

Package ‘texreg’

March 3, 2017

Version 1.36.23

Date 2017-03-03

Title Conversion of R Regression Output to LaTeX or HTML Tables

Author Philip Leifeld

Maintainer Philip Leifeld <philip.leifeld@glasgow.ac.uk>

Description Converts coefficients, standard errors, significance stars, and goodness-of-fit statistics of statistical models into LaTeX tables or HTML tables/MS Word documents or to nicely formatted screen output for the R console for easy model comparison. A list of several models can be combined in a single table. The output is highly customizable. New model types can be easily implemented.

URL <http://github.com/leifeld/texreg/>

BugReports <https://github.com/leifeld/texreg/issues>

Suggests nlme, survival, network, ergm, lme4 (>= 1.0), btergm, Zelig (>= 5.0-16)

Depends R (>= 2.15.0)

Imports methods, grDevices, graphics, stats

Enhances AER, betareg, brglm, censReg, dynlm, eha, erer, fGarch, gamlss, gee, geopack, gmm, h2o, latentnet, lmtest, lqmm, MASS, mfx, mgcv, mlogit, mnlogit, MuMIn, nnet, ordinal, plm, pscl, quantreg, relevent, rms, robustbase, RSiena, sampleSelection, simex, sna, spdep, survey, systemfit, tergm, VGAM, xergm

License GPL-2 | GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-03 21:26:00

R topics documented:

texreg-package	2
coeftoString	3

createTexreg	3
extract	5
plotreg	21
print.texregTable	26
texreg	26

Index	36
--------------	-----------

texreg-package	<i>Conversion of statistical model output in R to LaTeX and HTML tables</i>
----------------	---

Description

texreg converts coefficients, standard errors, significance stars, and goodness-of-fit statistics of statistical models into LaTeX tables or HTML tables/MS Word documents or to nicely formatted screen output for the R console for easy model comparison. A list of several models can be combined in a single table. The output is highly customizable. New model types can be easily implemented.

Details

Several packages like apsrtable, memisc, outreg, stargazer, or xtable are available for typesetting R regression output as LaTeX tables. However, texreg supports more flexible handling of new model types, supports multiple models side-by-side, supports confidence intervals and standard errors alike, has many options for customization, and beside LaTeX output, it can also print tables to the R console screen, save them as HTML or MS Word documents, or plot them as coefficient plots. If several models are submitted, they are merged by row labels of the coefficients, and they are inserted into the final table as separate columns. The package works with report generation tools like Sweave or knitr (including options for LaTeX, HTML, and Markdown). To display citation information, execute `citation("texreg")`.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[texreg](#) [plotreg](#)

coeftoString *Convert coefficients into standardized strings*

Description

Convert coefficients into standardized strings.

Usage

```
coeftoString(x, lead.zero = FALSE, digits = 2)
```

Arguments

x	A numeric object, for example a coefficient resulting from a regression model.
lead.zero	If the number starts with "0" or "-0": should the zero be retained or removed? If true, the leading zero is kept.
digits	The number of decimal places to be used.

Details

This function converts numbers into strings and standardizes them according to some simple rules. The function is used by the `texreg` function.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[texreg-package](#) `texreg`

Examples

```
coeftoString(-0.345, lead.zero = FALSE) # this should give "-.34"
```

createTexreg *Create a texreg object*

Description

Create a `texreg` object with coefficients and GOF statistics.

Usage

```
createTexreg(coef.names, coef, se, pvalues = numeric(0), ci.low = numeric(0),  
             ci.up = numeric(0), gof.names = character(0), gof = numeric(0),  
             gof.decimal = logical(0), model.name = character(0))
```

Arguments

<code>coef.names</code>	A vector of coefficient names.
<code>coef</code>	The coefficient values.
<code>se</code>	The standard errors. This is optional if the <code>ci.low</code> and <code>ci.up</code> slots are filled.
<code>pvalues</code>	The p-values of the model. This is optional.
<code>ci.low</code>	Lower bound of confidence interval (the actual values, not the confidence level). This is optional as long as <code>se</code> is available, but if it is provided, <code>ci.up</code> must also be provided.
<code>ci.up</code>	Upper bound of confidence interval (the actual values, not the confidence level). This is optional as long as <code>se</code> is available, but if it is provided, <code>ci.low</code> must also be provided.
<code>gof.names</code>	A vector of names of the goodness-of-fit statistics.
<code>gof</code>	A vector of goodness-of-fit statistics.
<code>gof.decimal</code>	A vector of boolean/logical values indicating for each GOF statistic if decimal places shall be used. This is optional.
<code>model.name</code>	The name of the model. In some cases, models consist of two separate columns because two separate data-generating processes are modeled. In these cases, it may make sense to specify default names for the columns (that is, for each <code>texreg</code> object). This argument is optional.

Details

This function creates a `texreg` object. A `texreg` object contains information about coefficients, standard errors, p values (optional), and about goodness-of-fit statistics. Instead of standard errors and p values, a `texreg` object may also contain upper and lower bounds of a confidence interval. `texreg` objects are used by the `texreg` command to create LaTeX tables and other representations of the model results.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

References

Leifeld, Philip (2013). `texreg`: Conversion of Statistical Model Output in R to LaTeX and HTML Tables. *Journal of Statistical Software*, 55(8), 1-24. <http://www.jstatsoft.org/v55/i08/>.

See Also

[texreg-package](#) `texreg`

Examples

```
library(nlme) #load library for fitting linear mixed effects models
model <- lme(distance ~ age, data = Orthodont, random = ~ 1) #estimate model
coefficient.names <- rownames(summary(model)$tTable) #extract coefficient names
coefficients <- summary(model)$tTable[, 1] #extract coefficient values
```

```

standard.errors <- summary(model)$tTable[, 2] #extract standard errors
significance <- summary(model)$tTable[, 5] #extract p values

lik <- summary(model)$logLik #extract log likelihood
aic <- summary(model)$AIC #extract AIC
bic <- summary(model)$BIC #extract BIC
n <- nobs(model) #extract number of observations
gof <- c(aic, bic, lik, n) #create a vector of GOF statistics
gof.names <- c("AIC", "BIC", "Log Likelihood", "Num. obs.") #names of GOFs
decimal.places <- c(TRUE, TRUE, TRUE, FALSE) #the last one is a count variable

#create the texreg object
tr <- createTexreg(
  coef.names = coefficient.names,
  coef = coefficients,
  se = standard.errors,
  pvalues = significance,
  gof.names = gof.names,
  gof = gof,
  gof.decimal = decimal.places
)

```

extract

Extract coefficients and GOF measures from a statistical object

Description

Extract coefficients and GOF measures from a statistical object.

Usage

```

extract(model, ...)

## S4 method for signature 'aftreg'
extract(model, include.loglik = TRUE,
  include.lr = TRUE, include.nobs = TRUE, include.events = TRUE,
  include.trisk = TRUE, ...)

## S4 method for signature 'Arima'
extract(model, include.pvalues = FALSE,
  include.aic = TRUE, include.loglik = TRUE, ...)

## S4 method for signature 'ARIMA'
extract(model, include.pvalues = FALSE,
  include.aic = TRUE, include.aicc = TRUE, include.bic = TRUE,
  include.loglik = TRUE, ...)

## S4 method for signature 'averaging'
extract(model, use.ci = FALSE,

