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

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; her research integrates personalized medicine, human-computer interaction, experimental design, and regulatory policy.

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."
>
> — Anonymous (2015)

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

> "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, at its current state the culture of data analysis and statistics actually enables human bias through improper model selection. All hypothesis tests and estimators are derived from statistical models, so to obtain valid estimates and inference it is critical that the statistical model contains the process that generated the data. Perhaps treatment was randomized or only depended on a small number of baseline covariates; this knowledge should and can be incorporated in the model. Alternatively, maybe the data is observational, and there is no knowledge about the data-generating process (DGP). If this is the case, then the statistical model should contain *all* data distributions. In practice; however, models are not selected based on knowledge of the DGP, instead models are often selected based on (1) the p-values they yield, (2) their convenience of implementation, and/or (3) an analysts loyalty to a particular 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 (Stark and Saltelli, 2018). This presents a fundamental drive behind the epidemic of false findings that scientific research is suffering from (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 targets the scientific question at hand and is counter to the current culture of “convenience statistics” which opens the door to biased estimation, misleading results, and false discoveries. Targeted Learning restores the fundamentals that formalized the field of statistics, such as the that facts that a statistical model represents real knowledge about the experiment that generated the data, and 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 inhibiting these all-too-human biases (e.g., hindsight bias, confirmation bias, and outcome bias) from infiltrating analysis.

> “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)

The objective for this handbook is to provide training to students, researchers, industry professionals, faculty in science, public health, statistics, and other fields to empower them with the necessary knowledge and skills to utilize the sound methodology of Targeted Learning — a technique that provides tailored pre-specified machines for answering queries, so that each data analysis is completely reproducible, and estimators are efficient, minimally biased, and provide formal statistical inference.

Just as the conscientious use of modern statistical methodology is necessary to ensure that scientific practice thrives, it remains critical to acknowledge the role that robust software plays in allowing practitioners direct access to published results. We recall that "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 aspect of science (Buckheit and Donoho, 1995).

For a statistical methodology to be readily accessible in practice, it is crucial that it is accompanied by robust user-friendly software (Pullenayegum et al., 2016; Stromberg et al., 2004). The `tlverse` software ecosystem was developed to fulfill this need for the Targeted Learning methodology. Not only does this software facilitate computationally reproducible and efficient analyses, it is also a tool for Targeted Learning education since its workflow mirrors that of the methodology. In particular, the `tlverse` paradigm 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 `R` 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` `R` packages share a core set of design principles centered on extensibility, allowing for them to be used in conjunction with each other and built upon one other in a cohesive fashion. For an introduction to Targeted Learning, we recommend the 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 intuitive explanation readers will build 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

The Roadmap for Targeted Learning

Learning Objectives

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

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.

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.

2.1 The Roadmap

Following the roadmap is a process of five stages.

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

(1) Data: A random variable with a probability distribution, $O \sim P_0$

The data set we're confronted with is the result of an experiment and we can view the data as a random variable, O, because if we repeat the experiment we would have a different realization of this experiment. In particular, if we repeat the experiment many times we could learn the probability distribution, P_0, of our data. So, the observed data O with probability distribution P_0 are n independent identically distributed (i.i.d.) observations of the random variable $O; O_1, \ldots, O_n$. Note that while not all data are i.i.d., there are ways to handle non-i.i.d. data, such as establishing conditional independence, stratifying data to create sets of identically distributed data, etc. It is crucial that researchers be absolutely clear about what they actually know about the data-generating distribution for a given problem of interest. Unfortunately, communication between statisticians and researchers is often fraught with misinterpretation. The roadmap provides a mechanism by which to ensure clear communication between research and statistician – it truly helps with this communication!

The empirical probability measure, P_n

Once we have n of such i.i.d. observations we have an empirical probability measure, P_n. The empirical probability measure is an approximation of the true probability measure P_0, allowing us to learn from our data. For example, we can define the empirical probability measure of a set, A, to be the proportion of observations which end up in A. That is,

$$P_n(A) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(O_i \in A)$$

In order to start learning something, we need to ask *"What do we know about the probability distribution of the data?"* This brings us to Step 2.

