

# Package ‘wfr’

December 7, 2023

**Type** Package

**Title** A Work-Flow to Assist Document Creation for R Data Analysis

**Version** 0.5.4

**VignetteBuilder** knitr

## Description

Greatly reduces the amount of non-statistical work in conducting large data analysis projects using R and compiling reports containing many table and figures; specifically: 1) It provides a way to systematically record the outputs from the data analysis R script, even including format information if the output is a table. 2) It can automatically create a R markdown file to produce either 'MS Word', HTML, or PDF output. 3) Numeric columns of the tables in the report are formatted automatically and properly based on the distribution of the column.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0), flextable (>= 0.5.5)

**Imports** utils, knitr, kableExtra, stringr, magrittr, captioner, officer, stats, ggplot2, openxlsx, xtable

**Suggests** bookdown,

pander,  
tinytex,  
webshot,  
testthat

**RoxygenNote** 7.2.3

**URL** <http://github.com/blueskypie/wfr>

**BugReports** <http://github.com/blueskypie/wfr/issues>

## R topics documented:

createRmd . . . . .	2
is.empty . . . . .	4
loadLibs . . . . .	4
nDecimal . . . . .	5

num2formattedStr . . . . .	6
rmdTable . . . . .	7
saveOutput . . . . .	10
saveWfrInfo . . . . .	12
setHtmlHeaderProperty . . . . .	13
setNsmall . . . . .	14
setWidths . . . . .	15
showObj . . . . .	16
str2list . . . . .	17
table.cross.ref . . . . .	18
template.files . . . . .	19
writeExcel . . . . .	19

<b>Index</b>	<b>21</b>
--------------	-----------

---

createRmd	<i>Create a R markdown file</i>
-----------	---------------------------------

---

## Description

Automatically creates a Rmd file by appending knitr code chunks to an existing [rmd.template](#)

## Usage

```
createRmd(outputRmdFn, outputListFn, rmdTemplateFn =
  system.file("extdata", "rmd.template.Rmd", package = "wfr"),
  libs = c("ggplot2"), tabPars=NULL)
```

## Arguments

outputRmdFn	(character) The file name of the produced Rmd file, can contain file paths.
outputListFn	(character) The file name of the text file containing the table from <a href="#">OUTPUTS</a>
rmdTemplateFn	(character, system.file("extdata", "rmd.template.Rmd", package = "wfr")) The file name of the <a href="#">rmd.template</a>
libs	(character, "ggplot2") The names of the libraries needed to execute the object displayed in each knitr code chunk
tabPars	(character, NULL) Table parameters passed to <a href="#">showObj</a> , excluding the 1st three parameters. For example, "fontSize=12, theme='plain'".

## Details

This function creates a Rmd file by appending knitr code chunks to the existing [rmd.template](#).

The first appended code chunk is for setting up: loading libraries in `libs` and [rmdTable](#), and reading `outputListFn` into a data.frame. The following code chunks are one for each row in `outputListFn`, ordered by `rmdInd` and filtered by `eval` in [saveOutput](#). [showObj](#) displays of the object (figure or table) in each row. By default, only its top three parameters are included in the code chunk; additional parameters for table display should be specified through `tabPars`.

Lines containing R markdown section header in their first cell can be inserted into the Excel file pointed by `outputListFn`. Here is an example where `.` represents empty space and `...` the remaining cells.

```
oPath rdsFileName oFileName caption rmdInd eval ...

# Method . . . 1 TRUE .

/path tab1.rds tab1.csv table1 2 TRUE ...
```

When the Excel file contains many lines, adding those section headers make it easier to see the structure of the R markdown file, and the tables and figures at each layer. `createRmd` writes those section headers directly into the produced R markdown file.

The final Rmd file should be edited, e.g. the `isDocx`, `outputFileName`, `title`, `author` and optionally `wordTemplateFileName` for a customized Word template, before knitting.

