

iemisc: Additional Examples from GNU Octave size Compatible Functions

Irucka Embry, E.I.T. (EcoC²S)

2024-06-05

Contents

size Examples (R style)	1
size Examples (GNU Octave style)	3
length_octave Examples (R style)	6
length Examples (GNU Octave style)	7
numel Examples (R style)	8
numel Examples (GNU Octave style)	9
ndims Examples (R style)	11
ndims Examples (GNU Octave style)	11
isrow Examples (R style)	12
isrow Examples (GNU Octave style)	12
iscolumn Examples (R style)	13
iscolumn Examples (GNU Octave style)	13
Works Cited	14
EcoC ² S Links	14
Copyright and License	14

size Examples (R style)

```
library("iemisc")
```

```
import::from(gsubfn, list)  
import::from(ramify, mat)
```

```

# Example from GNU Octave ndims function reference

size(matlab::ones(4, 1, 2, 1))

## [1] 4 1 2
# Examples from GNU Octave size function reference

object1 <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 3, ncol = 2, byrow = TRUE)

size(object1)

## [1] 3 2
list[nr, nc] <- size(matrix(c(1, 2, 3, 4, 5, 6), nrow = 3, ncol = 2, byrow = TRUE))
nr

## [1] 3
nc

## [1] 2
size(matrix(c(1, 2, 3, 4, 5, 6), nrow = 3, ncol = 2, byrow = TRUE), 2)

## [1] 2
# using ramify's mat

size(mat("1, 2; 3, 4; 5, 6"))

## [1] 3 2
size(mat("1, 2; 3, 4; 5, 6"), 2)

## [1] 2
list[nr, nc] <- size(mat("1, 2; 3, 4; 5, 6"))
nr

## [1] 3
nc

## [1] 2
size(matlab::ones(4, 3, 4, 8), 4)

## [1] NA
size(matlab::ones(4, 3, 4, 5), 3)

## [1] 4
## The following can't be done currently with this function:

# list[nr, remainder] <- size(matlab::ones(2, 3, 4, 5)); nr; remainder

## As a work around to get similar results to GNU Octave, do the following:

nr <- size(matlab::ones(2, 3, 4, 5), 1)
nr

## [1] 2

```

```

remainder <- size(matlab::ones(2, 3, 4, 5), 2)
remainder

## [1] 60
# Examples from pracma size

size(1:8)

## [1] 1 8
size(matrix(1:8, 2, 4))

## [1] 2 4
size(matrix(1:8, 2, 4), 2)

## [1] 4
size(matrix(1:8, 2, 4), 3)

## [1] NA
ss <- "object"

size(ss)

## [1] 1 6

```

size Examples (GNU Octave style)

```

% check against GNU Octave

% Example from GNU Octave ndims function reference
size(ones(4, 1, 2, 1))

% Examples from GNU Octave size function reference
object1 = [1, 2; 3, 4; 5, 6]

size(object1)

[nr, nc] = size([1, 2; 3, 4; 5, 6])

size([1, 2; 3, 4; 5, 6], 2)

size([1 2; 3 4; 5 6])

size([1 2; 3 4; 5 6], 2)

[nr, nc] = size([1, 2; 3, 4; 5, 6])

```

```

nr
nc

size(ones(2, 3, 4, 5))

[nr, remainder] = size(ones(2, 3, 4, 5))
nr
remainder

size(ones(4, 3, 4, 8), 4)

size(ones(4, 3, 4, 5), 3)

% Examples from pracma size

size(1:8)

object2 = [1 3 5 7; 2 4 6 8]

size(object2)

size(object2, 2)

size(object2, 3)

ss = 'object'

size(ss)

% results

>> % Example from GNU Octave ndims function reference
>>
>> size(ones(4, 1, 2, 1))
ans =
    4    1    2

>>
>>
>> % Examples from GNU Octave size function reference
>>
>> object1 = [1, 2; 3, 4; 5, 6]
object1 =
    1    2
    3    4
    5    6