```

```
adjusted.se = FALSE, include.nobs = TRUE, ...)  
  
## S4 method for signature 'bam'  
extract(model, include.smooth = TRUE,  
  include.aic = TRUE, include.bic = TRUE,  
  include.loglik = TRUE, include.deviance = TRUE,  
  include.dev.expl = TRUE, include.dispersion = TRUE,  
  include.rsquared = TRUE, include.gcv = TRUE,  
  include.nobs = TRUE, include.nsmooth = TRUE, ...)  
  
## S4 method for signature 'betamfx'  
extract(model, include.pseudors = TRUE,  
  include.loglik = TRUE, include.nobs = TRUE, ...)  
  
## S4 method for signature 'betaor'  
extract(model, include.pseudors = TRUE,  
  include.loglik = TRUE, include.nobs = TRUE, ...)  
  
## S4 method for signature 'betareg'  
extract(model, include.precision = TRUE,  
  include.pseudors = TRUE, include.loglik = TRUE,  
  include.nobs = TRUE, ...)  
  
## S4 method for signature 'brglm'  
extract(model, include.aic = TRUE,  
  include.bic = TRUE, include.loglik = TRUE,  
  include.deviance = TRUE, include.nobs = TRUE, ...)  
  
## S4 method for signature 'btergm'  
extract(model, level = 0.95,  
  include.nobs = TRUE, ...)  
  
## S4 method for signature 'censReg'  
extract(model, include.aic = TRUE,  
  include.bic = TRUE, include.loglik = TRUE,  
  include.nobs = TRUE, ...)  
  
## S4 method for signature 'clm'  
extract(model, include.thresholds = TRUE,  
  include.aic = TRUE, include.bic=TRUE, include.loglik = TRUE,  
  include.nobs = TRUE, ...)  
  
## S4 method for signature 'clmm'  
extract(model, include.thresholds = TRUE,  
  include.loglik = TRUE, include.aic = TRUE,  
  include.bic = TRUE, include.nobs = TRUE,  
  include.groups = TRUE, include.variance = TRUE, ...)
```

```
## S4 method for signature 'clogit'
extract(model, include.aic = TRUE,
        include.rsquared = TRUE, include.maxrs = TRUE,
        include.events = TRUE, include.nobs = TRUE,
        include.missings = TRUE, ...)

## S4 method for signature 'coeftest'
extract(model, ...)

## S4 method for signature 'coxph'
extract(model, include.aic = TRUE,
        include.rsquared = TRUE, include.maxrs=TRUE,
        include.events = TRUE, include.nobs = TRUE,
        include.missings = TRUE, include.zph = TRUE, ...)

## S4 method for signature 'coxph.penall'
extract(model, include.aic = TRUE,
        include.rsquared = TRUE, include.maxrs = TRUE,
        include.events = TRUE, include.nobs = TRUE,
        include.missings = TRUE, include.zph = TRUE, ...)

## S4 method for signature 'dynlm'
extract(model, include.rsquared = TRUE,
        include.adjrs = TRUE, include.nobs = TRUE,
        include.fstatistic = FALSE, include.rmse = TRUE, ...)

## S4 method for signature 'ergm'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE, ...)

## S4 method for signature 'ergmm'
extract(model, include.bic = TRUE, ...)

## S4 method for signature 'ets'
extract(model, include.pvalues = FALSE,
        include.aic = TRUE, include.aicc = TRUE, include.bic = TRUE,
        include.loglik = TRUE, ...)

## S4 method for signature 'felm'
extract(model, include.nobs = TRUE,
        include.rsquared = TRUE, include.adjrs = TRUE,
        include.fstatistic = FALSE, ...)

## S4 method for signature 'fGARCH'
extract(model, include.nobs = TRUE,
        include.aic = TRUE, include.loglik = TRUE, ...)

## S4 method for signature 'forecast'
```

```
extract(model, ...)

## S4 method for signature 'gam'
extract(model, include.smooth = TRUE,
        include.aic = TRUE, include.bic = TRUE,
        include.loglik = TRUE, include.deviance = TRUE,
        include.dev.expl = TRUE, include.dispersion = TRUE,
        include.rsquared = TRUE, include.gcv = TRUE,
        include.nobs = TRUE, include.nsmooth = TRUE, ...)

## S4 method for signature 'gamlss'
extract(model, robust = FALSE,
        include.nobs = TRUE, include.nagelkerke = TRUE,
        include.gaic = TRUE, ...)

## S4 method for signature 'gee'
extract(model, robust = TRUE,
        include.dispersion = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'geeglm'
extract(model, include.scale = TRUE,
        include.correlation = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'gel'
extract(model, include.obj.fcn = TRUE,
        include.overidentification = FALSE, include.nobs = TRUE,
        overIdentTest = c("LR", "LM", "J "), ...)

## S4 method for signature 'glm'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'glmerMod'
extract(model, method = c("naive",
        "profile", "boot", "Wald"), level = 0.95, nsim = 1000,
        include.aic = TRUE, include.bic = TRUE, include.dic = FALSE,
        include.deviance = FALSE, include.loglik = TRUE,
        include.nobs = TRUE, include.groups = TRUE,
        include.variance = TRUE, ...)

## S4 method for signature 'glmmadmb'
extract(model, include.variance = TRUE,
        include.dispersion = TRUE, include.zero = TRUE,
        include.aic = TRUE, include.bic = TRUE,
        include.loglik = TRUE, include.nobs = TRUE,
        include.groups = TRUE, ...)
```



```
## S4 method for signature 'glmPQL'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.nobs = TRUE, include.groups = TRUE,
        include.variance = FALSE, ...)

## S4 method for signature 'glmrob'
extract(model, include.nobs = TRUE, ...)

## S4 method for signature 'gls'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.nobs = TRUE, ...)

## S4 method for signature 'gmm'
extract(model, include.obj.fcn = TRUE,
        include.overidentification = FALSE, include.nobs = TRUE, ...)

## S4 method for signature 'H2OBinomialModel'
extract(model, standardized = FALSE,
        include.mse = TRUE, include.rsquared = TRUE,
        include.logloss = TRUE, include.meanerror = TRUE,
        include.auc = TRUE, include.gini = TRUE,
        include.deviance = TRUE, include.aic = TRUE, ...)

## S4 method for signature 'hurdle'
extract(model, beside = FALSE,
        include.count = TRUE, include.zero = TRUE, include.aic = TRUE,
        include.loglik = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'ivreg'
extract(model, include.rsquared = TRUE,
        include.adjrs = TRUE, include.nobs = TRUE,
        include.fstatistic = FALSE, include.rmse = TRUE, ...)

## S4 method for signature 'lm'
extract(model, include.rsquared = TRUE,
        include.adjrs = TRUE, include.nobs = TRUE,
        include.fstatistic = FALSE, include.rmse = TRUE, ...)

## S4 method for signature 'lme'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.nobs = TRUE, include.groups = TRUE,
        include.variance = FALSE, ...)

## S4 method for signature 'lme4'
extract(model, method = c("naive",
```

```
"profile", "boot", "Wald"), level = 0.95, nsim = 1000,
include.aic = TRUE, include.bic = TRUE, include.dic = FALSE,
include.deviance = FALSE, include.loglik = TRUE,
include.nobs = TRUE, include.groups = TRUE,
include.variance = TRUE, ...)

## S4 method for signature 'lmerMod'
extract(model, method = c("naive",
  "profile", "boot", "Wald"), level = 0.95, nsim = 1000,
  include.aic = TRUE, include.bic = TRUE, include.dic = FALSE,
  include.deviance = FALSE, include.loglik = TRUE,
  include.nobs = TRUE, include.groups = TRUE,
  include.variance = TRUE, ...)

## S4 method for signature 'lmrob'
extract(model, include.nobs = TRUE, ...)

## S4 method for signature 'lnam'
extract(model, include.rsquared = TRUE,
  include.adjrs = TRUE, include.aic = TRUE, include.bic = TRUE,
  include.loglik = TRUE, ...)

## S4 method for signature 'logitmfx'
extract(model, include.nobs = TRUE,
  include.loglik = TRUE, include.deviance = TRUE,
  include.aic = TRUE, include.bic = TRUE, ...)

## S4 method for signature 'logitor'
extract(model, include.nobs = TRUE,
  include.loglik = TRUE, include.deviance = TRUE,
  include.aic = TRUE, include.bic = TRUE, ...)

## S4 method for signature 'lqmm'
extract(model, include.aic = TRUE,
  include.bic = TRUE, include.loglik = TRUE,
  include.nobs = TRUE, include.groups = TRUE,
  include.tau = FALSE, use.ci = FALSE, beside = TRUE, ...)

## S4 method for signature 'lrm'
extract(model, include.pseudors = TRUE,
  include.lr = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'maBina'
extract(model, ...)

## S4 method for signature 'mer'
extract(model, method = c("naive",
  "profile", "boot", "Wald"), level = 0.95, nsim = 1000,
```

```
    include.aic = TRUE, include.bic = TRUE, include.dic = FALSE,
    include.deviance = FALSE, include.loglik = TRUE,
    include.nobs = TRUE, include.groups = TRUE,
    include.variance = TRUE, ...)
```

```
## S4 method for signature 'mnlogit'
extract(model, include.aic = TRUE,
        include.loglik = TRUE, include.nobs = TRUE,
        include.groups = TRUE, include.intercept = TRUE,
        include.iterations = FALSE, beside = FALSE, ...)
```

```
## S4 method for signature 'mlogit'
extract(model, include.aic = TRUE,
        include.loglik = TRUE, include.nobs = TRUE, ...)
```

```
## S4 method for signature 'model.selection'
extract(model, include.loglik = TRUE,
        include.aicc = TRUE, include.delta = TRUE,
        include.weight = TRUE, include.nobs = TRUE, ...)
```

```
## S4 method for signature 'mtergm'
extract(model, include.nobs = TRUE,
        include.aic = TRUE, include.bic = TRUE, include.loglik = TRUE,
        ...)
```

```
## S4 method for signature 'multinom'
extract(model, include.pvalues = TRUE,
        include.aic = TRUE, include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = TRUE,
        levels = model$lev, beside = TRUE, ...)
```

```
## S4 method for signature 'negbin'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = TRUE, ...)
```

```
## S4 method for signature 'negbinirr'
extract(model, include.nobs = TRUE,
        include.loglik = TRUE, include.deviance = TRUE,
        include.aic = TRUE, include.bic = TRUE, ...)
```

```
## S4 method for signature 'negbinmfx'
extract(model, include.nobs = TRUE,
        include.loglik = TRUE, include.deviance = TRUE,
        include.aic = TRUE, include.bic = TRUE, ...)
```

```
## S4 method for signature 'netlogit'
extract(model, include.aic = TRUE,
```

```
include.bic = TRUE, include.deviance = TRUE,
include.nobs = TRUE, ...)

## S4 method for signature 'nlme'
extract(model, include.aic = TRUE,
include.bic = TRUE, include.loglik = TRUE,
include.nobs = TRUE, include.groups = TRUE,
include.variance = FALSE, ...)

## S4 method for signature 'nlmerMod'
extract(model, method = c("naive",
"profile", "boot", "Wald"), level = 0.95, nsim = 1000,
include.aic = TRUE, include.bic = TRUE, include.dic = FALSE,
include.deviance = FALSE, include.loglik = TRUE,
include.nobs = TRUE, include.groups = TRUE,
include.variance = TRUE, ...)

## S4 method for signature 'ols'
extract(model, include.nobs = TRUE,
include.rsquared = TRUE, include.adjrs = TRUE,
include.fstatistic = FALSE, include.lr = TRUE, ...)

## S4 method for signature 'pgmm'
extract(model, include.nobs = TRUE,
include.sargan = TRUE, include.wald = TRUE, ...)

## S4 method for signature 'phreg'
extract(model, include.loglik = TRUE,
include.lr = TRUE, include.nobs = TRUE, include.events = TRUE,
include.trisk = TRUE, ...)

## S4 method for signature 'plm'
extract(model, include.rsquared = TRUE,
include.adjrs = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'pmg'
extract(model, include.nobs = TRUE, ...)

## S4 method for signature 'poissonirr'
extract(model, include.nobs = TRUE,
include.loglik = TRUE, include.deviance = TRUE,
include.aic = TRUE, include.bic = TRUE, ...)

## S4 method for signature 'poissonmfx'
extract(model, include.nobs = TRUE,
include.loglik = TRUE, include.deviance = TRUE,
include.aic = TRUE, include.bic = TRUE, ...)
```

```
## S4 method for signature 'polr'
extract(model, include.thresholds = FALSE,
        include.aic = TRUE, include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'probitmfx'
extract(model, include.nobs = TRUE,
        include.loglik = TRUE, include.deviance = TRUE,
        include.aic = TRUE, include.bic = TRUE, ...)

## S4 method for signature 'rem.dyad'
extract(model, include.nvertices = TRUE,
        include.events = TRUE, include.aic = TRUE,
        include.aicc = TRUE, include.bic = TRUE, ...)

## S4 method for signature 'rlm'
extract(model, include.nobs = TRUE, ...)

## S4 method for signature 'rq'
extract(model, include.nobs = TRUE,
        include.percentile = TRUE, ...)

## S4 method for signature 'sarlm'
extract(model, include.nobs = TRUE,
        include.loglik = TRUE, include.aic = TRUE, include.lr = TRUE,
        include.wald = TRUE, ...)

## S4 method for signature 'sclm'
extract(model, include.thresholds = TRUE,
        include.aic = TRUE, include.bic = TRUE, include.loglik = TRUE,
        include.nobs = TRUE, ...)

## S4 method for signature 'selection'
extract(model, prefix = TRUE,
        include.selection = TRUE, include.outcome = TRUE,
        include.errors = TRUE, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.rsquared = TRUE, include.adjrs = TRUE,
        include.nobs = TRUE, ...)

## S4 method for signature 'sienaFit'
extract(model, include.iterations = TRUE,
        ...)

## S4 method for signature 'simex'
extract(model, jackknife = TRUE,
        include.nobs = TRUE, ...)
```

```
## S4 method for signature 'stergm'
extract(model, beside = FALSE,
        include.formation = TRUE, include.dissolution = TRUE,
        include.nvertices = TRUE, include.aic = FALSE,
        include.bic = FALSE, include.loglik = FALSE, ...)

## S4 method for signature 'survreg'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'survreg.penal'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'svyglm'
extract(model, include.aic = FALSE,
        include.bic = FALSE, include.loglik = FALSE,
        include.deviance = TRUE, include.dispersion = TRUE,
        include.nobs = TRUE, ...)

## S4 method for signature 'systemfit'
extract(model, include.rsquared = TRUE,
        include.adjrs = TRUE, include.nobs = TRUE, beside = FALSE,
        include.suffix = FALSE, ...)

## S4 method for signature 'texreg'
extract(model, ...)

## S4 method for signature 'tobit'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = FALSE,
        include.censnobs = TRUE, include.wald=TRUE, ...)

## S4 method for signature 'vglm'
extract(model, include.loglik = TRUE,
        include.df = TRUE, include.nobs = TRUE, ...)

## S4 method for signature 'weibreg'
extract(model, include.loglik = TRUE,
        include.lr = TRUE, include.nobs = TRUE,
        include.events = TRUE, include.trisk = TRUE, ...)

## S4 method for signature 'wls'
extract(model, include.nobs = TRUE, ...)
```

```

## S4 method for signature 'zelig'
extract(model, include.aic = TRUE,
        include.bic = TRUE, include.loglik = TRUE,
        include.deviance = TRUE, include.nobs = TRUE,
        include.rsquared = TRUE, include.adjrs = TRUE,
        include.fstatistic = TRUE, ...)

## S4 method for signature 'Zelig'
extract(model, ...)

## S4 method for signature 'zeroinfl'
extract(model, beside = FALSE,
        include.count = TRUE, include.zero = TRUE, include.aic = TRUE,
        include.loglik = TRUE, include.nobs = TRUE, ...)

