking” is the shady practice of trying out many regression specifications until the *p*-value associated with an important test attains statistical significance. PAPs protect researchers by communicating to skeptics *when* design decisions were made. If the regression specification was detailed in a PAP posted before any data were collected, the test could not be the result of a *p*-hack.

PAPs are sometimes misinterpreted as a binding commitment to report all pre-registered analyses and nothing but. This view is unrealistic and unnecessarily rigid. While we think that researchers should report all pre-registered analyses *somewhere* (see Section 22.2 on “populated PAPs”), study writeups inevitably deviate in some way from the PAP – and that’s a good thing. Researchers learn more by conducting research. This learning can and should be reflected in the finalized answer strategy.

Our hunch is that the main consequence of actually writing a PAP is improving the research design itself. Just like research design declaration forces us to think through the details of our model, inquiry, data strategy, and answer strategy, describing those choices in a publicly-posted document surely causes deeper reflection about the design. In this way, the main audience for a PAP is the study authors themselves.

What belongs in a PAP? Recommendations for the set of decisions that should be specified in a PAP remain remarkably unclear and inconsistent across research communities. PAP templates and checklists are proliferating, and the number of items they suggest ranges from nine to sixty. PAPs themselves are becoming longer and more detailed. Some in the American Economic Association and Evidence in Governance and Politics (EGAP) study registries reach hundreds of pages as researchers seek to be ever more comprehensive. Some registries emphasize the registration of the hypotheses to be tested, while others emphasize the registration of the tests that will used. In a review of many PAPs, Ofosu and Posner (2021) find considerable variation in how often analytically-relevant pieces of information appear in posted plans.

In our view a PAP should center on a design declaration. Currently, most PAPs focus on the answer strategy *A*: what estimator to use, what covariates to condition on, and what subsets of the data to include. But of course, we also need to know the details of the data strategy *D*: how units will be sampled, how treatments will be assigned, and how the outcomes will be measured. We need these details to assess the properties of the design and gauge whether the principles of analysis respecting sampling, treatment assignment, and measurement procedures are being followed. We need to know about the inquiry *I* because we need to know the target of inference. A significant concern is “outcome switching,” wherein the eventual report focuses on different outcomes than initially intended. When we switch outcomes, we switch inquiries! We need enough of the model *M* in the plan to describe *I* in sufficient detail. In short, a design declaration is what belongs in a PAP because a design declaration specifies all of the analytically-relevant design decisions.

In addition to a design declaration, a PAP should include mock analyses conducted on simulated data. If the design declaration is made formally in code, creating simulated data that resemble the eventually realized data is straightforward. We think researchers should run their answer strategy on the mock data, creating mock figures and tables that will ultimately be made with real data. In our experience, *this* is the step that really causes researchers to think hard about all aspects of their design.

PAPs can, optionally, include design diagnoses in addition to declarations, since it can be informative to describe why a particular design was chosen. For this reason, a PAP might include estimates of diagnosands like power, root-mean-squared-error, or bias. If a researcher writes in a PAP that the power to detect a very small effect is large, then if the study comes back null, the eventual writeup can much more credibly rule out “low power” as an explanation for the null.

21.7.1 Example

In this section, we provide an example of how to supplement a PAP with a design declaration. We follow the actual PAP for Bonilla and Tillery (2020), which was posted to the As Predicted registry. The study’s goal is to estimate the causal effects of alternative framings of Black Lives Matter (BLM) on support for the movement among Black Americans overall and among subsets of the Black community. These study authors are models of research transparency: they prominently link to the PAP in the published article, they conduct no non-preregistered analyses except those requested during the review process, and their replication archive includes all materials required to confirm their analyses, all of which we were able to reproduce exactly with minimal effort. Our goal with this section is to show how design declaration can supplement and complement existing planning practices.

21.7.1.1 Model

The authors write in their PAP:

We hypothesize that: H1: Black Nationalist frames of the BLM movement will increase perceived effectiveness of BLM among African American test subjects. H2: Feminist frames of the BLM movement will increase perceived effectiveness of BLM among African American women, but decrease perceived effectiveness in male subjects. H3: LGBTQ and Intersectional frames of the BLM movement will have no effect (or a demobilizing effect) on the perceived effectiveness of BLM African American subjects.

These hypotheses reflect a model of coalition politics that emphasizes the tensions induced by overlapping group identities. Framing the BLM movement as feminist or pro-LGBTQ may increase support among Black women or Black LGBTQ identifiers, but that increase may come at the expense of support among Black men or Black Americans who do not identify as LGBTQ. Similarly, this

model predicts that subjects with stronger attachment to their Black identity will have a larger response to a Black nationalist framing of BLM than those with weaker attachments.

The model also includes beliefs about the distributions of gender, LGBTQ status, and Black identity strength. In the data strategy, Black identity was measured with the standard linked fate measure. Other background characteristics that may be correlated with BLM support include age, religiosity, income, education, and familiarity with the movement, so these are included in M as well.

The study's focus will be on the causal effects of nationalism, feminism, and intersectional frames relative to a general description of the Black Lives Matter movement. Model beliefs about treatment effect heterogeneity are embedded in the model declaration. The effect of the nationalism treatment is hypothesized to be stronger, the greater subjects' sense of linked fate; the effect of the feminism treatment should be negative for men but positive for women; the effect of the intersectionality treatment should be positive for LGBTQ identifiers, but negative for non-identifiers.

```

rescale <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

likert_cut <- function(x) {
  as.numeric(cut(x, breaks = c(-100, 0.1, 0.3, 0.6, 0.8, 100), labels = 1:5))
}

model <-
declare_model(
  N = 800,
  female = rbinom(N, 1, prob = 0.51),
  lgbtq = rbinom(N, 1, prob = 0.05),
  linked_fate = sample(1:5, N, replace = TRUE,
                       prob = c(0.05, 0.05, 0.15, 0.25, 0.5)),
  age = sample(18:80, N, replace = TRUE),
  religiosity = sample(1:6, N, replace = TRUE),
  income = sample(1:12, N, replace = TRUE),
  college = rbinom(N, 1, prob = 0.5),
  blm_familiarity = sample(1:4, N, replace = TRUE),
  U = runif(N),
  blm_support_latent = rescale(
    U + 0.1 * blm_familiarity +
    0.45 * linked_fate +
    0.001 * age +
    0.5)
)

```

```

    0.25 * lgbtq +
    0.01 * income +
    0.1 * college +
    -0.1 * religiosity),
# potential_outcomes
blm_support_Z_general =
likert_cut(blm_support_latent),
blm_support_Z_nationalism =
likert_cut(blm_support_latent + 0.01 +
            0.01 * linked_fate +
            0.01 * blm_familiarity),
blm_support_Z_feminism =
likert_cut(blm_support_latent - 0.02 +
            0.07 * female +
            0.01 * blm_familiarity),
blm_support_Z_intersectional =
likert_cut(blm_support_latent - 0.05 +
            0.15 * lgbtq +
            0.01 * blm_familiarity)
)

```

21.7.1.2 Inquiry

The inquiries for this study naturally include the average effects of all three treatments relative to the “general” framing, as well as the differences in average effects for subgroups. When describing their planned analyses, the authors write:

We will also look at differences in responses between those indicating a pre-treatment familiarity BLM (4-Extensive knowledge to 1-Never heard of BLM), gender (particularly on the Feminist treatment), linked fate (particularly on the Nationalist treatment), and LGBT+ affiliation (particularly on the LGBT+ treatment), though we are not necessarily expecting these moderations to have a strong effect because samples may lack adequate representation.

In the code below, we specify how each treatment effect changes with its corresponding covariate X with $\frac{\text{cov}(i, X)}{V(X)}$, which is identical to the difference-in-difference for the binary covariates (female and lgbtq) and is the slope of the best linear predictor of how the effect changes over the range of linked_fate, and blm_familiarity which we are treating as quasi-continuous here.

```

slope <- function(y, x) { cov(y, x) / var(x) }

inquiry <-
declare_inquiries(
  # Average effects
  ATE_nationalism =
    mean(blm_support_Z_nationalism - blm_support_Z_general),
  ATE_feminism =
    mean(blm_support_Z_feminism - blm_support_Z_general),
  ATE_intersectional =
    mean(blm_support_Z_intersectional - blm_support_Z_general),

  # Overall heterogeneity w.r.t. blm_familiarity
  DID_nationalism_familiarity =
    slope(blm_support_Z_nationalism - blm_support_Z_general,
          blm_familiarity),
  DID_feminism_familiarity =
    slope(blm_support_Z_feminism - blm_support_Z_general,
          blm_familiarity),
  DID_intersectional_familiarity =
    slope(blm_support_Z_intersectional - blm_support_Z_general,
          blm_familiarity),

  # Treatment-specific heterogeneity
  DID_nationalism_linked_fate =
    slope(blm_support_Z_nationalism - blm_support_Z_general,
          linked_fate),
  DID_feminism_gender =
    slope(blm_support_Z_feminism - blm_support_Z_general,
          female),
  DID_intersectional_lgbtq =
    slope(blm_support_Z_intersectional - blm_support_Z_general,
          lgbtq)
)

```

21.7.1.3 Data strategy

This study's subjects are 800 Black Americans recruited by the survey firm Qualtrics using a quota sampling procedure. We omit this sampling step in our declaration: 800 subjects are described in the model declaration above. The reason is that, as is common practice in the analysis of survey experiments on convenience samples, the authors do not formally extrapolate from their data to make generalizations about the population of Black Americans. The inquiries

they study are sample average effects. If the authors had used a different sampling strategy, such as using random sampling through random digit dialing, we would have defined the population from which they were sampling and the random sampling procedure.

After subjects' background characteristics were measured, they were assigned to one of four treatment conditions. Since the survey was conducted on Qualtrics, we assume that the authors used the built-in randomization tools, which use simple (Bernoulli) random assignment.

```
data_strategy <-
  declare_assignment(
    Z = simple_ra(
      N,
      conditions =
        c("general", "nationalism", "feminism", "intersectional"),
      simple = TRUE
    )
  ) +
  declare_measurement(blm_support = reveal_outcomes(blm_support ~ Z))
```

21.7.1.4 Answer strategy

The authors write:

We will run an OLS regression predicting the support for, effectiveness of, and trust in BLM on each treatment condition. [...] We will also look at differences in responses between those indicating a pre-treatment familiarity BLM (4-Extensive knowledge to 1-Never heard of BLM), gender (particularly on the Feminist treatment), linked fate (particularly on the Nationalist treatment), and LGBT+ affiliation (particularly on the LGBT+ treatment), though we are not necessarily expecting these moderations to have a strong effect because samples may lack adequate representation. We plan to conduct analyses without controls. As we will check for between group balance, we may also run OLS analyses with demographic controls (age, linked fate, gender, sexual orientation, religiosity, income, education, and ethnic or multi-racial backgrounds), and will report differences in OLS results.

In DeclareDesign, this corresponds to five estimators, with two shooting at the ATEs and three shooting at the differences-in-differences. We use OLS for all five. The majority of the code is bookkeeping to ensure we match the right regression coefficient with the appropriate inquiry.

```

answer_strategy <-
  declare_estimator(
    blm_support ~ Z,
    term = c("Znationalism", "Zfeminism", "Zintersectional"),
    inquiry =
      c("ATE_nationalism", "ATE_feminism", "ATE_intersectional"),
    label = "OLS") +
  declare_estimator(
    blm_support ~ Z + age + female + as.factor(linked_fate) + lgbtq,
    term = c("Znationalism", "Zfeminism", "Zintersectional"),
    inquiry =
      c("ATE_nationalism", "ATE_feminism", "ATE_intersectional"),
    label = "OLS with controls") +
  declare_estimator(
    blm_support ~ Z*blm_familiarity,
    term = c("Znationalism:blm_familiarity",
             "Zfeminism:blm_familiarity",
             "Zintersectional:blm_familiarity"),
    inquiry = c("DID_nationalism_familiarity",
               "DID_feminism_familiarity",
               "DID_intersectional_familiarity"),
    label = "DID_familiarity") +
  declare_estimator(
    blm_support ~ Z * linked_fate,
    term = "Zfeminism:linked_fate",
    inquiry = "DID_nationalism_linked_fate",
    label = "DID_nationalism_linked_fate") +
  declare_estimator(
    blm_support ~ Z * female,
    term = "Zfeminism:female",
    inquiry = "DID_feminism_gender",
    label = "DID_feminism_gender") +
  declare_estimator(
    blm_support ~ Z * lgbtq,
    term = "Zintersectional:lgbtq",
    inquiry = "DID_intersectional_lgbtq",
    label = "DID_intersectional_lgbtq")

```

21.7.1.5 Mock analysis

Putting it all together, we can declare the complete design and draw mock data from it.

