# Package 'wfr'

October 9, 2019

**Type** Package

**Title** A Workflow to Assist the Creation of Documents Containing Many Tables and Figures

**Version** 0.3.1

**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 output from the data analysis R script, even including format info if the output is a table. 2) It can automatically create the R markdown file to produce either DOCX, 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

**RoxygenNote** 6.1.1.9000

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

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

## R topics documented:

---

createRmd                              *Create a R markdown file*

---

### 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

outputListFn (character) The file name of the text file containing the table from OUTPUTS

rmdTemplateFn

                (character, system.file("extdata", "rmd.template.Rmd",package = "wfr"))
                The file name of the rmd.template

libs           (character, "ggplot2") The names of the libraries needed to execute the object, which is actually OBJ2, displayed in each knitr code chunk

tabPars        (character, NULL) Table parameters passed to showObj, 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.

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

### 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)
## Not run:
    saveOutput(df1, ofName="tab.1.csv", caption="this is a testing table")
    saveOutput(qplot(1:10,1:10), ofName="fig.1.png", caption="this is a testing plot"
    OUTPUTS[,"oPath"] = getwd()
    exFn="all.outputs.xlsx"
    writeExcel(OUTPUTS,exFn)

    createRmd("all.outputs.Rmd",exFn, tabPars="fontSize=12,theme='plain'")

## End(Not run)
```

---

| is.empty | *Determine if an object is empty* |
|---|---|

---

**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))
```

---

loadLibs                          *Load libraries into working space*

---

**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**

**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 *Format a vector of numbers based on their distribution*

---

## 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 accidently 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 sinificant 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 = T, nRowScroll = 20,
        nRowDisplay = 200, maxTableWidth = 7,
        theme = c("zebra","box","booktabs","vanilla","tron","vader"),
        underLine2Space=T, splitCamelCase=F,
        footerFontSize=9, minFontSize=9)

myFlexTable(dataDf,header=list(colnames(dataDf)), footer = NULL,
        colWidths = NULL, fontSize = 11, caption = NULL,
        rowHeaderInd = NULL, mergeBodyColumn = T, maxTableWidth = 7,
        theme = c("zebra","box","booktabs","vanilla","tron","vader"),
        underLine2Space=T, splitCamelCase=F,
        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))) |

- 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.

> • character: use '|' to separate cell, and '||' to separate rows. It is converted to list using str2list

footer        (character or list of character vectors, NULL)

> • list: each vector contains the following items:
>   1. (optional, character). The cell content which the footnote refers to.
>   2. (character). The content of the footer. use $\sim i$ and $\hat{}i$ to represent the sub/super-script of `i`
>   3. (optional, character). The super-script of #1
>   4. (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.
> • character: use '|' to separate cell, and '||' to separate rows. It is converted to list using `str2list`

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, T) `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 `F`, the merging of identical neighboring horizontal cells in table body, when the intention is to limit such merging to row header only.

isDocx        (boolean, T) if `T`, use `myFlexTable`; otherwise, `myKable`

nRowScroll    (integer, 20) For `myKable` only, the cutoff on nuber of rows to apply a scroll window.

nRowDisplay   (integer, 200) For `myFlexTable` only, the cutoff on nuber 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 setWidths. For `myFlexTable` only.

underLine2Space

> (boolean, T) Should the underlines '_' in the 1st row of column header be changed to space? If so, when a column header containing '_' is wrapped, the wrapping happens at a space in stead of the middle of a word.

splitCamelCase

> (boolean, F) 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
```

## Details

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

## Value

(flextable or kable object)

## Note

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

## See Also

[num2formattedStr](num2formattedStr), [flextable](flextable), [kable](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 | *Save the current R image or the object rds file and add its information to the matrix OUTPUTS* |
|---|---|

---

### 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 inforamtion related to its appearance in the R markdown file, to the matrix OUTPUTS

### Usage

