

知って得する！データサイエンスを快適にする MATLAB基本テクニック

MathWorks Japan
アプリケーションエンジニアリング部
田口美紗

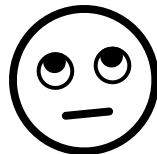
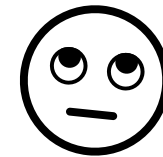
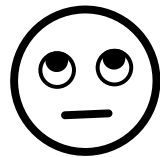
データサイエンス？

ニューラルネット？

K-NN？

ランダムフォレスト？

SVM？



ワークフロー全体を考える

データアクセス



センサー



ファイル



データベース

データ解析



データ 探索



前処理



専門的な
アルゴリズム

開発



AI モデル



アルゴリズム
開発



モデリング &
シミュレーショ
ン

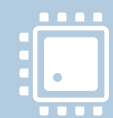
実装&展開



デスクトップ
アプリ



エンタープライ
ズシステム



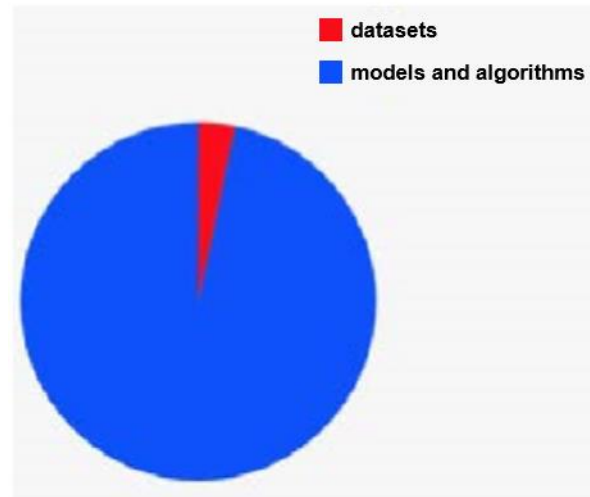
組み込み
デバイス

AIの作業で最も労力を要しているデータの準備…

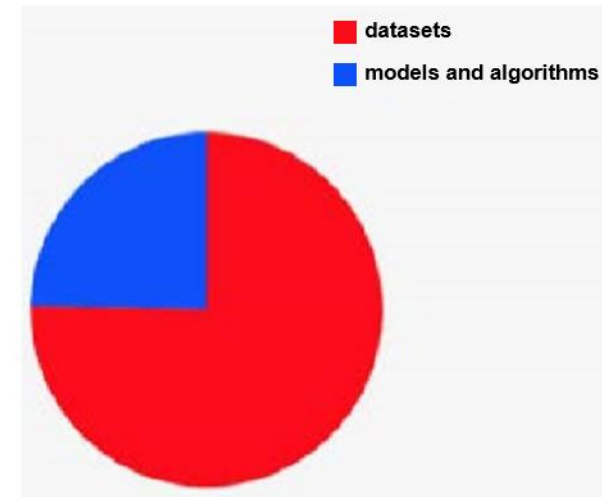
生データをモデリングや解析のために変換することは非常に重要なステップです。

何に時間をかけるか… データセット
モデル・アルゴリズム

PhD



Tesla



Source: Andrej Karpathy slide from TrainAI 2018

データサイエンス成功への鍵:

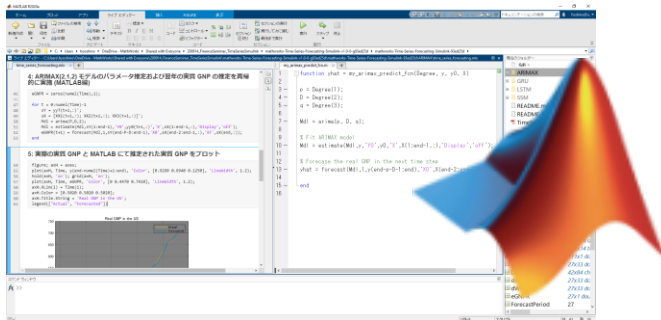
```
# -*- coding: utf-8 -*-
"""
Created on Wed Aug 19 14:04:47 2020

@author: hyoshino
"""

import unittest

def fun(x):
    return x + 1

class MyTest(unittest.TestCase):
    def test(self):
        self.assertEqual(fun(3), 4)
```



本質的な課題

非本質的作業:

ツールの使い方・コマンド、プログラミング、ライブラリの整合性、実装、バージョン管理、知財…

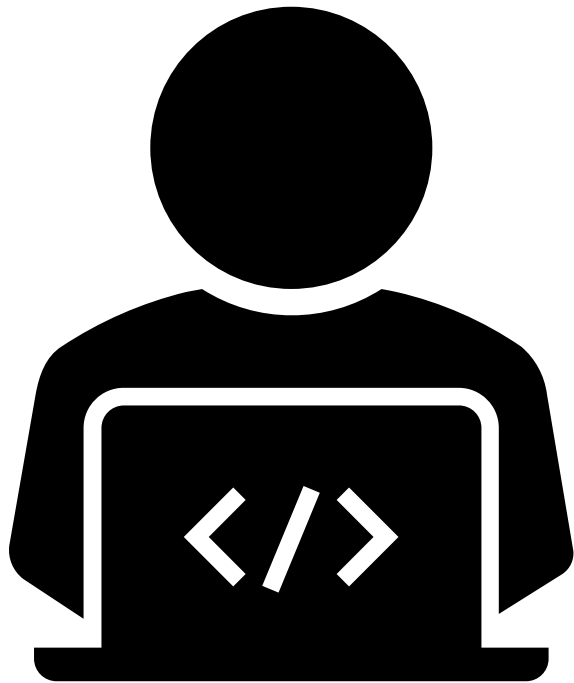
本質的な課題

最小限の
非本質的作業

改良 / 次のプロジェクト

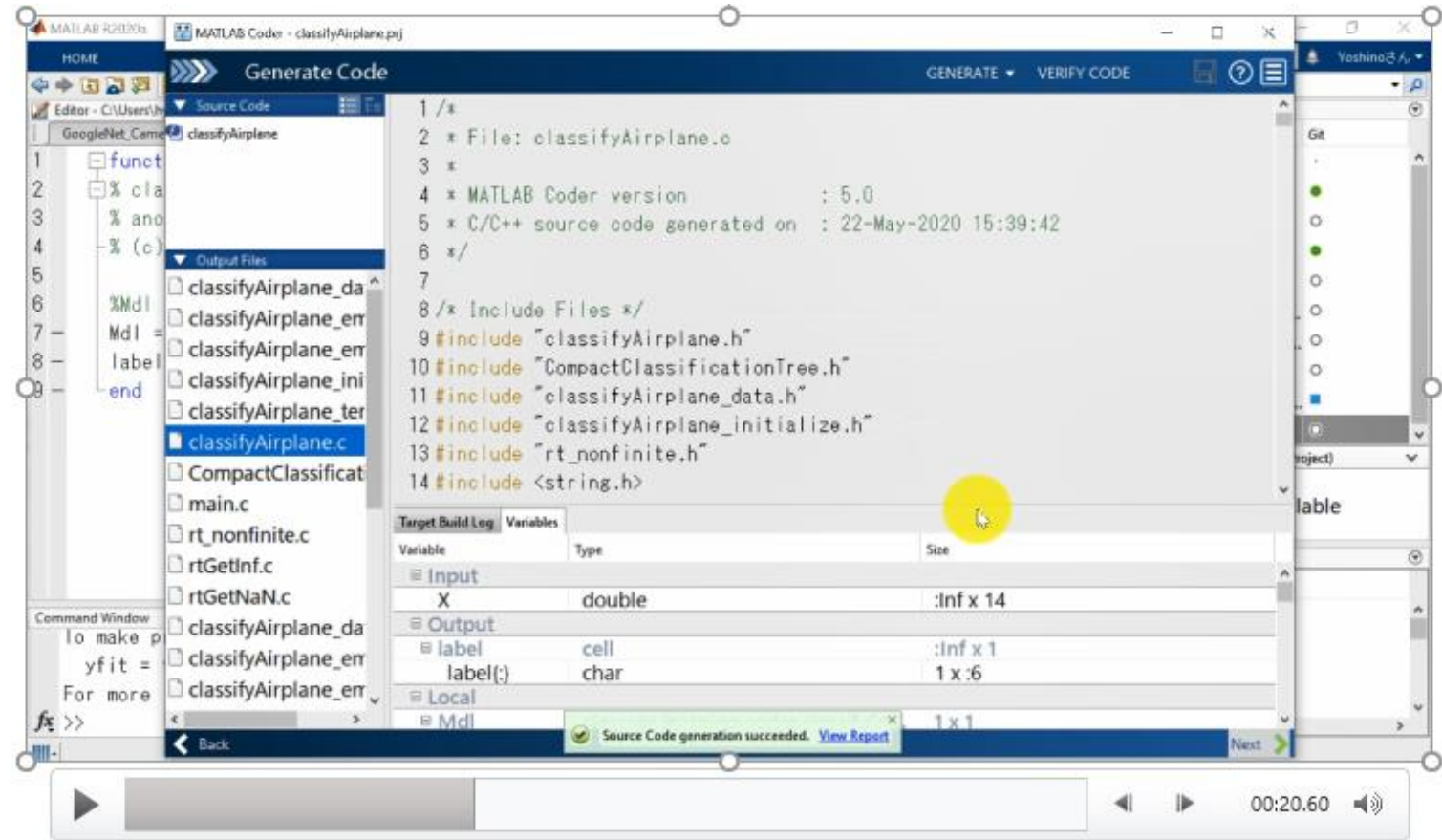
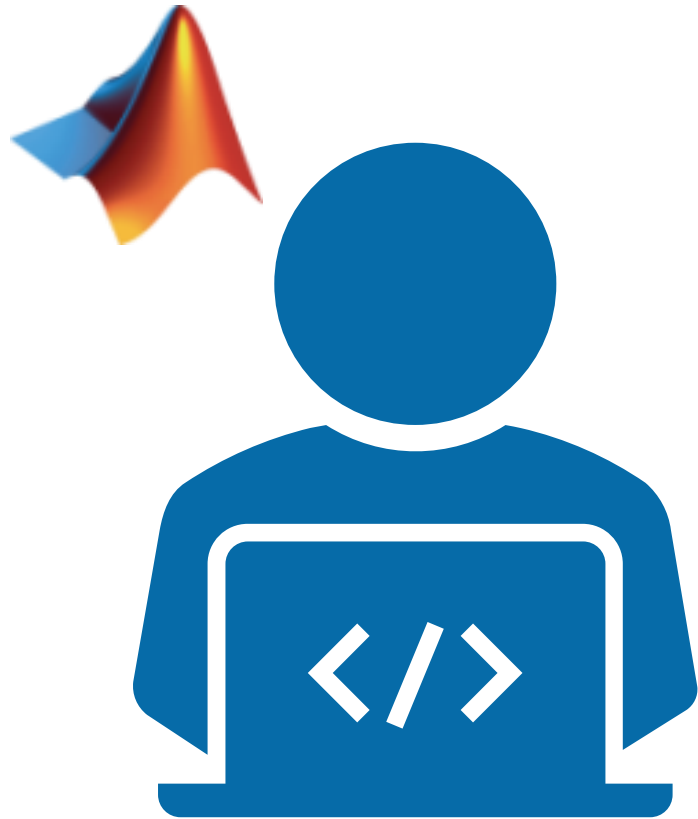
accelerating the pace of discovery, innovation, development, and learning in engineering and science