Table 21.1: Mock analysis from Bonilla and Tillery design.

ID	female	lgbtq	linked_fate	age	religiosity	income	college	blm_familiarity	U
001	0	0	3	27	1	11	1	4	0.976
002	0	0	4	44	6	4	1	1	0.274
003	1	0	3	78	2	9	1	2	0.349
004	0	0	5	23	5	4	1	3	0.929
005	0	0	4	69	5	1	0	4	0.351

Declaration 21.1.

```
design <- model + inquiry + data_strategy + answer_strategy
mock_data <- draw_data(design)
```

The table below shows a mock analysis of average effects (estimated with and without covariate adjustment) as well as the heterogeneous effects analyses with respect to the quasi-continuous moderators.

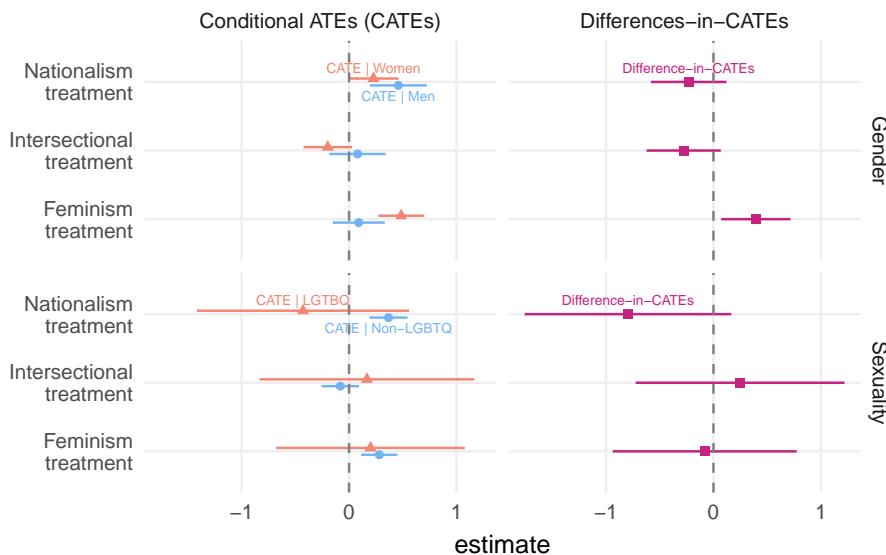


Figure 21.4: Mock coefficient plot from Bonilla and Tillery design.

	Model 1	Model 2	Model 3	Model 4
(Intercept)	3.629 (0.059)	1.283 (0.104)	1.383 (0.146)	3.284 (0.140)
Znationalism	0.339 (0.089)	0.305 (0.048)	0.101 (0.199)	0.056 (0.217)
Zfeminism	0.284 (0.084)	0.208 (0.049)	0.543 (0.207)	0.361 (0.198)
Zintersectional	0.073 (0.087)	0.041 (0.050)	0.169 (0.230)	0.031 (0.208)
female		0.071 (0.035)		
lgbtq		0.279 (0.110)		
age		0.000 (0.001)		
religiosity		0.147 (0.010)		
income		0.013 (0.005)		
college		0.165 (0.035)		
linked_fate		0.549 (0.015)	0.561 (0.035)	
blm_familiarity		0.173 (0.017)		0.141 (0.052)
Znationalism:linked_fate			0.067 (0.048)	
Zfeminism:linked_fate			0.075 (0.050)	
Zintersectional:linked_fate			0.061 (0.055)	
Znationalism:blm_familiarity				0.147 (0.081)
Zfeminism:blm_familiarity				0.034 (0.074)
Zintersectional:blm_familiarity				0.040 (0.079)
R ²	0.038	0.701	0.567	0.081
Adj. R ²	0.034	0.697	0.563	0.073
Num. obs.	800	800	800	800
RMSE	0.882	0.494	0.593	0.864

p < 0.001; *p* < 0.01; *p* < 0.05

Table 21.2: Mock regression table from Bonilla and Tillery design.

Table 21.3: Design diagnosis for Bonilla and Tillery design.

Estimand	Estimator	Bias	Power
ATE feminism	OLS	-0.002	0.527
ATE feminism	OLS with controls	-0.003	0.851
ATE intersectional	OLS	-0.005	0.174
ATE intersectional	OLS with controls	-0.005	0.326
ATE nationalism	OLS	-0.001	0.962
ATE nationalism	OLS with controls	-0.002	1.000
DID feminism familiarity	DID familiarity	0.003	0.091
DID feminism gender	DID feminism gender	0.004	0.461
DID intersectional familiarity	DID familiarity	-0.001	0.091
DID intersectional lgbtq	DID intersectional lgbtq	-0.011	0.359
DID nationalism familiarity	DID familiarity	-0.002	0.077
DID nationalism linked fate	DID nationalism linked fate	-0.045	0.053

21.7.1.6 Design diagnosis

Finally, while a design diagnosis is not a necessary component of a preanalysis plan, it can be useful to show readers why a particular design was chosen over others. This diagnosis indicates that the design produces unbiased estimates but is better powered from some inquiries than others (under the above assumptions about effect size, which were our own and not the original authors'). We are well-powered for the average effects, and the power increases when we include covariate controls. The design is probably too small for most of the heterogeneous effect analyses, which is a point directly conceded in the authors' original PAP.

Further reading.

- Casey, Glennister and Miguel (2012) introduces PAPs in Economics.
- Olken (2015) is a prominent skeptical take on PAPs.
- Green and Lin (2016) offers standard operating procedures as a way to plan for eventualities not covered in the PAP.
- Christensen and Miguel (2018) reviews many issues surrounding PAPs
- Coffman and Niederle (2015) argues for emphasizing replication studies over PAPs.
- Humphreys, de la Sierra and van der Windt (2013) introduces PAPs in Political Science.
- Miguel et al. (2014) describes PAP adoption by journals
- Ofosu and Posner (2020) consider the relationship between preregistration and publication
- Rennie (2004) and Zarin and Tse (2008) explore the consequences of pre-registration for medical trials

- Nosek et al. (2015) introduces the TOP guidelines
- Findley et al. (2016) describes results-blind review

396

Planning

21.7

Chapter 22

Realization

Realization, the implementation of a study, starts from the design declaration. Implementing the data strategy means sampling the units as planned, allocating treatments according to the randomization procedure, and executing the measurement protocol. Implementing the answer strategy means applying the planned summary functions to the realized data. Of course, implementation is much easier said than done. Inevitably, some portion of the design fails to go according to plan: subjects do not comply with treatments, others cannot be located to answer survey questions, or governments interfere with the study as a whole. Sometimes, the answer strategies are discovered to be biased or imprecise or otherwise wanting. Declared designs can be adapted as the study changes, both to help make choices when you need to pivot and so that at the end there is a “realized design” to compare to the “planned design.”

When implementation is complete, the design preregistered in an analysis plan can be “populated” to report on analyses as planned and the realized design reconciled with the planned design. In writing up the study, the design forms the center: why we should believe the answers that we report on. The declared design can be used in the writeup to convince reviewers of the study’s quality, and also a tool to assess the impact of reviewer suggestions on the design.

22.1 Pivoting

When something goes wrong or you learn things work differently from how you expect, you need to pivot. You face two decisions: go/no-go, and if go, should you alter your research design to account for the new reality. Redesigning the study and diagnosing the possible new designs can help you make these decisions. Your design declaration is a living document that you can keep updated and use as a tool to guide you along the research path, not just as a document to write at the beginning of the study and revisit when you are writing up. Keep-

ing the declaration updated as you make the changes along the way will make it easier to reconcile the planned design with the implemented design.

We illustrate two real choices we made, one in which we abandoned the study and one in which we changed the design radically. We link to design declarations and diagnoses of the studies pre- and post-pivot.

One of us was involved with a get-out-the-vote canvassing-experiment-gone-wrong during the 2014 Senate race in New Hampshire. We randomly assigned 4,230 of 8,530 subjects to treatment. However, approximately two weeks before the election, canvassers had only attempted 746 subjects (17.6% of the treatment group) and delivered treatment to just 152 subjects (3.6%). In essence, the implementer was overly optimistic about the number of subjects they would be able to contact in time for the election. Upon reflection, the organization estimated that they would only be able to attempt to contact 900 more voters and believed that their efforts would be best spent on voters with above-median vote propensities.

We faced a choice: should we spend (1) the remaining organizational capacity on treating 900 of the 3,484 remaining unattempted treatment group subjects or should we (2) conduct a new random assignment among above-median propensity voters only? The inquiry for both designs is a complier average causal effect (CACE), but who is classified as a never-taker or a complier differs across the two designs. The organization successfully contacts approximately 20% of those it attempts to contact. In the first design, those who are never even attempted are nevertakers (through no fault of their own!), and further deflate the intention-to-treat effect. We can't just drop them from design 1, because we don't know which units in the control group wouldn't have been attempted, had they been in the control group. In design 2, we conduct a brand-new assignment and the treatment group is only as large as the organization thinks it can handle. A design diagnosis reveals a clear course of action. Even though it decreases the overall sample size, restricting the study to the above-median propensity voters substantially increases the precision of the design. This conclusion follows the logic of the placebo-controlled design described in Section 17.7. Our goal is to restrict the experimental sample to compliers only.

Another of us faced another kind of noncompliance problem in a study in Nigeria: failure to deliver the correct treatment. We launched a cluster-randomized placebo-controlled 2x2 factorial trial of a film treatment and a text message blast treatment. A few days after treatment delivery began, we noticed that the number of replies was extremely similar in treatment and placebo communities, counter to our expectation. We discovered that our research partner, the cell phone company, delivered the treatment message to all communities, so placebo communities received the wrong treatment. But by that time, treatments had been delivered to 106 communities (about half the sample).

We faced the choice to abandon the study or pivot and adapt the study. We quickly agreed that we could not continue research in the 106 communities, be-

cause they had received at least partial treatment. We were left with 109 from our original sample of 200 plus 15 alternates that were selected in the same random sampling process. We determined we could not retain all four treatment conditions and the pure control. We decided that at most we could have two conditions, with about 50 units in each. But which ones? We were reticent to lose the text message or the film treatments, as both tested two distinct theoretical mechanisms for how to encourage prosocial behaviors. We decided to drop the pure control group, the fifth condition, as well as the placebo text message condition. In this way, we could learn about the effect of the film (compared to placebo) and about the effect of the text messages (compared to none).¹

22.2 Populated preanalysis plan

A preanalysis plan describes how study data will eventually be analyzed, but those plans may change in the during the process of producing a finished report, article, or book. Inevitably, authors of pre-analysis plans fail to anticipate how the data generated by the study will eventually be analyzed. Some reasons for discrepancies were discussed in the previous section on pivoting, but others intervene as well. A common reason is that PAPs promise too many analyses. In writing a concise paper, some analyses are dropped, others are combined, and still others are added during the writing and revision process. In the next section, we'll describe how to reconcile analyses-as-planned with analyses-as-implemented, but this present section is about what to do with your analysis plan immediately after getting the data back.

We echo proposals made in Banerjee et al. (2020) and Alrababa'h et al. (2020) that researchers should produce short reports that fulfill the promises made in their PAPs. Banerjee et al. (2020) emphasize that writing PAPs is difficult and usually time-constrained, so it is natural that the final paper will reflect further thinking about the full set of empirical approaches. A “populated PAP” serves to communicate the results of the promised analyses. Alrababa'h et al. (2020) cite the tendency of researchers to abandon the publication of studies that return null results. To address the resulting publication bias, they recommend “null results reports” that share the results of the pre-registered analyses.

We recommended in Section 21.7 that authors include mock analyses in their PAPs using simulated data. Doing so has the significant benefit of being specific about the details of the answer strategy. A further benefit comes when it is time to produce a populated PAP, since the realized data can quite straightforwardly be swapped in for the mock data. Given the time invested in building simulated analyses for the PAP, writing up a populated PAP takes only as much effort as is needed to clean the data (which will need to be done in any case).

¹We randomized half of the communities to receive the treatment film and half the placebo. We then used an over-time stepped-wedge design to study the effect of the text message, randomizing how many days after the film was distributed the text message was sent.

22.2.1 Example

In Section 21, we declared the design for Bonilla and Tillery (2020) following their preanalysis plan. In doing so, we declared an answer strategy in code. In our populated PAP, we can run that same answer strategy code, but swap out the simulated data for the real data collected during the study. We present first the regression table and then the coefficient plot in Figure 22.1.

