

Package: portion (via r-universe)

June 30, 2026

Type Package

Title Extracting a Data Portion

Version 0.1.3

Description Provides simple methods to extract data portions from various objects. The relative portion size and the way the portion is selected can be chosen.

License GPL (>= 3)

Encoding UTF-8

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports stats

URL <https://github.com/loelschlaeger/portion>

BugReports <https://github.com/loelschlaeger/portion/issues>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Lennart Oelschläger [aut, cre]

Maintainer Lennart Oelschläger <oelschlaeger.lennart@gmail.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-06-30 19:13:09 UTC

RemoteUrl <https://github.com/cran/portion>

RemoteRef HEAD

RemoteSha 0059723b902b497ed7e92e60e37d14066c58ed3e

Contents

portion	2
Index	5

portion	<i>Extract a data portion</i>
---------	-------------------------------

Description

Extract a relative portion from vectors, matrices, data frames, or lists of these objects.

Usage

```
portion(x, proportion, how = "random", centers = 2L, ...)  
  
## Default S3 method:  
portion(x, ...)  
  
## S3 method for class 'numeric'  
portion(x, proportion, how = "random", centers = 2L, ...)  
  
## S3 method for class 'character'  
portion(x, proportion, how = "random", ...)  
  
## S3 method for class 'logical'  
portion(x, proportion, how = "random", centers = 2L, ...)  
  
## S3 method for class 'matrix'  
portion(  
  x,  
  proportion,  
  how = "random",  
  centers = 2L,  
  byrow = TRUE,  
  ignore = integer(),  
  ...  
)  
  
## S3 method for class 'data.frame'  
portion(  
  x,  
  proportion,  
  how = "random",  
  centers = 2L,  
  byrow = TRUE,  
  ignore = integer(),  
  ...  
)  
  
## S3 method for class 'list'  
portion(x, proportion, how = "random", centers = 2L, ...)
```

Arguments

x	An object to be portioned.
proportion	[numeric(1)] The relative portion size as a number between 0 and 1. The absolute size is rounded up with <code>ceiling()</code> .
how	[character(1)] Exactly one of: <ul style="list-style-type: none"> • "random" (default), select at random • "first", select the first elements, rows, or columns • "last", select the last elements, rows, or columns • "similar", select similar elements, rows, or columns • "dissimilar", select dissimilar elements, rows, or columns Options "similar" and "dissimilar" are based on clustering via <code>kmeans</code> and require numeric clustering data without NA, NaN, or infinite values.
centers	[integer(1)] Only relevant if <code>how = "similar"</code> or <code>how = "dissimilar"</code> . A positive whole number passed to <code>kmeans</code> for clustering. If <code>centers</code> exceeds the number of distinct values or rows, it is reduced automatically.
...	Further arguments to be passed to or from other methods.
byrow	[logical(1)] Only relevant if <code>x</code> has two dimensions (rows and columns). In this case, set to <code>TRUE</code> to portion row-wise (default) or <code>FALSE</code> to portion column-wise.
ignore	[integer()] Only relevant for two-dimensional <code>x</code> and <code>how = "similar"</code> or <code>how = "dissimilar"</code> . Indices to exclude from the clustering data, but not from the returned object. With <code>byrow = TRUE</code> , these are column indices. With <code>byrow = FALSE</code> , these are row indices. This is useful, for example, to ignore identifier or non-numeric columns while selecting similar rows.

Value

A portion of `x`, preserving attributes where possible. Vectors return selected elements, matrices and data frames return selected rows or columns, and lists return a list with each element portioned. The selected indices are stored in the "indices" attribute of each returned object.

Examples

```
# can portion vectors, matrices, data.frames, and lists of such types
portion(
  list(
    1:10,
    matrix(LETTERS[1:12], nrow = 3, ncol = 4),
    data.frame(a = 1:6, b = -6:-1)
  ),
```

```
    proportion = 0.5,  
    how = "first"  
  )  
  
# can portion similar and dissimilar elements (based on kmeans clustering)  
set.seed(1)  
x <- c(1, 1, 2, 2)  
portion(x, proportion = 0.5, how = "similar")  
portion(x, proportion = 0.5, how = "dissimilar")  
  
# ignore non-numeric columns when clustering data frame rows  
x <- data.frame(value = c(1, 1, 5, 5), group = c("a", "a", "b", "b"))  
portion(x, proportion = 0.5, how = "similar", ignore = 2)  
  
# object attributes are preserved  
x <- structure(1:10, "test_attribute" = "test")  
x[1:5]  
portion(x, proportion = 0.5, how = "first")
```

Index

kmeans, 3

portion, 2