

# Package ‘paths’

May 9, 2026

**Type** Package

**Title** An Imputation Approach to Estimating Path-Specific Causal Effects

**Version** 0.1.2

**Date** 2025-12-22

**Description**

In causal mediation analysis with multiple causally ordered mediators, a set of path-specific effects are identified under standard ignorability assumptions. This package implements an imputation approach to estimating these effects along with a set of bias formulas for conducting sensitivity analysis (Zhou and Yamamoto <[doi:10.31235/osf.io/2rx6p](https://doi.org/10.31235/osf.io/2rx6p)>). It contains two main functions: `paths()` for estimating path-specific effects and `sens()` for conducting sensitivity analysis. Estimation uncertainty is quantified using the nonparametric bootstrap.

**Encoding** UTF-8

**LazyLoad** yes

**LazyData** yes

**License** GPL (>= 2)

**Depends** R (>= 4.5.0)

**Suggests** knitr, rmarkdown

**Imports** BART, boot, gbm, ggplot2, metR, stats, twang, tidyR

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Author** Minh Trinh [aut],  
Teppei Yamamoto [aut],  
Xiang Zhou [cre]

**Maintainer** Xiang Zhou <[xiang\\_zhou@fas.harvard.edu](mailto:xiang_zhou@fas.harvard.edu)>

**Repository** CRAN

**Date/Publication** 2025-12-22 17:00:09 UTC

## Contents

paths . . . . .	2
plot.paths . . . . .	5
plot.sens . . . . .	7
pred . . . . .	8
sens . . . . .	9
summary.paths . . . . .	11
tatar . . . . .	13
welfare . . . . .	15
<b>Index</b>	<b>17</b>

---

paths *Causal Paths Analysis*

---

## Description

paths estimates path-specific causal effects in the presence of  $K (\geq 1)$  causally ordered mediators. It implements the pure imputation estimator and the imputation-based weighting estimator (when a propensity score model is provided) as detailed in Zhou and Yamamoto (2020). The user supplies the names of the treatment, outcome, mediator variables,  $K + 1$  fitted models characterizing the conditional mean of the outcome given treatment, pretreatment confounders, and varying sets of mediators, and a data frame containing all the variables. The function returns  $K + 1$  path-specific causal effects that together constitute the total treatment effect. When  $K = 1$ , the path-specific causal effects are identical to the natural direct and indirect effects in standard causal mediation analysis.

## Usage

```
paths(
  a,
  y,
  m,
  models,
  ps_model = NULL,
  data,
  nboot = 500,
  conf_level = 0.95,
  ...
)

## S3 method for class 'paths'
print(x, digits = 3, ...)
```

**Arguments**

<code>a</code>	a character string indicating the name of the treatment variable. The treatment should be a binary variable taking either 0 or 1.
<code>y</code>	a character string indicating the name of the outcome variable.
<code>m</code>	a list of $K$ character vectors indicating the names of $K$ causally ordered mediators $M_1, \dots, M_K$ .
<code>models</code>	a list of $K + 1$ fitted model objects describing how the outcome depends on treatment, pretreatment confounders, and varying sets of mediators, where $K$ is the number of mediators. <ul style="list-style-type: none"> <li>the first element is a baseline model of the outcome conditional on treatment and pretreatment confounders.</li> <li>the <math>k</math>th element is an outcome model conditional on treatment, pretreatment confounders, and mediators <math>M_1, \dots, M_{k-1}</math>.</li> <li>the last element is an outcome model conditional on treatment, pretreatment confounders, and all of the mediators, i.e., <math>M_1, \dots, M_K</math>.</li> </ul> <p>The fitted model objects can be of type <code>lm</code>, <code>glm</code>, <code>gbm</code>, <code>wbart</code>, or <code>pbart</code>.</p>
<code>ps_model</code>	an optional propensity score model for treatment. It can be of type <code>glm</code> , <code>gbm</code> , <code>ps</code> , or <code>pbart</code> . When it is provided, the imputation-based weighting estimator is also used to compute path-specific causal effects.
<code>data</code>	a data frame containing all variables.
<code>nboot</code>	number of bootstrap iterations for estimating confidence intervals. Default is 500.
<code>conf_level</code>	the confidence level of the returned two-sided confidence intervals. Default is 0.95.
<code>...</code>	additional arguments to be passed to <code>boot::boot</code> , e.g. <code>parallel</code> and <code>ncpus</code> . For the <code>print</code> method, additional arguments to be passed to <code>print.default</code>
<code>x</code>	a fitted model object returned by the <code>paths</code> function.
<code>digits</code>	minimal number of significant digits printed.