```

Arguments

<code>model</code>	A statistical model object.
<code>adjusted.se</code>	If available: use adjusted rather than default standard errors?
<code>beside</code>	If available: should the model terms be arranged below each other or beside each other? For example, in a stergm model, the formation and dissolution coefficients can be arranged in two columns of the table.
<code>include.adjrs</code>	If available: should the adjusted R-squared be reported?
<code>include.aic</code>	If available: should Akaike's information criterion (AIC) be reported?
<code>include.aicc</code>	If available: should AICc be reported? This is a version of AIC with a correction for finite sample sizes.
<code>include.auc</code>	If available: should the area under the curve (AUC) be reported?
<code>include.bic</code>	If available: should the Bayesian information criterion (BIC) be reported?
<code>include.censnobs</code>	If available: should the total, right-censored, left-censored, and uncensored number of observations be reported?
<code>include.correlation</code>	If available: should the correlation parameter alpha and its standard error be reported (for geeglm models)?
<code>include.count</code>	If available: should the count model of a zero-inflated or hurdle regression be included in the coefficients block (before the zero-inflation or zero hurdle model)?
<code>include.delta</code>	If available: should the delta statistic be included?
<code>include.dev.expl</code>	If available: should the deviance explained be reported?
<code>include.deviance</code>	If available: should the deviance be reported?
<code>include.df</code>	If available: should the degrees of freedom be reported?
<code>include.dic</code>	If available: should the deviance information criterion (DIC) be reported?
<code>include.dispersion</code>	If available: should the dispersion or scale parameter be reported?

`include.dissolution` If available: should the coefficients for the dissolution phase in a STERGM be reported?

`include.errors` If available: should the error terms of a sample selection model be reported?

`include.events` If available: should the number of events be reported (in survival models)?

`include.formation` If available: should the coefficients for the formation phase in a STERGM be reported?

`include.fstatistic` If available: should the F statistic be reported?

`include.gaic` If available: should the Generalized Akaike's information criterion (GAIC) be reported?

`include.gini` If available: should the Gini coefficient be reported?

`include.gcv` If available: should the GCV score be reported (in GAMs)?

`include.groups` If available: should the number of groups in a mixed-effects model (or k alternatives in a multinomial choice model) be reported?

`include.intercept` If available: should the intercept be included in the GOF block?

`include.iterations` If available: should the number of iterations be included?

`include.loglik` If available: should the log-likelihood be reported?

`include.logloss` If available: should the log loss be reported?

`include.lr` If available: should the likelihood ratio test be reported?

`include.maxrs` If available: should the maximum possible R-squared be reported?

`include.meanerror` If available: should the mean per-class error be reported?

`include.missings` If available: should the number of missing observations be reported (in survival models)?

`include.mse` If available: should the mean square error be reported?

`include.nagelkerke` If available: should Nagelkerke's R-squared be reported?

`include.nobs` If available: should the number of observations be reported?

`include.nsmooth` If available: should the number of smooth terms be reported (in GAMs)?

`include.nvertices` If available: should the number of vertices be reported in a statistical network model?

`include.obj.fcn` If available: should the value of the objective function (= criterion function) be reported (for `gel` and `gmm` objects)? More precisely, this returns $E(g) \text{var}(g)^{-1} E(g)$.

`include.outcome` If available: should the outcome component of a sample selection model be reported?

`include.overidentification` If available: should the J-test for overidentification be reported (for `gel` and `gmm` objects)?

`include.percentile` If available: should the percentile (`tau`) be reported?

`include.precision` If available: should the precision estimates of a `betareg` fit (the `phi` coefficients) be reported as part of the coefficients block?

`include.pseudors` If available: should the pseudo R-squared be reported?

`include.pvalues` If available: should the p values be reported (naive p values are not recommended for `lme4` models, but see also the `mcmc.pvalues` argument)?

`include.rmse` If available: should the root-mean-square error (= residual standard deviation) be reported?

`include.rsquared` If available: should R-squared be reported?

`include.sargan` If available: should the Sargan test be reported?

`include.scale` If available: should the scale parameter `gamma` and its standard error be reported (for `geeglm` models)?

`include.selection` If available: should the selection component of a sample selection model be reported?

`include.smooth` If available: should the smooth terms of a GAM be reported? If they are reported, the EDF value is reported as the coefficient, and DF is included in parentheses (not standard errors because a chi-square test is used for the smooth terms).

`include.suffix` If available: include the name of the current model in parentheses after each model term (instead of before the model term).

`include.tau` If available: include `tau` in linear quantile mixed models?

`include.thresholds` If available: should the threshold parameters (that is, the intercepts for the class boundaries) be reported in ordinal models?

`include.trisk` If available: should the total time at risk be reported (in event-history models)?

`include.variance` If available: should group variances be reported?

`include.wald` If available: should the Wald statistic be included?

`include.weight` If available: should the weight be included?

`include.zero` If available: should the zero-inflation model of a zero-inflated regression or the zero hurdle model of a hurdle regression be included in the coefficients block (after the count model)?

include.zph	If available: should the Cox proportional hazards assumption be tested (resulting in a p value indicating whether the proportional hazards assumption of the model is violated)?
jackknife	If available: use Jackknife variance instead of Asymptotic variance.
level	Confidence level (1 - alpha) for computing confidence intervals.
levels	The names of the levels of a multinomial model that should be included in the table. Should be provided as a vector of character strings.
method	The method used to compute confidence intervals or p values. In lme4 models, the default value "naive" computes naive p values while the other methods compute confidence intervals using the confint function.
nsim	In linear mixed effects models: the MCMC sample size or number of bootstrapping replications on the basis of which confidence intervals are computed (only if the method argument does not specify "naive", which is the default behavior). Note: large values may take considerable computing time.
overIdentTest	If available: which test statistics should be included in an overidentification test (for gels and gmm objects)?
prefix	Include prefix before the label of the coefficient in order to identify the current model component.
robust	If available: report robust instead of naive standard errors.
standardized	If available: report standardized coefficients instead of raw coefficients?
use.ci	Use confidence intervals rather than standard errors.
...	Custom parameters which are handed over to subroutines.

Details

extract is a generic function which extracts coefficients and GOF measures from statistical model objects. There are several extract methods for the specific model types, which are called by the generic extract function if it encounters a model known to be handled by the specific method. The output is a texreg object, which is subsequently used by the [texreg](#) function.

