Package 'Tex4exams'

May 12, 2023

Type Package

Title Generating 'Sweave' Code for 'R/exams' Questions in Mathematics

Version 0.1.2

Description

When using the R package 'exams' to write mathematics questions in 'Sweave' files, the output of a lot of R functions need to be adjusted for display in mathematical formulas. Specifically, the functions were accumulated when writing questions for the topics of the mathematics courses College Algebra, Precalculus, Calculus, Differential Equations, Introduction to Probability, and Linear Algebra. The output of the developed functions can be used in 'Sweave' files.

Depends numbers, fractional, pracma, polynom

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Qingwen Hu [aut, cre, cph] (<https://orcid.org/0000-0002-0482-5873>)

Maintainer Qingwen Hu <huqwen@gmail.com>

Repository CRAN

Date/Publication 2023-05-12 14:20:02 UTC

R topics documented:

`c2p` *Rational number sequence to polynomial in TeX code*

Description

The function 'polynomial' in the 'polynom' package converts a sequence of rational numbers into a polynomial with decimal coefficients. This function 'c2p' converts the output of 'polynomial' into the TeX form of a polynomial where coefficients are of vertical fraction form using the package 'fractional'.

Usage

```
c2p(m)
```

Arguments

`m`	a list of rational numbers which are coefficients of a polynomial in descending order.

Details

The function uses 'polynomial' function from the package 'polynom' which defaults the polynomial in ascending order.

Value

The function returns a string of TeX code of the polynomial with rational coefficients.

See Also

c2str, c2strpm.

Examples

```
m <- sample(c(1:100),5)
m
c2p(m)
```

---

c2str *Sequence of rational numbers into comma separated string*

Description

Convert a sequence of rational numbers into a string separated with a comma where the fractions are in backslash form.

Usage

```
c2str(x)
```

Arguments

x	a list of rational numbers.

Details

The output string was originally designed for 'string' type answers of 'R/exams' when a sequence of rational numbers are the answers from multiple parts of the question.

Value

The output is a string of rational numbers with backslash denoting division.

See Also

c2strpm

Examples

```
x <- sample(c(1:100),5)/100
x
c2str(x)
```

c2strpm *Sequence of numbers into a string of TeX code with plus minus signs.*

Description

Convert a sequence of numbers into a string of TeX code for the sequence with plus minus signs for each number of the sequence, where the fractions are in vertical form.

Usage

```
c2strpm(x)
```

Arguments

x	a list of rational numbers.

Value

The output is a string of rational numbers with backslash denoting division and plus minus signs in front of each number.

See Also

c2str

Examples

```
x <- sample(c(1:100),5)
x
c2strpm(x)
```

cm2l *Convert a matrix into a comma separated string.*

Description

The output of cm2l is a vector of strings with length equal to the column size of the input matrix. The i-th entry is the string of the numbers from the i-th column of the matrix. For example, the 2 by 2 identity matrix is converted into c("1,0", "0,1").

Usage

```
cm2l(matrix)
```

Arguments

matrix	a matrix.

Value

a vector of the strings of the columns of x. Each entry is a string of the column.

See Also

m2l, m22l, rm2l, rcm2l

Examples

```
a <- matrix(sample(c((-10):10),12),nrow =3,byrow=TRUE)
cm2l(a)
```

---

costex *Triangle leg values to TeX code of the cosine value*

Description

Convert a cosine value into TeX code with simplification on the radical from the hypotenuse, where the input (a,b) are the integer lengths of the legs of the right triangle with a the vertical leg, b the horizontal leg. The simplification is provided by another function 'simpRad' in the same package.

Usage

```
costex(a,b)
```

Arguments

a	The vertical length of the right triangle, which can be negative.
b	The horizontal length of the right triangle, which can be negative.

Details

Given integer lengths of a the legs of a triangle, the function returns a string of tex code for the value of the associated cosine.

Value

The function returns a string of TeX code for the value of the associated cosine.

Note

Caution: Integer coordinates (x,y) in the plane should switch order to be the input (y,x) of the function.

See Also

sintex,simpRad

Examples

```
a <- sample(c(1:5),1)
b <- sample(c(1:5),1)

costex(a,b)
```

`cycledisplay` *Display the cycle notation of the permutation*

Description

Display the cycle notation of the permutation using the output matrix of the function permucycle().

Usage

```
cycledisplay(x)
```

Arguments

x	output matrix of the function 'permucycle'.

Value

A cycle notation of a permutation.

