Package ‘tableParser’

January 6, 2026

Title Parse Tabled Content to Text Vector and Extract Statistical Standard Results

Date 2026-01-05

Version 1.0.1

Maintainer Ingmar Böschen <ingmar.boeschen@uni-hamburg.de>

Description Features include the ability to extract tabled content from NISO-JATS-coded XML, any native HTML or HML file, DOCX, and PDF documents, and then collapse it into a text format that is readable by humans by mimicking the actions of a screen reader. As tables within PDF documents are extracted with the tabulapdf package, and the table captions and footnotes cannot be extracted, the results on tables within PDF documents have to be considered less precise. The function table2matrix() returns a list of the tables within a document as character matrices. table2text() collapses the matrix content into a list of character strings by imitating the behavior of a screen reader. The textual representation of characters and numbers can be unified with unifyMatrix() before parsing. The function table2stats() extracts the tabled statistical test results from the collapsed text with the function standardStats() from the JATSdecoder package and, if activated, checks the reported and coded p-values for consistency. Due to the great variability and potential complexity of table structures, parsing accuracy may vary.

Depends R (>= 4.1)

Imports utils,
JATSdecoder,
tabulapdf

License GPL-3

URL https://github.com/ingmarboeschen/tableParser

BugReports https://github.com/ingmarboeschen/tableParser/issues

Language en-US

Encoding UTF-8

RoxygenNote 7.3.2

R topics documented:

`docx2matrix` *docx2matrix*

Description

Extracts tables from DOCX documents and returns a list of character matrices.

Usage

```
docx2matrix(x, replicate = TRUE)
```

Arguments

`x`	File path to a DOCX input file with tables.
`replicate`	Logical. If TRUE, replicates content when splitting connected cells.

Value

List with extracted tables as character matrices.

get.caption *get.caption*

Description

Extracts the content of HTML <caption>-tags.

Usage

```
get.caption(x, rm.html = TRUE, sentences = FALSE, letter.convert = TRUE)
```

Arguments

`x`	A vector with HTML-coded tables.
`rm.html`	logical. If TRUE, all HTML tags are removed, <sub> converts to '_', and <sup> to '^'.
`sentences`	logical. If TRUE, a sentence vector is returned.
`letter.convert`	logical. If TRUE, hexadecimal letters are converted to Unicode and unified with JATSdecoder::letter.convert.

Value

A character vector with the extracted caption text and NULL for no caption text

get.footer *get.footer*

Description

Extracts the content of HTML <table-wrap-foot>-tag/s.

Usage

```
get.footer(x, rm.html = TRUE, sentences = FALSE, letter.convert = TRUE)
```

Arguments

`x`	A vector with HTML-coded tables.
`rm.html`	logical. If TRUE, all HTML tags are removed, <sub> converts to '_', and <sup> to '^'.
`sentences`	logical. If TRUE, a sentence vector is returned.
`letter.convert`	logical. If TRUE, hexadecimal letters are converted to Unicode and unified with JATSdecoder::letter.convert.

Value

A character vector with the extracted footer text and NULL for no footer text.

`get.HTML.tables` *get.HTML.tables*

Description

Extracts HTML tables as a vector of HTML-coded tables from plain HTML code, HTML, HML, XML, or CERMXML files.

Usage

```
get.HTML.tables(x)
```

Arguments

`x`	HTML, HML, XML, or CERMXML file or character object with HTML-encoded content.

Value

Character vector with one HTML-encoded table per cell.

`guessCaptionFooterDOCX` *guessCaptionFooterDOCX Extracts text blocks around tables within DOCX files in order to return the tables caption/footer.*

Description

guessCaptionFooterDOCX Extracts text blocks around tables within DOCX files in order to return the tables caption/footer.

Usage

```
guessCaptionFooterDOCX(x, MaxCaptionLength = 1, MaxFooterLength = 4)
```

Arguments

`x`	character. A file path to a DOCX file.
`MaxCaptionLength`	numeric. The maximum number of sentences within a text block that shall be treated as a caption. Text blocks that contain more sentences than this threshold are not extracted at all.
`MaxFooterLength`	numeric. The maximum number of sentences within a text block that shall be treated as a footer. Text blocks that contain more sentences than this threshold are not extracted at all.

