Package ‘ascii’

February 19, 2015

Maintainer David Hajage <dhajage@gmail.com>
License GPL (>= 2)
Title Export R objects to several markup languages
Type Package
Author David Hajage
Description Coerce R object to asciidoc, txt2tags, restructuredText, org, textile or pandoc syntax. Package comes with a set of drivers for Sweave.
Version 2.1

Date 2009-07-20
Depends R (>= 2.13), utils, methods
Suggests survival, Hmisc, xtable, R2HTML, cacheSweave, weaver
Collate 'asciiAnova.r' 'asciiDataFrame.r' 'asciiDefault.r'
  'asciiDensity.r' 'asciiDescr.r' 'asciiEpi.r' 'asciiGlm.r'
  'asciiHmisc.r' 'asciiHtest.r' 'asciiList.r' 'asciiLm.r'
  'asciiMatrix.r' 'asciiMemisc.r' 'asciiPrecomp.r'
  'asciiSmoothSpline.r' 'asciiSummaryTable.r' 'asciiSurvival.r'
  'asciiTable.r' 'asciiTs.r' 'asciiVector.r' 'cacheSweaveAscii.r'
  'cbind.r' 'export.r' 'generic.r' 'groups.r' 'interleave.r'
  'paste.matrix.r' 'plim.r' 'print.character.matrix.r'
  'RweaveAscii.r' 'show.asciidoc.r' 'show.org.r' 'show.pandoc.r'
  'show.r' 'show.rest.r' 'show.t2t.r' 'show.textile.r'
  'SweaveAscii.r' 'tocharac.r' 'weaverAscii.r' 'zzz.r' 'print.r'

Repository CRAN
Date/Publication 2011-09-29 11:27:34

NeedsCompilation no
Description

Convert an R object to an ascii object, which can then be printed with asciidoc, txt2tags, reStructuredText, org, textile or pandoc syntax.

Usage

```r
## Default S3 method:
ascii(x, include.rownames = TRUE, include.colnames = TRUE, rownames = NULL, colnames = NULL, format =

## S3 method for class 'describe'
ascii(x, condense = TRUE, ...)

## S3 method for class 'sessionInfo'
ascii(x, locale = TRUE, ...)
```
## S3 method for class 'survfit'

```r
class = 1, print.rmean = getOption("survfit.print.rmean"), rmean = getOption("survfit.rmean")
```

```r
ascii(x, ...)  
```

### Arguments

- **x**: An R object of class found among methods(ascii). If `x` is a list, it should be a list of character strings (it will produce a bulleted list output by default).

- **include.rownames**: logical. If TRUE the rows names are printed. Default value depends on class of `x`.

- **include.colnames**: logical. If TRUE the columns names are printed. Default value depends on class of `x`.

- **rownames**: Character vector (replicated or truncated as necessary) indicating rownames of the corresponding rows. If NULL (default) the row names are not modified.

- **colnames**: Character vector (replicated or truncated as necessary) indicating colnames of the corresponding columns. If NULL (default) the column names are not modified.

- **format**: Character vector or matrix indicating the format for the corresponding columns. These values are passed to the `formatC` function. Use "d" (for integers), "f", "e", "E", "g", "G", "fg" (for reals), or "s" (for strings). "f" gives numbers in the usual xxx.xxx format; "e" and "E" give n.ddde+nn or n.dddE+nn (scientific format); "g" and "G" put x[i] into scientific format only if it saves space to do so. "fg" uses fixed format as "f", but digits as number of significant digits. Note that this can lead to quite long result strings. Finally, "nice" is like "f", but with 0 digits if `x` is an integer. Default depends on the class of `x`.

- **digits**: Numeric vector of length equal to the number of columns of the resulting table (otherwise it will be replicated or truncated as necessary) indicating the number of digits to display in the corresponding columns. Default is 2.

- **decimal.mark**: The character to be used to indicate the numeric decimal point. Default is ".".

- **na.print**: The character string specifying how NA should be formatted specially. Default is "".

- **caption**: Character vector of length 1 containing the table's caption or title. Set to "" to suppress the caption. Default value is NULL.

- **caption.level**: Character or numeric vector of length 1 containing the caption's level. Can take the following values: 0 to 5, "." (block titles in asciidoc markup), "s" (strong), "e" (emphasis), "m" (monospaced) or "" (no markup). Default is NULL.