```

```

>>
>> size(object1)
ans =

    3    2

>>
>>
>> [nr, nc] = size([1, 2; 3, 4; 5, 6])
nr = 3
nc = 2
>>
>> size([1, 2; 3, 4; 5, 6], 2)
ans = 2
>>
>> size([1 2; 3 4; 5 6])
ans =

    3    2

>>
>> size([1 2; 3 4; 5 6], 2)
ans = 2
>>
>> [nr, nc] = size([1, 2; 3, 4; 5, 6])
nr = 3
nc = 2
>> nr
nr = 3
>> nc
nc = 2
>>
>> size(ones(2, 3, 4, 5))
ans =

    2    3    4    5

>>
>> [nr, remainder] = size(ones(2, 3, 4, 5))
nr = 2
remainder = 60
>> nr
nr = 2
>> remainder
remainder = 60
>>
>> size(ones(4, 3, 4, 8), 4)
ans = 8
>>
>> size(ones(4, 3, 4, 5), 3)
ans = 4
>>
>>

```

```

>>
>>
>> % Examples from pracma size
>>
>> size(1:8)
ans =

    1    8

>>
>> object2 = [1 3 5 7; 2 4 6 8]
object2 =

    1    3    5    7
    2    4    6    8

>>
>> size(object2)
ans =

    2    4

>>
>> size(object2, 2)
ans = 4
>>
>> size(object2, 3)
ans = 1
>>
>> ss = 'object'
ss = object
>>
>> size(ss)
ans =

    1    6

>>

```

length_octave Examples (R style)

```

library("iemisc")

import::from(matlab, ones)

# Example from pracma isempty

object1 <- matrix(0, 1, 0)

length_octave(object1)

```

```

## [1] 0
object2 <- 2

length_octave(object2)

## [1] 1
object3 <- 1:10

length_octave(object3)

## [1] 10
object4 <- ones(3, 4)

length_octave(object4)

## [1] 4
object5 <- "ss"

length_octave(object5)

## [1] 2
object6 <- list(letters, b <- 2)

length_octave(object6)

## [1] 2

```

length Examples (GNU Octave style)

```

% check against GNU Octave

object1 = [];

length(object1)

object2 = 2;

length(object2)

object3 = 1:10;

length(object3)

object4 = ones(3, 4);

```

```
length(object4)
```

```
object5 = 'ss';
```

```
length(object5)
```

```
%% results
```

```
>> object1 = [];
```

```
>>
```

```
>> length(object1)
```

```
ans = 0
```

```
>>
```

```
>>
```

```
>> object2 = 2;
```

```
>>
```

```
>> length(object2)
```

```
ans = 1
```

```
>>
```

```
>>
```

```
>> object3 = 1:10;
```

```
>>
```

```
>> length(object3)
```

```
ans = 10
```

```
>>
```

```
>>
```

```
>> object4 = ones(3, 4);
```

```
>>
```

```
>> length(object4)
```

```
ans = 4
```

```
>>
```

```
>>
```

```
>> object5 = 'ss';
```

```
>>
```

```
>> length(object5)
```

```
ans = 2
```

numel Examples (R style)

```
library("iemisc")
```

```
import::from(matlab, ones)
```

```
xx <- list(1:26, 1:10)
```

```
numel(xx)
```



```

## [1] 2
# Examples from GNU Octave numel

a <- 1

b <- ones(2, 3)

numel(a, b)

## [1] 6
a <- 2

b <- ones(2, 3)

c <- ones(3, 4)

numel(a, b)

## [1] 6
numel(a, b, c)

## [1] 72
f <- matrix(c(10, 12, 23, 21, 62, 93), nrow = 2, ncol = 3, byrow = TRUE)

g <- c(2, 4)

numel(f, g)

## [1] 2

```

numel Examples (GNU Octave style)

```

% check against GNU Octave

xx = {1:26, 1:10}

% Examples from GNU Octave numel
a = 1;

b = ones(2, 3);

numel(a, b)

a = 2;

b = ones(2, 3);