```
saveOutput(obj,ofName=NA, saveWorkspace=F, oPath='.', caption=NA ,
           rmdInd=NA, eval=T,objID=NA, header=NA, footer=NA,
           rowHeaderInd=NA, colWidths=NA, fontSize=11, nRowScroll = 20,
           nRowDisplay = 200, maxTableWidth = 7.2, theme = "zebra",...)
```

### Arguments

| | |
|---|---|
| obj | The target object, usually a data.frame or ggplot object. After passing in, its value assigns to the global variable OBJ2; therefore, later when the R image is loaded, this target object can be just referred as OBJ2. |
| ofName | (character, NA). The file name of the text (e.g. csv) or image (e.g. png) file of the obj to save. |
| saveWorkspace | (boolean, F)<br><br>• F: Save the rds file of the obj only. If ofName is provided, the rds file name is paste0(ofNamePrefix,'.rds'); otherwise, paste0(sprintf("%03d", OFCOUNTER),'.rds').<br>• T: Save the R image of the workspace where obj is generated as paste0(sprintf("%03d", OFCOUNTER),'.r.image.rdata') |
| oPath | (character, '.'). The output path. |
| caption | (character, NA). The caption of the target object. |
| rmdInd | (integer, OFCOUNTER). The order to display obj in the Rmd file. |
| eval | (boolean, T). 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 gsub("[^A-Za-z0-9]","",objID), and prefixed by tab or fig depending on the type of obj. If it's already present in OUTPUTS[,"objID"], OFCOUNTER is appended, i.e. objID = paste0(objID,OFCOUNTER), to make it unique. |
| header | (character, NA). The header of the obj, if it is displayed as a table in the Rmd file. See parameter header of rmdTable for more details and its character representation. |

footer            (character, NA). The footer of the `obj`, if it is displayed as a table in the Rmd
                  file. See parameter `footer` of `rmdTable` for more details and its character
                  representation.

colWidths         (character, NA). The column width of the `obj`, if it is displayed as a table in the
                  Rmd file. See parameter `colWidths` of `rmdTable` for more details and its
                  character representation.

rowHeaderInd, fontSize, nRowScroll, nRowDisplay, maxTableWidth, theme
                  Parameters for `obj`, if it is displayed as a table in the Rmd file. See same
                  parameters in `rmdTable` for more details.

...               parameters passed to `write.csv` to save the `obj`, if a data.frame or matrix, as
                  text file.

### Details

For the simpilicity of coding, three global variables are created for this function:

- OFCOUNTER (integer,1). A global variable to count the number of outputs, initial value is 1.
- OBJ2 (any data type, NULL). A global variable to hold the value of `obj` inside `saveOutput`
- 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 `ofName` is provided, save `obj` as `paste0(sprintf("%03d", OFCOUNTER),'.',ofName)` 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. Assign `obj` to the global variable OBJ2, so that when `obj` is passed in as a subset of a data.frame, e.g. `df1[1:5,]`, that subset can be easily referred using OBJ2.
5. Increment OFCOUNTER by 1

### Value

### See Also

OUTPUTS, OFCOUNTER, OBJ2

### Examples

```
library(wfr)
library(ggplot2)

print(OFCOUNTER)
print(OUTPUTS)
```

```
print(OBJ2)

df1=data.frame(A=c("a","a","b3"),
               B=c("b1","b2","b3"),
               C1=1:3,C2=2:4)
## Not run:
  saveOutput(df1, ofName="tab.1.csv", caption="this is a testing table")

## End(Not run)

print(OFCOUNTER)
print(OUTPUTS)
print(OBJ2)

## Not run:
  saveOutput(qplot(1:10,1:10), ofName="fig.1.png", caption="this is a testing plot")

## End(Not run)

print(OFCOUNTER)
print(OUTPUTS)
print(OBJ2)
```

---

```
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)
```

### Arguments

```
titleFontSize

titleFontFamily

titleColor
authorFontSize

authorFontFamily

authorColor
dateFontSize
dateFontFamily

dateColor
```

## References

stackoverflow link

## Examples

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

---

setNsmall                        *Set the nSmall of a number, and convert it to character*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| v | (numeric or integer vector) a number vector |
| nSmall | (integer) nsmall, i.e. nubmer of decimal points, of a number (0 <= nsmall <= 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 | *Set column widths of a table*

---

### 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 [rmdTable](). 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, staring 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 | *Display an object in knitr code chunk* |
| --- | --- |

---

## Description

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

## Usage

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

## Arguments

| | |
| --- | --- |
| `oDf` | (data.frame or matrix) Refers to OUTPUTS, the table containing the information of saved objects. |
| `isDocx` | (boolean, F) 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 identifed 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"
```

```
## Not run:
    saveOutput(df1,"t1.csv",caption = "1st testing table",header = header,
                footer = footer, colWidth = colWidth,fontSize = 12 )
    saveOutput(qplot(1:10,1:10),caption = "1st testing fig")

    OUTPUTS[,"oPath"]=getwd()
    write.csv(OUTPUTS,"all.outputs.csv")

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

## End(Not run)
```

---

str2list                    *Convert a character string to list of character vectors*

---

### Description

In the character string, '|' separates items inside a vector; '||' 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          *Create the cross-reference string, and the caption of a table with cross-*
                         *reference*

---

### 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 auto-matically producing table cross-reference, these functions are a workaround.

### Value

(character) If `isDocx` is F, `tRef` returns `Table \@ref(tab:label)` and `tCap` simply re-turns `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 *A R markdown and a Word template file*

---

### Description

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

### Details

This R markdown template can produce docx, html, or pdf file based on the flag `oFormat` on line 3. The style of docx Word format is based on the 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 *Save the* `OUTPUTS` *to an Excel file*

---

### Description

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

### Usage

```
writeExcel(df1, fileName, ...)
```

### Arguments

| | |
|---|---|
| `df1` | (data frame or matrix) Used to hold the value of `OUTPUTS` |
| `fileName` | (character) The name of the Excel file with extension 'xlsx' |
| `...` | parameters passed to `write.xlsx` |

### 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

### 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"

## Not run:
  saveOutput(df1,"t1.csv",caption = "1st testing table",header = header,
             footer = footer,colWidth = colWidth,fontSize = 12 )
  saveOutput(qplot(1:10,1:10),caption = "1st testing fig")
  saveOutput(df1,"t1.csv",caption = "1st testing table",header = header,
             footer = footer,colWidth = colWidth,fontSize = 12 )

  OUTPUTS[,"oPath"]=getwd()
  fn1="output.xlsx"
  writeExcel(OUTPUTS,fn1)
  df2=read.xlsx(fn1)

## End(Not run)
```

# Index