

## クイックスタートガイド

# MATLABを使った時系列データの前処理

このリファレンスでは一般的な使用例をいくつかご紹介いたします。

➤ をクリックするとより詳しい内容を MATLAB® ドキュメンテーションで確認することができます。

## Timetable

Timetable は時系列データを編成、処理するためのデータ型です。

### Timetable の内容

Times (2 options)		Variable Names				Data
datetime (absolute timestamp—e.g., year/month/day)	duration (offset from a reference time)	Time	Temperature	Humidity	Precipitation	Wind
		1	2	3	4	
		1	2	3	4	5
		2	46	35	Overcast	47
		3	44	35	Overcast	47
		4	43	34	Cloudy	44
		5	40	34	Cloudy	43
		6	44	35	Cloudy	44
		7	48	35	Cloudy	44
		8	50	32	Cloudy	44
		9	52	31	Cloudy	44
		10	54	30	Overcast	44
		11	56	28	Sunny	44
		12	58	29	Sunny	44
		13	62	27	Sunny	44
		14	62	27	Sunny	44
		15	61	23	Sunny	48

### Timetable の作成 ➤

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

(各変数の行数が同じである必要があります)

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

(t 内の最初の datetime または duration 変数が tt の行時間のベクトルになります)

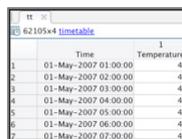
## Timetable の操作

### データへのアクセス

```
tt.Temperature
```

```
tt{:,'Temperature'}
```

```
tt{: ,1} ➤
```



### 新しい変数の追加

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

### 変数名の変更

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

(変数名は有効な MATLAB 識別子でなければなりません)

ヒント: `matlab.lang.makevalidname` を使って無効な文字から有効な変数名を作ります。

### Retime を使用してデータをリサンプリング

```
tt = retime(tt,newtimes,method); ➤
```

内挿方法は `method` で指定します。これは `synchronize` 関数でも同じです。(下記の「Timetableの結合」を参照)

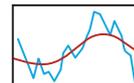
## データクリーニング

### データの平滑化 ➤

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

ノイズの多いデータの平滑化方法:

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

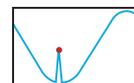


### 外れ値の検出 ➤

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

外れ値の検出方法

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

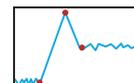


### 変更点の検出 ➤

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

変更点の検出方法:

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



## Timetable の結合

複数の timetable を共通の時間ベクトルに同期します。

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

Synchronize で同期した場合、データが測定されていない箇所は欠損データと見なされます。欠損データの処理方法を調整するには以下の方法があります。

➤

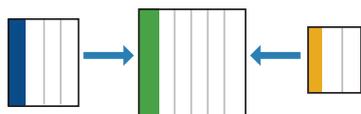
### 埋め込み:

```
'fillwithmissing','fillwithconstant'
```

内挿: 'linear','spline','pchip'

最近傍: 'previous','next','nearest'

集約: 'mean','min','max',@func,...



## 欠損値の処理

### 欠損値の検出

```
TF = ismissing(tt); ➤
```

### 欠損値の埋め込み

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

以下から指定した方法で前後のデータの欠損値を埋めます。

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

### 欠損値を含む不完全な行の削除

```
tt = rmmissing(tt);
```

Time	1	2
01-May-2007 01:00:00	Temperature 47	Humidity 35
01-May-2007 02:00:00	46	NaN
01-May-2007 03:00:00	NaN	35
01-May-2007 04:00:00	NaN	34
01-May-2007 05:00:00	40	34
01-May-2007 06:00:00	44	35
01-May-2007 07:00:00	48	35

## ビッグデータの処理

Tall 配列は大きすぎてメモリに収まらないデータを MATLAB 関数で操作できるようにします。

### Tall timetable の作成:

```
% データストアの作成
```

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

```
% データストアから tall テーブルの作成
```

```
t = tall(ds); ➤
```

```
% tall テーブルから tall timetable への変換
```

```
tt = table2timetable(t); ➤
```

Mx5 tall timetable					
Time	LATP	LONP	ALT	PTCH	ROLL
10-May-2001 16:24:12	39.055	-84.661	866	-0.37352	0.076902
10-May-2001 16:24:12	NaN	NaN	NaN	-0.37352	0.076902
10-May-2001 16:24:12	NaN	NaN	866	-0.37352	0.076902
10-May-2001 16:24:12	NaN	NaN	NaN	-0.37352	0.076902
10-May-2001 16:24:12	NaN	NaN	866	-0.37352	0.076902
10-May-2001 16:24:12	NaN	NaN	NaN	-0.37352	0.076902
10-May-2001 16:24:12	NaN	866	NaN	-0.37352	0.076902
10-May-2001 16:24:12	NaN	NaN	NaN	-0.37352	0.076902
:	:	:	:	:	:
:	:	:	:	:	: