Package: tableParser (via r-universe)

October 14, 2026

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

Date 2026-04-09

Version 1.0.5

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. A detailed description of how 'tableParser' works is provided here: <doi:10.48550/arXiv.2603.19756>.

Depends R (>= 4.1)

Imports utils, stats, 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

NeedsCompilation no

Author Ingmar Böschen [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1159-3991>)

Repository https://cran.r-universe.dev

Date/Publication 2026-04-09 09:44:20 UTC

RemoteUrl https://github.com/cran/tableParser

RemoteRef HEAD

RemoteSha 354432b5ca31af3902a5ffa184fc68df7ff10665

Contents

docx2matrix . . . 2
get.caption . . . 3
get.footer . . . 4
get.HTML.tables . . . 4
guessCaptionFootnote . . . 5
html2unicode . . . 6
legendCodings . . . 6
matrix2text . . . 7
parseMatrixContent . . . 9
prepareMatrix . . . 11
table2matrix . . . 12
table2stats . . . 14
table2text . . . 16
tableClass . . . 18
unifyMatrixContent . . . 19
unifyStats . . . 20

Index **21**

`docx2matrix` *docx2matrix*

Description

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

Usage

```
docx2matrix(x, unifyMatrix = FALSE, correctComma = FALSE, replicate = TRUE)
```

Arguments

x	File path to a DOCX input file with tables.
unifyMatrix	Logical. If TRUE, matrix cells are unified for better post-processing (see 'unifyMatrixContent()').
correctComma	Logical. If TRUE, commas used as decimal are converted to dots, big mark commas are removed.
replicate	Logical. If TRUE, replicates content when splitting connected cells.

Value

List with extracted tables as character matrices.

Examples

```
## Download an example DOCX file from tableParser's github repo to temp directory
d<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.docx'
tempFile<-paste0(tempdir(),"/","tableExamples.docx")

# on Windows with method="wget"
if(grepl("^[A-z]:",tempFile))
   download.file(d,tempFile,method="wget")
# on all other machines
if(!grepl("^[A-z]:",tempFile))
   download.file(d,tempFile)

Sys.sleep(.2)

# Extract tables as character matrices
docx2matrix(tempFile)
```

---

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, or XML files. If tables are nested within tables, only the inner tables are extracted.

Usage

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

Arguments

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

Value

Character vector with one HTML-encoded table per cell.

Examples

```
x<-readLines("https://en.wikipedia.org/wiki/R_(programming_language)",warn=FALSE)
get.HTML.tables(x)
```

---

guessCaptionFootnote *guessCaptionFootnote*

Description

Extracts text blocks around tables within DOCX, HTML, HML, XML, or NXML files in order to return the table captions and footnotes.

Usage

```
guessCaptionFootnote(x, MaxCaptionLength = 1, MaxFootnoteLength = 4)
```

Arguments

x	character. A file path.
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.
MaxFootnoteLength	numeric. The maximum number of sentences within a text block that shall be treated as a footnote. Text blocks that contain more sentences than this threshold are not extracted.

Value

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

Examples

```
## Download an example DOCX file from tableParser's github repo to temp directory
d<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.docx'
download.file(d,paste0(tempdir(),"/","tableExamples.docx"))

## Download an example HTML file from tableParser's github repo to temp directory
h<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.html'
download.file(h,paste0(tempdir(),"/","tableExamples.html"))

## Extract table captions and footnotes
# DOCX file
```

```
guessCaptionFootnote(paste0(tempdir(),"/","tableExamples.docx"))
# HTML file
guessCaptionFootnote(paste0(tempdir(),"/","tableExamples.html"))
```

`html2unicode` *html2unicode*

Description

Converts HTML encoded special letters to unicode.

Usage

```
html2unicode(x)
```

Arguments

x	A character vector or matrix.

Value

A character vector or matrix.

References

https://www.w3.org/TR/REC-html40/sgml/entities.html

Examples

```
html2unicode(x<-"&#34;, &#161;, &#162;.")
```

`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 footnote text.

Usage

```
legendCodings(x)
```

Arguments

x	An HTML-coded table or plain textual input of table caption and/or footnote 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,
  decodeP = FALSE,
  standardPcoding = FALSE,
  noSign2p = FALSE,
  bracketHandling = FALSE,
  dfHandling = 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 footnote with 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/footnote with legendCodings().
superscript2bracket	Logical. If TRUE, detected superscript codings are inserted inside parentheses.
decodeP	Logical. If TRUE, imputes the converts the detected p-value codings to text with seperator ';;' (e.g., '1.23*' -> '1.23;; p<.01')
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.
bracketHandling	Logical. If TRUE and if possible, decodes numbers in brackets.
dfHandling	Logical. If TRUE, detected sample size N in the caption/footnote 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 grouping column [,1]
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,
  decodeP = TRUE,
  standardPcoding = TRUE,
  noSign2p = TRUE,
  bracketHandling = TRUE,
  forceClass = NULL,
  expandAbbreviations = TRUE,
  superscript2bracket = FALSE,
  dfHandling = TRUE
)
```

Arguments

x	A character matrix or list with a character matrix as first and only element.
legend	The table's caption/footnote as a character vector.
decodeP	Logical. If TRUE, imputes the converts the detected p-value codings to text with seperator ';;' (e.g., '1.23*' -> '1.23;; p<.01')
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 the detected p-value codes to cells that do have a coding sign.
bracketHandling	Logical. If TRUE and if possible, decodes numbers in brackets.
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/footnotes with legendCodings().
superscript2bracket	Logical. If TRUE, detected superscript codings are inserted inside parentheses.
dfHandling	Logical. If TRUE, detected sample size N in the caption/footnotes is inserted as degrees of freedom (N-2) to r- and t-values that are reported without degrees of freedom, if the detected N>3.

Value

A text vector with the parsed matrix content.

Examples

