Package ‘broom’

February 19, 2015

Type Package

Title Convert Statistical Analysis Objects into Tidy Data Frames

Version 0.3.6

Date 2015-02-17

Maintainer David Robinson <admiral.david@gmail.com>

Description Convert statistical analysis objects from R into tidy data frames, so that they can more easily be combined, reshaped and otherwise processed with tools like dplyr, tidyr and ggplot2. The package provides three S3 generics: tidy, which summarizes a model’s statistical findings such as coefficients of a regression; augment, which adds columns to the original data such as predictions, residuals and cluster assignments; and glance, which provides a one-row summary of model-level statistics.

Imports plyr, dplyr, tidyr, psych, stringr

Suggests knitr, survival, gam, glmnet, lfe, Lahman, MASS, sp, maps, mapprotools, multcomp, testthat, lme4, zoo, ggplot2

URL http://github.com/dgretton/broom

BugReports http://github.com/dgretton/broom/issues

VignetteBuilder knitr

License MIT + file LICENSE

Author David Robinson [aut, cre], Matthieu Gomez [ctb], Boris Demeshev [ctb], Hadley Wickham [ctb]

NeedsCompilation no

Repository CRAN

Date/Publication 2015-02-18 19:31:14
R topics documented:

aareg_tidiers .................................................. 3
anova_tidiers .................................................. 4
Arima_tidiers .................................................. 5
augment .......................................................... 6
augment_columns .............................................. 7
bootstrap ....................................................... 7
cch_tidiers ..................................................... 8
compact ........................................................ 10
confint_tidy .................................................... 10
coxph_tidiers .................................................. 11
cv.glmnet_tidiers .............................................. 13
data.frame_tidiers ........................................... 14
delm_tidiers .................................................... 16
finish_glance .................................................. 18
fix_data_frame ............................................... 19
gam_tidiers ..................................................... 20
glance .......................................................... 21
glmnet_tidiers .................................................. 21
glm_tidiers ..................................................... 23
htest_tidiers ................................................... 24
inflate .......................................................... 25
insert_NAs ..................................................... 26
kmeans_tidiers .................................................. 26
lme4_tidiers .................................................... 28
lm_tidiers ....................................................... 30
matrix_tidiers ................................................. 33
multcomp_tidiers .............................................. 34
nlm_tidiers ..................................................... 35
pyears_tidiers .................................................. 37
ridgelm_tidiers ............................................... 38
rlm_tidiers ..................................................... 39
rowwise_df_tidiers ........................................... 40
sexpfit_tidiers ............................................... 42
smooth.spline_tidiers ....................................... 43
sp_tidiers ...................................................... 44
survfit_tidiers ............................................... 45
survreg_tidiers ............................................... 47
tidy ............................................................. 49
tidy.default .................................................. 50
tidy.density .................................................... 50
tidy.ftable ..................................................... 51
tidy.manova ................................................... 52
tidy.map ........................................................ 53
tidy.NULL ....................................................... 53
tidy.pairwise.htest ......................................... 54
tidy.spec ........................................................ 55
**Description**

These tidy the coefficients of Aalen additive regression objects.

**Usage**

```r
## S3 method for class 'aareg'
tidy(x, ...)

## S3 method for class 'aareg'
glance(x, ...)
```

**Arguments**

- `x`: an "aareg" object
- `...`: extra arguments (not used)

**Value**

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

- `tidy.aareg` returns one row for each coefficient, with the columns:
  - `term`: name of coefficient
  - `estimate`: estimate of the slope
  - `statistic`: test statistic for coefficient
  - `std.error`: standard error of statistic
  - `robust.se`: robust version of standard error estimate
  - `z`: z score
  - `p.value`: p-value

- `glance` returns a one-row data frame containing:
  - `statistic`: chi-squared statistic
  - `p.value`: p-value based on chi-squared statistic
  - `df`: degrees of freedom used by coefficients
Examples

```r
if (require("survival", quietly)) {
  afit <- aareg(Surv(time, status) ~ age + sex + ph.ecog, data=lung, dfbeta=TRUE)
  summary(afit)
  tidy(afit)
}
```

Description

Tidies the result of an analysis of variance into an ANOVA table. Only a tidy method is provided, not an augment or glance method.

Usage

```r
## S3 method for class 'anova'
tidy(x, ...)
```

```r
## S3 method for class 'aov'
tidy(x, ...)
```

```r
## S3 method for class 'aovlist'
tidy(x, ...)
```

Arguments

- `x` An object of class "anova", "aov", or "aovlist"
- `...` extra arguments (not used)

Details

Note that the "term" column of an ANOVA table can come with leading or trailing whitespace, which this tidying method trims.

Value

A data.frame with columns

- `term` Term within the model, or "Residuals"
- `df` Degrees of freedom used by this term in the model
- `sumsq` Sum of squares explained by this term
- `meansq` Mean of sum of squares among degrees of freedom
- `statistic` F statistic
- `p.value` P-value from F test

In the case of an "aovlist" object, there is also a `stratum` column describing the error stratum.
Examples

```r
a <- anova(lm(mpg ~ wt + qsec + disp, mtcars))
tidy(a)

a <- aov(mpg ~ wt + qsec + disp, mtcars)
tidy(a)

al <- aov(mpg ~ wt + qsec + Error(disp / am), mtcars)
tidy(al)
```

Description

These methods tidy the coefficients of ARIMA models of univariate time series.

Usage

```r
## S3 method for class 'Arima'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)

## S3 method for class 'Arima'
glance(x, ...)
```

Arguments

- `x`: An object of class "Arima"
- `conf.int`: whether to include a confidence interval
- `conf.level`: confidence level of the interval, used only if conf.int=TRUE
- `...`: extra arguments (not used)

Details

augment is not currently implemented, as it is not clear whether ARIMA predictions can or should be merged with the original data frame.

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

tidy returns one row for each coefficient in the model, with five columns:

- `term`: The term in the nonlinear model being estimated and tested
- `estimate`: The estimated coefficient
- `std.error`: The standard error from the linear model
augment

If conf.int = TRUE, also returns

conf.low  low end of confidence interval
conf.high high end of confidence interval

glance returns one row with the columns

sigma     the square root of the estimated residual variance
logLik    the data's log-likelihood under the model
AIC       the Akaike Information Criterion
BIC       the Bayesian Information Criterion

See Also

arima

Examples

```r
fit <- arima(lh, order = c(1, 0, 0))
tidy(fit)
glance(fit)
```

augment  

**Augment data according to a tidied model**

Description

Given an R statistical model or other non-tidy object, add columns to the original dataset such as predictions, residuals and cluster assignments.

Usage

```r
augment(x, data, ...)
```

Arguments

- `x`  
  model or other R object to convert to data frame
- `data`  
  original dataset, if needed (when possible this is extracted from the model)
- `...`  
  other arguments passed to methods

Details

This generic originated in the ggplot2 package, where it was called "fortify."

See Also

augment.lm
### augment_columns

**Description**

Add fitted values, residuals, and other common outputs to the value returned from `augment`.

**Usage**

```r
augment_columns(x, data, newdata, type, type.predict = type,
                 type.residuals = type, se.fit = TRUE, ...)
```

**Arguments**

- `x`: a model
- `data`: original data onto which columns should be added
- `newdata`: new data to predict on, optional
- `type`: Type of prediction and residuals to compute
- `type.predict`: Type of prediction to compute; by default same as `type`
- `type.residuals`: Type of residuals to compute; by default same as `type`
- `se.fit`: Value to pass to `predict`'s `se.fit`, or `NULL` for no value
- `...`: extra arguments (not used)

**Details**

In the case that a residuals or influence generic is not implemented for the model, fail quietly.

### bootstrap

**Description**

Set up bootstrap replicates of a dplyr operation

**Usage**

```r
bootstrap(df, m)
```

**Arguments**

- `df`: a data frame
- `m`: number of bootstrap replicates to perform
Details

This code originates from Hadley Wickham (with a few small corrections) here:
https://github.com/hadley/dplyr/issues/269
Some examples can be found at
https://github.com/dgrtwo/broom/blob/master/vignettes/bootstrapping.Rmd

Examples

```r
library(dplyr)
mtcars %>% bootstrap(10) %>% do(tidy(lm(mpg ~ wt, .)))
```

Description

Tidiers for case-cohort analyses: summarize each estimated coefficient, or test the overall model.

Usage

```r
## S3 method for class 'cch'
tidy(x, conf.level = 0.95, ...)

## S3 method for class 'cch'
glance(x, ...)
```

Arguments

```r
x a "cch" object
conf.level confidence level for CI
... extra arguments (not used)
```

Details

It is not clear what an augment method would look like, so none is provided. Nor is there currently any way to extract the covariance or the residuals.

