Package 'rmdHelpers'

July 11, 2024

Type Package

Title Helper Functions for Rmd Documents

Version 1.3.1

Date 2024-07-11

Author Mark Peterson

Maintainer Mark Peterson <mark.phillip.peterson@gmail.com>

Description

A series of functions to aid in repeated tasks for Rmd documents. All details are to my personal preference, though I am happy to add flexibility if there are use cases I am missing. I will continue updating with new functions as I add utility functions for myself.

License GPL

Depends knitr, dplyr

NeedsCompilation no

Repository CRAN

Date/Publication 2024-07-11 21:00:02 UTC

Contents

`rmdHelpers-package` *Helper Functions for Rmd Documents*

Description

A series of functions to aid in repeated tasks for Rmd documents. All details are to my personal preference, though I am happy to add flexibility if there are use cases I am missing. I will continue updating with new functions as I add utility functions for myself.

Details

The DESCRIPTION file:

Package:	rmdHelpers
Type:	Package
Title:	Helper Functions for Rmd Documents
Version:	1.3.1
Date:	2024-07-11
Author:	Mark Peterson
Maintainer:	Mark Peterson <mark.phillip.peterson@gmail.com>
Description:	A series of functions to aid in repeated tasks for Rmd documents. All details are to my personal preference, tho
License:	GPL
Depends:	knitr, dplyr

Index of help topics:

Basic functions that I use in multiple Rmd documents

Author(s)

Mark Peterson

Maintainer: Mark Peterson <mark.phillip.peterson@gmail.com>

`formatEffectTable` *Format an effect table*

Description

Performs general formatting of effect tables from ‘lm‘ suitable for basic printing. This includes merging estimates and confidence intervals, rounding, and optionally improving the display of factor and logical columns (using a colon and space, instead of just concatenating them).

Usage

```
formatEffectTable(object
                  , level = 0.95
                  , estDigits = 2
                  , pDigits = 4
                  , cleanFactors = TRUE)
```

Arguments

`object`	A fitted model object from ‘lm‘
`level`	The confidence level to be returned
`estDigits`	The number of digits to be displayed for the estimate and confidence thresholds
`pDigits`	The number of digits to be displayed for the p-values
`cleanFactors`	Logical. Should the parameter names for factors be cleaned by separating the parameter from the value with a colon and a space, or not. (This option may lead to issues with merging if set to ‘TRUE‘.)

Value

A data.frame formatted ready to be displayed (e.g. by ‘kable‘)

Note

Note that "NA" values are silently dropped (they are not returned by ‘summary.lm‘ in the coefficients table).

Author(s)

Mark Peterson

Examples

```
irisMod <- lm(Sepal.Length ~ ., data = iris)
formatEffectTable(irisMod)
```

formatP *Format p-value*

Description

A wrapper to sensibly control the printing of p-values because I was frustrated with playing with round. Calls format instead now.

Usage

```
formatP(p, digits = 3, scientific = FALSE, ...)
```

Arguments

p	Numeric vector of values to be displayed
digits	Numeric of length one giving the number of digits to display. Note that if p is longer than 1, all returned values will be to the same level of precision.
scientific	Logical, should scientific notation be used?
...	Further arguments passed to format

Value

Character vector of the p-value(s) formatted

Author(s)

Mark Peterson

See Also

format

Examples

```
formatP(0.049865465646)

formatP(0.00000013212354)

formatP(c(0.01564643131,.0003456463131, .45643131564), 2)
```

myFrac *Print Fraction for markdown*

Description

Converts fractions for inline rendering.

Usage

```
myFrac(num, denom, format = "markdown")
```

Arguments

num	Either a vector of values for the numerators, or, if denom is NULL, fractions using "/" as the separator (allows passing fractions directly, as from fractions)
denom	Vector of values for the denominators, or NULL (the default) if num is already formatted fractions.
format	Character vector of length one giving the format of the output. Default of "markdown" uses super script for the numerator, a slash, then subscript for the denominator. Alternatively, "latex" uses \frac and surrounds the fraction with $ to induce LaTeX conversion (via MathJax for html output)

Value

Character vector of the fractions

Note

If passing fractions, the function assumes that there is exactly one "/" in each fraction, and will return "NA" for the denominator if none is present or truncate the fraction if more than one "/" is included.

