ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Basic Indexing

A *cell array* is a data type with indexed data containers called cells. Each cell can contain any type of data. Cell arrays are often used to hold data from a file that has inconsistent formatting, such as columns that contain both numeric and text data.

For instance, consider a 2-by-3 cell array of mixed data.

```
C = {'one','two','three';
     100,200,rand(3,3)}
```

```
C=2×3 cell array
    {'one'}    {'two'}    {'three'   }
    {[100]}    {[200]}    {3x3 double}
```

Each element is within a cell. If you index into this array using standard parentheses, the result is a subset of the cell array that includes the cells.

```
C2 = C(1:2,1:2)
```

```
C2=2×2 cell array
    {'one'}    {'two'}
    {[100]}    {[200]}
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Basic Indexing

To read or write the contents within a specific cell, enclose the indices in curly braces.

```
R = C{2,3}
```

```
R = 3×3

    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575
```

```
C{1,3} = 'longer text in a third location'
```

```
C=2×3 cell array
    {'one'}    {'two'}    {'longer text in a third location'}
    {[100]}    {[200]}    {3x3 double                       }
```

To replace the contents of multiple cells at the same time, use parentheses to refer to the cells and curly braces to define an equivalently sized cell array.

```
C(1,1:2) = {'first','second'}
```

```
C=2×3 cell array
    {'first'}    {'second'}    {'longer text in a third location'}
    {[  100]}    {[   200]}    {3x3 double                       }
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Read Data from Multiple Cells

Most of the data processing functions in MATLAB® operate on a rectangular array with a uniform data type. Because cell arrays can contain a mix of types and sizes, you sometimes must extract and combine data from cells before processing that data. This section describes a few common scenarios.

Text in Specific Cells

When the entire cell array or a known subset of cells contains text, you can index and pass the cells directly to any of the text processing functions in MATLAB. For instance, find where the letter *t* appears in each element of the first row of `C`.

```
ts = strfind(C(1,:),'t')
```

```
ts=1×3 cell array
    {[5]}    {0x0 double}    {[8 11 18 28]}
```

Numbers in Specific Cells

The two main ways to process numeric data in a cell array are:

- Combine the contents of those cells into a single numeric array, and then process that array.
- Process the individual cells separately.

To combine numeric cells, use the `cell2mat` function. The arrays in each cell must have compatible sizes for concatenation. For instance, the first two elements of the second row of `C` are scalar values. Combine them into a 1-by-2 numeric vector.

```
v = cell2mat(C(2,1:2))
```

```
v = 1×2

   100   200
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Read Data from Multiple Cells

To process individual cells, you can use the `cellfun` function. When calling `cellfun`, specify the function to apply to each cell. Use the `@` symbol to indicate that it is a function and to create a function handle. For instance, find the length of each of the cells in the second row of `C`.

```
len = cellfun(@length,C(2,:))
```

```
len = 1×3

     1     1     3
```

Data in Cells with Unknown Indices

When some of the cells contain data that you want to process, but you do not know the exact indices, you can use one of these options:

- Find all the elements that meet a certain condition using logical indexing, and then process those elements.
- Check and process cells one at a time with a `for`- or `while`-loop.

For instance, suppose you want to process only the cells that contain character vectors. To take advantage of logical indexing, first use the `cellfun` function with `ischar` to find those cells.

```
idx = cellfun(@ischar,C)
```

```
idx = 2x3 logical array

   1   1   1
   0   0   0
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Read Data from Multiple Cells

Data in Cells with Unknown Indices

Then, use the logical array to index into the cell array, `C(idx)`. The result of the indexing operation is a column vector, which you can pass to a text processing function, such as `strlength`.

```
len = strlength(C(idx))
```

```
len = 3×1

     5
     6
    31
```

The other approach is to use a loop to check and process the contents of each cell. For instance, find cells that contain the letter *t* and combine them into a string array by looping through the cells. Track how many elements the loop adds to the string array in variable n.

```
n = 0;
for k = 1:numel(C)
    if ischar(C{k}) && contains(C{k},"t")
        n = n + 1;
        txt(n) = string(C{k});
    end
end
txt
```

```
txt = 1x2 string
    "first"    "longer text in a third location"
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Index into Multiple Cells

If you refer to multiple cells using curly brace indexing, MATLAB returns the contents of the cells as a *comma-separated list*. For example,

```
C{1:2,1:2}
```

is the same as

```
C{1,1}, C{2,1}, C{1,2}, C{2,2}.
```

Because each cell can contain a different type of data, you cannot assign this list to a single variable. However, you can assign the list to the same number of variables as cells.

```
[v1,v2,v3,v4] = C{1:2,1:2}
```

```
v1 =
'first'

v2 = 100

v3 =
'second'

v4 = 200
```

If each cell contains the same type of data with compatible sizes, you can create a single variable by applying the array concatenation operator [ ] to the comma-separated list.

