pstratreg: An R package

Ian Lundberg and Soonhong Cho

2025-08-26

Contents

Welcome!

> **Warning.** This website documents an R package that is in development. Expect changes. We encourage you to try the package and email us with suggestions.[1]. You can also see our **draft** and **slides** from a presentation of this project.

This website documents the **pstratreg** package for R. The package helps study causal effects when some treatment values cause outcomes to be non-existent for some units.

Possible applications include

- a labor market intervention where the outcome is hourly wage, but some people are unemployed
- a medical intervention where the outcome is a health metric, but some people die before it is measured
- a sociological study where the outcome involves one's spouse, but some people divorce or never marry and thus have no spouse

This package provides regression-based methods for principal stratification that rely on parametric models and are useful in settings where one hopes to adjust for many confounders.

Getting started

To get started, first install R and RStudio. Then install the package.

```
devtools::install_github("ilundberg/pstratreg")
```

Now head on to the next page to see the types of questions this package can help answer.

This project is joint work between Ian Lundberg (UCLA, ianlundberg@ucla.edu) and Soonhong Cho (UCLA, tnsehdtm@gmail.com).

[1]Ian Lundberg (ianlundberg@ucla.edu) and Soonhong Cho (soonhongcho@g.ucla.edu)

Chapter 1

The goal

Average causal effects are undefined when some outcomes do not exist. For example, consider the two people below who were eligible for a job training intervention.

Person	Wage with job training	Wage without job training	Causal effect
Javier	\$30	\$20	\$30-\$20=\$10
William	\$26	??	\$26-??=??
Average	**\$28**	**??**	**??**

A social scientist might study the average causal effect of the intervention on future hourly wages. But without job training, William would not be employed at all. His wage ?? does not exist. As a result, the average causal effect does not exist. Principal stratification is a method to estimate average causal effects in the subpopulation that excludes people like William, for whom the effect is undefined.

> **Where to read more.** For original ideas, see Frangakis & Rubin (2002) and Zhang & Rubin (2003). For a recent introduction, see Miratrix et al. (2018).

This page is a tutorial on the ideas behind principal stratification. The next page discusses the regression setting that is the focus of this package. Subsequent pages show package functionality.

1.1 Defining principal strata

Let M_i be a binary mediator (e.g., employment) that determines whether an outcome exists for unti i. Let Y_i be the outcome, where Y is undefined when $M = 0$. Let $\{M_i^1, M_i^0\}$ and $\{Y_i^1, Y_i^0\}$ be the potential values that the mediator and outcome would take for unit i under job training and under no job training. The table above gave values for $\{Y_i^1, Y_i^0\}$.

Define four **principal strata** of units by the effect that the intervention has on the mediator value

Stratum	Math	English
$S_i =$ Always	$M_i^1 = M_i^0 = 1$	Always employed, regardless of the intervention
$S_i =$ Induced	$M_i^1 > M_i^0$	Induced into employment by the intervention
$S_i =$ Blocked	$M_i^1 < M_i^0$	Blocked from employment by the intervention
$S_i =$ Never	$M_i^1 = M_i^0 = 0$	Never employed, regardless of the intervention

In our example, Javier is a member of the always stratum and William is a member of the induced stratum.

1.2 Target population: The always stratum

The average causal effect $E(Y^1 - Y^0)$ is only defined in the "Always" stratum.

- in the induced stratum, Y^0 is undefined, so the effect is $Y^1 - ??$
- in the blocked stratum, Y^1 is undefined, so the effect is $?? - Y^0$
- in the never stratum, both are undefined, so the effect is $?? - ??$

Our causal estimand is therefore the average causal effect in the always stratum.

$$E(Y^1 - Y^0 \mid S = \text{Always})$$

1.3 Fundamental problem: Strata are latent

Suppose William received job training. We would observe his $26 wage. We would not know that he would not have been employed without job training. If William received job training, then in the observed data we would know that he was either in the always stratum or the induced stratum.

What can be observed is the treatment value D_i and the mediator value M_i. Each combination of these values is a mixture of two principal strata.