**Value**

An object of class `paths`, which is a list containing the following elements

**pure** estimates of direct and path-specific effects via  $M_1, \dots, M_K$  based on the pure imputation estimator.

**hybrid** estimates of direct and path-specific effects via  $M_1, \dots, M_K$  based on the imputation-based weighting estimator.

**varnames** a list of character strings indicating the names of the pretreatment confounders ( $X$ ), treatment ( $A$ ), mediators ( $M_1, \dots, M_K$ ), and outcome ( $Y$ ).

**formulas** formulas for the outcome models.

**classes** classes of the outcome models.

**families** model families of the outcome models.

**args** a list containing arguments of the outcome models.

**ps\_formula** formula for the propensity score model.  
**ps\_class** class of the propensity score model.  
**ps\_family** model family of the propensity score model.  
**ps\_args** arguments of the propensity score model.  
**data** the original data.  
**nboot** number of bootstrap iterations.  
**conf\_level** confidence level for confidence intervals.  
**boot\_out** output matrix from the bootstrap iterations.  
**call** the matched call to the paths function.

## References

Zhou, Xiang and Tepei Yamamoto. 2020. "[Tracing Causal Paths from Experimental and Observational Data](#)".

## See Also

[summary.paths](#), [plot.paths](#), [sens](#)

## Examples

```
data(tatar)

m1 <- c("trust_g1", "victim_g1", "fear_g1")
m2 <- c("trust_g2", "victim_g2", "fear_g2")
m3 <- c("trust_g3", "victim_g3", "fear_g3")
mediators <- list(m1, m2, m3)

formula_m0 <- annex ~ kulak + prosoviet_pre + religiosity_pre + land_pre +
  orchard_pre + animals_pre + carriage_pre + otherprop_pre + violence
formula_m1 <- update(formula_m0, ~ . + trust_g1 + victim_g1 + fear_g1)
formula_m2 <- update(formula_m1, ~ . + trust_g2 + victim_g2 + fear_g2)
formula_m3 <- update(formula_m2, ~ . + trust_g3 + victim_g3 + fear_g3)
formula_ps <- violence ~ kulak + prosoviet_pre + religiosity_pre +
  land_pre + orchard_pre + animals_pre + carriage_pre + otherprop_pre

#####
# Causal Paths Analysis using GLM
#####

# outcome models
glm_m0 <- glm(formula_m0, family = binomial("logit"), data = tatar)
glm_m1 <- glm(formula_m1, family = binomial("logit"), data = tatar)
glm_m2 <- glm(formula_m2, family = binomial("logit"), data = tatar)
glm_m3 <- glm(formula_m3, family = binomial("logit"), data = tatar)
glm_models <- list(glm_m0, glm_m1, glm_m2, glm_m3)

# propensity score model
glm_ps <- glm(formula_ps, family = binomial("logit"), data = tatar)
```

```

# causal paths analysis using glm
# note: For illustration purposes only a small number of bootstrap replicates are used
paths_glm <- paths(a = "violence", y = "annex", m = mediators,
  glm_ymodels, ps_model = glm_ps, data = tatar, nboot = 3)

#####
# Causal Paths Analysis using GBM
#####

require(gbm)

# outcome models
gbm_m0 <- gbm(formula_m0, data = tatar, distribution = "bernoulli", interaction.depth = 3)
gbm_m1 <- gbm(formula_m1, data = tatar, distribution = "bernoulli", interaction.depth = 3)
gbm_m2 <- gbm(formula_m2, data = tatar, distribution = "bernoulli", interaction.depth = 3)
gbm_m3 <- gbm(formula_m3, data = tatar, distribution = "bernoulli", interaction.depth = 3)
gbm_ymodels <- list(gbm_m0, gbm_m1, gbm_m2, gbm_m3)

# propensity score model via gbm
gbm_ps <- gbm(formula_ps, data = tatar, distribution = "bernoulli", interaction.depth = 3)

# causal paths analysis using gbm
# note: For illustration purposes only a small number of bootstrap replicates are used
paths_gbm <- paths(a = "violence", y = "annex", m = mediators,
  gbm_ymodels, ps_model = gbm_ps, data = tatar, nboot = 3)