(2) The statistical model $\mathcal{M}$ such that $P_0 \in \mathcal{M}$

The statistical model $\mathcal{M}$ is defined by the question we asked at the end of Step 1. It is defined as the set of possible probability distributions for our observed data. Often $\mathcal{M}$ is very large (possibly infinite-dimensional), to reflect the fact that statistical knowledge is limited. In the case that $\mathcal{M}$ is infinite-dimensional, we deem this a nonparametric statistical model.

Alternatively, if the probability distribution of the data at hand is described by a finite number of parameters, then the statistical model is parametric. In this case, we subscribe to the belief that the random variable O being observed has, for example, a normal distribution with mean μ and variance σ^2. Formally, a parametric model may be defined

$$\mathcal{M} = \{P_\theta : \theta \in \mathbb{R}^d\}$$

Sadly, the assumption that the data-generating distribution has a specific, parametric form is all too common, especially since this is a leap of faith or an assumption made of convenience. This practice of oversimplification in the current culture of data analysis typically derails any attempt at trying to answer the scientific question at hand; alas, such statements as the ever-popular quip of Box that "All models are wrong but some are useful" encourage the data analyst to make arbitrary choices even when such a practice often forces starkly different answers to the same estimation problem. The Targeted Learning paradigm does not suffer from this bias since it defines the statistical model through a representation of the true data-generating distribution corresponding to the observed data.

Now, on to Step 3: *"What are we trying to learn from the data?"*

(3) The statistical target parameter Ψ and estimand $\Psi(P_0)$

The statistical target parameter, Ψ, is defined as a mapping from the statistical model, $\mathcal{M}$, to the parameter space (i.e., a real number) $\mathbb{R}$. That is, $\Psi : \mathcal{M} \to \mathbb{R}$. The estimand may be seen as a representation of the quantity that we wish to learn from the data, the answer to a well-specified (often causal) question of interest. In contrast to purely statistical estimands, causal estimands require *identification from the observed data*, based on causal models that include several untestable assumptions, described in more detail in the section on causal target parameters.

For a simple example, consider a data set which contains observations of a survival time on every subject, for which our question of interest is "What's the probability that someone lives longer than five years?" We have,

$$\Psi(P_0) = \mathbb{P}(O > 5)$$

This answer to this question is the **estimand,** $\Psi(P_0)$, which is the quantity we're trying to learn from the data. Once we have defined O, $\mathcal{M}$ and $\Psi(P_0)$ we have formally defined the statistical estimation problem.

(4) The estimator $\hat{\Psi}$ and estimate $\hat{\Psi}(P_n)$

To obtain a good approximation of the estimand, we need an estimator, an *a priori*-specified algorithm defined as a mapping from the set of possible empirical distributions, P_n, which live in a non-parametric statistical model, $\mathcal{M}_{NP}$ ($P_n \in \mathcal{M}_{NP}$), to the parameter space of the parameter of interest. That is, $\hat{\Psi} : \mathcal{M}_{NP} \to \mathbb{R}^d$. The estimator is a function that takes as input the observed data, a realization of P_n, and gives as output a value in the parameter space, which is the **estimate,** $\hat{\Psi}(P_n)$.

Where the estimator may be seen as an operator that maps the observed data and corresponding empirical distribution to a value in the parameter space, the numerical output that produced such a function is the estimate. Thus, it is an element of the parameter space based on the empirical probability distribution of the observed data. If we plug in a realization of P_n (based on a sample size n of the random variable O), we get back an estimate $\hat{\Psi}(P_n)$ of the true parameter value $\Psi(P_0)$.

In order to quantify the uncertainty in our estimate of the target parameter (i.e., to construct statistical inference), an understanding of the sampling distribution of our estimator will be necessary. This brings us to Step 5.

(5) A measure of uncertainty for the estimate $\hat{\Psi}(P_n)$

Since the estimator $\hat{\Psi}$ is a function of the empirical distribution P_n, the estimator itself is a random variable with a sampling distribution. So, if we repeat the experiment of drawing n observations we would every time end up with a different realization of our estimate and our estimator has a sampling distribution. The sampling distribution of some estimators can be theoretically validated to be approximately normally distributed by a Central Limit Theorem (CLT).

