

Package ‘ascii’

September 17, 2020

Maintainer Mark Clements <mark.clements@ki.se>

License GPL (>= 2)

Title Export R Objects to Several Markup Languages

Type Package

Description Coerce R object to 'asciidoc', 'txt2tags',
'restructuredText', 'org', 'textile' or 'pandoc' syntax.
Package comes with a set of drivers for 'Sweave'.

Version 2.4

URL <https://github.com/mclements/ascii>

BugReports <https://github.com/mclements/ascii/issues>

Date 2020-08-18

Depends R (>= 2.13), methods

Imports utils, digest, codetools, survival, stats, grDevices

Suggests Hmisc, xtable, R2HTML, knitr

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' 'bind.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'
'cache_expr.R' 'weaver.R' 'unexported.R'

RoxygenNote 7.0.2

NeedsCompilation no

Author David Hajage [aut],
Mark Clements [cre, ctb],
Seth Falcon [ctb],

Terry Therneau [ctb],
 Matti Pastell [ctb],
 Friedrich Leisch [ctb]

Repository CRAN

Date/Publication 2020-09-17 13:10:09 UTC

R topics documented:

ascii.anova	2
asciiCbind-class	23
AsciiDoc	23
asciiList-class	24
asciiMixed-class	25
asciiTable-class	26
cbind.ascii	26
convert	27
createreport	28
fig	30
out	31
paragraph	31
plim	32
print,asciiCbind-method	32
print.fig	35
print.out	35
print.paragraph	36
print.section	36
print.sexpr	37
print.verbatim	37
RtangleAscii	38
section	38
sexpr	39
verbatim	39
Index	40

ascii.anova

Export R objects to several markup languages

Description

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

Usage

```
## S3 method for class 'anova'
ascii(
  x,
  include.rownames = TRUE,
  include.colnames = TRUE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = TRUE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
  lgroup = NULL,
  n.lgroup = NULL,
  lalign = "c",
  lvalign = "middle",
  lstyle = "h",
  rgroup = NULL,
  n.rgroup = NULL,
  ralign = "c",
  rvalign = "middle",
  rstyle = "h",
  ...
)

## S3 method for class 'data.frame'
ascii(
```

```
x,  
include.rownames = TRUE,  
include.colnames = TRUE,  
rownames = NULL,  
colnames = NULL,  
format = "f",  
digits = 2,  
decimal.mark = ".",  
na.print = "",  
caption = NULL,  
caption.level = NULL,  
width = 0,  
frame = NULL,  
grid = NULL,  
valign = NULL,  
header = TRUE,  
footer = FALSE,  
align = NULL,  
col.width = 1,  
style = NULL,  
tgroup = NULL,  
n.tgroup = NULL,  
talign = "c",  
tvalign = "middle",  
tstyle = "h",  
bgroup = NULL,  
n.bgroup = NULL,  
balign = "c",  
bvalign = "middle",  
bstyle = "h",  
lgroup = NULL,  
n.lgroup = NULL,  
lalign = "c",  
lvalign = "middle",  
lstyle = "h",  
rgroup = NULL,  
n.rgroup = NULL,  
ralign = "c",  
rvalign = "middle",  
rstyle = "h",  
...  
)  
  
## Default S3 method:  
ascii(  
  x,  
  include.rownames = TRUE,  
  include.colnames = TRUE,
```

```
    rownames = NULL,
    colnames = NULL,
    format = "f",
    digits = 2,
    decimal.mark = ".",
    na.print = "",
    caption = NULL,
    caption.level = NULL,
    width = 0,
    frame = NULL,
    grid = NULL,
    valign = NULL,
    header = TRUE,
    footer = FALSE,
    align = NULL,
    col.width = 1,
    style = NULL,
    tgroup = NULL,
    n.tgroup = NULL,
    talign = "c",
    tvalign = "middle",
    tstyle = "h",
    bgroup = NULL,
    n.bgroup = NULL,
    balign = "c",
    bvalign = "middle",
    bstyle = "h",
    lgroup = NULL,
    n.lgroup = NULL,
    lalign = "c",
    lvalign = "middle",
    lstyle = "h",
    rgroup = NULL,
    n.rgroup = NULL,
    ralign = "c",
    rvalign = "middle",
    rstyle = "h",
    list.type = "bullet",
    ...
)

## S3 method for class 'glm'
ascii(
  x,
  include.rownames = TRUE,
  include.colnames = TRUE,
  rownames = NULL,
  colnames = NULL,
```

```
format = "f",
digits = 2,
decimal.mark = ".",
na.print = "",
caption = NULL,
caption.level = NULL,
width = 0,
frame = NULL,
grid = NULL,
valign = NULL,
header = TRUE,
footer = FALSE,
align = NULL,
col.width = 1,
style = NULL,
tgroup = NULL,
n.tgroup = NULL,
talign = "c",
tvalign = "middle",
tstyle = "h",
bgroup = NULL,
n.bgroup = NULL,
balign = "c",
bvalign = "middle",
bstyle = "h",
lgroup = NULL,
n.lgroup = NULL,
lalign = "c",
lvalign = "middle",
lstyle = "h",
rgroup = NULL,
n.rgroup = NULL,
ralign = "c",
rvalign = "middle",
rstyle = "h",
...
)

## S3 method for class 'summary.glm'
ascii(
  x,
  include.rownames = TRUE,
  include.colnames = TRUE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
```

```

na.print = "",
caption = NULL,
caption.level = NULL,
width = 0,
frame = NULL,
grid = NULL,
valign = NULL,
header = TRUE,
footer = FALSE,
align = NULL,
col.width = 1,
style = NULL,
tgroup = NULL,
n.tgroup = NULL,
talign = "c",
tvalign = "middle",
tstyle = "h",
bgroup = NULL,
n.bgroup = NULL,
balign = "c",
bvalign = "middle",
bstyle = "h",
lgroup = NULL,
n.lgroup = NULL,
lalign = "c",
lvalign = "middle",
lstyle = "h",
rgroup = NULL,
n.rgroup = NULL,
ralign = "c",
rvalign = "middle",
rstyle = "h",
...
)

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

## S3 method for class 'summary.formula.response'
ascii(
  x,
  vnames = c("labels", "names"),
  prUnits = TRUE,
  lgroup = list(dimnames(stats)[[1]], if (ul) vlabels else at$vname[at$vname != ""]),
  n.lgroup = list(1, at$nlevels),
  include.rownames = FALSE,
  include.colnames = TRUE,
  format = "nice",