```

---

plot.paths

*Plot Method for paths Objects*


---

## Description

Plot point estimates and confidence intervals for each individual path-specific effect from a paths object.

## Usage

```

## S3 method for class 'paths'
plot(
  x,
  mediator_names = NULL,
  estimator = c("pure", "hybrid", "both"),
  decomp = c("Type I", "Type II", "both"),
  horizontal = (decomp != "both"),
  ...
)

```

**Arguments**

x	an object of class paths returned by the <code>paths</code> function
mediator_names	a vector of character strings giving the labels for each mediator. It must contain as many elements as the number of mediators in the model. If not supplied, a set of default labels will be constructed from the fitted paths object.
estimator	either "pure", "hybrid", or "both", indicating whether the plot will display estimates obtained using the pure imputation estimator, the imputation-based weighting estimator, or both. Default is to show estimates from the pure imputation estimator.
decomp	either "Type I", "Type II", or "both", indicating whether the plot will display estimates obtained using Type I decomposition, Type II decomposition, or both Type I and Type II decompositions. Default is to show estimates from Type I decomposition.
horizontal	a logical variable indicating whether a horizontal plot should be used. Default is to use a horizontal plot when <code>decomp != both</code> .
...	additional arguments.

**Value**

a ggplot2 plot, which can be further customized by the user.

**Examples**

```
data(tatar)

m1 <- c("trust_g1", "victim_g1", "fear_g1")
m2 <- c("trust_g2", "victim_g2", "fear_g2")
m3 <- c("trust_g3", "victim_g3", "fear_g3")
mediators <- list(m1, m2, m3)

formula_m0 <- annex ~ kulak + prosoviet_pre + religiosity_pre + land_pre +
  orchard_pre + animals_pre + carriage_pre + otherprop_pre + violence
formula_m1 <- update(formula_m0, ~ . + trust_g1 + victim_g1 + fear_g1)
formula_m2 <- update(formula_m1, ~ . + trust_g2 + victim_g2 + fear_g2)
formula_m3 <- update(formula_m2, ~ . + trust_g3 + victim_g3 + fear_g3)
formula_ps <- violence ~ kulak + prosoviet_pre + religiosity_pre +
  land_pre + orchard_pre + animals_pre + carriage_pre + otherprop_pre

#####
# Causal Paths Analysis using GLM
#####

# outcome models
glm_m0 <- glm(formula_m0, family = binomial("logit"), data = tatar)
glm_m1 <- glm(formula_m1, family = binomial("logit"), data = tatar)
glm_m2 <- glm(formula_m2, family = binomial("logit"), data = tatar)
glm_m3 <- glm(formula_m3, family = binomial("logit"), data = tatar)
glm_ymodels <- list(glm_m0, glm_m1, glm_m2, glm_m3)
```

```

# propensity score model
glm_ps <- glm(formula_ps, family = binomial("logit"), data = tatar)

# causal paths analysis using glm
# note: For illustration purposes only a small number of bootstrap replicates are used
paths_glm <- paths(a = "violence", y = "annex", m = mediators,
  glm_ymodels, ps_model = glm_ps, data = tatar, nboot = 3)

# plot total, direct, and path-specific effects
plot(paths_glm, mediator_names = c("G1 identity", "G2 identity", "G3 identity"),
  estimator = "both")

```

---

plot.sens

*Plot Method for sens Objects*


---

## Description

Contour plot for sensitivity analysis objects.