```
v = [C{2,1:2}]
```

```
v = 1×2

   100   200
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Index into Multiple Cells

If the cell contents cannot be concatenated, store results in a new cell array, table, or other heterogeneous container. For instance, convert the numeric data in the second row of `C` to a table. Use the text data in the first row of `C` for variable names.

```
t = cell2table(C(2,:),VariableNames=C(1,:))
```

```
t=1×3 table
    first    second    longer text in a third location
    _____    ______    _______________________________

     100      200              {3x3 double}
```

Index into Arrays Within Cells

If a cell contains an array, you can access specific elements within that array using two levels of indices. First, use curly braces to access the contents of the cell. Then, use the standard indexing syntax for the type of array in that cell.

For example, `C{2,3}` returns a 3-by-3 matrix of random numbers. Index with parentheses to extract the second row of that matrix.

```
C{2,3}(2,:)
```

```
ans = 1×3

    0.9058    0.6324    0.5469
```

If the cell contains a cell array, use curly braces for indexing, and if it contains a structure array, use dot notation to refer to specific fields. For instance, consider a cell array that contains a 2-by-1 cell array and a scalar structure with fields `f1` and `f2`.

```
c = {'A'; ones(3,4)};
s = struct("f1",'B',"f2",ones(5,6));
C = {c,s}
```

```
C=1×2 cell array
    {2x1 cell}    {1x1 struct}
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

Index into Arrays Within Cells

Extract the arrays of ones from the nested cell array and structure.

```
A1 = C{1}{2}
```

```
A1 = 3×4

     1     1     1     1
     1     1     1     1
     1     1     1     1
```

```
A2 = C{2}.f2
```

```
A2 = 5×6

     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
```

You can nest any number of cell and structure arrays. Apply the same indexing rules to lower levels in the hierarchy. For instance, these syntaxes are valid when the referenced cells contain the expected cell or structure array.

```
C{1}{2}{3}
C{4}.f1.f2(1)
C{5}.f3.f4{1}
```

At any indexing level, if you refer to multiple cells, MATLAB returns a comma-separated list. For details, see Index into Multiple Cells.

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

cell

Cell array

Description

A *cell array* is a data type with indexed data containers called *cells*, where each cell can contain any type of data. Cell arrays commonly contain either lists of text, combinations of text and numbers, or numeric arrays of different sizes. Refer to sets of cells by enclosing indices in smooth parentheses, ( ). Access the contents of cells by indexing with curly braces, {}.

Creation

When you have data to put into a cell array, create the array using the cell array construction operator, {}.

```
C = {1,2,3;
     'text',rand(5,10,2),{11; 22; 33}}
```

```
C=2×3 cell array
    {[   1]}    {[          2]}    {[     3]}
    {'text'}    {5×10×2 double}    {3×1 cell}
```

You also can use {} to create an empty 0-by-0 cell array.

```
C = {}
```

```
C =

  0×0 empty cell array
```

To create a cell array with a specified size, use the `cell` function, described below.

You can use `cell` to preallocate a cell array to which you assign data later. `cell` also converts certain types of Java®, .NET, and Python® data structures to cell arrays of equivalent MATLAB® objects.

ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

cell

Cell array

Examples

Create Cell Array

When related pieces of data have different data types, you can keep them together in a cell array. Each cell contains a piece of data. To refer to elements of a cell array, use array indexing. You can index into a cell array using smooth parentheses, `()`, and into the contents of cells using curly braces, `{}`.

Create a cell array that contains several temperature readings taken on a given date. Specify a date as a character vector, and temperatures as an array of doubles. To store these pieces of data in a cell array, enclose them in curly braces.

```
C = {'2017-08-16',[56 67 78]}
```

```
C=1×2 cell array
    {'2017-08-16'}    {[56 67 78]}
```

Add readings for different dates to the cell array. One way to add more cells is to expand the cell array by assignment, just as you can expand an ordinary array.

```
C(2,:) = {'2017-08-17',[58 69 79]};
C(3,:) = {'2017-08-18',[60 68 81]}
```

```
C=3×2 cell array
    {'2017-08-16'}    {[56 67 78]}
    {'2017-08-17'}    {[58 69 79]}
    {'2017-08-18'}    {[60 68 81]}
```

Index into the first row of `C`. When you index with smooth parentheses, `()`, the result is a cell array that is a subset of the cell array.