	$D_i = 1$	$D_i = 0$
$M_i = 1$	Always and Induced	Always and Blocked
$M_i = 0$	Never and Blocked	Never and Induced

1.4 Solution step 1: Make an assumption

Assumptions can simplify the problem, and can be plausible in some settings. We focus on two versions of one assumption.

Assumption	Math	English
Positive monotonicity	$M_i^1 \geq M_i^0$	No one is blocked
Negative monotonicity	$M_i^1 \leq M_i^0$	No one is induced

In our example, the positive monotonicity assumption may be credible: job training is unlikely to block anyone from employment. In other settings, the negative monotonicity assumption is credible. If the treatment were having a criminal record on one's resume, that treatment might block employment but would be unlikely to induce employment.

Under positive monotonicity, some latent strata become observable.

	$D_i = 1$	$D_i = 0$
$M_i = 1$	Always and Induced	Always
$M_i = 0$	Never	Never and Induced

For example, suppose that Javier was assigned to no job training, and we observed that he was employed. Under positive monotonicity, we would know that Javier was in the always stratum because he was employed without job training, and thus surely would have been employed with job training.

1.5 Solution step 2: Bound the average effect in the always stratum

Suppose we assume positive monotonicity, and we observe data where four people have been randomized to each treatment condition.

In the no-job-training condition, we observe the following four people

Person	Treatment	Wage	Stratum
Jamal	No job training	\$20	Always
Sandra	No job training	\$16	Always
Oscar	No job training	??	Never or Induced
Nancy	No job training	??	Never or Induced

The stratum column is sometimes known by assumption: Jamal and Sandra were employed despite no job training, so they surely would have been employed in a counterfactual world with job training (by positive monotonicity). Oscar and Nancy might or might not have been employed in a counterfactual world where they received job training.

Four people were randomized to the treatment: job training

Person	Treatment	Wage	Stratum
Nia	Job training	\$40	Always or Induced
Steven	Job training	\$32	Always or Induced
Maya	Job training	\$29	Always or Induced
Hugo	Job training	\$25	Always or Induced

The stratum column for the four treated people is always unknown, because under positive monotonicity they might or might not have been employed despite no job training.

By comparing the frequency of ?? in the tables, we can estimate that job training reduces employment by 50 percentage points. This tells us the size of the induced stratum.

$$\hat{P}(S = \text{Induced}) = 0.5$$

Because all those receiving job training were employed, our data also suggest that the never-employed stratum is empty.

$$\hat{P}(S = \text{Never}) = 0$$

Thus, in our example we estimate that 50% of people are always employed and 50% would be induced into employment if exposed to job training. Thus half of {Nia, Steven, Maya, Hugo} are always-employed.

But which half? This is impossible to know. We therefore **bound** the expected outcome under treatment.

- for an *upper bound*, assume the highest-valued half of the treated units are the always-employed ones
  - Nia and Steven are always employed
  - Maya and Hugo are induced
  - $\hat{E}_\text{Upper}(Y^1 \mid S = \text{Always}) = \frac{32+40}{2} = \$36$
- for a *lower bound*, assume the lowest-valued half of the treated units are the always-employed ones
  - Nia and Steven are induced
  - Maya and Hugo are always employed
  - $\hat{E}_\text{Lower}(Y^1 \mid S = \text{Always}) = \frac{25+29}{2} = \$27$

Because under positive monotonicity all untreated individuals must be always-employed, we point estimate the outcome under control

$$\hat{E}(Y^0 \mid S = \text{Always}) = \$18$$

The difference tells us that the average causal effect of job training on the wages of the always employed is between \$27 - \$18 = \$9 and \$36 - \$18 = \$18.

Chapter 2

Why regression

The goal of this package is to support parametric model-based principal stratification. This page motivates that choice: why should we want a model?

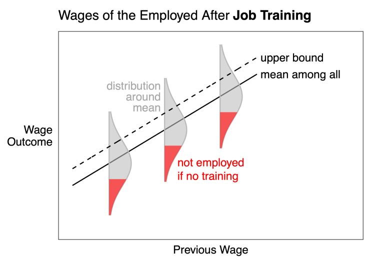
Parametric regression models make principal stratification bounds applicable in social science settings with many covariates. One such setting is causal inference in observational settings. In that setting, we might want to statistically adjust for a covariate vector $\vec{X}$.