```

```

caption = paste(at$ylabel, if (ns > 1) paste(" by", if (ul) at$strat.label else
  at$strat.name), " N = ", at$n, if (at$nmiss) paste(", ", at$nmiss, " Missing", sep =
  """), sep = """),
caption.level = "s",
header = TRUE,
...
)

```

```
## S3 method for class 'summary.formula.reverse'
```

```

ascii(
  x,
  digits,
  prn = any(n != N),
  pctdig = 0,
  npct = c("numerator", "both", "denominator", "none"),
  exclude1 = TRUE,
  vnames = c("labels", "names"),
  prUnits = TRUE,
  sep = "/",
  formatArgs = NULL,
  round = NULL,
  prtest = c("P", "stat", "df", "name"),
  prmsd = FALSE,
  pdig = 3,
  eps = 0.001,
  caption = paste("Descriptive Statistics", if (length(x$group.label)) paste(" by",
    x$group.label) else paste(" (N = ", x$N, ")"), sep = """), sep = """),
  caption.level = "s",
  include.rownames = FALSE,
  include.colnames = TRUE,
  colnames = gl,
  header = TRUE,
  lgroup = lgr,
  n.lgroup = n.lgr,
  rgroup = rgr,
  n.rgroup = n.rgr,
  rstyle = "d",
  ...
)

```

```
## S3 method for class 'summary.formula.cross'
```

```

ascii(
  x,
  twoway = nvar == 2,
  prnmiss = any(stats$Missing > 0),
  prn = TRUE,
  formatArgs = NULL,
  caption = a$heading,

