Package 'visa'

March 19, 2025

Type Package

Title Vegetation Imaging Spectroscopy Analyzer

Version 1.0.0

Description Provides easy-to-use tools for data analysis and visualization for hyperspectral remote sensing (also known as imaging spectroscopy), with a particular focus on vegetation hyperspectral data analysis. It consists of a set of functions, ranging from the organization of hyperspectral data in the proper data structure for spectral feature selection, calculation of vegetation index, multivariate analysis, as well as to the visualization of spectra and results of analysis in the 'ggplot2' style.

License GPL-3

LazyData true

Encoding UTF-8

URL https://github.com/kang-yu/visa

BugReports https://github.com/kang-yu/visa/issues

Depends R (>= 3.5.0)

Imports ggplot2, ggpmisc, methods, magrittr, Matrix, plot3D, plotly, reshape2, RColorBrewer

Suggests devtools, testthat, knitr, rmarkdown, stringi

RoxygenNote 7.3.2

VignetteBuilder knitr

Collate 'visa.R' 'Spectra-class.R' 'cm.nsr.R' 'cm.rbd3.R' 'cm.sr.R' 'data-specdf.R' 'data-speclib.R' 'find.bestBands.R' 'ggplot-method.R' 'ggplot.cm.R' 'ggplot.lmfit.R' 'ggplot.spectra.R' 'ndvi2.R' 'nsr.R' 'plt.2dcm.R' 'plt.3dcm_best.R' 'spectra.R' 'sr.R' 'wavelength.R'

NeedsCompilation no

Author Kang Yu [aut, cre]

Maintainer Kang Yu <kang.yu@outlook.com>

Repository CRAN

Date/Publication 2025-03-19 19:50:02 UTC

Contents

as.spectra.dataframe . . . 2
cm.nsr . . . 3
cm.rbd3 . . . 4
cm.sr . . . 6
find.bestBands . . . 7
ggplot-method . . . 8
ggplot.cm . . . 9
ggplot.spectra . . . 10
ndvi2 . . . 11
NSpec.DF . . . 12
NSpec.Lib . . . 13
nsr . . . 13
plt.2dcm . . . 14
plt.3dcm_best . . . 15
spectra . . . 16
Spectra-class . . . 17
SpectraDataframe-class . . . 18
SpectraLibrary-class . . . 19
SpectraMatrix-class . . . 19
sr . . . 20
wavelength . . . 21

Index **23**

`as.spectra.dataframe` *Create a SpectraDataframe*

Description

Constructor `as.spectra.dataframe` function creates a SpectraDataframe object, which is equivalent to the use of as.specdf.

Usage

```
as.spectra.dataframe(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = character(0),
  data = data.frame(0),
  ...
)
```

Arguments

spectra	A matrix
wavelength	A numeric vector
w.unit	A character string
data	A data.frame
...	Other options for similar format of variables

Value

sdf	Returns a SpectraDataframe.

Examples

```
sdf <- as.spectra.dataframe(matrix(1:10, 1), 1:10, "nm", data.frame(a = 1, b =2))
str(sdf)
```

---

cm.nsr *Selecting the best 2-Band combinations for Normalized Simple Ratio (NSR)*

Description

This function develops an optimization algorithm based on correlation analysis between the spectral matrix `spectra` and the vegetation variable of interest `x`. It determines the best spectral band combinations (i, j) of the full spectrum that are most predictive for `x`.

Usage

```
cm.nsr(S, x, w = wavelength(S), w.unit = NULL, cm.plot = FALSE)
```

Arguments

S	A matrix of spectral data, where each row is a spectrum across all spectral bands.
x	A numeric vector (e.g., a vegetation variable).
w	A numeric vector of wavelengths; by default it is derived using `wavelength(S)`.
w.unit	A character string specifying the unit of wavelengths (default is `NULL`).
cm.plot	Logical. If `TRUE`, the correlation coefficient matrix is plotted.

Details

For every possible pair of distinct bands (i, j), the function calculates

$$\mathrm{NSR} = \frac{R_j - R_i}{R_j + R_i}$$

and then computes the squared Pearson correlation (R^2) between `x` and the resulting NSR values.