See Also

permucycle, permuorder

Examples

```
cycledisplay(permucycle(c(3,2,1)))
paste0(cycledisplay(permucycle(c(3,2,1))),collapse ="")
```

delzero — *Delete the first zero in the output remainder of 'polydiv'*

Description

The output remainder of polydiv(x,y)[2] from the package 'pracma' may contain a zero in the first place which is not needed for text presentation. The function will modify the output by deleting the first zero, if any.

Usage

```
delzero(x)
```

Arguments

x	a list of numbers whose first entry may be zero or close to zero with absolute value less than or equal to 1e-10.

Value

The function truncates the first zero entry of the list.

Examples

```
x <- c(0,sample(c((-10):10),5))
delzero(x)
```

fmt4 — *Convert a decimal number into exactly 4 decimal places.*

Description

Convert a decimal number into exactly 4 decimal places without scientific notation.

Usage

```
fmt4(x)
```

Arguments

x	a decimal number.

Details

Round a decimal number into exactly 4 decimal places without scientific notation.

Value

A decimal number rounded into exactly 4 decimal places.

See Also

fmtN

Examples

```
x <- sin (sample(c(1:5),1))
fmt4(x)
```

---

fmtN *Convert a decimal number into exactly n decimal places.*

Description

Convert a decimal number into exactly n decimal places without scientific notation.

Usage

```
fmtN(x,n)
```

Arguments

x,n	where x is a decimal number and n is the numbers of decimal places to keep.

Details

Round a decimal number into exactly n decimal places.

Value

A decimal number rounded into exactly n decimal places.

See Also

fmt4 is a special case of fmtN but is simpler to use with one argument.

Examples

```
x <- sin (sample(c(1:5),1))
n <- sample(c(4:10),1)

fmtN(x,n)
```

G3S *Apply the Gram-Schmidt process to orthogonize a group of 3 vectors*

Description

Apply the Gram-Schmidt process to convert the group of 3 vectors (x,y,z) into an orthogonal group (u,v,w) without normalizing to unit vectors. The output is a matrix with columns (u,v,w).

Usage

```
G3S(x,y,z)
```

Arguments

x,y,z	a group of 3 vectors (x,y,z)

Details

Need the one dimensional projection function 'myGS' from the same package.

Value

The output is a matrix with columns (u,v,w) which are an orthogonal set of vectors.

See Also

myGS

Examples

```
G3S(c(1,2,3),c(3,2,1),c(4,5,9))
fractional(G3S(c(1,2,3),c(3,2,1),c(4,5,9)))
```

hypotex *TeX code of the hypotenuse*

Description

Given the lengths of the integer legs of a right triangle, the function generates the TeX code of the length of the hypotenuse in simplified form.

Usage

```
hypotex(a,b)
```

Arguments

a,b	a pair of the integer leg lengths of a right triangle.

Value

The function generates the TeX code of the hypotenuse in simplified form of the radicals.

See Also

simpRad, sintex, costex

Examples

```
a <- sample(c(1:5),1)
b <- sample(c(1:5),1)
hypotex(a,b)
```

---

inversions *Count the number of total inversions of a permutation*

Description

inversions() counts the number of inversions of a permutation.

Usage

```
inversions(x)
```

Arguments

x	a permutation of 1, 2, ..., n.

Details

Input must be a list of numbers.

Value

The total number of inversions in a list.

See Also

inversionv

Examples

```
inversions(c(3,1,2))
```

inversionv *Generating a vector of inversions for a permutation*

Description

inversionv() returns the vector of inversions for each entry in the permutation. For example, inversionv(1,2,0) = (1,1,0), inversionv(0,1,2) = (0,0,0).

Usage

```
inversionv(x)
```

Arguments

x	a list of numbers, or a permutation.

Value

inversionv returns a vector of inversions for each member of the permutation.

See Also

inversions

Examples

```
inversionv(c(3,1,2,4))
```

m22l *Converting a matrix into TeX code of a matrix without brackets*

Description

m22l converts a matrix into a matrix without brackets or parentheses around the array of numbers.

Usage

```
m22l(matrix)
```

Arguments

matrix	a matrix.

Value

array a numbers without parentheses.

See Also

m2l, rm2l, rcm2l, cm2l

Examples

```
a <- matrix(sample(c(-10:10),12),nrow =3,byrow=TRUE)
m22l(a)
```

---

m2l *Converting a matrix into TeX code of a matrix with square brackets*

Description

m2l converts a matrix into TeX code with square brackets.