```
C(1,:)
```

```
ans=1×2 cell array
    {'2017-08-16'}    {[56 67 78]}
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

cell

Cell array

Index into the contents of a cell. When you index with curly braces, {}, the result is the piece of data that is contained in the specified cell.

```
C{1,2}
```

```
ans = 1×3

    56    67    78
```

Square Cell Array

Create a 3-by-3 cell array of empty matrices.

```
C = cell(3)
```

```
C=3×3 cell array
    {0x0 double}    {0x0 double}    {0x0 double}
    {0x0 double}    {0x0 double}    {0x0 double}
    {0x0 double}    {0x0 double}    {0x0 double}
```

Tips

- Creating a cell array of empty matrices with the `cell` function is equivalent to assigning an empty matrix to the last index of a new cell array. For example, these two statements are equivalent:

```
C = cell(3,4,2);
C{3,4,2} = [];
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

cell2mat

Convertire l'array di celle in un array ordinario del tipo di dati sottostante

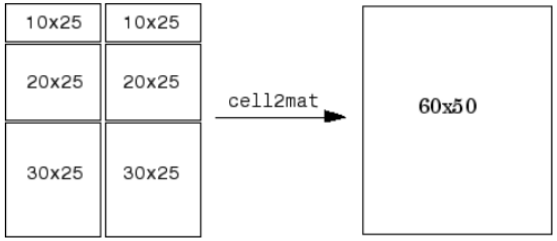
Sintassi

```
A = cell2mat(C)
```

Descrizione

`A = cell2mat(C)` converte un array di celle in un array ordinario. Gli elementi dell'array di celle devono contenere tutti lo stesso tipo di dati e l'array risultante avrà lo stesso tipo di dati.

I contenuti di `C` devono supportare la concatenazione in un rettangolo di N dimensioni. Diversamente, i risultati saranno indefiniti. Ad esempio, i contenuti delle celle nella stessa colonna devono avere lo stesso numero di colonne, sebbene non debbano avere lo stesso numero di righe (vedere la figura).



ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Cell Array

cell2mat

Convertire l'array di celle in un array ordinario del tipo di dati sottostante

Esempi

Conversione di array di celle in array numerici

Convertire gli array numerici a quattro celle di un array di celle in un array numerico.

```
C = {[1],    [2 3 4];
     [5; 9], [6 7 8; 10 11 12]}
```

```
C=2×2 cell array
    {[       1]}    {[   2 3 4]}
    {2x1 double}    {2x3 double}
```

```
A = cell2mat(C)
```

```
A = 3×4

     1     2     3     4
     5     6     7     8
     9    10    11    12
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

A table is a container that stores column-oriented data in variables. To access the data in a table, you can index into the table by specifying rows and variables, just as you can index into a matrix by specifying rows and columns. Table variables have names, just as the fields of a structure have names. The rows of a table also can have names, but row names are not required. To index into a table, specify rows and variables using either positions, names, or data types. The result can be either an array or a table.

This topic describes the different table indexing syntaxes and when to use each type. Additional examples show the different ways to apply these table indexing types. The table at the end of the topic summarizes the indexing syntaxes, how to specify rows and variables, and the resulting outputs.

Table Indexing Syntaxes

Depending on the type of indexing you use, the result is either an array extracted from the table or a new table. Indexing with:

- **Dot notation,** as in `T.varname` or `T.(expression)`, extracts an array from one table variable.
- **Curly braces,** as in `T{rows,vars}`, extract an array from the specified rows and variables. The variables must have compatible data types so that they can be concatenated into one array.
- **Parentheses,** as in `T(rows,vars)`, return a table that has only the specified rows and variables.

