Tables

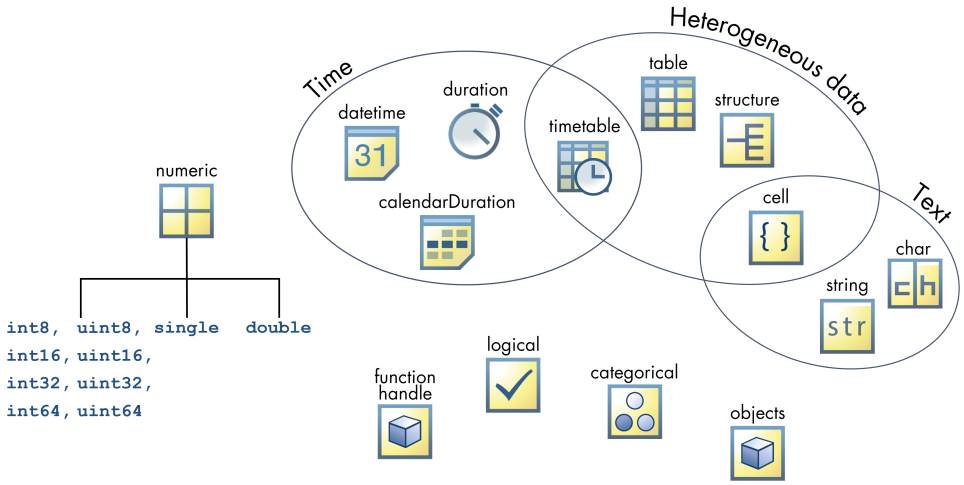
Work with Tables

Introduction

MATLAB Data Types

The different data types in MATLAB have different uses.

Each data type is intended to store data with a characteristic organization. In this course, you use the `table` data type to store tabular data where the rows are observations, which all have the same set of measured variables, but the different variables (columns) can be of different types.



Summary

Summary: Work with Tables

Store Data in a Table

The [readtable](#) function creates a table in MATLAB from a data file.

The [table](#) function can create a table from workspace variables.

```
EPL = readtable("EPLresults.csv",TextType="string");


teamWins = table(team,wins)
```

```
teamWins =

            Team              Wins
    ____________________      ____

    "Arsenal"                  20
    "Chelsea"                  12
    "Leicester City"           23
    "Manchester United"        19
```

The `array2table` function can convert a numeric array to a table. Specify the `VariableNames` name-value argument as a string array of variable names for the table.

```
stats = array2table(WDL, ...
    VariableNames=["Wins" "Draws" "Losses"])
```

```
stats =

    Wins    Draws    Losses
    ____    _____    ______

     20       11        7
     12       14       12
     23       12        3
     19        9       10
```

The `summary` function displays summary statistics for variables in a table.

```
summary(EPL)
```

Sort Table Data

The `sortrows` function sorts the table by the specified variable in ascending order, by default.

Use the optional `"descend"` parameter to sort the table in descending order.

You can also sort by multiple variables, in order, by specifying a string array of variable names.

```
EPL = sortrows(EPL,"HomeWins");
```

```
EPL = sortrows(EPL,"HomeWins","descend");
```

```
EPL = sortrows(EPL,["HomeWins" "AwayWins"],"descend");
```

Extract Portions of a Table

Display the original table.

To extract a portion of the table, index into it by using parentheses and the table row and variable numbers.

```
EPL
```

```
EPL =
          Team              HW    HD    HL    AW    AD    AL
    ____________________    __    __    __    __    __    __
    "Leicester City"        12     6     1    11     6     2
    "Arsenal"               12     4     3     8     7     4
    "Manchester City"       12     2     5     7     7     5
    "Manchester United"     12     5     2     7     4     8
    "Southampton"           11     3     5     7     6     6
    "Tottenham Hotspur"     10     6     3     9     7     3
    "West Ham United"        9     7     3     7     7     5
```

```
EPL(2:4,[1 2 5])
```

You can also index using the variable names. To reference more than one variable, use a string array containing the variable names.

```
ans =
            Team             HW    AW
    ____________________    __    __
    "Arsenal"               12    8
    "Manchester City"       12    7
    "Manchester United"     12    7
```

```
EPL(2:4,["Team" "HW" "AW"])
```

```
ans =
            Team             HW    AW
    ____________________    __    __
    "Arsenal"               12     8
    "Manchester City"       12     7
    "Manchester United"     12     7
```