Value

A list with the extracted table captions and footers as vectors of length=number of tables.

`legendCodings` *legendCodings*

Description

Extracts the coding of p-values, brackets, abbreviations, superscripts, diagonal content, and the reported sample size/s with 'N=number' from table captions and footer notes/text.

Usage

```
legendCodings(x)
```

Arguments

`x`	An HTML-coded table or plain textual input of table caption and/or footer text.

Value

A list with detected p-value and superscript codings, abbreviations, and reported sample size/s.

Examples

```
x<-"+ p>.05, ^**p<.01, SSq, Sum of Squares, ^a t-test, n=120.
POS: perceived organizational support, JP; job performance.
Numbers in parenthesis are standard errors.
Bold values indicate significance at p<.05."
legendCodings(x)
```

`matrix2text` *matrix2text*

Description

Converts character matrix content to a screen reader-like readable character string. The parsing is performed row-wise in standard mode.

Usage

```
matrix2text(
  x,
  legend = NULL,
  unifyMatrix = TRUE,
  correctComma = FALSE,
  na.rm = TRUE,
  forceClass = NULL,
  expandAbbreviations = TRUE,
  superscript2bracket = TRUE,
  standardPcoding = FALSE,
  noSign2p = FALSE,
  addDF = TRUE,
  rotate = FALSE,
  unlist = FALSE,
  addTableName = TRUE,
  split = FALSE
)
```

Arguments

`x` A character matrix or list of character matrices.

`legend` A list with table legend codes extracted from table caption and/or footer with tableParser::legendCodings().

`unifyMatrix` Logical. If TRUE, matrix cells are unified for better post-processing.

`correctComma` Logical. If TRUE and unifyMatrix=TRUE, decimal sign commas are converted to dots.

`na.rm` Logical. If TRUE, NA cells are set to empty cells.

`forceClass` character. Set matrix-specific handling to one of c("tabled result", "correlation", "matrix", "text").

`expandAbbreviations`
Logical. If TRUE, detected abbreviations are expanded to label detected in table caption/footer with tableParser::legendCodings().

`superscript2bracket`
Logical. If TRUE, detected superscript codings are inserted inside parentheses.

`standardPcoding`
Logical. If TRUE, and no other detection of p-value coding is detected, standard coding of p-values is assumed to be: * p<.05, ** p<.01, and *** p<.001.

`noSign2p` Logical. If TRUE, imputes 'p>maximum of coded p-values' to cells that are not coded to be significant.

`addDF` Logical. If TRUE, detected sample size N in the caption/footer is inserted as degrees of freedom (N-2) to r- and t-values that are reported without degrees of freedom.

`rotate` Logical. If TRUE, matrix content is parsed by column.

`unlist` Logical. If TRUE, output is returned as a vector with parsed text from all listed matrices; else, a list with parsed text from each matrix is returned as a list.

`addTableName` Logical. If TRUE and unlist=TRUE, the table number is added in front of unlisted text lines.

`split` Logical. If TRUE, multi-model tables are split before being processed.

Value

Character vector with a parsed and human-readable form of the input table. The result vector can be further processed with standardStats() to extract and structure the statistical standard test results only.

Examples

```
# some random data
x<-rnorm(100)
y<-x+rnorm(100)
# a model result table...
mod<-round(summary(lm(y~x))$coefficients,3)
rnames<-c(””,rownames(mod))
cnames<-colnames(mod)
mod<-rbind(cnames,mod)
mod<-cbind(rnames,mod)
# ...as character result matrix
x<-unname(mod);x
## parse matrix to text vector
# -as is
matrix2text(x,unifyMatrix=FALSE)
# -with unified content
matrix2text(x,unifyMatrix=TRUE)
## processing of a matrix with two header lines
x<-rbind(c(””,”A”,”A”,”B”,”B”),x);x
matrix2text(x,unifyMatrix=FALSE)
## processing of a matrix with two header lines and naming columns
x<-cbind(c(””,””,”C”,”D”),x);x
matrix2text(x,unifyMatrix=FALSE)
```