## Value

none

## See Also

[OUTPUTS](#), [rmd.template](#)

## Examples

```
library(wfr)
library(ggplot2)

df1=data.frame(A=c("a","a","b3"),
               B=c("b1","b2","b3"),
               C1=1:3,C2=2:4)

tmpDir = tempdir()
ofn = file.path(tmpDir, "tab.1.csv")
saveOutput(df1, oFileName = ofn, caption="this is a testing table")

ofn = file.path(tmpDir, "fig.1.png")
saveOutput(qplot(1:10,1:10), oFileName=ofn, caption="this is a testing plot")

exFn=file.path(tmpDir, "all.outputs.xlsx")
rmdFn=file.path(tmpDir, "all.outputs.Rmd")
writeExcel(exFn)
createRmd(rmdFn,exFn, tabPars="fontSize=12,theme='plain'")
```

---

<code>is.empty</code>	<i>Determine if an object is empty</i>
-----------------------	--

---

### Description

Determine if an object is zero length list or factor, or a vector full of NULL, space, dot, NA, Inf, or NaN. It returns a **single** boolean value.

### Usage

```
is.empty(x)
```

### Arguments

`x` (any object)

### Value

(boolean) a single value

### Examples

```
is.empty(c(' ', '.'))
is.empty(list(NULL))
```

---

<code>loadLibs</code>	<i>Load libraries into working space</i>
-----------------------	--

---

### Description

It loads libraries successfully in certain scenario where `require/library` fails.

### Usage

```
loadLibs(failedPackages)
```

### Arguments

`failedPackages` (character vector) names of libraries to load

### Details

This function loads libraries into the working space. It iterates over each path in `.libPaths()` using `require` until a library is loaded successfully. In comparison, `require(x)` for a library `x` fails if `x` is present in multiple search paths but the 1st presence causes loading error, e.g. due to dependency or other issues.

It raises error if not all libraries are loaded successfully.

**Value**

none

**Examples**

```
library(wfr)
loadLibs(c("ggplot2", "flextable"))
```

---

nDecimal

*Return the number of decimal points of a number*

---

**Description**

Trailing zeros are not counted.

**Usage**

```
nDecimal(x)
```

**Arguments**

x (numeric or integer vector) a number vector

**Value**

(integer vector) numbers of decimal points, excluding trailing zeros, or NA if x is NA

**Note**

This is an internal function used in [num2formattedStr](#)

**See Also**

[num2formattedStr](#)

**Examples**

```
library(wfr)
nDecimal(c(1.010, 0.3, NA))
```

---

num2formattedStr	<i>Format a vector of numbers based on their distribution</i>
------------------	---

---

## Description

Format a vector of numbers based on their distribution, removing non-informative digits.

## Usage

```
isVecNumeric(v)
num2formattedStr(v)
```

## Arguments

v (vector) a vector

## Details

isVecNumeric determines whether a vector, numeric or not, contains numbers only. This function is used because often numeric columns are accidentally formatted as character or factor classes. If TRUE from isVecNumeric, num2formattedStr makes decision on the following four aspects of formatting based on the min, median, max values of the vector, so that the formatted numbers carry enough information and are in a length less than 10 characters.

- number of significant digits
- number of decimal points
- whether to apply 1000 separator ','
- whether to apply scientific notation

## Value

(boolean) by isVecNumeric

(character vector) formatted numbers in character by num2formattedStr

## Examples

```
library(wfr)
v1=c(-1032.789, 389.4789, 78.00)
num2formattedStr(v1)
v1=c(3.00, 8.00, -10.000)
num2formattedStr(v1)
v1=c(-0.1289, 0.0489, 0.0003765)
num2formattedStr(v1)
v1=c(-0.1289, 0.0489, 0.03765, NA, Inf, -Inf, NaN)
num2formattedStr(v1)
v1=c(-0.1289, 0.0489, 0.03765, NA, '', Inf, -Inf, NaN)
class(v1)
num2formattedStr(v1)
```

rmdTable

*Construct a table***Description**

Construct a table in html, pdf, or word document, format the numeric columns, and automatically setting column widths.