天才エンジニアの先輩の話



- 音声合成の研究者
- AIアルゴリズムをオープンソースプラットフォームを使って開発
- C++, C の使い手
- アルゴリズム開発 → ハードウェア実装 (C++) もできてしまう天才

私はこうします



講演のメッセージ

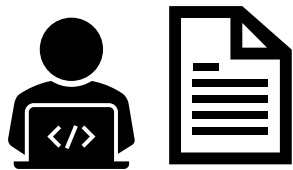
“本質的な課題”の解決
こそが価値があること



本日のアジェンダ

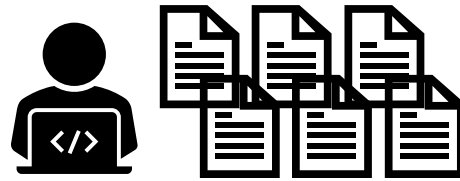
①アドホックな解析

- データ読み込み
- データアクセス
- 前処理
- 可視化



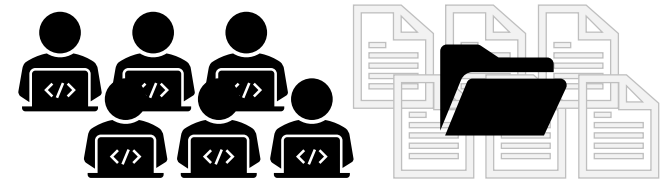
②ビッグデータ解析

- ビッグデータの読み込み
- ビッグデータの解析・深掘り



③プロジェクト化

- プロジェクトの作成・共有
- 大人数で解析

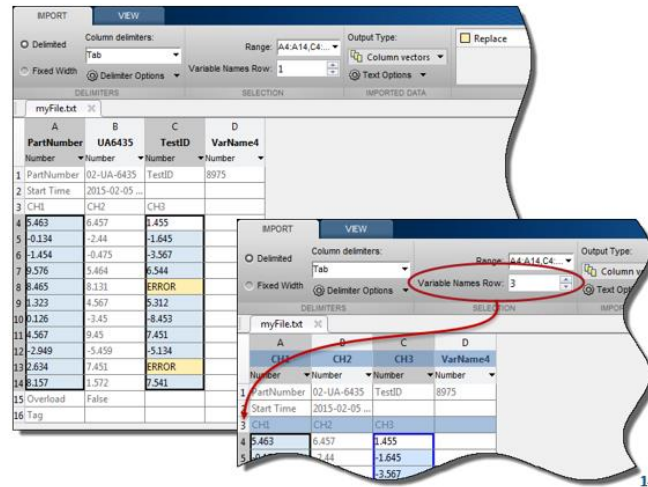


データインポートツールアプリ



データインポートツールアプリ

- GUIで即座にインポート
 - 取り込み範囲の指定
 - 変数型の指定
 - 欠損値処理の指定
- MATLABコード生成機能
 - 2回目以降はコマンド



(1)

[紹介] ライブエディター



- 高機能スクリプトエディター
 - 数式 (LaTeX対応)
 - 文字、画像を埋め込み
 - 計算の出力、グラフを表示
- 半GUIを実装
 - ライブタスク, e.g., 最適化、前処理
 - ライブコントロール, e.g., スライダー
- レポート機能
 - Word, pdf, LaTeX, ...

(2)

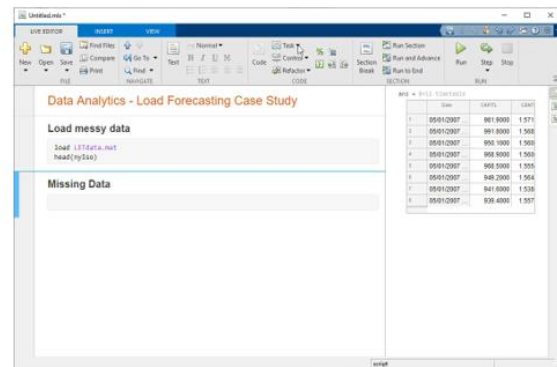


16

ライブタスクから最適な前処理設定を素早く探索



- 半GUI操作で試行錯誤が簡単に
- 対応するMATLABコードを自動生成



22

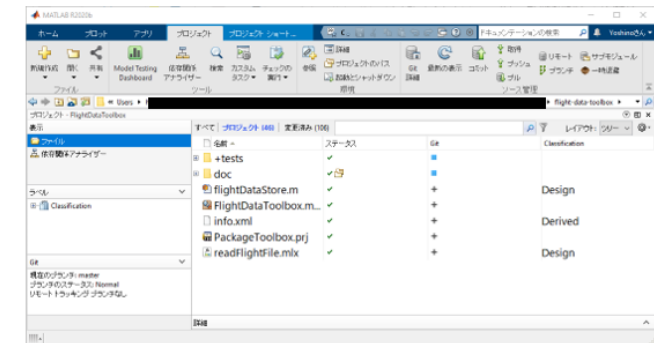
(3)

プロジェクト化
共同作業・バージョン管理

- ファイル、データ、テスト等を1か所で管理
- 作業の整理、チームでの共同作業を円滑にする
- Git™, SVN®を使用したソース管理ツール (GitはMATLABに同梱)
- 依存関係の解析

(4)

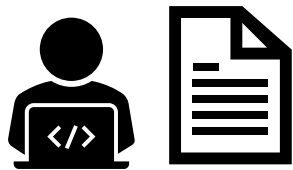
MATLABプロジェクトについて



41

①アドホックな解析

- データ読み込み
- データアクセス
- 前処理
- 可視化



フライトデータの解析例

振動センサ

燃料センサ

緯度・経度



火災センサ

時間

データの読み込み

(例) テキストファイルからのデータの読み込み

低水準インポート関数

- ファイルポインタ
- ループで読み込み
- 詳細に設定が必要

```
filename = 'mymeas.dat';
measrows = 4;
meascols = 4;

% open the file
fid = fopen(filename);

% read the file headers, find N (one value)
N = fscanf(fid, '%*s %*s\nN=%d\n\n', 1);

% read each set of measurements
for n = 1:N
    mystruct(n).mtime = fscanf(fid, '%s', 1);
    mystruct(n).mdate = fscanf(fid, '%s', 1);

    % fscanf fills the array in column order,
    % so transpose the results
    mystruct(n).meas = ...
        fscanf(fid, '%f', [measrows, meascols]);
end

% close the file
fclose(fid);
```