## Usage

```

## S3 method for class 'sens'
plot(
  x,
  outcome_name = "Outcome",
  x_axis = c("eta_k", "gamma_k"),
  other = list(`eta_k-1` = NULL, `gamma_k-1` = NULL),
  ...
)

```

## Arguments

x	an object of class sens returned by the <a href="#">sens</a> function
outcome_name	a character string indicating the name of the outcome.
x_axis	sensitivity analysis parameter shown on the x axis. Default is "eta_k", i.e., the difference in the prevalence of an unobserved confounder $U$ between treated and untreated units given pretreatment covariates $X$ and mediators $M_1, \dots, M_k$ . Alternatively, it can be "gamma_k", i.e., the average effect of the unobserved confounder $U$ on the outcome given pretreatment covariates $X$ , treatment $A$ , and mediators $M_1, \dots, M_k$ .
other	a named list indicating the values at which other sensitivity analysis parameters, namely, <code>`eta_k-1`</code> and <code>`gamma_k-1`</code> , are held. This is needed only when the bias formula involves $\eta_{k-1}$ and $\gamma_{k-1}$ as well as $\eta_k$ and $\gamma_k$ . The default is to set both $\eta_{k-1}$ and $\gamma_{k-1}$ at their average values in the sens object.
...	additional arguments

**Value**

a ggplot2 plot, which can be further customized by the user.

**Examples**

```
data(tatar)

m1 <- c("trust_g1", "victim_g1", "fear_g1")
m2 <- c("trust_g2", "victim_g2", "fear_g2")
m3 <- c("trust_g3", "victim_g3", "fear_g3")
mediators <- list(m1, m2, m3)

formula_m0 <- annex ~ kulak + prosoviet_pre + religiosity_pre + land_pre +
  orchard_pre + animals_pre + carriage_pre + otherprop_pre + violence
formula_m1 <- update(formula_m0, ~ . + trust_g1 + victim_g1 + fear_g1)
formula_m2 <- update(formula_m1, ~ . + trust_g2 + victim_g2 + fear_g2)
formula_m3 <- update(formula_m2, ~ . + trust_g3 + victim_g3 + fear_g3)
formula_ps <- violence ~ kulak + prosoviet_pre + religiosity_pre +
  land_pre + orchard_pre + animals_pre + carriage_pre + otherprop_pre

#####
# Causal Paths Analysis using GLM
#####

# outcome models
glm_m0 <- glm(formula_m0, family = binomial("logit"), data = tatar)
glm_m1 <- glm(formula_m1, family = binomial("logit"), data = tatar)
glm_m2 <- glm(formula_m2, family = binomial("logit"), data = tatar)
glm_m3 <- glm(formula_m3, family = binomial("logit"), data = tatar)
glm_ymodels <- list(glm_m0, glm_m1, glm_m2, glm_m3)

# propensity score model
glm_ps <- glm(formula_ps, family = binomial("logit"), data = tatar)

# causal paths analysis using glm
# note: For illustration purposes only a small number of bootstrap replicates are used
paths_glm <- paths(a = "violence", y = "annex", m = mediators,
  glm_ymodels, ps_model = glm_ps, data = tatar, nboot = 3)

# sensitivity analysis for the path-specific effect via M1
sens_glm <- sens(paths_glm, confounded = "M1", estimand = "via M1",
  gamma_values = - seq(0, 0.5, 0.005), eta_values = seq(-0.5, 0.5, 0.005))

# plot(sens_glm)
```

**Description**

Generic function that returns predicted outcomes from `lm`, `glm`, `gbm`, `wbart`, and `pbart` objects with new data.

