Mark van der Laan, Jeremy Coyle, Nima Hejazi, Ivana Malenica, Rachael Phillips, Alan Hubbard

Targeted Learning in R

Causal Data Science with the tlverse Software Ecosystem

Contents

List of Tables

List of Figures

About this book

Targeted Learning in `R`*: Causal Data Science with the* `tlverse` *Software Ecosystem* is an open source, reproducible electronic handbook for applying the Targeted Learning methodology in practice using the `tlverse` software ecosystem. This work is currently in an early draft phase and is available to facilitate input from the community. To view or contribute to the available content, consider visiting the GitHub repository.

0.1 Outline

The contents of this handbook are meant to serve as a reference guide for applied research as well as materials that can be taught in a series of short courses focused on the applications of Targeted Learning. Each section introduces a set of distinct causal questions, motivated by a case study, alongside statistical methodology and software for assessing the causal claim of interest. The (evolving) set of materials includes

- Motivation: Why we need a statistical revolution
- The Roadmap and introductory case study: the WASH Beneifits data
- Introduction to the `tlverse` software ecosystem
- Cross-validation with the `origami` package
- Ensemble machine learning with the `sl3` package
- Targeted learning for causal inference with the `tmle3` package
- Optimal treatments regimes and the `tmle3mopttx` package
- Stochastic treatment regimes and the `tmle3shift` package
- Causal mediation analysis with the `tmle3mediate` package
- *Coda*: Why we need a statistical revolution

What this book is not

The focus of this work is **not** on providing in-depth technical descriptions of current statistical methodology or recent advancements. Instead, the goal is to convey key details of state-of-the-art techniques in an manner that is both clear and complete, without burdening the reader with extraneous information. We hope that the presentations herein will serve as references for researchers – methodologists and domain specialists alike – that empower them to deploy the central tools of Targeted Learning in an efficient manner. For technical details and in-depth descriptions of both classical theory and recent advances in the field of Targeted Learning, the interested reader is invited to consult van der Laan and Rose (2011) and/or van der Laan and Rose (2018) as appropriate. The primary literature in statistical causal inference, machine learning, and non/semiparametric theory include many of the most recent advances in Targeted Learning and related areas.

About the authors

Mark van der Laan

Mark van der Laan, PhD, is Professor of Biostatistics and Statistics at UC Berkeley. His research interests include statistical methods in computational biology, survival analysis, censored data, adaptive designs, targeted maximum likelihood estimation, causal inference, data-adaptive loss-based learning, and multiple testing. His research group developed loss-based super learning in semiparametric models, based on cross-validation, as a generic optimal tool for the estimation of infinite-dimensional parameters, such as nonparametric density estimation and prediction with both censored and uncensored data. Building on this work, his research group developed targeted maximum likelihood estimation for a target parameter of the data-generating distribution in arbitrary semiparametric and nonparametric models, as a generic optimal methodology for statistical and causal inference. Most recently, Mark's group has focused in part on the development of a centralized, principled set of software tools for targeted learning, the `tlverse`.

Jeremy Coyle

Jeremy Coyle, PhD, is a consulting data scientist and statistical programmer, currently leading the software development effort that has produced the `tlverse` ecosystem of R packages and related software tools. Jeremy earned his PhD in Biostatistics from UC Berkeley in 2016, primarily under the supervision of Alan Hubbard.

Nima Hejazi

Nima Hejazi is a PhD candidate in biostatistics, working under the collaborative direction of Mark van der Laan and Alan Hubbard. Nima is affiliated with UC Berkeley's Center for Computational Biology and NIH Biomedical Big Data training program, as well as with the Fred Hutchinson Cancer Research Center. Previously, he earned an MA in Biostatistics and a BA (with majors in Molecular and Cell Biology, Psychology, and Public Health), both at UC Berkeley. His research interests fall at the intersection of causal inference and machine learning, drawing on ideas from non/semi-parametric estimation in large, flexible statistical models to develop efficient and robust statistical procedures for evaluating complex target estimands in observational and randomized studies. Particular areas of current emphasis include mediation/path analysis, outcome-dependent sampling designs, targeted loss-based estimation, and vaccine efficacy trials. Nima is also passionate about statistical computing and open source software development for applied statistics.

Ivana Malenica

Ivana Malenica is a PhD student in biostatistics advised by Mark van der Laan. Ivana is currently a fellow at the Berkeley Institute for Data Science, after serving as a NIH Biomedical Big Data and Freeport-McMoRan Genomic Engine fellow. She earned her Master's in Biostatistics and Bachelor's in Mathematics, and spent some time at the Translational Genomics Research Institute. Very broadly, her research interests span non/semi-parametric theory, probability theory, machine learning, causal inference and high-dimensional statistics. Most of her current work involves complex dependent settings (dependence through time and network) and adaptive sequential designs.

Rachael Phillips

Rachael Phillips is a PhD student in biostatistics, advised by Alan Hubbard and Mark van der Laan. She has an MA in Biostatistics, BS in Biology, and BA in Mathematics. A student of targeted learning and causal inference, Rachael's research focuses on statistical estimation and inference in realistic statistical models. Her

current projects involve personalized online machine learning from EHR streaming data of vital signs, automated learning with highly adaptive lasso, and causal effect estimation for community-level interventions. She is also working on an FDA-funded project led Dr. Susan Gruber, A Targeted Learning Framework for Causal Effect Estimation Using Real-World Data. Rachael is an active contributor to the `hal9001` and `sl3` R packages in the `tlverse`.

Alan Hubbard

Alan Hubbard is Professor of Biostatistics, former head of the Division of Biostatistics at UC Berkeley, and head of data analytics core at UC Berkeley's SuperFund research program. His current research interests include causal inference, variable importance analysis, statistical machine learning, estimation of and inference for data-adaptive statistical target parameters, and targeted minimum loss-based estimation. Research in his group is generally motivated by applications to problems in computational biology, epidemiology, and precision medicine.