**Usage**

```
rmdTable(dataDf, header = list(colnames(dataDf)), footer = NULL,
         colWidths = NULL, fontSize = 11, caption = NULL,
         rowHeaderInd = NULL, isDocx = TRUE, nRowScroll = 20,
         nRowDisplay = 200, maxTableWidth = 7,
         theme = c("zebra", "box", "booktabs", "vanilla", "tron", "vader"),
         char2space=NULL, splitCamelCase=FALSE,
         footerFontSize=9, minFontSize=9,...)

myFlexTable(dataDf,header=list(colnames(dataDf)), footer = NULL,
            colWidths = NULL, fontSize = 11, caption = NULL,
            rowHeaderInd = NULL, mergeBodyColumn = TRUE, maxTableWidth = 7,
            theme = c("zebra", "box", "booktabs", "vanilla", "tron", "vader"),
            char2space=NULL, splitCamelCase=FALSE,
            footerFontSize=9, minFontSize=9,...)

myKable(dataDf,header = list(colnames(dataDf)), footer = NULL,
        caption = NULL, rowHeaderInd = NULL, nRowScroll = 20,
        theme = c("zebra", "box", "vanilla"),...)

setFlexTableFontSize(ft,fontSize,footerFontSize=9)
```

**Arguments**

dataDf	(data.frame or matrix) the content of the table to be displayed
header	(character or list of character vectors, list(colnames(dataDf))) <ul style="list-style-type: none"> <li>• list: each vector is the column title. The last vector replaces the colnames(dataDf). Neighboring cells of identical content in the header will be merged into one cell. For myKable, merge is only horizontal and not on the last row.</li> <li>• character: use 'l' to separate cell, and 'll' to separate rows. It is converted to list using <a href="#">str2list</a></li> </ul>
footer	(character or list of character vectors, NULL) <ul style="list-style-type: none"> <li>• list: each vector contains the following items:               <ol style="list-style-type: none"> <li>1. (optional, character). The cell content which the footnote refers to.</li> <li>2. (character). The content of the footer. use \$~i\$ and \$^i\$ to represent the sub/super-script of i</li> </ol> </li> </ul>

	<ol style="list-style-type: none"> <li>(optional, character). The super-script of #1</li> <li>(optional, character, "header" or "body") the portion of the table where #1 is to be searched. If "body", only the columns in rowHeaderInd are searched.</li> <li>character: use 'l' to separate cell, and 'll' to separate rows. It is converted to list using <a href="#">str2list</a></li> </ol>
colWidths	(character or numeric vector, NULL). For myFlexTable only, the column widths, a numerical vector of the length of ncol(dataDf). Unit is inch. It can also be a character string where the numbers are separated by ',', e.g. "2,1,1,1".
fontSize	(integer, 10) For myFlexTable only, the font size of the header and body. Font size of footer is fontSize - 2.
caption	(character, NULL) the caption of the table.
rowHeaderInd	(integer, NULL). Row headers are the columns in the left of table body serving as headers for rows in the table body. rowHeaderInd are the last index of those columns, so the column indices of row headers is 1:rowHeaderInd. If rowHeaderInd is specified, the font of row header becomes bold, and neighboring cells of identical content are collapsed. The merge can be both horizontal and vertical in myFlexTable, and only vertical in myKable.
mergeBodyColumn	(boolean, TRUE) if (mergeBodyColumn && rowHeaderInd > 1), neighboring horizontal cells in table body are merged if they are in rows where exists identical neighboring horizontal cells in row header. This flag is to prevent, if set to FALSE, the merging of identical neighboring horizontal cells in table body, when the intention is to limit such merging to row header only.
isDocx	(boolean, TRUE) if TRUE, use myFlexTable; otherwise, myKable
nRowScroll	(integer, 20) For myKable only, the cutoff on number of rows to apply a scroll window.
nRowDisplay	(integer, 200) For myFlexTable only, the cutoff on number of rows to display. If there are more than nRowDisplay rows in the table, the caption of the table is appended "(top nRowDisplay rows only)".
theme	(character) The theme of the table.
maxTableWidth, minFontSize	see those parameters in <a href="#">setWidths</a> . For myFlexTable only.
char2space	(character string, NULL) A regular expression. Should the characters represented by this regular expression in the bottom row of column header be changed to space? If so, when the column header is wrapped, the wrapping happens at a space in stead of the middle of a word. For example, setting char2space = '[^A-z0-9]' changes all non-letter and non-digit characters into space.
splitCamelCase	(boolean, FALSE) Should the camel cases in the 1st row of column header be split into separated words? for example, change "youMadeItLOL" into "you Made It LOL". If so, when a column header in camel case is wrapped, the wrapping happens at a space in stead of the middle of a word.
footerFontSize	(integer, 9) For myFlexTable only; the size of footer font.
ft	a flextable object
...	if isDocx, passed to flextable::flextable(), and kableExtra::kable() otherwise.