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.
tidy returns a data.frame with one row for each term
```
term name of term
estimate estimate of coefficient
```
stderr  standard error
statistic  Z statistic
p.value  p-value
conf.low  low end of confidence interval
conf.high  high end of confidence interval

glance returns a one-row data.frame with the following columns:

score  score
rscore  rscore
p.value  p-value from Wald test
iter  number of iterations
n  number of predictions
nevent  number of events

See Also
cch

Examples

if (require("survival", quietly = TRUE)) {
  # examples come from cch documentation
  subcoh <- nwtco$in.subcohort
  selccoh <= with(nwtco, rel==1|subcoh==1)
  ccoh.data <- nwtco[selccoh]
  ccoh.data$subcohort <- subcoh[selccoh]
  # central-lab histology
  ccoh.data$histol <- factor(ccoh.data$histol,labels=c("FH","UH"))
  # tumour stage
  ccoh.data$stage <= factor(ccoh.data$stage,labels=c("I","II","III","IV"))
  ccoh.data$age <- ccoh.data$age/12 # Age in years
  fit.ccP <- cch(Surv(edrel, rel) ~ stage + histol + age, data = ccoh.data,
                 subcoh = ~subcohort, id= ~seqno, cohort.size = 4028)
  tidy(fit.ccP)
  # coefficient plot
  library(ggplot2)
  ggplot(tidy(fit.ccP), aes(x = estimate, y = term)) + geom_point() +
         geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0) +
         geom_vline()
  # compare between methods
  library(dplyr)
  fits <- data_frame(method = c("Prentice", "SelfPrentice", "LinYing")) %>%
              group_by(method) %>%
              do(tidy(cch(Surv(edrel, rel) ~ stage + histol + age, data = ccoh.data,
subcoh = ~subcohort, id= ~seqno, cohort.size = 4028,
method = $.method))

# coefficient plots comparing methods
ggplot(fits, aes(x = estimate, y = term, color = method)) + geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline()
}

compact

Remove NULL items in a vector or list

Description

Remove NULL items in a vector or list

Usage

compact(x)

Arguments

x a vector or list

confint_tidy

Calculate confidence interval as a tidy data frame

Description

Return a confidence interval as a tidy data frame. This directly wraps the
confint function, but ensures it follows broom conventions: column names of conf.low and conf.high, and no row
names

Usage

confint_tidy(x, conf.level = 0.95, ...)

Arguments

x a model object for which confint can be calculated
conf.level confidence level
... extra arguments passed on to confint

Value

A data frame with two columns: conf.low and conf.high.
Tidiers for coxph object

Description

Tidy the coefficients of a Cox proportional hazards regression model, construct predictions, or summarize the entire model into a single row.

Usage

## S3 method for class 'coxph'
tidy(x, exponentiate = FALSE, conf.int = 0.95, ...)

## S3 method for class 'coxph'
augment(x, data = model.frame(x), newdata,
          type.predict = "lp", type.residuals = "martingale", ...)

## S3 method for class 'coxph'
glance(x, ...)

Arguments

x "coxph" object
exponentiate whether to report the estimate and confidence intervals on an exponential scale
conf.int confidence level to be used for CI
... Extra arguments, not used
data original data for augment
newdata new data on which to do predictions
type.predict type of predicted value (see predict.coxph)
type.residuals type of residuals (see residuals.coxph)

Details

When the modeling was performed with na.action = "na.omit" (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with na.action = "na.exclude", one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to augment and na.action = "na.exclude", a warning is raised and the incomplete rows are dropped.
Value

tidy returns a data.frame with one row for each term, with columns

- **estimate**: estimate of slope
- **std.error**: standard error of estimate
- **statistic**: test statistic
- **p.value**: p-value

augment returns the original data.frame with additional columns added:

- **.fitted**: predicted values
- **.se.fit**: standard errors
- **.resid**: residuals (not present if newdata is provided)

glance returns a one-row data.frame with statistics calculated on the cox regression.

See Also

na.action

Examples

```r
if (require("survival", quietly = TRUE)) {
  cfit <- coxph(Surv(time, status) ~ age + sex + lung)

  tidy(cfit)
tidy(cfit, exponentiate = TRUE)

  lp <- augment(cfit, lung)
  risks <- augment(cfit, lung, type.predict = "risk")
  expected <- augment(cfit, lung, type.predict = "expected")

  glance(cfit)

  # also works on clogit models
  resp <- levels(logan$occupation)
n <- nrow(logan)
indx <- rep(1:n, length(resp))
logan2 <- data.frame(logan[indx,],
  id = indx,
  tocc = factor(rep(resp, each=n)))
logan2$case <- (logan2$occupation == logan2$tocc)

  cl <- clogit(case ~ tocc + tocc:education + strata(id), logan2)
tidy(cl)
glance(cl)

 library(ggplot2)
ggplot(lp, aes(age, .fitted, color = sex)) + geom_point()
ggplot(risks, aes(age, .fitted, color = sex)) + geom_point()
ggplot(expected, aes(time, .fitted, color = sex)) + geom_point()
}
```
Tidiers for glmnet cross-validation objects

Description

Tidying methods for cross-validation performed by glmnet.cv, summarizing the mean-squared-error across choices of the penalty parameter lambda.

Usage

```r
## S3 method for class 'cv.glmnet'
tidy(x, ...)

## S3 method for class 'cv.glmnet'
glance(x, ...)
```

Arguments

- `x`: a "cv.glmnet" object
- `...`: extra arguments (not used)

Details

No `augment` method exists for this class.

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

`tidy` produces a data.frame with one row per choice of lambda, with columns

- `lambda`: penalty parameter lambda
- `estimate`: estimate (median) of mean-squared error or other criterion
- `std.error`: standard error of criterion
- `conf.high`: high end of confidence interval on criterion
- `conf.low`: low end of confidence interval on criterion
- `nzero`: number of parameters that are zero at this choice of lambda

`glance` returns a one-row data.frame with the values

- `nulldev`: null deviance
- `npasses`: total passes over the data across all lambda values
data.frame_tidiers

Examples

```r
if (require("glmnet", quietly = TRUE)) {
  set.seed(2014)

  nobs <- 100
  nvar <- 50
  real <- 5

  x <- matrix(rnorm(nobs * nvar), nobs, nvar)
  beta <- c(rnorm(real, 0, 1), rep(0, nvar - real))
  y <- c(t(beta) %*% t(x)) + rnorm(nvar, sd = 3)

  cvfit1 <- cv.glmnet(x, y)
  head(tidy(cvfit1))
  glance(cvfit1)

  library(ggplot2)
  tidied_cv <- tidy(cvfit1)
  glance_cv <- glance(cvfit1)

  # plot of MSE as a function of lambda
  g <- ggplot(tidied_cv, aes(lambda, estimate)) + geom_line() + scale_x_log10()
  g

  # plot of MSE as a function of lambda with confidence ribbon
  g <- g + geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)
  g

  # plot of MSE as a function of lambda with confidence ribbon and choices
  # of minimum lambda marked
  g <- g + geom_vline(xintercept = glance_cv$lambda.min) +
      geom_vline(xintercept = glance_cv$lambda.1se, lty = 2)
  g

  # plot of number of zeros for each choice of lambda
  ggplot(tidied_cv, aes(lambda, nzero)) + geom_line() + scale_x_log10()

  # coefficient plot with min lambda shown
  tidied <- tidy(cvfit1$glmnet.fit)
  ggplot(tidied, aes(lambda, estimate, group = term)) + scale_x_log10() + geom_line() +
         geom_vline(xintercept = glance_cv$lambda.min) +
         geom_vline(xintercept = glance_cv$lambda.1se, lty = 2)
}
```

data.frame_tidiers Tidiers for data.frame objects
**data.frame_tidiers**

**Description**

These perform tidy summaries of data.frame objects. tidy produces summary statistics about each column, while glance simply reports the number of rows and columns. Note that augment.data.frame will throw an error.

**Usage**

```r
## S3 method for class 'data.frame'
tidy(x, ...)
## S3 method for class 'data.frame'
augment(x, data, ...)
## S3 method for class 'data.frame'
glance(x, ...)
```

**Arguments**

- **x** A data.frame
- **...** extra arguments: for tidy, these are passed on to describe from psych package
- **data** data, not used

**Details**

The tidy method calls the psych method describe directly to produce its per-columns summary statistics.

