

0.1 `model.frame.multiple`: Extracting the “environment” of a model formula

Description

Use `model.frame.multiple` after `parse.par` to create a data frame of the unique variables identified in the formula (or list of formulas).

Usage

```
model.frame.multiple(formula, data, eqn = NULL, ...)
```

Arguments

<code>formula</code>	a list of formulas of class "multiple", returned from <code>parse.par</code>
<code>data</code>	a data frame containing all the variables used in <code>formula</code>
<code>eqn</code>	an optional character string or vector of character strings specifying the equations (specified in <code>describe.mymodel</code>) for which you would like to pull out the relevant variables.
<code>...</code>	additional arguments passed to <code>model.frame.default</code>

Value

The output is a data frame (with a `terms` attribute) containing all the unique explanatory and response variables identified in the list of formulas. By default, missing (NA) values are listwise deleted.

If `as.factor` appears on the left-hand side, the response variables will be returned as an indicator (0/1) matrix with columns corresponding to the unique levels in the factor variable.

If any formula contains more than one `tag` statement, `model.frame.multiple` will return the original variable in the data frame and use the `tag` information in the `terms` attribute only.

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>; Ferdinand Alimadhi <falimadhi@iq.harvard.edu>

See Also

`model.matrix.default`, `parse.formula` and the full Zelig manual at <http://gking.harvard.edu/zelig>

Examples

```
## Not run:  
data(sanction)  
formulae <- list(import ~ coop + cost + target,  
                  export ~ coop + cost + target)  
fml <- parse.formula(formulae, model = "bivariate.logit")  
D <- model.frame(fml, data = sanction)  
## End(Not run)
```