

# Package ‘tableParser’

January 17, 2026

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

**Date** 2026-01-16

**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 . . . . . 2

get.caption . . . . .	3
get.footer . . . . .	3
get.HTML.tables . . . . .	4
guessCaptionFootnote . . . . .	4
html2unicode . . . . .	5
legendCodings . . . . .	6
matrix2text . . . . .	6
parseMatrixContent . . . . .	8
prepareMatrix . . . . .	10
table2matrix . . . . .	11
table2stats . . . . .	13
table2text . . . . .	15
tableClass . . . . .	17
unifyMatrixContent . . . . .	18
unifyStats . . . . .	19
<b>Index</b>	<b>20</b>

---

docx2matrix	<i>docx2matrix</i>
-------------	--------------------

---

**Description**

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

**Usage**

```
docx2matrix(x, unifyMatrix = TRUE, 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()).
- 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'
download.file(d,paste0(tempdir(),"/", "tableExamples.docx"),method="wget")

# Extract tables as character matrices
docx2matrix(paste0(tempdir(),"/", "tableExamples.docx"))
```

---

get.caption	<i>get.caption</i>
-------------	--------------------

---

**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	<i>get.footer</i>
------------	-------------------

---

**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	<i>get.HTML.tables</i>
-----------------	------------------------

---

### 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	<i>guessCaptionFootnote</i>
----------------------	-----------------------------

---

### Description

Extracts text blocks around tables within DOCX or HTML/HML/XML/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/ingmarboeschon/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/ingmarboeschon/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	<i>legendCodings</i>
---------------	----------------------

---

### 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	<i>matrix2text</i>
-------------	--------------------

---

### 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

<code>x</code>	A character matrix or list of character matrices.
<code>legend</code>	A list with table legend codes extracted from table caption and/or footnote with <code>tableParser::legendCodings()</code> .
<code>unifyMatrix</code>	Logical. If TRUE, matrix cells are unified for better post-processing.
<code>correctComma</code>	Logical. If TRUE and <code>unifyMatrix=TRUE</code> , decimal sign commas are converted to dots.
<code>na.rm</code>	Logical. If TRUE, NA cells are set to empty cells.
<code>forceClass</code>	character. Set matrix-specific handling to one of <code>c("tabled result", "correlation", "matrix", "text")</code> .
<code>expandAbbreviations</code>	Logical. If TRUE, detected abbreviations are expanded to label detected in table caption/footnote with <code>tableParser::legendCodings()</code> .
<code>superscript2bracket</code>	Logical. If TRUE, detected superscript codings are inserted inside parentheses.
<code>decodeP</code>	Logical. If TRUE, imputes the converts the detected p-value codings to text with separator ';;' (e.g., '1.23*' -> '1.23;; p<.01')
<code>standardPcoding</code>	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.
<code>noSign2p</code>	Logical. If TRUE, imputes 'p>maximum of coded p-values' to cells that are not coded to be significant.
<code>bracketHandling</code>	Logical. If TRUE and if possible, decodes numbers in brackets.
<code>dfHandling</code>	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.
<code>rotate</code>	Logical. If TRUE, matrix content is parsed by column.
<code>unlist</code>	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)
```



## 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 tableParser::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.

**Value**

A text vector with the parsed matrix content.

**Examples**

```
# Example matrix
m<-rbind(c("", "B", "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	<i>prepareMatrix</i>
---------------	----------------------

---

**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)
```

**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

**Examples**

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

x

# apply function
prepareMatrix(x)
```

---

table2matrix	<i>table2matrix</i>
--------------	---------------------

---

**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,
  replicate = FALSE,
  repNums = FALSE,
  rm.html = FALSE,
  rm.empty.row.col = FALSE,
  collapseHeader = TRUE,
  header2colnames = FALSE
)
```

**Arguments**

<code>x</code>	A file path to a DOCX, PDF, or HTML encoded file, or text with HTML code.
<code>unifyMatrix</code>	Logical. If TRUE, matrix cells are unified for better post-processing (see <code>unifyMatrixContent()</code> ).
<code>letter.convert</code>	Logical. If TRUE, html and hexadecimal encoded letters will be unified and converted to Unicode with <code>html2unicode()</code> and <code>JATSdecoder::letter.convert()</code> .
<code>greek2text</code>	Logical. If TRUE and 'letter.convert=TRUE', converts and unifies various Greek letters to a text-based form (e.g.: 'alpha', 'beta').
<code>replicate</code>	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/ingmarboesch/en/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/ingmarboesch/en/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/ingmarboesch/en/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	<i>table2stats</i>
-------------	--------------------

---

**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,
  dfHandling = 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.
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.
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').

## Examples