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

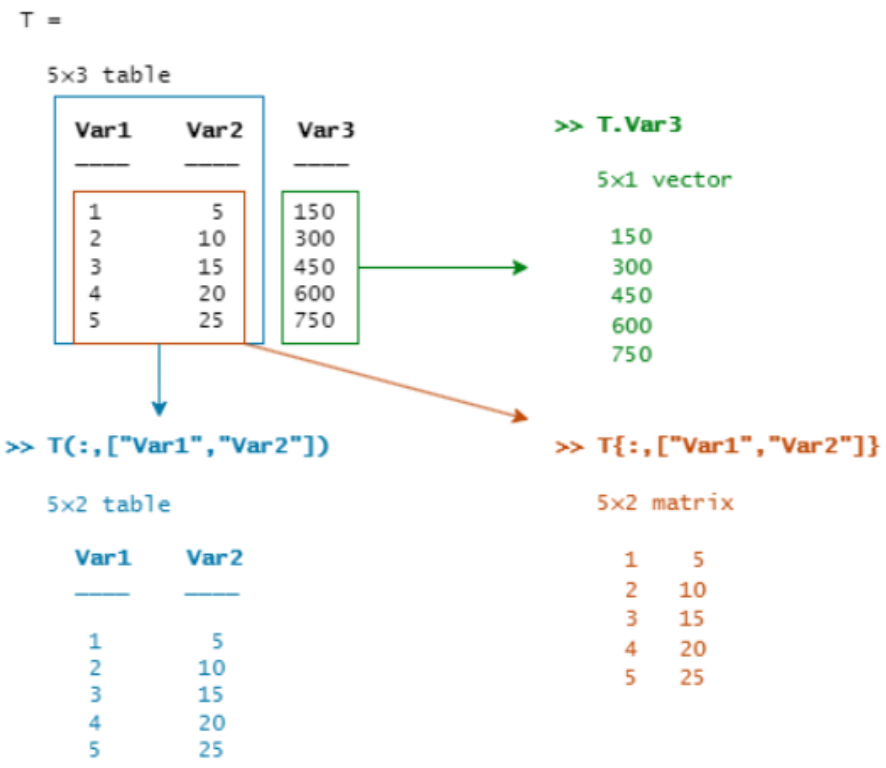
Argomenti

Indicizzazione di array

Access Data in Tables

Table Indexing Syntaxes

This diagram shows the three types of table indexing.



ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Recommended Indexing Syntaxes

The recommended way to access the contents of a table depends on the result you want and the number of variables that you specify. In these syntax examples, `T` is a table that has variables named `Var1`, `Var2`, and `Var3`. (If you do not specify variable names when calling the `table` function, then these names are the default names.)

```
T = table([1;2;3;4;5],[5;10;15;20;25],[150;300;450;600;750])
```

- **To access one table variable, use dot notation.** Specify a variable name or an expression that matches the name or position of a variable.

  Using literal names of variables is faster than using expressions. For example, `T.Var1` and `T.(1)` both access the first variable of `T`, but using the expression is slower.

```
X = T.Var1
Y = T.Var1(1:3)
Z = T.(1)
T.Var1 = T.Var1 .* 10
```

  You can also specify one variable using curly braces. However, accessing a variable using dot notation is faster than accessing a variable using curly braces.

- **To access multiple table variables, use curly braces.**

```
X = T{:,["Var1","Var2"]}
Y = T{1:3,["Var1","Var2"]}
T{:,["Var1","Var2"]} = T{:,["Var1","Var2"]} .* 10
```

- **To return a table that has only specified rows and variables, use parentheses.**

```
T2 = T(:,["Var1","Var2"])
T2 = T(1:3,["Var1","Var2"])

A = rand(5,1)
B = rand(5,1)
T(:,["Var1","Var2"]) = table(A,B)
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

You can index into tables by specifying numeric indices, row and variable names, or variable data types.

Create a table. Load arrays of data from the sample `patients.mat` file. Then create a table from these arrays using the `table` function. The names of the input arrays become the names of the table variables. Row names are optional. To specify row names, use the `RowNames` name-value argument.

```
load patients.mat Age Height Weight Smoker LastName
T = table(Age,Height,Weight,Smoker,RowNames=LastName)
```

```
T=100×4 table
               Age    Height    Weight    Smoker
               ___    ______    ______    ______

    Smith       38       71       176      true
    Johnson     43       69       163      false
    Williams    38       64       131      false
    Jones       40       67       133      false
    Brown       49       64       119      false
    Davis       46       68       142      false
    Miller      33       64       142      true
    Wilson      40       68       180      false
    Moore       28       68       183      false
    Taylor      31       66       132      false
    Anderson    45       68       128      false
    Thomas      42       66       137      false
    Jackson     25       71       174      false
    White       39       72       202      true
    Harris      36       65       129      false
    Martin      48       71       181      true
      ⋮
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

Index by Position

You can index into tables by specifying positions as numeric indices. You can also use colons and the `end` keyword.

For example, index into the first three rows of `T`. This syntax is a compact way to return a table with the specified number of rows.

```
firstRows = T(1:3,:)
```

```
firstRows=3×4 table
                Age    Height    Weight    Smoker
                ___    ______    ______    ______

    Smith       38       71        176      true
    Johnson     43       69        163      false
    Williams    38       64        131      false
```

Return a table with the first two variables and the last three rows of `T`.

```
lastRows = T(end-2:end,1:2)
```

```
lastRows=3×2 table
                Age    Height
                ___    ______

    Griffin     49       70
    Diaz        45       68
    Hayes       48       66
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

Index by Position

If the variables have compatible data types, then you can use curly braces to return the extracted data as an array.

```
lastRowsAsArray = T{end-2:end,1:2}
```

```
lastRowsAsArray = 3×2

    49    70
    45    68
    48    66
```

Index by Variable Names

You can index into tables by specifying variable names using a string array. Table variable names do not have to be valid MATLAB® identifiers. They can include spaces and non-ASCII characters, and can start with any character.

For example, return a table that has only the first three rows of T and the `Height` and `Weight` variables.