Methods

aftreg An extract method for aftreg objects from the **eha** package.

Arima An extract method for Arima objects from the **stats** package.

ARIMA An extract method for ARIMA objects from the **forecast** package.

averaging An extract method for averaging objects from the **MuMIn** package.

bam An extract method for bam objects from the **mgcv** package.

betamfx An extract method for betamfx objects from the **mfx** package.

betaor An extract method for betaor objects from the **mfx** package.

betareg An extract method for betareg objects from the **betareg** package.

brglm An extract method for brglm objects from the **brglm** package.

btergm An extract method for btergm objects from the **xergm** package.

censReg An extract method for censReg objects from the **censReg** package.

clm An extract method for clm objects from the **ordinal** package.

clmm An extract method for clmm objects from the **ordinal** package.

clogit An extract method for clogit objects from the **survival** package.

coefstest An extract method for coefstest objects from the **lmtest** package.

coxph An extract method for coxph objects from the **survival** package.

coxph.penal An extract method for coxph.penal objects from the **survival** package.

dynlm An extract method for dynlm objects from the **dynlm** package.

ergm An extract method for ergm objects from the **ergm** package.

ergmm An extract method for ergmm objects from the **latentnet** package.

ets An extract method for ets objects from the **forecast** package.

felm An extract method for felm objects from the **lfe** package.

fGARCH An extract method for fGARCH objects from the **fGarch** package.

forecast An extract method for forecast objects from the **forecast** package.

gam An extract method for gam objects from the **mgcv** package.

gamlss An extract method for gamlss objects from the **gamlss** package.

gee An extract method for gee objects from the **gee** package.

geeglm An extract method for geeglm objects from the **geepack** package.

gel An extract method for gel objects from the **gmm** package.

glm An extract method for glm objects from the **stats** package.

glmerMod An extract method for glmerMod objects from the (old) **lme4** package.

glmmadmb An extract method for glmmadmb objects from the **glmmADMB** package.

glmmPQL An extract method for glmmPQL objects from the **MASS** package.

glmrob An extract method for glmrob objects from the **robustbase** package.

gls An extract method for gls objects from the **nlme** package.

gmm An extract method for gmm objects from the **gmm** package.

H2OBinomialModel An extract method for H2OBinomialModel objects from the **h2o** package.

hurdle An extract method for hurdle objects from the **pscl** package.

ivreg An extract method for ivreg objects from the **AER** package.

lm An extract method for lm objects from the **stats** package.

lme An extract method for lme objects from the **nlme** package.

lme4 An extract method for lme4 objects from the **lme4** package.

lmerMod An extract method for lmerMod objects from the (old) **lme4** package.

lmrob An extract method for lmrob objects from the **robustbase** package.

lnam An extract method for lnam objects from the **sna** package.

logitmfx An extract method for logitmfx objects from the **mfx** package.

logitor An extract method for logitor objects from the **mfx** package.

`lqmm` An extract method for `lqmm` objects from the **lqmm** package.

`lrm` An extract method for `lrm` objects from the **Design** or **rms** package.

`maBina` An extract method for `maBina` objects from the **erer** package.

`mer` An extract method for `mer` objects from the (old) **lme4** package.

`mlogit` An extract method for `mlogit` objects from the **mlogit** package.

`mnlogit` An extract method for `mnlogit` objects from the **mnlogit** package.

`model.selection` An extract method for `model.selection` objects from the **MuMIn** package.

`mtergm` An extract method for `mtergm` objects from the **btergm** package.

`multinom` An extract method for `multinom` objects from the **nnet** package.

`negbin` An extract method for `negbin` objects from the **MASS** package.

`negbinirr` An extract method for `negbinirr` objects from the **mfX** package.

`negbinmfx` An extract method for `negbinmfx` objects from the **mfX** package.

`netlogit` An extract method for `netlogit` objects from the **sna** package.

`nlme` An extract method for `nlme` objects from the **nlme** package.

`nlmerMod` An extract method for `nlmerMod` objects from the (old) **lme4** package.

`ols` An extract method for `ols` objects from the **rms** package.

`pgmm` An extract method for `pgmm` objects from the **plm** package.

`phreg` An extract method for `phreg` objects from the **eha** package.

`plm` An extract method for `plm` objects from the **plm** package.

`pmg` An extract method for `pmg` objects from the **plm** package.

`poissonirr` An extract method for `poissonirr` objects from the **mfX** package.

`poissonmfx` An extract method for `poissonmfx` objects from the **mfX** package.

`polr` An extract method for `polr` objects from the **MASS** package.

`probitmfx` An extract method for `probitmfx` objects from the **mfX** package.

`rem.dyad` An extract method for `rem.dyad` objects from the **relevent** package.

`rlm` An extract method for `rlm` objects from the **MASS** package.

`rq` An extract method for `rq` objects from the **quantreg** package.

`sarlm` An extract method for `sarlm` objects from the **spdep** package.

`sclm` An extract method for `sclm` objects from the **ordinal** package.

`selection` An extract method for `selection` objects from the **sampleSelection** package.

`sienaFit` An extract method for `sienaFit` objects from the **RSiena** package.

`simex` An extract method for `simex` objects from the **simex** package.

`stergm` An extract method for `stergm` objects from the **tergm** package.

`survreg` An extract method for `survreg` objects from the **survival** package.

`survreg.penal` An extract method for `survreg.penal` objects from the **survival** package.

`svyglm` An extract method for `svyglm` objects from the **survey** package.

`systemfit` An extract method for `systemfit` objects from the **systemfit** package.

texreg An extract method for texreg objects from the **texreg** package. The purpose is to allow for easy manipulation of the output. texreg objects can be created using the [createTexreg](#) function or using the [extract](#) function. After manipulating the object, it can be handed back to the [screenreg](#), [texreg](#), or [htmlreg](#) functions for creating a table.

tobit An extract method for tobit objects from the **AER** package.

vglm An extract method for vglm objects from the **VGAM** package.

weibreg An extract method for weibreg objects from the **eha** package.

wls An extract method for wls objects from the **metaSEM** package.

zelig An extract method for zelig objects from the **Zelig** package.

Zelig An extract method for Zelig objects from the **Zelig** package.

When fitting models, **Zelig** often wraps additional information around a model object produced by a different R library. It is often possible to recover that model object using the `from_zelig_model` function from **Zelig** ($\geq 5.0-16$). If that underlying model is supported by texreg, tables will be produced as usual, automatically. To identify the relevant model-specific arguments (e.g., `include.adjrs = TRUE`), identify the class of the underlying model (`class(Zelig::from_zelig_model(model))`), and check the appropriate `extract.*` function in texreg.

zeroinfl An extract method for zeroinfl objects from the **pscl** package.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

References

Leifeld, Philip (2013). texreg: Conversion of Statistical Model Output in R to LaTeX and HTML Tables. Journal of Statistical Software, 55(8), 1-24. <http://www.jstatsoft.org/v55/i08/>.

See Also

[texreg-package](#) [texreg](#)

plotreg

Create coefficient plots from statistical model output

Description

Create coefficient plots from statistical model output.

Usage

```
plotreg(l, file = NULL, custom.model.names = NULL,
        custom.coef.names = NULL, custom.note = NULL,
        override.coef = 0, override.se = 0, override.pval = 0,
        override.ci.low = 0, override.ci.up = 0,
        omit.coef = NULL, reorder.coef = NULL, ci.level = 0.95,
        use.se = FALSE, mfrow = TRUE, xlim = NULL, cex = 2.5,
        lwd.zerobar = 4, lwd.vbars = 1, lwd.inner = 7,
        lwd.outer = 5, ylab.cex = 1.0, signif.light = "#fbc9b9",
        signif.medium = "#f7523a", signif.dark = "#bd0017",
        insignif.light = "#c5dbe9", insignif.medium = "#5a9ecc",
        insignif.dark = "#1c5ba6", ...)

coefplot(labels, estimates, lower.inner = NULL,
         upper.inner = NULL, lower.outer = NULL,
         upper.outer = NULL, signif.outer = TRUE,
         xlab = "Coefficients and confidence intervals",
         main = "Coefficient plot", xlim = NULL,
         cex = 2.5, lwd.zerobar = 4, lwd.vbars = 1,
         lwd.inner = 7, lwd.outer = 5, ylab.cex = 1.0,
         signif.light = "#fbc9b9", signif.medium = "#f7523a",
         signif.dark = "#bd0017", insignif.light = "#c5dbe9",
         insignif.medium = "#5a9ecc", insignif.dark = "#1c5ba6",
         ...)
```

Arguments

- | | |
|--------------------|---|
| l | A statistical model or a list of statistical models. Lists of models can be specified as <code>l = list(model.1, model.2, ...)</code> . Different object types can also be mixed. |
| file | Using this argument, the resulting table is written to a file rather than to the R prompt. The file name can be specified as a character string. The file extension is automatically recognized. pdf, ps, png, bmp, jpg, and tiff are supported. |
| custom.model.names | A character vector of labels for the models. By default, the models are named Model 1, Model 2, etc. Specifying <code>custom.model.names = c("My name 1", "My name 2")</code> etc. overrides the default behavior. |
| custom.coef.names | By default, plotreg uses the coefficient names which are stored in the models. The <code>custom.coef.names</code> argument can be used to replace them by other character strings in the order of appearance. For example, if a model shows a total of three coefficients (including the intercept), the argument <code>custom.coef.names = c("Intercept", "vari</code> will replace their names in this order. |
| custom.note | With this argument, a replacement text for the xlab note below the diagram can be provided. If an empty character object is provided (<code>custom.note = ""</code>), the note will be omitted completely. |