A **Central Limit Theorem** (CLTs) is a statement regarding the convergence of the **sampling distribution of an estimator** to a normal distribution. In general, we will construct estimators whose limit sampling distributions may be shown to be approximately normal distributed as sample size increases. For large enough n we have,

$$\hat{\Psi}(P_n) \sim N\left(\Psi(P_0), \frac{\sigma^2}{n}\right),$$

permitting statistical inference. Now, we can proceed to quantify the uncertainty of our chosen estimator by construction of hypothesis tests and confidence intervals. For example, we may construct a confidence interval at level $(1-\alpha)$ for our estimand, $\Psi(P_0)$:

$$\hat{\Psi}(P_n) \pm z_{1-\frac{\alpha}{2}}\left(\frac{\sigma}{\sqrt{n}}\right),$$

where $z_{1-\frac{\alpha}{2}}$ is the $(1-\frac{\alpha}{2})^{\text{th}}$ quantile of the standard normal distribution. Often, we will be interested in constructing 95% confidence intervals, corresponding to mass $\alpha = 0.05$ in either tail of the limit distribution; thus, we will typically take $z_{1-\frac{\alpha}{2}} \approx 1.96$.

Note: we will typically have to estimate the standard error, $\frac{\sigma}{\sqrt{n}}$.

A 95% confidence interval means that if we were to take 100 different samples of size n and compute a 95% confidence interval for each sample, then approximately 95 of the 100 confidence intervals would contain the estimand, $\Psi(P_0)$. More practically, this means that there is a 95% probability that the confidence interval procedure generates intervals containing the true estimand value (or 95% confidence of "covering" the true value). That is, any single estimated confidence interval either will contain the true estimand or will not (also called "coverage").

2.2 Summary of the Roadmap

Data, O, is viewed as a random variable that has a probability distribution. We often have n units of independent identically distributed units with probability distribution P_0, such that $O_1, \ldots, O_n \sim P_0$. We have statistical knowledge about the experiment that generated this data. In other words, we make a statement that the true data distribution P_0 falls in a certain set called a statistical model, $\mathcal{M}$. Often these sets are very large because statistical knowledge is very limited - hence, these statistical models are often infinite dimensional models. Our statistical query is, "What are we trying to learn from the data?" denoted by the statistical target parameter, Ψ, which maps the P_0 into the estimand, $\Psi(P_0)$. At this point the statistical estimation problem is formally defined and now we will need statistical theory to guide us in the construction of estimators. There's a lot of statistical theory we will review in this course that, in particular, relies on the Central Limit Theorem, allowing us to come up with estimators that are approximately normally distributed and also allowing us to come with statistical inference (i.e., confidence intervals and hypothesis tests).

2.3 Causal Target Parameters

In many cases, we are interested in problems that ask questions regarding the effect of an intervention on a future outcome of interest. These questions can be represented as causal estimands.

The Causal Model

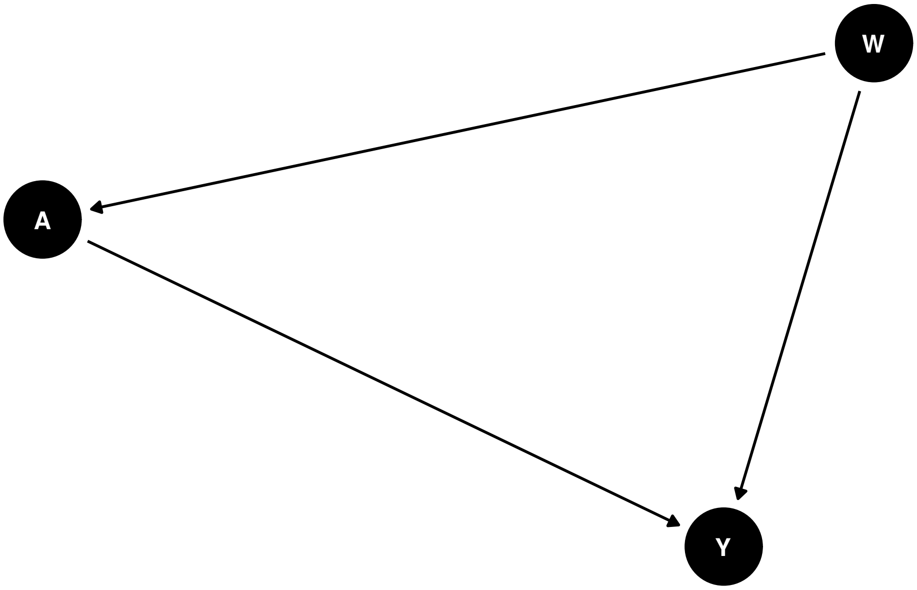
After formalizing the data and the statistical model, we can define a causal model to express causal parameters of interest. Directed acyclic graphs (DAGs) are one useful tool to express what we know about the causal relations among variables. Ignoring exogenous U terms (explained below), we assume the following ordering of the variables in the observed data O. We do this below using `DAGitty` (Textor et al., 2011):

```
library(dagitty)
library(ggdag)

# make DAG by specifying dependence structure
dag <- dagitty(
  "dag {
    W -> A
    W -> Y
    A -> Y
    W -> A -> Y
  }"
)
exposures(dag) <- c("A")
outcomes(dag) <- c("Y")
```

```r
tidy_dag <- tidy_dagitty(dag)

# visualize DAG
ggdag(tidy_dag) +
  theme_dag()
```