```

```

c = ones(3, 4);

numel(a, b)

numel(a, b, c)

f = [10 12 23; 21 62 93];

g = [2 4];

numel(f, g)

% results

>> xx = {1:26, 1:10}
xx =
{
  [1,1] =

    Columns 1 through 15:

         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15

    Columns 16 through 26:

        16    17    18    19    20    21    22    23    24    25    26

  [1,2] =

         1     2     3     4     5     6     7     8     9    10

}
>>
>>
>> % Examples from GNU Octave numel
>> a = 1;
>>
>> b = ones(2, 3);
>>
>> numel(a, b)
ans = 6
>>
>>
>> a = 2;
>>
>> b = ones(2, 3);
>>
>> c = ones(3, 4);
>>
>> numel(a, b)

```

```
ans = 6
>>
>> numel(a, b, c)
ans = 72
>>
>>
>> f = [10 12 23; 21 62 93];
>>
>> g = [2 4];
>>
>> numel(f, g)
ans = 2
>>
```

ndims Examples (R style)

```
library("iemisc")

# Examples from GNU Octave ndims

b <- matlab::ones(c(4, 1, 2, 1))

ndims(b)

## [1] 3
```

ndims Examples (GNU Octave style)

```
% check against GNU Octave

% Example from GNU Octave ndims

ndims(ones(4, 1, 2, 1))

% results

>> ndims(ones(4, 1, 2, 1))
ans = 3
```

isrow Examples (R style)

```
library("iemisc")
```

```
# Examples
```

```
xx <- ramify::mat("1, 2")  
xx
```

```
-----  
1 2  
-----
```

```
isrow(xx)
```

```
## [1] TRUE
```

```
xy <- ramify::mat("1, 2; 3, 4")  
xy
```

```
-----  
1 2  
3 4  
-----
```

```
isrow(xy)
```

```
## [1] FALSE
```

isrow Examples (GNU Octave style)

```
% check against GNU Octave
```

```
isrow([1 2])
```

```
isrow([1 2; 3 4])
```

```
% results
```

```
>> isrow([1 2])
```

```
ans = 1
```

```
>>
```

```
>> isrow([1 2; 3 4])
```

```
ans = 0
```

```
>>
```

iscolumn Examples (R style)

```
library("iemisc")
```

```
# Examples
```

```
xxx <- ramify::mat("1, 2")  
xxx
```

```
-----  
1 2  
-----
```

```
iscolumn(xxx)
```

```
## [1] FALSE
```

```
xyy <- ramify::mat("1; 2")  
xyy
```

```
-  
1  
2  
-
```

```
iscolumn(xyy)
```

```
## [1] TRUE
```

iscolumn Examples (GNU Octave style)

```
% check against GNU Octave
```

```
iscolumn([1 2])
```

```
iscolumn([1; 2])
```

```
% results
```

```
>> iscolumn([1 2])
```

```
ans = 0
```

```
>>
```

```
>> iscolumn([1; 2])
```

```
ans = 1
```

```
>>
```

Works Cited

John W. Eaton, David Bateman, Søren Hauberg, and Rik Wehbring (November 2022). *GNU Octave: A high-level interactive language for numerical computations*: Edition 7 for Octave version 7.3.0. <https://docs.octave.org/octave.pdf>. Pages 46-48, 68.

EcoC²S Links

EcoC²S Home – <https://www.ecoccs.com/>

About EcoC²S – https://www.ecoccs.com/about_ecoc2s.html

Services – <https://www.ecoccs.com/services.html>

1 Stop Shop – https://www.ecoccs.com/other_biz.html

Products – <https://www.questionuniverse.com/products.html>

Media – <https://www.ecoccs.com/media.html>

Resources – <https://www.ecoccs.com/resources.html>

R Trainings and Resources provided by EcoC²S (Irucka Embry, E.I.T.) – <https://www.ecoccs.com/rtraining.html>

Copyright and License

All R code written by Irucka Embry is distributed under the GPL-3 (or later) license, see the [GNU General Public License {GPL} page](#).

All written content originally created by Irucka Embry is copyrighted under the Creative Commons Attribution-ShareAlike 4.0 International License. All other written content retains the copyright of the original author(s).

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).