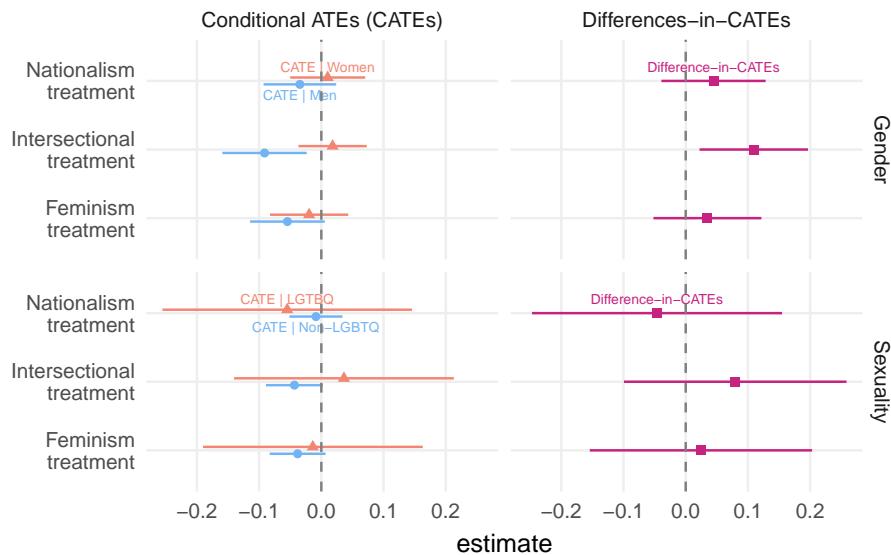


Figure 22.1: Coefficient plot from Bonilla and Tillery design based on the study's realized data.

22.3 Reconciliation

Research design as implemented will differ in some way from research designs as planned. Treatments cannot be executed as conceived, some people cannot be found to interview, and sometimes what we learn from baseline measures informs how we measure later. Understanding how your research design changed from conception to implementation is crucial to understanding what was learned from the study.

Suppose the original design described a three-arm trial: one control and two treatments, but the design as implemented drops all subjects assigned to the second treatment. Sometimes, this is an entirely appropriate and reasonable design modification. Perhaps the second treatment was simply not delivered due to an implementation failure. Other times, these modifications are less benign. Perhaps the second treatment effect estimate did not achieve statistical

	Model 1	Model 2	Model 3	Model 4
(Intercept)	0.84 (0.02)	0.41 (0.04)	0.61 (0.06)	0.54 (0.07)
Znationalism	0.01 (0.02)	0.00 (0.02)	0.02 (0.08)	0.09 (0.09)
Zfeminism	0.04 (0.02)	0.01 (0.02)	0.01 (0.08)	0.02 (0.09)
Zintersectional	0.04 (0.02)	0.03 (0.02)	0.08 (0.08)	0.01 (0.10)
female		0.03 (0.01)		
lgbtq		0.02 (0.02)		
age		0.00 (0.03)		
religiosity		0.01 (0.02)		
income		0.00 (0.03)		
college		0.02 (0.02)		
linked_fate		0.27 (0.03)	0.30 (0.07)	
blm_familiarity		0.07 (0.01)		0.10 (0.02)
Znationalism:linked_fate			0.04 (0.09)	
Zfeminism:linked_fate			0.05 (0.09)	
Zintersectional:linked_fate			0.05 (0.09)	
Znationalism:blm_familiarity				0.03 (0.03)
Zfeminism:blm_familiarity				0.01 (0.03)
Zintersectional:blm_familiarity				0.01 (0.03)
R ²	0.00	0.20	0.14	0.09
Adj. R ²	0.00	0.19	0.13	0.09
Num. obs.	849	849	849	849
RMSE	0.23	0.20	0.21	0.22

p < 0.001; *p* < 0.01; *p* < 0.05

Table 22.1: Statistical models

significance, so the author omitted it from the analysis.

For this reason, we recommend that authors reconcile the design as planned with the design as implemented. A reconciliation can be a plain description of the deviations from the PAP, with justifications where appropriate. A more involved reconciliation would include a declaration of the planned design, a declaration of the implemented design, and a list of the differences. This “diff” of the designs can be automated through the declaration of both designs in computer code, then comparing the two design objects line-by-line (see the function `compare_designs()` in `DeclareDesign`).

In some cases, reconciliation will lead to additional learning beyond what can be inferred from the final design itself. When some units refuse to be included in the study sample or some units refuse measurement, we learn that important information about those units. Understanding sample exclusions, noncompliance, and attrition not only may inform future research design planning choices but contribute substantively to our understanding of the social setting.

22.3.1 Example

In Section 21, we described the preanalysis plan registered by Bonilla and Tillery (2020). We reconcile the set of conditional average treatment effect (CATE) analyses planned in that PAP, the analyses reported in the paper, and those reported in the appendix at the request of reviewers in Table 22.2. In column two, we see that the authors planned four CATE estimations: effects by familiarity with Black Lives Matter; by gender; LGBTQ status; and linked fate. Only two of those are reported in the paper; the others may have been excluded for space reasons. Another way to handle these uninteresting results would be to present them in a populated PAP posted on their Web site or in the paper’s appendix.

In their appendix, the authors report on a set of analyses requested by reviewers. We see this as a perfect example of transparently presenting the set of planned analyses and highlighting the analyses that were added afterward and why they were added. They write:

We have been asked to consider other pertinent moderations beyond gender and LGBTQ+ status. They are contained in the four following sections.

This small table describes the heterogeneous effects analyses the researchers planned, those reported in the paper, and those reported in the appendix at the request of reviewers.

Table 22.2: Reconciliation of Bonilla and Tillery preanalysis plan.

Covariate	In the preanalysis plan	In the paper	In the appendix (at the request of reviewers)
Familiarity with BLM	X		
Gender	X	X	
LGBTQ status	X	X	
Linked fate	X		
Religiosity			X
Region			X
Age			X
Education			X

22.4 Writing

When writing up an empirical paper, authors have two sets of goals. First, they want to convince reviewers and readers that the research question they are tackling is important and their research design provides useful answers to that question. Second, they want to influence scholars but also decisionmakers who may make choices about what to believe and what to do on the basis of the study, including policymakers, businesses, and the public.

A common model for social science empirical papers has five sections: introduction, theory, design, results, and discussion. We discuss each in turn.

The introduction section should highlight each aspect of *MIDA* in brief. The reader is brought quickly up to speed on the whole research design, as well as expectations and actual findings.

The theory, evidence review, and hypotheses section is particularly important for the second goal of empirical papers, integration into a research literature and decisionmaking. The theory and hypotheses clarify many elements of the study's model *M* and also its inquiry *I*. The theory and review of past evidence on the same and related inquiries will be used to structure prior beliefs about the question and related questions, and also to identify which part of past scholarship the study's inquiry speaks to. Without explicitly linking the present inquiry to those of past studies, we can't explain to readers how the study updates our understanding over previous work.

With the theoretical relationship of the present inquiry *I* to the inquiries in past work clarified, reviews of the findings of those past studies represent research designs unto themselves. We should try to prevent general research design issues such as selection on the dependent variable by ensuring we discuss all

literature and not only present views consonant with our hypotheses. A meta-analysis or systematic review of past evidence could provide a systematic summary of past answers, or an informal literature review could be offered. In summarizing the literature, the research designs of past studies should be accounted for. Meta-analyses often formally account for the quality of the research designs of past studies by weighting by the inverse of their precision (upweighting more informative studies and down-weighting less informative ones). Literature reviews may do so informally. However, this accounts only for the variance of past studies, not potential biases, which should be accounted for in how studies are selected (filtering out biased research designs).

The research design section should lay out the details of D and A , building on the description of M and I in the preceding section. The section could refer to a design declared in code found in an appendix.

The results and discussion sections report on the realized answer and clarify what inferences from this result can be drawn. These sections implement the answer strategy of the study, but not only in obvious ways. Of course, regression tables and visualizations of the data report on the application of estimation procedures to the realized data. But the text in a discussion section is also part of the answer strategy: it is the application of a strategy for translating numerical and visual results into a qualitative description of the findings. What this translation function is may depend on how the data turn out, which is good and bad. Good in that we should learn as much as we can from our data, and some tests may not be obvious to us before. Bad in that it is hard to imagine what discursive procedure we would use under *alternative* realizations of the data. It may be helpful to write out the interpretations you would give to plausible ways the study could come out.

In the conclusion section, we turn back to the second goal of writing up the paper: influencing future scholarship and decisionmaking. The conclusion section should, formally where possible, integrate the new findings into past findings and leave readers with the authors' view of what is now known to date on the inquiry. Bayesian integration could take the form of updating priors formed based on a meta-analysis of past studies, a likelihood function, and the results from the present study as the new data. Informal integration could follow this strategy qualitatively, assessing what was known and how confident we were and what we learned in this study and how confident we are in the findings. An additional way a conclusion section can be written to influence future scholarship is to provide new research designs — new *MIDAs* — that future scholars can implement. By providing a statement of the posterior beliefs of the study and new research designs that address specific empirical threats to the results, later scholars can move forward in an informed way.

22.4.1 Example

In the Figure below, we annotate Mousa (2020) by highlighting where in the article each design component is discussed. The study reports on the results of a randomized experiment in which Iraqi Christians were assigned either to an all-Christian soccer team or a team in which they would play alongside Muslims. The experiment tested whether being on a mixed team affected intergroup attitudes and behaviors, both among teammates and back at home after the games were over. We highlight in color areas discussing the model M in yellow, the inquiry I in green, the data strategy D in blue, and the answer strategy A in pink.

The model and the inquiry largely appear in the abstract and introductory portion of the paper, though aspects of the model are discussed later on. Much of the first three pages are devoted to the data strategy, while the answer strategy only appears briefly. This division makes sense: in this paper, the action is all in the experimental design whereas the answer strategy follows straightforwardly from it. The paper mostly describes M and D , with only a small amount of text devoted to I and A . Finally, it is notable that the data strategy is interspersed with aspects of the model. The reason is that the author is justifying choices about randomization and measurement using features of the model.

22.5 Publication

Publication in a peer-reviewed journal is a major goal of many (but not all) research projects. The advice we gave in the previous section on writing papers was to build the case for your findings by grounding your conclusions in the specifics of the research design. By detailing M , I , D , and A in ways that leave little room for confusion or ambiguity, you greatly improve the chances that reviewers and, later, readers will understand your paper.

Ideally, studies would be selected for publication on the basis of design rather than on the basis of results. This ideal can be hard to achieve. Reviewers and editors must decide whether to devote scarce journal space and editing bandwidth to publishing a paper. Criteria may include the topic fit with the journal, the importance of the question, and how much was learned from the research. The publication filter problem – publishing only studies with statistically-significant or splashy results – has long been recognized as a cause both of “false” findings making their way into the literature and publication bias due to missing null findings.

One major barrier to fixing this problem is that design quality is hard to convey to reviewers, so they substitute their own judgments of design based on the results. When the estimate turns out to be statistically significant, reviewers infer that the design must have been well-enough-powered to discover a statistically-significant result. The trouble with this approach is that a significant result might come from a study with 80% power or it might be a lucky draw from a

describe *M*, *I*, *D*, and *A* in enough detail that reviewers can understand the empirical thrust of the study. If in addition to this information, authors provide diagnostic information about the ability of the study to generate credible inferences, we may be to induce reviewers to evaluate studies on the basis of design, not results.

A study's research design is not set in stone until the final version is posted on a journal Web site or published in print. Until then, journal editors and reviewers may ask for design changes that would, in their view, improve the paper. If the changes improve the design, then adopting their suggestions is easy. Some changes are irrelevant to the design – like reporting the reviewer's preferred descriptive statistics – so why not comply. The trouble comes when reviewers propose changes that actively undermine the design. When this happens, diagnosing the reviewer's alternative design can effectively demonstrate that the proposed changes would harm the research design. The design context also helps editors, who have to resolve the dispute one way or the other.

408

Realization

22.5

Chapter 23

Integration

After publication, research studies leave the hands of their authors and enter the public domain.

Most immediately, authors share their findings with the public through the media and with decisionmakers. Design information is useful for helping journalists to emphasize design quality rather than splashy findings. Decisionmakers may act on evidence from studies, and researchers who want to influence policymaking and business decisions may wish to consider diagnosands about the decisions these actors make.