If the two bands are identical or the standard deviation of computed `VI` (for a given band combination) is zero, the correlation is set to 0, thereby avoiding warnings.

Value

cm — A correlation coefficient matrix with squared Pearson correlation values.

See Also

cor

Examples

```
## Not run:
  library(visa)
  data(NSpec.DF)
  X <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 5)]  # resampled to 5 nm steps
  y <- NSpec.DF$N  # nitrogen
  cm <- cm.nsr(X, y, cm.plot = TRUE)

## End(Not run)
```

---

cm.rbd3 — *Calculate 3-Band Correlation Array for Spectral Data correlating with another variable x*

Description

This function computes the squared Pearson correlation (R^2) between a response vector x and a derived variable V for every possible combination of three distinct spectral bands. The derived variable V is calculated using the formula:

$$V = \frac{R_k - R_j}{R_j - R_i}$$

where R_i, R_j, and R_k represent the reflectance values at bands i, j, and k, respectively.

Usage

```
cm.rbd3(
  S,
  x,
  w = wavelength(S),
  w.unit = NULL,
  cm.plot = FALSE,
  plot.method = "default"
)
```

Arguments

S	A spectral data object or matrix. Each column corresponds to a spectral band.
x	A numeric vector representing the response variable (e.g., chlorophyll).
w	A numeric vector of wavelengths; by default, it is derived using `wavelength(S)`.
w.unit	Character string specifying the unit of wavelengths (optional).
cm.plot	Logical. If TRUE, a 3D slice plot of the correlation array is generated.
plot.method	Character string specifying the plotting method. Currently, the plotting option uses `plot3D`.

Details

The function prints the maximum R^2 value and the corresponding band wavelengths. Optionally, it can produce a 3D slice plot of the correlation array using `plot3D::slice3D`.

For every combination of three distinct bands (i, j, k), the function computes

$$V = \frac{R_k - R_j}{R_j - R_i}$$

and then calculates the squared Pearson correlation between `x` and `V`. The maximum R^2 value and its associated band combination are printed.

If `cm.plot` is set to TRUE, the function generates a 3D slice plot of the correlation array using the best band combination, where the slices correspond to the wavelengths of the bands.

Value

A 3-dimensional array of squared correlation (R^2) values with dimensions corresponding to the combinations of bands i, j, and k.

Examples

```
## Not run:
  library(visa)
  data(NSpec.DF)
  x <- NSpec.DF$N # nitrogen
  # Below resamples spectra to 20 nm for fast computation
  S <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 20)]
  # S is a spectral data object and x is a numeric vector.
  Rsq3 <- cm.rbd3(S, x, cm.plot = TRUE)

## End(Not run)
```

cm.sr *Selecting the best 2-Band combinations for Simple Ratio (SR)*

Description

This function develops an optimization algorithm based on correlation analysis between the spectral matrix `spectra` and the vegetation variable of interest `x`. It determines the best spectral band combinations of the full spectrum that are most predictive for `x`.

Usage

```
cm.sr(S, x, w = wavelength(S), w.unit = NULL, cm.plot = FALSE)
```

Arguments

`S`	A matrix of spectral data, where each row is a spectrum across all spectral bands.
`x`	A numeric vector (e.g., a vegetation variable).
`w`	A numeric vector of wavelengths; by default it is derived using `wavelength(S)`.
`w.unit`	A character string specifying the unit of wavelengths (default is `NULL`).
`cm.plot`	Logical. If `TRUE`, the correlation coefficient matrix is plotted.

Details

This function runs a calculation of

$$SR = \lambda_i / \lambda_j$$

using the spectra data for all the possible pairs/combinations of any two bands (i, j) within the full spectrum range. Next, correlation analysis is then performed between all possible SR values and another vector variable `y`, and it returns the squared Pearson correlation (R^2) which indicates the predictive performance of each SR and its corresponding two-band combination. The output is the wavelength (nm) indicating the best two bands that produce the highest value of R^2.

Value

`cm`	Returns a correlation coefficients matrix.