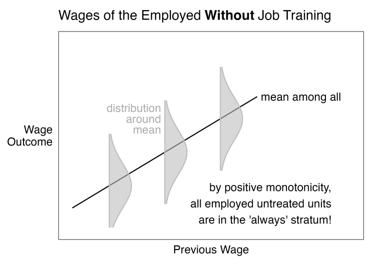
- when $\vec{X}$ can take only a few discrete values, one can carry out principal stratification as above within each stratum of $\vec{X}$
- when $\vec{X}$ can take many values, it is possible that each vector value $\vec{X}_i$ is unique

In the latter case, we need model-based principal stratification.

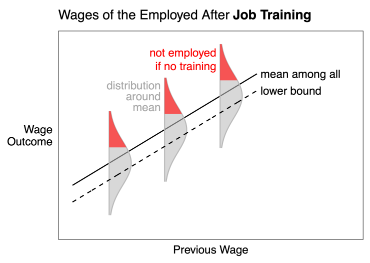
1) Model the mediator
    - Fit a model for the mediator M
    - Estimate the probability of each stratum for each unit
2) Model the outcome
    - Fit a model for the outcome $Y \mid A = 1, M = 1, \vec{X}$
        - Example: Model the distribution of wages for employed job training recipients
    - Fit a model for the outcome $Y \mid A = 0, M = 1, \vec{X}$
        - Example: Model the distribution of wages for employed job training recipients
    - Bound by assuming that the proportion induced (from 1) is the upper or lower portion conditional on covariates

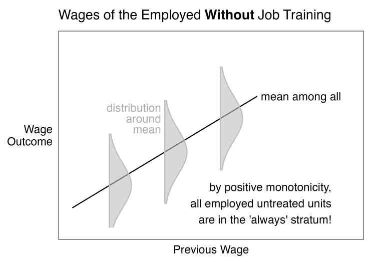
2.0.1 Upper bound: Left - Right




Wages of the Employed **Without** Job Training
mean among all
distribution around mean
Wage Outcome
by positive monotonicity, all employed untreated units are in the 'always' stratum!
Previous Wage

2.0.2 Lower bound: Left - Right




Wages of the Employed **Without** Job Training
mean among all
distribution around mean
Wage Outcome
by positive monotonicity, all employed untreated units are in the 'always' stratum!
Previous Wage

Chapter 3

Basic functionality

This page illustrates the basic functionality of the `pstratreg` function. This function conducts parametric principal stratification analysis to estimate average causal effect among the always-valid subgroup whose outcome would exist in either treatment condition.

> Jargon? Start with the first page on the goal!

The package automates the process to

- estimate a mediator model
- estimate an outcome model
  - allowing heteroskedasticity if needed
- calculate the conditional probability of being always-valid
- implement monotonicity assumptions
- bound estimates using the conditional outcome distribution and the proportion in the always-valid subgroup
- return estimates, conditional on population subgroups if requested

3.1 Simulate data

We first simulate some data for illustration.

The data has four variables

- continuous confounder `x`
- binary treatment `a`

- binary mediator `m`
- continuous outcome `y`
  - `y` is `NA` when `m = FALSE`

```
library(tidyverse)
library(pstratreg)
data <- pstratreg_sim(n = 100)
```

```
#>            x     a     s          y
#> 1 -1.9122208 FALSE FALSE         NA
#> 2 -0.7676647 FALSE  TRUE -0.1775097
#> 3 -1.6331539 FALSE  TRUE -1.9658900
#> 4 -0.4652854 FALSE  TRUE -1.6908824
#> 5 -1.0495272 FALSE FALSE         NA
```

3.2 The pstratreg function

The call below runs a principal stratification regression analysis with default options, returning estimates of the average treatment effect among the latent stratum who would have valid outcomes regardless of treatment.

```
result <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  data = data,
  treatment_name = "a"
)
```

```
#> Effect on survival, where S = 1 indicates the outcome exists
#> # A tibble: 1 x 3
#>      s0    s1 effect_s
#>   <dbl> <dbl>    <dbl>
#> 1 0.809 0.848   0.0390
#>
#> Effect on outcome among those who would have a valid outcome regardless of treatmen
#> # A tibble: 1 x 2
#>   effect_y_lower effect_y_upper
#>            <dbl>          <dbl>
#> 1          0.695           1.83
```

3.3 Positive monotonicity

If you believe that the `TRUE` value of the treatment never causes the outcome to be undefined, then you might add the `monotonicity_positive = TRUE` option.