```



```
caption.level = "s",
include.rownames = FALSE,
include.colnames = TRUE,
header = TRUE,
format = "nice",
lggroup = v,
n.lgroup = rep(length(z), length(v)),
...
)
```

```
## S3 method for class 'htest'
```

```
ascii(
  x,
  include.rownames = TRUE,
  include.colnames = TRUE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = TRUE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
  lgroup = NULL,
  n.lgroup = NULL,
  lalign = "c",
  lvalign = "middle",
  lstyle = "h",
  rgroup = NULL,
```

```
n.rgroup = NULL,  
ralign = "c",  
rvalign = "middle",  
rstyle = "h",  
...  
)  
  
## S3 method for class 'list'  
ascii(x, caption = NULL, caption.level = NULL, list.type = "bullet", ...)  
  
## S3 method for class 'packageDescription'  
ascii(x, caption = NULL, caption.level = NULL, list.type = "label", ...)  
  
## S3 method for class 'sessionInfo'  
ascii(x, locale = TRUE, ...)  
  
## S3 method for class 'lm'  
ascii(  
  x,  
  include.rownames = TRUE,  
  include.colnames = TRUE,  
  rownames = NULL,  
  colnames = NULL,  
  format = "f",  
  digits = 2,  
  decimal.mark = ".",  
  na.print = "",  
  caption = NULL,  
  caption.level = NULL,  
  width = 0,  
  frame = NULL,  
  grid = NULL,  
  valign = NULL,  
  header = TRUE,  
  footer = FALSE,  
  align = NULL,  
  col.width = 1,  
  style = NULL,  
  tgroup = NULL,  
  n.tgroup = NULL,  
  talign = "c",  
  tvalign = "middle",  
  tstyle = "h",  
  bgroup = NULL,  
  n.bgroup = NULL,  
  balign = "c",  
  bvalign = "middle",  
  bstyle = "h",
```

```
    lgroup = NULL,
    n.lgroup = NULL,
    lalign = "c",
    lvalign = "middle",
    lstyle = "h",
    rgroup = NULL,
    n.rgroup = NULL,
    ralign = "c",
    rvalign = "middle",
    rstyle = "h",
    ...
)

## S3 method for class 'summary.lm'
ascii(
  x,
  include.rownames = TRUE,
  include.colnames = TRUE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = TRUE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
  lgroup = NULL,
  n.lgroup = NULL,
  lalign = "c",
```

```
    lvalign = "middle",
    lstyle = "h",
    rgroup = NULL,
    n.rgroup = NULL,
    ralign = "c",
    rvalign = "middle",
    rstyle = "h",
    ...
)

## S3 method for class 'matrix'
ascii(
  x,
  include.rownames = FALSE,
  include.colnames = FALSE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = FALSE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
  lgroup = NULL,
  n.lgroup = NULL,
  lalign = "c",
  lvalign = "middle",
  lstyle = "h",
  rgroup = NULL,
```

```
    n.rgroup = NULL,
    ralign = "c",
    rvalign = "middle",
    rstyle = "h",
    ...
)

## S3 method for class 'survfit'
ascii(
  x,
  scale = 1,
  print.rmean = getOption("survfit.print.rmean"),
  rmean = getOption("survfit.rmean"),
  include.rownames = TRUE,
  include.colnames = TRUE,
  header = TRUE,
  ...
)

## S3 method for class 'table'
ascii(
  x,
  include.rownames = TRUE,
  include.colnames = TRUE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = TRUE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
```

```
    balign = "c",
    bvalign = "middle",
    bstyle = "h",
    lgroup = NULL,
    n.lgroup = NULL,
    lalign = "c",
    lvalign = "middle",
    lstyle = "h",
    rgroup = NULL,
    n.rgroup = NULL,
    ralign = "c",
    rvalign = "middle",
    rstyle = "h",
    ...
)

## S3 method for class 'integer'
ascii(
  x,
  include.rownames = FALSE,
  include.colnames = FALSE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = FALSE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
```

```
    lgroup = NULL,
    n.lgroup = NULL,
    lalign = "c",
    lvalign = "middle",
    lstyle = "h",
    rgroup = NULL,
    n.rgroup = NULL,
    ralign = "c",
    rvalign = "middle",
    rstyle = "h",
    ...
)

## S3 method for class 'numeric'
ascii(
  x,
  include.rownames = FALSE,
  include.colnames = FALSE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = FALSE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
  lgroup = NULL,
  n.lgroup = NULL,
  lalign = "c",
```

```
    lvalign = "middle",
    lstyle = "h",
    rgroup = NULL,
    n.rgroup = NULL,
    ralign = "c",
    rvalign = "middle",
    rstyle = "h",
    ...
)

## S3 method for class 'character'
ascii(
  x,
  include.rownames = FALSE,
  include.colnames = FALSE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = FALSE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
  lgroup = NULL,
  n.lgroup = NULL,
  lalign = "c",
  lvalign = "middle",
  lstyle = "h",
  rgroup = NULL,
```



```
n.rgroup = NULL,
ralign = "c",
rvalign = "middle",
rstyle = "h",
...
)

## S3 method for class 'factor'
ascii(
  x,
  include.rownames = FALSE,
  include.colnames = FALSE,
  rownames = NULL,
  colnames = NULL,
  format = "f",
  digits = 2,
  decimal.mark = ".",
  na.print = "",
  caption = NULL,
  caption.level = NULL,
  width = 0,
  frame = NULL,
  grid = NULL,
  valign = NULL,
  header = FALSE,
  footer = FALSE,
  align = NULL,
  col.width = 1,
  style = NULL,
  tgroup = NULL,
  n.tgroup = NULL,
  talign = "c",
  tvalign = "middle",
  tstyle = "h",
  bgroup = NULL,
  n.bgroup = NULL,
  balign = "c",
  bvalign = "middle",
  bstyle = "h",
  lgroup = NULL,
  n.lgroup = NULL,
  lalign = "c",
  lvalign = "middle",
  lstyle = "h",
  rgroup = NULL,
  n.rgroup = NULL,
  ralign = "c",
  rvalign = "middle",
```

```

  rstyle = "h",
  ...
)

## S3 method for class 'proc_time'
ascii(x, include.rownames = FALSE, include.colnames = TRUE, ...)

ascii(x, ...)