See Also

`cm.nsr`

Examples

```
## Not run:
  library(visa)
  data(NSpec.DF)
  # Using the example spectra matrix of the spectra.dataframe
  X <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 10)]  # resampled to 10 nm steps
  y <- NSpec.DF$N  # nitrogen
```

```
  cm <- cm.sr(X, y, cm.plot = FALSE)

## End(Not run)
```

`find.bestBands` *Find Best Band Combinations*

Description

This function identifies the best band combination from the cm.nsr or cm.sr calculated correlation coefficients - a numeric matrix (2D), or the cm.rbd3 returned array (3D) by locating the maximum value in the input `R` and returning the corresponding wavelengths from `w`. For a 2D matrix, it returns a two-element vector (i, j); for a 3D array, a three-element vector (i, j, k).

Usage

```
find.bestBands(R, w)
```

Arguments

`R`	A numeric matrix (2D) or array (3D) containing metric values (e.g., correlation values).
`w`	A numeric vector of wavelengths corresponding to the bands.

Details

The function first verifies that `R` has dimensions. It then computes the maximum value in `R`, retrieves the indices corresponding to that value, and extracts the wavelengths from `w` based on the dimensionality of `R`. If `R` is 2D, the order is assumed to be (i, j); if `R` is 3D, the order is (i, j, k).

Value

A vector of wavelengths corresponding to the best band combination.

Examples

```
# Example for a 2D matrix:
R_mat <- matrix(c(0.2, 0.8, 0.5, 0.3), nrow = 2)
wavelengths <- c(450, 550)
bestBands <- find.bestBands(R_mat, wavelengths)

# Example for a 3D array:
R_arr <- array(runif(27), dim = c(3, 3, 3))
wavelengths <- c(400, 450, 500)
bestBands <- find.bestBands(R_arr, wavelengths)
```

`ggplot-method` *Plot functions*

Description

This function plots a linear model fit using ggplot2. It creates a scatter plot with a regression line, and displays the regression equation along with the R^2 value.

Usage

```
## S3 method for class 'lmfit'
ggplot(data, mapping = NULL, ..., environment = parent.frame())
```

Arguments

`data`	Either a numeric vector (to be used as x) or an object containing the data (e.g., a data frame).
`mapping`	Either a numeric vector (to be used as y when data is numeric) or an aesthetic mapping created with `ggplot2::aes()`. If `mapping` is missing and `data` is a data frame, the default mapping `aes(x, y)` is used.
`...`	Other arguments passed to ggplot2 components.
`environment`	The environment in which to evaluate the plot. Defaults to `parent.frame()`.

Details

When provided with two numeric vectors, the function treats them as x and y values, respectively, constructs a data frame, and applies a default mapping. Alternatively, if a data frame is provided, an aesthetic mapping (or default mapping) will be used.

Value

A ggplot object.

Examples

```
## Not run:
  library(visa)
  # Using numeric vectors for x and y:
  ggplot.lmfit(1:10, 2:11)

  # Using a data frame:
  df <- data.frame(x = runif(10, 1, 10), y = runif(10, 2, 11) + 0.5)
  ggplot.lmfit(df, aes(x, y))

## End(Not run)
```

ggplot.cm *Create a ggplot Plot from Correlation Coefficients (Deprecated)*

Description

Deprecated: This function is deprecated and will be removed in a future release. Please use `plt.2dcm` for 2D correlation matrices or the appropriate new functions for 3D plots.

Usage

```
## S3 method for class 'cm'
ggplot(
  data,
  mapping = NULL,
  ...,
  show.stat = TRUE,
  environment = parent.frame()
)
```

Arguments

`data`	A numeric 2D matrix of correlation coefficients. (For 3D arrays, a warning is issued.)
`mapping`	Optional ggplot2 aesthetic mapping.
`...`	Additional arguments passed to `ggplot`.
`show.stat`	Logical. If `TRUE`, prints the best R^2 value and corresponding bands.
`environment`	The environment in which to evaluate the plot. Defaults to `parent.frame()`.