`parseMatrixContent` *parseMatrixContent*

Description

Parses character matrix content into a text vector. This is the basic function of tableParser, which is implemented in matrix2text(), table2text(), and table2stats(). Row and column names are parsed to cell content with operators that depend on the cell content. Numeric cells are parsed with "=", and textual cell content with ":". Cells that start with an operator ('<', '=' or '>') are parsed without a separator. Detected codings for (e.g., p-values, abbreviations) from table legend text can be used to extend the tabled content to a fully written-out form.

Usage

```
parseMatrixContent(
  x,
  legend = NULL,
  standardPcoding = TRUE,
  noSign2p = TRUE,
  forceClass = NULL,
  expandAbbreviations = TRUE,
  superscript2bracket = FALSE,
  addDF = TRUE
)
```

Arguments

`x` A character matrix or list with a character matrix as first and only element.

`legend` The table's caption/footer notes as a character vector.

`standardPcoding`
Logical. If TRUE, and no other detection of p-value coding is detected, standard coding of p-values is assumed to be: * p<.05, ** p<.01, and *** p<.001.

`noSign2p` Logical. If TRUE, imputes 'p>maximum of coded p-values' to cells that are not coded to be significant.

`forceClass` Character. Set a fixed table class for extraction heuristic. One of c("tabled result", "correlation", "matrix", "text").

`expandAbbreviations`
Logical. If TRUE, detected abbreviations are expanded to label detected in table caption/footer with tableParser::legendCodings().

`superscript2bracket`
Logical. If TRUE, detected superscript codings are inserted inside parentheses.

`addDF` Logical. If TRUE, detected sample size N in the caption/footer is inserted as degrees of freedom (N-2) to r- and t-values that are reported without degrees of freedom.

Value

A text vector with the parsed matrix content.

prepareMatrix *prepareMatrix*

Description

Prepares character matrix content for parsing.

Usage

```
prepareMatrix(x, split = FALSE, forceClass = NULL, na.rm = TRUE)
```

Arguments

`x` character matrix

`split` logical. If TRUE, multi-model matrices are split into a list of single-model matrices.

`forceClass` character. Set matrix-specific handling to one of c("tabled result", "correlation", "matrix, "text").

`na.rm` Logical. If TRUE, NA cells are set to empty cells.

Value

A character matrix

`table2matrix` *table2matrix*

Description

Extracts tables from HTML, HML, XML, CERMXML, DOCX, PDF files, or plain HTML code to a list of character matrices.

Usage

```
table2matrix(
  x,
  unifyMatrix = FALSE,
  letter.convert = TRUE,
  greek2text = FALSE,
  replicate = FALSE,
  repNums = FALSE,
  rm.html = FALSE,
  rm.empty.row.col = FALSE,
  collapseHeader = TRUE,
  header2colnames = FALSE
)
```

Arguments

`x` File path to a DOCX, PDF, or HTML-encoded file, or text with HTML code.

`unifyMatrix` Logical. If TRUE, matrix cells are unified for better post-processing (see unifyMatrixContent()).

`letter.convert` Logical. If TRUE, hex codes will be unified and converted to Unicode with JATSdecoder::letter.convert().

`greek2text` Logical. If TRUE and 'letter.convert=TRUE', converts and unifies various Greek letters to a text-based form (e.g.: 'alpha', 'beta').

`replicate` Logical. If TRUE, the content of cells with row/col span > 1 is replicated in all connected cells; if FALSE, the value will only be placed in the first of the connected cells.

`repNums` Logical. If TRUE, cells with numbers that have row/col span > 1 are replicated in every connected cell.

`rm.html` Logical. If TRUE, all HTML tags are removed, except <sub> and <sup>, and </break> is converted to space.

