Package ‘csurvey’

December 4, 2025

Type Package

Title Constrained Regression for Survey Data

Version 1.14

Date 2025-12-04

Maintainer Xiyue Liao <xliao@sdsu.edu>

Description Domain mean estimation with monotonicity or block monotone constraints. See Xu X, Meyer MC and Opsomer JD (2021)<doi:10.1016/j.jspi.2021.02.004> for more details.

License GPL (>= 2)

Depends survey (>= 4.2-1), cgam (>= 1.27), R (>= 4.1.1)

Imports coneproj, purrr, data.table, stats, igraph, graphics, grDevices, MASS, Matrix, tibble, ggplot2, dplyr, zeallot

Suggests knitr, NHANES, rmarkdown, testthat (>= 3.0.0)

NeedsCompilation no

ByteCompile true

Repository CRAN

Author Xiyue Liao [aut, cre] (ORCID: <https://orcid.org/0000-0002-4508-9219>)

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

Date/Publication 2025-12-04 12:30:07 UTC

Contents

block.Ord *Specify a Block Monotonic Shape-Restriction in a CSVY Formula*

Description

A symbolic routine to define that a vector of domain means follows a monotonic ordering in a predictor in a formula argument to csvy. This is the unsmoothed version.

Usage

```
block.Ord(x, order = NULL, numknots = 0, knots = 0, space = "E")
```

Arguments

`x` A numeric predictor which has the same length as the response vector.

`order` A $1 \times M$ vector defining the order of domains when the shape constraint is block ordering.

`numknots` The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.

`knots` The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.

`space` A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "E".

Value

The vector x with five attributes, i.e., name: the name of x; shape: 9("block ordering"); numknots: the numknots argument in "block.Ord"; knots: the knots argument in "block.Ord"; space: the space argument in "block.Ord".

Author(s)

Xiyue Liao

See Also

`csvy`

csvy *Estimation of Domain Means with Monotonicity or Convexity Constraints*

Description

The csvy function performs design-based domain mean estimation with monotonicity and block-monotone shape constraints.

For example, in a one dimensional situation, we assume that $\bar{y}_{U_t}$ are non-decreasing over T domains. If this monotonicity is not used in estimation, the population domain means can be estimated by the Horvitz-Thompson estimator or the Hajek estimator. To use the monotonicity information, this csvy function starts from the Hajek estimates $\bar{y}_{S_t} = (\sum_{k \in S_t} y_k/\pi_k)/N_t$ and the isotonic estimator $(\hat{\theta}_1, \ldots, \hat{\theta}_T)^T$ minimizes the weighted sum of squared deviations from the sample domain means over the set of ordered vectors; that is, $\hat{\boldsymbol{\theta}}$ is the minimizer of $(\tilde{\boldsymbol{y}}_S - \boldsymbol{\theta})^T \boldsymbol{W}_s (\tilde{\boldsymbol{y}}_S - \boldsymbol{\theta})$ subject to $\boldsymbol{A\theta} \geq \boldsymbol{0}$, where $\boldsymbol{W}_S$ is the diagonal matrix with elements $\hat{N}_1/\hat{N}, \ldots, \hat{N}_D/\hat{N}$, and $\hat{N} = \sum_{t=1}^{T} \hat{N}_t$ and $\boldsymbol{A}$ is a $m \times T$ constraint matrix imposing the monotonicity constraint.

Domains can also be formed from multiple covariates. In that case, a grid will be used to represent the domains. For example, if there are two predictors x_1 and x_2, and x_1 has values on D_1 domains: $1, \ldots, D_1$, x_2 has values on D_2 domains: $1, \ldots, D_2$, then the domains formed by x_1 and x_2 will be a $D_1 \times D_2$ by 2 grid.

To get $100(1-\alpha)\%$ approximate confidence intervals or surfaces for the domain means, we apply the method in Meyer, M. C. (2018). $\hat{p}_J$ is the estimated probability that the projection of y_s onto $\mathcal{C}$ lands on $\mathcal{F}_{\mathcal{J}}$, and the $\hat{p}_J$ values are obtained by simulating many normal random vectors with estimated domain means and covariance matrix I, where I is a $M \times M$ matrix, and recording the resulting sets J.

