

Package ‘encode’

January 3, 2018

Type Package

Title Represent Ordered Lists and Pairs as Strings

Version 0.3.3

Author Tim Bergsma

Maintainer Tim Bergsma <bergsmat@gmail.com>

Description Interconverts between ordered lists and compact string notation.

Useful for capturing code lists, and pair-wise codes and decodes, for text storage.

Analogous to factor levels and labels. Generics encode() and decode()

perform interconversion, while codes() and decodes() extract components of an encoding.

The function encoded() checks whether something is interpretable as an encoding.

License GPL-3

LazyData TRUE

Imports

Suggests

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-03 13:06:13 UTC

R topics documented:

codes	2
codes.default	2
decode	3
decode.default	3
decodes	4
decodes.default	4
encode	5
encode.character	6
encode.default	7
encode.list	7

encoded	8
encoded.default	8

Index	9
--------------	----------

codes	<i>Extract Codes from an Object</i>
-------	-------------------------------------

Description

Extracts Codes from an object. Default method is supplied.

Usage

```
codes(x, ...)
```

Arguments

x	object
...	passed arguments

See Also

[codes.default](#)

codes.default	<i>Extract Codes by Default from an Object</i>
---------------	--

Description

Extracts codes from an object using the default method.

Usage

```
## Default S3 method:
codes(x, simplify = TRUE, ...)
```

Arguments

x	object
simplify	whether to convert length one list to vector
...	passed arguments

Value

list, or vector if simplify = TRUE

decode	<i>Decode an Object</i>
--------	-------------------------

Description

Decodes an object. Default method supplied.

Usage

```
decode(x, ...)
```

Arguments

x	object
...	passed arguments

See Also

[decode.default](#)

decode.default	<i>Decode an Object by Default</i>
----------------	------------------------------------

Description

Decodes an object using the default method. Typically x is a character vector containing codes that can be extracted from encoding. Corresponding decodes are returned as a factor. If encoding is NULL, it is replaced with an encoding such that levels and labels are both unique(x).

Usage

```
## Default S3 method:  
decode(x, encoding = NULL, ...)
```

Arguments

x	object
encoding	length one character that is itself encoded
...	passed arguments

Value

factor

decodes *Extract Decodes from an Object*

Description

Extracts decodes from an object. Default method is supplied.

Usage

```
decodes(x, ...)
```

Arguments

x	object
...	passed arguments

See Also

[decodes.default](#)

decodes.default *Extract Decodes by Default from an Object*

Description

Extracts decodes from an object using the default method.

Usage

```
## Default S3 method:  
decodes(x, simplify = TRUE, ...)
```

Arguments

x	object
simplify	whether to convert length one list to vector
...	passed arguments

Value

list, or vector if simplify = TRUE

`encode`*Encode Factor-like Levels and Labels as a Simple String*

Description

For compact storage, `encode` combines a set of levels and labels (codes and decodes) into a simple string. The default method converts its argument to character. The `list` method operates element-wise, expecting an equal number of label elements, each of which have the same length as the corresponding element of `x`.

Usage

```
encode(x, ...)
```

Arguments

<code>x</code>	object
<code>...</code>	passed arguments

Details

An empty 'encoding' consists of four identical characters, e.g. `////`.

A non-empty encoding must be at least 5 characters long, beginning and ending with two instances of `sep` e.g. `//1//`. Levels are likewise separated from each other by double separators, e.g. `//1//2//`.

If a label (decode) is available for a level, it follows the corresponding level: the two are separated by a single instance of `sep`, e.g. `//1/a//2/b//`.

Encodings may be combined as elements of a character vector, i.e. an encoded vector. Choice of separator may vary among elements, but must be consistent within elements.

Labels (decodes) may be zero-length, but not levels (codes), e.g. `//1///` is valid but `///a//` is not. A zero-length decode is extracted as an empty string.

See Also

[encode.character](#) [encode.default](#) [encode.list](#) [codes](#) [decodes](#) [decode](#) [encoded](#)

Examples

```
a <- encode(
  x = list(
    c('M', 'F'),
    c(1:4)
  ),
  labels = list(
    c('male', 'female'),
    c('caucasian', 'asian', 'african', NA)
  )
)
```

```

)
)
b <- encode(c(1:2),c('pediatric','adult'))
a
b
c <- c('a',NA,'##b##')
encoded(a)
encoded(b)
encoded(c)
encoded(' //4// ')
codes(a)
codes(b)
codes(b,simplify=FALSE)
codes(c)
codes('..1..')
decodes(a)
decodes(b)
decodes(c)
decode(1:4,'//1/a//2/b//3/c//')
```

encode.character *Encode Character.*

Description

Encodes character. If sep is NULL, it is replaced with the first of these that is not otherwise present in the result: /!\~!@#&

Usage

```
## S3 method for class 'character'
encode(x, labels = NULL, sep = NULL, ...)
```

Arguments

x	object
labels	same length as x if supplied
sep	a single character not present in x or labels
...	passed arguments

Value

character

encode.default	<i>Encode Default.</i>
----------------	------------------------

Description

Encodes using default method: coerces to character and encodes the result.

Usage

```
## Default S3 method:  
encode(x, labels = NULL, ...)
```

Arguments

x	object
labels	same length as x if supplied
...	passed arguments

Value

character

encode.list	<i>Encode a List</i>
-------------	----------------------

Description

Encodes a list.

Usage

```
## S3 method for class 'list'  
encode(x, labels = NULL, ...)
```

Arguments

x	object
labels	same length as x if supplied
...	passed arguments

Value

list

encoded	<i>Check If Object is Encoded</i>
---------	-----------------------------------

Description

Checks if object is encoded.

Usage

```
encoded(x, ...)
```

Arguments

x	object
...	passed arguments

See Also

[encoded.default](#)

encoded.default	<i>Check If Default Object is Encoded</i>
-----------------	---

Description

Checks if object is encoded, using default methodology. Always returns logical, telling whether the corresponding element represents an encoding of levels and labels. Objects with zero length give FALSE.

Usage

```
## Default S3 method:
encoded(x, ...)
```

Arguments

x	object
...	passed arguments

Value

logical

Index

codes, [2](#), [5](#)

codes.default, [2](#), [2](#)

decode, [3](#), [5](#)

decode.default, [3](#), [3](#)

decodes, [4](#), [5](#)

decodes.default, [4](#), [4](#)

encode, [5](#)

encode.character, [5](#), [6](#)

encode.default, [5](#), [7](#)

encode.list, [5](#), [7](#)

encoded, [5](#), [8](#)

encoded.default, [8](#), [8](#)