特定ファイル形式 インポート関数

- 比較的簡単に読み込み
- ループ要らず

```
opts = delimitedTextImportOptions("NumVariables", 89);

% 範囲と区切り記号の指定
opts.DataLines = [3, Inf];
opts.Delimiter = ",";

% 列名と型の指定
opts.VariableNames = ["Time", "ABRK", "ACMT", "AIL_1", "AIL_2"];
opts.VariableTypes = ["string", "double", "double", "double"];

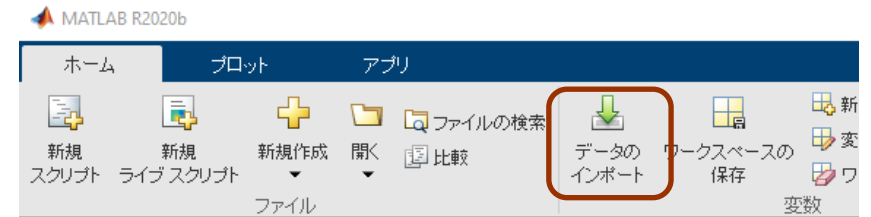
% ファイル レベルのプロパティを指定
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% 変数プロパティを指定
opts = setvaropts(opts, "Time", "WhitespaceRule", "preserve");
opts = setvaropts(opts, "Time", "EmptyFieldRule", "auto");

% データのインポート
flightData1Hz = readtable("mydata.dat", opts);
```

データインポート ツールアプリ

- GUIで読み込み

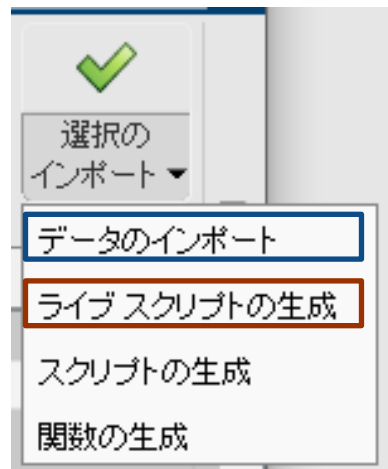


The image shows the 'Import Data' dialog box with 'flightData1Hz.csv' selected. The 'Import Data' button is highlighted. Below the dialog, the resulting table is displayed.

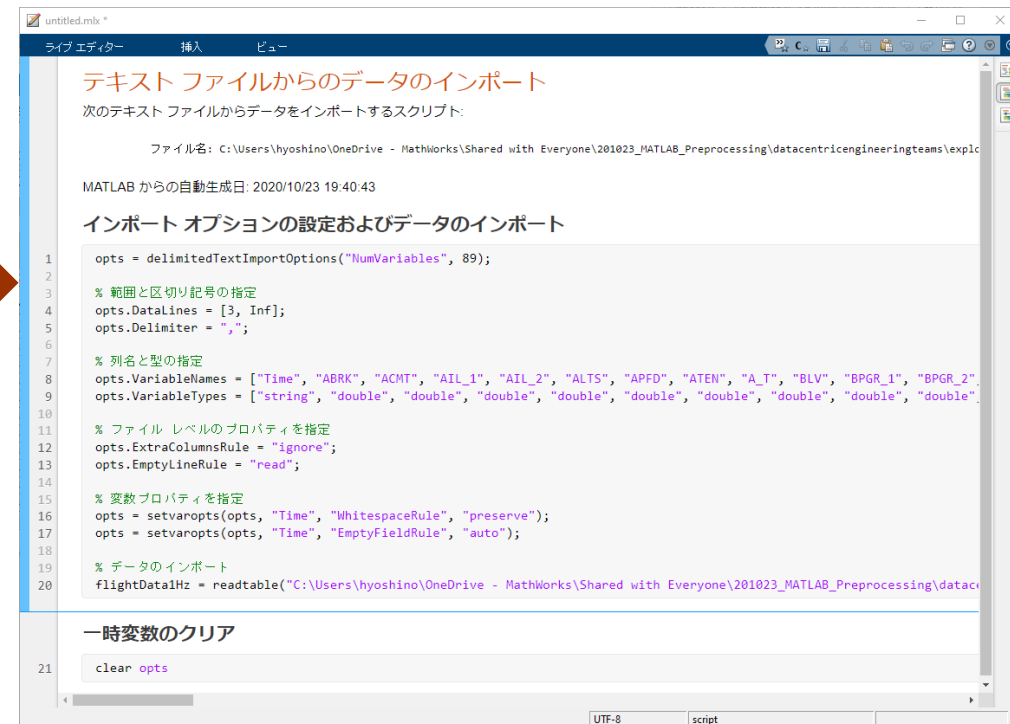
	Time	ABRK	ACMT	AIL_1	AIL_2	ALTS	APFD	ATEN	A.T	BLV	BPGR_1	BPGR_2	BPYR_1	BPYR_2
1	Time	ABRK	ACMT	AIL_1	AIL_2	ALTS	APFD	ATEN	A.T	BLV	BPGR_1	BPGR_2	BPYR_1	BPYR_2
2	DEG	DEG	DEG	DEG	DEG	FEET								
3	10-May-20...	119.983558...	59	91.8788909...	91.5924835...	6000	2	0	1	0	0	43.9454231...	2373.05297...	0
4	10-May-20...	119.983558...	59	91.8993530...	91.5924835...	6000	2	0	1	0	0	43.9454231...	2373.05297...	0
5	10-May-20...	119.983558...	60	91.8993530...	91.5924835...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
6	10-May-20...	119.983558...	59	91.8993530...	91.5720214...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
7	10-May-20...	119.983558...	60	91.8788909...	91.5720214...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
8	10-May-20...	119.983558...	60	91.9607238...	91.5720214...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
9	10-May-20...	119.983558...	59	91.9198150...	91.5515670...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
10	10-May-20...	119.983558...	60	91.8788909...	91.5924835...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
11	10-May-20...	119.983558...	59	91.8175201...	91.5515670...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
12	10-May-20...	119.983558...	60	91.8175201...	91.5311126...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
13	10-May-20...	119.983558...	60	91.8175201...	91.5311126...	6000	2	0	1	0	0	43.9454231...	2368.17016...	0
14	10-May-20...	119.983558...	59	91.8175201...	91.5106506...	6000	2	0	1	0	0	43.9454231...	2363.28735...	0