```
variablesByName = T(1:3,["Height","Weight"])
```

```
variablesByName=3×2 table
               Height    Weight
               ______    ______

    Smith        71        176
    Johnson      69        163
    Williams     64        131
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

Index by Variable Names

You can also use dot notation to index into one variable. In fact, dot notation is more efficient when you access just one variable.

```
heightAsArray = T.Height
```

```
heightAsArray = 100×1

    71
    69
    64
    67
    64
    68
    64
    68
    68
    66
     ⋮
```

Use dot notation to return the first three rows of the `Height` variable as an array.

```
firstHeights = T.Height(1:3)
```

```
firstHeights = 3×1

    71
    69
    64
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

Index by Row Names

If a table has row names, you can index into it by row name, not just by row number. For example, return rows of `T` for three specific patients.

```
rowsByName = T(["Griffin","Diaz","Hayes"],:)
```

```
rowsByName=3×4 table
              Age    Height    Weight    Smoker
              ___    ______    ______    ______

    Griffin    49       70       186      false
    Diaz       45       68       172      true
    Hayes      48       66       177      false
```

You can also use curly braces to return the data as an array.

```
arraysFromRows = T{["Griffin","Diaz","Hayes"],:}
```

```
arraysFromRows = 3×4

    49    70   186     0
    45    68   172     1
    48    66   177     0
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

Index into One Element

To index into one element of a table, specify one row and one variable. Use curly braces to return the element as an array, a scalar value in this case.

```
oneElement = T{"Diaz","Height"}
```

```
oneElement = 68
```

To return that element as a table with one row and one variable, use parentheses.

```
oneElementTable = T("Diaz","Height")
```

```
oneElementTable=table
            Height
            ______

    Diaz      68
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

Index by Variable Data Type

To index into a table by specifying variables that have the same data type, create a data type subscript using the `vartype` function.

For example, create a data type subscript to match numeric table variables.

```
numSubscript = vartype("numeric")
```

```
numSubscript =
        table vartype subscript:

                Select table variables matching the type 'numeric'
```

Return a table that has only the numeric variables of `T`. The `Smoker` variable is not included because it is a logical variable.

```
onlyNumVariables = T(:,numSubscript)
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Index by Specifying Rows and Variables

Index by Variable Data Type

```
onlyNumVariables=100×3 table
               Age    Height    Weight
               ___    ______    ______

    Smith      38       71        176
    Johnson    43       69        163
    Williams   38       64        131
    Jones      40       67        133
    Brown      49       64        119
    Davis      46       68        142
    Miller     33       64        142
    Wilson     40       68        180
    Moore      28       68        183
    Taylor     31       66        132
    Anderson   45       68        128
    Thomas     42       66        137
    Jackson    25       71        174
    White      39       72        202
    Harris     36       65        129
    Martin     48       71        181
      ⋮
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Assign Values to Table

You can use any indexing syntax to assign values to a table. You can assign values to variables, rows, or individual elements.

Assign Values to Variables

Import power outage data from a spreadsheet into a table using the `readtable` function.

```
outages = readtable("outages.csv",TextType="string")
```

```
outages=1468×6 table
      Region          OutageTime          Loss      Customers      RestorationTime          Cause
    ___________    ________________    ______    _____________    ________________    _________________

    "SouthWest"    2002-02-01 12:18    458.98    1.8202e+06       2002-02-07 16:50    "winter storm"
    "SouthEast"    2003-01-23 00:49    530.14    2.1204e+05                  NaT      "winter storm"
    "SouthEast"    2003-02-07 21:15     289.4    1.4294e+05       2003-02-17 08:14    "winter storm"
    "West"         2004-04-06 05:44    434.81    3.4037e+05       2004-04-06 06:10    "equipment fault"
    "MidWest"      2002-03-16 06:18    186.44    2.1275e+05       2002-03-18 23:23    "severe storm"
    "West"         2003-06-18 02:49         0             0       2003-06-18 10:54    "attack"
    "West"         2004-06-20 14:39    231.29           NaN       2004-06-20 19:16    "equipment fault"
    "West"         2002-06-06 19:28    311.86           NaN       2002-06-07 00:51    "equipment fault"
    "NorthEast"    2003-07-16 16:23    239.93         49434       2003-07-17 01:12    "fire"
    "MidWest"      2004-09-27 11:09    286.72         66104       2004-09-27 16:37    "equipment fault"
    "SouthEast"    2004-09-05 17:48    73.387         36073       2004-09-05 20:46    "equipment fault"
    "West"         2004-05-21 21:45    159.99           NaN       2004-05-22 04:23    "equipment fault"
    "SouthEast"    2002-09-01 18:22    95.917         36759       2002-09-01 19:12    "severe storm"
    "SouthEast"    2003-09-27 07:32       NaN    3.5517e+05       2003-10-04 07:02    "severe storm"
    "West"         2003-11-12 06:12    254.09    9.2429e+05       2003-11-17 02:04    "winter storm"
    "NorthEast"    2004-09-18 05:54         0             0                  NaT      "equipment fault"
      ⋮
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Assign Values to Table