Researchers can prepare for the integration of their studies into scholarly debates through better archiving practices and better reporting of research designs in the published article. Future researchers may build on the results of a past study in three ways. First, they may *reanalyze* the original data. Reanalysts must be cognizant of the original data strategy D when working with the realized data d . Changes to the answer strategy A must respect D , regardless of whether the purpose of the reanalysis is to answer the original inquiry I or to answer a different inquiry I' . Second, future researchers may *replicate* the design. Typically, replicators provide a new answer to the same I with new data, possibly improving elements of D and A along the way. If the inquiry of the replication is too different from the inquiry of the original study, the fidelity of the replication study may be compromised. Lastly, future researchers may *meta-analyze* study's answer with other past studies. Meta-analysis is most meaningful when all of the included studies target a similar enough inquiry and when all studies rely on credible design. Otherwise, the procedure produces a meta-analytic average that is difficult to interpret.

All three of these activities depend on an accurate understanding of the study design. Reanalysts, replicators, and meta-analysts all need access to the study data and materials, of course. They also need to be sure of the critical design information in M , I , D , and A . Later in this section, we outline how archiving

procedures that preserve study data and study design can enable new scientific purposes and describe strategies for doing each of these three particular integration tasks.

23.1 Communicating

The findings from studies are communicated to other scholars through academic publications. But some of the most important audiences – policymakers, businesses, journalists, and the public at large – do not read academic journals. These audiences learn about the study in other in other ways. Authors write opeds, blog posts, and policy reports that translate research for nonspecialist audiences. Press offices pitch research studies for coverage by the media. Researchers present findings directly to decisionmakers and to their research partners.

These new outputs are for different audiences, so they are necessarily diverse in their tone and approach. Some things don't change: we still need to communicate the quality of the research design and what we learn from the study. But some things do: we need to translate specialist language about the substance of the study to a nonspecialist audience, and translate the features of the research design in a way that nonspecialists can understand.

Too often, a casualty of translating the study from academic to other audiences is the design information. When researchers write for popular blogs or give interviews, emphasis is placed on the study results, not on the reasons why the results of the study are to be believed. In sharing the research for nonspecialist audiences, we revert to saying *that* the findings are true and not *why we know* the findings are true. Explaining why we know requires explaining the research design, which in our view ought to be part of any public-facing communication about research.

Of course, even when authors do emphasize design, journalists do not always care. Science reporting is commonly criticized for ignoring study design when picking which studies to publicize, so weak studies are not appropriately filtered out of coverage. Furthermore, journalists emphasize results they believe will drive people to pick up a newspaper or click on a headline. Flashy, surprising, or pandering findings receive far more attention than deserved, with the result that boring but correct findings are crowded out of the media spotlight.

In a review we conducted of recent studies published in *The New York Times* Well section on health and fitness, we found that two dimensions of design quality were commonly ignored. First, experimental studies on new fitness regimens with tiny samples, sometimes fewer than 10 units, are commonly highlighted. When both academic journals and reporters promote tiny studies, the likely result is that the published record is full of statistical flukes driven by noise, not new discoveries. Second, very large studies that draw observational comparisons between large samples of dieters and non-dieters with millions of

observations receive outsize attention. These designs are prone to bias from confounding, but these concerns are swept under the rug.

This state of affairs is not entirely or even mostly the journalists' fault, since, in the absence of design information, it can be challenging to separate the weak designs from the strong ones. Statistical significance and the stamp of approval from peer review are too-easy heuristics to follow.

How can we improve this scientific communication dilemma? The market incentives for both journalists and authors reward flash over substance, and any real solution to the problem would require addressing those incentives. Short of that, we recommend that authors who wish to communicate the high quality of their designs to the media do so by providing the design information in *M*, *I*, *D*, and *A* in lay terms. Science communicators should clearly state the research question (*I*) and explain why applying the data and answer strategies is likely to yield a good answer to the question. The actual result is, of course, also important to communicate, but *why* it is a credible answer to the research question is just as important to share (Principle 3.11). Building confidence in scientific results requires building confidence in scientific practice.

Here's an example of how we could cite a (hypothetical) study in a way that conveys at least some design information. "Using a randomized experiment, the researchers (Authors, Year) found that donating to a campaign causes a large increase in the number of subsequent donation requests from other candidates, which is consistent with theories of party behavior that predict intra-party cooperation.".

Citations can't convey the entirety of *MIDA* in one sentence, but they can give an inkling. The citation explains that the data strategy included some kind of randomized experiment (we don't know how many treatment arms or subjects, among other details), and that the answer strategy probably compared the counts of donation requests from any campaign (email requests, or phone, we don't know) among the groups of subjects that were assigned to donate to a particular campaign. The citation mentions the models described in an unspecified area of the scientific literature on party politics, which all predict cooperation like the sharing of donor lists. We can reason that, if the inquiry, "Is the population average treatment effect effect of donating to one campaign on the number of donation requests from other campaigns positive?" were put to each of these theories, they would all respond "Yes." The citation serves as a useful shorthand for the reader of what the claim of the paper is and why they should think it's credible. By contrast, a citation like "The researchers found that party members cooperate (Author, Year)." doesn't communicate any design information at all.

23.2 Decisionmaking

Policymakers, businesses, humanitarian organizations, and individuals make decisions based on social science research. Research designs, however, are often

constructed without considering who will be informed by the evidence and how they will use evidence in decisions. We can optimize our designs for both scientific publication and decisionmaking. The first step is eliciting the inquiries decisionmakers have, and the second is diagnosing how their decisions change depending on the results. How often would the decisionmaker make the right decision, with and without the study? A design that exhibits high statistical power and a high rate of making the right decision will influence not only the scientific literature but also decisions made by the public.

We illustrate this process by declaring an experimental design that compares a status quo policy with an alternative. As the researcher, your inquiry is the average treatment effect, but the policymaker has a subtlety different inquiry. The policymaker would like to know which policy to implement: the status quo or the alternative. Imagine you meet with the policymaker and ask how they would use the evidence you plan to produce. The policymaker says that they would like to switch to the alternative if it is better than the status quo. However, they face a switching cost to adopt the new policy, so for now, they would like to adopt the alternative only if it is at least 0.1 standard deviations better than the status quo.

In your design declaration, you add two new components to assess your design's probability of the policymaker making the right decision. The first is you add a new inquiry, which is, is the treatment at least 0.1 standard deviations better than the control condition? In addition, you add a statistical test to target this inquiry, which tests whether the treatment effect is larger than 0.1. It does so by testing the null hypothesis that $0.1 = 0$.

Declaration 23.1.

```
# compare status quo to a new proposed policy,
# given cost of switching
N <- 100
effect_size <- 0.1

design <-
  declare_model(N = N,
                U = rnorm(N),
                potential_outcomes(Y ~ effect_size * Z + U)) +
  declare_inquiry(
    ATE = mean(Y_Z_1 - Y_Z_0),
    alternative_better_than_sq = if_else(ATE > 0.1, TRUE, FALSE)
  ) +
  declare_assignment(Z = complete_ra(N)) +
  declare_measurement(Y = reveal_outcomes(Y ~ Z)) +
  declare_estimator(Y ~ Z,
```

```

model = difference_in_means,
inquiry = "ATE",
label = "dim") +
declare_estimator(Y ~ Z,
                   model = lh_robust,
                   linear_hypothesis = "Z - 0.05 = 0",
                   label = "decision")

```

In addition to power, we set up a diagnosand for the proportion of times the policymaker will make the right decision given the evidence you provide. We redesign to consider alternative sample sizes and we diagnose under different possible true effect sizes, some negative (in which case the policymaker should retain the status quo); no difference (status quo should be retained because the effect is not a big enough improvement to justify switching costs); and positive with different sizes.

In Figure 23.1, we show the probability of retaining the status quo policy (left facet) and the probability of switching to the treatment (right) by different true effect sizes. On the left, we see that there is a very high probability of selecting the right policy when the true effect size is very low. This pattern occurs because when the effect size is low, we are likely to fail to reject the null. With small sample sizes, we are likely to also select the status quo even when we should not, because of the imprecision of the estimates. Looking at the right graph, even when the true effect size is large (i.e., 0.25), we need to have a large sample size, about 1500, to achieve 80% probability of correctly choosing the treatment.

The sample size we might choose based on this analysis of the policymaker's choice is different than if we only considered statistical power. This is because the decision curve only reaches 80% power at 1500, while power reaches 80% at just over 500. The reason for the divergence is to make a correct decision, we need evidence that the treatment effect is greater than 0.1, as compared to statistical power which considers whether the effect is greater than 0.0.

23.3 Archiving

One of the biggest successes in the push for greater research transparency has been changing norms surrounding data sharing and analysis code after studies have been published. It has become *de rigueur* at many journals to post these materials at publicly-available repositories like the OSF or Dataverse. This development is undoubtedly a good thing. In older manuscripts, sometimes data or analyses are described as being “available upon request,” but of

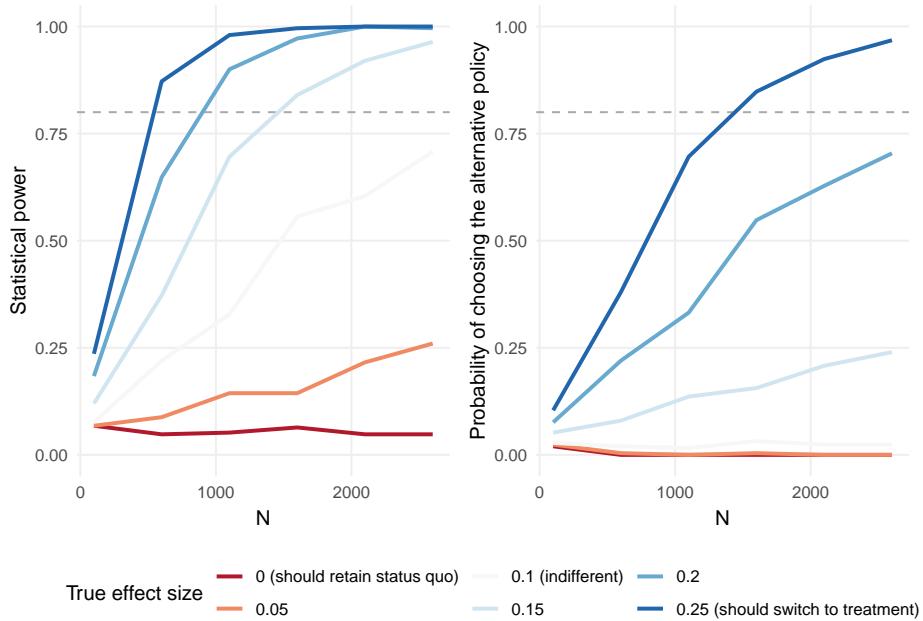


Figure 23.1: WIP: Research design diagnosis for study of effectiveness of a policy change compared to the status quo, where a policymaker wishes to switch to the treatment policy only if it is at least 0.05 standard deviations better than the status quo. On the left, we display the statistical power of the study to detect an effect in either direction. On the right, we display the rate of making the right decision to switch policies or not.

course, such requests are sometimes ignored. Furthermore, a century from now, study authors will no longer be with us even if they wanted to respond to such requests. Public repositories have a much better chance of preserving study information for the future.

What belongs in a replication archive? Enough documentation, data, and design detail that those who wish to reanalyze, replicate, and meta-analyze results can do so without contacting the authors.

Data. First, the realized data itself. Sometimes this is the raw data. Sometimes it is only the “cleaned” data that is actually used by analysis scripts. Where ethically possible, we think it is preferable to post as much of the raw data as possible after removing information like IP addresses and geographic locations that could be used to identify subjects. The output of cleaning scripts – the cleaned data – should also be included in the replication archive.

Reanalyses often reexamine and extend studies by exploring the use of alternative outcomes, varying sets of control variables, and new ways of grouping data. As a result, replication data ideally includes all data collected by the au-

thors even if the variables are not used in the final published results. Often authors exclude these to preserve their own ability to publish on these other variables or because they are worried alternative analyses will cast doubt on their results. We hope norms will change such that study authors instead want to enable future researchers to build on their research by being expansive in what information is included.

Analysis code. Replication archives also include the answer strategy A , or the set of functions that produce results when applied to the data. We need the actual analysis code because the natural-language descriptions of A that are typically given in written reports are imprecise. As a small example, many articles describe their answer strategies as “ordinary least squares” but do not fully describe the set of covariates included or the particular approach to variance estimation. These choices can substantively affect the quality of the research design – and nothing makes these choices explicit like the actual analysis code. Analysis code is needed not only for reanalysis but also replication and meta-analysis. Replication practice today involves inferring most of these details from descriptions in text. Reanalyses may directly reuse or modify analysis code and replication projects need to know the exact details of analyses to ensure they can implement the same analyses on the data they collect. Meta-analysis authors may take the estimates from the past studies directly, so understanding the exact analysis procedure conducted is important. Other times, meta-analyses reanalyze data to ensure comparability in estimation. Conducting analyses with and without covariates, with clustering when it was appropriate, or with a single statistical model when they vary across studies all require having the exact analysis code.