**Details**

myFlexTable and myKable are wrapper functions of [flextable](#) and [kable](#), and rmdTable is a wrapper of the two wrappers with isDocx TRUE referring to myFlexTable and FALSE to myKable

**Value**

(flextable or kable object)

**Note**

Numeric columns are formatted using [num2formattedStr](#). If a numeric column is not formatted in the displayed table, probably it is because its data type in dataDf is not numeric or integer.

**See Also**

[num2formattedStr](#), [flextable](#), [kable](#)

**Examples**

```
library(wfr)
df1=data.frame(A=c("a","a","b3"),
               B=c("b1","b2","b3"),
               C1=c(1001.123,58.04,32.01),
               C2=c(-0.00321, 0.0121, 0.325))

header=list(c('A','A','C','C'),
            c('A','A','C1','C2'))

footer=list(c("A","Arkansas$~ref$",'1','header'),
            c("C1","Kansas$^ref$",'x','header'),
            c('a',"Arizona",'2','body'))

rmdTable(df1,header = header,
         rowHeaderInd = 2,
         footer = footer,
         caption = "my first table",
         colWidths = c(2,1,1,1),
         fontSize = 12, isDocx = TRUE)

colWidths = "2,1,1,1"
header = "A | A | C | C || A | A | C1 | C2"
footer = "A|Arkansas$~ref$|1|header
         || C1|Kansas$^ref$|x|header
         || a|Arizona|2|body"

rmdTable(df1,header = header,
         rowHeaderInd = 2,
         footer = footer,
         caption = "my second table",
         colWidths = colWidths,
         fontSize = 12, isDocx = TRUE)
```

---

saveOutput	<i>Save the current R image or the object rds file and add its information to the matrix OUTPUTS</i>
------------	--

---

## Description

Save the current R image or the object rds file and optionally other formats of the object (e.g. csv for tables and png for figures), its caption, and other information related to its appearance in the R markdown file, to the matrix OUTPUTS

## Usage

```
saveOutput(obj=NULL, oFileName, saveWorkspace=FALSE, oPath=getwd(), caption=NA ,
           rmdInd=NA, eval=TRUE, objID=NA, header=NA, footer=NA,
           rowHeaderInd=NA, colWidths=NA, fontSize=11, nRowScroll = 20,
           nRowDisplay = 200, maxTableWidth = 7.2, theme = "zebra",
           numberOutputFiles=TRUE,...)
```

```
ofcmm(num=1)
```

## Arguments

obj	The target object, usually a data.frame or ggplot object. If NULL, oFileName can be used to insert an existing figure file into OUTPUTS, so that the figure file can be auto-included into the Rmd file produced by <a href="#">createRmd</a> function.
oFileName	(character). The file name of the text (e.g. csv) or image (e.g. png) file of the obj to save. Can contain relative or absolute file paths; Use, e.g. ./path1/my.csv, instead of path1/my.csv to include a relative path into the file name. If oFileName contains either a relative or absolute path, the oPath will be replaced by the whole path.
saveWorkspace	(boolean, FALSE) <ul style="list-style-type: none"> <li>FALSE: Save the rds file of the obj only. If oFileName is provided, the rds file name is paste0(ofNamePrefix, '.rds'); otherwise, paste0(sprintf ("%03d", OFCOUNTER), '.rds').</li> <li>TRUE: In addition to the rds file, save the R image of the workspace where obj is generated as paste0(sprintf ("%03d", OFCOUNTER), '.r.image.rdata')</li> </ul>
oPath	(character, getwd()). The output path.
caption	(character, NA). The caption of the target object.
rmdInd	(integer, <a href="#">OFCOUNTER</a> ). The order to display obj in the Rmd file.
eval	(boolean, TRUE). Should the obj be included/evaluated in the Rmd file?
objID	(character, NA, or paste0('tab', OFCOUNTER), or paste0('fig', OFCOUNTER)). The label of the obj in the Rmd code chunk. Its default value depends on the data type of obj: paste0('tab', OFCOUNTER) for data.frame or matrix, paste0('fig', OFCOUNTER) for ggplot, and NA otherwise. If provided, it is