0.2 Learning resources

To effectively utilize this handbook, the reader need not be a fully trained statistician to begin understanding and applying these methods. However, it is highly recommended for the reader to have an understanding of basic statistical concepts such as confounding, probability distributions, confidence intervals, hypothesis tests, and regression. Advanced knowledge of mathematical statistics may be useful but is not necessary. Familiarity with the `R` programming language will be essential. We also recommend an understanding of introductory causal inference.

For learning the `R` programming language we recommend the following (free) introductory resources:

- Software Carpentry's *Programming with* `R`
- Software Carpentry's `R` *for Reproducible Scientific Analysis*
- Garret Grolemund and Hadley Wickham's `R` *for Data Science*

For a general introduction to causal inference, we recommend

- Miguel A. Hernán and James M. Robins' *Causal Inference: What If*, 2021

- Jason A. Roy's *A Crash Course in Causality: Inferring Causal Effects from Observational Data* on Coursera

0.3 Setup instructions

0.3.1 R and RStudio

R and **RStudio** are separate downloads and installations. R is the underlying statistical computing environment. RStudio is a graphical integrated development environment (IDE) that makes using R much easier and more interactive. You need to install R before you install RStudio.

0.3.1.1 Windows

0.3.1.1.1 If you already have R and RStudio installed

- Open RStudio, and click on "Help" > "Check for updates". If a new version is available, quit RStudio, and download the latest version for RStudio.
- To check which version of R you are using, start RStudio and the first thing that appears in the console indicates the version of R you are running. Alternatively, you can type `sessionInfo()`, which will also display which version of R you are running. Go on the CRAN website and check whether a more recent version is available. If so, please download and install it. You can check here for more information on how to remove old versions from your system if you wish to do so.

0.3.1.1.2 If you don't have R and RStudio installed

- Download R from the CRAN website.
- Run the `.exe` file that was just downloaded
- Go to the RStudio download page
- Under *Installers* select **RStudio x.yy.zzz - Windows XP/Vista/7/8** (where x, y, and z represent version numbers)
- Double click the file to install it
- Once it's installed, open RStudio to make sure it works and you don't get any error messages.

0.3.1.2 macOS / Mac OS X

0.3.1.2.1 If you already have R and RStudio installed

- Open RStudio, and click on "Help" > "Check for updates". If a new version is available, quit RStudio, and download the latest version for RStudio.
- To check the version of R you are using, start RStudio and the first thing that appears on the terminal indicates the version of R you are running. Alternatively, you can type `sessionInfo()`, which will also display which version of R you are running. Go on the CRAN website and check whether a more recent version is available. If so, please download and install it.

0.3.1.2.2 If you don't have R and RStudio installed

- Download R from the CRAN website.
- Select the `.pkg` file for the latest R version
- Double click on the downloaded file to install R
- It is also a good idea to install XQuartz (needed by some packages)
- Go to the RStudio download page
- Under *Installers* select **RStudio x.yy.zzz - Mac OS X 10.6+ (64-bit)** (where x, y, and z represent version numbers)
- Double click the file to install RStudio
- Once it's installed, open RStudio to make sure it works and you don't get any error messages.

0.3.1.3 Linux

- Follow the instructions for your distribution from CRAN, they provide information to get the most recent version of R for common distributions. For most distributions, you could use your package manager (e.g., for Debian/Ubuntu run `sudo apt-get install r-base`, and for Fedora `sudo yum install R`), but we don't recommend this approach as the versions provided by this are usually out of date. In any case, make sure you have at least R 3.3.1.
- Go to the RStudio download page
- Under *Installers* select the version that matches your distribution, and install it with your preferred method (e.g., with Debian/Ubuntu `sudo dpkg -i rstudio-x.yy.zzz-amd64.deb` at the terminal).
- Once it's installed, open RStudio to make sure it works and you don't get any error messages.

These setup instructions are adapted from those written for Data Carpentry: R for Data Analysis and Visualization of Ecological Data.

1

Robust Statistics and Reproducible Science

> "One enemy of robust science is our humanity – our appetite for being right, and our tendency to find patterns in noise, to see supporting evidence for what we already believe is true, and to ignore the facts that do not fit."
>
> — *Nature* Editorial (Anonymous) (2015b)

Scientific research is at a unique point in its history. The need to improve rigor and reproducibility in our field is greater than ever; corroboration moves science forward, yet there is growing alarm that results cannot be reproduced or validated, suggesting the possibility that many discoveries may be false (Baker, 2016). Consequences of not meeting this need will result in further decline in the rate of scientific progress, the reputation of the sciences, and the public's trust in scientific findings (Munafò et al., 2017; *Nature* Editorial (Anonymous), 2015a).

> "The key question we want to answer when seeing the results of any scientific study is whether we can trust the data analysis."
>
> — Peng (2015)

Unfortunately, in its current state, the culture of statistical data analysis enables, rather than precludes, the manner in which human bias may affect the results of (ideally objective) data analytic efforts. A significant degree of human bias enters statistical analysis efforts in the form improper model selection. All procedures for estimation and hypothesis testing are derived based on a choice of statistical model; thus, obtaining valid estimates and statistical inference relies critically on the chosen statistical model containing an accurate representation of the process that generated the data. Consider, for example, a hypothetical study in which a treatment was assigned to a group of patients: Was the treatment assigned randomly or were characteristics of the individuals (i.e., baseline covariates) used in making the treatment decision? Such knowledge can should be incorporated in the statistical model. Alternatively, the data could be from an observational study, in which there is no control over the treatment assignment mechanism. In such cases, available knowledge about the data-generating process (DGP) is more limited still. If this is the

case, then the statistical model should contain *all* possible distributions of the data. In practice, however, models are not selected based on scientific knowledge available about the DGP; instead, models are often selected based on (1) the philosophical leanings of the analyst, (2) the relative convenience of implementation of statistical methods admissible within the choice of model, and (3) the results of significance testing (i.e., p-values) applied within the choice of model.

This practice of "cargo-cult statistics — the ritualistic miming of statistics rather than conscientious practice," (Stark and Saltelli, 2018) is characterized by arbitrary modeling choices, even though these choices often result in different answers to the same research question. That is, "increasingly often, [statistics] is used instead to aid and abet weak science, a role it can perform well when used mechanically or ritually," as opposed to its original purpose of safeguarding against weak science by providing formal techniques for evaluating the veracity of a claim using properly collected data (Stark and Saltelli, 2018). This presents a fundamental drive behind the epidemic of false findings from which scientific research is suffering (van der Laan and Starmans, 2014).