```

Arguments

<code>x</code>	An R object of class found among <code>methods(ascii)</code> . If <code>x</code> is a list, it should be a list of character strings (it will produce a bulleted list output by default).
<code>include.rownames</code>	logical. If TRUE the rows names are printed. Default value depends of class of <code>x</code> .
<code>include.colnames</code>	logical. If TRUE the columns names are printed. Default value depends of class of <code>x</code> .
<code>rownames</code>	Character vector (replicated or truncated as necessary) indicating rownames of the corresponding rows. If NULL (default) the row names are not modified
<code>colnames</code>	Character vector (replicated or truncated as necessary) indicating colnames of the corresponding columns. If NULL (default) the column names are not modified
<code>format</code>	Character vector or matrix indicating the format for the corresponding columns. These values are passed to the <code>formatC</code> 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 <code>x[i]</code> into scientific format only if it saves space to do so. "fg" uses fixed format as "f", but <code>digits</code> as number of <i>significant</i> digits. Note that this can lead to quite long result strings. Finally, "nice" is like "f", but with 0 digits if <code>x</code> is an integer. Default depends on the class of <code>x</code> .
<code>digits</code>	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.
<code>decimal.mark</code>	The character to be used to indicate the numeric decimal point. Default is ".".
<code>na.print</code>	The character string specifying how NA should be formatted specially. Default is "".
<code>caption</code>	Character vector of length 1 containing the table's caption or title. Set to "" to suppress the caption. Default value is NULL.
<code>caption.level</code>	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.
<code>width</code>	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.
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
...	Additional arguments. (Currently ignored.)
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".
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).
vnames	By default, tables and plots are usually labeled with variable labels (see summary.formula in package Hmisc).
prUnits	set to FALSE to suppress printing or latexing units attributes of variables (see summary.formula in package Hmisc).
prn	set to TRUE to print the number of non-missing observations on the current (row) variable (see summary.formula in package Hmisc).
pctdig	number of digits to the right of the decimal place for printing percentages (see summary.formula in package Hmisc).
npct	specifies which counts are to be printed to the right of percentages (see summary.formula in package Hmisc).
exclude1	by default, method="reverse" objects will be printed, plotted, or typeset by removing redundant entries from percentage tables for categorical variables (see summary.formula in package Hmisc).
sep	character to use to separate quantiles when printing method="reverse" tables (see summary.formula in package Hmisc).
formatArgs	a list containing other arguments to pass to format.default (see summary.formula in package Hmisc).
round	Specify round to round the quantiles and optional mean and standard deviation to round digits after the decimal point (see summary.formula in package Hmisc).