**Value**

tidy.data.frame produces a data frame with one row per original column, containing summary statistics of each:

- **column** name of original column
- **n** Number of valid (non-NA) values
- **mean** mean
- **sd** standard deviation
- **median** median
- **trimmed** trimmed mean, with trim defaulting to .1
- **mad** median absolute deviation (from the median)
- **min** minimum value
- **max** maximum value
- **range** range
- **skew** skew
- **kurtosis** kurtosis
- **se** standard error
glance returns a one-row data.frame with

- `nrow`: number of rows
- `ncol`: number of columns
- `complete.obs`: number of rows that have no missing values
- `na.fraction`: fraction of values across all rows and columns that are missing

See Also
describe

Examples

```r
td <- tidy(mtcars)
td

glance(mtcars)

library(ggplot2)
# compare mean and standard deviation
ggplot(td, aes(mean, sd)) + geom_point() +
  geom_text(aes(label = column), hjust = 1, vjust = 1) +
  scale_x_log10() + scale_y_log10() + geom_abline()
```

---

felm_tidiers

*Tidying methods for models with multiple group fixed effects*

Description

These methods tidy the coefficients of a linear model with multiple group fixed effects

Usage

```r
## S3 method for class 'felm'
tidy(x, conf.int = FALSE, conf.level = 0.95, fe = FALSE,
    fe.error = fe, ...)

## S3 method for class 'felm'
augment(x, data = NULL, ...)

## S3 method for class 'felm'
glance(x, ...)
```
Arguments

- **x**: felm object
- **conf.int**: whether to include a confidence interval
- **conf.level**: confidence level of the interval, used only if **conf.int**=TRUE
- **fe**: whether to include estimates of fixed effects
- **fe.error**: whether to include standard error of fixed effects
- ...: extra arguments (not used)
- **data**: Original data, defaults to extracting it from the model

Details

If **conf.int**=TRUE, the confidence interval is computed

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

**tidy.felm** returns one row for each coefficient. If **fe**=TRUE, it also includes rows for fixed effects estimates. There are five columns:

- **term**: The term in the linear model being estimated and tested
- **estimate**: The estimated coefficient
- **std.error**: The standard error from the linear model
- **statistic**: t-statistic
- **p.value**: two-sided p-value

If **cont.int**=TRUE, it also includes columns for **conf.low** and **conf.high**, computed with **confint**.

**augment.felm** returns one row for each observation, with multiple columns added to the original data:

- **.fitted**: Fitted values of model
- **.resid**: Residuals

If fixed effect are present,

- **.comp**: Connected component
- **.fe_**: Fixed effects (as many columns as factors)

**glance.lm** returns a one-row data.frame with the columns

- **r.squared**: The percent of variance explained by the model
- **adj.r.squared**: r.squared adjusted based on the degrees of freedom
- **sigma**: The square root of the estimated residual variance
- **statistic**: F-statistic
- **p.value**: p-value from the F test
- **df**: Degrees of freedom used by the coefficients
- **df.residual**: residual degrees of freedom
finish_glance

**Examples**

```r
if (require("lfe", quietly = TRUE)) {
  N = 1e2
  DT <- data.frame(  
    id = sample(5, N, TRUE),  
    v1 = sample(5, N, TRUE),  
    v2 = sample(1e6, N, TRUE),  
    v3 = sample(round(runif(100, max = 100), 4), N, TRUE),  
    v4 = sample(round(runif(100, max = 100), 4), N, TRUE)
  )

  result_felm <- felm(v2 ~ v3, DT)  
  tidy(result_felm)  
  augment(result_felm)
  result_felm <- felm(v2 ~ v3 | id + v1, DT)  
  tidy(result_felm, fe = TRUE)  
  augment(result_felm)
  v1 <- DT$v1  
  v2 <- DT$v2  
  v3 <- DT$v3  
  id <- DT$id
  result_felm <- felm(v2 ~ v3 | id + v1)  
  tidy(result_felm)  
  augment(result_felm)
  glance(result_felm)
}
```

---

**Description**

A helper function for several functions in the glance generic. Methods such as `logLik`, `AIC`, and `BIC` are defined for many prediction objects, such as `lm`, `glm`, and `nls`. This is a helper function that adds them to a glance data.frame can be performed. If any of them cannot be computed, it fails quietly.

**Usage**

```r
finish_glance(ret, x)
```

**Arguments**

- `ret` a one-row data frame (a partially complete glance)
- `x` the prediction model
In one special case, deviance for objects of the \texttt{lmerMod} class from \texttt{lme4} is computed with \texttt{deviance(x, REML=FALSE)}.

### Value

A one-row data frame with additional columns added, such as

- `logLik` log likelihoods
- `AIC` Akaike Information Criterion
- `BIC` Bayesian Information Criterion
- `deviance` deviance
- `df.residual` residual degrees of freedom

Each of these are produced by the corresponding generics.

---

**Description**

Ensure an object is a data frame, with rownames moved into a column

**Usage**

\[
\texttt{fix.data.frame(x, newnames = \texttt{NULL}, newcol = "term")}
\]

**Arguments**

- `x` a \texttt{data.frame} or matrix
- `newnames` new column names, not including the rownames
- `newcol` the name of the new rownames column

**Value**

A data frame, with rownames moved into a column and new column names assigned
Description

These methods tidy the coefficients of a "gam" object (generalized additive model) into a summary, augment the original data with information on the fitted values and residuals, and construct a one-row glance of the model's statistics.

Usage

```r
## S3 method for class 'gam'
tidy(x, ...)

## S3 method for class 'gam'
glance(x, ...)
```

Arguments

- `x` : gam object
- `...` : extra arguments (not used)

Details

The "augment" method is handled by `lm_tidiers`.

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

- `tidy.gam` returns the tidied output of the parametric ANOVA, with one row for each term in the formula. The columns match those in `anova_tidiers`.
- `glance.gam` returns a one-row data.frame with the columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>df</td>
<td>Degrees of freedom used by the coefficients</td>
</tr>
<tr>
<td>logLik</td>
<td>the data's log-likelihood under the model</td>
</tr>
<tr>
<td>AIC</td>
<td>the Akaike Information Criterion</td>
</tr>
<tr>
<td>BIC</td>
<td>the Bayesian Information Criterion</td>
</tr>
<tr>
<td>deviance</td>
<td>deviance</td>
</tr>
<tr>
<td>df.residual</td>
<td>residual degrees of freedom</td>
</tr>
</tbody>
</table>

See Also

- `lm_tidiers`, `anova_tidiers`
**Examples**

```r
if (require("gam", quietly = TRUE)) {
  data(kyphosis)
  g <- gam(Kyphosis ~ s(Age,4) + Number, family = binomial, data = kyphosis)
  tidy(g)
  augment(g)
  glance(g)
}
```

---

**glance**

Construct a single row summary "glance" of a model, fit, or other object

**Description**

Glance methods always return either a one-row data frame, or NULL

**Usage**

```r
glance(x, ...)
```

**Arguments**

- `x` model or other R object to convert to single-row data frame
- `...` other arguments passed to methods

---

**glmnet_tidiers**

Tidiers for LASSO or elasticnet regularized fits

**Description**

Tidying methods for regularized fits produced by glmnet, summarizing the estimates across values of the penalty parameter lambda.

**Usage**

```r
# S3 method for class 'glmnet'
tidy(x, ...)
```

```r
# S3 method for class 'glmnet'
glance(x, ...)
```

**Arguments**

- `x` a "glmnet" object
- `...` extra arguments (not used)
Details

Note that while this representation of GLMs is much easier to plot and combine than the default structure, it is also much more memory-intensive. Do not use for extremely large, sparse matrices.

No augment method is yet provided even though the model produces predictions, because the input data is not tidy (it is a matrix that may be very wide) and therefore combining predictions with it is not logical. Furthermore, predictions make sense only with a specific choice of lambda.

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

tidy produces a data.frame with one row per combination of coefficient (including the intercept) and value of lambda, with the columns:

- term: coefficient name (V1...VN by default, along with "(Intercept)")
- step: which step of lambda choices was used
- estimate: estimate of coefficient
- lambda: value of penalty parameter lambda
- dev.ratio: fraction of null deviance explained at each value of lambda

glance returns a one-row data.frame with the values

- nulldev: null deviance
- npasses: total passes over the data across all lambda values

Examples

```r
if (require("glmnet", quietly = TRUE)) {
  set.seed(2014)
  x <- matrix(rnorm(100*20), 100, 20)
  y <- rnorm(100)
  fit1 <- glmnet(x, y)

  head(tidy(fit1))
  glance(fit1)

  library(dplyr)
  library(ggplot2)

  tidied <- tidy(fit1) %>% filter(term != "(Intercept)"

  ggplot(tidied, aes(step, estimate, group = term)) + geom_line()
  ggplot(tidied, aes(lambda, estimate, group = term)) + geom_line() + scale_x_log10()
  ggplot(tidied, aes(lambda, dev.ratio)) + geom_line()

  # works for other types of regressions as well, such as logistic
  g2 <- sample(1:2, 100, replace=TRUE)
}
```
Description

Tidy a glm object. The tidy and augment methods are handled by \texttt{lm_tidiers}.

Usage

\begin{verbatim}
## S3 method for class 'glm'
glance(x, ...)
\end{verbatim}

Arguments

- \texttt{x} glm object
- \texttt{...} extra arguments, not used

Value

tidy and augment return the same values as do \texttt{tidy.lm} and \texttt{augment.lm}.
glance returns a one-row data.frame with the columns

- \texttt{null.deviance} the deviance of the null model
- \texttt{df.null} the residual degrees of freedom for the null model
- \texttt{logLik} the data's log-likelihood under the model
- \texttt{AIC} the Akaike Information Criterion
- \texttt{BIC} the Bayesian Information Criterion
- \texttt{deviance} deviance
- \texttt{df.residual} residual degrees of freedom

See Also

\texttt{tidy.lm} and \texttt{augment.lm}. Also \texttt{glm}, which computes the values reported by the \texttt{glance} method.