> "We suggest that the weak statistical understanding is probably due to inadequate"statistics lite" education. This approach does not build up appropriate mathematical fundamentals and does not provide scientifically rigorous introduction into statistics. Hence, students' knowledge may remain imprecise, patchy, and prone to serious misunderstandings. What this approach achieves, however, is providing students with false confidence of being able to use inferential tools whereas they usually only interpret the p-value provided by black box statistical software. While this educational problem remains unaddressed, poor statistical practices will prevail regardless of what procedures and measures may be favored and/or banned by editorials."
>
> — Szucs and Ioannidis (2017)

Our team at the University of California, Berkeley is uniquely positioned to provide such an education. Spearheaded by Professor Mark van der Laan, and spreading rapidly by many of his students and colleagues who have greatly enriched the field, the aptly named "Targeted Learning" methodology emphasizes a focus of (i.e., "targeting of") the scientific question at hand, running counter to the current culture problem of "convenience statistics," which opens the door to biased estimation, misleading analytic results, and erroneous discoveries. Targeted Learning embraces the fundamentals that formalized the field of statistics, notably including the notions that a statistical model must represent real knowledge about the experiment that generated the data and that

a target parameter represents what we are seeking to learn from the data as a feature of the distribution that generated it (van der Laan and Starmans, 2014). In this way, Targeted Learning defines a truth and establishes a principled standard for estimation, thereby curtailing our all-too-human biases (e.g., hindsight bias, confirmation bias, and outcome bias) from infiltrating our objective analytic efforts.

> "The key for effective classical [statistical] inference is to have well-defined questions and an analysis plan that tests those questions."
>
> — Nosek et al. (2018)

This handbook aims to provide practical training to students, researchers, industry professionals, and academicians in the sciences (whether biological, physical, economic, or social), public health, statistics, and numerous other fields, to equip them with the necessary knowledge and skills to utilize the the methodological developments of Targeted Learning — a technique that provides tailored pre-specified machines for answering queries — taking advantage of estimators that are efficient, minimally biased, and that provide formal statistical inference — so that each and every data analysis incorporates state-of-the-art statistical methodology, all while ensuring compatibility with the guiding principles of computational reproducibility.

Just as the conscientious use of modern statistical methodology is necessary to ensure that scientific practice thrives, robust, well-tested software plays a critical role in allowing practitioners to direct access the published results of a given scientific investigation. In fact, "an article. . . in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures," thus making the availability and adoption of robust statistical software key to enhancing the transparency that is an inherent (and assumed) aspect of the scientific process (Buckheit and Donoho, 1995).

For a statistical methodology to be readily accessible in practice, it is crucial that it is accompanied by user-friendly software (Pullenayegum et al., 2016; Stromberg et al., 2004). The `tlverse` software ecosystem, composed of a set of package for the `R` language and environment for statistical computing (R Core Team, 2021), was developed to fulfill this need for the Targeted Learning methodological framework. Not only does this suite of software tools facilitate computationally reproducible and efficient analyses, it is also a tool for Targeted Learning education, since its workflow mirrors the central aspects of the statistical methodology. In particular, the programming paradigm central to the `tlverse` ecosystem does not focus on implementing a specific estimator or a small set of related estimators. Instead, the focus is on exposing the

statistical framework of Targeted Learning itself — all software packages in the `tlverse` ecosystem directly model the key objects defined in the mathematical and theoretical framework of Targeted Learning. What's more, the `tlverse` software packages share a core set of design principles centered on extensibility, allowing for them all to be used in conjunction with each other and even used cohesively as building blocks for formulating sophisticated statistical analyses. For an introduction to the Targeted Learning framework, we recommend a recent review paper from Coyle et al. (2021).

In this handbook, the reader will embark on a journey through the `tlverse` ecosystem. Guided by `R` programming exercises, case studies, and intuition-building explanations, readers will learn to use a toolbox for applying the Targeted Learning statistical methodology, which will translate to real-world causal inference analyses. Some preliminaries are required prior to this learning endeavor – we have made available a list of recommended learning resources.

2

Meet the Data

Targeted Learning is all about learning from data. We'll use a few example datasets throughout this book. We introduce them in this chapter.

2.1 Learning Objectives

By the end of this chapter you will be able to:

1. Understand the various datasets we'll use as case studies
2. Have a vague understanding of the kinds of scientific questions we might want to answer with them

2.2 Schematic Example

This is an entirely artificial example with three variables that's helpful for illustrating key concepts.

This dataset is loaded with

```
data(schematic, package="tlverse")
```

Here's a table with a few rows of the data:

```
   W A         Y
1 10 1  4.72968
2  6 0 -0.20798
3  5 0 -0.25256
4  9 0 -2.04532
5  5 0 -0.25444
6  6 1  0.73052
```

2.2.1 Schematic Variables

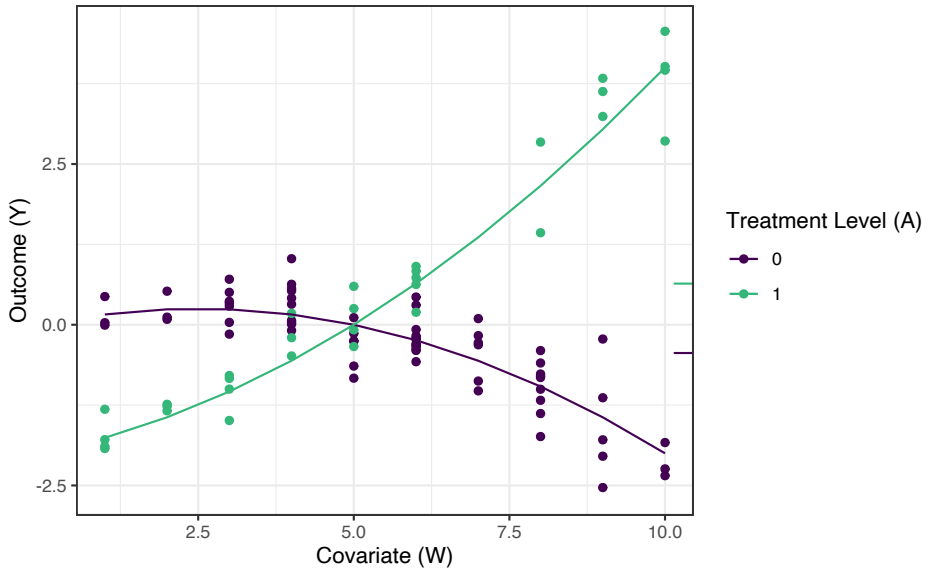
The variables should be interpreted as follows:

W — a baseline covariate, in this case an integer ranging from 1 to 10. You can think of this as someone's age or some other feature about a person. Usually you have a lot of these, but in this case we have only one.

A — a treatment or intervention, in this case it's either 0 or 1. You can think of this as some treatment we're interested in learning the effects of. We can say that 1 means a person got the treatment and 0 means that a person didn't (they got a placebo or nothing at all)

Y — an outcome, in this case it's a continuous measure that has a range roughly between -4 and 4. You can think of it as some outcome we're interested in, like death. Maybe it's a good outcome, and we hope that by giving the treatment we'll increase it. Maybe it's a bad outcome, and we hope that by giving the treatment we'll decrease it.

Because it's so simple, it's easy to visualize on a single plot:



We want to use the data to figure out the effect of the treatment A on outcome Y, while adjusting for covariate(s) W (we'll see what that's important later). Generally speaking, a lot of data questions can be framed this way. Of course, the devil is in the details. We'll see later how important it is to get the details correctly specified.

2.3 WASH Benefits

These data come from a study of the effect of water quality, sanitation, hand washing, and nutritional interventions on child development in rural Bangladesh (WASH Benefits Bangladesh): a cluster randomized controlled trial (Tofail et al., 2018). For reference, this trial was registered with ClinicalTrials.gov as NCT01590095. The study enrolled pregnant women in their first or second trimester from the rural villages of Gazipur, Kishoreganj, Mymensingh, and Tangail districts of central Bangladesh, with an average of eight women per cluster. Groups of eight geographically adjacent clusters were block randomized, using a random number generator, into six intervention groups (all of which received weekly visits from a community health promoter for the first 6 months and every 2 weeks for the next 18 months) and a double-sized control group (no intervention or health promoter visit). In this book, we concentrate on child growth (size for age) as the outcome of interest

This dataset is loaded with

```
data(tlverse_washb)
```

TODO: table

The six intervention groups were:

1. chlorinated drinking water;
2. improved sanitation;
3. hand-washing with soap;
4. combined water, sanitation, and hand washing;
5. improved nutrition through counseling and provision of lipid-based nutrient supplements; and
6. combined water, sanitation, handwashing, and nutrition.

We have 28 variables measured. This outcome, Y, is the weight-for-height Z-score (`whz` in `dat`); the treatment of interest, A, is the randomized treatment group (`tr` in `dat`); and the adjustment set, W, consists simply of *everything else*.

2.4 International Stroke Trial (Rachael to replace with different dataset for exercises)

The International Stroke Trial database contains individual patient data from the International Stroke Trial (IST), a multi-national randomized trial conducted between 1991 and 1996 (pilot phase between 1991 and 1993) that aimed to assess whether early administration of aspirin, heparin, both aspirin and heparin, or neither influenced the clinical course of acute ischaemic stroke (Sandercock et al., 1997). The IST dataset includes data on 19,435 patients with acute stroke, with 99% complete follow-up. De-identified data are available for download at `https://datashare.is.ed.ac.uk/handle/10283/128`. This study is described in more detail in Sandercock et al. (2011). The example data for this handbook considers a sample of 5,000 patients and the binary outcome of recurrent ischemic stroke within 14 days after randomization. Also in this example data, we ensure that we have subjects with a missing outcome.

We have 26 variables measured, and the outcome of interest, Y, indicates recurrent ischemic stroke within 14 days after randomization (`DRSISC` in `ist`); the treatment of interest, A, is the randomized aspirin vs. no aspirin treatment allocation (`RXASP` in `ist`); and the adjustment set, W, consists of all other variables measured at baseline.

This dataset is loaded with

```
data(tlverse_ist)
```

Like before, we can summarize the variables measured in the IST sample data set with `skimr`:

TODO: table

3

The Roadmap for Targeted Learning

In this chapter you will...

1. Translate scientific questions to statistical questions.
2. Define a statistical model based on the knowledge of the experiment that generated the data.
3. Identify a causal parameter as a function of the observed data distribution.
4. Explain the following causal and statistical assumptions and their implications: i.i.d., consistency, interference, positivity, SUTVA.

3.1 Introduction

The roadmap of statistical learning is concerned with the translation from real-world data applications to a mathematical and statistical formulation of the relevant estimation problem. This involves data as a random variable having a probability distribution, scientific knowledge represented by a statistical model, a statistical target parameter representing an answer to the question of interest, and the notion of an estimator and sampling distribution of the estimator.

3.2 The Roadmap

Following the roadmap is a process of five steps.

1. Data: Data as a random variable with a probability distribution, $O \sim P_0$
2. Model: The statistical model $\mathcal{M}$ such that $P_0 \in \mathcal{M}$
3. Parameter: The statistical target parameter Ψ and estimand $\Psi(P_0)$.
4. Estimation: The estimator $\hat{\Psi}$ and estimand $\hat{\Psi}(P_n)$.
5. Inference: A measure of uncertainty for the estimate $\hat{\Psi}(P_n)$

3.3 Schematic Example

Remember the schematic from last chapter? Let's start a roadmap for it before going on to talk about the steps in more detail

3.3.1 Data Step

We can describe the data as a set of observations about an individual (it's more general to say experimental unit) and, for our schematic example, we can denote an observation like so:

$$O \equiv (W, A, Y)$$

a collection of facts (here W, A, and Y) about an individual observation O. We think of a set of data as a set of such observations. We think of that observation as a random draw from a distribution of possible observations we denote P_0 (here the subscript 0 denotes the real one, we'll use other subscripts to denote theoretical or estimated distributions). We call P_0 the probability distribution or data generating process (DGP).