Access Data in a Table

Display the original table.

```
EPL
```

```
EPL =
            Team             HW    HD    HL    AW    AD    AL
    ____________________    __    __    __    __    __    __
    "Leicester City"        12     6     1    11     6     2
    "Arsenal"               12     4     3     8     7     4
    "Manchester City"       12     2     5     7     7     5
    "Manchester United"     12     5     2     7     4     8
```

You can use dot notation to access data for use in calculations or plotting.

```
tw = EPL.HW + EPL.AW
```

```
tw =
    23
    20
    19
    19
```

You can also use dot notation to create new variables in a table.

```
EPL.TW = EPL.HW + EPL.AW
```

```
EPL =
            Team             HW    HD    HL    AW    AD    AL    TW
    ____________________    __    __    __    __    __    __    __
    "Leicester City"        12     6     1    11     6     2    23
    "Arsenal"               12     4     3     8     7     4    20
    "Manchester City"       12     2     5     7     7     5    19
    "Manchester United"     12     5     2     7     4     8    19
```

To extract multiple variables, index using curly braces.

```
draws = EPL{:,["HD" "AD"]}
```

Specify row indices to extract specific rows.

```
draws =
     6     6
     4     7
     2     7
     5     4


draws13 = EPL{[1 3],["HD" "AD"]}
draws13 =
     6     6
     2     7
```

Export Tables

You can use the `writetable` function to create a file that contains the table data.

```
writetable(tableName,"myFile.txt",Delimiter="\t")
```

The file format is based on the specified file extension, such as `.txt`, `.csv`, or `.xlsx`. For delimited text files, you can also specify a delimiter.

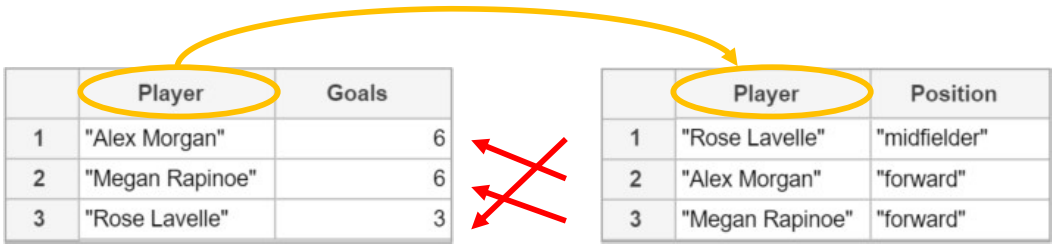
Manage Tables of Data

Summary

Summary: Manage Tables of Data

Combine Tables

You can combine two tables by merging them with a join. In a join, rows with matching values in common variables are merged.



	Player	Goals
1	"Alex Morgan"	6
2	"Megan Rapinoe"	6
3	"Rose Lavelle"	3

uswntTop3

	Player	Position
1	"Rose Lavelle"	"midfielder"
2	"Alex Morgan"	"forward"
3	"Megan Rapinoe"	"forward"

posTop3

The `join` function combines tables with a common variable.

```
top3 = join(uswntTop3,posTop3)
top3 =
          Player          Goals        Position
    ____________________    _____    ____________

    "Alex Morgan"             6      "forward"
    "Megan Rapinoe"           6      "forward"
    "Rose Lavelle"            3      "midfielder"
```

Table Properties

Display the table properties.

```
EPL.Properties
```

```
ans =
  Table Properties with properties:

            Description: ''
               UserData: []
         DimensionNames: {'Row'  'Variable'}
          VariableNames: {1×11 cell}
    VariableDescriptions: {1×11 cell}
          VariableUnits: {}
      VariableContinuity: []
                RowNames: {}
       CustomProperties: No custom properties are set.
```

You can access an individual property of `Properties` using dot notation.

```
EPL.Properties.VariableNames
```

```
ans =
  1×11 cell array
  Columns 1 through 4
    {'Team'}    {'HomeWins'}    {'HomeDraws'}    {'HomeLosses'}
  Columns 5 through 8
    {'HomeGF'}    {'HomeGA'}    {'AwayWins'}    {'AwayDraws'}
  Columns 9 through 11
    {'AwayLosses'}    {'AwayGF'}    {'AwayGA'}
```

Manage Table Variables

`convertvars`	*Convert variables to specified data type.*
`renamevars`	*Rename variables.*
`movevars`	*Reorder variables.*
`removevars`	*Remove variables.*

You can use text functions, such as `contains`, `startsWith`, `endsWith`, and `replace`, to identify and modify variables based on patterns in variable names. For more information about text functions, see Search and Replace Text in the documentation.