`rm.empty.row.col`
Logical. If TRUE, empty rows/columns are removed from output.

`collapseHeader` Logical. If TRUE, header cells are collapsed for each column if the header has 2 or more lines.

`header2colnames`
Logical. If TRUE and 'collapseHeader=TRUE', the first table row is used for column names and removed from the table.

Value

List with detected tables as character matrices.

Examples

```
x<-readLines("https://en.wikipedia.org/wiki/R_(programming_language)",warn=FALSE)
table2matrix(x,rm.html=TRUE)
```

table2stats *table2stats*

Description

Extracts tabulated statistical results from documents in XML, HTML, HML, DOCX, or PDF format.

Usage

```
table2stats(
  x,
  standardPcoding = FALSE,
  noSign2p = TRUE,
  correctComma = FALSE,
  rotate = FALSE,
  expandAbbreviations = TRUE,
  superscript2bracket = TRUE,
  stats.mode = "all",
  checkP = FALSE,
  alpha = 0.05,
  criticalDif = 0.02,
  alternative = "undirected",
  estimateZ = FALSE,
  T2t = FALSE,
  addDF = TRUE,
  collapse = TRUE,
  addTableName = FALSE,
  rm.na.col = TRUE
)
```

Arguments

`x` Input. Either a file path to an XML, HTML, HML, DOCX, or PDF file; or a matrix object; or a vector of plain HTML-coded tables.

`standardPcoding` Logical. If TRUE, and no other detection of coding is detected, then standard coding of p-values is assumed to be * p<.05, ** p<.01, and ***p<.001.

`noSign2p` Logical. If TRUE, imputes ’p>maximum of coded p-values’ to cells that are not coded to be significant.

`correctComma` Logical. If TRUE, decimal sign commas are converted to dots.

`rotate` Logical. If TRUE, matrix content is parsed by column.

`expandAbbreviations` Logical. If TRUE, detected abbreviations are expanded to label from table caption/footer.

`superscript2bracket` Logical. If TRUE, detected superscript codings are inserted inside parentheses.

`stats.mode` Select a subset of test results by p-value checkability for output. One of: c("all", "checkable", "computable", "uncomputable").

`checkP` Logical. If TRUE, detected p-values and recalculated p-values will be checked for consistency.

`alpha` Numeric. Defines the alpha level to be used for error assignment.

`criticalDif` Numeric. Sets the absolute maximum difference in reported and recalculated p-values for error detection.

`alternative` Character. Select test sidedness for recomputation of p-values from t-, r-, and beta-values. One of c("undirected", "directed"). If "directed" is specified, p-values for directed null hypotheses are added to the table but still require a manual inspection of the consistency of the direction.

`estimateZ` Logical. If TRUE, detected beta-/d-values are divided by the reported standard error "SE" to estimate Z-values ("Zest") for observed beta/d and computation of p-values. Note: This is only valid if Gauss-Markov assumptions are met and a sufficiently large sample size is used. If a Z- or t-value is detected in a report of a beta-/d-coefficient with SE, no estimation will be performed, although set to TRUE.

`T2t` Logical. If TRUE, capital letter T is treated as a t-statistic.

`addDF` Logical. If TRUE, detected sample size N in the caption/footer is inserted as degrees of freedom (N-2) to r- and t-values that are reported without degrees of freedom.

`collapse` Logical. If TRUE, the result is collapsed to a single data frame object. Else, a list of data frames with length = n matrices is returned.

`addTableName` Logical. If TRUE, the table number is added in front of the extracted results.

`rm.na.col` Logical. If TRUE, removes all columns with only NA.

Value

A data.frame object with the extracted statistical standard results, recalculated p-values and a rudimentary, optional consistency check for reported p-values (if 'checkP=TRUE').

`table2text` *table2text*

Description

Parses tabled content from HTML-coded content, or HTML, DOCX, or PDF file to human-readable text vector. Before parsing, header lines are collapsed and connected cells are broken up.