The user needs to provide a survey design, which is specified by the svydesign function in the survey package, and also a data frame containing the response, predictor(s), domain variable, sampling weights, etc. So far, only stratified sampling design with simple random sampling without replacement (STSI) is considered in the examples in this package.

Note that when there is any empty domain, the user must specify the total number of domains in the nD argument.

Usage

```
csvy(formula, design, family=stats::gaussian(), multicore=getOption("csurvey.multicore"),
  level=0.95, n.mix=100L, test=FALSE, subset=NULL)
## S3 method for class 'csvy'
summary(object,...)
## S3 method for class 'csvy'
vcov(object,...)
## S3 method for class 'csvy'
coef(object,...)
## S3 method for class 'csvy'
confint(object, parm=NULL, level = 0.95, type = c("link", "response"),...)
## S3 method for class 'csvy'
```

```
predict(object, newdata = NULL, type = c("link", "response"),
    se.fit = TRUE, level = 0.95, n.mix = 100,...)
```

Arguments

`formula` A formula object which gives a symbolic description of the model to be fitted. It has the form "response ~ predictor". The response is a vector of length n. A predictor can be a non-parametrically modelled variable with a monotonicity or convexity restriction, or a combination of both. In terms of a non-parametrically modelled predictor, the user is supposed to indicate the relationship between the domain mean and a predictor x in the following way:

Assume that μ is the vector of domain means and x is a predictor:

incr(x): μ is increasing in x.

decr(x): μ is decreasing in x.

block.Ord(x): μ is has a block ordering in x.

`design` A survey design, which must be specified by the svydesign routine in the survey package.

`family` A parameter indicating the error distribution and link function to be used in the model. It can be a character string naming a family function or the result of a call to a family function. This is borrowed from the glm routine in the stats package. There are four families used in csvy: Gaussian, binomial, poisson, and Gamma.

`multicore` A parameter retrieving the current global option for "csurvey.multicore" and assigns its value to the multicore variable, allowing the function to respect a user-defined setting for parallel processing behavior across the package.

`level` Confidence level of the approximate confidence surfaces. The default is 0.95.

`n.mix` The number of simulations used to get the approximate confidence intervals or surfaces. If n.mix = 0, no simulation will be done and the face of the final projection will be used to compute the covariance matrix of the constrained estimate. The default is n.mix = 100L.

`test` A logical scalar. If test == TRUE, then the p-value for the test $H_0 : \theta$ is in V versus $H_1 : \theta$ is in C is returned. C is the constraint cone of the form $\{\beta : A\beta \geq 0\}$, and V is the null space of A. The default is test = FALSE.

`subset` Expression to select a subpopulation.

`...` Extra arguments.

The `coef` function returns estimated systematic component of a csvy object.

The `confint` function returns the confidence interval of a csvy object. If type = "response", then the interval is for the mean; if type = "link", then the interval is for the systematic component.

`parm` An argument in the generic confint function in the stats package. For now, this argument is not in use.

The following arguments are used in the `predict` function.

`object` A csvy object.

`newdata` A data frame in which to look for variables with which to predict. If omitted, the fitted values are used.

`type` If the response is Gaussian, type = "response" and type = "link" give the predicted mean; if the response is binomial, poisson or Gamma, type = "response" gives the predicted mean, and type = "link" gives the predicted systematic component.

`se.fit` Logical switch indicating if confidence intervals are required.

Details

For binomial and Poisson families use `family=quasibinomial()` and `family=quasipoisson()` to avoid a warning about non-integer numbers of successes. The 'quasi' versions of the family objects give the same point estimates and standard errors and do not give the warning.

`predict` gives fitted values and sampling variability for specific new values of covariates. When `newdata` are the population mean it gives the regression estimator of the mean, and when `newdata` are the population totals and `total` is specified it gives the regression estimator of the population total. Regression estimators of mean and total can also be obtained with `calibrate`.

Value

The output is a list of values used for estimation, inference and visualization. Main output include:

`survey.design` The survey design used in the model.

`etahat` Estimated shape-constrained domain systematic component.

`etahatu` Estimated unconstrained domain systematic component.

`muhat` Estimated shape-constrained domain means.