**Usage**

```
pred(object, newdata, ...)

## S3 method for class 'lm'
pred(object, newdata, ...)

## S3 method for class 'glm'
pred(object, newdata, ...)

## S3 method for class 'gbm'
pred(object, newdata, method = "OOB", ...)

## S3 method for class 'pbart'
pred(object, newdata, ...)

## S3 method for class 'wbart'
pred(object, newdata, ...)
```

**Arguments**

<code>object</code>	a fitted model object, which can be of class <code>lm</code> , <code>glm</code> , <code>gbm</code> , <code>wbart</code> , or <code>pbart</code> .
<code>newdata</code>	a data frame containing predictor variables.
<code>...</code>	additional arguments passed to the predict methods.
<code>method</code>	Method used to determine the optimal number of boosting iterations for <code>gbm</code> objects.

**Value**

a vector of expected outcomes for `newdata`

---

sens	<i>Sensitivity Analysis for Unobserved Confounding on Path-Specific Causal Effects</i>
------	--

---

**Description**

`sens` implements a set of bias formulas detailed in Zhou and Yamamoto (2020) for assessing the sensitivity of estimated path-specific effects to an unobserved confounder  $U$  of a mediator-outcome relationship. The user provides a fitted `paths` object, the mediator whose relationship with the outcome is potentially confounded, the estimand whose sensitivity to unobserved confounding is being investigated, type of estimator, type of decomposition, and possible values of the  $\gamma$  and  $\eta$  parameters.

**Usage**

```

sens(
  object,
  confounded = "M1",
  estimand = "via M1",
  estimator = c("pure", "hybrid"),
  decomp = c("Type I", "Type II"),
  gamma_values = NULL,
  eta_values = NULL
)

```

**Arguments**

object	a fitted model object returned by the <a href="#">paths</a> function.
confounded	a character string indicating the mediator whose relationship with the outcome is potentially confounded. One of {"M1", "M2", ...}.
estimand	a character string indicating the estimand whose sensitivity to unobserved confounding is being investigated. One of {"M1", "M2", ...} or "direct".
estimator	type of estimator, the pure imputation estimator ("pure") or the imputation-based weighting estimator ("hybrid").
decomp	type of decomposition, "Type I" or "Type II".
gamma_values	potential values of the $\gamma$ parameter, which denotes the average effect of the unobserved confounder $U$ on the outcome given pretreatment covariates $X$ , treatment $A$ , and mediators $M_1, \dots, M_k$ . If not provided, it is defaulted to a range of 20 values from $-sd(Y)$ to $sd(Y)$ , where $sd$ denotes standard deviation and $Y$ denotes the outcome variable.
eta_values	potential values of the $\eta$ parameter, which denotes the difference in the prevalence of the unobserved confounder $U$ between treated and untreated units given pretreatment covariates $X$ and mediators $M_1, \dots, M_k$ . If not provided, it is defaulted to a range of 20 values from $-sd(A)$ to $sd(A)$ , where $sd$ denotes standard deviation and $A$ denotes the treatment variable.

**Value**

A list containing the following elements

**original** original estimate of the corresponding path-specific effect.

**adjusted** a data frame where each row represents a potential combination of  $\gamma$  and  $\eta$ , the corresponding bias, bias-adjusted estimate, and an indicator for whether the bias-adjusted estimate is of the opposite sign to the original estimate.

**References**

Zhou, Xiang and Teppei Yamamoto. 2020. "[Tracing Causal Paths from Experimental and Observational Data](#)".