Usage

```
table2text(
  x,
  unifyMatrix = TRUE,
  unifyStats = FALSE,
  expandAbbreviations = TRUE,
  superscript2bracket = TRUE,
  standardPcoding = FALSE,
  noSign2p = FALSE,
  addDF = FALSE,
  rotate = FALSE,
  correctComma = FALSE,
  na.rm = TRUE,
  addDescription = TRUE,
  unlist = FALSE,
  addTableName = TRUE
)
```

Arguments

`x` A vector with HTML tables, or a single file path to an HTML, XML, CERMXML, HML, PDF, or DOCX file.

`unifyMatrix` Logical. If TRUE, matrix cells are unified for better post-processing.

`unifyStats` Logical. If TRUE, output is unified for better post-processing (e.g., "p-value"->"p").

`expandAbbreviations`
Logical. If TRUE, detected abbreviations are expanded to label from table caption/footer.

`superscript2bracket`
Logical. If TRUE, detected superscript codings are inserted inside parentheses.

`standardPcoding`
Logical. If TRUE, and no other detection of coding is detected, standard coding of p-values is assumed to be * p<.05, ** p<.01, and ***p<.001.

`noSign2p` Logical. If TRUE, imputes 'p>maximum of coded p-values' to cells that are not coded to be significant.

`addDF` Logical. If TRUE, detected sample size N in the caption/footer is inserted as degrees of freedom (N-2) to r- and t-values that are reported without degrees of freedom.

`rotate` Logical. If TRUE, matrix content is parsed by column.

`correctComma` Logical. If TRUE and unifyMatrix=TRUE, decimal sign commas are converted to dots.

`na.rm` Logical. If TRUE, NA cells are set to empty cells.

`addDescription` Logical. If TRUE, the table caption and footer are added before the extracted table content for better readability.

`unlist` Logical. If TRUE, output is returned as a vector.

`addTableName` Logical. If TRUE and unlist=TRUE, the table number is added in front of unlisted text lines.

Value

A list with text vectors of the parsed table content by table. The text vector in each list element can be further processed with JATSdecoder::standardStats() to extract and structure the statistical standard test results.

`tableClass` *tableClass*

Description

Classifies matrix content to either 'tabled results', 'correlation', 'matrix', 'text', 'vector', 'model with model statistics', or 'multi model with model statistics'.

Usage

```
tableClass(x, legend = NULL)
```

Arguments

`x` A character matrix

`legend` A text vector with the tables caption and/or footer.

Value

A character object of length=1 with the table's class.

`unifyMatrixContent` *unifyMatrixContent*

Description

Unifies textual and numerical content of character matrices. Unifies hyphens, spaces, hexadecimal and Greek letters, and performs space and comma corrections. Big marks in numbers are removed. HTML tags <sup> and <sub> are converted to '^' and '_' respectively. All other HTML tags are removed.

Usage

```
unifyMatrixContent(
  x,
  letter.convert = TRUE,
  greek2text = TRUE,
  text2num = TRUE,
  correctComma = FALSE,
  na.rm = TRUE
)
```

Arguments

`x` a character matrix.

`letter.convert` Logical. If TRUE, hexadecimal-coded letters will be unified and converted to Unicode with JATSdecoder::letter.convert().

`greek2text` Logical. If TRUE and 'letter.convert=TRUE', converts and unifies various Greek letters to a text-based form (e.g., 'alpha', 'beta').

`text2num` Logical. If TRUE, textual representations of numbers (words, exponents, fractions) are converted to digit numbers.

`correctComma` Logical. If TRUE, commas used as numeric separators are converted to dots.

`na.rm` Logical. If TRUE, cells with NA, or only minus, hyphen, slash, or dot are set to empty cells.

`unifyStats` *unifyStats*

Description

Unifies many textual representations of statistical results in text vectors created with table2text(). This uniformization is needed for a more precise extraction of standard results with the function standardStats() from the JATSdecoder package.

Usage

```
unifyStats(x)
```

Arguments

`x`	A text vector with the parsed table content.

Value

A unified text string.

Index