	cleaned by <code>gsub("[^A-Za-z0-9]", "", objID)</code> , and prefixed by <code>tab</code> or <code>fig</code> depending on the type of <code>obj</code> . If it's already present in <code>OUTPUTS[, "objID"]</code> , <code>OFCOUNTER</code> is appended, i.e. <code>objID = paste0(objID, OFCOUNTER)</code> , to make it unique.
header	(character, NA). The header of the <code>obj</code> , if it is displayed as a table in the Rmd file. See parameter <code>header</code> of <code>rmdTable</code> for more details and its character representation.
footer	(character, NA). The footer of the <code>obj</code> , if it is displayed as a table in the Rmd file. See parameter <code>footer</code> of <code>rmdTable</code> for more details and its character representation.
colWidths	(character, NA). The column width of the <code>obj</code> , if it is displayed as a table in the Rmd file. See parameter <code>colWidths</code> of <code>rmdTable</code> for more details and its character representation.
rowHeaderInd, fontSize, nRowScroll, nRowDisplay, maxTableWidth, theme	Parameters for <code>obj</code> , if it is displayed as a table in the Rmd file. See same parameters in <code>rmdTable</code> for more details.
numberOutputFiles	(boolean, TRUE). Should the output files be numbered?
...	parameters passed to <code>write.csv</code> or <code>ggsave</code> to save the <code>obj</code> , if a <code>data.frame</code> or matrix, as text file.
num	the number to deduct from <code>OFCOUNTER</code>

## Details

For the simplicity of coding, two global variables are created for this function:

- `OFCOUNTER` (integer, 1). A global variable to count the number of outputs, initial value is 1.
- `OUTPUTS` (character matrix, NULL). A global variable to record the information of the current r image file and `obj`. At the 1st run of `saveOutput`, it is assigned to be a character matrix of following columns:
  - `rImageName` The name of saved R image or rds file, depending on the flag `saveWorkspace`.
  - all other parameters of `saveOutput` except `obj`, and values assigned by them.

`saveOutput` does the following:

1. Save the current R image as `paste0(sprintf("%03d", OFCOUNTER), '.r.image.rdata')` or the object rds file in the `oPath` directory.
2. If `oFileName` is provided, save `obj` as `paste0(sprintf("%03d", OFCOUNTER), '.', oFileName)` in the `oPath` directory, using either `utils::write.csv` or `ggplot2::ggsave` depending on the data type of `obj`
3. Create the matrix `OUTPUTS` if it is NULL, and assign the values of all other parameters to corresponding columns in `OUTPUTS`
4. Increment `OFCOUNTER` by 1

`ofcmm` deducts `OFCOUNTER` by `num`. It can be used to remove the information appended to `OUTPUTS` from running `saveOutput`.

**Value**

none

**See Also**

[OUTPUTS](#), [OFCOUNTER](#)

**Examples**

```
library(wfr)
library(ggplot2)

print(OFCOUNTER)
print(OUTPUTS)

df1=data.frame(A=c("a","a","b3"),
               B=c("b1","b2","b3"),
               C1=1:3,C2=2:4)

tmpDir = tempdir()
ofn = file.path(tmpDir, "tab.1.csv")
saveOutput(df1, oFileName=ofn, caption="this is a wrong caption")
print(OFCOUNTER)
print(OUTPUTS)
ofcmm()
print(OFCOUNTER)
saveOutput(df1, oFileName=ofn, caption="this is the correct caption")
print(OUTPUTS)

print(OFCOUNTER)
print(OUTPUTS)

ofn = file.path(tmpDir, "fig.1.png")
saveOutput(qplot(1:10,1:10), oFileName=ofn, caption="this is a testing plot")

print(OFCOUNTER)
print(OUTPUTS)
```

---

saveWfrInfo

*save and read back the values of OFCOUNTER and OUTPUTS*

---

**Description**

these functions are used in R scripts that are separate and run in order, using the SAME [OFCOUNTER](#) and [OUTPUTS](#)

**Usage**

```
saveWfrInfo(rdsFileName)
restoreWfrInfo(rdsFileName)
```

**Arguments**

rdsFileName      The name of the rds file keeping the values of [OFCOUNTER](#) and [OUTPUTS](#)

**Details**

The rds file is a list(ofc=OFCOUNTER, outp=OUTPUTS)

**Value**

(list) by restoreWfrInfo

**See Also**

[OFCOUNTER](#) and [OUTPUTS](#)

---

setHtmlHeaderProperty    *Set the formatting properties of Title, Author, and Date in Rmd file*

---

**Description**

Font size in px, font family, and color can be set. Alignment is center.