**See Also**[paths](#), [plot.sens](#)**Examples**

```

data(tatar)

m1 <- c("trust_g1", "victim_g1", "fear_g1")
m2 <- c("trust_g2", "victim_g2", "fear_g2")
m3 <- c("trust_g3", "victim_g3", "fear_g3")
mediators <- list(m1, m2, m3)

formula_m0 <- annex ~ kulak + prosoviet_pre + religiosity_pre + land_pre +
  orchard_pre + animals_pre + carriage_pre + otherprop_pre + violence
formula_m1 <- update(formula_m0, ~ . + trust_g1 + victim_g1 + fear_g1)
formula_m2 <- update(formula_m1, ~ . + trust_g2 + victim_g2 + fear_g2)
formula_m3 <- update(formula_m2, ~ . + trust_g3 + victim_g3 + fear_g3)
formula_ps <- violence ~ kulak + prosoviet_pre + religiosity_pre +
  land_pre + orchard_pre + animals_pre + carriage_pre + otherprop_pre

#####
# Causal Paths Analysis using GLM
#####

# outcome models
glm_m0 <- glm(formula_m0, family = binomial("logit"), data = tatar)
glm_m1 <- glm(formula_m1, family = binomial("logit"), data = tatar)
glm_m2 <- glm(formula_m2, family = binomial("logit"), data = tatar)
glm_m3 <- glm(formula_m3, family = binomial("logit"), data = tatar)
glm_ymodels <- list(glm_m0, glm_m1, glm_m2, glm_m3)

# propensity score model
glm_ps <- glm(formula_ps, family = binomial("logit"), data = tatar)

# causal paths analysis using glm
# note: For illustration purposes only a small number of bootstrap replicates are used
paths_glm <- paths(a = "violence", y = "annex", m = mediators,
  glm_ymodels, ps_model = glm_ps, data = tatar, nboot = 3)

# sensitivity analysis for the path-specific effect via M1
sens_glm <- sens(paths_glm, confounded = "M1", estimand = "via M1",
  gamma_values = seq(0, 0.5, 0.005), eta_values = seq(-0.5, 0.5, 0.005))

# plot(sens_glm)

```

**Description**

Function to report results from causal paths analysis. Report point estimates and standard errors for the total effect, direct effect, and each individual indirect effect, separately for Type I and Type II decompositions.

**Usage**

```
## S3 method for class 'paths'
summary(object, ...)

## S3 method for class 'summary.paths'
print(x, ...)
```

**Arguments**

object	an object of class paths returned by the <a href="#">paths</a> function.
...	additional arguments to be passed to printCoefmat for the print method
x	an object of class summary.paths

**Details**

print.summary.paths tries to smartly format the point estimates and confidence intervals, and provides 'significance stars' through the [printCoefmat](#) function.

It also prints out the names of the treatment, outcome, mediator variables as well as pretreatment covariates, which are extracted from the formulas argument of the call to paths so that users can verify if the model formulas have been correctly specified.

**Value**

An object of class summary.paths, which is a list containing the call, varnames, formulas, classes, args, ps\_formula, ps\_class, ps\_args, nboot, conf\_level components from the paths object, plus

**nobs** number of observations in data

**estimates** a list containing four matrices, corresponding to effect estimates obtained using the pure imputation estimator and the imputation-based weighting estimator, each with Type I and Type II decompositions. Each matrix contains the point estimates, standard errors, and confidence intervals of the total effect, direct effect, and each individual indirect effect for the corresponding decomposition. The elements in each matrix are extracted from the paths object.

**See Also**

[paths](#), [print.paths](#), [plot.paths](#)