Data strategy materials. Increasingly, replication archives include the materials needed to implement treatments and measurement strategies. Without the survey questionnaires in their original languages and formats, we cannot exactly replicate them in future studies, which hinders our ability to build on and adapt them. The treatment stimuli used in the study should also be included. Data strategies are needed for reanalyses and meta-analyses too: answer strategies should respect data strategies, so understanding the details of sampling, treatment assignment, and measurement can shape reanalysts’ decisions and meta-analysis authors’ decisions about what studies to include and which estimates to synthesize.

Design declaration. While typical replication archives include the data and code, we think that future replication archives should also have a design declaration that fully describes M , I , D , and A . This should be done in code and words. A diagnosis can also be included, demonstrating the properties as understood by the author and indicating the diagnosands that the author considered in judging the quality of the design.

Design details help future scholars not only assess, but replicate, reanalyze, and extend the study. Reanalysts need to understand the answer strategy to modify or extend it and the data strategy used to ensure that their new analysis

respects the details of the sampling, treatment assignment, and measurement procedures. Data and analysis sharing enables reanalysts to adopt or adapt the analysis strategy, but a declaration of the data strategy would help more. The same is true of meta-analysis authors, who need to understand the designs' details to make good decisions about which studies to include and how to analyze them. Replicators who wish to exactly replicate or even just provide an answer to the same inquiry need to understand the inquiry, data strategy, and answer strategy.

The result is disputes that result after the replication is sent out for peer review. The original authors may disagree with inferences the replicators made about the inquiry or data strategy or answer strategy. To protect the original authors and the replicators, including a research design declaration specifying each of these elements resolves these issues so that replication and extension can focus on the substance of the research question and innovation in research design.

Figure 23.2 below shows the file structure for an example replication. Our view on replication archives shares much in common with the TIER protocol. It includes raw data in a platform-independent format (.csv) and cleaned data in a language-specific format (.rds, a format for R data files). Data features like labels, attributes, and factor levels are preserved when imported by the analysis scripts. The analysis scripts are labeled by the outputs they create, such as figures and tables. A master script is included that runs the cleaning and analysis scripts in the correct order. The documents folder consists of the paper, the supplemental appendix, the pre-analysis plan, the populated analysis plan, and codebooks that describe the data. A README file explains each part of the replication archive. We also suggest that authors include a script that consists of a design declaration and diagnosis.

Further Reading

- Peer, Orr and Coppock (2021) propose that researchers should “actively maintain” their replication archives by checking that they still run and making updates to obsolete code. In this way, the information about *A* that is contained in the replication archive stays current and scientifically useful.
- Elman, Kapiszewski and Lupia (2018) argues that the benefits of data transparency in political science outweigh its costs.
- Bowers (2011) describes how good archiving is like collaborating with your future self.
- Alvarez, Key and Núñez (2018) provide guidance on how to create good replication archives.

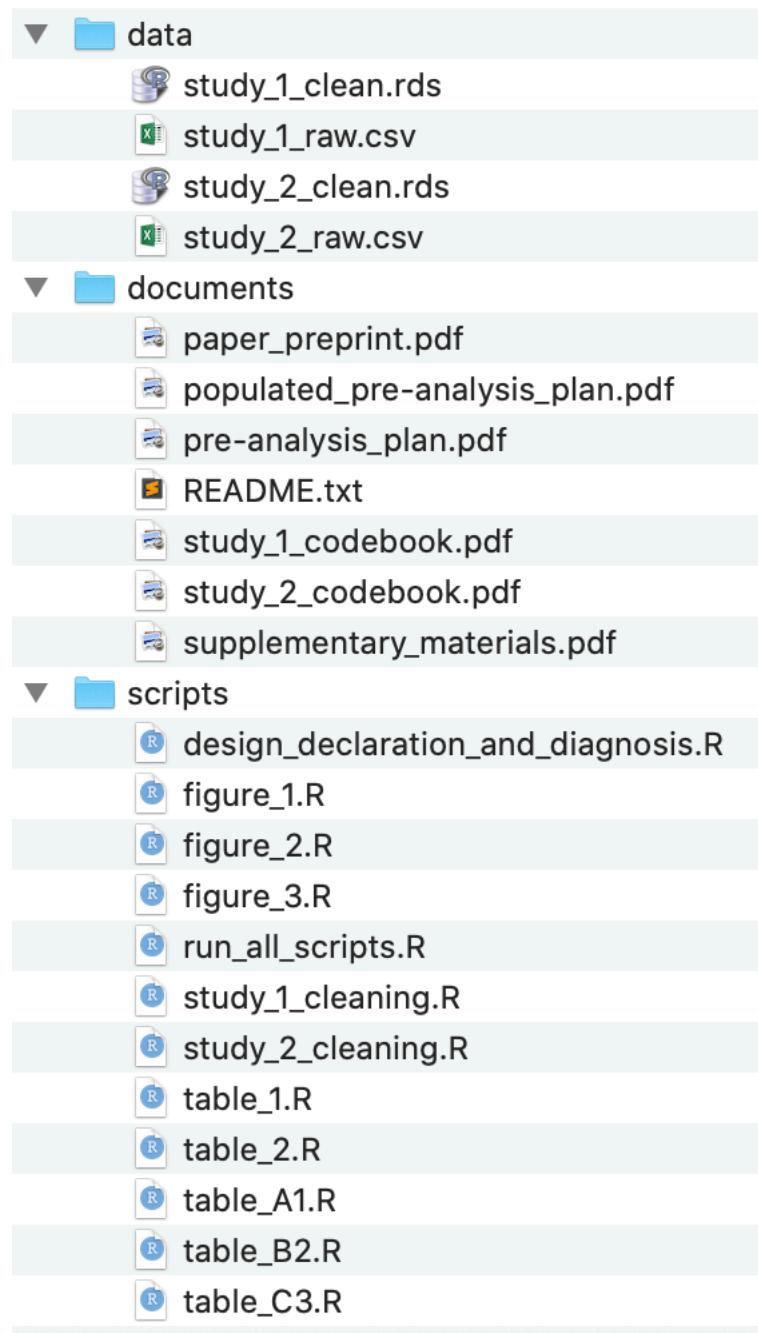


Figure 23.2: File structure for archiving

23.4 Reanalysis

A reanalysis of an existing study is a follow-up study that reuses the original realized data for some new purpose. The reanalysis is a study with a research design that can be described in terms of M , I , D , and A . Reanalyses are fundamentally constrained by the data strategy of the original study. The data strategy D and the resulting data are set in stone – but reanalysts can make changes to the answer strategy A and sometimes also to the model M or inquiry I .

We can learn from reanalyses in several ways. First, we can fix errors in the original answer strategy. Reanalyses fixed simple mathematical errors, typos in data transcription, or failures to account for features of the data strategy when analyzing the data. These reanalyses show whether the original results do or do not depend on these corrections. Second, we can reassess the study in light of new information about the world learned after the original study was published. That is, sometimes M changes in ways that color our interpretation of past results. Perhaps we learned about new confounders or alternative causal channels that undermine the original design's credibility. When reanalyzed, demonstrating the results do (or do not) change when new model features are incorporated improves our understanding of the inquiry. Third, reanalyses may also aim to answer new questions that were not considered by the original study but for which the realized data can provide useful answers.

Lastly, many reanalyses show that original findings are not “robust” to alternative answer strategies. These are better conceptualized as claims about robustness to alternative models: one model may imply one answer strategy, and a different model, with another confounder, suggests another. If both models are plausible, a good answer strategy should be robust to both and even help distinguish between them. A reanalysis could uncover robustness to these alternative models or lack thereof.

Reanalyses are themselves research designs. Just like any design, whether a reanalysis is a strong research design depends on *possible* realizations of the data (as determined by the data strategy), not just the realized data. Because the realized data is fixed in a reanalysis, analysts are often instead tempted to judge the reanalysis based on whether it overturns or confirms the original study's results. A successful reanalysis in this way of thinking demonstrates, by showing that the original results are changed under an alternative answer strategy, that the results are not robust to other plausible models.

This way of thinking can lead to incorrect assessments of reanalyses. We need to consider what answers we would obtain under the original answer strategy A and the reanalysis strategy A' under many *possible* realizations of the data. A good reanalysis strategy reveals with high probability the set of models of the world under which we can make credible claims about the inquiry. Whether or not the results change under the answer strategies A and A' tells us little about this probability because the realized data is only one draw.

23.4.1 Example

In this section, we illustrate the flaw in assessing reanalyses based on changing significance of results alone. We demonstrate how to assess the properties of reanalysis plans, comparing the properties of original answer strategies to proposed reanalysis answer strategies.

The design we consider is an observational study with a binary treatment Z that may or may not be confounded by a covariate X . Suppose that the original researcher had in mind a model in which Z is not confounded by X :

```
# X is not a confounder and is measured pretreatment
model_1 <-
  declare_model(
    N = 100,
    U = rnorm(N),
    X = rnorm(N),
    Z = rbinom(N, 1, prob = plogis(0.5)),
    potential_outcomes(Y ~ 0.1 * Z + 0.25 * X + U),
    Y = reveal_outcomes(Y ~ Z)
  )
```

The reanalyst has in mind a different model. In this second model, X confounds the relationship between Z and Y :

```
# X is a confounder and is measured pretreatment
model_2 <-
  declare_model(
    N = 100,
    U = rnorm(N),
    X = rnorm(N),
    Z = rbinom(N, 1, prob = plogis(0.5 + X)),
    potential_outcomes(Y ~ 0.1 * Z + 0.25 * X + U),
    Y = reveal_outcomes(Y ~ Z)
  )
```

The original answer strategy A is a regression of the outcome Y on the treatment Z . The reanalyst collects the covariate X and proposes to control for it in a linear regression; call that strategy A_prime.

```
A <- declare_estimator(Y ~ Z, model = lm_robust, label = "A")
```

```
A_prime <- declare_estimator(Y ~ Z + X, model = lm_robust, label = "A_prime")
```

Applying the two answer strategies, we get differing results. The treatment effect estimate is significant under A but not under A_{prime} . Commonly, reanalysts would infer from this that the answer strategy A_{prime} is preferred and that the original result was incorrect.

```
draw_estimates(model_2 + A + A_prime)
```

estimator	estimate	std.error	p.value
A	0.385	0.176	0.031
A_{prime}	0.219	0.188	0.246

As we show now, these claims depend on the validity of the model and should be assessed with design diagnosis. Consider a third model in which X is affected by Z and Y . (In many observational settings, which variables are causally prior or posterior to others can be difficult to know with certainty). We now diagnose both answer strategies under all three models.

```
# X is not a confounder and is measured posttreatment
model_3 <-
  declare_model(
    N = 100,
    U = rnorm(N),
    Z = rbinom(N, 1, prob = plogis(0.5)),
    potential_outcomes(Y ~ 0.1 * Z + U),
    Y = reveal_outcomes(Y ~ Z),
    X = 0.1 * Z + 5 * Y + rnorm(N)
  )

I <- declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))

design_1 <- model_1 + I + A + A_prime
design_2 <- model_2 + I + A + A_prime
design_3 <- model_3 + I + A + A_prime
```

What we see in the diagnosis below is that A_{prime} is only preferred if we know for sure that X is measured pretreatment. In design 3, where X is measured posttreatment, A is preferred, because controlling for X leads to posttreatment bias. This diagnosis indicates that the reanalyst needs to justify their beliefs

Table 23.1: Diagnosis of the reanalysis design under alternative models

design	estimator	bias
design_1	A	-0.003
design_1	A_prime	-0.002
design_2	A	0.218
design_2	A_prime	0.013
design_3	A	0.008
design_3	A_prime	-0.116

about the causal ordering of X and Z to claim that A_{prime} is preferred to A . The reanalyst should not conclude on the basis of the realized estimates only that their answer strategy is preferred.

Three principles emerge from the idea that changing A to A' should be justified by diagnosis, not the comparison of the realized results of the two answer strategies.