<code>prtest</code>	a vector of test statistic components to print if <code>test=TRUE</code> (see <code>summary.formula</code> in package <code>Hmisc</code>).
<code>prmsd</code>	set to <code>TRUE</code> to print mean and SD after the three quantiles, for continuous variables (see <code>summary.formula</code> in package <code>Hmisc</code>).
<code>pdig</code>	number of digits to the right of the decimal place for printing P-values. (see <code>summary.formula</code> in package <code>Hmisc</code>).
<code>eps</code>	P-values less than <code>eps</code> will be printed as <code>< eps</code> (see <code>summary.formula</code> in package <code>Hmisc</code>).
<code>twoway</code>	controls whether the resulting table will be printed in enumeration format or as a two-way table (the default) (see <code>summary.formula</code> in package <code>Hmisc</code>).
<code>prnmiss</code>	set to <code>FALSE</code> to suppress printing counts of missing values
<code>locale</code>	show locale information?
<code>scale</code>	A numeric value to rescale the survival time, e.g., if the input data to <code>survfit</code> were in days, <code>scale=365</code> would scale the printout to years (see <code>print.survfit()</code> in package <code>survival</code>).
<code>print.rmean</code>	Option for computation and display of the restricted mean (see <code>print.survfit()</code> in package <code>survival</code>).
<code>rmean</code>	Option for computation and display of the restricted mean (see <code>print.survfit()</code> in package <code>survival</code>).

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

```
op <- options(asciiType = "org")
local({x <- 1:10; y <- rnorm(length(x),1+x); ascii(anova(lm(y~x)))})
options(op)
op <- options(asciiType = "org")
ascii(data.frame(a = 1:3, b = 2), include.rownames = FALSE, digits = 0)
options(op)
```

```

op <- options(asciiType = "org")
local({x <- 1:10; y <- rnorm(length(x), 1+x); ascii(glm(y~x)) })
options(op)
op <- options(asciiType = "org")
local({x <- 1:10; y <- rnorm(length(x), 1+x); ascii(summary(glm(y~x))) })
options(op)
op <- options(asciiType = "org")
local({x <- rnorm(100); ascii(t.test(x))})
options(op)
op <- options(asciiType = "org")
ascii(list(a=1,b=2), list.type="label")
options(op)
op <- options(asciiType = "org")
ascii(sessionInfo())
options(op)
op <- options(asciiType = "org")
local({x <- 1:10; y <- rnorm(length(x), 1+x); ascii(lm(y~x)) })
options(op)
op <- options(asciiType = "org")
local({x <- 1:10; y <- rnorm(length(x), 1+x); ascii(summary(lm(y~x))) })
options(op)
op <- options(asciiType = "org")
ascii(matrix(1:4,2,2,FALSE,list(1:2,c("A","B"))), TRUE, TRUE, digits=0)
options(op)
op <- options(asciiType = "org")
ascii(table(rbinom(100,5,.3)), digits=0)
options(op)
op <- options(asciiType = "org")
ascii(c(a=1L,b=2L),FALSE,TRUE,digits=0)
options(op)
op <- options(asciiType = "org")
ascii(seq(0,1,length=11),digits=1)
options(op)
op <- options(asciiType = "org")
ascii(c(a="A",b="B"),FALSE,TRUE,header=TRUE)
options(op)
op <- options(asciiType = "org")
ascii(factor(c("A","B")),FALSE)
options(op)
op <- options(asciiType = "org")
ascii(system.time(sum(1:1e6)), header=TRUE)
options(op)
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))

```

asciiCbind-class	<i>ascii table generator</i>
------------------	------------------------------

Description

ascii table generator

Author(s)

David Hajage

Asciiidoc	<i>Sweave wrappers</i>
-----------	------------------------

Description

Sweave wrappers

Usage

```
Asciiidoc(  
  file,  
  driver = RweaveAsciiidoc,  
  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 = "",
    ...
)

```

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](#)

Examples

```

## Not run:
testfile <- system.file("examples", "Org-test-1.nw", package = "ascii")

## enforce par(ask = FALSE)
options(device.ask.default = FALSE)

## create an org file - in the current working directory, getwd():
Org(testfile)
Org(testfile, driver=weaverOrg)

## This can be edited in and exported from Org Mode

## End(Not run)

