0.1 setx: Setting Explanatory Variable Values

Description

The setx command uses the variables identified in the formula generated by zelig and sets the values of the explanatory variables to the selected values. Use setx after zelig and before sim to simulate quantities of interest.

Usage

```
x.out <- setx(object, fn = list(numeric = mean, ordered = median,
                                others = mode),
              data = NULL, cond = FALSE, ...)
```

Arguments

object	the saved output from zelig.
fn	a list of functions to apply to three types of variables:
numeric	numeric variables are set to their mean by default, but you may select any mathematical function to apply to numeric variables.
ordered	ordered factors are set to their meidan by default, and most mathematical operations will work on them. If you select ordered = mean, however, setx will default to median with a warning.
other	variables may consist of unordered factors, character strings, or logical variables. The other variables may only be set to their mode. If you wish to set one of the other variables to a specific value, you may do so using ... below.
	In the special case fn = NULL, setx will return all of the observations without applying any function to the data.
data	a new data frame used to set the values of explanatory variables. If data = NULL (the default), the data frame called in zelig is used.
cond	a logical value indicating whether unconditional (default) or conditional (choose cond = TRUE) prediction should be performed. If you choose cond = TRUE, setx will coerce fn = NULL and ignore the additional arguments in .... If cond = TRUE and data = NULL, setx will prompt you for a data frame.
...	user-defined values of specific variables overwriting the default values set by the function fn. For example, adding var1 = mean(data$var1) or x1 = 12 explicitly sets the value of x1 to 12. In addition, you may specify one explanatory variable as a range of values, creating one observation for every unique value in the range of values.

Value

For unconditional prediction, `x.out` is a model matrix based on the specified values for the explanatory variables. For multiple analyses (i.e., when choosing the `by` option in `zelig`, `setx` returns the selected values calculated over the entire data frame. If you wish to calculate values over just one subset of the data frame, the 5th subset for example, you may use: `x.out <- setx(z.out[[5]])`

For conditional prediction, `x.out` includes the model matrix and the dependent variables. For multiple analyses (when choosing the `by` option in `zelig`), `setx` returns the observed explanatory variables in each subset.

Author(s)

Kosuke Imai <⟨kimai@princeton.edu⟩>; Gary King <⟨king@harvard.edu⟩>; Olivia Lau <⟨olau@fas.harvard.edu⟩>

See Also

The full Zelig manual may be accessed online at `http://gking.harvard.edu/zelig`.

Examples

```
# Unconditional prediction:
data(turnout)
z.out <- zelig(vote ~ race + educate, model = "logit", data = turnout)
x.out <- setx(z.out)
s.out <- sim(z.out, x = x.out)

# Unconditional prediction with all observations:
x.out <- setx(z.out, fn = NULL)
s.out <- sim(z.out, x = x.out)

# Unconditional prediction with out of sample data:
z.out <- zelig(vote ~ race + educate, model = "logit",
               data = turnout[1:1000,])
x.out <- setx(z.out, data = turnout[1001:2000,])
s.out <- sim(z.out, x = x.out)

# Using a user-defined function in fn:
## Not run:
quants <- function(x)
  quantile(x, 0.25)
x.out <- setx(z.out, fn = list(numeric = quants))
## End(Not run)
```

```
# Conditional prediction:
## Not run:
library(MatchIt)
data(lalonde)
match.out <- matchit(treat ~ age + educ + black + hispan + married +
                     nodegree + re74 + re75, data = lalonde)
z.out <- zelig(re78 ~ distance, data = match.data(match.out, "control"),
               model = "ls")
x.out <- setx(z.out, fn = NULL, data = match.data(match.out, "treat"),
              cond = TRUE)
s.out <- sim(z.out, x = x.out)
## End(Not run)
```