- **width**: Numeric vector of length one containing the table width relative to the available width (expressed as a percentage value, 1...99). Default is 0 (all available width).

- **frame**: Character vector of length one. Defines the table border, and can take the following values: "topbot" (top and bottom), "all" (all sides), "none" and "sides" (left and right). The default value is NULL.

- **grid**: Character vector of length one. Defines which ruler lines are drawn between table rows and columns, and can take the following values: "all", "rows", "cols" and "none". Default is NULL.
valign
Vector or matrix indicating vertical alignment of all cells in table. Can take the following values: "top", "bottom" and "middle". Default is "".

header
logical or numeric. If TRUE or 1, 2, ..., the first line(s) of the table is (are) emphasized. The default value depends of class of x.

footer
logical or numeric. If TRUE or 1, the last line(s) of the table is (are) emphasized. The default value depends of class of x.

align
Vector or matrix indicating the alignment of the corresponding columns. Can be composed with "r" (right), "l" (left) and "c" (center). Default value is NULL.

col.width
Numeric vector of length equal to the number of columns of the resulting table (otherwise it will be replicated or truncated as necessary) indicating width of the corresponding columns (integer proportional values). Default is 1.

style
Character vector or matrix indicating the style of the corresponding columns. Can be composed with "d" (default), "s" (strong), "e" (emphasis), "m" (monospaced), "h" (header) "a" (cells can contain any of the AsciiDoc elements that are allowed inside document), "l" (literal), "v" (verse; all line breaks are retained). Default is NULL.

tgroup
Character vector or a list of character vectors defining major top column headings. The default is to have none (NULL).

n.tgroup
A numeric vector or a list of numeric vectors containing the number of columns for which each element in tgroup is a heading. For example, specify tgroup=c("Major 1", "Major 2"), n.tgroup=c(3,3) if "Major 1" is to span columns 1-3 and "Major 2" is to span columns 4-6.

talign
Character vector of length one defining alignment of major top column headings.

tvalign
Character vector of length one defining vertical alignment of major top column headings.

tstyle
Character vector of length one indicating the style of major top column headings

bgroup
Character vector or list of character vectors defining major bottom column headings. The default is to have none (NULL).

n.bgroup
A numeric vector containing the number of columns for which each element in bgroup is a heading.

balign
Character vector of length one defining alignment of major bottom column headings.

bvalign
Character vector of length one defining vertical alignment of major bottom column headings.

bstyle
Character vector of length one indicating the style of major bottom column headings

lgroup
Character vector or list of character vectors defining major left row headings. The default is to have none (NULL).

n.lgroup
A numeric vector containing the number of rows for which each element in lgroup is a heading. Column names count in the row numbers if include.colnames = TRUE.

lalign
Character vector of length one defining alignment of major left row headings.

lvalign
Character vector of length one defining vertical alignment of major left row headings.
ASCII Documentation

**ASCII Default**

- **lstyle**: Character vector of length one indicating the style of major left row headings.
- **rgroup**: Character vector or list of character vectors defining major right row headings. The default is to have none (NULL).
- **n.rgroup**: A numeric vector containing the number of rows for which each element in rgroup is a heading. Column names count in the row numbers if `include.colnames = TRUE`.
- **ralign**: Character vector of length one defining alignment of major right row headings.
- **rvalign**: Character vector of length one defining vertical alignment of major right row headings.
- **rstyle**: Character vector of length one indicating the style of major right row headings.
- **list.type**: Character vector of length one indicating the list type ("bullet", "number", "label" or "none"). If "label", `names(list)` is used for labels. Default is "bullet".
- **...**: Additional arguments. (Currently ignored.)
- **condense**: default is TRUE to condense the output with regard to the 5 lowest and highest values and the frequency table (`describe()` in package `Hmisc`).
- **locale**: show locale information?
- **scale**: A numeric value to rescale the survival time, e.g., if the input data to `survfit()` were in days, `scale=365` would scale the printout to years (see `print.survfit()` in package `survival`).
- **print.rmean**: Option for computation and display of the restricted mean (see `print.survfit()` in package `survival`).
- **rmean**: Option for computation and display of the restricted mean (see `print.survfit()` in package `survival`).

**Details**

The nature of the generated output depends on the class of `x`. For example, `summary.table` objects produce a bulleted list while `data.frame` objects produce a table of the entire `data.frame`.

Sometimes, arguments are not active, depending of the features implemented in the markup language generated. All arguments are active when asciidoc syntax is produced.

The available method functions for `ascii` are given by `methods(ascii)`. Users can extend the list of available classes by writing methods for the generic function `ascii`. All method functions should return an object of class "ascii".

**Value**

This function returns an object of class "asciiTable", "asciiList" or "asciiMixed".

**Author(s)**

David Hajage <dhajage@gmail.com>
Examples

data(esoph)
ascii(esoph[1:10,])
tab <- table(esoph$agegp, esoph$alcgp)
ascii(tab)
print(ascii(tab), type = "t2t")
print(ascii(tab), type = "rest")
print(ascii(tab), type = "org")
ascii(summary(tab))

---

**asciiBind**

*ascii table generator*

---

Description

ascii table generator

Author(s)

David Hajage

---

**Asciidoc**

*Sweave wrappers*

---

Description

Sweave wrappers

Usage

Asciidoc(file, driver = RweaveAsciidoc, syntax = SweaveSyntaxNoweb, encoding = "", ...)

T2t(file, driver = RweaveT2t, syntax = SweaveSyntaxNoweb, encoding = "", ...)

ReST(file, driver = RweaveReST, syntax = SweaveSyntaxNoweb, encoding = "", ...)

Org(file, driver = RweaveOrg, syntax = SweaveSyntaxNoweb, encoding = "", ...)

Textile(file, driver = RweaveTextile, syntax = SweaveSyntaxNoweb, encoding = "", ...)

Pandoc(file, driver = RweavePandoc, syntax = SweaveSyntaxNoweb, encoding = "", ...)
*asciiList*

**Arguments**

- **file**: Name of Sweave source file.
- **driver**: Sweave driver
- **syntax**: Sweave syntax
- **encoding**: Encoding
- **...**: Further arguments passed to the driver’s setup function.

**Author(s)**

David Hajage <dhajage@gmail.com>

**See Also**

*Sweave*

---

**asciilist**: *ascii list generator*

**Description**

ascii list generator

**Author(s)**

David Hajage

---

**asciimixed**: *ascii mixed generator*

**Description**

ascii mixed generator

**Author(s)**

David Hajage


**asciiTable**

*ascii table generator*

Description

ascii table generator

Author(s)

David Hajage

---

**cbind.ascii**

*Cbind two ascii objects*

Description

Cbind two ascii objects

Usage

cbind.ascii(..., caption = NULL, caption.level = NULL, frame = NULL, grid = NULL, col.width = 1, width = 1)

Arguments

- ...: ascii objects
- caption: see ?ascii
- caption.level: see ?ascii
- frame: see ?ascii
- grid: see ?ascii
- col.width: see ?ascii
- width: see ?ascii

Details

This function binds cols of two ascii table.

Value

An "asciiCbind" object.

Author(s)

David Hajage
**convert**

*Convert a file with specified backend*

**Description**

Convert a file with specified backend

**Usage**

```r
convert(i, d = NULL, f = NULL, e = NULL, o = NULL, backend = getOption("asciidoc"), cygwin = FALSE, open = FALSE)
```

**Arguments**

- `i` input file
- `d` output directory
- `f` format
- `e` encoding
- `o` other options
- `backend` backend ("asciidoc", "t2t" or "pandoc")
- `cygwin` use cygwin?
- `open` open resulting file?

**Details**

This function convert a file with asciidoc, txt2tags or pandoc backend

**Value**

Nothing

**Author(s)**

David Hajage
creatoreport  

**Report creation**

**Description**

Produce a report
Report generator

**Usage**

```r
createreport(..., list = NULL, file = NULL, format = NULL, open = TRUE, backend = getOption("asciidBack", TRUE), encoding = NULL, options = NULL, cygwin = FALSE, title = NULL, author = NULL, email = NULL, date = NULL)
```

**Arguments**

- `...`: R objects (not used if "list" is not NULL)
- `list`: list of R objects
- `file`: name of the output file (without extension)
- `format`: format of the output file
- `open`: open resulting file?
- `backend`: backend
- `encoding`: encoding
- `options`: other options
- `cygwin`: use cygwin?
- `title`: title of the report
- `author`: author of the report
- `email`: email of the author
- `date`: date

**Details**

Produce a report from a list of R objects. This function can be used directly, or through a `Report` object (see examples). `Report$new()` creates a new object, `Report$create()` produce a report. Exportation options can be specified with `Report$option <- option` or directly in `Report$create(nameoftheoption = option)`. 

Special objects can be used to create sections (see `?section`), paragraphs (see `?paragraph`), verbatim environment (see `?verbatim`), and to insert figures (see `?fig`) or inline results (see `?sexpr`). Helpers exist: `Report$addSection()`, `Report$addParagraph()`, `Report$addVerbatim()`, `Report$addFig()`. 

It needs a working installation of asciidoc, a2x tool chain, txt2tags, pandoc and/or markdown2pdf.

**Value**

Nothing
Author(s)
David Hajage
David Hajage

Examples

```r
## Not run:
options(asciiType = "asciidoc")
createreport(head(esoph))

r <- Report$new(author = "David Hajage", email = "dhajage at gmail dot com")
r$add(section("First section", 2))
r$add(paragraph("The data set has", nrow(esoph), " lines. See yourself:"), esoph)
r$addSection("Second subsection: age and alc group", 2)

tab <- with(esoph, table(alcgp, agegp))
r$add(ascii(tab), ascii(summary(tab), format = "nice"))
r$create()
r$format <- "slidy"
r$createt()

r$title <- "R report example"
r$author <- "David Hajage"
r$email <- "dhajage at gmail dot com"
options(asciiType = "pandoc")
r$backend <- "pandoc"
r$format <- "odt"
r$create()

r$create(backend = "markdown2pdf", format = "pdf")

## End(Not run)
```

---

**fig**

*Insert figure*

Description

graph can be used with export function to insert an R graphic.

Usage

```r
fig(file = NULL, graph = NULL, format = NULL, ...)
```

Arguments

- **file** character string (a recordedplot, a lattice plot, a ggplot, or an expression producing a plot (optional if the file already exists))
format: jpg, png or pdf (or guessed with the file name)

... additional arguments (passed to format options)

Value

A fig object

Author(s)

David Hajage

Description

out can be used with export function to insert an R results

Usage

out(x, results = "verbatim")

Arguments

x: an R object
results: if 'verbatim', the output is included in a verbatim environment. If 'ascii', the output is taken to be already proper markup and included as is.

Value

An out object

Author(s)

David Hajage
paragraph

Create a paragraph

Description
paragraph can be used with export function to add...a paragraph

Usage
paragraph(..., new = TRUE)

Arguments
... strings composing the paragraph
new whether to create a new paragraph or to continue a preceding one

Value
A paragraph object.

Author(s)
David Hajage

plim format p values

Description
format p values

Usage
plim(p, digits = 4)

Arguments
p p values
digits number of digits

Value
formatted p values

Author(s)
David Hajage
print

Print ascii object

Description

Function displaying the asciidoc, txt2tags, reStructuredText, org or textile code associated with the supplied object of class ascii.

Show method for ascii objects

Arguments

- **x**
  - An object of class "asciiTable", "asciilist", "asciimixed", "asciicbind" or "Report".

- **type**
  - Type of syntax produce. Possible values for type are "asciidoc", "t2t", "rest", "org", "textile" or "pandoc". Default value produce asciidoc syntax.

- **file**
  - A character string naming the file to print to. Default is NULL (print to the console).

- **append**
  - If TRUE, code will be appended to file instead of overwriting it. Default value is FALSE.

- **escape**
  - If TRUE, characters in list.escape will be printed with a \\. Default value is FALSE.

- **list.escape**
  - Character vector. Default value is c("\_", "\^")

- **help**
  - logical print help? (objects of class "Report")

- **...**
  - Additional arguments. (Currently ignored.)

object

ascii or Report object

Details

The package provides the new global option asciitype. Default value is "asciidoc" (see examples).