Sometimes, the monotonicity assumption disagrees with the empirical estimates in at least some cases. For example, we assume that treatment never prevents a valid outcome but empirically we estimate that the treatment increases the probability that the treatment increases the value of the mediator in some subgroups. When this happens, the package issues a warning.

Empirical monotonicity violations may be non-troubling; they can occur in estimates due to random chance from sampling variability. Because the user has assumed monotonicity, the package assumes that any violations arise purely from estimation errors. The predicted values of the mediator under each treatment condition in these cases are forced to be equal, at the midpoint of the two predicted values.

Generally, if the warning tells you that monotonicity is violated in only a small percent of cases, it may be warranted to proceed. If monotonicity is empirically violated in many cases, then you may need to rethink this assumption.

```r
result <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  data = data,
  treatment_name = "a",
  monotonicity_positive = TRUE
)
#> Warning in pstratreg(formula_y = formula(y ~ x * a), formula_s = formula(s ~ : Monotonicity v
#> Forcing s1_trunc = s0_trunc at midpoint of estimates for those
```

```
#> Effect on survival, where S = 1 indicates the outcome exists
#> # A tibble: 1 x 3
#>      s0    s1 effect_s
#>   <dbl> <dbl>    <dbl>
#> 1 0.809 0.848   0.0390
#>
#> Effect on outcome among those who would have a valid outcome regardless of treatment
#> # A tibble: 1 x 2
#>   effect_y_lower effect_y_upper
#>            <dbl>          <dbl>
#> 1           1.20           1.33
```

3.4 Negative monotonicity

Conversely, you can assume negative monotonicity with `monotonicity_negative = TRUE`. If you are not sure what negative monotonicity is, see the previous page on the big idea!

In this particular simulation, negative monotonicity does not hold and you see below that the warning appropriately alerts us that monotonicity is frequently empirically violated.

```
result <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  data = data,
  treatment_name = "a",
  monotonicity_negative = TRUE
)
#> Warning in pstratreg(formula_y = formula(y ~ x * a), formula_s = formula(s ~ : Mono
#> Forcing s1_trunc = s0_trunc at midpoint of estimates for those
```

```
#> Effect on survival, where S = 1 indicates the outcome exists
#> # A tibble: 1 x 3
#>      s0    s1 effect_s
#>   <dbl> <dbl>    <dbl>
#> 1 0.809 0.848   0.0390
#>
#> Effect on outcome among those who would have a valid outcome regardless of treatmen
#> # A tibble: 1 x 2
#>   effect_y_lower effect_y_upper
#>            <dbl>          <dbl>
#> 1           1.26           1.37
```

3.5 Aggregate in subgroups

Instead of sample average effect estimates, you might want the estimate within subgroups defined by a grouping variable in the data. The `group_vars` argument allows you to specify a vector of variable names from `data` within which to aggregate results.

First we create some groups for illustration

```
data_with_groups <- data %>%
  mutate(group1 = x < -.5,
         group2 = x > .5)
```

```
#>            x     a     s           y group1 group2
#> 1 -1.9122208 FALSE FALSE          NA   TRUE  FALSE
#> 2 -0.7676647 FALSE  TRUE -0.17750966   TRUE  FALSE
#> 3 -1.6331539 FALSE  TRUE -1.96588997   TRUE  FALSE
#> 4 -0.4652854 FALSE  TRUE -1.69088236  FALSE  FALSE
#> 5 -1.0495272 FALSE FALSE          NA   TRUE  FALSE
#> 6  0.1214910 FALSE  TRUE  0.04738583  FALSE  FALSE
```