Examples

\begin{verbatim}
g <- glm(am ~ mpg, mtcars, family = "binomial")
glance(g)
\end{verbatim}
Tidying methods for an htest object

Description

Tidies hypothesis test objects, such as those from cor.test, t.test, and wilcox.test, into a one-row data frame.

Usage

```r
## S3 method for class 'htest'
tidy(x, ...)

## S3 method for class 'htest'
glance(x, ...)
```

Arguments

- `x`: An object of class "htest"
- `...`: extra arguments (not used)

Details

No augment method is provided for "htest", since there is no sense in which a hypothesis test generates one value for each observation.

Value

Both `tidy` and `glance` return the same output, a one-row data frame with one or more of the following columns:

- `estimate`: Estimate of the effect size
- `statistic`: Test statistic used to compute the p-value
- `p.value`: P-value
- `parameter`: Parameter field in the htest, typically degrees of freedom
- `conf.low`: Lower bound on a confidence interval
- `conf.high`: Upper bound on a confidence interval
- `estimate1`: Sometimes two estimates are computed, such as in a two-sample t-test
- `estimate2`: Sometimes two estimates are computed, such as in a two-sample t-test

Which columns are included depends on the hypothesis test used.
Examples

```r
tt <- t.test(rnorm(10))
tidy(tt)
glance(tt)  # same output for all htests

tt <- t.test(mpg ~ am, data = mtcars)
tidy(tt)

wt <- wilcox.test(mpg ~ am, data = mtcars)
tidy(wt)

c <- cor.test(mtcars$wt, mtcars$mpg)
tidy(c)
```

inflate

Expand a dataset to include all factorial combinations of one or more variables

Description

Expand a dataset to include all factorial combinations of one or more variables

Usage

`inflate(.data, ..., stringsAsFactors = FALSE)`

Arguments

- `.data` a tbl
- `...` arguments
- `stringsAsFactors` logical specifying if character vectors are converted to factors.

Value

A tbl, grouped by the arguments in ...
**insert_NAs**

*insert a row of NAs into a data frame wherever another data frame has NAs*

**Description**

insert a row of NAs into a data frame wherever another data frame has NAs

**Usage**

```r
insert_NAs(x, original)
```

**Arguments**

- `x` data frame that has one row for each non-NA row in `original`
- `original` data frame with NAs

---

**kmeans_tiders**

*Tidying methods for kmeans objects*

**Description**

These methods summarize the results of k-means clustering into three tidy forms. `tidy` describes the center and size of each cluster, `augment` adds the cluster assignments to the original data, and `glance` summarizes the total within and between sum of squares of the clustering.

**Usage**

```r
## S3 method for class 'kmeans'
tidy(x, col.names = paste0("x", 1:ncol(x$centers)), ...)

## S3 method for class 'kmeans'
augment(x, data, ...)

## S3 method for class 'kmeans'
glance(x, ...)
```

**Arguments**

- `x` kmeans object
- `col.names` The names to call each dimension of the data in `tidy`. Defaults to `x1`, `x2`...
- `...` extra arguments, not used
- `data` Original data (required for `augment`)
Value

All tidying methods return a data.frame without rownames. The structure depends on the method chosen.

tidy returns one row per cluster, with one column for each dimension in the data describing the center, followed by:

- **size**: The size of each cluster
- **withinss**: The within-cluster sum of squares
- **cluster**: A factor describing the cluster from 1:k

augment returns the original data with one extra column:

- **.cluster**: The cluster assigned by the k-means algorithm

glance returns a one-row data.frame with the columns:

- **totss**: The total sum of squares
- **tot.withinss**: The total within-cluster sum of squares
- **betweenss**: The total between-cluster sum of squares
- **iter**: The number of (outer) iterations

See Also

- kmeans

Examples

```r
library(dplyr)
library(ggplot2)

set.seed(2014)
centers <- data.frame(cluster=factor(1:3), size=c(100, 150, 50),
                      x1=c(5, 0, -3), x2=c(-1, 1, -2))
points <- centers %>% group_by(cluster) %>%
do(data.frame(x1=rnorm(.$size[1], .$x1[1]),
              x2=rnorm(.$size[1], .$x2[1])))

k <- kmeans(points %>% dplyr::select(x1, x2), 3)
tidy(k)
head(augment(k, points))
glance(k)

ggplot(augment(k, points), aes(x1, x2)) +
  geom_point(aes(color = .cluster)) +
  geom_text(aes(label = cluster), data = tidy(k), size = 10)
```
lme4_tidiers

Tidying methods for mixed effects models

Description

These methods tidy the coefficients of mixed effects models, particularly responses of the merMod class.

Usage

```r
## S3 method for class 'merMod'
tidy(x, effects = "random", ...)

## S3 method for class 'merMod'
augment(x, data = model.frame(x), newdata, ...)

## S3 method for class 'merMod'
glance(x, ...)
```

Arguments

- `x` An object of class merMod, such as those from `lmer`, `glmer`, or `nlmer`
- `effects` Either "random" (default) or "fixed"
- `...` extra arguments (not used)
- `data` original data this was fitted on; if not given this will attempt to be reconstructed
- `newdata` new data to be used for prediction; optional

Details

When the modeling was performed with `na.action = "na.omit"` (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with `na.action = "na.exclude"`, one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to `augment` and `na.action = "na.exclude"`, a warning is raised and the incomplete rows are dropped.

Value

All tidying methods return a data.frame without rownames. The structure depends on the method chosen.

- `tidy` returns one row for each estimated effect, either random or fixed depending on the `effects` parameter. If `effects = "random"`, it contains the columns
  - `group` the group within which the random effect is being estimated
  - `level` level within group
  - `term` term being estimated
estimate estimated coefficient

If effects="fixed", tidy returns the columns

term fixed term being estimated
estimate estimate of fixed effect
std.error standard error
statistic t-statistic
p.value P-value computed from t-statistic (depending on the model, this may or may not be calculated and included)

augment returns one row for each original observation, with columns (each prepended by a .) added. Included are the columns

.fitted predicted values
.resid residuals
.fixed predicted values with no random effects

Also added for "merMod" objects, but not for "mer" objects, are values from the response object within the model (of type lmResp, glmResp, nlsResp, etc). These include "mu", "offset", "sqrtXwt", "sqrtrwt", "...

glance returns one row with the columns

sigma the square root of the estimated residual variance
logLik the data’s log-likelihood under the model
AIC the Akaike Information Criterion
BIC the Bayesian Information Criterion
deviance deviance

See Also

na.action

Examples

if (require("lme4")) {
  # example regressions are from lme4 documentation
  lmm1 <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
  tidy(lmm1)
  tidy(lmm1, effects = "fixed")
  head(augment(lmm1, sleepstudy))
  glance(lmm1)

  glmm1 <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd),
                 data = cbpp, family = binomial)
  tidy(glmm1)
  tidy(glmm1, effects = "fixed")
  head(augment(glmm1, cbpp))
  glance(glmm1)}
Description

These methods tidy the coefficients of a linear model into a summary, augment the original data with information on the fitted values and residuals, and construct a one-row glance of the model’s statistics.

Usage

```r
## S3 method for class 'lm'
tidy(x, conf.int = FALSE, conf.level = 0.95,
    exponentiate = FALSE, ...)

## S3 method for class 'lm'
augment(x, data = model.frame(x), newdata, type.predict,
    type.residuals, ...)

## S3 method for class 'lm'
glance(x, ...)
```

Arguments

- `x` lm object
- `conf.int` whether to include a confidence interval
- `conf.level` confidence level of the interval, used only if `conf.int=TRUE`
- `exponentiate` whether to exponentiate the coefficient estimates and confidence intervals (typi- cal for logistic regression)
- `...` extra arguments (not used)
- `data` Original data, defaults to the extracting it from the model
- `newdata` If provided, performs predictions on the new data
- `type.predict` Type of prediction to compute for a GLM; passed on to `predict.glm`
- `type.residuals` Type of residuals to compute for a GLM; passed on to `residuals.glm`
Details

If you have missing values in your model data, you may need to refit the model with na.action = na.exclude.

If conf.int=TRUE, the confidence interval is computed with the confint function.

When the modeling was performed with na.action = "na.omit" (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with na.action = "na.exclude", one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to augment and na.action = "na.exclude", a warning is raised and the incomplete rows are dropped.

Value

All tidying methods return a data.frame without rownames. The structure depends on the method chosen.

tidy.lm returns one row for each coefficient, with five columns:

- term: The term in the linear model being estimated and tested
- estimate: The estimated coefficient
- std.error: The standard error from the linear model
- statistic: t-statistic
- p.value: two-sided p-value

If conf.int=TRUE, it also includes columns for conf.low and conf.high, computed with confint.