`muhatu` Estimated unconstrained domain means.

`lwr` Approximate lower confidence band or surface for the shape-constrained domain mean estimate.

`upp` Approximate upper confidence band or surface for the shape-constrained domain mean estimate.

`lwru` Approximate lower confidence band or surface for the unconstrained domain mean estimate.

`uppu` Approximate upper confidence band or surface for the unconstrained domain mean estimate.

`amat` The $k \times M$ constraint matrix imposing shape constraints in each dimension, where M is the total number of domains.

`grid` A $M \times p$ grid, where p is the total number of predictors or dimensions.

`nd` A vector of sample sizes in all domains.

`Ds` A vector of the number of domains in each dimension.

`acov` Constrained mixture covariance estimate of domain means.

`cov.un` Unconstrained covariance estimate of domain means.

`CIC` The cone information criterion proposed in Meyer(2013a). It uses the "null expected degrees of freedom" as a measure of the complexity of the model. See Meyer(2013a) for further details of cic.

`CIC.un` The cone information criterion for the unconstrained estimator.

`zeros_ps` Index of empty domain(s).

`nd` Sample size of each domain.

`pval` p-value of the one-sided test.

`family` The family parameter defined in a csvy formula.

`df.residual` The observed degree of freedom for the residuals of a csvy fit.

`df.null` The degree of freedom for the null model of a csvy fit.

`domain` Index of each domain in the data set contained in the survey.design object.

`null.deviance` The deviance for the null model of a csvy fit.

`deviance` The residual deviance of a csvy fit.

Author(s)

Xiyue Liao

References

Xu, X. and Meyer, M. C. (2021) One-sided testing of population domain means in surveys.

Oliva, C., Meyer, M. C., and Opsomer, J.D. (2020) Estimation and inference of domain means subject to qualitative constraints. *Survey Methodology*

Meyer, M. C. (2018) A Framework for Estimation and Inference in Generalized Additive Models with Shape and Order Restrictions. *Statistical Science* ***33(4)*** 595–614.

Wu, J., Opsomer, J.D., and Meyer, M. C. (2016) Survey estimation of domain means that respect natural orderings. *Canadian Journal of Statistics* ***44(4)*** 431–444.

Meyer, M. C. (2013a) Semi-parametric additive constrained regression. *Journal of Nonparametric Statistics* ***25(3)***, 715.

Lumley, T. (2004) Analysis of complex survey samples. *Journal of Statistical Software* ***9(1)*** 1–19.

See Also

`plotpersp`, to create a 3D Plot for a csvy Object

`incr`, to specify an increasing shape-restriction in a csvy Formula

`decr`, to specify an decreasing shape-restriction in a csvy Formula

Examples

```
# Example 1: monotonic in 1 predictor
data(nhdat2, package = 'csurvey')
#specify the design:
dstrat <- svydesign(ids = ~id, strata = ~str, data = nhdat2, weight = ~wt)
```

```
#uncomment to use parallel computing:
#options(csurvey.multicore=TRUE)
#mixture-variance-covariance matrix is simulated
set.seed(1)
ans <- csvy(chol ~ incr(age), design = dstrat, n.mix=5)

#check the constrained fit vs the unconstrained fit
summary(ans)
plot(ans, type = 'both')

## Not run:
# Example 2: monotonic in 2 predictors and unconstrained in a 3rd predictor
data(nhdat2, package = 'csurvey')
#specify the design:
dstrat <- svydesign(ids = ~id, strata = ~str, data = nhdat2, weight = ~wt)

#use parallel computing:
#options(csurvey.multicore=TRUE)
#mixture-variance-covariance matrix is simulated
#Average cholestorel level increases in age, waist, and unconstrained in income
set.seed(1)
ans <- csvy(chol ~ incr(age)*incr(wcat)*icat, design = dstrat, test=FALSE, n.mix=5)

#visualize the constrained estimation with confidence bands
plot(ans, x1='icat', x2='wcat')

#create control object
ctl <- plot_csvy_control(ribbon_fill = "pink",x1lab = 'income',x2lab = 'waist')

plot(ans, x1='icat', x2='wcat', control=ctl)

## End(Not run)

# Example 3: example with a binomial response
library(NHANES)
library(survey)
data(NHANES)

nh <- subset(NHANES, !is.na(Education) & !is.na(BMI) & !is.na(Weight))
nh$DiabetesBin <- as.integer(nh$Diabetes == "Yes")
nh$BMIgroup <- cut(nh$BMI, breaks = c(0, 18.5, 25, 30, 35, 40, Inf), labels = FALSE)

# specify the design
dsgn <- svydesign(ids = ~1, strata = ~BMIgroup, weights = ~Weight, data = nh)

ans <- csvy(DiabetesBin ~ decr(Education) * incr(BMIgroup), design = dsgn,
  family = quasibinomial(link='logit'), n.mix=5)

summary(ans)

plot(ans, x1 = 'BMIgroup', x2 = 'Education')

ctl <- plot_csvy_control(
  x1size = 1.5, x2size = 2,
  angle = 45, hjust = .3)

plot(ans, x1 = 'Education', x2 = 'BMIgroup', control = ctl)
```