While directed acyclic graphs (DAGs) like above provide a convenient means by which to visualize causal relations between variables, the same causal relations among variables can be represented via a set of structural equations, which define the non-parametric structural equation model (NPSEM):

$$\begin{aligned} W &= f_W(U_W) \\ A &= f_A(W, U_A) \\ Y &= f_Y(W, A, U_Y), \end{aligned}$$

where U_W, U_A, and U_Y represent the unmeasured exogenous background characteristics that influence the value of each variable. In the NPSEM, f_W, f_A and f_Y denote that each variable (for W, A and Y, respectively) is a function of its parents and unmeasured background characteristics, but note that there is no imposition of any particular functional constraints(e.g., linear, logit-linear, only one interaction, etc.). For this reason, they are called non-parametric structural equation models (NPSEMs). The DAG and set of nonparametric structural equations represent exactly the same information and so may be used interchangeably.

The first hypothetical experiment we will consider is assigning exposure to the whole population and observing the outcome, and then assigning no exposure to the whole population and observing the outcome. On the nonparametric structural equations, this corresponds to a comparison of the outcome distribution in the population under two interventions:

1. A is set to 1 for all individuals, and
2. A is set to 0 for all individuals.

These interventions imply two new nonparametric structural equation models. For the case $A = 1$, we have

$$\begin{aligned} W &= f_W(U_W) \\ A &= 1 \\ Y(1) &= f_Y(W, 1, U_Y), \end{aligned}$$

and for the case $A = 0$,

$$\begin{aligned} W &= f_W(U_W) \\ A &= 0 \\ Y(0) &= f_Y(W, 0, U_Y). \end{aligned}$$

In these equations, A is no longer a function of W because we have intervened on the system, setting A deterministically to either of the values 1 or 0. The new symbols $Y(1)$ and $Y(0)$ indicate the outcome variable in our population if it were generated by the respective NPSEMs above; these are often called *counterfactuals* (since they run contrary-to-fact). The difference between the means of the outcome under these two interventions defines a parameter that is often called the “average treatment effect” (ATE), denoted

$$ATE = \mathbb{E}_X(Y(1) - Y(0)), \tag{2.1}$$

where $\mathbb{E}_X$ is the mean under the theoretical (unobserved) full data $X = (W, Y(1), Y(0))$.

Note, we can define much more complicated interventions on NPSEM’s, such as interventions based upon rules (themselves based upon covariates), stochastic rules, etc. and each results in a different targeted parameter and entails different identifiability assumptions discussed below.

Identifiability

Because we can never observe both $Y(0)$ (the counterfactual outcome when $A = 0$) and $Y(1)$ (similarly, the counterfactual outcome when $A = 1$), we cannot estimate the quantity in Equation (2.1) directly. Instead, we have to make assumptions under which this quantity may be estimated from the observed data $O \sim P_0$ under the data-generating distribution P_0. Fortunately, given the causal model specified in the NPSEM above, we can, with a handful of untestable assumptions, estimate the ATE, even from observational data. These assumptions may be summarized as follows.