## Examples

```

# **For illustration purposes a small number of bootstrap replicates are used**

data(tatar)

m1 <- c("trust_g1", "victim_g1", "fear_g1")
m2 <- c("trust_g2", "victim_g2", "fear_g2")
m3 <- c("trust_g3", "victim_g3", "fear_g3")
mediators <- list(m1, m2, m3)

formula_m0 <- annex ~ kulak + prosoviet_pre + religiosity_pre + land_pre +
  orchard_pre + animals_pre + carriage_pre + otherprop_pre + violence
formula_m1 <- update(formula_m0, ~ . + trust_g1 + victim_g1 + fear_g1)
formula_m2 <- update(formula_m1, ~ . + trust_g2 + victim_g2 + fear_g2)
formula_m3 <- update(formula_m2, ~ . + trust_g3 + victim_g3 + fear_g3)
formula_ps <- violence ~ kulak + prosoviet_pre + religiosity_pre +
  land_pre + orchard_pre + animals_pre + carriage_pre + otherprop_pre

#####
# Causal Paths Analysis using GLM
#####

# outcome models
glm_m0 <- glm(formula_m0, family = binomial("logit"), data = tatar)
glm_m1 <- glm(formula_m1, family = binomial("logit"), data = tatar)
glm_m2 <- glm(formula_m2, family = binomial("logit"), data = tatar)
glm_m3 <- glm(formula_m3, family = binomial("logit"), data = tatar)
glm_ymodels <- list(glm_m0, glm_m1, glm_m2, glm_m3)

# propensity score model
glm_ps <- glm(formula_ps, family = binomial("logit"), data = tatar)

# causal paths analysis using glm
# note: For illustration purposes only a small number of bootstrap replicates are used
paths_glm <- paths(a = "violence", y = "annex", m = mediators,
  glm_ymodels, ps_model = glm_ps, data = tatar, nboot = 3)
# plot total, direct, and path-specific effects
summary(paths_glm)

```

---

tatar

*The Legacy of Political Violence among Crimean Tatars*


---

## Description

A dataset of 427 Crimean Tatars including variables on ancestor victimization, political identities of first-, second- and third-generation respondents, and political attitudes toward Russia's annexation of Crimea (Lupu and Peisakhin 2017).

**Usage**

tatar

**Format**

A data frame with 427 rows and 19 columns:

**kulak** Whether the first-generation respondent had close relatives subject to dekulakization. 0: no; 1: yes.

**prosoviet\_pre** Whether the first-generation respondent's close relatives privately supported or opposed Soviet authorities. 1: opposed; 2: indifferent; 3: supported.

**religiosity\_pre** How important it was for the first-generation respondent's family to follow Islamic customs and traditions while in deportation. 1: not at all important; 2: somewhat important; 3: very important.

**land\_pre** Whether the first-generation respondent's close relatives owned agricultural land. 0: no; 1: some; 2: a lot.

**orchard\_pre** Whether the first-generation respondent's close relatives owned orchards. 0: no; 1: yes.

**animals\_pre** Whether the first-generation respondent's close relatives owned pasture animals. 0: no; 1: some; 2: a lot.

**carriage\_pre** Whether the first-generation respondent's close relatives owned horse-drawn carriages. 0: no; 1: yes.

**otherprop\_pre** Whether the first-generation respondent's close relatives owned other substantial property. 0: no; 1: yes.

**violence** Whether the first-generation respondent had a family member who died due to poor conditions during the 1944-45 deportation to Crimea. 0: no; 1: yes.

**trust\_g1** Whether the first-generation respondent trusts Crimean Tatars more than Russians. 1: trusts Crimean Tatars more; 0: indifferent; -1: trusts Russians more.

**victim\_g1** Whether the first-generation respondent consider them or their close relatives victims of the Soviet political system. 0: no; 1: yes.

**fear\_g1** Whether the first-generation respondent started to fear concerning their future after the March referendum. 0: no; 1: yes.

**trust\_g2** The degree to which second-generation respondents trust Crimean Tatars more than Russians, ranging from -1 to 1 (averaged over multiple respondents).

**victim\_g2** The degree to which second-generation respondents consider them or their close relatives victims of the Soviet political system, ranging from 0 to 1 (averaged over multiple respondents).

**fear\_g2** The degree to which second-generation respondents started to fear concerning their future after the March referendum, ranging from 0 to 1 (averaged over multiple respondents).