nhdat — *Subset of NHANES 2009 to 2010 Data for Constrained Survey Analysis*

Description

A structured subset of the 2009 to 2010 NHANES data designed for illustrating constrained survey estimation methods.

Usage

```
data(nhdat)
```

Format

A data frame with 1,680 observations on the following 7 variables:

- `id` Cluster identifier derived from NHANES sequence number (`SEQN`).
- `chol` Binary indicator of high total cholesterol: 1 if total cholesterol > 200 mg/dL, 0 otherwise. Derived from `LBXTC` in `TCHOL_F.XPT`.
- `wcat` Four-level ordinal variable for waist to height ratio category, based on `BMXWAIST` and `BMXHT` from `BMX_F.XPT`.
- `gender` Gender of the participant (1 = male, 2 = female), from `RIAGENDR` in `DEMO_F.XPT`.
- `age` Age in years (continuous), from `RIDAGEYR` in `DEMO_F.XPT`.
- `wt` Sampling weight within strata, based on `WTINT2YR` from `DEMO_G.XPT`.
- `str` Stratum identifier, based on `SDMVPSU` from `DEMO_G.XPT`.

Details

This subset includes participants aged 21 to 45 years, selected for illustrating the estimation of the probability of high cholesterol using order constrained survey methods.

Source

National Center for Health Statistics. NHANES 2009 to 2010 Public Use Data Files.\ https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?BeginYear=2009 `nhdat2` *Subset of NHANES 2009 to 2010 Data for Constrained Survey Analysis*

Description

A structured subset of the 2009 to 2010 NHANES data designed for illustrating constrained survey estimation methods.

Usage

```
data(nhdat2)
```

Format

A data frame with 1,933 observations and the following 8 variables:

`id` An identification vector specifying cluster ids from largest level to smallest level, derived from NHANES sequence number `SEQN`.

`chol` Total cholesterol, measured in mg/dL. This variable is derived from `LBXTC` in the laboratory file `TCHOL_F.XPT`.

`wcat` A 4 level ordinal categorical variable representing waist to height ratio categories, computed from `BMXWAIST` (waist circumference in cm) and `BMXHT` (height in cm) in the body measures file `BMX_F.XPT`.

`icat` A 4 level ordinal categorical variable. It categorizes income based on the ratio of family income to the federal poverty level (INDFMPIR), with cutpoints at 0.75, 1.38, and 3.5 to reflect meaningful policy thresholds. It is derived from `INDFMIN2` in the demographics file `DEMO_F.XPT`.

`gender` Gender of the participant, with values 1 (male) and 2 (female), derived from `RIAGENDR` in `DEMO_F.XPT`.

`age` Age in years (continuous), derived from `RIDAGEYR` in the demographics file `DEMO_F.XPT`.

`wt` Sampling weight within each stratum, derived from (`WTINT2YR`) from `DEMO_G.XPT`.

`str` Stratum identifier, derived from (`SDMVPSU`) from `DEMO_G.XPT`.

Details

This subset includes participants aged 21 through 45, selected to demonstrate estimation of domain means using order constrained methods.