- `override.coef` Set custom values for the coefficients. New coefficients are provided as a list of numeric vectors. The list contains vectors of coefficients for each model. There must be as many vectors of coefficients as there are models. For example, if there are two models with three model terms each, the argument could be specified as `override.coef = list(c(0.1, 0.2, 0.3), c(0.05, 0.06, 0.07))`. If there is only one model, custom values can be provided as a plain vector (not embedded in a list). For example: `override.coef = c(0.05, 0.06, 0.07)`.
- `override.se` Set custom values for the standard errors. This only has an effect where standard errors are converted into confidence intervals because no other CIs are present. New standard errors are provided as a list of numeric vectors. The list contains vectors of standard errors for each model. There must be as many vectors of standard errors as there are models. For example, if there are two models with three coefficients each, the argument could be specified as `override.se = list(c(0.1, 0.2, 0.3), c(0.05, 0.06, 0.07))`. If there is only one model, custom values can be provided as a plain vector (not embedded in a list). For example: `override.se = c(0.05, 0.06, 0.07)`. Overriding standard errors can be useful for the implementation of robust SEs, for example.
- `override.pval` Set custom values for the p values. This only has an effect where standard errors are converted into confidence intervals because no other CIs are present. In this case, significance is derived from the p values rather than the confidence intervals. New p values are provided as a list of numeric vectors. The list contains vectors of p values for each model. There must be as many vectors of p values as there are models. For example, if there are two models with three coefficients each, the argument could be specified as `override.pval = list(c(0.1, 0.2, 0.3), c(0.05, 0.06, 0.07))`. If there is only one model, custom values can be provided as a plain vector (not embedded in a list). For example: `override.pval = c(0.05, 0.06, 0.07)`. Overriding p values can be useful for the implementation of robust SEs and p values, for example.
- `override.ci.low` Set custom lower confidence interval bounds. This works like the other override arguments, with one exception: if confidence intervals are provided here and in the `override.ci.up` argument, the standard errors and p values as well as the `ci.force` argument are ignored.
- `override.ci.up` Set custom upper confidence interval bounds. This works like the other override arguments, with one exception: if confidence intervals are provided here and in the `override.ci.low` argument, the standard errors and p values as well as the `ci.force` argument are ignored.
- `omit.coef` A character string which is used as a regular expression to remove coefficient rows from the table. For example, `omit.coef = "group"` deletes all coefficient rows from the diagram where the name of the coefficient contains the character sequence "group". More complex regular expressions can be used to filter out several kinds of model terms, for example `omit.coef = "(thresh)|(ranef)"` to remove all model terms matching either "thresh" or "ranef". The `omit.coef` argument is processed after the `custom.coef.names` argument, so the regular expression should refer to the custom coefficient names.
- `reorder.coef` Reorder the rows of the coefficient block of the resulting table in a custom way. The argument takes a vector of the same length as the number of coefficients.

For example, if there are three coefficients, `reorder.coef = c(3, 2, 1)` will put the third coefficient in the first row and the first coefficient in the third row. Reordering can be sensible because interaction effects are often added to the end of the model output although they were specified earlier in the model formula. Note: Reordering takes place after processing custom coefficient names and after omitting coefficients, so the `custom.coef.names` and `omit.coef` arguments should follow the original order.

<code>ci.level</code>	If standard errors are converted to confidence intervals (because a model does not natively support CIs), what confidence level should be used for the outer confidence interval? By default, 0.95 is used (i.e., an alpha value of 0.05).
<code>use.se</code>	Use one standard error for the inner horizontal bar and two standard errors from the estimate for the outer horizontal bar (instead of confidence intervals). Only available if standard errors can be extracted from the model using the respective extract function.
<code>mfrow</code>	If multiple models are handed over as the <code>l</code> argument, several plots are produced. If <code>mfrow = TRUE</code> is set, multiple diagrams are aligned on the same page. If <code>mfrow = FALSE</code> is set, each diagram per model comes out as a separate plot.
<code>xlim</code>	Horizontal limits. In the <code>coefplot</code> function, they must be provided as a vector with two numeric, e.g., <code>xlim = c(-5, 5)</code> for displaying a range from -5 to +5. In the <code>plotreg</code> function, they can be provided either as such a vector with two values or as a list of vectors (with each entry corresponding to a model in <code>l</code>).
<code>lwd.zerobar</code>	Line width of the vertical gray bar at the <code>x</code> value of 0. To remove the line, set <code>lwd.zerobar = 0</code> .
<code>lwd.vbars</code>	Line width of the thin vertical gray bars. To remove them completely, set <code>lwd.vbars = 0</code> .
<code>labels</code>	The names of the model terms. They are arranged on the left axis.
<code>estimates</code>	The coefficients (point estimates) of the model terms. They are depicted as bold dots in the coefficient plot.
<code>lower.inner</code>	The lower bounds of the inner confidence intervals, provided as a vector. Inner CI means more relaxed (lower confidence level, higher alpha) because fewer observations have to fall into the CI, therefore the CI gets smaller.
<code>upper.inner</code>	The upper bounds of the inner confidence intervals, provided as a vector. Inner CI means more relaxed (lower confidence level, higher alpha) because fewer observations have to fall into the CI, therefore the CI gets smaller.
<code>lower.outer</code>	The lower bounds of the outer confidence intervals, provided as a vector. Outer CI means stricter or narrower (higher confidence level, lower alpha) because more observations have to fall into the CI, therefore the CI gets larger.
<code>upper.outer</code>	The upper bounds of the outer confidence intervals, provided as a vector. Outer CI means stricter or narrower (higher confidence level, lower alpha) because more observations have to fall into the CI, therefore the CI gets larger.
<code>signif.outer</code>	Different colors are used for significant estimates and confidence intervals. If <code>signif.outer = TRUE</code> , the outer CIs are used to evaluate significance, otherwise the inner CIs are used.
<code>xlab</code>	The label of the <code>x</code> axis.

<code>main</code>	The main title or heading of the plot.
<code>cex</code>	Size of the point representing the estimate.
<code>lwd.inner</code>	Line width of the inner confidence interval.
<code>lwd.outer</code>	Line width of the outer confidence interval.
<code>ylab.cex</code>	Size of the coefficient labels. The size of the x axis labels can be adjusted by using argument <code>cex.axis = 1.0</code> .
<code>signif.light</code>	Color of outer confidence intervals for significant model terms.
<code>signif.medium</code>	Color of inner confidence intervals for significant model terms.
<code>signif.dark</code>	Color of point estimates and labels for significant model terms.
<code>insignif.light</code>	Color of outer confidence intervals for insignificant model terms.
<code>insignif.medium</code>	Color of inner confidence intervals for insignificant model terms.
<code>insignif.dark</code>	Color of point estimates and labels for insignificant model terms.
<code>...</code>	Custom options to be passed on to the <code>extract</code> function or the graphics device. See the help entries of <code>extract</code> and <code>extract-methods</code> for more information.

Details

The `coefplot` function produces coefficient plots (i.e., forest plots applied to point estimates and confidence intervals). It accepts raw data (the lower and upper bounds of inner and outer confidence intervals as well as the point estimates and their names) as input data. Significant coefficients and intervals can be plotted in a different color.

The `plotreg` function is a wrapper for the `coefplot` function and works much like the `screenreg`, `texreg`, and `htmlreg` functions. It accepts a single or multiple statistical models as input and internally extracts the relevant data from the models. If confidence intervals are not defined in the `extract` method of a statistical model (see `extract` and `extract-methods`), the default standard errors are converted to confidence intervals. Most of the arguments work either like in the `screenreg`, `texreg`, and `htmlreg` functions, or they work like in the `coefplot` function.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[texreg-package](#) [extract](#) [extract-methods](#) [texreg](#)

Examples

```
#example from the 'lm' help file:
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2,10,20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
screenreg(lm.D9) # print model output to the R console
plotreg(lm.D9) # plot model output as a diagram
```

`print.texregTable` *Print the output of a screenreg, texreg, or htmlreg call*

Description

Print the output of a screenreg, texreg, or htmlreg call.

Usage

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

Arguments

`x` A texregTable object. This is basically a simple character object with an additional class name called texregTable.

`...` Additional arguments to be handed over to the cat function.

Details

This function prints a texregTable object, which results from a screenreg, texreg, or htmlreg call. Most of the time, this function is called implicitly by just entering the name of the object.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[texreg-package texreg](#)

`texreg` *Convert regression output to LaTeX or HTML tables*

Description

Conversion of R regression output to LaTeX or HTML tables.

Usage

```
texreg(l, file = NULL, single.row = FALSE, stars = c(0.001,
  0.01, 0.05), custom.model.names = NULL,
  custom.coef.names = NULL, custom.coef.map = NULL,
  custom.gof.names = NULL, custom.note = NULL, digits = 2,
  leading.zero = TRUE, symbol = "\\cdot", override.coef = 0,
  override.se = 0, override.pvalues = 0, override.ci.low = 0,
  override.ci.up = 0, omit.coef = NULL, reorder.coef = NULL,
  reorder.gof = NULL, ci.force = FALSE, ci.force.level = 0.95,
  ci.test = 0, groups = NULL, custom.columns = NULL,
  custom.col.pos = NULL, bold = 0.00, center = TRUE,
  caption = "Statistical models", caption.above = FALSE,
  label = "table:coefficients", booktabs = FALSE,
  dcolumn = FALSE, lyx = FALSE, sideways = FALSE,
  longtable = FALSE, use.packages = TRUE, table = TRUE,
  no.margin = FALSE, fontsize = NULL, scalebox = NULL,
  float.pos = "", ...)
```

```
htmlreg(l, file = NULL, single.row = FALSE, stars = c(0.001,
  0.01, 0.05), custom.model.names = NULL,
  custom.coef.names = NULL, custom.coef.map = NULL,
  custom.gof.names = NULL, custom.note = NULL, digits = 2,
  leading.zero = TRUE, symbol = "&middot;", override.coef = 0,
  override.se = 0, override.pvalues = 0, override.ci.low = 0,
  override.ci.up = 0, omit.coef = NULL, reorder.coef = NULL,
  reorder.gof = NULL, ci.force = FALSE, ci.force.level = 0.95,
  ci.test = 0, groups = NULL, custom.columns = NULL,
  custom.col.pos = NULL, bold = 0.00, center = TRUE,
  caption = "Statistical models", caption.above = FALSE,
  star.symbol = "*", inline.css = TRUE, doctype = TRUE,
  html.tag = FALSE, head.tag = FALSE, body.tag = FALSE,
  indentation = "", vertical.align.px = 0, ...)
```

```
screenreg(l, file = NULL, single.row = FALSE, stars = c(0.001,
  0.01, 0.05), custom.model.names = NULL,
  custom.coef.names = NULL, custom.coef.map = NULL,
  custom.gof.names = NULL, custom.note = NULL, digits = 2,
  leading.zero = TRUE, symbol = ".", override.coef = 0,
  override.se = 0, override.pvalues = 0, override.ci.low = 0,
  override.ci.up = 0, omit.coef = NULL, reorder.coef = NULL,
  reorder.gof = NULL, ci.force = FALSE, ci.force.level = 0.95,
  ci.test = 0, groups = NULL, custom.columns = NULL,
  custom.col.pos = NULL, column.spacing = 2, outer.rule = "=",
  inner.rule = "-", ...)
```