Usage

```
m2l(matrix)
```

Arguments

matrix	a matrix

Value

TeX code of the matrix in 'Sweave' file.

See Also

rm2l, m22l, rcm2l, cm2l

Examples

```
a <- matrix(sample(c(-10:10),12),nrow =3,byrow=TRUE)
m2l(a)
```

mfrac — *Convert multiple decimals into tex code of vertical fractions*

Description

The function converts multiple decimals into tex code of vertical fractions.

Usage

```
mfrac(z)
```

Arguments

z	a list of decimal numbers.

Value

The function returns a string of tex code for the numbers in vertical fraction form.

Note

The function used function 'rfrac' in the same package which depends on the package 'fractional'.

See Also

smfrac,rfracF,rfrac

Examples

```
z <- sample(c(1:55),6)/100
mfrac(z)
```

myGS — *Orthogonal projection of y onto x*

Description

Orthogonal projection of y onto x.

Usage

```
myGS(x,y)
```

Arguments

x,y	a pairs of vectors.

Value

myGS returns a vector of the projection of y onto x.

See Also

G3S

Examples

```
x <-  sample(c((-10):10),3)
y <-  sample(c((-10):10),3)
myGS(x,y)
fractional(myGS(x,y))
```

---

perm *Lists permutations of a vector with distinct entries*

Description

perm(v) lists permutations of a vector v with distinct entries. The output is a matrix with each row a permutation. It cannot distinguish identical permutations.

Usage

```
perm(v)
```

Arguments

v	is a vector with distinct entries.

Value

an n! by n matrix with each row a permuation of the enties of v.

See Also

permucycle, permuorder,cycledisplay

Examples

```
x <-  sample(c((-10):10),3)
perm(x)
```

permucycle *Matrix representation of the cycles of a permutation.*

Description

permucycle returns a matrix containing information of a permutation. See the Value section for details.

Usage

```
permucycle(x)
```

Arguments

x	a permutation

Value

Given a permutation x of the numbers 1, 2, ..., n, and i with value from 1 to n. permucycle() returns a (n+1) by (n+1) matrix A with

A[1,1]	the total number of cycles
A[(i+1),1]	the length of the i-th cycle
A[(i+1),2:(n+1)]	the members of the i-th cycle

For example, permucycle(c(3,2,1)) will produce the following matrix:

[,1] [,2] [,3] [,4]

[1,] 2 0 0 0

[2,] 2 1 3 0

[3,] 1 2 0 0

[4,] 0 0 0 0

The 2 in the first row means there are two cycles; The second row means there is a cycle of length 2, with members (1,3); The third row means there is a cycle of length 1, with member (2); The fourth row is redundant for this specific case. One can read from the output of permucycle() to obtain cycle notation (13)(2) of the permutation, and other information.

See Also

perm, permuorder,cycledisplay

Examples

```
permucycle(c(3,2,1))
```

permuorder *Order of permutation*

Description

Computing the order of the permutation x of the numbers 1, 2, ..., n, using the first column of the output matrix of the function 'permucycle', whose first entry N = permucycle()[1,1] is the total number of the cycles in the permutation, and permucycle()[2:(n+1),1] are the lengths of each cycle. Note: Since matrix can be regarded as a 1 dimensional vector with each column attached with the previous one, the argument of permuorder() can be the whole output of permucycle() when permuorder() just uses the first N+1 entries in the first column. Certainly specifying the first column of permucycle() to be the input will save some memory usage.

Usage

```
permuorder(x)
```

Arguments

x	output matrix of the function permucycle()

Value

The order of the permutation, which is the least common multiple of the orders of each contained cycle.

See Also

permucycle, cycledisplay

Examples

```
permuorder(permucycle(c(3,1,2)))
```

pos *TeX code for positivity of randomized values*

Description

Produce the TeX code of positivity of randomaized values.

Usage

```
pos(x)
```

Arguments

x	a numeric number.

Value

The function returns one of the symbols "<0", ">0", or "=0".

See Also

signF

Examples

```
x <- sample(c((-10):10),1)
pos(x)
```

---

rcm2l *Converting a matrix into a comma separated string*

Description

The output of rcm2l is a vector of strings with length equal to the row size of the input matrix. The i-th entry is the string of the numbers from the i-th row of the matrix. For example, the standard 2 by 2 Jordan block with 2 in the main diagonal is converted into c("2,1", "0,2").