and then we apply the function to estimate within those groups.

```
result <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  data = data_with_groups,
  treatment_name = "a",
  group_vars = c("group1","group2")
)
```

```
#> Effect on survival, where S = 1 indicates the outcome exists
#> # A tibble: 3 x 5
#>   group1 group2    s0    s1 effect_s
#>   <lgl>  <lgl>  <dbl> <dbl>    <dbl>
#> 1 FALSE  FALSE  0.902 0.876  -0.0267
#> 2 FALSE  TRUE   0.991 0.959  -0.0318
#> 3 TRUE   FALSE  0.486 0.678   0.192
#>
#> Effect on outcome among those who would have a valid outcome regardless of treatment
#> # A tibble: 3 x 4
#> # Groups:   group1, group2 [3]
#>   group1 group2 effect_y_lower effect_y_upper
#>   <lgl>  <lgl>           <dbl>          <dbl>
#> 1 TRUE   FALSE          -0.273          4.16
#> 2 FALSE  FALSE           1.04           1.92
#> 3 FALSE  TRUE            0.759          0.980
```

3.6 Sample weights

If you have case weights from sampling with unequal probabilities, they can be provided in a vector of length `nrow(data)` using the `weights` argument.

Here we generate some hypothetical weights

```
sim_weights <- runif(nrow(data))
```

and then call the function. Note that the `glm()` used internally to estimate logistic regression will create a warning for `non-integer #successes in a binomial glm!` which is to be expected when weights are used in this function.

```
result <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  data = data,
  treatment_name = "a",
  weights = sim_weights
)
#> Warning in eval(family$initialize): non-integer #successes
#> in a binomial glm!
```

```
#> Effect on survival, where S = 1 indicates the outcome exists
#> # A tibble: 1 x 3
#>      s0    s1 effect_s
#>   <dbl> <dbl>    <dbl>
#> 1 0.812 0.803 -0.00923
#>
#> Effect on outcome among those who would have a valid outcome regardless of treatmen
#> # A tibble: 1 x 2
#>   effect_y_lower effect_y_upper
#>            <dbl>          <dbl>
#> 1          0.576           1.64
```

Chapter 4

Relaxing homoskedasticity

Model-based principal stratification bounds involve a model for the conditional distribution of the outcome, not just the conditional mean. For that goal, one might be concerned about the homoskedasticity assumption.

> Homoskedasticity: The assumption that the conditional outcome variance is equal at all values of the predictors.

While commonly assumed in Ordinary Least Squares, homoskedasticity is an assumption that can be relaxed. We currently support parametric model for heteroskedasticity based on the ideas of variance function regression (Western & Bloome 2009).

4.1 Model the conditional mean

We begin with an ordinary least squares outcome model, as in the homoskedastic case,

$$E(Y \mid \vec{X}, A, M = 1) = \alpha + A\hat{\beta} + \vec{X}\vec{\gamma} + A\vec{X}'\vec{\eta}$$

where A is a binary treatment, $\vec{X}$ is a vector of measured confounders, and $M = 1$ is the mediator indicating that the outcome is valid. Let $\hat{Y}_i$ be the predicted value for each unit i from this model.

4.2 Model the conditional variance

We next allow the conditional variance of Y to vary as a function of A and $\vec{X}$. We first define the squared residual

$$\hat{\epsilon}^2 = \left(Y - \hat{Y}\right)^2$$

Under the assumption that $\hat{\epsilon}$ is normally distributed, the squared residual $\hat{\epsilon}^2$ follows a Gamma distribution with mean equal to the conditional variance $\sigma^2(\vec{X}, A, M = 1)$. We therefore model $\hat{\epsilon}^2$ by a Gamma GLM with a log link function, using a parametric linear predictor such as the one below.

$$\log(\sigma^2(\vec{X}, A, M = 1)) = \lambda + A\delta + \vec{X}'\vec{\nu} + A\vec{X}'\vec{\omega}$$

Predictions from this model (exponentiated) are estimates of the conditional variance, $\hat{V}(Y \mid \vec{X}, A, M = 1)$ at the observed predictors for each unit. We make predictions under the treatment and control conditions.

$$\begin{aligned}\hat{V}(Y \mid \vec{X}, A = 1, M = 1) &= \exp\left[\hat{\lambda} + \hat{\delta} + \vec{X}'\left(\hat{\vec{\nu}} + \hat{\vec{\omega}}\right)\right] \\ \hat{V}(Y \mid \vec{X}, A = 0, M = 1) &= \exp\left[\hat{\lambda} + \vec{X}'\hat{\vec{\nu}}\right]\end{aligned}$$