When newdata is not supplied augment.lm returns one row for each observation, with seven columns added to the original data:

- .hat: Diagonal of the hat matrix
- .sigma: Estimate of residual standard deviation when corresponding observation is dropped from model
- .cooks: Cooks distance, cooks.distance
- .fitted: Fitted values of model
- .se.fit: Standard errors of fitted values
- .resid: Residuals
- .std.resid: Standardised residuals

(Some unusual "lm" objects, such as "rlm" from MASS, may omit .cooks and .std.resid)

When newdata is supplied, augment.lm returns one row for each observation, with three columns added to the new data:

- .fitted: Fitted values of model
- .se.fit: Standard errors of fitted values
- .resid: Residuals of fitted values on the new data

glance.lm returns a one-row data.frame with the columns
The percent of variance explained by the model

r.squared

adj.r.squared

The percent of variance explained by the model adjusted based on the degrees of freedom

sigma

The square root of the estimated residual variance

statistic

F-statistic

p.value

p-value from the F test, describing whether the full regression is significant

df

Degrees of freedom used by the coefficients

logLik

the data’s log-likelihood under the model

AIC

the Akaike Information Criterion

BIC

the Bayesian Information Criterion

deviance

df.residual

residual degrees of freedom

See Also

summary.lm

na.action

Examples

library(ggplot2)
library(dplyr)

mod <- lm(mpg ~ wt + qsec, data = mtcars)

tidy(mod)

glance(mod)

# coefficient plot
d <- tidy(mod) %>% mutate(low = estimate - std.error,
                        high = estimate + std.error)
ggplot(d, aes(estimate, term, xmin = low, xmax = high, height = 0)) +
  geom_point() + geom_vline() + geom_errorbarh()

head(augment(mod))

head(augment(mod, mtcars))

# predict on new data
newdata <- mtcars %>% head(6) %>% mutate(wt = wt + 1)
augment(mod, newdata = newdata)

au <- augment(mod, data = mtcars)

plot(mod, which = 1)
qplot(.fitted, .resid, data = au) +
  geom_hline(yintercept = 0) +
  geom_smooth(se = FALSE)
qplot(.fitted, .std.resid, data = au) +
  geom_hline(yintercept = 0) +
matrix_tidiers

Tidders for matrix objects

Description

These perform tidying operations on matrix objects. tidy turns the matrix into a data.frame while bringing rownames, if they exist, in as a column called .rownames (since results of tidying operations never contain rownames). glance simply reports the number of rows and columns. Note that no augment method exists for matrices.

Usage

```r
## S3 method for class 'matrix'
tidy(x, ...)

## S3 method for class 'matrix'
glance(x, ...)
```
**Arguments**

- **x**: A matrix
- **...**: extra arguments, not used

**Value**

- `tidy.matrix` returns the original matrix converted into a data.frame, except that it incorporates rownames (if they exist) into a column called `.rownames`.
- `glance` returns a one-row data.frame with

  - `nrow`: number of rows
  - `ncol`: number of columns
  - `complete.obs`: number of rows that have no missing values
  - `na.fraction`: fraction of values across all rows and columns that are missing

**Examples**

```r
mat <- as.matrix(mtcars)
tidy(mat)
glance(mat)
```

---

**Description**

These methods originated in ggplot2, as "fortify." In broom, they were renamed "tidy" because they summarize terms and tests, rather than adding columns to a dataset.

**Usage**

```r
## S3 method for class 'glht'
tidy(x, ...)

## S3 method for class 'confint.glht'
tidy(x, ...)

## S3 method for class 'summary.glht'
tidy(x, ...)

## S3 method for class 'cld'
tidy(x, ...)
```

**Arguments**

- **x**: an object of class `glht`, `confint.glht`, `summary.glht` or `cld`
- **...**: extra arguments (not used)
Examples

```r
if (require("multcomp") && require("ggplot2")) {
  amod <- aov(breaks ~ wool + tension, data = warpbreaks)
  wht <- glht(amod, linfct = mcp(tension = "Tukey"))

  tidy(wht)
  ggplot(wht, aes(lhs, estimate)) + geom_point()

  CI <- confint(wht)
  tidy(CI)
  ggplot(CI, aes(lhs, estimate, ymin = lwr, ymax = upr)) +
  geom_pointrange()

  tidy(summary(wht))
  ggplot(mapping = aes(lhs, estimate)) +
  geom_linerange(aes(ymin = lwr, ymax = upr), data = CI) +
  geom_point(aes(size = p), data = summary(wht)) +
  scale_size(trans = "reverse")

  cld <- cld(wht)
  tidy(cld)
}
```

---

### nls_tidders

**Tidying methods for a nonlinear model**

**Description**

These methods tidy the coefficients of a nonlinear model into a summary, augment the original data with information on the fitted values and residuals, and construct a one-row glance of the model’s statistics.

**Usage**

```r
## S3 method for class 'nls'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)

## S3 method for class 'nls'
augment(x, data = NULL, newdata = NULL, ...)

## S3 method for class 'nls'
glance(x, ...)
```

**Arguments**

- **x**: An object of class "nls"
- **conf.int**: whether to include a confidence interval
- **conf.level**: confidence level of the interval, used only if conf.int=TRUE
... extra arguments (not used)

data original data this was fitted on; if not given this will attempt to be reconstructed from nls (may not be successful)

newdata new data frame to use for predictions

Details

When the modeling was performed with na.action = "na.omit" (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with na.action = "na.exclude", one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to augment and na.action = "na.exclude", a warning is raised and the incomplete rows are dropped.

Value

All tidying methods return a data.frame without rownames. The structure depends on the method chosen.

tidy returns one row for each coefficient in the model, with five columns:

- term: The term in the nonlinear model being estimated and tested
- estimate: The estimated coefficient
- std.error: The standard error from the linear model
- statistic: t-statistic
- p.value: two-sided p-value

augment returns one row for each original observation, with two columns added:

- .fitted: Fitted values of model
- .resid: Residuals

If newdata is provided, these are computed on based on predictions of the new data.

glance returns one row with the columns

- sigma: the square root of the estimated residual variance
- isConv: whether the fit successfully converged
- finTol: the achieved convergence tolerance
- logLik: the data’s log-likelihood under the model
- AIC: the Akaike Information Criterion
- BIC: the Bayesian Information Criterion
- deviance: deviance
- df.residual: residual degrees of freedom

See Also

na.action

nls and summary.nls
Examples

```
n <- nls(mpg ~ k * e ^ wt, data = mtcars, start = list(k = 1, e = 2))
tidy(n)
augment(n)
glance(n)
```

```
library(ggplot2)
ggplot(augment(n), aes(wt, mpg)) + geom_point() + geom_line(aes(y = .fitted))
```

```
# augment on new data
newdata <- head(mtcars)
newdata$wt <- newdata$wt + 1
augment(n, newdata = newdata)
```

Description

These tidy the output of `pyears`, a calculation of the person-years of follow-up time contributed by a cohort of subject. Since the output of `pyears$data` is already tidy (if the `data.frame = TRUE` argument is given), this does only a little work and should rarely be necessary.

Usage

```
## S3 method for class 'pyears'
tidy(x, ...)

## S3 method for class 'pyears'
glance(x, ...)
```

Arguments

- `x`: a "pyears" object
- `...`: extra arguments (not used)

Value

`tidy` returns a data.frame with the columns

- `pyyears`: person-years of exposure
- `n`: number of subjects contributing time
- `event`: observed number of events
- `expected`: expected number of events (present only if a `ratetable` term is present)

If the `data.frame = TRUE` argument is supplied to `pyears`, this is simply the contents of `x$data`. `glance` returns a one-row data frame with
total number of person-years tabulated
offtable total number of person-years off table

This contains the values printed by `summary.pyears`.

See Also

`pyears`

Examples

```r
if (require("survival", quietly = TRUE)) {
  temp.yr <- tcut(mgus$dxyr, 55:92, labels=as.character(55:91))
  temp.age <- tcut(mgus$age, 34:101, labels=as.character(34:100))
  ptime <- ifelse(is.na(mgus$pctime), mgus$futime, mgus$pctime)
  pstat <- ifelse(is.na(mgus$pctime), 0, 1)
  pfit <- pyears(Surv(ptime/365.25, pstat) ~ temp.yr + temp.age + sex, mgus,
                  data.frame=TRUE)
  head(tidy(pfit))
  glance(pfit)

  # if data.frame argument is not given, different information is present in
  # output
  pfit2 <- pyears(Surv(ptime/365.25, pstat) ~ temp.yr + temp.age + sex, mgus)
  head(tidy(pfit2))
  glance(pfit2)
}
```

ridgelm_tidiers  Tidying methods for ridgelm objects from the MASS package

Description

These methods tidies the coefficients of a ridge regression model chosen at each value of lambda into a data frame, or constructs a one-row glance of the model’s choices of lambda (the ridge constant).