Usage

```
rcm2l(matrix)
```

Arguments

matrix	a matrix

Value

a vector of strings with length equal to the row size of the input matrix. The i-th entry is the string of the numbers from the i-th row of the matrix.

See Also

m2l, m22l, rm2l, cm2l

Examples

```
a <- matrix(sample(c(-10:10),12),nrow =3,byrow=TRUE)
rcm2l(a)
```

rfrac *Convert a decimal into TeX code of vertical fractions*

Description

The function converts a decimal into TeX code of vertical fractions, using functions 'denominators' and 'numerators' from the package 'fractional'.

Usage

```
rfrac(x)
```

Arguments

x	a list of decimal numbers.

Value

The function returns a string of TeX code for the number in vertical fraction form.

Note

The function depends on the package 'fractional'.

See Also

mfrac,smfrac,rfracF

Examples

```
x <-  sample(c(1:55),1)/100
rfrac(x)
```

rfracF *Convert a decimal into TeX code of vertical fractions with the sign in front of the fraction*

Description

The function converts a decimal into TeX code of vertical fractions, using functions 'denominators' and 'numerators' from the packaege 'fractional'. This function differs from 'rfrac' only in the place of the negative signs. 'rfracF' returns the fraction with the sign in front.

Usage

```
rfracF(x)
```

Arguments

x	a list of decimal numbers.

Value

The function returns a string of tex code for the number in vertical fraction form with the sign in front of the fraction.

Note

The function depends on the package 'fractional'.

See Also

mfrac,smfrac, rfrac

Examples

```
x <-  sample(c(1:55),1)/100
rfrac(x)
```

---

rm2l *Convert a matrix into its transpose in TeX code*

Description

rm2l converts a matrix into its transpose in TeX code.

Usage

```
rm2l(matrix)
```

Arguments

matrix	a matrix.

Value

It return the transpose of the input matrix in latex code.

See Also

m2l, m22l, rcm2l, cm2l

Examples

```
a <- matrix(sample(c((-10):10),12),nrow =3,byrow=TRUE)
rm2l(a)
```

signF *Show minus sign in front of fractions*

Description

The minus sign in fractions should be in front of the fraction for display purpose, while many numerical algorithms may produce fractions with minus sign in the numerator. This function modifies the output of those packages into the correct display form in TeX code.

Usage

```
signF(a)
```

Arguments

a	a numeric number.

Value

The function returns one of the symbols "-" for negative numbers, or empty "" for nonnegative ones.

See Also

rfracF

Examples

```
a <- sample(c((-10):10),1)/100
signF(a)
```

simpRad *Simplify square roots of positive integers*

Description

SimpRad uses the package 'numbers' to simplify square roots of positive integers.

Usage

```
simpRad(n)
```

Arguments

n	a positive integer.

Value

The function returns a string of TeX code for radical in simplified form.

Note

The function used functions 'primFactors' and 'radical' from the package of 'numbers'.

See Also

radical, primeFactors

Examples

```
n <- sample(c(4:100),1)
simpRad(n)
```

`sintex` *Triangle leg values to TeX code of the cosine value*

Description

Convert a sine value into TeX code with simplification on the radical from the hypotenuse, where the input (a,b) are the integer lengths of the legs of the right triangle with a the vertical leg, b the horizontal leg. The simplification is provided by another function 'simpRad' in the same package.

Usage

```
sintex(a,b)
```

Arguments

a	The vertical length of the right triangle, which can be negative.
b	The horizontal length of the right triangle, which can be negative.

Details

Given integer lengths of the legs of a triangle, the function returns a string of TeX code for the value of the associated sine.

Value

The function returns a string of TeX code for the value of the associated sine.

Note

Caution: Integer coordinates (x,y) in the plane should switch order to be the input (y,x) of the function.

See Also

costex,simpRad

Examples

```
a <- sample(c(1:5),1)
b <- sample(c(1:5),1)

sintex(a,b)
```

---

smfrac *Convert multiple decimals into TeX code of back slash fractions*

Description

The function converts multiple decimals into TeX code of back slash fractions.

Usage

```
smfrac(z)
```

Arguments

z	a list of decimal numbers.

Value

The function returns a string of TeX code for the numbers in back slash fraction form.

Note

The function used function 'rfrac' in the same package which depends on the package 'fractional'.

See Also

mfrac,rfracF,rfrac

Examples

```
z <- sample(c(1:55),6)/100
smfrac(z)
```

Index