Details

This function creates a ggplot visualization from a 2D correlation matrix. It attempts to extract numeric wavelengths from the column names of the input matrix.

This function extracts numeric wavelengths from the column names of `data`. If these cannot be determined, sequential indices are used instead.

Value

A ggplot object visualizing the correlation matrix. For 3D arrays, returns `NULL`.

Examples

```
## Not run:
  library(visa)
  data(NSpec.DF)
  x <- NSpec.DF$N  # nitrogen
   # resampled to 10 nm steps
```

```
  S <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 10)]
  cm2d <- cm.sr(S, x, cm.plot = FALSE)
  p2d <- ggplot.cm(cm2d)
  print(p2d)


## End(Not run)
```

`ggplot.spectra` *Create a new ggplot plot with a geom_line() layer from spectra data*

Description

`ggplot()` initializes a ggplot object. It can be used to declare the input spectra object for a graphic and to optionally specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.

Usage

```
## S3 method for class 'spectra'
ggplot(
  data,
  mapping = NULL,
  ...,
  wl = NULL,
  w.unit = "nm",
  environment = parent.frame()
)
```

Arguments

`data`	Default spectra database to use for plot. If not a spectra database, the methods used will be those defined in package `ggplot2`. See `ggplot`. If not specified, must be supplied in each layer added to the plot.
`mapping`	Default list of aesthetic mappings to use for plot. If not specified, in the case of spectra objects, a default mapping will be used.
`...`	Other arguments passed on to methods. Not currently used.
`wl`	numeric The wavelength vector.
`w.unit`	character The wavelength unit of the spectra.
`environment`	If an variable defined in the aesthetic mapping is not found in the data, ggplot will look for it in this environment. It defaults to using the environment in which `ggplot()` is called.

Details

ggplot() is typically used to construct a plot incrementally, using the + operator to add layers to the existing ggplot object. This is advantageous in that the code is explicit about which layers are added and the order in which they are added. For complex graphics with multiple layers, initialization with ggplot is recommended.

Note

Current implementation does not merge default mapping with user supplied mapping. If user supplies a mapping, it is used as is. To add to the default mapping, aes() can be used by itself to compose the ggplot.

See Also

?ggpmisc::ggplot()

Examples

```
library(visa)
library(ggplot2)
ggplot.spectra(NSpec.DF)
```

---

ndvi2 *Calculate and plot a 2-band NDVI.*

Description

This function calculates a 2-band NDVI using the nsr function.

Usage

```
ndvi2(s, b1, b2)
```

Arguments

s	Spectral data in the format of visa's Spectra object, spectra.dataframe or spectra.matrix.
b1	A integer number which defines the wavelength of the 1st spectral band.
b2	A integer number which defines the wavelength of the 2nd spectral band.

Details

Calculate a NDVI with two specific bands of choice. The new NDVI follows the the standard formula

$$NDVI = (\lambda_i + \lambda_j)/(\lambda_i - \lambda_j)$$

. Bands i and j correspond to the b1 and b2 input arguments, respectively. Wavelength indexes are determined based on the first argument 's'.

Value

ndvi The returned values are the new NDVI.

Examples

```
library(visa)
s <- NSpec.DF$spectra
ndvi2(s, 780, 680)
```

NSpec.DF *Example data in the Spectra.Dataframe format*

Description

A dataset containing the plant Nitrogen content and spectra. The Spectra matrix is stored as a variable (in a column) of a data.frame.

Usage

```
NSpec.DF
```

Format

A data frame with 19 rows and 2 variables:

N Plant nitrogen content

spectra A variable of Matrix of plant spectra ...

See Also

data.frame and NSpec.Lib

Examples

```
library(visa)
data(NSpec.DF)
str(NSpec.DF)
```

NSpec.Lib *Example data in the Spectra/SpectraLibrary format.*

Description

A S4 data structure containing the plant spectra and nitorgen (N) content. Spectra is organized as a matrix and is stored as a slot, named 'spectra'. The corresponding N content is stored in the slot 'data', which is a data.frame used for storing supporting data and plant/vegetation traits, such as here the plant N content.