1. The causal graph implies $Y(a) \perp A$ for all $a \in \mathcal{A}$, which is the *randomization* assumption. In the case of observational data, the analogous assumption is *strong ignorability* or *no unmeasured confounding* $Y(a) \perp A \mid W$ for all $a \in \mathcal{A}$;
2. Although not represented in the causal graph, also required is the assumption of no interference between units, that is, the outcome for unit i Y_i is not affected by exposure for unit j A_j unless $i = j$;
3. *Consistency* of the treatment mechanism is also required, i.e., the outcome for unit i is $Y_i(a)$ whenever $A_i = a$, an assumption also known as "no other versions of treatment";
4. It is also necessary that all observed units, across strata defined by W, have a bounded (non-deterministic) probability of receiving treatment – that is, $0 < \mathbb{P}(A = a \mid W) < 1$ for all a and W). This assumption is referred to as *positivity* or *overlap*.

Remark: Together, (2) and (3), the assumptions of no interference and consistency, respectively, are jointly referred to as the *stable unit treatment value assumption* (SUTVA).

Given these assumptions, the ATE may be re-written as a function of P_0, specifically

$$ATE = \mathbb{E}_0(Y(1) - Y(0)) = \mathbb{E}_0\left(\mathbb{E}_0[Y \mid A = 1, W] - \mathbb{E}_0[Y \mid A = 0, W]\right). \tag{2.2}$$

In words, the ATE is the difference in the predicted outcome values for each subject, under the contrast of treatment conditions ($A = 0$ versus $A = 1$), in the population, averaged over all observations. Thus, a parameter of a theoretical "full" data distribution can be represented as an estimand of the observed data distribution. Significantly, there is nothing about the representation in Equation (2.2) that requires parameteric assumptions; thus, the regressions on the right hand side may be estimated freely with machine learning. With different parameters, there will be potentially different identifiability assumptions and the resulting estimands can be functions of different components of P_0. We discuss several more complex estimands in later sections of this handbook.

3

Welcome to the `tlverse`

Learning Objectives

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

What is the `tlverse`?

The `tlverse` is a new framework for doing Targeted Learning in R, inspired by the `tidyverse` ecosystem of R packages.

By analogy to the `tidyverse`:

> 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` is

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

Anatomy of the tlverse

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

3.1 Installation

The tlverse ecosystem of packages are currently hosted at https://github.com/tlverse, not yet on CRAN. You can use the usethis package to install them:

```
install.packages("devtools")
devtools::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 (PAT).
2. Create a PAT simply by clicking "Generate token" at the bottom of the page.
3. Copy your PAT, 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.

    a. If your `.Renviron` file does not pop-up after calling `usethis::edit_r_environ()`; then try inputting `Sys.setenv(GITHUB_PAT = "yourPAT")`, replacing your PAT with inside the quotes. If this does not error, then skip to step 8.

5. In your `.Renviron` file, type `GITHUB_PAT=` and then paste your PAT after the equals symbol with no space.
6. In your `.Renviron` file, press the enter key to ensure that your `.Renviron` ends with a new line.
7. Save your `.Renviron` file. The example below shows how this syntax should look.

```
GITHUB_PAT <- yourPAT
```

8. Restart R. You can restart R via the drop-down menu on RStudio's "Session" tab, which is located at the top of the RStudio interface. You have to restart R for the changes to take effect!

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

4

Meet the Data

4.1 WASH Benefits Example Dataset

The 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). 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). 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.

In the handbook, we concentrate on child growth (size for age) as the outcome of interest. For reference, this trial was registered with ClinicalTrials.gov as NCT01590095.