1. **Home ground dominance.** Holding the original M constant (i.e., the home ground of the original study), if you can show that a new answer strategy A' yields better diagnosands than the original A , then A' can be justified by home ground dominance. In the example above, model 1 is the “home ground,” and the reanalyst’s A' is preferred to A on this home ground.
2. **Robustness to alternative models.** A second justification for a change in answer strategy is that you can show that a new answer strategy is robust to both the original model M and a new, also plausible, M' . In observational studies, we are uncertain about many features of the model, such as the existence of unobserved confounders. In the example above, A' is robust to models 1 and 2 but is not robust to model 3. By contrast, A is robust to models 1 and 3 but not to model 2.
3. **Model plausibility.** If the diagnosands for a design with A' are worse than those with A under M but better under M' , then the switch to A' can only be justified by a claim or demonstration that M' is more plausible than M . As we saw in the example, neither A nor A' was robust to all three alternative models. A claim about model plausibility would have to be invoked to justify controlling for X . Such a claim could be made on the basis of substantive knowledge or additional data. For example, the reanalyst could demonstrate that data collection of X took place before the treatment was realized in order to rule out model 3.

23.5 Replication

After your study is completed, it may one day be replicated. Replication differs from reanalysis in that a replication study involves collecting new data to study the same inquiry. A new model, data strategy, or answer strategy may also be proposed.

So-called “exact” replications hold key features of I , D , and A fixed, but draw a new dataset from the data strategy and apply the same answer strategy A to the new data to produce a fresh answer. Replications are said to “succeed” when the new and old answer are similar and to “fail” when they are not. Dichotomizing replication attempts into successes and failures is usually not that helpful, and it would be better to simply characterize how similar the old and new answers are. Literally exact replication is impossible: at least some elements of M have changed between the first study and the replication. Specifying how they might have changed, e.g., how outcomes vary with time, will help judge differences observed between old and new answers.

Replication studies can benefit enormously from the knowledge gains produced by the original studies. For example, we learn a large amount about the model M and the value of the inquiry from the original study. The M of the replication study can and should incorporate this new information. For example, if we learn from the original study that the estimand is positive, but it might be small, the replication study could respond by changing D to increase the sample size. Design diagnosis can help you learn about how to change the replication study’s design in light of the original research.

When changes to the data strategy D or answer strategy A can be made to produce more informative answers about the same inquiry I , exact replication may not be preferred. Holding the treatment and outcomes the same may be required to provide an answer to the same I , but increasing the sample size or sampling individuals rather than villages or other changes may be preferable to exact replication. Replication designs can also take advantage of new best practices in research design.

So-called “conceptual” replications alter both M and D , but keep I and A as similar as possible. That is, a conceptual replication tries to ascertain whether a relationship in one context also holds in a new context. The trouble and promise of conceptual replications lie in the designer’s success at holding I constant. Too often, a conceptual replication fails because in changing M , too much changes about I , muddying the “concept” under replication.

A summary function is needed to interpret the difference between the original answer and the replication answer. This might take the new one and throw out the old if design was poor in the first. It might be taking the average. It might be a precision-weighted average. Specifying this function *ex ante* may be useful to avoid the choice of summary depending on the replication results. This summary function will be reflected in A and in the discussion section of the

replication paper.

23.5.1 Example

Here we have an original study design of size 1000. The original study design's true sample average treatment effect (SATE) is 0.2 because the original authors happened to study a very treatment-responsive population. We seek to replicate the original results, whatever they may be. We want to characterize the probability of concluding that we "failed" to replicate the original results. We have four alternative metrics for assessing replication failure.

1. Are the original and replication estimates statistically significantly different from each other? If yes, we conclude that we failed to replicate the original results, and if no, we conclude that the study replicated.
2. Is the replication estimate within the original 95% confidence interval?
3. Is the original estimate within the replication 95% confidence interval?
4. Do we fail to affirm equivalence¹ between the replication and original estimate, using a tolerance of 0.2?

Figure 23.3 shows that no matter how big we make the replication, we find that the rate of concluding the difference-in-SATEs is nonzero only occurs about 10% of the time. Similarly, the replication estimate is rarely outside of the original confidence interval, because it's rare to be more extreme than a wide confidence interval. The relatively high variance of the original study means that it is so uncertain, it's tough to distinguish it from any number in particular.

If we turn to the third metric, we become more and more likely to conclude that the study fails to replicate as the replication study grows. At very large sample sizes, the replication confidence intervals become extremely small, so in the limit, it will always exclude the original study estimate.

The last metric, equivalence testing, has the nice property that as the sample size grows, we get closer to the correct answer – the true SATEs are indeed within 0.2 standard units of each other. However, again because the original study is so noisy, it is difficult to affirm its equivalence with anything, even when the replication study is quite large.

The upshot of this exercise is that, curiously, when original studies are weak (in that they generate imprecise estimates), it becomes harder to conclusively affirm that they did not replicate. This set of incentives is somewhat perverse: designers of original studies benefit from a lack of precision if it means they can't "fail to replicate."

¹For an introduction to equivalence testing see Hartman and Hidalgo (2018)

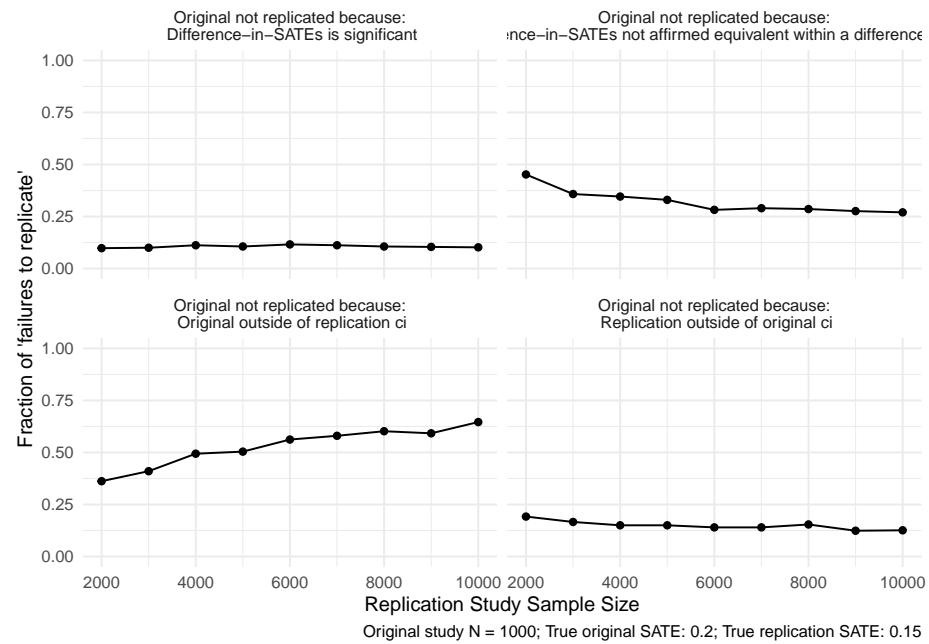


Figure 23.3: Rates of 'Failure to Replicate' according to four diagnosands

23.6 Meta-analysis

One of the last stages of the lifecycle of a research design is its eventual incorporation in to our common scientific understanding of the world. Research findings are synthesized into our broader scientific understanding through systematic reviews and meta-analysis. In this section, we describe how a meta-analysis project itself comprises a new research design, whose properties we can investigate through declaration and diagnosis.

Research synthesis takes two basic forms. The first is meta-analysis, in which a series of estimates are analyzed together in order to better understand features of the distribution of answers obtained in the literature (see Section 18.4). Studies can be averaged together in ways that are better and worse. Sometimes the answers are averaged together according to their precision. A precision-weighted average gives more weight to precise estimates and less weight to studies that are noisy. Sometimes studies are “averaged” by counting up how many of the estimates are positive and significant, how many are negative and significant, and how many are null. This is the typical approach taken in a literature review. Regardless of the averaging approach, the goal of this kind of synthesis is to learn as much as possible about a particular inquiry I by drawing on evidence from many studies.

A second kind of synthesis is an attempt to bring together the results many

designs, each of which targets a different inquiry about a common model. This is the kind of synthesis that takes place across an entire research literature. Different scholars focus on different nodes and edges of the common model, so a synthesis needs to incorporate the diverse sources of evidence.

How can you best anticipate how your research findings will be synthesized? For the first kind of synthesis – meta-analysis – you must be cognizant of keeping a commonly understood I in mind. You want to select inquiries not for their novelty, but because of their commonly-understood importance. We want *many* studies on the effects of having women versus men elected officials on public goods because we want to understand this particular I in great detail and specificity. While the specifics of the model M might differ from study to study, the fact that the I s are all similar enough to be synthesized allows for a specific kind of knowledge accumulation.

For the second kind of synthesis – literature-wide progress on a full causal model – even greater care is required. Specific studies cannot make up bespoke models M but instead must understand how the specific M adopted in the study is a special case of some broader M that is in principle agreed to by a wider research community. The nonstop, neverending proliferation of study-specific theories is a threat to this kind of knowledge accumulation. In a telling piece, McPhetres et al. (2020) document that in a decade of research articles published in *Psychological Science*, 359 specific theories were named, 70% were named only once and a further 12% were named just twice.

Since either kind of synthesis is a research design of its own, declaring it and diagnosing its properties can be informative. The data strategy for any research synthesis includes the process of collecting past studies. Search strategies are sampling strategies, and they can be biased in the same ways as convenience samples of individuals. Conducting a full census of past literature on a topic is usually not possible since not all research is made public, but selecting only published studies may reinforce publication biases. Proactively collecting working papers and soliciting unpublished or abandoned research on a topic are strategies to mitigate these risks. The choice of answer strategy for research synthesis depends on model assumptions about how studies are related. The model for declaring a research synthesis thus might include assumptions not only about how studies reach you as the synthesizer, but also how the contexts and units were selected in those original studies. Three common inquiries for meta-analysis include the average effect across contexts, the extent to which effects vary across contexts, and the best estimate of effects in specific contexts. Diagnosis can help assess the conditions under which your analysis strategies will provide unbiased, efficient estimates of true effects either in a subset of contexts which were studied or about a broader population.

Further Reading

- McPhetres et al. (2020) on the proliferation of theories in psychology
- Samii (2016) on the role of “causal empiricists,” as distinct from the role

of theorists.

Part V

Epilogue

Chapter 24

Epilogue

Social science is undergoing a period of structural change. The credibility of decades of research findings has been questioned and many studies have been found to be flawed.

The first prong of the response to the upheaval was work to identify new research designs that could deliver credible evidence to answer social scientific questions. Randomized experiments rapidly rose in popularity in economics and political science. Observational causal inference methods such as regression discontinuities and difference-in-differences designs have become popular in sociology, political science, and economics. And sample sizes have increased and new populations outside university students have been explored in psychology and experimental economics.

The second prong has focused on communication. Open science practices are motivated by the idea that even when a credible *general* design for drawing inferences is adopted, myriad small design decisions may influence the validity of the results. Sharing the plans, computer code, and materials used to implement the research as well as the data that result allows peer reviewers and readers to assess the large and small decisions the authors made and come to a their own judgment about what was learned. Preregistration of plans before implementing research provides additional clarity: which of these decisions were made before seeing data and results and which were made after.

The two prongs are closely related to each other. Open science practices are meant to reinforce the work on credible designs: transparency of research methods incentivizes researchers to select credible research designs in the first place. Common to both is the idea that research design matters.

Strikingly, however, these advances have been made without a clear common understanding of what a design is or how to evaluate one. In this book we provide a flexible approach to defining a design and a procedure for assessing

its qualities. We identify four generic elements of a research design: the Model, the Inquiry, the Data Strategy and the Answer strategy. “Declaring” these four elements makes it easier to communicate the most important analytic features of a design, enabling “diagnosis” the credibility of claims that depend on them.

We hope our effort adds two new steps to the workflow promoted by open science advocates. First, we want scholars to develop designs by declaring them in code, diagnosing their properties in terms of scientific, logistical, and ethical goals, and redesigning across feasible designs to select the final design. Second, we want scholars to share their designs so they can be more easily understood, more easily interrogated, and more easily built upon.

We see these steps as deeply complementary to the credibility revolution and the open science movement.

Declaring and diagnosing designs can make designs stronger. Many design choices can be made on the basis of analytic results, and these should be used when possible. but oftentimes analytic results provide incomplete answers. Sampling and eligibility procedures can interact with treatment allocation schemes, so causal identification results can be insufficient to assess the unbiasedness of the design for a sample average treatment effect. Moreover, many theoretical results about research design are conditional on certain sample sizes, correlations between variables, or the correctness of functional forms. Assessing how designs perform based on the specific research setting and its sample size and empirical correlations between variables augments the general theoretical guidance. Of course, the theoretical results guide how to set up the design itself: identifying what kinds of problems can emerge in a model is an exercise shaped by theoretical results.

Sharing research designs in code complements common open science practices in use today. By providing the design in code, the study can be replicated exactly in a new setting or a later time period, reanalyzed with the realized data but new estimators, and the diagnosands reassessed on the authors’ original terms and under new conjectures about the model. Declarations also complement current practices in preregistration. Considerable debates surrounds what should be included in a preanalysis plan. Declarations in code provide an answer: you should declare sufficient information to enable to diagnose the design in terms of study-relevant diagnosands.