データ読み込み用のコード生成

- データのインポート
- コードの生成
 - ライブスクリプト
 - スクリプト
 - 関数化



名前	値	クラス	サイズ	バイト
flightData1Hz	4636x89 t...	table	4636x89	3682582



データインポートツールアプリ

データインポートツールアプリ

- GUIで即座にインポート
 - 取り込み範囲の指定
 - 変数型の指定
 - 欠損値処理の指定
- MATLABコード生成機能
 - 2回目以降はコマンド

The top screenshot shows the 'VIEW' tab of the Data Importer tool. The 'Delimited' radio button is selected. The 'Column delimiters' dropdown is set to 'Tab'. The 'Range' is 'A4:A14,C4:...'. The 'Output Type' is 'Column vectors'. The 'Variable Names Row' is set to 1. The data table below shows the following data:

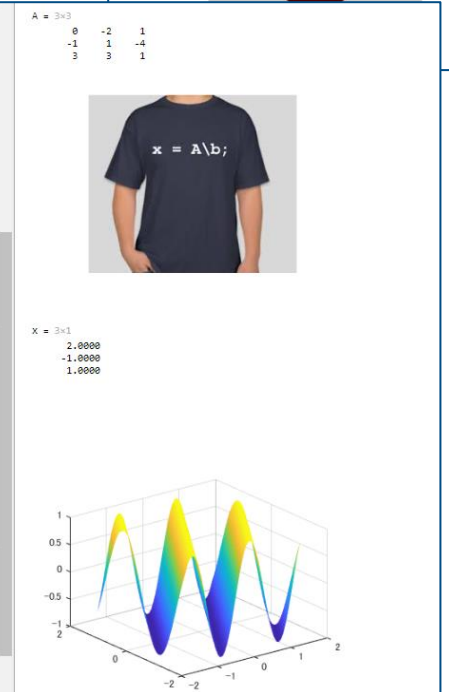
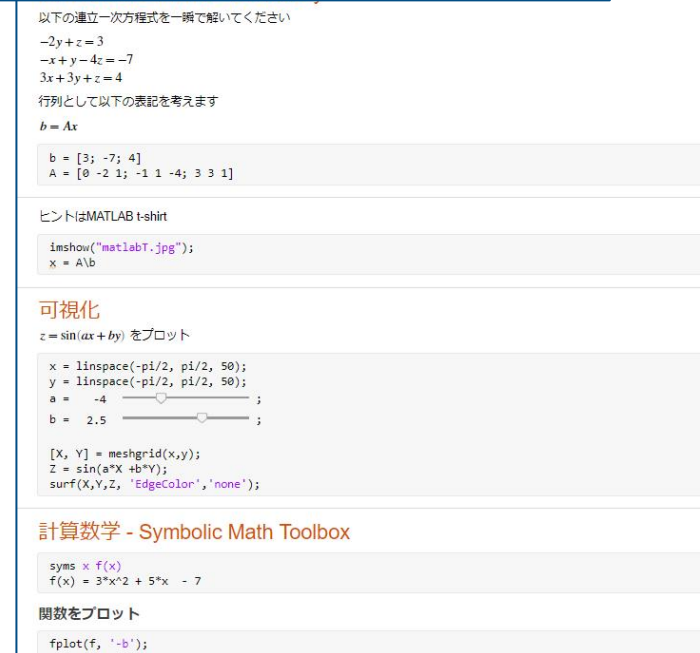
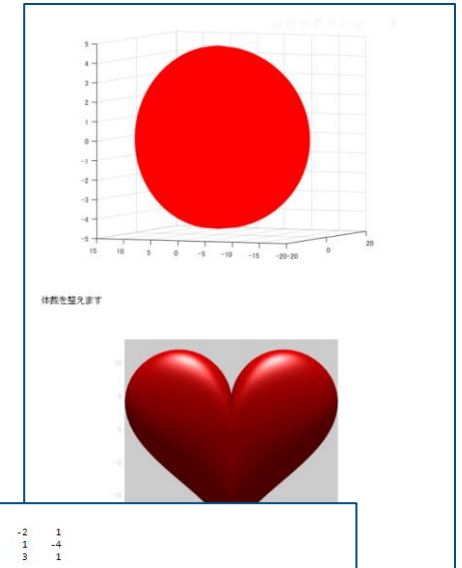
	A	B	C	D
	PartNumber	UA6435	TestID	VarName4
	Number	Number	Number	Number
1	PartNumber	02-UA-6435	TestID	8975
2	Start Time	2015-02-05 ...		
3	CH1	CH2	CH3	
4	5.463	6.457	1.455	
5	-0.134	-2.44	-1.645	
6	-1.454	-0.475	-3.567	
7	9.576	5.464	6.544	
8	8.465	8.131	ERROR	
9	1.323	4.567	5.312	
10	0.126	-3.45	-8.453	
11	4.567	9.45	7.451	
12	-2.949	-5.459	-5.134	
13	2.634	7.451	ERROR	
14	8.157	1.572	7.541	
15	Overload	False		
16	Tag			

The bottom screenshot shows the same tool with 'Variable Names Row' set to 3. A red circle highlights this change. A red arrow points from the 'Variable Names Row' field to the corresponding row in the data table below it.