How the observation is drawn from the sample is important. That's called the experiment. How to translate between a real world experiment and a probability model is outside the scope of this book. For now, we'll focus on what we call independent and identically distributed (i.i.d.) data. That means that each unit O got drawn from the same P_0 in the same way. No other sample can change another samples outcome, and all samples get drawn from the same imaginary box. Options and modifications of our methodology are available for for complex and biased samples, repeated measures, and other sampling concerns.

Luckily for us, we have just such data in our schematic dataset.

3.3.2 Model Step

Just like we had a set of observations we called a dataset, we have a set of possible probability distributions. You might think we call that a distribution set, but we don't, we call it a model. We denote it $\mathcal{M}$ and we write:

$$P_0 \in \mathcal{M}$$

To indicate that the true DGP is part of our model. This is important, because if our model doesn't contain the truth, it will be impossible for us to get the right answer, even with infinite data!

Well, what can we say about $\mathcal{M}$? That is, what can we say for sure about what P_0 might look like. Given that I haven't told you much about the data or the experiment, really very little! We'll see in later chapters how some statisticians want to do statistics in small models, that we can be quite sure don't contain P_0 , because it makes the statistics easier. For now we'll just say that $\mathcal{M}$ is nonparametric, which essentially means that we can't make any assumptions about it.

The truth is, we can make a few assumptions based on the observed data types and our belief that we've observed all the values of some of the variables. For example, we think A can only be 0 or 1, and Wranges between 1 and 10. It also seems like Y varies in a small range, so we could incorporate that as a modeling assumption if we were fairly confident that that's its true range. We don't often write these things as part of the model explicitly, but they are part of it.

3.3.3 Parameter Step

We said that we want to know about the effect of the treatment A on the outcome Y. There's a lot of ways we could formalize that mathematically, but here's one we like:

$$\Psi_{0,\text{TSM}} = E_W[E_{Y|A,W}[Y|A = 1, W]]$$

we call this a Treatment Specific Mean (TSM):

Basically, we want to know the mean of Y for every W, when we set $A = 1$. We then want to take a mean across Ws, which we call "marginalizing". We say call this a treatment specific mean because it's the mean outcome Y we'd expect under the specific treatment $A = 1$. That tells us something about how treatment affects outcome. However, we'd often like to compare outcomes under two conditions. We can use a pair of TSMs to make an Average Treatment Effect (ATE):

$$\Psi_{0,\text{ATE}} = E_W[E_{Y|A,W}[Y|A = 1, W]] - E_W[E_{Y|A,W}[Y|A = 0, W]]$$

Many other types of parameters like relative risks and odds ratios can be defined by simple combinations of TSMs. We'll see later how we can use the Delta Method to estimate parameters like these starting with estimates of TSMs.

3.3.4 Estimation Step

Explain plug-ins here

Say we'll see more in next two chapters

3.3.5 Inference Step

We'll cover this later

3.4 WASH Benefits Example

3.4.1 Data Step

We still say $O \equiv (W, A, Y)$, except now W is a vector of many covariates.

For the purposes of this handbook, we will say that the sample was generated i.i.d as before. This study had a cluster design, so this is not actually the case. We could, with available options, account for the clustering of the data.

3.4.2 Model Step

We still don't know anything, so we'll stick with a nonparametric model $\mathcal{M}$.

3.4.3 Parameter Step

We would like to estimate TSMs for every treatment level, as well as ATEs between some treatment levels and the control treatment.

3.4.4 Estimation Step

3.4.5 Inference Step

3.5 Causal Concerns

Current roadmap text goes here

3.6 Exercises

4

Welcome to the `tlverse`

4.1 In this chapter you will...

1. Understand the `tlverse` ecosystem conceptually
2. Identify the core components of the `tlverse`
3. Install `tlverse` `R` packages
4. Understand the Targeted Learning roadmap
5. Learn about the WASH Benefits example data

By now you're probably thinking. I thought I bought a book about software. Where's my receipt? I want my money back. Maybe you're reading this online and thinking boy I'm sure glad I didn't buy the print copy. But your patience has paid off. We're finally ready to talk about software.

4.2 What is the `tlverse`?

The `tlverse` is a new framework for doing Targeted Learning in R, inspired by the [`tidyverse` ecosystem](`https://tidyverse.org/`) of R packages.

By analogy to the [`tidyverse`](`https://tidyverse.org/`):

> The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

So, the [`tlverse`](`https://tlverse.org/`) is

- an opinionated collection of R packages for Targeted Learning
- sharing an underlying philosophy, grammar, and set of data structures

4.3 Anatomy of the tlverse

These are the main packages that represent the **core** of the tlverse:

- [sl3](https://github.com/tlverse/sl3): Modern Super Learning with Pipelines
  - *What?* A modern object-oriented re-implementation of the Super Learner algorithm, employing recently developed paradigms for R programming.
  - *Why?* A design that leverages modern tools for fast computation, is forward-looking, and can form one of the cornerstones of the tlverse.
- [tmle3](https://github.com/tlverse/tmle3): An Engine for Targeted Learning
  - *What?* A generalized framework that simplifies Targeted Learning by identifying and implementing a series of common statistical estimation procedures.
  - *Why?* A common interface and engine that accommodates current algorithmic approaches to Targeted Learning and is still flexible enough to remain the engine even as new techniques are developed.

In addition to the engines that drive development in the tlverse, there are some supporting packages – in particular, we have two...

- [origami](https://github.com/tlverse/origami): A Generalized Framework for Cross-Validation
  - *What?* A generalized framework for flexible cross-validation
  - *Why?* Cross-validation is a key part of ensuring error estimates are honest and preventing overfitting. It is an essential part of the both the Super Learner algorithm and Targeted Learning.
- [delayed](https://github.com/tlverse/delayed): Parallelization Framework for Dependent Tasks
  - *What?* A framework for delayed computations (futures) based on task dependencies.
  - *Why?* Efficient allocation of compute resources is essential when deploying large-scale, computationally intensive algorithms.