Better software tools will come along to declare, diagnose, and redesign studies in code. A body of domain-specific knowledge will develop about what models design must be assessed against to assure robustness. What we hope will remain is the idea that research designs can be thought of as interrogable objects, defined by the specific steps in the procedures used to generate data and analyze it to provide answers to specific inquiries that are themselves well defined with respect to specified representations of the world. We hope that these ideas and tools will enable scholars to better respond to changed incentives in social sciences to adopt credible research designs for the questions they are asking and

to communicate that they have done so to reviewers and readers.

Part VI

References

Bibliography

- Abadie, Alberto, Susan Athey, Guido W. Imbens and Jeffrey Wooldridge. 2017. “When should you adjust standard errors for clustering?”.
- Abell, Peter and Ofer Engel. 2019. “Subjective Causality and Counterfactuals in the Social Sciences: Toward an Ethnographic Causality?” *Sociological Methods & Research* p. 0049124119852373.
- Alrababa'h, Ala', Scott Williamson, Andrea Dillon, Jens Hainmueller, Dominik Hangartner, Michael Hotard, David Laitin, Duncan Lawrence and Jeremy Weinstein. 2020. “Learning from Null Effects: A Bottom-Up Approach.”. [URL: osf.io/preprints/socarxiv/5epy](https://osf.io/preprints/socarxiv/5epy)
- Alvarez, R. Michael, Ellen M. Key and Lucas Núñez. 2018. “Research Replication: Practical Considerations.” *PS: Political Science & Politics* 51(2):422–426.
- Angrist, Joshua D. and Jörn-Steffen Pischke. 2008. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton: Princeton University Press.
- Aronow, Peter M. and Benjamin T. Miller. 2019. *Foundations of Agnostic Statistics*. Cambridge University Press.
- Aronow, Peter M. and Cyrus Samii. 2017. “Estimating average causal effects under general interference, with application to a social network experiment.” *The Annals of Applied Statistics* 11(4):1912–1947.
- Aronow, Peter M., Donald P. Green and Donald K. K. Lee. 2014. “Sharp Bounds On The Variance In Randomized Experiments.” *The Annals of Statistics* 42(3):850–871.
- Aronow, Peter M., Jonathon Baron and Lauren Pinson. 2019. “A note on dropping experimental subjects who fail a manipulation check.” *Political Analysis* 27(4):572–589.
- Baird, Sarah, J. Aislinn Bohren, Craig McIntosh and Berk Ozler. 2018. “Optimal Design of Experiments in the Presence of Interference.” *Review of Economics & Statistics* 5(100):844–860.
- Banerjee, Abhijit, Esther Duflo, Amy Finkelstein, Lawrence F Katz, Benjamin A Olken and Anja Sautmann. 2020. In Praise of Moderation: Suggestions for the

- Scope and Use of Pre-Analysis Plans for RCTs in Economics. Working Paper 26993 National Bureau of Economic Research.
URL: <http://www.nber.org/papers/w26993>
- Bansak, Kirk, Jens Hainmueller, Daniel J. Hopkins and Teppei Yamamoto. 2019. "Beyond the breaking point? Survey satisficing in conjoint experiments." *Political Science Research and Methods* pp. 1–19.
- Baron, Hannah and Lauren Young. 2020. "From principles to practice: Methods for increasing the transparency of research ethics in violent contexts.". Working paper.
- Baumgartner, Michael and Alrik Thiem. 2017. "Often Trusted but Never (Properly) Tested: Evaluating Qualitative Comparative Analysis." *Sociological Methods & Research* . Forthcoming.
- Benjamin, Daniel J., James O. Berger, Magnus Johannesson, Brian A. Nosek, E. J. Wagenmakers, Richard Berk, Kenneth A. Bollen, Björn Brembs, Lawrence Brown, Colin Camerer, David Cesarini, Christopher D. Chambers, Merlise Clyde, Thomas D. Cook, Paul De Boeck, Zoltan Dienes, Anna Dreber, Kenny Easwaran, Charles Efferson, Ernst Fehr, Fiona Fidler, Andy P. Field, Malcolm Forster, Edward I. George, Richard Gonzalez, Steven Goodman, Edwin Green, Donald P. Green, Anthony G. Greenwald, Jarrod D. Hadfield, Larry V. Hedges, Leonhard Held, Teck Hua Ho, Herbert Hoijtink, Daniel J. Hruschka, Kosuke Imai, Guido Imbens, John P. A. Ioannidis, Minjeong Jeon, James Holland Jones, Michael Kirchler, David Laibson, John List, Roderick Little, Arthur Lupia, Edouard Machery, Scott E. Maxwell, Michael McCarthy, Don A. Moore, Stephen L. Morgan, Marcus Munafó, Shinichi Nakagawa, Brendan Nyhan, Timothy H. Parker, Luis Pericchi, Marco Perugini, Jeff Rouder, Judith Rousseau, Victoria Savalei, Felix D. Schönbrodt, Thomas Sellke, Betsy Sinclair, Dustin Tingley, Trisha Van Zandt, Simine Vazire, Duncan J. Watts, Christopher Winship, Robert L. Wolpert, Yu Xie, Cristobal Young, Jonathan Zinman and Valen E. Johnson. 2018. "Redefine statistical significance." *Nature Human Behaviour* 2(1):6–10.
- Bennett, Andrew. 2015. Appendix. In *Process Tracing: From Metaphor to Analytic Tool*, ed. Andrew Bennett and Jeffrey Checkel. New York: Cambridge University Press.
- Bennett, Andrew and Jeffrey Checkel. 2015. Process Tracing: From Philosophical Roots to Best Practices. In *Process Tracing: From Metaphor to Analytic Tool*, ed. Andrew Bennett and Jeffrey Checkel. New York: Cambridge University Press pp. 3–37.
- Blair, Graeme, Alexander Coppock and Margaret Moor. 2020. "When to Worry About Sensitivity Bias: A Social Reference Theory and Evidence from 30 Years of List Experiments." *American Political Science Review* 114(4):1297–1315.
- Blair, Graeme, Jasper Cooper, Alexander Coppock and Macartan Humphreys.

2020. "Declaring and Diagnosing Research Designs." *American Political Science Review* .
- Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. 2015. "Design and analysis of the randomized response technique." *Journal of the American Statistical Association* 110(511):1304–1319.
- Bonilla, Tabitha and Alvin B. Tillery. 2020. "Which Identity Frames Boost Support for and Mobilization in the #BlackLivesMatter Movement? An Experimental Test." *American Political Science Review* 114(4):947–962.
- Bowers, Jake. 2011. "Six steps to a better relationship with your future self." .
URL: https://cpb-us-e1.wpmucdn.com/blogs.rice.edu/dist/d/2418/files/2013/09/tpm_v18_n2.pdf
- Brady, Henry E. 2004. "Data-set observations versus causal-process observations: The 2000 US presidential election." *Rethinking social inquiry: Diverse tools, shared standards* pp. 267–272.
- Butler, Daniel M. and Charles Crabtree. 2017. "Moving Beyond Measurement: Adapting Audit Studies to Test Bias-Reducing Interventions." *Journal of Experimental Political Science* 4(1):57–67.
- Calonico, Sebastian, Matias D Cattaneo and Rocio Titiunik. 2014. "Robust non-parametric confidence intervals for regression-discontinuity designs." *Econometrica* 82(6):2295–2326.
- 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.
- Christensen, Garret and Edward Miguel. 2018. "Transparency, Reproducibility, and the Credibility of Economics Research." *Journal of Economic Literature* 56(3):920–80.
URL: <https://www.aeaweb.org/articles?id=10.1257/jel.20171350>
- Clingingsmith, David, Asim Ijaz Khwaja and Michael Kremer. 2009. "Estimating the impact of the Hajj: religion and tolerance in Islam's global gathering." *The Quarterly Journal of Economics* 124(3):1133–1170.
- Coffman, Lucas C. and Muriel Niederle. 2015. "Pre-analysis plans have limited upside, especially where replications are feasible." *The Journal of Economic Perspectives* 29(3):81–97.
- Collier, David, David A Freedman, James D Fearon, David D Laitin, John Gerding and Gary Goertz. 2008. "Symposium: Case Selection, Case Studies, and Causal Inference." *Qualitative & Multimethod Research* 6(2).
- Coppock, Alexander. 2019. "Avoiding Post-Treatment Bias in Audit Experiments." *Journal of Experimental Political Science* 6(1):1–4.
- Coppock, Alexander and Dipin Kaur. 2021. "Qualitative Imputation of Missing Potential Outcomes." *American Journal of Political Science* . Forthcoming.

- Creighton, Mathew J. and Amaney Jamal. 2015. “Does Islam play a role in anti-immigrant sentiment? An experimental approach.” *Social Science Research* 53:89–103.
- Dawid, A. Philip. 2000. “Causal inference without counterfactuals.” *Journal of the American Statistical Association* 95(450):407–424.
- De Chaisemartin, Clément and Xavier d’Haultfoeuille. 2020. “Two-way fixed effects estimators with heterogeneous treatment effects.” *American Economic Review* 110(9):2964–96.
- Deaton, Angus S. 2010. “Instruments, randomization, and learning about development.” *Journal of Economic Literature* 48(2):424–55.
- Druckman, James N and Cindy D Kam. 2011. “Students as experimental participants.” *Cambridge handbook of experimental political science* 1:41–57.
- Dunning, Thad. 2012. *Natural Experiments in the Social Sciences: A Design-Based Approach*. Cambridge: Cambridge University Press.
- Egami, Naoki and Erin Hartman. 2021. “Elements of External Validity: Framework, Design, and Analysis.” *Unpublished manuscript* .
- Elman, Colin, Diana Kapiszewski and Arthur Lupia. 2018. “Transparent Social Inquiry: Implications for Political Science.” *Annual Review of Political Science* 21(1):29–47.
- Fairfield, Tasha and Andrew E Charman. 2017. “Explicit Bayesian analysis for process tracing: Guidelines, opportunities, and caveats.” *Political Analysis* 25(3):363–380.
- Fang, Albert H, Andrew M Guess and Macartan Humphreys. 2019. “Can the government deter discrimination? Evidence from a randomized intervention in New York City.” *The Journal of Politics* 81(1):127–141.
- Fearon, James D and David D Laitin. 2008. Integrating qualitative and quantitative methods. In *The Oxford Handbook of Political Science*.
- Fenno, Richard F. 1978. *Home style: House members in their districts*. Pearson College Division.
- Findley, Michael G., Nathan M. Jensen, Edmund J. Malesky and Thomas B. Pepinsky. 2016. “Can Results-Free Review Reduce Publication Bias? The Results and Implications of a Pilot Study.” *Comparative Political Studies* 49(13):1667–1703.
- Fisher, Ronald A. 1935. “The logic of inductive inference.” *Journal of the royal statistical society* 98(1):39–82.
- Fisher, Ronald Aylmer. 1937. *The design of experiments*. Oliver And Boyd; Edinburgh; London.

- Frangakis, Constantine E and Donald B Rubin. 2002. "Principal stratification in causal inference." *Biometrics* 58(1):21–29.
- Freedman, David A. 2008. "On regression adjustments to experimental data." *Advances in Applied Mathematics* 40(2):180–193.
- Geddes, Barbara. 2003. *Paradigms and Sand Castles: Theory building and research design in comparative politics*. Ann Arbor, Michigan: University of Michigan Press.
- Gelman, Andrew and Guido Imbens. 2017. "Why high-order polynomials should not be used in regression discontinuity designs." *Journal of Business & Economic Statistics* (Forthcoming).
- Gelman, Andrew, Jennifer Hill and Aki Vehtari. 2020. *Regression and other stories*. Cambridge University Press.
- 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.
- Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.
- Gerring, John and Lee Cojocaru. 2016. "Selecting cases for intensive analysis: A diversity of goals and methods." *Sociological Methods & Research* 45(3):392–423.
- Goertz, Gary. 2008. "Choosing cases for case studies: A qualitative logic." *Newsletter of the APSA Section on Qualitative & Multi-Method Research* 6(2):11–4.
- Goertz, Gary and James Mahoney. 2012. *A tale of two cultures: Qualitative and quantitative research in the social sciences*. Princeton: Princeton University Press.
- Green, Donald P. and Andrej Tusicisny. 2012. "Statistical analysis of results from laboratory studies in experimental economics: A critique of current practice." Available at SSRN 2181654 .
- Green, Donald P. and Winston Lin. 2016. "Standard Operating Procedures: A Safety Net for Pre-Analysis Plans." *PS: Political Science & Politics* 49(3):495–499.
- Gulzar, Saad and Muhammad Yasir Khan. 2021. ““Good Politicians:” Experimental Evidence on Motivations for Political Candidacy and Government Performance.” Working paper.
- Hainmueller, Jens, Daniel J. Hopkins and Teppei Yamamoto. 2014. "Causal Inference in Conjoint Analysis: Understanding Multidimensional Choices via Stated Preference Experiments." *Political Analysis* 22(1):1–30.