**Usage**

```
setHtmlHeaderProperty(
  titleFontSize = NULL, titleFontFamily = NULL, titleColor = NULL,
  authorFontSize = NULL, authorFontFamily = NULL, authorColor = NULL,
  dateFontSize = NULL, dateFontFamily = NULL, dateColor = NULL,
  pageWidth=NULL, tableWidth = 100)
```

**Arguments**

titleFontSize  
titleFontFamily

titleColor  
authorFontSize  
authorFontFamily

authorColor  
dateFontSize

dateFontFamily  
 dateColor  
 pageWidth           (numeric) the page width in px  
 tableWidth         (numeric) the percentage of table width over page width

## References

[stackoverflow link](#)

## Examples

```
setHtmlHeaderProperty(titleFontSize=18,
  titleFontFamily='"Times New Roman", Times, serif',
  titleColor='DarkBlue')
```

---

setNsmall	<i>Set the nSmall of a number, and convert it to character</i>
-----------	--

---

## Description

set the nsmall, i.e. number of decimal points, of a number, and convert it to character; 1000 separator ',' is added if  $\max(\text{abs}(v)) > 999$ .

## Usage

```
setNsmall(v, nSmall)
```

## Arguments

v                   (numeric or integer vector) a number vector  
 nSmall             (integer) nsmall, i.e. number of decimal points, of a number ( $0 \leq \text{nsmall} \leq 20$ )

## Value

(character vector) formatted numbers in character

## Note

This is an internal function used in [num2formattedStr](#)

## See Also

[format](#), [num2formattedStr](#)

## Examples

```
setNsmall(c(1.003, 2.1), 2)
```

---

setWidths	<i>Set column widths of a table</i>
-----------	-------------------------------------

---

### Description

Properly setting column widths given the maximum width of the table, for non-html output only.

### Usage

```
setWidths(x, header1, maxTableWidth = 7, rowHeaderInd = NULL,
          minFontSize=9, nRowPerRowHeader=NULL)
breakRatio(aStr)
```

### Arguments

x	A flextable object
header1	(character vector) the first row of the column header
maxTableWidth	(numeric, 7.0) the maximum width of the table in inch. The default 7.0 corresponds to "PAGE LAYOUT" > "size" > "letter"; "Margins" > "Moderate" in MS Word.
rowHeaderInd	(integer, NULL) See rowHeaderInd in <a href="#">rmdTable</a> . Because row headers are bold font, their lengths in inch is increased by 10%.
minFontSize	(integer, 9) The minimum font size in table body.
nRowPerRowHeader	(integer vector, NULL) The average number of rows spanned by each row header under a row header index. So it's a vector of length rowHeaderInd.
aStr	(a character string) The string is the content of a header cell.

### Details

setWidths sets the width of each column of a table to fit maxTableWidth. Here is a brief description of its algorithm:

```
let HBWidths = mapply(max, wHeader, wBody)
```

1. if(sum(HBWidths) <= maxTableWidth), set HBWidths to be the final table widths.
2. Else if(sum(HBWidths)/maxTableWidth < 1.08), reducing font size by 1, set HBWidths\*0.92 to be the final table widths.
3. Else
  - (a) For columns where the header is longer than the body and the header is a single word, wrap the header at a non-letter character closest to the middle of the header.
  - (b) If the table still doesn't fit, further wrapping the columns of row headers if any, and if the cells under those columns span vertically across multiple cells.
  - (c) If the table still doesn't fit, reduce font size up to minFontSize
  - (d) If the table still doesn't fit, wrap the table body, starting from the longest table columns, until the table fits.

breakRatio computes the wrapping point of a header.

Value

(numeric) by breakRatio, the ratio of the original length after wrapping. (list) by setWidths. The list contains 1)widths: a numeric vector of column widths and 2)fs: the new font size.