Usage

```
NSpec.Lib
```

Format

A Spectra object with 19 rows and 4 slots (spectra, wavelength, w.unit, data).

spectra A matrix of plant spectral data

wavelength A vector of wavelength for the 'spectra' data

w.unit A character string of wavelength unit (default "nm")

data A data.frame of vegetation traits, here plant nitrogen content ...{currently not used}

Examples

```
library(visa)
data(NSpec.Lib)
str(NSpec.Lib)
```

nsr *Calculate Normalized Simple Ratio (NSR) index.*

Description

It is a normalization of SR by doing NSR = (1-SR)/(1+SR), with the same two spectral bands.

Usage

```
nsr(s, b1, b2)

lm.nsr(s, b1, b2, y)
```

Arguments

s	Spectral data in the format of visa's Spectra object, spectra.dataframe or spectra.matrix.
b1	A integer number which defines the wavelength of the 1st spectral band.
b2	A integer number which defines the wavelength of the 2nd spectral band.
y	A numeric variable to correlate with SR

Details

As it exactly reads in its name, it is a normalization of the SR and ranges in (0,1).

Value

nsr	Returns a NSR index.
p	Returns a ggplot object.

Examples

```
s <- NSpec.DF$spectra
nsr1 <- nsr(s, 480, 550)

s <- NSpec.DF
y <- NSpec.DF$N
lm.nsr(s,600,500,y)
```

---

plt.2dcm *Create a ggplot Plot from a 2D Correlation Matrix*

Description

This function creates a ggplot visualization from a 2D correlation matrix, such as those produced by `cm.sr` or `cm.nsr`. The function attempts to extract numeric wavelengths from the column names. If the extraction fails, sequential indices are used.

Usage

```
plt.2dcm(
  data,
  mapping = NULL,
  ...,
  show.stat = TRUE,
  environment = parent.frame()
)
```

Arguments

data	A numeric 2D matrix of correlation coefficients.
mapping	Optional ggplot2 aesthetic mapping.
...	Additional arguments passed to ggplot.
show.stat	Logical. If TRUE, prints the best R^2 value and corresponding bands.
environment	The environment in which to evaluate the plot. Defaults to parent.frame().

Details

It replaces the former ggplot.cm().

The function extracts numeric wavelengths from the column names of data. If these cannot be determined, sequential indices are used instead.

Value

A ggplot object visualizing the correlation matrix.

Examples

```
## Not run:
  library(visa)
  data(NSpec.DF)
  x <- NSpec.DF$N  # nitrogen
  S <- NSpec.DF$spectra[, seq(1, ncol(NSpec.DF$spectra), 10)]  # resampled to 10 nm steps
  cm2d <- cm.sr(S, x, cm.plot = FALSE)
  p2d <- plt.2dcm(cm2d)
  print(p2d)

## End(Not run)
```

---

plt.3dcm_best *Create a 3D Slice Plot of Correlation Array*

Description

This function creates an interactive Plotly 3D slice plot from a 3D correlation array. The function uses the array's dimnames to define the coordinate values. If no dimnames are present, a warning is issued and sequential indices are used. The plot displays three surfaces corresponding to slices along each dimension (i, j, k) at the best band combination (where the correlation value is maximal).

Usage

```
plt.3dcm_best(R3, colorscale = "Spectral")
```

Arguments

R3	A 3D numeric array of correlation coefficients.
colorscale	A character string specifying the colorscale to use. If "Spectral", a custom colorscale is created using colorRampPalette(rev(RColorBrewer::brewer.pal(11, "Spectral")), space = "Lab")(100). Otherwise, the provided value is used (default is "Spectral").

Details

The function first checks if the input 3D array has proper dimnames. If not, it issues a warning and assigns sequential indices as dimnames. It then melts the array to find the best band combination based on the maximum correlation value. Using the dimnames, it finds the numeric indices for the best bands and creates grid matrices for each slice. A custom colorscale is built when colorscale = "Spectral", and the Plotly figure is constructed with a single color bar.

Value

An interactive Plotly figure showing three surfaces corresponding to constant slices along dimensions i, j, and k.