Source

National Center for Health Statistics. NHANES 2009 to 2010 Public Use Data Files. https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?BeginYear=2009 plot.csvy *Plot method for csvy objects Generates a diagnostic or summary plot from a fitted "csvy" object. Supports both single-factor and two-factor visualization. Aesthetic settings can be customized using* plot_csvy_control()*.*

Description

Plot method for csvy objects Generates a diagnostic or summary plot from a fitted "csvy" object. Supports both single-factor and two-factor visualization. Aesthetic settings can be customized using plot_csvy_control().

Usage

```
## S3 method for class 'csvy'
plot(
  x,
  x1 = NULL,
  x2 = NULL,
  domains = NULL,
  type = c("constrained", "unconstrained", "both"),
  control = plot_csvy_control(),
  ...
)
```

Arguments

x	An object of class "csvy".
x1	Optional. Name of the first factor to display in two-factor plots. Defaults to the first added variable.
x2	Optional. Name of the second factor to display in two-factor plots. Defaults to the second added variable.
domains	Optional. A data frame containing some domain(s) to be emphasized on the plot. Defaults to be NULL.
type	Character string, either "constrained", "unconstrained", or "both". Defaults to "constrained".
control	A list of display options returned by plot_csvy_control. Defaults to plot_csvy_control().
...	Additional arguments passed to ggplot2::geom_line() or geom_point(), such as linewidth, size, etc.

Value

A ggplot2 object.

See Also

plot_csvy_control for a full list of customizable settings.

Examples

```
# plot.csvy(fit)
# plot.csvy(fit, x1 = "education", x2 = "region", control = plot_csvy_control(x1lab = "Education"))
```

---

plotpersp_csvy_control

Create a Control List for plotpersp.csvy

Description

Constructs a list of control parameters for use with plotpersp.csvy. It extends the default settings from plotpersp_control with options specific to csvy plots.

Usage

```
plotpersp_csvy_control(surface = c("C", "U"), x1nm = NULL, x2nm = NULL, categ = NULL,
  col = NULL, random = FALSE, ngrid = 12, xlim = NULL, ylim = NULL, zlim = NULL,
  xlab = NULL, ylab = NULL, zlab = NULL, th = NULL, ltheta = NULL, main = NULL,
  sub = NULL, ticktype = "simple", ci = c("none", "lwr", "up", "both"),
  cex = 1, categnm = NULL, type = c("response", "link"),
  cex.main = 0.8, box = TRUE, axes = TRUE, nticks = 5, palette = NULL,
  NCOL = NULL, transpose = FALSE)
```

Arguments

surface	Plot the constrained ("C") or unconstrained ("U") fit.
x1nm, x2nm	Character strings naming the predictor variables for x and y axes.
categ	Optional character string naming a categorical covariate to stratify plots.
col	Color(s) for surfaces. Can be a palette or custom colors.
random	If TRUE, colors are assigned randomly.
ngrid	Number of grid points along each axis.
xlim, ylim, zlim	Optional limits for x, y, z axes.
xlab, ylab, zlab	Axis labels.
th, ltheta	Viewing and lighting angles for the plot.
main, sub	Plot title and subtitle.
ticktype	Type of ticks: "simple" or "detailed".
ci	Confidence interval display mode: "none", "lwr", "up", "both".
cex	Scaling factor for labels.
categnm	Labels for each level of the categorical covariate.

`type` Scale of the surface: `"response"` (default) or `"link"`.

`cex.main` Scaling factor for main title text.

`box, axes` Logical flags to show box and axes.

`nticks` Number of tick marks along axes.

`palette` Vector of colors for multi-surface plots.

`NCOL` Number of columns in multi-panel layout.

`transpose` Logical; if `TRUE`, transpose layout of multi-panel plots.

Value

A named list of graphical settings for use in `plotpersp.csvy`.

See Also

`plotpersp.csvy`, `plotpersp_control`, `persp`

Examples

```
ctrl <- plotpersp_csvy_control(col = "topo", ci = "both", transpose = TRUE)
```

`plot_csvy_control` *Control settings for plot.csvy*

Description

Creates a list of graphical options to customize plots generated by `plot.csvy`. This includes labels, text sizes, colors, shapes, themes, and other display features.