4.3 In code

You can relax the homskedasticity assumption with the `homoskedastic = FALSE` argument. Doing so estimates a variance function regression (as above) using the same predictor model formula as in `formula_y`.

```r
library(tidyverse)
library(pstratreg)
data <- pstratreg_sim(n = 100)
```

With the data setup above, we can estimate the models.

```r
result <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  data = data,
  treatment_name = "a",
  homoskedastic = FALSE
)
```

```
#> Effect on survival, where S = 1 indicates the outcome exists
#> # A tibble: 1 x 3
#>      s0    s1 effect_s
#>   <dbl> <dbl>    <dbl>
#> 1 0.708 0.782   0.0734
```

```
#>
#> Effect on outcome among those who would have a valid outcome regardless of treatment
#> # A tibble: 1 x 2
#>   effect_y_lower effect_y_upper
#>            <dbl>          <dbl>
#> 1          0.192           2.65
```

Optionally, you can specify a separate model formula for the model of squared residuals that may be simpler than the model formula used for Y, which might be done if the model formula involves many parameters and you see errors from the internal `glm()` call about convergence for the variance regression.

```
result <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  formula_sq_resid = formula(~ x + a),
  data = data,
  treatment_name = "a",
  homoskedastic = FALSE
)
```

```
#> Effect on survival, where S = 1 indicates the outcome exists
#> # A tibble: 1 x 3
#>      s0    s1 effect_s
#>   <dbl> <dbl>    <dbl>
#> 1 0.708 0.782   0.0734
#>
#> Effect on outcome among those who would have a valid outcome regardless of treatment
#> # A tibble: 1 x 2
#>   effect_y_lower effect_y_upper
#>            <dbl>          <dbl>
#> 1          0.194           2.65
```

Chapter 5

Standard errors

Appropriate standard errors depend on application-specific knowledge about how the data were generated. For example, appropriate standard errors are not the same for sampling variability in simple random samples, sampling variability in complex samples, and finite-sample inference with variation from random treatment assignment.

For the particular case of simple random samples, the package supports standard error estimation by the nonparametric bootstrap and 95% confidence intervals by a Normal approximation using the estimated standard error.

We first set up the environment

```r
library(tidyverse)
library(pstratreg)
```

and simulate data

```r
data <- pstratreg_sim(n = 100)
```

and produce point estimates with a call to `pstratreg`.

```r
pstratreg.out <- pstratreg(
  formula_y = formula(y ~ x*a),
  formula_s = formula(s ~ x*a),
  data = data,
  treatment_name = "a"
)
```

To estimate standard errors and confidence intervals, hand the output of the call to `pstratreg` to the `pstratreg_se` function, optionally specifying the number `r` of bootstrap samples.

```
result_with_se <- pstratreg_se(pstratreg.out, r = 100)
```

The output of a call to `pstratreg_se` is a data frame containing estimates and inferential quantities.

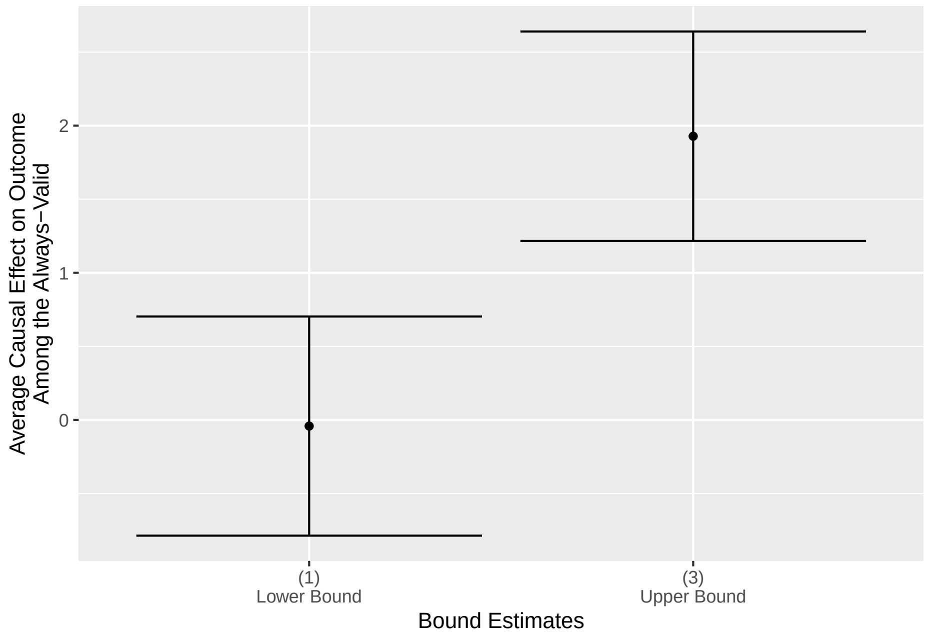
```
#> # A tibble: 2 x 5
#>   estimand       estimate    se ci.min ci.max
#>   <chr>             <dbl> <dbl>  <dbl>  <dbl>
#> 1 effect_y_lower  -0.0415 0.380 -0.786  0.703
#> 2 effect_y_upper   1.93   0.363  1.22   2.64
```