```

asciilist-class

ascii list generator

Description

ascii list generator

Methods

```
show.asciidoc( x = .self$x, caption = .self$caption, caption.level = .self$caption.level, list.type = .self$list.type )  
    print a list with asciidoc markup  
show.org( x = .self$x, caption = .self$caption, caption.level = .self$caption.level, list.type = .self$list.type )  
    print a list with org markup  
show.pandoc( x = .self$x, caption = .self$caption, caption.level = .self$caption.level, list.type = .self$list.type )  
    print a list with pandoc markup  
show.rest( x = .self$x, caption = .self$caption, caption.level = .self$caption.level, list.type = .self$list.type )  
    print a list with rest markup  
show.t2t( x = .self$x, caption = .self$caption, caption.level = .self$caption.level, list.type = .self$list.type )  
    print a list with t2t markup  
show.textile( x = .self$x, caption = .self$caption, caption.level = .self$caption.level, list.type = .self$list.type )  
    print a list with textile markup
```

Author(s)

David Hajage

asciiMixed-class	<i>ascii mixed generator</i>
------------------	------------------------------

Description

ascii mixed generator

Methods

```
show.asciidoc() print everything with asciidoc markup  
show.org() print everything with org markup  
show.pandoc() print everything with pandoc markup  
show.rest() print everything with rest markup  
show.t2t() print everything with t2t markup  
show.textile() print everything with textile markup
```

Author(s)

David Hajage

asciiTable-class	<i>ascii table generator</i>
------------------	------------------------------

Description

ascii table generator

Methods

```
show.asciidoc( x = .self$x, include.rownames = .self$include.rownames, include.colnames = .self$include.colnames )
  print a table with asciidoc markup
show.org( x = .self$x, include.rownames = .self$include.rownames, include.colnames = .self$include.colnames )
  print a table with org-mode markup
show.pandoc( x = .self$x, include.rownames = .self$include.rownames, include.colnames = .self$include.colnames )
  print a table with pandoc markup
show.rest( x = .self$x, include.rownames = .self$include.rownames, include.colnames = .self$include.colnames )
  print a table with restructuredText markup
show.t2t( x = .self$x, include.rownames = .self$include.rownames, include.colnames = .self$include.colnames )
  print a table with txt2tags markup
show.textile( x = .self$x, include.rownames = .self$include.rownames, include.colnames = .self$include.colnames )
  print a table with textile markup
```

Author(s)

David Hajage

cbind.ascii	<i>Cbind two ascii objects</i>
-------------	--------------------------------

Description

Cbind two ascii objects

Usage

```
## S3 method for class 'ascii'
cbind(
  ...,
  caption = NULL,
  caption.level = NULL,
  frame = NULL,
  grid = NULL,
  col.width = 1,
  width = 0
)
```

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	<i>Convert a file with specified backend</i>
---------	--

Description

Convert a file with specified backend

Usage

```
convert(  
  i,  
  d = NULL,  
  f = NULL,  
  e = NULL,  
  O = NULL,  
  backend = getOption("asciiBackend"),  
  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

createreport

Report creation

Description

Produce a report

Usage

```
createreport(  
  ...,  
  list = NULL,  
  file = NULL,  
  format = NULL,  
  open = TRUE,  
  backend = getOption("asciiBackend"),  
  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$nameoftheoption <- option` or directly in `Report$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 and/or pandoc (NB: `mark-down2pdf` uses pandoc with latex).

Value

Nothing

Author(s)

David Hajage

David Hajage

Examples

```
## Not run:
op <- options(asciiType = "asciidoc")
createreport(head(esoph))

r <- Report$new(author = "David Hajage", email = "dhajage at gmail dot com")
r$add(section("First section"))
r$addSection("First subsection", 2)
```

```

r$add(paragraph("The data set has", sexpr(nrow(esoph)), " lines. See yourself:"), esoph)
r$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$format <- "slidy"
r$create()

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")
options(op)

## End(Not run)

```

 fig

Insert figure

Description

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

Usage

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

Arguments

file	character string (
graph	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

out *Export R objects*

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

<code>p</code>	p values
<code>digits</code>	number of digits

Value

formatted p values

Author(s)

