|  |  |
| --- | --- |
| aggregate {stats} | R Documentation |

**Compute Summary Statistics of Data Subsets**

**Description**

Splits the data into subsets, computes summary statistics for each, and returns the result in a convenient form.

**Usage**

aggregate(x, ...)

## Default S3 method:

aggregate(x, ...)

## S3 method for class 'data.frame'

aggregate(x, by, FUN, ..., simplify = TRUE, drop = TRUE)

## S3 method for class 'formula'

aggregate(formula, data, FUN, ...,

subset, na.action = na.omit)

## S3 method for class 'ts'

aggregate(x, nfrequency = 1, FUN = sum, ndeltat = 1,

ts.eps = getOption("ts.eps"), ...)

**Arguments**

|  |  |
| --- | --- |
| x | an R object. |
| by | a list of grouping elements, each as long as the variables in the data frame x. The elements are coerced to factors before use. |
| FUN | a function to compute the summary statistics which can be applied to all data subsets. |
| simplify | a logical indicating whether results should be simplified to a vector or matrix if possible. |
| drop | a logical indicating whether to drop unused combinations of grouping values. |
| formula | a [formula](http://127.0.0.1:14695/library/stats/help/formula), such as y ~ x or cbind(y1, y2) ~ x1 + x2, where the y variables are numeric data to be split into groups according to the grouping x variables (usually factors). |
| data | a data frame (or list) from which the variables in formula should be taken. |
| subset | an optional vector specifying a subset of observations to be used. |
| na.action | a function which indicates what should happen when the data contain NA values. The default is to ignore missing values in the given variables. |
| nfrequency | new number of observations per unit of time; must be a divisor of the frequency of x. |
| ndeltat | new fraction of the sampling period between successive observations; must be a divisor of the sampling interval of x. |
| ts.eps | tolerance used to decide if nfrequency is a sub-multiple of the original frequency. |
| ... | further arguments passed to or used by methods. |

**Details**

aggregate is a generic function with methods for data frames and time series.

The default method, aggregate.default, uses the time series method if x is a time series, and otherwise coerces x to a data frame and calls the data frame method.

aggregate.data.frame is the data frame method. If x is not a data frame, it is coerced to one, which must have a non-zero number of rows. Then, each of the variables (columns) in x is split into subsets of cases (rows) of identical combinations of the components of by, and FUN is applied to each such subset with further arguments in ... passed to it. The result is reformatted into a data frame containing the variables in by and x. The ones arising from by contain the unique combinations of grouping values used for determining the subsets, and the ones arising from x the corresponding summaries for the subset of the respective variables in x. If simplify is true, summaries are simplified to vectors or matrices if they have a common length of one or greater than one, respectively; otherwise, lists of summary results according to subsets are obtained. Rows with missing values in any of the by variables will be omitted from the result. (Note that versions of **R** prior to 2.11.0 required FUN to be a scalar function.)

aggregate.formula is a standard formula interface to aggregate.data.frame.

aggregate.ts is the time series method, and requires FUN to be a scalar function. If x is not a time series, it is coerced to one. Then, the variables in x are split into appropriate blocks of length frequency(x) / nfrequency, and FUN is applied to each such block, with further (named) arguments in ... passed to it. The result returned is a time series with frequency nfrequency holding the aggregated values. Note that this make most sense for a quarterly or yearly result when the original series covers a whole number of quarters or years: in particular aggregating a monthly series to quarters starting in February does not give a conventional quarterly series.

FUN is passed to [match.fun](http://127.0.0.1:14695/library/stats/help/match.fun), and hence it can be a function or a symbol or character string naming a function.

**Value**

For the time series method, a time series of class "ts" or class c("mts", "ts").

For the data frame method, a data frame with columns corresponding to the grouping variables in by followed by aggregated columns from x. If the by has names, the non-empty times are used to label the columns in the results, with unnamed grouping variables being named Group.*i* for by[[*i*]].

**Author(s)**

Kurt Hornik, with contributions by Arni Magnusson.

**References**

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

**See Also**

[apply](http://127.0.0.1:14695/library/stats/help/apply), [lapply](http://127.0.0.1:14695/library/stats/help/lapply), [tapply](http://127.0.0.1:14695/library/stats/help/tapply).

**Examples**

## Compute the averages for the variables in 'state.x77', grouped

## according to the region (Northeast, South, North Central, West) that

## each state belongs to.

aggregate(state.x77, list(Region = state.region), mean)

## Compute the averages according to region and the occurrence of more

## than 130 days of frost.

aggregate(state.x77,

list(Region = state.region,

Cold = state.x77[,"Frost"] > 130),

mean)

## (Note that no state in 'South' is THAT cold.)

## example with character variables and NAs

testDF <- data.frame(v1 = c(1,3,5,7,8,3,5,NA,4,5,7,9),

v2 = c(11,33,55,77,88,33,55,NA,44,55,77,99) )

by1 <- c("red", "blue", 1, 2, NA, "big", 1, 2, "red", 1, NA, 12)

by2 <- c("wet", "dry", 99, 95, NA, "damp", 95, 99, "red", 99, NA, NA)

aggregate(x = testDF, by = list(by1, by2), FUN = "mean")

# and if you want to treat NAs as a group

fby1 <- factor(by1, exclude = "")

fby2 <- factor(by2, exclude = "")

aggregate(x = testDF, by = list(fby1, fby2), FUN = "mean")

## Formulas, one ~ one, one ~ many, many ~ one, and many ~ many:

aggregate(weight ~ feed, data = chickwts, mean)

aggregate(breaks ~ wool + tension, data = warpbreaks, mean)

aggregate(cbind(Ozone, Temp) ~ Month, data = airquality, mean)

aggregate(cbind(ncases, ncontrols) ~ alcgp + tobgp, data = esoph, sum)

## Dot notation:

aggregate(. ~ Species, data = iris, mean)

aggregate(len ~ ., data = ToothGrowth, mean)

## Often followed by xtabs():

ag <- aggregate(len ~ ., data = ToothGrowth, mean)

xtabs(len ~ ., data = ag)

## Compute the average annual approval ratings for American presidents.

aggregate(presidents, nfrequency = 1, FUN = mean)

## Give the summer less weight.

aggregate(presidents, nfrequency = 1,

FUN = weighted.mean, w = c(1, 1, 0.5, 1))