Examples

```
## Not run:
  # Assume cm3d is a 3D correlation array with proper dimnames.
  plt.3dcm_best(cm3d)

## End(Not run)
```

---

spectra *Access the spectra data of 'SpectraLibrary'.*

Description

Functions to access slot data of the Class Spectra.

Usage

```
spectra(object, ...)

## S4 method for signature 'Spectra'
spectra(object, ...)

## S4 method for signature 'data.frame'
spectra(object, ...)

## S4 method for signature 'matrix'
spectra(object, ...)
```

Arguments

object	A Spectra object, spectra.data.frame, or spectra.matrix.
...	Other options.

Details

Construct generic functions for the Spectra object, spectra.data.frame, and spectra.matrix.

Examples

```
# For the S4 class 'Spectra'
library(visa)
data(NSpec.Lib)
spectra_matrix <- spectra(NSpec.Lib)
# For the spectra data.frame
data(NSpec.DF)
spectra_matrix <- spectra(NSpec.DF)
```

Spectra-class *Create a Spectra or SpectraLibrary*

Description

Constructor `as.spectra` creates a Spectra object.

Constructor `as.spectra.library` creates a SpectraLibrary object.

Usage

```
as.spectra(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = "nm",
  data = data.frame(),
  ...
)

as.spectra.library(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = "nm",
  data = data.frame(),
  ...
)
```

Arguments

spectra	A matrix
wavelength	A numeric vector
w.unit	A character string
data	A data.frame
...	Other parameters

Slots

spectra A matrix

wavelength A numeric vector

w.unit A character string

data A data.frame

Examples

```
s <- as.spectra(matrix(1:100, 4), 1:25, "nm", data.frame(x = letters[1:4]))
str(s)

s <- as.spectra.library(matrix(1:100, 4), 1:25, "nm", data.frame(x = letters[1:4]))
str(s)
```

SpectraDataframe-class

Class 'SpectraDataframe'

Description

SpectraDataframe is an extended 'Spectra' class, with associated vegetation data ('data') in a data.frame.

Slots

spectra A matrix

wavelength A numeric vector

w.unit A character string

data A data.frame of vegetation data corresponding to the spectra

SpectraLibrary-class *Class 'SpectraLibrary'*

Description

SpectraLibrary is an extended 'Spectra' class, with associated vegetation data ('data') in a data.frame.

Slots

`spectra` A matrix

`wavelength` A numeric vector

`w.unit` A character string

`data` A data.frame of vegetation data corresponding to the spectra

SpectraMatrix-class *Class 'SpectraMatrix'*

Description

SpectraMatrix is a extended 'Spectra' class.

Constructor `as.spectra.matrix` creates a SpectraMatrix object.

Usage

```
as.spectra.matrix(
  spectra = matrix(0),
  wavelength = numeric(0),
  w.unit = character(0)
)
```

Arguments

`spectra`	A matrix
`wavelength`	A numeric vector
`w.unit`	A character string

Value

`sdf`	Returns a SpectraDataframe.

Examples

```
smatrix <- as.spectra.matrix(matrix(1:10, 1), 1:10, "nm")
str(smatrix)
```

sr *Calculate Simple Ratio (SR).*

Description

Simple Ratio is the ratio of the spectra (mostly reflectance) between two bands in the format of

$$SR = \lambda_i / \lambda_j$$

Usage

```
sr(s, b1, b2)

lm.sr(s, b1, b2, y)
```

Arguments

s	Spectral data in the format of visa's Spectra object, spectra.dataframe or spectra.matrix.
b1	A integer number which defines the wavelength of the 1st spectral band.
b2	A integer number which defines the wavelength of the 2nd spectral band.
y	A numeric variable to correlate with SR

Details

Simple ratio and NDVI looking indices are the two groups of mostly used spectral indices in remote sensing.

Value

sr	Returns a simple ratio index.
p	Returns a ggplot object.

Examples

```
library(visa)
s <- NSpec.DF$spectra
sr1 <- sr(s, 480, 550)

s <- NSpec.DF
y <- NSpec.DF$N
lm.sr(s,600,500,y)
```

wavelength *Retrieve Wavelength Information from Spectra Objects*

Description

This function extracts the wavelength information from various representations of spectra. It supports the S4 class `Spectra`, as well as data.frame and matrix representations.