	A	B	C	D
	CH1	CH2	CH3	VarName4
	Number	Number	Number	Number
1	PartNumber	02-UA-6435	TestID	8975
2	Start Time	2015-02-05 ...		
3	CH1	CH2	CH3	
4	5.463	6.457	1.455	
5	-0.134	-2.44	-1.645	
6	-1.454	-0.475	-3.567	

[紹介] ライブエディター

- 高機能スクリプトエディター
 - 数式 (LaTeX対応)
 - 文字、画像を埋め込み
 - 計算の出力、グラフを表示
- 半GUIを実装
 - ライブタスク, e.g., 最適化、前処理
 - ライブコントロール, e.g., スライダー
- レポート機能
 - Word, pdf, LaTeX, ...



ライブスクリプトで微分方程式論の授業

LiveEditor



微分方程式論(Introduction of Differential Equations)

version 2.0.0 (5.19 MB) by Ben T. Nohara

第1部: 基礎編 第1章: 微分方程式とは 第2章: 求積法, 解析解と数値解 第3章: ベクトル場, 相図: 線形微分方程式と非線形微分方程式の違い, Hartman-Grobman定理 第4章: 数値解法の基礎事項: オイラー法, ... その他

★★★★★ 0 Ratings

6 Downloads

Updated 29 Jul 2019

[View License](#)

[+ Follow](#)

[Download](#)

Overview **Examples**

学問としての微分方程式論の重要性は、諸現象の核心部分を抽出しモデル化して、物理量の動態を把握することにより、これにより未来時間での動態を予測できることにある。インストラクターはつぎの事項に注力する。すなわち、微分方程式の基礎理論をマスターすることは勿論であるが、ともすれば理学部数学科の理論倒れになるのをさけるため、理論と応用のバランスに心がけ（理論のない学問は発展せず、応用のない学問は空虚である）、MATLABを駆使して専門科目（機械工学、電気工学、化学、建築など）の持つ課題解決への発展を目指す。

受講生はこの講義の内容を修得すれば、

- (1) 線形常微分方程式の解の導出や解軌跡を相図で表現できる。
- (2) 解くことが不可能な非線形常微分方程式の厳密解をその線形化により表現できるようになり、大域的な解の動態を把握できる。
- (3) 歴史上重要な方程式（van der Pol方程式やLotka-Volterra方程式など）のモデル化を学び、安定性の概念や非線形の取り扱いを修得できる。
- (4) MATLABを通して微分方程式の数値解法を修得でき、解の正確さを吟味できるようになる。
- (5) エンジニアの将来として、未解決問題を微分方程式によりモデル化が可能になり、技術革新の基盤形成ができる。

ともすれば無味乾燥した微分方程式という対象の面白さを伝えるためにはどうしてもコンピュータ援用学習が必要になります。その点、MATLABのLive Scriptは理にかなった構造と機能を持っています。Live Scriptは説明文（平文）とコードが混在した形態であり、理論を（概念的に）理解した後に、実際にコードを実行させて概念的な理解を具現化するように設計されています。

この講座の原稿はすべてLive Scriptで書いてあり、例や問題を皆さんが実行し、考察した結果を印刷すれば皆さん一人一人の独り占めになるでしょう。Live Scriptというヨットに乗って微分方程式の大海に漕ぎ出してみませんか。皆さんがLive Scriptを操ることにより、安全な航海ができ微分方程式の提供する実りある目的地にきっと入港できるでしょう。Bon voyage! どうぞ、よい!

Cite As

Ben T. Nohara (2019). 微分方程式論(Introduction of Differential Equations) (<https://www.mathworks.com/matlabcentral/fileexchange/71912-introduction-of-differential-equations>), MATLAB Central File Exchange. Retrieved December 9, 2019.

MATLAB Release Compatibility

Created with R2019a
Compatible with any release

Platform Compatibility

☒ Windows ☒ macOS ☒ Linux

Tags

[Add Tags](#)

[differential equa...](#) [japanese](#)
[lecture note](#) [live script](#)

Others Also Downloaded

Apollo 11 Moon Landing - 50th Anniversary Model
40 Downloads



Ben T. Nohara
Tokyo City Univ.

9.1 van der Pol方程式

van der Pol 方程式は、 $x = x(t)$ として

$$(9.1) \quad \ddot{x} - \epsilon(1 - x^2)\dot{x} + x = 0$$

という非線形の常微分方程式で表せられる。ここに、 ϵ は正の定数である。式(9.1)のような非線形微分方程式は一般に解を得ない。

さて、これをつぎのように書き改めるとリミットサイクルの現象を理解するのに都合がよい。

$$(9.2) \quad \begin{cases} \dot{x} = y - f(x) \\ \dot{y} = -x \end{cases}$$

ここに、

$$f(x) = -\epsilon \left(x - \frac{1}{3}x^3 \right)$$

である。

9.2 シミュレーション

まず、シミュレーションにより式(9.2)の動態を観察してみよう。

```
% CODE_9100
% van der Pol 方程式
clc;clear;
epsilon = 10;%epsilonの設定
tRange = [0 20];
x0 = [-4;1];
```


データアクセス

様々なデータを一緒に扱う

読み込んだデータ

時間 × (時間列+センサ)

名前 ▲	値	クラス
 flightData1Hz	4636x89 table	table

	A	B	C	D
1	Time	ABRK	ACMT	AIL_1
2		DEG		DEG
3	2001/5/10 17:03	119.9836	59	91.87889
4	2001/5/10 17:03	119.9836	59	91.89935
5	2001/5/10 17:03	119.9836	60	91.89935
6	2001/5/10 17:03	119.9836	59	91.89935
7	2001/5/10 17:03	119.9836	60	91.87889
8	2001/5/10 17:03	119.9836	60	91.96072
9	2001/5/10 17:04	119.9836	59	91.91982