To assign values to one variable, use dot notation. For example, scale the `Loss` variable by a factor of 100.

```
outages.Loss = outages.Loss .* 100
```

`outages=1468×6 table`

Region	OutageTime	Loss	Customers	RestorationTime	Cause
"SouthWest"	2002-02-01 12:18	45898	1.8202e+06	2002-02-07 16:50	"winter storm"
"SouthEast"	2003-01-23 00:49	53014	2.1204e+05	NaT	"winter storm"
"SouthEast"	2003-02-07 21:15	28940	1.4294e+05	2003-02-17 08:14	"winter storm"
"West"	2004-04-06 05:44	43481	3.4037e+05	2004-04-06 06:10	"equipment fault"
"MidWest"	2002-03-16 06:18	18644	2.1275e+05	2002-03-18 23:23	"severe storm"
"West"	2003-06-18 02:49	0	0	2003-06-18 10:54	"attack"
"West"	2004-06-20 14:39	23129	NaN	2004-06-20 19:16	"equipment fault"
"West"	2002-06-06 19:28	31186	NaN	2002-06-07 00:51	"equipment fault"
"NorthEast"	2003-07-16 16:23	23993	49434	2003-07-17 01:12	"fire"
"MidWest"	2004-09-27 11:09	28672	66104	2004-09-27 16:37	"equipment fault"
"SouthEast"	2004-09-05 17:48	7338.7	36073	2004-09-05 20:46	"equipment fault"
"West"	2004-05-21 21:45	15999	NaN	2004-05-22 04:23	"equipment fault"
"SouthEast"	2002-09-01 18:22	9591.7	36759	2002-09-01 19:12	"severe storm"
"SouthEast"	2003-09-27 07:32	NaN	3.5517e+05	2003-10-04 07:02	"severe storm"
"West"	2003-11-12 06:12	25409	9.2429e+05	2003-11-17 02:04	"winter storm"
"NorthEast"	2004-09-18 05:54	0	0	NaT	"equipment fault"
⋮					

ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Assign Values to Table

You can also assign data to multiple variables by using curly braces. The variables must have compatible data types. For example, scale `Loss` and `Customers` by a factor of 1/10,000.

```
outages{:,["Loss","Customers"]} = outages{:,["Loss","Customers"]} ./ 10000
```

```
outages=1468×6 table
      Region          OutageTime          Loss      Customers    RestorationTime          Cause
    ___________    ________________    _______    _________    _________________    _________________

    "SouthWest"    2002-02-01 12:18     4.5898      182.02      2002-02-07 16:50    "winter storm"
    "SouthEast"    2003-01-23 00:49     5.3014      21.204                 NaT      "winter storm"
    "SouthEast"    2003-02-07 21:15      2.894      14.294      2003-02-17 08:14    "winter storm"
    "West"         2004-04-06 05:44     4.3481      34.037      2004-04-06 06:10    "equipment fault"
    "MidWest"      2002-03-16 06:18     1.8644      21.275      2002-03-18 23:23    "severe storm"
    "West"         2003-06-18 02:49          0           0      2003-06-18 10:54    "attack"
    "West"         2004-06-20 14:39     2.3129         NaN      2004-06-20 19:16    "equipment fault"
    "West"         2002-06-06 19:28     3.1186         NaN      2002-06-07 00:51    "equipment fault"
    "NorthEast"    2003-07-16 16:23     2.3993      4.9434      2003-07-17 01:12    "fire"
    "MidWest"      2004-09-27 11:09     2.8672      6.6104      2004-09-27 16:37    "equipment fault"
    "SouthEast"    2004-09-05 17:48    0.73387      3.6073      2004-09-05 20:46    "equipment fault"
    "West"         2004-05-21 21:45     1.5999         NaN      2004-05-22 04:23    "equipment fault"
    "SouthEast"    2002-09-01 18:22    0.95917      3.6759      2002-09-01 19:12    "severe storm"
    "SouthEast"    2003-09-27 07:32         NaN      35.517      2003-10-04 07:02    "severe storm"
    "West"         2003-11-12 06:12     2.5409      92.429      2003-11-17 02:04    "winter storm"
    "NorthEast"    2004-09-18 05:54          0           0                 NaT      "equipment fault"
      ⋮
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Assign Values to Table

Assign Values to Rows

To assign one row to a table, you can use either a one-row table or a cell array. In this case, using a cell array can be more convenient than creating and assigning a one-row table.

For example, assign data to a new row at the end of `outages`. Display the end of the table.

```
outages(end+1,:) = {"East",datetime("now"),17.3,325,datetime("tomorrow"),"unknown"};
outages(end-2:end,:)
```

```
ans=3×6 table
      Region          OutageTime          Loss      Customers    RestorationTime          Cause
    ___________    ________________    ________    _________    _________________    _______________

    "SouthEast"    2013-12-20 19:52    0.023096     0.10382     2013-12-20 23:29    "thunder storm"
    "SouthEast"    2011-09-14 11:55     0.45042      1.1835     2011-09-14 13:28    "equipment fault"
    "East"         2024-02-13 00:34        17.3         325     2024-02-14 00:00    "unknown"
```

To assign data to multiple rows, assign values from another table that has variables with the same names and data types. For example, create a new two-row table.

```
newOutages = table(["West";"North"], ...
                   datetime(2024,1,1:2)', ...
                   [3;4], ...
                   [300;400], ...
                   datetime(2024,1,3:4)',["unknown";"unknown"], ...
                   VariableNames=outages.Properties.VariableNames)
```

```
newOutages=2×6 table
    Region      OutageTime      Loss    Customers    RestorationTime      Cause
    _______    ____________    ____    _________    _______________    _________

    "West"     01-Jan-2024      3          300        03-Jan-2024      "unknown"
    "North"    02-Jan-2024      4          400        04-Jan-2024      "unknown"
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Assign Values to Table

Assign Values to Rows

Assign the two-row table to the first two rows of outages. Then display the first four rows of outages.

```
outages(1:2,:) = newOutages;
outages(1:4,:)
```

```
ans=4×6 table
      Region          OutageTime          Loss      Customers    RestorationTime          Cause
    ___________    ________________    ______    _________    _________________    _________________

    "West"         2024-01-01 00:00         3          300    2024-01-03 00:00    "unknown"
    "North"        2024-01-02 00:00         4          400    2024-01-04 00:00    "unknown"
    "SouthEast"    2003-02-07 21:15     2.894       14.294    2003-02-17 08:14    "winter storm"
    "West"         2004-04-06 05:44    4.3481       34.037    2004-04-06 06:10    "equipment fault"
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Assign Values to Table

Assign Values to Elements

To assign values to elements of a table, use curly braces. For example, assign causes for the first two outages.

```
outages{1,"Cause"} = "severe storm";
outages{2,"Cause"} = "attack";
outages(1:4,:)
```

```
ans=4×6 table
      Region          OutageTime          Loss      Customers    RestorationTime          Cause
    ___________    ________________    _______    _________    _________________    ________________

    "West"         2024-01-01 00:00          3          300    2024-01-03 00:00     "severe storm"
    "North"        2024-01-02 00:00          4          400    2024-01-04 00:00     "attack"
    "SouthEast"    2003-02-07 21:15      2.894       14.294    2003-02-17 08:14     "winter storm"
    "West"         2004-04-06 05:44     4.3481       34.037    2004-04-06 06:10     "equipment fault"
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Find Table Rows Where Values Meet Conditions

To find the rows of a table where values meet conditions, use logical indexing. Specify the table variables that have values of interest and create an array of row indices where values in those variables meet conditions that you specify. Index into the table using the row indices.

First, import power outage data from a spreadsheet into a table.

```
outages = readtable("outages.csv",TextType="string")
```

outages=*1468×6 table*

Region	OutageTime	Loss	Customers	RestorationTime	Cause
"SouthWest"	2002-02-01 12:18	458.98	1.8202e+06	2002-02-07 16:50	"winter storm"
"SouthEast"	2003-01-23 00:49	530.14	2.1204e+05	NaT	"winter storm"
"SouthEast"	2003-02-07 21:15	289.4	1.4294e+05	2003-02-17 08:14	"winter storm"
"West"	2004-04-06 05:44	434.81	3.4037e+05	2004-04-06 06:10	"equipment fault"
"MidWest"	2002-03-16 06:18	186.44	2.1275e+05	2002-03-18 23:23	"severe storm"
"West"	2003-06-18 02:49	0	0	2003-06-18 10:54	"attack"
"West"	2004-06-20 14:39	231.29	NaN	2004-06-20 19:16	"equipment fault"
"West"	2002-06-06 19:28	311.86	NaN	2002-06-07 00:51	"equipment fault"
"NorthEast"	2003-07-16 16:23	239.93	49434	2003-07-17 01:12	"fire"
"MidWest"	2004-09-27 11:09	286.72	66104	2004-09-27 16:37	"equipment fault"
"SouthEast"	2004-09-05 17:48	73.387	36073	2004-09-05 20:46	"equipment fault"
"West"	2004-05-21 21:45	159.99	NaN	2004-05-22 04:23	"equipment fault"
"SouthEast"	2002-09-01 18:22	95.917	36759	2002-09-01 19:12	"severe storm"
"SouthEast"	2003-09-27 07:32	NaN	3.5517e+05	2003-10-04 07:02	"severe storm"
"West"	2003-11-12 06:12	254.09	9.2429e+05	2003-11-17 02:04	"winter storm"
"NorthEast"	2004-09-18 05:54	0	0	NaT	"equipment fault"
⋮					