A key principle of the tlverse is extensibility. That is, we want to support new Targeted Learning estimators as they are developed. The model for this is new estimators are implemented in additional packages using the core packages above. There are currently two featured examples of this:

- [tmle3mopttx](https://github.com/tlverse/tmle3mopttx): Optimal Treatments in tlverse
  - *What?* Learn an optimal rule and estimate the mean outcome under the rule
  - *Why?* Optimal Treatment is a powerful tool in precision healthcare and other settings where a one-size-fits-all treatment approach is not appropriate.
- [tmle3shift](https://github.com/tlverse/tmle3shift): Shift Interventions in tlverse
  - *What?* Shift interventions for continuous treatments
  - *Why?* Not all treatment variables are discrete. Being able to estimate the effects of continuous treatment represents a powerful extension of the Targeted Learning approach.

4.4 Installation

The tlverse ecosystem of packages are currently hosted at https://github.com/tlverse, not yet on CRAN. You can use the [usethis package](https://usethis.r-lib.org/) to install them:

```
install.packages("usethis")usethis::install_github("tlverse/tlverse")
```

The tlverse depends on a large number of other packages that are also hosted on GitHub. Because of this, you may see the following error:

```
Error: HTTP error 403.
  API rate limit exceeded for 71.204.135.82. (But here's the good news:
  Authenticated requests get a higher rate limit. Check out the documentation
  for more details.)

  Rate limit remaining: 0/60
  Rate limit reset at: 2019-03-04 19:39:05 UTC

  To increase your GitHub API rate limit
  - Use `usethis::browse_github_pat()` to create a Personal Access Token.
  - Use `usethis::edit_r_environ()` and add the token as `GITHUB_PAT`.
```

This just means that R tried to install too many packages from GitHub in too short of a window. To fix this, you need to tell R how to use GitHub as your user (you'll need a GitHub user account). Follow these two steps:

1. Type `usethis::browse_github_pat()` in your R console, which will direct you to GitHub's page to create a New Personal Access Token.
2. Create a Personal Access Token simply by clicking "Generate token" at the bottom of the page.
3. Copy your Personal Access Token, a long string of lowercase letters and numbers.
4. Type `usethis::edit_r_environ()` in your R console, which will open your `.Renviron` file in the source window of RStudio. If you are not able to access your `.Renviron` file with this command, then try inputting `Sys.setenv(GITHUB_PAT = )` with your Personal Access Token inserted as a string after the equals symbol; and if this does not error, then skip to step 8.
5. In your `.Renviron` file, type `GITHUB_PAT=` and then paste your Personal Access Token after the equals symbol with no space.
6. In your `.Renviron` file, press the enter key to ensure that your `.Renviron` ends with a newline.
7. Save your `.Renviron` file.
8. Restart R for changes to take effect. You can restart R via the drop-down menu on the "Session" tab. The "Session" tab is at the top of the RStudio interface.

After following these steps, you should be able to successfully install the package which threw the error above.

5

Cross-validation

5.1 Introduction

5.1.1 Roadmap Review

5.1.2 We want to fit the data to estimate Q

5.1.3 We can propose and test models

5.2 Learning Objectives

By the end of this chapter you will be able to:

1. Differentiate between training, validation and test sets.
2. Understand the concept of a loss function, risk and cross-validation.
3. Select a loss function that is appropriate for the functional parameter to be estimated.
4. Understand and contrast different cross-validation schemes for i.i.d. data.
5. Understand and contrast different cross-validation schemes for time dependent data.
6. Setup the proper fold structure, build custom fold-based function, and cross-validate the proposed function using the `origami` `R` package.
7. Setup the proper cross-validation structure for the use by the Super Learner using the the `origami` `R` package.

5.3 schematic example

5.3.1 show overfit on test set

5.3.2 show cross-validation

5.4 Conceptual

5.5 washb example

5.6 advanced usage

6

Super (Machine) Learning

Learning Objectives

By the end of this chapter you will be able to:

1. Select an objective function that (i) aligns with the intention of the analysis and (ii) is optimized by the target parameter.
2. Assemble a diverse library of learners to be considered in the Super Learner ensemble. In particular, you should be able to:
    a. Customize a learner by modifying it's tuning parameters.
    b. Create several different versions of the same learner at once by specifying a grid of tuning parameters.
    c. Curate covariate screening pipelines in order to pass a screener's output, a subset of covariates, as input for another learner that will use the subset of covariates selected by the screener to model the data.
3. Specify the learner for ensembling (the metalearner) such that it corresponds to your objective function.
4. Fit the Super Learner ensemble with nested cross-validation to obtain an estimate of the performance of the ensemble itself on out-of-sample data.
5. Obtain `sl3` variable importance metrics.

6. Interpret the fit for discrete and continuous Super Learners' from the cross-validated risk table and the coefficients.
7. Justify the base library of machine learning algorithms and the ensembling learner in terms of the prediction problem, statistical model $\mathcal{M}$, data sparsity, and the dimensionality of the covariates.

6.2 Setup

```
library(sl3)
library(tlverse)
library(data.table)
library(ggplot2)
```

6.3 Schematic Example

We can define a `sl3` task as follows:

```
data(schematic, package="tlverse")
task <- make_sl3_Task(schematic,
                      covariates=c("A","W"),
                      outcome="Y")
```

This is a way of organizing data and metadata together. In essence, it's telling the `tlverse` about the *Data* step of the roadmap.