```
# Example matrix
m<-rbind(c("","ß","Standard Error","Pr(>|t|)"),
         c("(Intercept)","1,234.5","123.4","1.3e-4"),
         c("Variable 1","1,2",".04","2.4*10^-5"),
         c("R^2",".23","*","-"))
m

# apply function
parseMatrixContent(m)
```

prepareMatrix *prepareMatrix*

Description

Prepares character matrix content for parsing. Removes empty rows and columns, extends content from plausible grouping cells to sparse cells, collapses multiple header rows, and splits multiple model tables to a list of single model tables.

Usage

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

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.
legend	Character. Optional, text from table caption and/or footnote for table class specific processing.

Value

A character matrix

Examples

```
# example matrix
x<-cbind(c("","","name","","",""),
         c("group","name","A","B","","C"),
         c("value","","1","2","","3"))
x

# apply function
prepareMatrix(x)
```

table2matrix *table2matrix*

Description

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

Usage

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

Arguments

x	A 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, html and hexadecimal encoded letters will be unified and converted to Unicode with html2unicode() and 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').
correctComma	Logical. If TRUE, commas used as decimal are converted to dots, big mark commas are removed.
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

```
## - Download example DOCX file
d<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.docx'
download.file(d,paste0(tempdir(),"/","tableExamples.docx"))

# Extract tables from example file as matrices
table2matrix(paste0(tempdir(),"/","tableExamples.docx"))

## - Download example HTML file
h<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.html'
download.file(h,paste0(tempdir(),"/","tableExamples.html"))

# Extract tables from example file as matrices
table2matrix(paste0(tempdir(),"/","tableExamples.html"),rm.html=TRUE)

## - Download example PDF file
p<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.pdf'
download.file(p,paste0(tempdir(),"/","tableExamples.pdf"))

# Extract tables from example file as matrices

table2matrix(paste0(tempdir(),"/","tableExamples.pdf"))

# Note: The extraction of tables within PDF documents with tabulapdf::extract_tables()
# does not work properly here.
# Also, the table captions and footnotes cannot be used for decoding (e.g., p-values).

tabulapdf::extract_tables(paste0(tempdir(),"/","tableExamples.pdf"))


## Another example with a website that contains simple and nested HTML-tables

# download file
x<-readLines("https://en.wikipedia.org/wiki/R_(programming_language)",warn=FALSE)

# apply function
table2matrix(x,rm.html=TRUE,unifyMatrix=TRUE)
```

table2stats *table2stats*

Description

Extracts tabulated statistical results from documents in XML, HTML, HML, DOCX, or PDF format. The tabled content is collapsed into a text string with table2text(), which is then processed with standardStats() from the 'JATSdecoder' package. It detects most standard statistics (t, Z, chi^2, F, r, d, beta, SE, r, d, eta^2, omega^2, OR, RR, p-values), decodes encoded p-values to text and recalculates and checks p-values if possible.

Usage

```
table2stats(
  x,
  standardPcoding = FALSE,
  checkP = FALSE,
  noSign2p = FALSE,
  criticalDif = 0.02,
  alternative = "undirected",
  estimateZ = FALSE,
  T2t = FALSE,
  correctComma = TRUE,
  stats.mode = "all",
  alpha = "auto",
  dfHandling = TRUE,
  collapse = TRUE,
  addTableName = TRUE,
  rotate = FALSE,
  expandAbbreviations = TRUE,
  superscript2bracket = TRUE,
  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 * for p<.05, ** for p<.01, and *** for p<.001.
checkP	Logical. If TRUE, detected p-values and recalculated p-values will be checked for consistency.
noSign2p	Logical. If TRUE, imputes 'p>maximum of coded p-values' to cells that are not coded to be significant.

`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.

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

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

`alpha` Numeric or "auto". Defines the alpha level to be used for error assignment. If set to "auto", table notes are screened for reports of alpha levels, 1-alpha confidence intervals and correction procedures for multiple testing. If no reported alpha levels is detected, the value is set to the widely used standard 'alpha=.05'.

`dfHandling` 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, when collapsed to a single data frame with 'collapse=TRUE'.

`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.

`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').

See Also

`get.stats` for extracting statistical results from textual resources.