Usage

```r
## S3 method for class 'ridgelm'
tidy(x, ...)

## S3 method for class 'ridgelm'
glance(x, ...)
```

Arguments

- `x` An object of class "ridgelm"
- `...` extra arguments (not used)
Value

All tidying methods return a data frame without rownames. The structure depends on the method chosen.
tidy.ridgelm returns one row for each combination of choice of lambda and term in the formula, with columns:

- **lambda**: choice of lambda
- **GCV**: generalized cross validation value for this lambda
- **term**: the term in the ridge regression model being estimated
- **estimate**: estimate of coefficient using this lambda

glance.ridgelm returns a one-row data.frame with the columns

- **khkb**: modified HKB estimate of the ridge constant
- **klw**: modified L-W estimate of the ridge constant
- **lambdagcv**: choice of lambda that minimizes GCV

This is similar to the output of select.ridgelm, but it is returned rather than printed.

Examples

```r
names(longley)[1] <- "y"
fit1 <- MASS::lm.ridge(y ~ ., longley)
tidy(fit1)
fit2 <- MASS::lm.ridge(y ~ ., longley, lambda = seq(0.001, .05, .001))
td2 <- tidy(fit2)
g2 <- glance(fit2)

# coefficient plot
library(ggplot2)
ggplot(td2, aes(lambda, estimate, color = term)) + geom_line()

# GCV plot
ggplot(td2, aes(lambda, GCV)) + geom_line()

# add line for the GCV minimizing estimate
ggplot(td2, aes(lambda, GCV)) + geom_line() +
  geom_vline(xintercept = g2$lambdagcv, col = "red", lty = 2)
```

---

**rlm_tidiers**

Tidying methods for an rlm (robust linear model) object

Description

This method provides a glance of an "rlm" object. The tidy and augment methods are handled by lm_tidiers.
rowwise_df_tidiers

Usage

```r
## S3 method for class 'rlm'
glance(x, ...)
```

Arguments

- `x`: rlm object
- `...`: extra arguments (not used)

Value

`glance.rlm` returns a one-row data.frame with the columns

- `sigma`: The square root of the estimated residual variance
- `converged`: whether the IWLS converged
- `logLik`: the data's log-likelihood under the model
- `AIC`: the Akaike Information Criterion
- `BIC`: the Bayesian Information Criterion
- `deviance`: deviance

See Also

- `lm_tidiers`

Examples

```r
library(MASS)

r <- rlm(stack.loss ~ ., stackloss)
tidy(r)
augment(r)
glance(r)
```

Description

These tidy, augment and glance methods are for performing tidying on each row of a rowwise data frame created by dplyr’s group_by and do operations. They first group a rowwise data frame based on all columns that are not lists, then perform the tidying operation on the specified column. This greatly shortens a common idiom of extracting tidy/augment/glance outputs after a do statement.
rowwise_df_tidiers

Usage

```r
## S3 method for class 'rowwise_df'
tidy(x, data, ...)
## S3 method for class 'rowwise_df'
tidy_(x, data, ...)
## S3 method for class 'rowwise_df'
augment(x, data, ...)
## S3 method for class 'rowwise_df'
augment_(x, data, ...)
## S3 method for class 'rowwise_df'
glance(x, data, ...)
## S3 method for class 'rowwise_df'
glance_(x, data, ...)
```

Arguments

- **x**: a rowwise_df
- **data**: the column name of the column containing the models to be tidied. For tidy, augment, and glance it should be the bare name; for _ methods it should be quoted. Note that this argument is named data so as to be consistent with the augment generic.
- **...**: additional arguments to pass on to the respective tidying method

Details

Note that this functionality is currently implemented for data.tables, since the result of the do operation is difficult to distinguish from a regular data.table.

Value

A "grouped_df", where the non-list columns of the original are used as grouping columns alongside the tidied outputs.

Examples

```r
library(dplyr)
regressions <- mtcars %>% group_by(cyl) %>% do(mod = lm(mpg ~ wt, .))
regressions

regressions %>% tidy(mod)
regressions %>% augment(mod)
regressions %>% glance(mod)

# we can provide additional arguments to the tidying function
```
**Description**

This constructs a summary across time points or overall of an expected survival curve. Note that this contains less information than most survfit objects.

**Usage**

```r
## S3 method for class 'survexp'
tidy(x, ...)

## S3 method for class 'survexp'
glance(x, ...)
```

**Arguments**

- `x` 
  "survexp" object
- `...` 
  extra arguments (not used)

**Value**

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

- `tidy` returns a one row for each time point, with columns
  - `time`  
    time point
  - `estimate`  
    estimated survival
  - `n.risk`  
    number of individuals at risk

- `glance` returns a one-row data.frame with the columns:
  - `n.max`  
    maximum number of subjects at risk
  - `n.start`  
    starting number of subjects at risk
  - `timepoints`  
    number of timepoints

**Examples**

```r
if (require("survival", quietly = TRUE)) {
  sexpfit <- survexp(futime ~ 1, rmap=list(sex="male", year=accept.dt, age=(accept.dt-birth.dt)),
                     method='conditional', data=jas)

tidy(sexpfit)
glance(sexpfit)
}
```
smooth.spline_tidiers  tidying methods for smooth.spline objects

Description

This combines the original data given to smooth.spline with the fit and residuals

Usage

## S3 method for class 'smooth.spline'
augment(x, data = x$data, ...)

## S3 method for class 'smooth.spline'
glance(x, ...)

Arguments

x  a smooth.spline object
data  defaults to data used to fit model
...  not used in this method

Details

No tidy method is provided for smooth.spline objects.

Value

augment returns the original data with extra columns:

.fitted  Fitted values of model
.resid  Residuals

glance returns one row with columns

spar  smoothing parameter
lambda  choice of lambda corresponding to spar
df  equivalent degrees of freedom
crit  minimized criterion
pen.crit  penalized criterion
cv.crit  cross-validation score
Examples

```r
spl <- smooth.spline(mtcars$wt, mtcars$mpg, df = 4)
head(augment(spl, mtcars))
head(augment(spl))  # calls original columns x and y

library(ggplot2)
ggplot(augment(spl, mtcars), aes(wt, mpg)) +
  geom_point() + geom_line(aes(y = .fitted))
```

Description

Tidy classes from the sp package to allow them to be plotted using ggplot2. To figure out the correct variable name for region, inspect `as.data.frame(x)`.

Usage

```r
## S3 method for class 'SpatialPolygonsDataFrame'
tidy(x, region = NULL, ...)

## S3 method for class 'SpatialPolygons'
tidy(x, ...)

## S3 method for class 'Polygons'
tidy(x, ...)

## S3 method for class 'Polygon'
tidy(x, ...)

## S3 method for class 'SpatialLinesDataFrame'
tidy(x, ...)

## S3 method for class 'Lines'
tidy(x, ...)

## S3 method for class 'Line'
tidy(x, ...)
```

Arguments

- **x**: SpatialPolygonsDataFrame to convert into a dataframe.
- **region**: name of variable used to split up regions
- **...**: not used by this method
Details

These functions originated in the ggplot2 package as "fortify" functions.

Examples

```r
if (require("maptools")) {
  sids <- system.file("shapes/sids.shp", package="maptools")
  ncl <- readShapePoly(sids,
                      proj4string = CRS("+proj=longlat +datum=NAD27"))
  ncl_df <- tidy(ncl)
}
```

Description

Construct tidied data frames showing survival curves over time.

Usage

```r
## S3 method for class 'survfit'
 tidy(x, ...)

## S3 method for class 'survfit'
 glance(x, ...)
```

Arguments

- `x` : "survfit" object
- `...` : extra arguments, not used

Details

glance does not work on multi-state survival curves, since the values glance outputs would be calculated for each state. tidy does work for multi-state survival objects, and includes a state column to distinguish between them.

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

tidy returns a row for each time point, with columns

- `time` : timepoint
- `n.risk` : number of subjects at risk at time t0
- `n.event` : number of events at time t
n.censor  number of censored events
estimate  estimate of survival
std.error standard error of estimate
conf.high upper end of confidence interval
conf.low lower end of confidence interval

glance returns one-row data.frame with the columns displayed by print.survfit

Examples

```r
if (require("survival", quietly = TRUE)) {
  cfit <- coxph(Surv(time, status) ~ age + sex, lung)
  sfit <- survfit(cfit)

  head(tidy(sfit))
  glance(sfit)

  library(ggplot2)
  ggplot(tidy(sfit), aes(time, estimate)) + geom_line() +
    geom_ribbon(aes(ymin=conf.low, ymax=conf.high), alpha=.25)

  # multi-state
  fitCI <- survfit(Surv(stop, status * as.numeric(event), type = "mstate") ~ 1,
                   data = mgus1, subset = (start == 0))
  td_multi <- tidy(fitCI)
  head(td_multi)
  tail(td_multi)
  ggplot(td_multi, aes(time, estimate, group = state)) +
    geom_line(aes(color = state)) +
    geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)

  # perform simple bootstrapping
  library(dplyr)
  bootstraps <- lung %>% bootstrap(100) %>%
    do(tidy(survfit(coxph(Surv(time, status) ~ age + sex, .))))
  ggplot(bootstraps, aes(time, estimate, group = replicate)) +
    geom_line(alpha = .25)

  bootstraps_bytime <- bootstraps %>% group_by(time) %>%
    summarize(median = median(estimate),
              conf.low = quantile(estimate, .025,
                          conf.high = quantile(estimate, .975)))
```
survreg_tidiers

Tidiers for a parametric regression survival model

Description

Tidies the coefficients of a parametric survival regression model, from the "survreg" function, adds fitted values and residuals, or summarizes the model statistics.