David Hajage

`print,asciiCbind-method`
Print ascii object

Description

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

Usage

```
## S4 method for signature 'asciiCbind'
print(
  x,
  type = getOption("asciiType"),
  file = NULL,
  append = FALSE,
  escape = FALSE,
  list.escape = c("\\_", "\\^"),
  ...
)
```



```
## S4 method for signature 'asciiCbind'
show(object)

## S4 method for signature 'asciiTable'
print(
  x,
  type = getOption("asciiType"),
  file = NULL,
  append = FALSE,
  escape = FALSE,
  list.escape = c("\_", "\^"),
  ...
)

## S4 method for signature 'asciiTable'
show(object)

## S4 method for signature 'asciiList'
print(
  x,
  type = getOption("asciiType"),
  file = NULL,
  append = FALSE,
  escape = FALSE,
  list.escape = c("\_", "\^"),
  ...
)

## S4 method for signature 'asciiList'
show(object)

## S4 method for signature 'asciiMixed'
print(
  x,
  type = getOption("asciiType"),
  file = NULL,
  append = FALSE,
  escape = FALSE,
  list.escape = c("\_", "\^"),
  ...
)

## S4 method for signature 'asciiMixed'
show(object)

## S4 method for signature 'Report'
print(x, help = FALSE, ...)
```

```
## S4 method for signature 'Report'
show(object)
```

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("_", "^")
...	Additional arguments. (Currently ignored.)
object	ascii or Report object
help	logical print help? (objects of class "Report")

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](#)

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")
```

print.fig	<i>Print an graph object</i>
-----------	------------------------------

Description

Print an graph object

Usage

```
## 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	<i>Print an out object</i>
-----------	----------------------------

Description

Print an out object

Usage

```
## 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"), ...)
```

Arguments

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

Author(s)

David Hajage

print.sexpr	<i>Print a sexpr object</i>
-------------	-----------------------------

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	<i>Print a verbatim object</i>
----------------	--------------------------------

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`*RtangleAscii*

Description`RtangleAscii`**Usage**`RtangleAscii()`

`section`*Create a 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	<i>Insert an inline R result</i>
-------	----------------------------------

Description

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

Usage

```
sexpr(x)
```

Arguments

x an R results (of length one)

Value

A sexpr object.

Author(s)

David Hajage

verbatim	<i>Create a verbatim paragraph</i>
----------	------------------------------------

Description

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

Usage

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

Arguments

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

Value

A verbatim object.

Author(s)

David Hajage

Index

- * **IO**
 - Asciidoc, 23
- * **file**
 - Asciidoc, 23
- * **print**
 - ascii.anova, 2
 - print,asciiCbind-method, 32
- ascii, 34
- ascii (ascii.anova), 2
- ascii.anova, 2
- asciiCbind (asciiCbind-class), 23
- asciiCbind-class, 23
- Asciidoc, 23
- asciiList (asciiList-class), 24
- asciiList-class, 24
- asciiMixed (asciiMixed-class), 25
- asciiMixed-class, 25
- asciiTable (asciiTable-class), 26
- asciiTable-class, 26
- cbind.ascii, 26
- convert, 27
- createreport, 28
- fig, 30
- graph (fig), 30
- Org (Asciidoc), 23
- out, 31
- package-ascii (ascii.anova), 2
- Pandoc (Asciidoc), 23
- paragraph, 31
- plim, 32
- print,asciiCbind-method, 32
- print,asciiList-method
 - (print,asciiCbind-method), 32
- print,asciiMixed-method
 - (print,asciiCbind-method), 32
- print,asciiTable-method
 - (print,asciiCbind-method), 32
- print,Report-method
 - (print,asciiCbind-method), 32
- print.fig, 35
- print.out, 35
- print.paragraph, 36
- print.section, 36
- print.sexpr, 37
- print.verbatim, 37
- Report (createreport), 28
- Report-class (createreport), 28
- ReST (Asciidoc), 23
- RtangleAscii, 38
- section, 38
- sexpr, 39
- show,asciiCbind-method
 - (print,asciiCbind-method), 32
- show,asciiList-method
 - (print,asciiCbind-method), 32
- show,asciiMixed-method
 - (print,asciiCbind-method), 32
- show,asciiTable-method
 - (print,asciiCbind-method), 32
- show,Report-method
 - (print,asciiCbind-method), 32
- Sweave, 24
- T2t (Asciidoc), 23
- Textile (Asciidoc), 23
- verbatim, 39