See the end of this page for a glossary of estimand terms in the output.

5.1 Visualize the result

One way to use this output to create visualizations using `ggplot`. For example, the code below visualizes bounded effects on the outcome.

```
result_with_se %>%
  filter(estimand %in% c("effect_y_lower","effect_y_naive","effect_y_upper")) %>%
  mutate(estimand_label = case_when(
    estimand == "effect_y_lower" ~ "(1)\nLower Bound",
    estimand == "effect_y_naive" ~ "(2)\nEstimate if No\nPost-Treatment\nSelection",
    estimand == "effect_y_upper" ~ "(3)\nUpper Bound"
  )) %>%
  ggplot(aes(x = estimand_label, y = estimate,
             ymin = ci.min, ymax = ci.max)) +
  geom_point() +
  geom_errorbar() +
  ylab("Average Causal Effect on Outcome\nAmong the Always-Valid") +
  scale_x_discrete(name = "Bound Estimates")
```



5.2 Glossary of estimands that result

Value of `estimand`	Definition	English
`effect_y_naive`	$\hat{E}_{\text{Naive}}(Y^1 - Y^0 \mid S = \text{Always})$	Average effect of treatment on outcome in always-valid stratum, estimated by assumption that potential outcomes are conditionally independent of M
`effect_y_lower`	$\hat{E}_{\text{Lower}}(Y^1 - Y^0 \mid S = \text{Always})$	Lower bound average effect of treatment on outcome in always-valid stratum
`effect_y_upper`	$\hat{E}_{\text{Lower}}(Y^1 - Y^0 \mid S = \text{Always})$	Upper bound average effect of treatment on outcome in always-valid stratum

Value of `estimand`	Definition	English
`yhat_1_naive`	$\hat{E}_{\text{Naive}}(Y^1 \mid S = \text{Always})$	Average outcome under treatment in the always-valid stratum, estimated by assumption potential outcomes are conditionally independent of M
`yhat_0_naive`	$\hat{E}_{\text{Naive}}(Y^0 \mid S = \text{Always})$	Analogous to above, for outcome under control
`yhat_1_lower`	$\hat{E}_{\text{Lower}}(Y^1 \mid S = \text{Always})$	Lower bound average outcome under treatment in the always-valid stratum
`yhat_0_lower`	$\hat{E}_{\text{Lower}}(Y^0 \mid S = \text{Always})$	Analogous to above, for outcome under control
`yhat_1_upper`	$\hat{E}_{\text{Upper}}(Y^1 \mid S = \text{Always})$	Upper bound average outcome under treatment in the always-valid stratum
`yhat_0_upper`	$\hat{E}_{\text{Upper}}(Y^0 \mid S = \text{Always})$	Analogous to above, for outcome under control
`effect_m`	$\hat{E}(M^1 - M^0)$	Average effect of treatment on mediator
`mhat1`	$\hat{E}(M^1)$	Average mediator under treatment
`mhat1_trunc`		The above, truncated to comply with the user-defined positivity assumption
`mhat0`	$\hat{E}(M^0)$	Average mediator under control
`mhat0_trunc`		The above, truncated to comply with the user-defined positivity assumption