```
## - Download example DOCX file
d<-'https://github.com/ingmarboesch/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/ingmarboesch/en/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/ingmarboesch/en/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)
```

---

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 = TRUE,
  na.rm = TRUE,
  addDescription = TRUE,
  unlist = FALSE,
  addTableName = TRUE
)
```

**Arguments**

<code>x</code>	A vector with HTML tables, or a single file path to an HTML, XML, HML, PDF, or DOCX file.
<code>unifyMatrix</code>	Logical. If TRUE, matrix cells are unified for better post-processing.
<code>unifyStats</code>	Logical. If TRUE, output is unified for better post-processing (e.g., "p-value" -> "p").
<code>expandAbbreviations</code>	Logical. If TRUE, detected abbreviations are expanded to label from table caption/footnote.
<code>superscript2bracket</code>	Logical. If TRUE, detected superscript codings are inserted inside parentheses.
<code>standardPcoding</code>	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.
<code>decodeP</code>	Logical. If TRUE, imputes the converts the detected p-value codings to text with seperator ';;' (e.g., '1.23*' -> '1.23;; p<.01')
<code>noSign2p</code>	Logical. If TRUE, imputes 'p>maximum of coded p-values' to cells that are not coded to be significant.
<code>bracketHandling</code>	Logical. If TRUE and if possible, decodes numbers in brackets.
<code>dfHandling</code>	Logical. If TRUE, the 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.
<code>rotate</code>	Logical. If TRUE, matrix content is parsed by column.
<code>correctComma</code>	Logical. If TRUE and unifyMatrix=TRUE, decimal sign commas are converted to dots.
<code>na.rm</code>	Logical. If TRUE, NA cells are set to empty cells.
<code>addDescription</code>	Logical. If TRUE, the attributes table caption and table footnote are added in front of the extracted character content for better readability.
<code>unlist</code>	Logical. If TRUE, output is returned as a vector.
<code>addTableName</code>	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/ingmarboeschon/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/ingmarboeschon/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/ingmarboeschon/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', '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 footnote.

## Value

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

---

unifyMatrixContent	<i>unifyMatrixContent</i>
--------------------	---------------------------

---

## 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

<code>x</code>	a character matrix.
<code>letter.convert</code>	Logical. If TRUE, hexadecimal- and html-encoded letters will be unified and converted to Unicode with <code>JATSdecoder::letter.convert()</code> .
<code>greek2text</code>	Logical. If TRUE and <code>'letter.convert=TRUE'</code> , converts and unifies various Greek letters to a text-based form (e.g., <code>'alpha'</code> , <code>'beta'</code> ).
<code>text2num</code>	Logical. If TRUE, textual representations of numbers (words, exponents, fractions) are converted to digit numbers.
<code>correctComma</code>	Logical. If TRUE, commas used as numeric separators are converted to dots.
<code>na.rm</code>	Logical. If TRUE, cells with NA, or only minus, hyphen, slash, or dot are set to empty cells.

## Examples

```
# Example matrix
m<-rbind(c("", "B", "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)
```

**Arguments**

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

**Value**

A unified text string.

**Examples**

```
# Example matrix
m<-rbind(c("", "β", "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`, [3](#)

`get.HTML.tables`, [4](#)

`guessCaptionFootnote`, [4](#)

`html2unicode`, [5](#)

`legendCodings`, [6](#)

`matrix2text`, [6](#)

`parseMatrixContent`, [8](#)

`prepareMatrix`, [10](#)

`table2matrix`, [11](#)

`table2stats`, [13](#)

`table2text`, [15](#)

`tableClass`, [17](#)

`unifyMatrixContent`, [18](#)

`unifyStats`, [19](#)