---

showObj	<i>Display an object in knitr code chunk</i>
---------	--

---

Description

Display an object in knitr code chunk and set the caption with cross reference.

Usage

```
showObj(oDf, objID=NULL, isDocx=FALSE, ...)
```

Arguments

- oDf (data.frame or matrix) Refers to [OUTPUTS](#), the table containing the information of saved objects.
- isDocx (boolean, FALSE) is the Rmd output word\_document2?
- objID (character, NULL) The objID of the object in oDf. If NULL, it is assigned the label of the code chunk. But when running the code chunk alone, e.g. for debugging, the label is not accessible and must be explicitly supplied.
- ... parameters passed to [rmdTable](#) to define table properties. Unsupplied properties are taken from the corresponding columns on the row identified by objID, if those cells are not NA.

Value

An object identified by objID, can be a ggplot2 or a table (flextable if isDocx T; kable otherwise)

See Also

[rmdTable](#) [OUTPUTS](#)

Examples

```
library(wfr)
library(ggplot2)
isDocx=TRUE

df1=data.frame(A=c("a", "a", "b3"),
               B=c("b1", "b2", "b3"),
               C1=1:3, C2=2:4)
colWidth = "2,1,1,1"
header = "A | A | C | C || A | A | C1 | C2"
footer = "A|Arkansas$~ref$|1|header"
```



```

|| C1|Kansas$^ref$|x|header
|| a|Arizona|2|body"

tmpDir = tempdir()
ofn = file.path(tmpDir, "t1.csv")
saveOutput(df1,ofn,caption = "1st testing table",header = header,
           footer = footer, colWidth = colWidth,fontSize = 12 )
saveOutput(qplot(1:10,1:10),oPath=tmpDir, caption = "1st testing fig")

ofn = file.path(tmpDir, "all.outputs.csv")
write.csv(OUTPUTS,ofn)

oDf=read.csv(ofn,stringsAsFactors = FALSE)
showObj(oDf,isDocx,objID = "tab1",rowHeaderInd=2)
showObj(oDf,objID = "fig2")

```

str2list

*Convert a character string to list of character vectors***Description**

In the character string, 'l' separates items inside a vector; 'll' separates vectors.

**Usage**

```
str2list(x)
```

**Arguments**

x (character) a character string

**Value**

(list of character vectors)

**Note**

This is an internal function used in [rmdTable](#) so that header and footer can accept character string.

**See Also**

[rmdTable](#)

**Examples**

```
str2list("A | A | C | C || A | A | C1 | C2")
```

---

table.cross.ref	<i>Create the cross-reference string, and the caption of a table with cross-reference</i>
-----------------	---

---

## Description

tRef creates the cross-reference string, and tCap creates the caption.

## Usage

```
tRef(label, isDocx)
tCap(cap, label, isDocx)
```

## Arguments

cap	(character) the original caption string
label	(character) the label of the table in the knitr code chunk
isDocx	(boolean) is the output format of the Rmd file word_document2?

## Details

Because the current version of [flextable](#) v0.5.5 does not work with [bookdown](#) v0.12 in automatically producing table cross-reference, these functions are a workaround.

## Value

(character) If isDocx is FALSE, tRef returns Table \@ref(tab:label) and tCap simply returns cap; otherwise, tRef returns Table. x and tCap returns Table. x cap, where x is the ordered index of the table.

## Examples

```
library(wfr)
tCap("first table", "tab1", FALSE)
tCap("first table", "tab1", TRUE)
tCap("second table", "tab2", TRUE)
tCap("first table", "tab1", TRUE)

tRef("tab2", TRUE)
tRef("tab1", TRUE)
```

---

template.files	<i>A R markdown and a Word template file</i>
----------------	--

---

**Description**

A R markdown template file that can produce 'MS Word', html, or pdf file based on a flag, and a 'MS Word' template file on which the style of Word output file is based on.

**Details**

This R markdown template can produce 'MS Word', html, or pdf file based on the flag oFormat on line 3. The style of 'MS Word' format is based on the 'MS Word' template file. They can be accessed by

- `system.file("extdata", "rmd.template.Rmd", package = "wfr")`
- `system.file("extdata", "word.template.for.Rmd.docx", package = "wfr")`

Function `createRmd` appends knitr code chunks to this Rmd template to form the final Rmd file.