Next, we can make some guesses about the *Model* step of the roadmap. This is similar to what we did in the [#origami] chapter:

```
# this defaults to a linear fit, so ~ A + W in this case
gl <- make_learner(Lrnr_glm)
gl_interaction <- make_learner(Lrnr_glm,
                               formula = "~ A*W")
gl_poly2 <- make_learner(Lrnr_glm,
                         formula = "~ A*(W + I(W^2))")
gl_poly4 <- make_learner(Lrnr_glm,
                         formula = "~ A*(W + I(W^2) + I(W^3) + I (W^4))")
```

We can train one of these on our task and generate predictions:

```
gl_fit <- gl$train(task)
preds <- gl_fit$predict(task)
head(preds)
[1]  1.74048 -0.33978 -0.52219  0.20743 -0.52219  1.01086
```

We can also ‘ensemble''these together using a special learner calledLrnr_sl‘. We’ll explain more in a minute what this means.

```
learners <- list(linear = gl,
                 interaction = gl_interaction,
                 poly2 = gl_poly2,
                 poly4 = gl_poly4)
sl <- make_learner(Lrnr_sl, learners)
sl_fit <- sl$train(task)
```

`Lrnr_sl` applies cross-validation to all the learners, so we can get CV risk estimates, similar to those we generated by hand with origami.

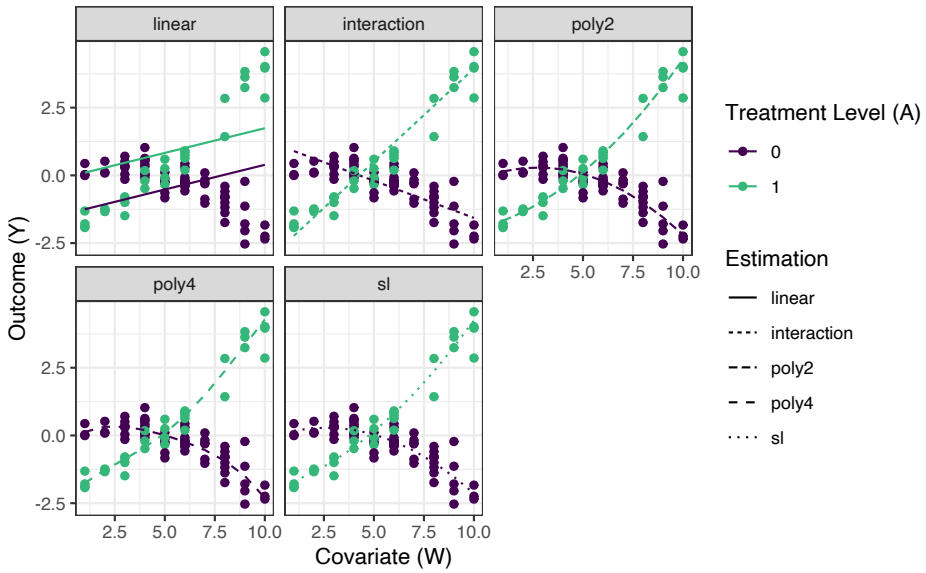
```
sl_fit$cv_risk(loss_squared_error)
        learner coefficients    risk       se  fold_sd fold_min_risk
1:       linear    0.0069332 2.06255 0.250298 0.831859      0.729272
2:  interaction    0.0503607 0.28348 0.037538 0.090305      0.120734
3:        poly2    0.9427061 0.18797 0.036528 0.088623      0.060315
4:        poly4    0.0000000 0.20638 0.039118 0.092897      0.078726
5: SuperLearner           NA 0.18753 0.035458 0.083596      0.061988
   fold_max_risk
1:       3.33138
2:       0.41529
3:       0.32614
4:       0.32564
5:       0.31375
```

```
data(schematic_grid, package="tlverse")
grid_task <- make_sl3_Task(schematic_grid,
                           covariates=c("A","W"),
                           outcome="Y")

schematic_grid$sl <- sl_fit$fit_object$full_fit$predict(grid_task)
library_grid_preds <- sl_fit$fit_object$full_fit$fit_object$learner_fits[[1]]$predict(g
sl_grid <- cbind(schematic_grid,library_grid_preds)
long <- melt(sl_grid, id=c("W","A"),
             measure=c("sl",names(learners)),
             variable.name = "type",
             value.name = "Y")

# TODO: why doesn't this control facet ordering?
```

```
long[,type:=factor(type, levels = c(names(learners),"sl"))]
schematic[,type:=NULL]
data(schematic_meta, package="tlverse")
tlverse::plot_schematic(schematic, long, type="facet")
```



6.4 Conceptual

6.5 washb example

6.6 advanced usage

7

The TMLE Framework

7.1 Introduction

7.1.1 Roadmap Review

7.1.2 We want to estimate psi better

7.1.3 We also want inference

7.2 Learning Objectives

By the end of this chapter, you will be able to

1. Understand why we use TMLE for effect estimation.
2. Use `tmle3` to estimate an Average Treatment Effect (ATE).
3. Understand how to use `tmle3` "Specs" objects.
4. Fit `tmle3` for a custom set of target parameters.
5. Use the delta method to estimate transformations of target parameters.

7.3 Setup

```
library(sl3)
library(tmle3)
library(ggplot2)
```

7.4 Schematic Example

We'll load the data like before:

```
data(schematic, package="tlverse")
```

We'll need some learners like before:

```
gl <- make_learner(Lrnr_glm)
ml <- make_learner(Lrnr_mean)
gl_interaction <- make_learner(Lrnr_glm,
                               formula = "~ A*W")
gl_poly2 <- make_learner(Lrnr_glm,
                         formula = "~ A*(W + I(W^2))")
gl_poly4 <- make_learner(Lrnr_glm,
                         formula = "~ A*(W + I(W^2) + I(W^3) + I (W^4))")

Y_learners <- list(linear = gl,
                   interaction = gl_interaction,
                   poly2 = gl_poly2,
                   poly4 = gl_poly4)
sl_Y <- make_learner(Lrnr_sl, Y_learners)

A_learners <- list(mean = ml,
                   linear = gl)
sl_A <- make_learner(Lrnr_sl, A_learners)

learner_list <- list(A = sl_A, Y = sl_Y)

ate_spec <- tmle_ATE(treatment_level = 1,
                     control_level = 0)

node_list <- list(W = "W", A = "A", Y = "Y")
tmle3_fit <- tmle3(ate_spec, schematic, node_list, learner_list)

print(tmle3_fit)
A tmle3_Fit that took 1 step(s)
   type                 param init_est tmle_est      se   lower  upper
1:  ATE ATE[Y_{A=1}-Y_{A=0}]   1.3193   1.3036 0.28075 0.75333 1.8538
   psi_transformed lower_transformed upper_transformed
1:          1.3036           0.75333            1.8538
```

7.5 Conceptual

7.6 washb example

7.7 advanced usage

8

A Primer on the R6 Class System

A central goal of the Targeted Learning statistical paradigm is to estimate scientifically relevant parameters in realistic (usually nonparametric) models.