Usage

```r
## S3 method for class 'survreg'
tidy(x, conf_level = 0.95, ...)

## S3 method for class 'survreg'
augment(x, data = model.frame(x), newdata, type.predict = "response", type.residuals = "response", ...)

## S3 method for class 'survreg'
glance(x, conf_level = 0.95, ...)
```

Arguments

- `x` a "survreg" model
- `conf.level` confidence level for CI
- `...` extra arguments (not used)
- `data` original data; if it is not provided, it is reconstructed as best as possible with `model.frame`
- `newdata` New data to use for prediction; optional
- `type.predict` type of prediction, default "response"
- `type.residuals` type of residuals to calculate, default "response"
Details

When the modeling was performed with `na.action = "na.omit"` (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with `na.action = "na.exclude"`, one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to `augment` and `na.action = "na.exclude"`, a warning is raised and the incomplete rows are dropped.

Value

All tidying methods return a data.frame without rownames, whose structure depends on the method chosen.

tidy returns a data.frame with one row for each term

<table>
<thead>
<tr>
<th>term</th>
<th>name of term</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimate</td>
<td>estimate of coefficient</td>
</tr>
<tr>
<td>stderr</td>
<td>standard error</td>
</tr>
<tr>
<td>statistic</td>
<td>Z statistic</td>
</tr>
<tr>
<td>p.value</td>
<td>p-value</td>
</tr>
<tr>
<td>conf.low</td>
<td>low end of confidence interval</td>
</tr>
<tr>
<td>conf.high</td>
<td>high end of confidence interval</td>
</tr>
</tbody>
</table>

augment returns the original data.frame with the following additional columns:

| .fitted    | Fitted values of model        |
| .se.fit    | Standard errors of fitted values|
| resid      | Residuals                     |

glance returns a one-row data.frame with the columns:

| iter       | number of iterations          |
| df         | degrees of freedom            |
| statistic  | chi-squared statistic         |
| p.value    | p-value from chi-squared test |
| logLik     | log likelihood                |
| AIC        | Akaike information criterion  |
| BIC        | Bayesian information criterion|
| df.residual| residual degrees of freedom   |

See Also

`na.action`
Examples

```r
if (require("survival", quietly = TRUE)) {
  sr <- survreg(Surv(futime, fustat) ~ ecog.ps + rx, ovarian,
                dist="exponential")

  td <- tidy(sr)
  augment(sr, ovarian)
  augment(sr)
  glance(td)

  # coefficient plot
  library(ggplot2)
  ggplot(td, aes(estimate, term)) + geom_point() +
        geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0) +
        geom_vline()
}
```

Description

The output of tidy is always a data.frame with disposable row names. It is therefore suited for further manipulation by packages like dplyr, reshape2, ggplot2 and ggvis.

Usage

```r
tidy(x, ...)
```

Arguments

- `x` An object to be converted into a tidy data.frame
- `...` extra arguments

Value

a data.frame
tidy.default  

**Default tidying method**

**Description**

By default, tidy uses `as.data.frame` to convert its output. This is dangerous, as it may fail with an uninformative error message. Generally tidy is intended to be used on structured model objects such as `lm` or `htest` for which a specific S3 object exists.

**Usage**

```r
## Default S3 method:
tidy(x, ...)  
```

**Arguments**

- `x`  
  an object to be tidied
- `...`  
  extra arguments (not used)

**Details**

If you know that you want to use `as.data.frame` on your untidy object, just use it directly.

**Value**

A data frame, from `as.data.frame` applied to the input `x`.

---

tidy.density  

**tidy a density objet**

**Description**

Given a "density" object, returns a tidy data frame with two columns: points `x` where the density is estimated, points `y` for the estimate.

**Usage**

```r
## S3 method for class 'density'
tidy(x, ...)  
```

**Arguments**

- `x`  
  an object of class "density"
- `...`  
  extra arguments (not used)
tidy.ftable

Value

a data frame with "x" and "y" columns

d <- density(faithful$eruptions, bw = "sj")
head(tidy(d))
library(ggplot2) ggplot(tidy(d), aes(x, y)) + geom_line()

See Also

density

tidyNftable

---

tidy.ftable  tidy an ftable object

Description

An ftable contains a "flat" contingency table. This melts it into a data.frame with one column for each variable, then a Freq column. It directly uses the stats::as.data.frame.ftable function.

Usage

## S3 method for class 'ftable'
tidy(x, ...)

Arguments

x

An object of class "ftable"

...

Extra arguments (not used)

See Also

ftable

Examples

tidy(ftable(Titanic, row.vars = 1:3))
tidy.manova  

*tidy a MANOVA object*

**Description**

Constructs a data frame with one row for each of the terms in the model, containing the information from `summary.manova`.

**Usage**

```r
## S3 method for class 'manova'
tidy(x, ...)
```

**Arguments**

- `x`: object of class "manova"
- `...`: additional arguments passed on to `summary.manova`, such as `test`

**Value**

A data.frame with the columns:

- `term`: Term in design
- `statistic`: Approximate F statistic
- `num.df`: Degrees of freedom
- `p.value`: P-value

**See Also**

`summary.manova`

**Examples**

```r
npk2 <- within(npk, foo <- rnorm(24))
npk2.aov <- manova(cbind(yield, foo) ~ block + N*P*K, npk2)
```
tidy.map

Tidy method for map objects.

Description

This function turns a map into a data frame.

Usage

## S3 method for class 'map'
tidy(x, ...)

Arguments

- x: map object
- ...: not used by this method

Details

This code and documentation originated in ggplot2, but was called "fortify." In broom, "fortify" became "augment", which is reserved for functions that *add* columns to existing data (based on a model fit, for example) so these functions were renamed as "tidy."

Examples

```r
if (require("maps") && require("ggplot2")) {
  ca <- map("county", "ca", plot = FALSE, fill = TRUE)
  head(tidy(ca))
  qplot(long, lat, data = ca, geom = "polygon", group = group)

  tx <- map("county", "texas", plot = FALSE, fill = TRUE)
  head(tidy(tx))
  qplot(long, lat, data = tx, geom = "polygon", group = group,
       colour = I("white"))
}
```

tidy.NULL

Tidy on a NULL input

Description

tidy on a NULL input returns an empty data frame, which means it can be combined with other data frames (treated as "empty")
### Usage

```r
## S3 method for class 'NULL'
tidy(x, ...)
```

### Arguments

- `x`  
  A value `NULL`

- `...`  
  extra arguments (not used)

### Value

An empty data.frame

---

**tidy.pairwise.htest**  
*tidy a pairwise hypothesis test*

### Description

Tidy a pairwise.htest object, containing (adjusted) p-values for multiple pairwise hypothesis tests.

### Usage

```r
## S3 method for class 'pairwise.htest'
tidy(x, ...)
```

### Arguments

- `x`  
  a "pairwise.htest" object

- `...`  
  extra arguments (not used)

### Details

Note that in one-sided tests, the alternative hypothesis of each test can be stated as "group1 is greater/less than group2".

Note also that the columns of group1 and group2 will always be a factor, even if the original input is (e.g.) numeric.

### Value

A data frame with one row per group/group comparison, with columns

- `group1`  
  First group being compared

- `group2`  
  Second group being compared

- `p.value`  
  (Adjusted) p-value of comparison
Description

Given a "spec" object, which shows a spectrum across a range of frequencies, returns a tidy data frame with two columns: "freq" and "spec"

Usage

## S3 method for class 'spec'
tidy(x, ...)

Arguments

x an object of class "spec"

... extra arguments (not used)

Value

a data frame with "freq" and "spec" columns

Examples

spc <- spectrum(lh)
tidy(spc)

library(ggplot2)
ggplot(tidy(spc), aes(freq, spec)) + geom_line()
Description

A table, typically created by the `table` function, contains a contingency table of frequencies across multiple vectors. This directly calls the `as.data.frame.table` method, which熔s it into a data frame with one column for each variable and a `Freq` column.