Examples

```
## - Download example DOCX file
d<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.docx'
download.file(d,paste0(tempdir(),"/","tableExamples.docx"))

# Extract the detected statistical standard results and validate the reported and coded
# p-values with the recalculated p-values.
table2stats(paste0(tempdir(),"/","tableExamples.docx"), checkP=TRUE, estimateZ=TRUE)

## - Download example HTML file
h<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.html'
download.file(h,paste0(tempdir(),"/","tableExamples.html"))
# Extract the detected statistical standard results and validate the reported and coded
# p-values with the recalculated p-values.
table2stats(paste0(tempdir(),"/","tableExamples.html"), checkP=TRUE, estimateZ=TRUE)
# - Download example PDF file

p<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.pdf'
download.file(p,paste0(tempdir(),"/","tableExamples.pdf"))

# Extract the detected statistical standard results and validate the reported and
# standard coded as well as not coded p-values with the recalculated p-values.
table2stats(paste0(tempdir(),"/","tableExamples.pdf"), checkP=TRUE, estimateZ=TRUE,
standardPcoding=TRUE, noSign2p=FALSE)
# Note: Due to the messy table extraction with tabulapdf::extract_tables(), the
# extraction of the statistical results is less precise here.
```

`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,
  decodeP = TRUE,
  noSign2p = FALSE,
  bracketHandling = TRUE,
  dfHandling = 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, 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/footnote.
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.
decodeP	Logical. If TRUE, imputes the converts the detected p-value codings to text with seperator ';;' (e.g., '1.23*' -> '1.23;; p<.01')
noSign2p	Logical. If TRUE, imputes 'p>maximum of coded p-values' to cells that are not coded to be significant.
bracketHandling	Logical. If TRUE and if possible, decodes numbers in brackets.
dfHandling	Logical. If TRUE, the detected sample size N in a row is inserted as degrees of freedom (N-2) to r- and t-values that are reported without degrees of freedom (df) in that row. In ANOVA tables, the detected residual df (df2) is imputed behind the faktor df (df1).
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 attributes table caption and table footnote are added in front of the extracted character 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.

Examples

```
## - Download example DOCX file
d<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.docx'
download.file(d,paste0(tempdir(),"/","tableExamples.docx"))

# Parse tabled content from example file to text vectors.
table2text(paste0(tempdir(),"/","tableExamples.docx"))

## - Download example HTML file
h<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.html'
download.file(h,paste0(tempdir(),"/","tableExamples.html"))

# Parse tabled content from example file to text vectors.
table2text(paste0(tempdir(),"/","tableExamples.html"),unlist=TRUE,addDescription=TRUE)


## - Download example PDF file
p<-'https://github.com/ingmarboeschen/tableParser/raw/refs/heads/main/tableExamples.pdf'
download.file(p,paste0(tempdir(),"/","tableExamples.pdf"))

# Parse tabled content from example file to text vectors.
table2text(paste0(tempdir(),"/","tableExamples.pdf"),decodeP=TRUE,standardPcoding=TRUE)
```

`tableClass` *tableClass*

Description

Classifies matrix content to either 'tabled results', 'matrix', 'correlation', 'vector', 'text', '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 footnote.

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 or list of character matrices.
letter.convert	Logical. If TRUE, hexadecimal- and html-encoded 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 decimal are converted to dots, big mark commas are removed.
na.rm	Logical. If TRUE, cells with NA, or only minus, hyphen, slash, or dot are set to empty cells.

Value

A unified character matrix or list of character matrices.

Examples

```
# Example matrix
m<-rbind(c("","ß","Standard Error","Pr(>|t|)"),
         c("(Intercept)","1,234.5","123.4","1.3e-4"),
         c("Variable 1","1,2",".04","2.4*10^-5"),
         c("R^2",".23","*","-"))
```

```
m

# apply function
unifyMatrixContent(m, correctComma = TRUE)
```

---

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, dfHandling = FALSE)
```

Arguments

x	A text vector with the parsed table content.
dfHandling	Logical. If TRUE, ANOVA specific handling of degrees of freedom (df) is performed. The detected residual df (df2) is imputed behind the faktor df (df1). Note: Should only be activated, when unifyStats() is applied to the collapsed content of a single table.

Value

A unified text string.

Examples

```
# Example matrix
m<-rbind(c("","ß","Standard Error","Pr(>|t|)"),
c("(Intercept)","1,234.5","123.4","1.3e-4"),
c("Variable 1","1,2",".04","2.4*10^-5"),
c("R^2",".23","*","-"))
m

# parsed content
text<-parseMatrixContent(unifyMatrixContent(m, correctComma = TRUE))
text

# unified stats
unifyStats(text)
```

Index

docx2matrix, 2

get.caption, 3
get.footer, 4
get.HTML.tables, 4
get.stats, *15*
guessCaptionFootnote, 5

html2unicode, 6

legendCodings, 6

matrix2text, 7

parseMatrixContent, 9
prepareMatrix, 11

table2matrix, 12
table2stats, 14
table2text, 16
tableClass, 18

unifyMatrixContent, 19
unifyStats, 20