Usage

```
wavelength(object, ...)

## S4 method for signature 'Spectra'
wavelength(object, ...)

## S4 method for signature 'data.frame'
wavelength(object, ...)

## S4 method for signature 'matrix'
wavelength(object, ...)
```

Arguments

`object`	An object containing spectra data. This can be an S4 object of class `Spectra`, a `data.frame`, or a `matrix`.
`...`	Additional arguments for future extensions (currently not used).

Details

For an object of class `Spectra`, the method returns the value stored in the `wavelength` slot. For a data.frame or matrix, it extracts numeric values from the column names (by removing non-digit characters) of the spectra data.

Value

A numeric vector representing the wavelength information extracted from the object.

Examples

```
## Not run:
  library(visa)

  # For an S4 Spectra object
  wavelengths <- wavelength(NSpec.Lib)

  # For spectra stored in a data.frame
  wavelengths <- wavelength(NSpec.DF)
```

```
  # For spectra stored in a matrix
  wavelengths <- wavelength(spectra_matrix)

## End(Not run)
```

Index

∗ **datasets**
NSpec.DF, 12
NSpec.Lib, 13

as.specdf, *2*
as.specdf *(*as.spectra.dataframe*)*, 2
as.spectra *(*Spectra-class*)*, 17
as.spectra.dataframe, 2
as.spectra.matrix
*(*SpectraMatrix-class*)*, 19

cm.nsr, 3, *6, 7*
cm.rbd3, 4, *7*
cm.sr, 6, *7*
cor, *4*

Data-speclib, *(*NSpec.Lib*)*, 13
Data-Spectra *(*NSpec.Lib*)*, 13
Data-Spectra.Dataframe *(*NSpec.DF*)*, 12
Data-SpectraLibrary, *(*NSpec.Lib*)*, 13
data.frame, *12*, *18, 19*

find.bestBands, 7

ggplot, *10*
ggplot-method, 8
ggplot.cm, 9
ggplot.lmfit *(*ggplot-method*)*, 8
ggplot.spectra, 10

lm.nsr *(*nsr*)*, 13
lm.sr *(*sr*)*, 20

ndvi2, 11
NSpec.DF, 12
NSpec.Lib, *12*, 13
nsr, *11*, 13

plt.2dcm, 14
plt.3dcm_best, 15

speclib *(*SpectraLibrary-class*)*, 19
Spectra *(*Spectra-class*)*, 17
spectra, 16
spectra,data.frame,ANY-method
*(*spectra*)*, 16
spectra,data.frame-method *(*spectra*)*, 16
spectra,matrix,ANY-method *(*spectra*)*, 16
spectra,matrix-method *(*spectra*)*, 16
spectra,Spectra,ANY-method *(*spectra*)*, 16
Spectra,Spectra-class *(*Spectra-class*)*,
17
spectra,Spectra-method *(*spectra*)*, 16
Spectra-class, 17
spectra.dataframe
*(*SpectraDataframe-class*)*, 18
spectra.maxtrix *(*SpectraMatrix-class*)*,
19
SpectraDataframe,
*(*SpectraDataframe-class*)*, 18
SpectraDataframe-class, 18
SpectraLibrary-class, 19
SpectraLibrary-class,spectra.library,
*(*SpectraLibrary-class*)*, 19
SpectraMatrix-class, 19
SpectraMatrix-class,
*(*SpectraMatrix-class*)*, 19
sr, 20

wavelength, 21
wavelength,data.frame,ANY-method
*(*wavelength*)*, 21
wavelength,data.frame-method
*(*wavelength*)*, 21
wavelength,matrix,ANY-method
*(*wavelength*)*, 21
wavelength,matrix-method *(*wavelength*)*,
21
wavelength,Spectra,ANY-method
*(*wavelength*)*, 21
wavelength,Spectra-method *(*wavelength*)*,
21

wavlen *(*wavelength*)*, 21