Usage

```r
## S3 method for class 'table'
tidy(x, ...)
```

Arguments

- `x` An object of class "table"
- `...` Extra arguments (not used)

See Also

`as.data.frame.table`

Examples

```r
tab <- with(airquality, table(cut(Temp, quantile(Temp)), Month))
tidy(tab)
```

description tidy a table object

 tidy.ts tidy a ts timeseries object

Description

Turn a ts object into a tidy data frame. Right now simply uses `as.data.frame.ts`.

Usage

```r
## S3 method for class 'ts'
tidy(x, ...)
```

Arguments

- `x` a "ts" object
- `...` extra arguments (not used)
tidy.TukeyHSD

Value

a tidy data frame

See Also

as.data.frame.ts

tidyNtukeyhsd

tidy a TukeyHSD object

Description

Returns a data.frame with one row for each pairwise comparison

Usage

## S3 method for class 'TukeyHSD'
tidy(x, ...)

Arguments

x object of class "TukeyHSD"

... additional arguments (not used)

Value

A data.frame with one row per comparison, containing columns

- comparison: Levels being compared, separated by -
- estimate: Estimate of difference
- conf.low: Low end of confidence interval of difference
- conf.high: High end of confidence interval of difference
- adj.p.value: P-value adjusted for multiple comparisons

See Also

TukeyHSD

Examples

```r
fm1 <- aov(breaks ~ wool + tension, data = warpbreaks)
thsd <- TukeyHSD(fm1, "tension", ordered = TRUE)
tidy(thsd)
```
unrownname  
(strip rownames from an object)

Description
strip rownames from an object

Usage
unrownname(x)

Arguments
x  a data frame

zoo_tidiers  
(Tidying methods for a zoo object)

Description
Tidies zoo (Z's ordered observations) time series objects. zoo objects are not tidy by default because they contain one row for each index and one series per column, rather than one row per observation per series.

Usage
## S3 method for class 'zoo'
tidy(x, ...)

Arguments
x  An object of class "zoo"
...  extra arguments (not used)

Value
tidy returns a data frame with one row for each observation in each series, with the following columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>Index (usually date) for the zoo object</td>
</tr>
<tr>
<td>series</td>
<td>Name of the series</td>
</tr>
<tr>
<td>value</td>
<td>Value of the observation</td>
</tr>
</tbody>
</table>
Examples

```r
if (require("zoo", quietly = TRUE)) {
  set.seed(1071)

  # data generated as shown in the zoo vignette
  Z.index <- as.Date(sample(12450:12500, 10))
  Z.data <- matrix(rnorm(30), ncol = 3)
  colnames(Z.data) <- c("Aa", "Bb", "Cc")
  Z <- zoo(Z.data, Z.index)

  tidy(Z)

  if (require("ggplot2", quietly = TRUE)) {
    ggplot(tidy(Z), aes(index, value, color = series)) + geom_line()
    ggplot(tidy(Z), aes(index, value)) + geom_line() +
      facet_wrap(~ series, ncol = 1)

    Zrolled <- rollmean(Z, 5)
    ggplot(tidy(Zrolled), aes(index, value, color = series)) + geom_line()
  }
}
```
Index

aareg_tidiers, 3
anova_tidiers, 4, 20
arima, 6
Arima_tidiers, 5
as.data.frame.table, 56
as.data.frame.ts, 57
augment, 6
augment.coxph (coxph_tidiers), 11
augment.data.frame
  (data.frame_tidiers), 14
augment.felm (felm_tidiers), 16
augment.kmeans (kmeans_tidiers), 26
augment.lm, 6, 23
augment.lm (lm_tidiers), 30
augment.lmerMod (lmer_tidiers), 28
augment.nls (nls_tidiers), 35
augment.rowwise_df
  (rowwise_df_tidiers), 40
augment.smooth.spline
  (smooth.spline_tidiers), 43
augment.survreg (survreg_tidiers), 47
augment_.rowwise_df
  (rowwise_df_tidiers), 40
augment_columns, 7
bootstrap, 7
cch, 9
cch_tidiers, 8
cld, 34
compact, 10
confint, 10, 11, 17, 31
confint_tidy, 10
cooks.distance, 31
coxph_tidiers, 11
cv.glmnet_tidiers, 13
data.frame_tidiers, 14
density, 51
describe, 15, 16
felm_tidiers, 16
finish_glance, 18
fix_data_frame, 19
ftable, 51
gam_tidiers, 20
glance, 21
    glance.aareg (aareg_tidiers), 3
    glance.Arima (Arima_tidiers), 5
    glance.cch (cch_tidiers), 8
    glance.coxph (coxph_tidiers), 11
    glance.cv.glmnet (cv.glmnet_tidiers), 13
    glance.data.frame (data.frame_tidiers), 14
    glance.felm (felm_tidiers), 16
    glance.gam (gam_tidiers), 20
    glance.glm (glm_tidiers), 23
    glance.glmnet (glmnet_tidiers), 21
    glance.htest (htest_tidiers), 24
    glance.kmeans (kmeans_tidiers), 26
    glance.lm (lm_tidiers), 30
    glance.matrix (matrix_tidiers), 33
    glance.merMod (merMod_tidiers), 28
    glance.nls (nls_tidiers), 35
    glance.pyears (pyears_tidiers), 37
    glance.ridgelm (ridgelm_tidiers), 38
    glance.rlm (rlm_tidiers), 39
    glance.rowwise_df (rowwise_df_tidiers), 40
    glance.smooth.spline
      (smooth.spline_tidiers), 43
    glance.survexp (survexp_tidiers), 42
    glance.survfit (survfit_tidiers), 45
    glance.survreg (survreg_tidiers), 47
    glance_.rowwise_df
      (rowwise_df_tidiers), 40
glm, 23
glm_tidiers, 23
glmnet_tidiers, 21

60
htest_tidiers, 24
inflate, 25
insert_NAs, 26
kmeans, 27
kmeans_tidiers, 26
lm_tidiers, 20, 23, 30, 39, 40
lme4_tidiers, 28
matrix_tidiers, 33
model.frame, 47
multcomp_tidiers, 34
na.action, 12, 29, 32, 36, 48
nls, 36
nls_tidiers, 35
pairwise.t.test, 55
pairwise.wilcox.test, 55
predict.coxph, 11
predict.glm, 30
print.survfit, 46
pyears, 38
pyears_tidiers, 37
residuals.coxph, 11
residuals.glm, 30
ridgelm_tidiers, 38
rlm_tidiers, 39
rowwise_df_tidiers, 40
sexpfit_tidiers, 42
smooth.spline_tidiers, 43
sp_tidiers, 44
summary.lm, 32
summary.manova, 52
summary.nls, 36
survfit_tidiers, 45
survreg_tidiers, 47
table, 56
tidy, 49
tidy.aareg (aareg_tidiers), 3
tidy.anova (anova_tidiers), 4
tidy.aov (anova_tidiers), 4
tidy.aovlist (anova_tidiers), 4
tidy.Arima (Arima_tidiers), 5
tidy.cch (cch_tidiers), 8
tidy.cld (multcomp_tidiers), 34
tidy.confint.glht (multcomp_tidiers), 34
tidy.coxph (coxph_tidiers), 11
tidy.cv.glmmnet (cv.glmmnet_tidiers), 13
tidy.data.frame (data.frame_tidiers), 14
tidy.default, 50
tidy.density, 50
tidy.felm (felm_tidiers), 16
tidy.ftable, 51
tidy.gam (gam_tidiers), 20
tidy.glht (multcomp_tidiers), 34
tidy.glmmnet (glmnet_tidiers), 21
tidy.htest (htest_tidiers), 24
tidy.kmeans (kmeans_tidiers), 26
tidy.Line (sp_tidiers), 44
tidy.Lines (sp_tidiers), 44
tidy.lm, 23
tidy.lm (lm_tidiers), 30
tidy.manova, 52
tidy.map, 53
tidy.matrix (matrix_tidiers), 33
tidy.merMod (lme4_tidiers), 28
tidy.nls (nls_tidiers), 35
tidy.NULL, 53
tidy.pairwise.htest, 54
tidy.Polygon (sp_tidiers), 44
tidy.Polygons (sp_tidiers), 44
tidy.pyears (pyears_tidiers), 37
tidy.ridgelm (ridgelm_tidiers), 38
tidy.rowwise_df (rowwise_df_tidiers), 40
tidy.SpatialLinesDataFrame
 (sp_tidiers), 44
tidy.SpatialPolygons (sp_tidiers), 44
tidy.SpatialPolygonsDataFrame
 (sp_tidiers), 44
tidy.spec, 55
tidy.summary.glht (multcomp_tidiers), 34
tidy.survexp (sexpfit_tidiers), 42
tidy.survfit (survfit_tidiers), 45
tidy.survreg (survreg_tidiers), 47
tidy.table, 56
tidy.ts, 56
tidy.TukeyHSD, 57
tidy.zoo (zoo_tidiers), 58
tidy.rowwise_df (rowwise_df_tidiers), 40
TukeyHSD, 57
unrowname, 58
zoo_tidiers, 58