```r
library(readr)
# read in data via readr::read_csv
dat <- read_csv(
  paste0(
    "https://raw.githubusercontent.com/tlverse/tlverse-data/master/",
    "wash-benefits/washb_data.csv"
  )
)
```

For the purposes of this handbook, we start by treating the data as independent and identically distributed (i.i.d.) random draws from a very large target population. We could, with available options, account for the clustering of the data (within sampled geographic units), but, for simplification, we avoid these details in the handbook, although modifications of our methodology for biased samples, repeated measures, and related complications, are available.

We have 28 variables measured, of which a single variable is set to be the outcome of interest. 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*. This results in our observed data structure being n i.i.d. copies of $O_i = (W_i, A_i, Y_i)$, for $i = 1, \ldots, n$.

Using the `skimr` package, we can quickly summarize the variables measured in the WASH Benefits data set:

skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	character.empty
character	tr	0	1.00000	3	15	0
character	fracode	0	1.00000	2	6	0
character	sex	0	1.00000	4	6	0
character	momedu	0	1.00000	12	15	0
character	hfiacat	0	1.00000	11	24	0
numeric	whz	0	1.00000	NA	NA	NA
numeric	month	0	1.00000	NA	NA	NA
numeric	aged	0	1.00000	NA	NA	NA
numeric	momage	18	0.99617	NA	NA	NA
numeric	momheight	31	0.99340	NA	NA	NA
numeric	Nlt18	0	1.00000	NA	NA	NA
numeric	Ncomp	0	1.00000	NA	NA	NA
numeric	watmin	0	1.00000	NA	NA	NA
numeric	elec	0	1.00000	NA	NA	NA
numeric	floor	0	1.00000	NA	NA	NA
numeric	walls	0	1.00000	NA	NA	NA
numeric	roof	0	1.00000	NA	NA	NA
numeric	asset_wardrobe	0	1.00000	NA	NA	NA
numeric	asset_table	0	1.00000	NA	NA	NA
numeric	asset_chair	0	1.00000	NA	NA	NA
numeric	asset_khat	0	1.00000	NA	NA	NA
numeric	asset_chouki	0	1.00000	NA	NA	NA
numeric	asset_tv	0	1.00000	NA	NA	NA
numeric	asset_refrig	0	1.00000	NA	NA	NA
numeric	asset_bike	0	1.00000	NA	NA	NA
numeric	asset_moto	0	1.00000	NA	NA	NA
numeric	asset_sewmach	0	1.00000	NA	NA	NA
numeric	asset_mobile	0	1.00000	NA	NA	NA

A convenient summary of the relevant variables is given just above, complete with a small visualization describing the marginal characteristics of each covariate. Note that the *asset* variables reflect socio-economic status of the study participants. Notice also the uniform distribution of the treatment groups (with twice as many controls); this is, of course, by design.

4.2 International Stroke Trial Example Dataset

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.

```
# read in data
ist <- read_csv(
  paste0(
    "https://raw.githubusercontent.com/tlverse/tlverse-handbook/master/",
    "data/ist_sample.csv"
  )
)
```

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. In this data, the outcome is occasionally missing, but there is no need to create a variable indicating this missingness (such as Δ) for analyses in the `tlverse`, since it is automatically detected when `NA` are present in the outcome. This observed data structure can be denoted as n i.i.d. copies of $O_i = (W_i, A_i, \Delta_i, \Delta Y_i)$, for $i = 1, \ldots, n$, where Δ denotes the binary indicator that the outcome is observed.

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

skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	char
character	RCONSC	0	1.000	1	1	
character	SEX	0	1.000	1	1	
character	RSLEEP	0	1.000	1	1	
character	RATRIAL	0	1.000	1	1	
character	RCT	0	1.000	1	1	
character	RVISINF	0	1.000	1	1	
character	RHEP24	0	1.000	1	1	
character	RASP3	0	1.000	1	1	
character	RDEF1	0	1.000	1	1	
character	RDEF2	0	1.000	1	1	
character	RDEF3	0	1.000	1	1	
character	RDEF4	0	1.000	1	1	
character	RDEF5	0	1.000	1	1	
character	RDEF6	0	1.000	1	1	
character	RDEF7	0	1.000	1	1	
character	RDEF8	0	1.000	1	1	
character	STYPE	0	1.000	3	4	
character	RXHEP	0	1.000	1	1	
character	REGION	0	1.000	10	26	
numeric	RDELAY	0	1.000	NA	NA	
numeric	AGE	0	1.000	NA	NA	
numeric	RSBP	0	1.000	NA	NA	
numeric	MISSING_RATRIAL_RASP3	0	1.000	NA	NA	
numeric	MISSING_RHEP24	0	1.000	NA	NA	
numeric	RXASP	0	1.000	NA	NA	
numeric	DRSISC	10	0.998	NA	NA	

4.3 NHANES I Epidemiologic Follow-up Study (NHEFS)

This data is from the National Health and Nutrition Examination Survey (NHANES) Data I Epidemiologic Follow-up Study. More coming soon.