---

writeExcel	<i>Save the <a href="#">OUTPUTS</a> to an Excel file</i>
------------	--

---

**Description**

While writing the [OUTPUTS](#) to an Excel file, column oFileName is embedded with URL `file.path(df1[, "oPath"], df1[, "o`

**Usage**

```
writeExcel(fileName, ...)
```

**Arguments**

- |          |  |
|----------|--|
| fileName | (character) The name of the Excel file with extension 'xlsx'. Can contain relative or absolute file paths. |
| ...      | parameters passed to <code>write.xlsx</code>   |

**Details**

It is not a general function to save a data.frame or matrix to an Excel file. And there is no append mode; the target file will be covered if already present.

**Value**

none

**See Also**[write.xlsx](#)**Examples**

```

library(wfr)
library(openxlsx)
library(ggplot2)
df1=data.frame(A=c("a","a","b3"),
               B=c("b1","b2","b3"),
               C1=1:3,C2=2:4)
colWidth = "2,1,1,1"
header = "A | A | C | C || A | A | C1 | C2"
footer = "A|Arkansas$~ref$|1|header
|| C1|Kansas$^ref$|x|header
|| a|Arizona|2|body"

tmpDir = tempdir()
ofn = file.path(tmpDir, "tab.1.csv")
saveOutput(df1,ofn,caption = "1st testing table",header = header,
           footer = footer,colWidth = colWidth,fontSize = 12 )

saveOutput(qplot(1:10,1:10),oPath=tmpDir,caption = "1st testing fig")

fn1=file.path(tmpDir, "output.xlsx")
writeExcel(fn1)
df2=read.xlsx(fn1)

```

# Index

- \* **MS Word**
  - template.files, 19
- \* **empty**
  - is.empty, 4
- \* **excel**
  - writeExcel, 19
- \* **format**
  - nDecimal, 5
  - num2formattedStr, 6
  - setNsmall, 14
- \* **html**
  - setHtmlHeaderProperty, 13
- \* **library**
  - loadLibs, 4
- \* **list**
  - str2list, 17
- \* **number**
  - nDecimal, 5
  - num2formattedStr, 6
  - setNsmall, 14
- \* **rmd**
  - createRmd, 2
  - rmdTable, 7
  - showObj, 16
  - table.cross.ref, 18
  - template.files, 19
- \* **string**
  - str2list, 17
- \* **table**
  - rmdTable, 7
  - saveOutput, 10
  - setWidths, 15
  - table.cross.ref, 18
- \* **template**
  - createRmd, 2
  - template.files, 19
- \* **widths**
  - setWidths, 15
- \* **width**
  - setWidths, 15
- breakRatio (setWidths), 15
- createRmd, 2, 10, 19
- flextable, 9, 18
- format, 14
- ggsave, 11
- Global.Variable (saveOutput), 10
- is.empty, 4
- isVecNumeric (num2formattedStr), 6
- kable, 9
- loadLibs, 4
- myFlexTable (rmdTable), 7
- myKable (rmdTable), 7
- nDecimal, 5
- num2formattedStr, 5, 6, 9, 14
- ofcmm (saveOutput), 10
- OFCOUNTER, 10–13
- OFCOUNTER (saveOutput), 10
- OUTPUTS, 2, 3, 12, 13, 16, 19
- OUTPUTS (saveOutput), 10
- restoreWfrInfo (saveWfrInfo), 12
- rmd.template, 2, 3
- rmd.template (template.files), 19
- rmdTable, 2, 7, 11, 15–17
- saveOutput, 2, 10
- saveWfrInfo, 12
- setFlexTableFontSize (rmdTable), 7
- setHtmlHeaderProperty, 13
- setNsmall, 14
- setWidths, 8, 15

showObj, [2](#), [16](#)  
str2list, [7](#), [8](#), [17](#)  
  
table.cross.ref, [18](#)  
tCap (table.cross.ref), [18](#)  
template.files, [19](#)  
tRef (table.cross.ref), [18](#)  
  
word.template (template.files), [19](#)  
write.csv, [11](#)  
write.xlsx, [19](#), [20](#)  
writeExcel, [19](#)