Package 'paths'

May 6, 2020

Type Package

Title An Imputation Approach to Estimating Path-Specific Causal
Effects

Version 0.1.0

Date 2020-05-03

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>). 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 (>= 3.6.0)

Suggests knitr, rmarkdown

Imports BART, boot, gbm, ggplot2, metR, parallel, pryr, stats, twang,
tidyr

RoxygenNote 7.1.0

NeedsCompilation no

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

Maintainer Minh Trinh <mdtrinh@mit.edu>

Repository CRAN

Date/Publication 2020-05-06 20:10:03 UTC

R topics documented:

---

paths-package   *paths: An Imputation Approach to Estimating Path-specific Causal Effects*

Description

This package implements an imputation approach to estimating path-specific causal effects as detailed in Zhou and Yamamoto (2020). Statistical inference is conducted using the nonparametric bootstrap.

Author(s)

Minh Trinh, Massachusetts Institute of Technology, <mdtrinh@mit.edu>; Teppei Yamamoto, Massachusetts Institute of Technology, <teppei@mit.edu>; Xiang Zhou, Harvard University, <xiang_zhou@fas.harvard.edu>

References

Zhou, Xiang and Teppei Yamamoto. 2020. "Tracing Causal Paths from Experimental and Observational Data".

---

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

a	a character string indicating the name of the treatment variable. The treatment should be a binary variable taking either 0 or 1.
y	a character string indicating the name of the outcome variable.
m	a list of K character vectors indicating the names of K causally ordered mediators $M_1, \ldots, M_K$.
models	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.

- the first element is a baseline model of the outcome conditional on treatment and pretreatment confounders.
- the kth element is an outcome model conditional on treatment, pretreatment confounders, and mediators $M_1, \ldots, M_{k-1}$.
- the last element is an outcome model conditional on treatment, pretreatment confounders, and all of the mediators, i.e., $M_1, \ldots, M_K$.

The fitted model objects can be of type lm, glm, gbm, wbart, or pbart.

ps_model	an optional propensity score model for treatment. It can be of type glm, gbm, ps, or pbart. When it is provided, the imputation-based weighting estimator is also used to compute path-specific causal effects.
data	a data frame containing all variables.
nboot	number of bootstrap iterations for estimating confidence intervals. Default is 500.
conf_level	the confidence level of the returned two-sided confidence intervals. Default is 0.95.
...	additional arguments to be passed to boot::boot, e.g. parallel and ncpus. For the print method, additional arguments to be passed to print.default
x	a fitted model object returned by the paths function.
digits	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 $M_1, \ldots, M_K$ based on the pure imputation estimator.

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

varnames a list of character strings indicating the names of the pretreatment confounders (X), treatment(A), mediators ($M_1, \ldots, M_K$), 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 Teppei 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_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)


####################################################
# 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 [paths](#) 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 decomp != both.
...	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 [sens](#) 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 $M_1, \ldots, M_k$. 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 $M_1, \ldots, M_k$.
other	a named list indicating the values at which other sensitivity analysis parameters, namely, `eta_k-1` and `gamma_k-1`, are held. This is needed only when the bias formula involves η_{k-1} and γ_{k-1} as well as η_k and γ_k. The default is to set both η_{k-1} and γ_{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)
```

---

pred   *Obtaining predicted values from fitted models*

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

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

Value

a vector of expected outcomes for `newdata`

---

sens	*Sensitivity Analysis for Unobserved Confounding on Path-Specific Causal Effects*

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 γ and η 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 paths function.
`confounded`	a character string indicating the mediator whose relationship with the outcome is potentially confounded. One of $\{"M1", "M2", \dots\}$.
`estimand`	a character string indicating the estimand whose sensitivity to unobserved confounding is being investigated. One of $\{"M1", "M2", \dots\}$ 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 γ 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 η 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 γ and η, 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)
```

---

summary.paths   *Summarizing Output from Causal Paths Analysis*

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 [paths](#) 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 recipiency to internal factors, measured on a 0-1 scale.

W2 The degree to which the respondent attributes welfare recipiency 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