Arguments

- l** A statistical model or a list of statistical models. Lists of models can be specified as `l = list(model.1, model.2, ...)`. Different object types can also be mixed.
- file** Using this argument, the resulting table is written to a file rather than to the R prompt. The file name can be specified as a character string. Writing a table to a file can be useful for working with MS Office or LibreOffice. For example, using the `htmlreg` function, an HTML table can be written to a file with the extension `.doc` and opened with MS Word. The table can then be simply copied into any Word document, retaining the formatting of the table. Note that LibreOffice can import only plain HTML; CSS decorations are not supported; the resulting tables do not retain the full formatting in LibreOffice.
- single.row** By default, a model parameter takes up two lines of the table: the standard error is listed in parentheses under the coefficient. This saves a lot of horizontal space on the page and is the default table format in most academic journals. If `single.row = TRUE` is activated, however, both coefficient and standard error are placed in a single table cell in the same line.
- stars** The significance levels to be used to draw stars. Between 0 and 4 threshold values can be provided as a numeric vector. For example, `stars = numeric(0)` will not print any stars and will not print any note about significance levels below the table. `stars = 0.05` will attach one single star to all coefficients where the p value is below 0.05. `stars = c(0.001, 0.01, 0.05, 0.1)` will print one, two, or three stars, or a symbol as specified by the `symbol` argument depending on the p values.
- custom.model.names**
A character vector of labels for the models. By default, the models are named Model 1, Model 2, etc. Specifying `model.names = c("My name 1", "My name 2")` etc. overrides the default behavior.
- custom.coef.names**
By default, **texreg** uses the coefficient names which are stored in the models. The `custom.coef.names` argument can be used to replace them by other character strings in the order of appearance. For example, if a table shows a total of three different coefficients (including the intercept), the argument `custom.coef.names = c("Intercept", "variable 1", "variable 2")` will replace their names in this order.
Sometimes it happens that the same variable has a different name in different models. In this case, the user can use this function to assign identical names. If possible, the rows will then be merged into a single row unless both rows contain values in the same column.
Where the argument contains an NA value, the original name of the coefficient is kept. For example, `custom.coef.names = c(NA, "age", NA)` will only replace the second coef name and leave the first and third name as they are in the original model.
- custom.coef.map**
The `custom.coef.map` argument can be used to select, omit, rename, and re-order coefficients.

Users must supply a named list of this form: `list('x' = 'First variable', 'y' = NA, 'z' = 'Third variable')`.
 With that particular example of `custom.coef.map`,

1. coefficients will be presented in order: x, y, z.
2. variable x will appear as "First variable", variable y will appear as "y", and variable "z" will appear as "Third variable".
3. all variables not named "x", "y", or "z" will be omitted from the table.

<code>custom.gof.names</code>	A character vector which is used to replace the names of the goodness-of-fit statistics at the bottom of the table. The vector must have the same length as the number of GOF statistics in the final table. The argument works like the <code>custom.coef.names</code> argument, but for the GOF values. NA values can be included where the original GOF name should be kept.
<code>custom.note</code>	With this argument, a replacement text for the significance note below the table can be provided. If an empty character object is provided (<code>custom.note = ""</code>), the note will be omitted completely. If some character string is provided (e.g., <code>custom.note = "My note"</code>), the significance legend is replaced by My note. The original significance legend can be included by inserting the <code>%stars</code> wildcard. For example, a custom note can be added right after the significance legend by providing <code>custom.note = "%stars. My note"</code> .
<code>digits</code>	Set the number of decimal places for coefficients, standard errors and goodness-of-fit statistics. Do not use negative values! The argument works like the <code>digits</code> argument in the <code>round</code> function of the base package.
<code>leading.zero</code>	Most journals require leading zeros of coefficients and standard errors (for example, <code>0.35</code>). This is also the default <code>texreg</code> behavior. Some journals, however, require omission of leading zeros (for example, <code>.35</code>). This can be achieved by setting <code>leading.zero = FALSE</code> .
<code>symbol</code>	If four threshold values are handed over to the <code>stars</code> argument, p values smaller than the largest threshold value but larger than the second-largest threshold value are denoted by this symbol. The default symbol is <code>"\\cdot"</code> for the LaTeX dot, <code>"&middot;"</code> for the HTML dot, or simply <code> "."</code> for the ASCII dot. If the <code>texreg</code> function is used, any other mathematical LaTeX symbol or plain text symbol can be used, for example <code>symbol = "\\circ"</code> for a small circle (note that backslashes must be escaped). If the <code>htmlreg</code> function is used, any other HTML character or symbol can be used. For the <code>screenreg</code> function, only plain text characters can be used.
<code>override.coef</code>	Set custom values for the coefficients. New coefficients are provided as a list of numeric vectors. The list contains vectors of coefficients for each model. There must be as many vectors of coefficients as there are models. For example, if there are two models with three model terms each, the argument could be specified as <code>override.coef = list(c(0.1, 0.2, 0.3), c(0.05, 0.06, 0.07))</code> . If there is only one model, custom values can be provided as a plain vector (not embedded in a list). For example: <code>override.coef = c(0.05, 0.06, 0.07)</code> .
<code>override.se</code>	Set custom values for the standard errors. New standard errors are provided as a list of numeric vectors. The list contains vectors of standard errors for each model. There must be as many vectors of standard errors as there are models. For example, if there are two models with three coefficients each, the argument

could be specified as `override.se = list(c(0.1, 0.2, 0.3), c(0.05, 0.06, 0.07))`. If there is only one model, custom values can be provided as a plain vector (not embedded in a list). For example: `override.se = c(0.05, 0.06, 0.07)`. Overriding standard errors can be useful for the implementation of robust SEs, for example.

`override.pvalues`

Set custom values for the p values. New p values are provided as a list of numeric vectors. The list contains vectors of p values for each model. There must be as many vectors of p values as there are models. For example, if there are two models with three coefficients each, the argument could be specified as `override.pvalues = list(c(0.1, 0.2, 0.3), c(0.05, 0.06, 0.07))`. If there is only one model, custom values can be provided as a plain vector (not embedded in a list). For example: `override.pvalues = c(0.05, 0.06, 0.07)`. Overriding p values can be useful for the implementation of robust SEs and p values, for example.

`override.ci.low`

Set custom lower confidence interval bounds. This works like the other override arguments, with one exception: if confidence intervals are provided here and in the `override.ci.up` argument, the standard errors and p values as well as the `ci.force` argument are ignored.

`override.ci.up`

Set custom upper confidence interval bounds. This works like the other override arguments, with one exception: if confidence intervals are provided here and in the `override.ci.low` argument, the standard errors and p values as well as the `ci.force` argument are ignored.

`omit.coef`

A character string which is used as a regular expression to remove coefficient rows from the table. For example, `omit.coef = "group"` deletes all coefficient rows from the table where the name of the coefficient contains the character sequence "group". More complex regular expressions can be used to filter out several kinds of model terms, for example `omit.coef = "(thresh)|(ranef)"` to remove all model terms matching either "thresh" or "ranef". The `omit.coef` argument is processed after the `custom.coef.names` argument, so the regular expression should refer to the custom coefficient names. To omit GOF entries instead of coefficient entries, use the custom arguments of the extract functions instead (see the help entry of the [extract](#) function or [extract-methods](#)).

`reorder.coef`

Reorder the rows of the coefficient block of the resulting table in a custom way. The argument takes a vector of the same length as the number of coefficients. For example, if there are three coefficients, `reorder.coef = c(3, 2, 1)` will put the third coefficient in the first row and the first coefficient in the third row. Reordering can be sensible because interaction effects are often added to the end of the model output although they were specified earlier in the model formula. Note: Reordering takes place after processing custom coefficient names and after omitting coefficients, so the `custom.coef.names` and `omit.coef` arguments should follow the original order.