**trust\_g3** Whether the third-generation respondent trusts Crimean Tatars more than Russians. 1: trusts Crimean Tatars more; 0: indifferent; -1: trusts Russians more.

**victim\_g3** Whether the third-generation respondent considers them or their close relatives victims of the Soviet political system. 0: no; 1: yes.

**fear\_g3** Whether the third-generation respondent started to fear concerning their future after the March referendum. 0: no; 1: yes.

**annex** Whether the third-generation respondent supports Russia's annexation of Crimea. 0: no; 1: yes.

## References

Lupu, Noam and Leonid Peisakhin. 2017. "The Legacy of Political Violence across Generations." *American Journal of Political Science* 61(4):836-851.

---

welfare

*Issue Framing and Support for Welfare Reform*

---

## Description

A dataset of 213 Danish students containing variables on gender, education, political interest, ideology, political knowledge, extremity of political values, treatment assignment (job/poor frame), beliefs about why some people receive welfare benefits, perceived importance of different considerations related to welfare policy, and support for a proposed welfare reform (Slothuus 2008; Imai and Yamamoto 2013).

## Usage

welfare

## Format

A data frame with 213 rows and 15 columns:

**gender1** Gender. 0: Female; 1: Male.

**educ1** Level of education. 1: the municipal primary and lower secondary school before ninth form; 2: the municipal primary and lower secondary school after ninth or tenth form; 3: Basic schooling; 4: Vocational education; 5: Higher preparatory examination course student; 6: Upper secondary school student; 7: Higher commercial examination student; 8: Higher technical examination student; 9: Short-term further education; 10: Medium-term further education; 11: Long-term further education; 12: Foreign education; 13: Else.

**polint1** Political interest, measured on a 0-4 scale.

**ideo1** Ideological self-placement on a 1-8 scale. A larger value denotes a more right-wing position.

**know1** Political knowledge. 1: low; 2: medium; 3: high.

**value1** Extremity of political values. 0: moderate. 1: extreme.

**ttt** Treatment assignment. Whether the respondent read a newspaper article that highlighted the positive effect of welfare reform on job creation (1) versus one emphasizing its negative effect on the poor (0).

**W1** The degree to which the respondent attributes welfare reciprocity to internal factors, measured on a 0-1 scale.

- W2** The degree to which the respondent attributes welfare reciprocity to external factors, measured on a 0-1 scale.
- M1** How important the respondent thinks that there should always be an incentive for people to take a job instead of receiving welfare benefits, measured on a 0-1 scale.
- M2** How important the respondent thinks that nobody should live in poverty, measured on a 0-1 scale.
- M3** How important the respondent thinks that government expenditures on welfare benefits should not be too expensive, measured on a 0-1 scale.
- M4** How important the respondent thinks that no defrauder should receive welfare benefits, measured on a 0-1 scale.
- M5** How important the respondent thinks that the unemployed should have benefit rates making it possible to maintain a decent standard of living conditions, measured on a 0-1 scale.
- Y** Support for the proposed welfare reform, measured on a seven-point scale.

### References

- Slothuus, Rune. 2008. "More than Weighting Cognitive Importance: A Dual-process Model of Issue Framing Effects." *Political Psychology* 29(1):1-28.
- Imai, Kosuke and Teppei Yamamoto. 2013. "Identification and Sensitivity Analysis for Multiple Causal Mechanisms: Revisiting Evidence from Framing Experiments." *Political Analysis* 21(2):141-171.

# Index

## \* datasets

tatar, 13

welfare, 15

gbm, 3

glm, 3

lm, 3

paths, 2, 3, 6, 10–12

pbart, 3

plot.paths, 4, 5, 12

plot.sens, 7, 11

pred, 8

print.paths, 12

print.paths (paths), 2

print.summary.paths (summary.paths), 11

printCoefmat, 12

ps, 3

sens, 4, 7, 9

summary.paths, 4, 11

tatar, 13

wbart, 3

welfare, 15