Author(s)

David Hajage <dhajage@gmail.com>

See Also

ascii
print.fig

**Examples**

data(esoph)
ascii(esoph[1:10,])
print(ascii(esoph[1:10,]), type = "t2t")
print(ascii(esoph[1:10,]), type = "rest")
print(ascii(esoph[1:10,]), type = "org")
print(ascii(esoph[1:10,]), type = "textile")
print(ascii(esoph[1:10,]), type = "pandoc")
options(asciiType = "rest")
ascii(esoph[1:10,])
options(asciiType = "asciidoc")

---

**Description**

Print an graph object

**Usage**

```r
## S3 method for class 'fig'
print(x, backend = getOption("asciiBackend"), ...)
```

**Arguments**

- `x`: an graph object
- `backend`: ascii backend
- `...`: not used

**Author(s)**

David Hajage

---

print.out

**Description**

Print an out object

**Usage**

```r
## S3 method for class 'out'
print(x, backend = getOption("asciiBackend"), ...)
```
Arguments

x an out object
backend ascii backend
... not used

Author(s)

David Hajage

print.paragraph

Print a paragraph object

Description

Print a paragraph object

Usage

## S3 method for class 'paragraph'
print(x, ...)

Arguments

x a paragraph object
... not used

Author(s)

David Hajage

print[section]

Print a section object

Description

Print a section object

Usage

## S3 method for class 'section'
print(x, backend = getOption("asciiBackend"), ...)

print.section
print.sexpr

Arguments

x a section object
backend ascii backend
... not used

Author(s)

David Hajage

print.sexpr  Print a sexpr object

Description

Print a sexpr object

Usage

## S3 method for class 'sexpr'
print(x, ...)

Arguments

x a sexpr object
... not used

Author(s)

David Hajage

print.verbatim  Print a verbatim object

Description

Print a verbatim object

Usage

## S3 method for class 'verbatim'
print(x, backend = getOption("asciiBackend"), ...)

print.verbatim  Print a verbatim object

Description

Print a verbatim object

Usage

## S3 method for class 'verbatim'
print(x, backend = getOption("asciiBackend"), ...)
Arguments

- `x`: a verbatim object
- `backend`: ascii backend
- ... not used

Author(s)

David Hajage

---

RtangleAscii

Description

RtangleAscii

---

section

Description

section can be used with export function to add...a section

Usage

`section(caption, caption.level = 1)`

Arguments

- `caption`: a string
- `caption.level`: caption level

Value

A section object.

Author(s)

David Hajage
**sexpr**

*Insert an inline R result*

**Description**

sexpr can be used with export function to insert an inline R results

**Usage**

```r
sexpr(x)
```

**Arguments**

- `x` an R results (of length one)

**Value**

A sexpr object.

**Author(s)**

David Hajage

---

**verbatim**

*Create a verbatim paragraph*

**Description**

verbatim can be used with export function to add a verbatim paragraph

**Usage**

```r
verbatim(...)```

**Arguments**

- `...` strings composing the paragraph (line by line)

**Value**

A verbatim object.

**Author(s)**

David Hajage
Index

*Topic IO
  Asciidoc, 6
*Topic file
  Asciidoc, 6
*Topic print
  ascii.default, 2
  print, 14
  ascii, 14
  ascii.default, 2
  asciiCbind, 6
  Asciidoc, 6
  asciiList, 7
  asciiMixed, 7
  asciiTable, 8
  cbind.ascii, 8
  convert, 9
  createreport, 10
  fig, 11
  graph(fig), 11
  Org(Asciidoc), 6
  out, 12
  Pandoc(Asciidoc), 6
  paragraph, 13
  plim, 13
  print, 14
  print.asciiCbind-method(print), 14
  print.asciiList-method(print), 14
  print.asciiMixed-method(print), 14
  print.asciiTable-method(print), 14
  print,Report-method(print), 14
  print.fig, 15
  print.out, 15
  print.paragraph, 16
  print.section, 16
  print,sexpr, 17
  print,verbatim, 17
  Report(createreport), 10
  ReST(Asciidoc), 6
  RangleAscii, 18
  section, 18
  sexpr, 19
  show.asciiCbind-method(print), 14
  show.asciiList-method(print), 14
  show.asciiMixed-method(print), 14
  show.asciiTable-method(print), 14
  show,Report-method(print), 14
  Sweave, 7
  T2t(Asciidoc), 6
  Textile(Asciidoc), 6
  verbatim, 19