`reorder.gof`

Reorder the rows of the goodness-of-fit block of the resulting table in a custom way. The argument takes a vector of the same length as the number of GOF statistics. For example, if there are three goodness-of-fit rows, `reorder.gof = c(3, 2, 1)` will exchange the first and the third row. Note: Reordering takes place after pro-

- cessing custom GOF names, so the `custom.gof.names` argument should follow the original order.
- `ci.force` Should confidence intervals be used instead of the default standard errors and p values? Most models implemented in the **texreg** package report standard errors and p values by default while few models report confidence intervals. However, the functions in the **texreg** package can convert standard errors and into confidence intervals if desired. To enforce confidence intervals instead of standard errors, the `ci.force` argument accepts either a logical value indicating whether all models or none of the models should be forced to report confidence intervals (`ci.force = TRUE` for all and `ci.force = FALSE` for none) or a vector of logical values indicating for each model separately whether the model should be forced to report confidence intervals (e.g., `ci.force = c(FALSE, TRUE, FALSE)`). Confidence intervals are computed using the standard normal distribution (z values based on the `qnorm` function).
- `ci.force.level` If the `ci.force` argument is used to convert standard errors to confidence intervals, what confidence level should be used? By default, 0.95 is used (i.e., an alpha value of 0.05).
- `ci.test` If confidence intervals are reported, the `ci.test` argument specifies the reference value to establish whether a coefficient/CI is significant. The default value `ci.test = 0`, for example, will attach a significance star to coefficients if the confidence interval does not contain 0. If no star should be printed at all, `ci.test = NULL` can be used. The `ci.test` argument works both for models with native support for confidence intervals and in cases where the `ci.force` argument is used.
- `groups` This argument can be used to group the rows of the table into blocks. For example, there could be one block for hypotheses and another block for control variables. Each group has a heading, and the row labels within a group are indented. The partitions must be handed over as a list of named numeric vectors, where each number is a row index and each name is the heading of the group. Example: `groups = list("first group" = 1:4, "second group" = 7:8)`.
- `custom.columns` An optional list of additional text columns to be inserted into the table, for example coefficient types. The list should contain one or more character vectors with as many character or numeric elements as there are rows. If the vectors in the list are named, the names are used as labels in the table header. For example, `custom.columns = list(type = c("a", "b", "c"), 1:3)` will add two columns; the first one is labeled while the second one is not. Note that the numeric elements of the second column will be converted to character objects in this example. The consequence is that decimal alignment with the **dcolumn** package is switched off in these columns. Note that this argument is processed after any arguments that affect the number of rows.
- `custom.col.pos` An optional integer vector of positions for the columns given in the `custom.columns` argument. For example, if there are three custom columns, `custom.col.pos = c(1, 3, 3)` will insert the first custom column before the first column of the original table and the remaining two custom columns after the second column of the original table. By default, all custom columns are placed after the first column, which usually contains the coefficient names.

<code>bold</code>	[only in the <code>texreg</code> and <code>htmlreg</code> functions] The <code>p</code> value threshold below which the coefficient shall be formatted in a bold font. For example, <code>bold = 0.05</code> will cause all coefficients which are significant at the 95% level to be formatted in bold. Note that this is not compatible with the <code>dcolumn</code> argument in the <code>texreg</code> function. If both are TRUE, <code>dcolumn</code> is switched off and a warning message appears. Note also that it is advisable to use <code>stars = FALSE</code> together with the <code>bold</code> argument because having both bolded coefficients and significance stars usually does not make any sense.
<code>center</code>	[only in the <code>texreg</code> and <code>htmlreg</code> functions] Should the table be horizontally aligned at the center of the page?
<code>caption</code>	[only in the <code>texreg</code> and <code>htmlreg</code> functions] Set the caption of the table.
<code>caption.above</code>	[only in the <code>texreg</code> and <code>htmlreg</code> functions] Should the caption of the table be placed above the table? By default, it is placed below the table.
<code>label</code>	[only in the <code>texreg</code> function] Set the label of the table environment.
<code>booktabs</code>	[only in the <code>texreg</code> function] Use the <code>booktabs</code> LaTeX package to get thick horizontal rules in the output table (recommended).
<code>dcolumn</code>	[only in the <code>texreg</code> function] Use the <code>dcolumn</code> LaTeX package to get a nice alignment of the coefficients (recommended).
<code>lyx</code>	[only in the <code>texreg</code> function] logical; if TRUE, each newline in the output is doubled, which facilitates transferring the output into the LyX document processor.
<code>sideways</code>	[only in the <code>texreg</code> function] If <code>sideways = TRUE</code> is set, the table floating environment is replaced by a <code>sidewaystable</code> float, and the <code>rotating</code> package is loaded in the preamble. The argument only has an effect if <code>table = TRUE</code> is also set.
<code>longtable</code>	[only in the <code>texreg</code> function] If <code>longtable = TRUE</code> is set, the <code>longtable</code> environment from the <code>longtable</code> LaTeX package is used to set tables across multiple pages. Note that this argument is not compatible with the <code>sideways</code> and <code>scalebox</code> arguments. These arguments will be automatically switched off when <code>longtable = TRUE</code> is set.
<code>use.packages</code>	[only in the <code>texreg</code> function] If this argument is set to TRUE (= the default behavior), the required LaTeX packages are loaded in the beginning. If set to FALSE, the use package statements are omitted from the output.
<code>table</code>	[only in the <code>texreg</code> function] By default, <code>texreg</code> puts the actual tabular object in a table floating environment. To get only the tabular object without the whole table header, set <code>table = FALSE</code> .
<code>no.margin</code>	[only in the <code>texreg</code> function] In order to save space, inner margins of tables can be switched off.
<code>fontsize</code>	[only in the <code>texreg</code> function] The <code>fontsize</code> argument serves to change the font size used in the table. Valid values are "tiny", "scriptsize", "footnotesize", "small", "normalsize", "large", "Large", "LARGE", "huge", and "Huge". Note that the <code>scalebox</code> argument often achieves better results when the goal is to change the size of the table.

scalebox	[only in the texreg function] The scalebox argument serves to resize the table. For example, scalebox = 1.0 is equivalent to the normal size, scalebox = 0.5 decreases the size of the table by one half, and scalebox = 2.0 doubles the space occupied by the table. Note that the scalebox argument does not work when the longtable argument is used.
float.pos	[only in the texreg function] This argument specifies where the table should be located on the page or in the document. By default, no floating position is specified, and LaTeX takes care of the position automatically. Possible values include h (here), p (page), t (top), b (bottom), any combination thereof, e.g. tb, or any of these values followed by an exclamation mark, e.g. t!, in order to enforce this position. The square brackets do not have to be specified.
star.symbol	[only in the htmlreg function] Alternative characters for the significance stars can be specified. This is useful if knitr and Markdown are used for HTML report generation. In Markdown, asterisks or stars are interpreted as special characters, so they have to be escaped. To make htmlreg compatible with Markdown, specify star.symbol = "*". Note that some other modifications are recommended for usage with knitr in combination with Markdown or HTML (see the inline.css, doctype, html.tag, head.tag, and body.tag arguments).
inline.css	[only in the htmlreg function] Should the CSS stylesheets be embedded directly in the code of the table (inline.css = TRUE), or should the CSS stylesheets be enclosed in the <head> tag, that is, separated from the table code (inline.css = FALSE)? Having inline CSS code makes the code of the table more complex, but sometimes it may be helpful when only the table shall be printed, without the head of the HTML file (for example when the table is embedded in a knitr report). As a rule of thumb: use inline CSS if the table is not saved to a file.
doctype	[only in the htmlreg function] Should the first line of the HTML code contain the DOCTYPE definition? If TRUE, the HTML 4 TRANSITIONAL version is used. If FALSE, no DOCTYPE will be included. Omitting the DOCTYPE can be helpful when the knitr package is used to generate HTML code because knitr requires only the plain table, not the whole HTML document including the document type declaration. Including the DOCTYPE can be helpful when the code is saved to a file, for example as an MS Word document.
html.tag	[only in the htmlreg function] Should the table code (and possibly the <body> and <head> tags) be enclosed in an <html> tag? Suppressing this tag is recommended when knitr is used for dynamic HTML or Markdown report generation. Including this tag is recommended when the code is saved to a file, for example as an MS Word document.
head.tag	[only in the htmlreg function] Should the <head> tag (including CSS definitions and title/caption) be included in the HTML code? Suppressing this tag is recommended when knitr is used for dynamic HTML or Markdown report generation. Including this tag is recommended when the code is saved to a file, for example as an MS Word document.
body.tag	[only in the htmlreg function] Should the table code be enclosed in a <body> HTML tag? Suppressing this tag is recommended when knitr is used for dynamic HTML or Markdown report generation. Including this tag is recommended when the code is saved to a file, for example as an MS Word document.

<code>indentation</code>	[only in the <code>htmlreg</code> function] Characters used for indentation of the HTML code. By default, <code>indentation = ""</code> uses no indentation. Any number of spaces or characters can be used instead. For example, <code>indentation = " "</code> uses two spaces of (additional) indentation for each subelement.
<code>vertical.align.px</code>	[only in the <code>htmlreg</code> function] Vertical alignment of significance stars. Browsers differ in their ways of displaying superscripted significance stars; in some browsers the stars are elevated by default, and in other browsers the stars are aligned vertically with the text, without any actual superscripting. This argument controls by how many additional pixels the stars are elevated. The default setting of 0 uses the defaults of the browser. In RStudio's internal browser, this looks OK, but in Firefox, this looks too low. A value of 4 looks OK in Firefox, for example, but is above the line in RStudio's internal browser.
<code>column.spacing</code>	[only in the <code>screenreg</code> function] The amount of space between any two columns of a table. By default, two spaces are used. If the tables do not fit on a single page horizontally, the value can be set to 1 or 0.
<code>outer.rule</code>	[only in the <code>screenreg</code> function] The character which is used to draw the outer horizontal line above and below a table. If an empty character object is provided (i.e., <code>outer.rule = ""</code>), there will be no outer horizontal lines. Recommended values are <code>" "</code> , <code>"="</code> , <code>"-"</code> , <code>"_"</code> , or <code>"#"</code> .
<code>inner.rule</code>	[only in the <code>screenreg</code> function] The character which is used to draw the inner horizontal line above and below a table. If an empty character object is provided (i.e., <code>outer.rule = ""</code>), there will be no inner horizontal lines. Recommended values are <code>" "</code> , <code>"-"</code> , or <code>"_"</code> .
<code>...</code>	Custom options to be passed on to the <code>extract</code> function. For example, most <code>extract</code> methods provide custom options for the inclusion or exclusion of specific goodness-of-fit statistics. See the help entries of extract and extract-methods for more information.