- Halpern, Joseph Y. 2000. "Axiomatizing causal reasoning." *Journal of Artificial Intelligence Research* 12:317–337.
- Hartman, Erin and F. Daniel Hidalgo. 2018. "An equivalence approach to balance and placebo tests." *American Journal of Political Science* 62(4):1000–1013.
- Hedayat, A.S., Hansheng Cheng and Jennifer Pajda-De La O. 2019. "Existence of unbiased estimation for the minimum, maximum, and median in finite population sampling." *Statistics & Probability Letters* 153:192–195.
- Herron, Michael C. and Kevin M. Quinn. 2016. "A careful look at modern case selection methods." *Sociological Methods & Research* 45(3):458–492.
- Humphreys, Macartan. 2015. "Reflections on the ethics of social experimentation." *Journal of Globalization and Development* 6(1):87–112.
- Humphreys, Macartan and Alan Jacobs. 2017. "Qualitative inference from causal models." *Draft manuscript (version 0.2)*. Retrieved November 27:2017.
- Humphreys, Macartan and Alan M. Jacobs. 2015. "Mixing methods: A Bayesian approach." *American Political Science Review* 109(4):653–673.
- Humphreys, Macartan, Raul de la Sierra and Peter van der Windt. 2013. "Fishing, Commitment, and Communication: A Proposal for Comprehensive Non-binding Research Registration." *Political Analysis* 21(1):1–20.
- Imai, Kosuke. 2011. "Multivariate Regression Analysis for the Item Count Technique." *Journal of the American Statistical Association* 106(494):407–416.
- Imai, Kosuke, Gary King and Clayton Nall. 2009. "The essential role of pair matching in cluster-randomized experiments, with application to the Mexican universal health insurance evaluation." *Statistical Science* 24(1):29–53.
- 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. 2010. "Better LATE than nothing: Some comments on Deaton (2009) and Heckman and Urzua (2009)." *Journal of Economic Literature* 48(2):399–423.
- Imbens, Guido W. and Donald B. Rubin. 2015. *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge: Cambridge University Press.
- Jamison, Julian C. 2019. "The Entry of Randomized Assignment into the Social Sciences." *Journal of Causal Inference* 7(1):1–16.
- Johnson, Noel D and Alexandra A Mislin. 2011. "Trust games: A meta-analysis." *Journal of economic psychology* 32(5):865–889.
- King, Gary, Emmanuela Gakidou, Nirmala Ravishankar, Ryan T Moore, Jason Lakin, Manett Vargas, Martha María Téllez-Rojo, Juan Eugenio Hernán-

- dez Ávila, Mauricio Hernández Ávila and Héctor Hernández Llamas. 2007. "A "politically robust" experimental design for public policy evaluation, with application to the Mexican universal health insurance program." *Journal of Policy Analysis and Management* 26(3):479–506.
- King, Gary and Melissa Sands. 2015. "How Human Subjects Research Rules Mislead You and Your University, and What to Do About it." *Unpublished manuscript*.
- Kirkland, Patricia A. and Alexander Coppock. 2018. "Candidate Choice Without Party Labels: New Insights from Conjoint Survey Experiments." *Political Behavior* 40(3):571–591.
- Kling, Jeffrey R, Jeffrey B Liebman and Lawrence F Katz. 2007. "Experimental analysis of neighborhood effects." *Econometrica* 75(1):83–119.
- Levy, Jack S. 2008. "Case studies: Types, designs, and logics of inference." *Conflict management and peace science* 25(1):1–18.
- Lieberman, Evan S. 2005. "Nested analysis as a mixed-method strategy for comparative research." *American Political Science Review* 99(3):435–452.
- Lin, Winston. 2013. "Agnostic notes on regression adjustments to experimental data: Reexamining Freedman's critique." *Annals of Applied Statistics* 7(1):295–318.
- Lyall, Jason. 2020. "Preregister Your Ethical Redlines.". Working paper.
- Martin, Lisa. 1992. *Coercive Cooperation: Explaining Multilateral Economic Sanctions*. Princeton: Princeton University Press.
- McPhetres, Jonathon, Nihan Albayrak-Aydemir, Ana Barbosa Mendes, Elvina C. Chow, Patricio Gonzalez-Marquez, Erin Loukras, Annika Maus, Aoife O'Mahony, Christina Pomareda, Maximilian Primbs, Shalaine L. Sackman, Conor J. R. Smithson and Kirill Volodko. 2020. "A decade of theory as reflected in Psychological Science (2009-2019)." PsyArXiv.
- Mellan, Jonathan. 2021. "Rain, Rain, Go Away: 176 potential exclusion-restriction violations for studies using weather as an instrumental variable.". URL: osf.io/preprints/socarxiv/9qj4f
- Middleton, Joel A. 2008. "Bias of the regression estimator for experiments using clustered random assignment." *Statistics & probability letters* 78(16):2654–2659.
- Miguel, Edward, Colin Camerer, Katherine Casey, Joshua Cohen, Kevin M Esterling, Alan S. Gerber, Rachel Glennerster, Donald P. Green, Macartan Humphreys, Guido Imbens et al. 2014. "Promoting transparency in social science research." *Science* 343(6166):30.
- Mill, John Stuart. 1884. *A system of logic, ratiocinative and inductive: Being a*

- connected view of the principles of evidence and the methods of scientific investigation.* Harper.
- Miller, Judith Droitcour. 1984. A new survey technique for studying deviant behavior PhD thesis George Washington University.
- Montgomery, Jacob M. and Erin L. Rossiter. 2020. “So many questions, so little time: Integrating adaptive inventories into public opinion research.” *Journal of Survey Statistics and Methodology* 8(4):667–690.
- Morris, Tim P., Ian R. White and Michael J. Crowther. 2019. “Using simulation studies to evaluate statistical methods.” *Statistics in Medicine* .
- Mousa, Salma. 2020. “Building social cohesion between Christians and Muslims through soccer in post-ISIS Iraq.” *Science* 369(6505):866–870.
- Nosek, Brian A., George Alter, George C. Banks, Denny Borsboom, Sara D. Bowman, Steven J. Breckler, Stuart Buck, Christopher D. Chambers, Gilbert Chin, Garret Christensen et al. 2015. “Promoting an open research culture: Author guidelines for journals could help to promote transparency, openness, and reproducibility.” *Science* 348(6242):1422.
- Offer-Westort, Molly, Alexander Coppock and Donald P. Green. 2021. “Adaptive Experimental Design: Prospects and Applications in Political Science.” *American Journal of Political Science* .
- Oforosu, George and Daniel Posner. 2021. “Pre-analysis Plans: A Stocktaking.” *Perspectives on Politics* . Forthcoming.
- Oforosu, George K. and Daniel N. Posner. 2020. “Do Pre-analysis Plans Hamper Publication?” *AEA Papers and Proceedings* 110:70–74.
- Olken, Benjamin A. 2015. “Promises and Perils of Pre-Analysis Plans.” *Journal of Economic Perspectives* 29(3):61–80.
- Pearl, Judea. 1999. “Probabilities of causation: three counterfactual interpretations and their identification.” *Synthese* 121(1-2):93–149.
- Pearl, Judea. 2009. *Causality: Models, Reasoning and Inference*. Second edition ed. Cambridge: Cambridge University Press.
- Pearl, Judea and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic Books.
- Peer, Limor, Lilla Orr and Alexander Coppock. 2021. “Active Maintenance: A Proposal for the Long-term Computational Reproducibility of Scientific Results.” *PS: Political Science & Politics* . Forthcoming.
- Plümper, Thomas, Vera E. Troeger and Eric Neumayer. 2010. “Case selection and causal inference in qualitative research.” *British Journal of Political Science* .
- Porter, Ethan and Yamil Velez. 2021. “Placebo Selection In Survey Experiments: An Agnostic Approach.” *Policy and Society* . Forthcoming.

- Reiss, Peter C and Frank A Wolak. 2007. "Structural econometric modeling: Rationales and examples from industrial organization." *Handbook of econometrics* 6:4277–4415.
- Rennie, Drummond. 2004. "Trial registration." *JAMA: the Journal of the American Medical Association* 292(11):1359–1362.
- Rohlfing, Ingo. 2018. "Power and False Negatives in Qualitative Comparative Analysis: Foundations, Simulation and Estimation for Empirical Studies." *Political Analysis* 26(1):72–89.
- Rubin, Donald B. 1980. "Randomization Analysis of Experimental Data: The Fisher Randomization Test Comment." *Journal of the American Statistical Association* 75(371):591–593.
- Rubin, Donald B. 1984. "Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician." *The Annals of Statistics* 12(4):1151–1172.
- Rubinstein, Ariel. 1982. "Perfect equilibrium in a bargaining model." *Econometrica: Journal of the Econometric Society* pp. 97–109.
- Samii, Cyrus. 2016. "Causal Empiricism in Quantitative Research." *The Journal of Politics* 78(3):941–955.
- Samii, Cyrus and Peter M. Aronow. 2012. "On equivalencies between design-based and regression-based variance estimators for randomized experiments." *Statistics & Probability Letters* 82(2):365–370.
- Schrag, Zachary M. 2010. *Ethical imperialism: Institutional review boards and the social sciences, 1965–2009*. Johns Hopkins University Press.
- Schwarz, Susanne and Alexander Coppock. 2020. "What Have We Learned About Gender From Candidate Choice Experiments? A Meta-analysis of 67 Factorial Survey Experiments." *Journal of Politics* . Forthcoming.
- Seawright, Jason and John Gerring. 2008. "Case Selection Techniques in Case Study Research: A Menu of Qualitative and Quantitative Options." *Political Research Quarterly* 61(2):294–308.
- Shadish, William, Thomas D. Cook and Donald Thomas Campbell. 2002. *Experimental and quasi-experimental designs for generalized causal inference*. Boston: Houghton Mifflin.
- Sinclair, Betsy, Margaret McConnell and Donald P. Green. 2012. "Detecting Spillover Effects: Design and Analysis of Multilevel Experiments." *American Journal of Political Science* 56(4):1055–1069.
- Skocpol, Theda. 1979. *States and social revolutions: A comparative analysis of France, Russia and China*. Cambridge University Press.
- Slough, Tara. 2020. "The Ethics of Electoral Experimentation: Design-Based Recommendations." *Unpublished Manuscript* .

- Swank, Duane. 2002. *Global Capital, Political Institutions, and Policy Change in Developed Welfare States*. New York: Cambridge University Press.
- Teele, Dawn. 2021. Virtual Consent: A Bronze Standard for Experimental Ethics. In *Cambridge Handbook of Experimental Political Science*, ed. Donald P. Green and James N. Druckman. Cambridge University Press.
- Van Evera, Stephen. 1997. *Guide to Methods for Students of Political Science*. Ithaca: Cornell University Press.
- Wager, Stefan and Susan Athey. 2018. “Estimation and inference of heterogeneous treatment effects using random forests.” *Journal of the American Statistical Association* 113(523):1228–1242.
- White, Ariel R., Noah L. Nathan and Julie K. Faller. 2015. “What Do I Need to Vote? Bureaucratic Discretion and Discrimination by Local Election Officials.” *American Political Science Review* 109(1):129–142.
- Wilke, Anna M., Donald P. Green and Jasper Cooper. 2020. “A placebo design to detect spillovers from an education–entertainment experiment in Uganda.” *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 183(3):1075–1096.
- Wilke, Anna M. and Macartan Humphreys. 2020. “Field Experiments, Theory, and External Validity.”
- Wood, Elisabeth Jean. 2006. “The Ethical Challenges of Field Research in Conflict Zones.” *Qualitative Sociology* 29(3):373–386.
- Yamamoto, Teppei. 2012. “Understanding the past: Statistical analysis of causal attribution.” *American Journal of Political Science* 56(1):237–256.
- Zarin, Deborah A. and Tony Tse. 2008. “Moving towards transparency of clinical trials.” *Science* 319(5868):1340–1342.
- Zhang, Junni L. and Donald B. Rubin. 2003. “Estimation of causal effects via principal stratification when some outcomes are truncated by “death”.” *Journal of Educational and Behavioral Statistics* 28(4):353–368.