テーブルデータへアクセス

T(rows,vars)
T{rows,vars}
T.var
T.(varindex)
T.var(rows)
T.Variables
S = vartype(type);
T(rows,S)
S = vartype(type);
T{rows,S}

データ. 変数名

flightData1Hz.A_T






 A_T
 ABRK
 ACMT
 AIL_1
 AIL_2

table → timetable 型へ
 時系列データを扱うための専用の型

```
myTimeTable = table2timetable(myTable)
```

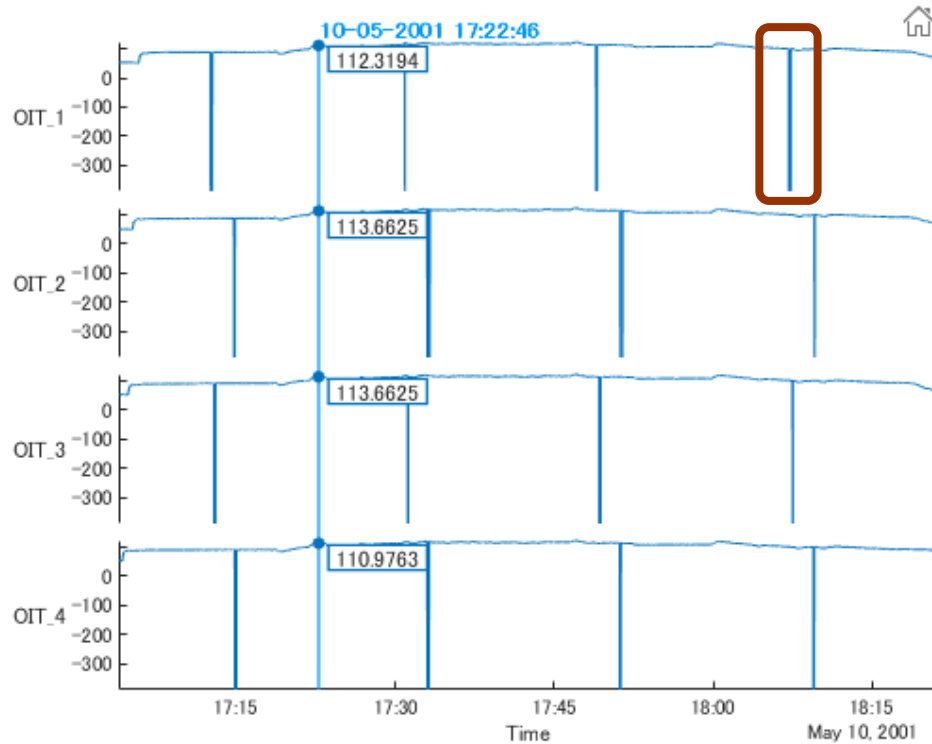
Time	Humidity_indoors	AirQuality	Humidity_outdoors	TemperatureF	PressureHg
2015-11-15 00:00:24	36	80	49	51.3	29.61
2015-11-15 01:13:35	36	80	NaN	NaN	NaN
2015-11-15 01:30:24	NaN	NaN	48.9	51.5	29.61
2015-11-15 02:26:47	37	79	NaN	NaN	NaN
2015-11-15 03:00:24	NaN	NaN	48.9	51.5	29.61

