Preprocessing Time Series Data with MATLAB

This reference shows common use cases, but is by no means comprehensive. The icon provides links to relevant sections of the MATLAB® documentation to learn more.

### Timetable

MATLAB datatype designed to organize and work with time series data.

#### Components of a Timetable

- **Date**: Double, 1x1 logical, char, cell, and other arrays
- **duration**: Offset from a reference time
- **datetime**: Absolute timestamp—e.g., year/month/day

#### Create Timetables

```matlab
tt = timetable(times, var1, var2, ... ,varN);
```

(All variables must have the same number of rows.)

```matlab
tt = table2timetable(t);
```

(The first datetime or duration variable in "t" becomes the row times.)

### Timetable Manipulation

#### Access Data

These return the same array:

```matlab
tt.Temperature
tt(:, 'Temperature')
```

#### Add a New Variable

```matlab
tt.newVar = zeros(height(tt), 1);
```

#### Change Variable Names

```matlab
tt.properties.VariableNames = newNames;
```

(Names must be valid MATLAB identifiers)

Tip: Use `matlab.lang.makevalidname` to create valid names from potentially invalid names.

### Data Cleaning

#### Smooth Data

```matlab
B = smoothdata(A, method);
```

Smooth noisy data with methods:

- `'movmean'`, `'movmedian'`, `'gaussian'`, `'lowess'`, `'loess'`, `'rlowess'`, `'rloess'`, `'sgolay'`

#### Detect Outliers

```matlab
TF = isoutlier(A, method);
```

Identify outliers with methods:

- `'median'`, `'mean'`, `'quartiles'`, `'grubbs'`, `'gesd'`

#### Detect Change Points

```matlab
TF = ischange(A, method);
```

Find abrupt changes with methods:

- `'mean'`, `'variance'`, `'linear'`

### Merge Timetables

Synchronize multiple timetables to a common time vector.

```matlab
tt = synchronize(tt1, tt2, ..., ttN);
```

(All variables must have the same number of rows.)

### Missing Data

#### Find Missing Values

```matlab
TF = ismissing(tt);
```

#### Fill Missing Values

```matlab
tt = fillmissing(tt, method);
```

Replace missing values with values calculated from nearby points with methods:

- `'previous'`, `'next'`, `'nearest'`
- `'linear'`, `'spline'`, `'pchip'`

### Big Data

Tall arrays extend MATLAB functions to work on data too big to load into memory.

#### Create a "tall" timetable:

```matlab
ds = datastore(*.csv');
```

Create a tall table from the datastore:

```matlab
t = tall(ds);
```

Convert to a timetable:

```matlab
tt = tall2timetable(t);
```