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Find Table Rows Where Values Meet Conditions

Next, create row indices that match the rows where a variable meets a condition. For example, create indices for rows where the region is `West`.

```
rows = matches(outages.Region,"West")
```

```
rows = 1468x1 logical array

   0
   0
   0
   1
   0
   1
   1
   1
   0
   0
   ⋮
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Find Table Rows Where Values Meet Conditions

You can index into a table with logical indices. Display the rows of the table for the outages that occur in the `West` region.

```
outages(rows,:)
```

```
ans=354×6 table
    Region       OutageTime          Loss      Customers     RestorationTime          Cause
    ______    ________________    ______    _________    ________________    _________________

    "West"    2004-04-06 05:44    434.81    3.4037e+05    2004-04-06 06:10    "equipment fault"
    "West"    2003-06-18 02:49         0             0    2003-06-18 10:54    "attack"
    "West"    2004-06-20 14:39    231.29           NaN    2004-06-20 19:16    "equipment fault"
    "West"    2002-06-06 19:28    311.86           NaN    2002-06-07 00:51    "equipment fault"
    "West"    2004-05-21 21:45    159.99           NaN    2004-05-22 04:23    "equipment fault"
    "West"    2003-11-12 06:12    254.09    9.2429e+05    2003-11-17 02:04    "winter storm"
    "West"    2004-12-21 18:50    112.05     7.985e+05    2004-12-29 03:46    "winter storm"
    "West"    2002-12-16 13:43    70.752    4.8193e+05    2002-12-19 09:38    "winter storm"
    "West"    2005-06-29 08:37    601.13         32005    2005-06-29 08:57    "equipment fault"
    "West"    2003-04-14 07:11    276.41        1.5647    2003-04-14 08:52    "equipment fault"
    "West"    2003-10-21 17:25    235.12         51496    2003-10-21 19:43    "equipment fault"
    "West"    2005-10-21 08:33       NaN         52639    2005-11-22 22:10    "fire"
    "West"    2003-08-28 23:46    172.01    1.6964e+05    2003-09-03 02:10    "wind"
    "West"    2005-03-01 14:39    115.47         82611    2005-03-03 05:58    "equipment fault"
    "West"    2005-09-26 06:32    258.18    1.3996e+05    2005-09-26 06:33    "earthquake"
    "West"    2003-12-22 03:40    232.26    3.9462e+05    2003-12-24 16:32    "winter storm"
      ⋮
```

ESERCITAZIONI
MATLAB – Nozioni fondamentali

MathWorks®

Matrici e array

Argomenti

Indicizzazione di array

Access Data in Tables

Find Table Rows Where Values Meet Conditions

You can match multiple conditions with one logical expression. For example, find the rows where outages affected more than one million customers in the `West` or `MidWest` regions.

```
rows = (outages.Customers > 1e6 & (matches(outages.Region,"West") | matches(outages.Region,"MidWest")));
outages(rows,:)
```

```
ans=10×6 table
     Region        OutageTime        Loss      Customers     RestorationTime          Cause
    _________    _________________    ______    _________    _________________    _________________

    "MidWest"    2002-12-10 10:45     14493     3.0879e+06    2002-12-11 18:06    "unknown"
    "West"       2007-10-20 20:56    3537.5     1.3637e+06    2007-10-20 22:08    "equipment fault"
    "West"       2006-12-28 14:04    804.05     1.5486e+06    2007-01-04 14:26    "severe storm"
    "MidWest"    2006-07-16 00:05    1817.9      3.295e+06    2006-07-27 14:42    "severe storm"
    "West"       2006-01-01 11:54    734.11       4.26e+06    2006-01-11 01:21    "winter storm"
    "MidWest"    2008-09-19 23:31    4801.1     1.2151e+06    2008-10-03 14:04    "severe storm"
    "MidWest"    2008-09-07 23:35       NaN      3.972e+06    2008-09-19 17:19    "severe storm"
    "West"       2011-07-24 02:54    483.37     1.1212e+06    2011-07-24 12:18    "wind"
    "West"       2010-01-24 18:47    348.91     1.8865e+06    2010-01-30 01:43    "severe storm"
    "West"       2010-05-17 09:10    8496.6     2.0768e+06    2010-05-18 22:43    "equipment fault"
```

Riferimenti Bibliografici

[1] https://it.mathworks.com