Usage

```
plot_csvy_control(
  x1lab = NULL,
  x1_labels = TRUE,
  x2lab = NULL,
  x2_labels = TRUE,
  x3lab = NULL,
  x3_labels = TRUE,
  x4_vals = NULL,
  x4_labels = NULL,
  ynm = NULL,
  ci = TRUE,
  legend = TRUE,
  ylab = TRUE,
  x1size = 3.8,
  x2size = 3.8,
  constrained_color = "cornflowerblue",
  unconstrained_color = "#A3C99A",
  constrained_shape = 16,
  unconstrained_shape = 18,
  ribbon_fill = "lightblue",
  line_color = "black",
  base_theme = ggplot2::theme_minimal(),
  subtitle.size = 12,
  angle = 0,
  hjust = 0.1
)
```

Arguments

`x1lab` Character. Label for the first covariate (x-axis). Default is `NULL`.

`x1_labels` Logical or Character vector. Custom tick labels for the first covariate. Default is `TRUE`. If it is TRUE, tick labels will be created; if it is a character vector, then this vector will be used as the tick labels.

`x2lab` Character. Label for the second covariate (x-axis). Default is `NULL`.

`x2_labels` Logical or Character vector. Custom tick labels for the second covariate. Default is `TRUE`. Default is `TRUE`. If it is TRUE, tick labels will be created; if it is a character vector, then this vector will be used as the tick labels.

`x3lab` Character. Label for the third covariate, if used (for subtitles or grouping). Default is `NULL`.

`x3_labels` Logical or Character vector. Custom labels for the third covariate. Default is `TRUE`. If it is TRUE, labels will be created; if it is a character vector, then this vector will be used as the labels.

`x4_vals` Character vector. For models with more than three predictors, specifies the category to use for each additional predictor. Defaults to NULL, using the mode of each.

`x4_labels` Character vector. Custom labels for the fourth covariate. Default is `NULL`.

`ynm` Character. Label for the response. Default is `NULL`.

`ci` Logical. If TRUE, confidence bands are displayed. Defaults to TRUE.

`legend` Logical. If TRUE, legend for constrained fit or unconstrained fit will be shown. Defaults to TRUE.

`ylab` Logical. If TRUE, the response name will be shown on the y-axis. Defaults to TRUE.

`x1size` Numeric. Font size for annotation labels on the x1 axis. Default is `3.8`.

`x2size` Numeric. Font size for annotation labels on the x2 axis. Default is `3.8`.

`constrained_color`
Character. Color used to display fitted values and intervals from the constrained model. Default is `"cornflowerblue"`.

`unconstrained_color`
Character. Color used to display fitted values and intervals from the unconstrained model. Default is `"#A3C99A"`.

constrained_shape
: Integer. Shape code (used by ggplot2) for points corresponding to constrained fits. Default is 16 (solid circle).

unconstrained_shape
: Integer. Shape code for points from unconstrained fits. Default is 18 (solid diamond).

ribbon_fill
: Character. Fill color for the confidence ribbon around the fitted lines. Default is "lightblue".

line_color
: Character. Color of the lines connecting the fitted values. Default is "black".

base_theme
: A ggplot2 theme object used as the base plot theme. Default is ggplot2::theme_minimal().

subtitle.size
: Numeric. Font size for the subtitle text in the plot. Default is 12.

angle
: Numeric. Angle (in degrees) to rotate x-axis labels (typically for x1). Default is 0.

hjust
: Numeric. Horizontal justification for rotated x-axis labels. Default is .1 (right-aligned).

Value

A named list of graphical control parameters to be passed to the control argument in plot.csvy.

Examples

```
data(nhdat2, package = 'csurvey')
dstrat <- svydesign(ids = ~id, strata = ~str, data = nhdat2, weight = ~wt)
ans <- csvy(chol ~ incr(age), design = dstrat, n.mix=5)
#check the constrained fit vs the unconstrained fit with default aesthetics
ctl <- plot_csvy_control()
plot(ans, type = 'both', control = ctl)
#check the fit with user-defined aethetics
ctl <- list(x1lab = "Age Group", x2lab = "Region", constrained_color = "cornflowerblue",
unconstrained_color = "gray80", x1size = 4.5)
plot(ans, type = 'both', control = ctl)
```

Index