The `tlverse` is designed using basic OOP principles and the `R6` OOP framework. While we've tried to make it easy to use the `tlverse` packages without worrying much about OOP, it is helpful to have some intuition about how the `tlverse` is structured. Here, we briefly outline some key concepts from OOP. Readers familiar with OOP basics are invited to skip this section.

8.1 Classes, Fields, and Methods

The key concept of OOP is that of an object, a collection of data and functions that corresponds to some conceptual unit. Objects have two main types of elements:

1. *fields*, which can be thought of as nouns, are information about an object, and
2. *methods*, which can be thought of as verbs, are actions an object can perform.

Objects are members of classes, which define what those specific fields and methods are. Classes can inherit elements from other classes (sometimes called base classes) – accordingly, classes that are similar, but not exactly the same, can share some parts of their definitions.

Many different implementations of OOP exist, with variations in how these concepts are implemented and used. R has several different implementations, including `S3`, `S4`, reference classes, and `R6`. The `tlverse` uses the `R6` implementation. In `R6`, methods and fields of a class object are accessed using the `$` operator. For a more thorough introduction to `R`'s various OOP systems, see `http://adv-r.had.co.nz/OO-essentials.html`, from Hadley Wickham's *Advanced R* (Wickham, 2014).

8.2 Object Oriented Programming: Python and R

OO concepts (classes with inherentence) were baked into Python from the first published version (version 0.9 in 1991). In contrast, R gets its OO "approach" from its predecessor, S, first released in 1976. For the first 15 years, S had no support for classes, then, suddenly, S got two OO frameworks bolted on in rapid succession: informal classes with S3 in 1991, and formal classes with S4 in 1998. This process continues, with new OO frameworks being periodically released, to try to improve the lackluster OO support in R, with reference classes (R5, 2010) and R6 (2014). Of these, R6 behaves most like Python classes (and also most like OOP focused languages like C++ and Java), including having method definitions be part of class definitions, and allowing objects to be modified by reference.

Bibliography

Baker, M. (2016). Is there a reproducibility crisis? a nature survey lifts the lid on how researchers view the crisis rocking science and what they think will help. *Nature*, 533(7604):452–455.

Buckheit, J. B. and Donoho, D. L. (1995). Wavelab and reproducible research. In *Wavelets and statistics*, pages 55–81. Springer.

Coyle, J. R., Hejazi, N. S., Malenica, I., Phillips, R. V., Arnold, B. F., Mertens, A., Benjamin-Chung, J., Cai, W., Dayal, S., Colford Jr., J. M., Hubbard, A. E., and van der Laan, M. J. (2021). Targeting Learning: Robust statistics for reproducible research. *arXiv*.

Munafò, M. R., Nosek, B. A., Bishop, D. V., Button, K. S., Chambers, C. D., Du Sert, N. P., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., and Ioannidis, J. P. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, 1(1):0021.

Nature Editorial (Anonymous) (2015a). How scientists fool themselves — and how they can stop. *Nature*, 526(7572).

Nature Editorial (Anonymous) (2015b). Let's think about cognitive bias. *Nature*, 526(7572).

Nosek, B. A., Ebersole, C. R., DeHaven, A. C., and Mellor, D. T. (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606.

Peng, R. (2015). The reproducibility crisis in science: A statistical counterattack. *Significance*, 12(3):30–32.

Pullenayegum, E. M., Platt, R. W., Barwick, M., Feldman, B. M., Offringa, M., and Thabane, L. (2016). Knowledge translation in biostatistics: a survey of current practices, preferences, and barriers to the dissemination and uptake of new statistical methods. *Statistics in medicine*, 35(6):805–818.

R Core Team (2021). R: A language and environment for statistical computing.

Sandercock, P., Collins, R., Counsell, C., Farrell, B., Peto, R., Slattery, J., and Warlow, C. (1997). The international stroke trial (ist): a randomized trial of aspirin, subcutaneous heparin, both, or neither among 19,435 patients with acute ischemic stroke. *Lancet*, 349(9065):1569–1581.

Sandercock, P. A., Niewada, M., and Członkowska, A. (2011). The international stroke trial database. *Trials*, 12(1):101.

Stark, P. B. and Saltelli, A. (2018). Cargo-cult statistics and scientific crisis. *Significance*, 15(4):40–43.

Stromberg, A. et al. (2004). Why write statistical software? the case of robust statistical methods. *Journal of Statistical Software*, 10(5):1–8.

Szucs, D. and Ioannidis, J. (2017). When null hypothesis significance testing is unsuitable for research: a reassessment. *Frontiers in Human Neuroscience*, 11:390.

Tofail, F., Fernald, L. C., Das, K. K., Rahman, M., Ahmed, T., Jannat, K. K., Unicomb, L., Arnold, B. F., Ashraf, S., Winch, P. J., et al. (2018). Effect of water quality, sanitation, hand washing, and nutritional interventions on child development in rural bangladesh (wash benefits bangladesh): a cluster-randomised controlled trial. *The Lancet Child & Adolescent Health*, 2(4):255–268.

van der Laan, M. J. and Rose, S. (2011). *Targeted learning: causal inference for observational and experimental data.* Springer Science & Business Media.

van der Laan, M. J. and Rose, S. (2018). *Targeted Learning in Data Science: Causal Inference for Complex Longitudinal Studies.* Springer Science & Business Media.

van der Laan, M. J. and Starmans, R. J. (2014). Entering the era of data science: Targeted learning and the integration of statistics and computational data analysis. *Advances in Statistics*, 2014.

Wickham, H. (2014). *Advanced r.* Chapman and Hall/CRC.