Author(s)

Mark Peterson

See Also

fractions

Examples

```
myFrac(3,4)

myFrac(1:3,4:6)

myFrac(1:3,4:6, "latex")
```

```
myFrac(letters[1:5], LETTERS[1:5])

myFrac( c("1/2","3/4", "9856/5646") )

myFrac( c("1/2","3/4", "9856/5646"), format = "latex" )
```

---

myKable *Wrapper for kable*

Description

A small wrapper for the knitr kable function to allow automated bolding of row and/or column names. Additional functionality may be added/

Usage

```
myKable(x, row.names = NA, boldRowNames = TRUE, boldColNames = TRUE, ...)
```

Arguments

x	Table or matrix to be passed to kable
row.names	Logical: should row names be included? Defaults to NULL which includes row names if they are not just numeric in order.
boldRowNames	Logical: should row names be bolded?
boldColNames	Logical: should column names be bolded?
...	Additional arguments to be passed to kable

Details

Currently bolds in markdown format, so needs to be passed through interpreter after running.

Value

A formatted table from kable

Author(s)

Mark Peterson

See Also

kable

Examples

```
tempTable <- matrix(LETTERS[6:20], nrow = 5)
colnames(tempTable) <- LETTERS[24:26]
row.names(tempTable) <- LETTERS[1:5]
myKable(tempTable)

myKable(tempTable, boldColNames =  FALSE)
```

`printList` *Print nice lists*

Description

Generate a list formatted for printing from a vector.

Usage

```
printList(toPrint = letters[1:3], finalSepWord = "and", midSep = ",")
```

Arguments

`toPrint`	Vector that you want to turn into a text list.
`finalSepWord`	The last word to include, defaults to "and" but could be "or" or similar
`midSep`	Separator between items, defaults to ","

Details

Note that this function includes an Oxford comma.

Value

Character vector of length 1 with the values of `toPrint` concatenated and separated as specified in the text.

Author(s)

Mark Peterson

Examples

```
printList()

printList(LETTERS[1:5])

printList(letters[1:5], "or", ";")
```

refNote *Generate a popup reference note*

Description

Generates the html needed to include a popup reference note. Note requires inclusion of javascript libraries for this to function.

Usage

```
refNote(text = "This is a test note", number = "*")
```

Arguments

text	Note to include in popup. Line breaks occasionally cause problems. As this is generally for short notes; I have not come up with a general solution yet. If your note text is complicated, ensure that you check the result.
number	The label to identify the note in the text.

Value

The html for the popup note

Note

The html page also needs to include the javascript and css to process the notes for these to work. Specifically, need to include the following (generally in the header):

```
<!-- The javascript to run the popup refernce notes -->
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
    <script>
      jQuery.noConflict();
      jQuery(function() {
        jQuery(".refbody").hide();
        jQuery(".refnum").click(function(event) {
          jQuery(this.nextSibling).toggle();
          event.stopPropagation();
        });
        jQuery("body").click(function(event) {
          jQuery(".refbody").hide();
        });
      });
    </script>
```

And the following to wherever you include you css definitions:

```
/* the reference tooltips style starts here */
/* This was stolen from the What If? css sytle sheet
 *  at https://what-if.xkcd.com/css/style.css
 *  in an effort to emulate that style  */

.ref {
    position: relative;
    vertical-align: baseline;
   }

.refnum {
    position: relative;
    left: 2px;
    bottom: 1ex;
    font-family: Verdana, sans-serif;
    color: #005994;
    font-size: .7em;
    font-weight: bold;
    text-decoration: underline;
    cursor: pointer;
}

.refbody {
    font-family: Verdana, sans-serif;
    font-size: .7em;
    line-height: 1.1;
    display: block;
    min-width: 20em;
    position: absolute;
    left: 25px;
    bottom: 5px ;
    border: 1px solid;
    padding: 5px;
    background-color: #fff;
    word-wrap: break-word;
    z-index: 9999;
    overflow: auto;
}
```

Author(s)

Mark Peterson

References

This was based on the popups on the https://what-if.xkcd.com/ site

Examples

```
cat(refNote())

cat(refNote("Any text can go in here", 42))
```

`thisFileName` *Identify current file*

Description

Identify the file currently being processed

Usage

```
thisFileName()
```

Details

Currently only works for files being processed with `knit`; may try to add more

Value

The current file name

References

Based on https://github.com/krlmlr/kimisc/blob/master/R/thisfile.R

See Also

`thisfile_knit`

Examples

```
thisFileName()
```

`thisfile_knit` *Identify the file currently being knitted*

Description

Identifies the file that knitr is processing when the function is called.

Usage

```
thisfile_knit()
```

Value

Returns the current file name

Author(s)

Mark Peterson

References

Based on https://github.com/krlmlr/kimisc/blob/master/R/thisfile.R

See Also

`thisFileName`

Examples

```
thisfile_knit()
```

Index