```
# read in data
nhefs_data <- read_csv(
  paste0(
    "https://raw.githubusercontent.com/tlverse/tlverse-handbook/master/",
    "data/NHEFS.csv"
  )
)
```

A snapshot of the data set is shown below:

skim_type	skim_variable	n_missing	complete_rate	numeric.mean	numeric.sd	numeric.p0	nume
numeric	seqn	0	1.00000	16552.36464	7498.91820	233.00000	10607
numeric	qsmk	0	1.00000	0.26274	0.44026	0.00000	0
numeric	death	0	1.00000	0.19521	0.39649	0.00000	0
numeric	yrdth	1311	0.19521	87.56918	2.65941	83.00000	85
numeric	modth	1307	0.19767	6.25776	3.61530	1.00000	3
numeric	dadth	1307	0.19767	15.87267	8.90549	1.00000	8
numeric	sbp	77	0.95273	128.70941	19.05156	87.00000	116
numeric	dbp	81	0.95028	77.74483	10.63486	47.00000	70
numeric	sex	0	1.00000	0.50952	0.50006	0.00000	0
numeric	age	0	1.00000	43.91529	12.17043	25.00000	33
numeric	race	0	1.00000	0.13198	0.33858	0.00000	0
numeric	income	62	0.96194	17.94767	2.66328	11.00000	17
numeric	marital	0	1.00000	2.50338	1.08237	2.00000	2
numeric	school	0	1.00000	11.13505	3.08960	0.00000	10
numeric	education	0	1.00000	2.70350	1.19010	1.00000	2
numeric	ht	0	1.00000	168.74096	9.05313	142.87500	161
numeric	wt71	0	1.00000	71.05213	15.72959	36.17000	59
numeric	wt82	63	0.96133	73.46922	16.15805	35.38020	61
numeric	wt82_71	63	0.96133	2.63830	7.87991	-41.28047	-1
numeric	birthplace	92	0.94352	31.59532	14.50050	1.00000	22
numeric	smokeintensity	0	1.00000	20.55126	11.80375	1.00000	10
numeric	smkintensity82_71	0	1.00000	-4.73788	13.74136	-80.00000	-10
numeric	smokeyrs	0	1.00000	24.87109	12.19807	1.00000	15
numeric	asthma	0	1.00000	0.04850	0.21488	0.00000	0
numeric	bronch	0	1.00000	0.08533	0.27946	0.00000	0
numeric	tb	0	1.00000	0.01412	0.11802	0.00000	0
numeric	hf	0	1.00000	0.00491	0.06993	0.00000	0
numeric	hbp	0	1.00000	1.05095	0.95821	0.00000	0
numeric	pepticulcer	0	1.00000	0.10374	0.30502	0.00000	0
numeric	colitis	0	1.00000	0.03376	0.18067	0.00000	0
numeric	hepatitis	0	1.00000	0.01719	0.13001	0.00000	0
numeric	chroniccough	0	1.00000	0.05402	0.22613	0.00000	0
numeric	hayfever	0	1.00000	0.08963	0.28573	0.00000	0
numeric	diabetes	0	1.00000	0.97974	0.99579	0.00000	0
numeric	polio	0	1.00000	0.01412	0.11802	0.00000	0
numeric	tumor	0	1.00000	0.02333	0.15099	0.00000	0
numeric	nervousbreak	0	1.00000	0.02885	0.16744	0.00000	0
numeric	alcoholpy	0	1.00000	0.87600	0.33887	0.00000	1
numeric	alcoholfreq	0	1.00000	1.92020	1.30714	0.00000	1
numeric	alcoholtype	0	1.00000	2.47575	1.20816	1.00000	1
numeric	alcoholhowmuch	417	0.74401	3.28713	2.98470	1.00000	2
numeric	pica	0	1.00000	0.97545	0.99785	0.00000	0
numeric	headache	0	1.00000	0.62983	0.48300	0.00000	0
numeric	otherpain	0	1.00000	0.24616	0.43091	0.00000	0
numeric	weakheart	0	1.00000	0.02210	0.14705	0.00000	0
numeric	allergies	0	1.00000	0.06200	0.24123	0.00000	0
numeric	nerves	0	1.00000	0.14426	0.35146	0.00000	0

5

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.

5.1 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).

5.2 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

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

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*.

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

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.

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.

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.

Textor, J., Hardt, J., and Knüppel, S. (2011). Dagitty: a graphical tool for analyzing causal diagrams. *Epidemiology*, 22(5):745.

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.