Details

`texreg` converts coefficients, standard errors, significance stars, and goodness-of-fit statistics of statistical models into LaTeX tables or HTML tables or into nicely formatted screen output for the R console. A list of several models can be combined in a single table. The output is customizable. New model types can be easily implemented. Confidence intervals can be used instead of standard errors and p values.

The `texreg()` function creates LaTeX code for inclusion in a LaTeX document or for usage with **Sweave** or **knitr**.

The `htmlreg()` function creates HTML code. Tables in HTML format can be saved with a ".html" extension and displayed in a web browser. Alternatively, they can be saved with a ".doc" extension and opened in MS Word for inclusion in office documents. `htmlreg()` also works with **knitr** and HTML or Markdown. Note that the `inline.css`, `doctype`, `html.tag`, `head.tag`, and `body.tag` arguments must be adjusted for the different purposes (see the description of the arguments).

The `screenreg()` function creates text representations of tables and prints them to the R console. This is an alternative to the `summary` method and serves easy model comparison. Moreover, once a table has been prepared in the R console, it can be later exported to LaTeX or HTML with little extra effort because the majority of arguments of the three functions is identical.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

References

Leifeld, Philip (2013). texreg: Conversion of Statistical Model Output in R to LaTeX and HTML Tables. Journal of Statistical Software, 55(8), 1-24. <http://www.jstatsoft.org/v55/i08/>.

See Also

[texreg-package](#) [extract](#) [extract-methods](#) [plotreg](#)

Examples

```
#Linear mixed-effects models
library(nlme)
model.1 <- lme(distance ~ age, data = Orthodont, random = ~ 1)
model.2 <- lme(distance ~ age + Sex, data = Orthodont, random = ~ 1)
texreg(list(model.1, model.2), booktabs = TRUE, dcolumn = TRUE)

#Ordinary least squares model (example from the 'lm' help file)
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2,10,20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
table.string <- texreg(lm.D9, return.string = TRUE)
cat(table.string)

#Create a 'fake' Office document containing a regression table
htmlreg(list(model.1, model.2), file = "texreg.doc",
        inline.css = FALSE, doctype = TRUE, html.tag = TRUE,
        head.tag = TRUE, body.tag = TRUE)
unlink("texreg.doc")
```

Index

*Topic **IO**

plotreg, 21
texreg, 26
texreg-package, 2

*Topic **methods**

extract, 5

*Topic **misc**

plotreg, 21
texreg, 26
texreg-package, 2

*Topic **print**

plotreg, 21
texreg, 26
texreg-package, 2

*Topic **utilities**

plotreg, 21
texreg, 26
texreg-package, 2

coefplot (plotreg), 21

coeftoString, 3

createTexreg, 3, 21

extract, 5, 21, 24, 25, 30, 34, 35

extract, aftreg-method (extract), 5

extract, ARIMA-method (extract), 5

extract, Arima-method (extract), 5

extract, averaging-method (extract), 5

extract, bam-method (extract), 5

extract, betamfx-method (extract), 5

extract, betaor-method (extract), 5

extract, betareg-method (extract), 5

extract, brglm-method (extract), 5

extract, btergm-method (extract), 5

extract, censReg-method (extract), 5

extract, clm-method (extract), 5

extract, clmm-method (extract), 5

extract, clogit-method (extract), 5

extract, coefstest-method (extract), 5

extract, coxph-method (extract), 5

extract, coxph.penal-method (extract), 5

extract, dynlm-method (extract), 5

extract, ergm-method (extract), 5

extract, ergmm-method (extract), 5

extract, ets-method (extract), 5

extract, felm-method (extract), 5

extract, fGARCH-method (extract), 5

extract, forecast-method (extract), 5

extract, gam-method (extract), 5

extract, gamlss-method (extract), 5

extract, gee-method (extract), 5

extract, geeglm-method (extract), 5

extract, gel-method (extract), 5

extract, glm-method (extract), 5

extract, glmerMod-method (extract), 5

extract, glmmadmb-method (extract), 5

extract, glmmPQL-method (extract), 5

extract, glmrob-method (extract), 5

extract, gls-method (extract), 5

extract, gmm-method (extract), 5

extract, H20BinomialModel-method

(extract), 5

extract, hurdle-method (extract), 5

extract, ivreg-method (extract), 5

extract, lm-method (extract), 5

extract, lme-method (extract), 5

extract, lme4-method (extract), 5

extract, lmerMod-method (extract), 5

extract, lmrob-method (extract), 5

extract, lnam-method (extract), 5

extract, logitmfx-method (extract), 5

extract, logitor-method (extract), 5

extract, lqmm-method (extract), 5

extract, lrm-method (extract), 5

extract, maBina-method (extract), 5

extract, mer-method (extract), 5

extract, mlogit-method (extract), 5

extract, mnlogit-method (extract), 5

extract, model.selection-method

- (extract), 5
- extract,mtergm-method (extract), 5
- extract,multinom-method (extract), 5
- extract,negbin-method (extract), 5
- extract,negbinirr-method (extract), 5
- extract,negbinmfx-method (extract), 5
- extract,netlogit-method (extract), 5
- extract,nlme-method (extract), 5
- extract,nlmerMod-method (extract), 5
- extract,ols-method (extract), 5
- extract,pgmm-method (extract), 5
- extract,phreg-method (extract), 5
- extract,plm-method (extract), 5
- extract,pmg-method (extract), 5
- extract,poissonirr-method (extract), 5
- extract,poissonmfx-method (extract), 5
- extract,polr-method (extract), 5
- extract,probitmfx-method (extract), 5
- extract,rem.dyad-method (extract), 5
- extract,rlm-method (extract), 5
- extract,rq-method (extract), 5
- extract,sarlm-method (extract), 5
- extract,sclm-method (extract), 5
- extract,selection-method (extract), 5
- extract,sienaFit-method (extract), 5
- extract,simex-method (extract), 5
- extract,stergm-method (extract), 5
- extract,survreg-method (extract), 5
- extract,survreg.penalty-method (extract), 5
- extract,svyglm-method (extract), 5
- extract,systemfit-method (extract), 5
- extract,texreg-method (extract), 5
- extract,tobit-method (extract), 5
- extract,vglm-method (extract), 5
- extract,weibreg-method (extract), 5
- extract,wls-method (extract), 5
- extract,Zelig-method (extract), 5
- extract,zelig-method (extract), 5
- extract,zeroinfl-method (extract), 5
- extract-methods, 25, 30, 34
- extract-methods (extract), 5
- extract.aftreg (extract), 5
- extract.ARIMA (extract), 5
- extract.Arima (extract), 5
- extract.averaging (extract), 5
- extract.bam (extract), 5
- extract.betamfx (extract), 5
- extract.betaor (extract), 5
- extract.betareg (extract), 5
- extract.brglm (extract), 5
- extract.btergm (extract), 5
- extract.censReg (extract), 5
- extract.clm (extract), 5
- extract.clmm (extract), 5
- extract.clogit (extract), 5
- extract.coefest (extract), 5
- extract.coxph (extract), 5
- extract.dynlm (extract), 5
- extract.ergm (extract), 5
- extract.ergmm (extract), 5
- extract.ets (extract), 5
- extract.felm (extract), 5
- extract.fGARCH (extract), 5
- extract.forecast (extract), 5
- extract.gam (extract), 5
- extract.gamlss (extract), 5
- extract.gee (extract), 5
- extract.geeglm (extract), 5
- extract.gel (extract), 5
- extract.glm (extract), 5
- extract.glmerMod (extract), 5
- extract.glmadmb (extract), 5
- extract.glmPQL (extract), 5
- extract.glmrob (extract), 5
- extract.gls (extract), 5
- extract.gmm (extract), 5
- extract.H20BinomialModel (extract), 5
- extract.hurdle (extract), 5
- extract.ivreg (extract), 5
- extract.lm (extract), 5
- extract.lme (extract), 5
- extract.lme4 (extract), 5
- extract.lmerMod (extract), 5
- extract.lmrob (extract), 5
- extract.lnam (extract), 5
- extract.logitmfx (extract), 5
- extract.logitor (extract), 5
- extract.lqmm (extract), 5
- extract.lrm (extract), 5
- extract.maBina (extract), 5
- extract.mer (extract), 5
- extract.mlogit (extract), 5
- extract.mnlogit (extract), 5
- extract.model.selection (extract), 5
- extract.mtergm (extract), 5

extract.multinom (extract), 5
extract.negbin (extract), 5
extract.negbinirr (extract), 5
extract.negbinmfx (extract), 5
extract.netlogit (extract), 5
extract.nlme (extract), 5
extract.nlmerMod (extract), 5
extract.ols (extract), 5
extract.pgmm (extract), 5
extract.phreg (extract), 5
extract.plm (extract), 5
extract.pmg (extract), 5
extract.poissonirr (extract), 5
extract.poissonmfx (extract), 5
extract.polr (extract), 5
extract.probitmfx (extract), 5
extract.rem.dyad (extract), 5
extract.rlm (extract), 5
extract.rq (extract), 5
extract.sarlm (extract), 5
extract.sclm (extract), 5
extract.selection (extract), 5
extract.sienaFit (extract), 5
extract.simex (extract), 5
extract.stergm (extract), 5
extract.survreg (extract), 5
extract.svyglm (extract), 5
extract.systemfit (extract), 5
extract.texreg (extract), 5
extract.tobit (extract), 5
extract.vglm (extract), 5
extract.weibreg (extract), 5
extract.wls (extract), 5
extract.Zelig (extract), 5
extract.zelig (extract), 5
extract.zeroinfl (extract), 5

htmlreg, 21, 25
htmlreg (texreg), 26

plotreg, 2, 21, 35
print.texregTable, 26

screenreg, 21, 25
screenreg (texreg), 26

texreg, 2–4, 18, 21, 25, 26, 26
texreg-package, 2
texregTable (print.texregTable), 26