`t3=synchronize(t1,t2)`  `t3,t1,t2,:timetable`

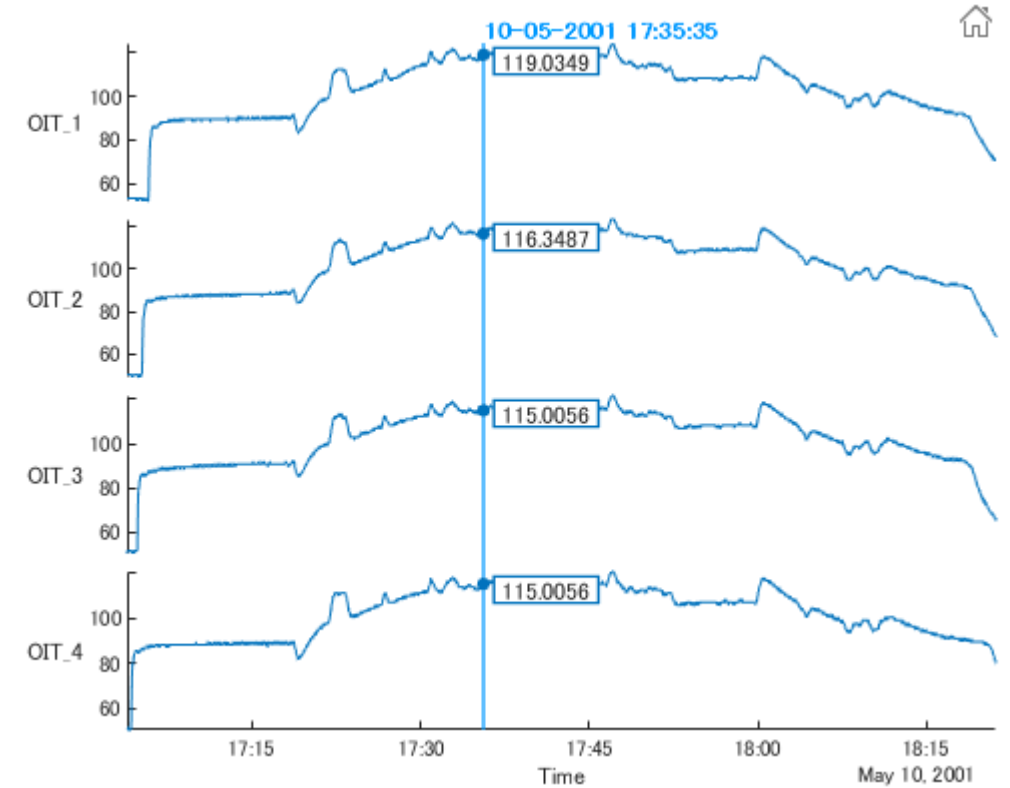
t3

Time	Humidity_indoors	AirQuality	Humidity_outdoors	TemperatureF	PressureHg
2015-11-15 00:00:24	36	80	49	51.3	29.61
2015-11-15 01:13:35	36	80	48.919	51.463	29.61
2015-11-15 01:30:24	36.23	79.77	48.9	51.5	29.61
2015-11-15 02:26:47	37	79	48.9	51.5	29.61
2015-11-15 03:00:24	37	80.378	48.9	51.5	29.61

可視化と前処理



filloutliers



```
stackedplot(t1hz,{'OIT_1','OIT_2','OIT_3','OIT_4'});
```

↑
timetable

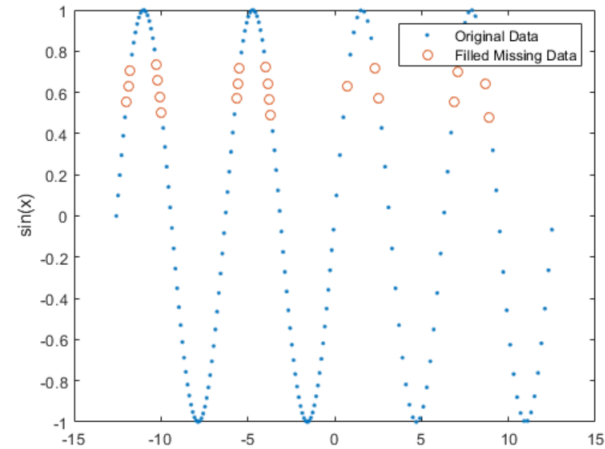
↑
テーブル内のセンサー名

便利な前処理関数

コマンド一行で前処理は完結させる

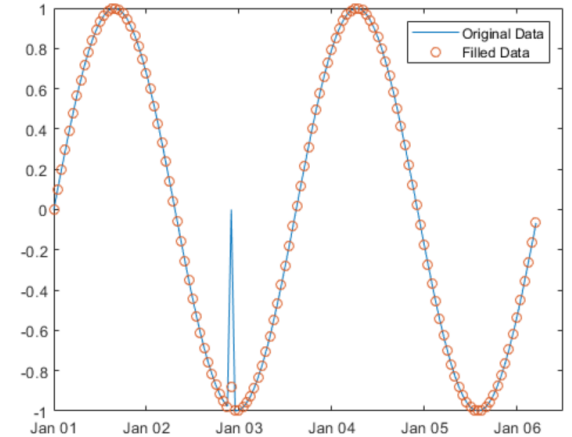
欠損値

`ismissing`
`rmissing`
`fillmissing`



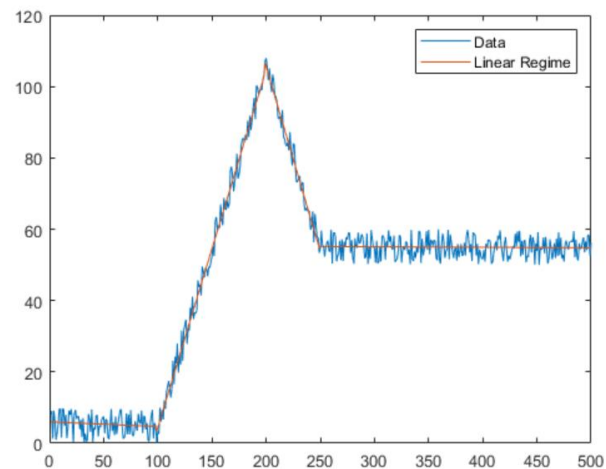
外れ値

`isoutlier`
`rmoutliers`
`filloutliers`



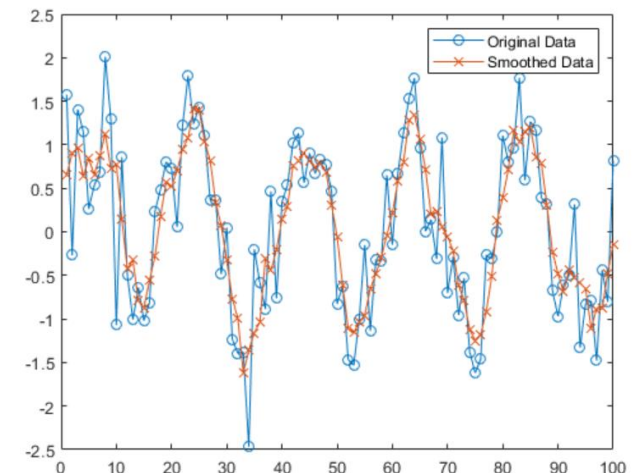
急激な変化点

`ischange`



平滑化

`smoothdata`



ライブタスクから最適な前処理設定を素早く探索

データ前処理

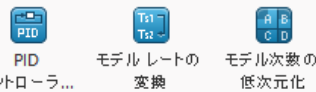


Table
Timetable



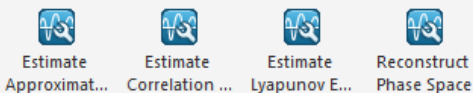
制御システム
設計と解析

制御システム設計と解析



予知保全

PREDICTIVE MAINTENANCE



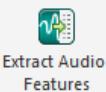
システム同定

SYSTEM IDENTIFICATION



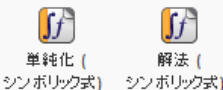
信号処理と通信

信号処理と通信



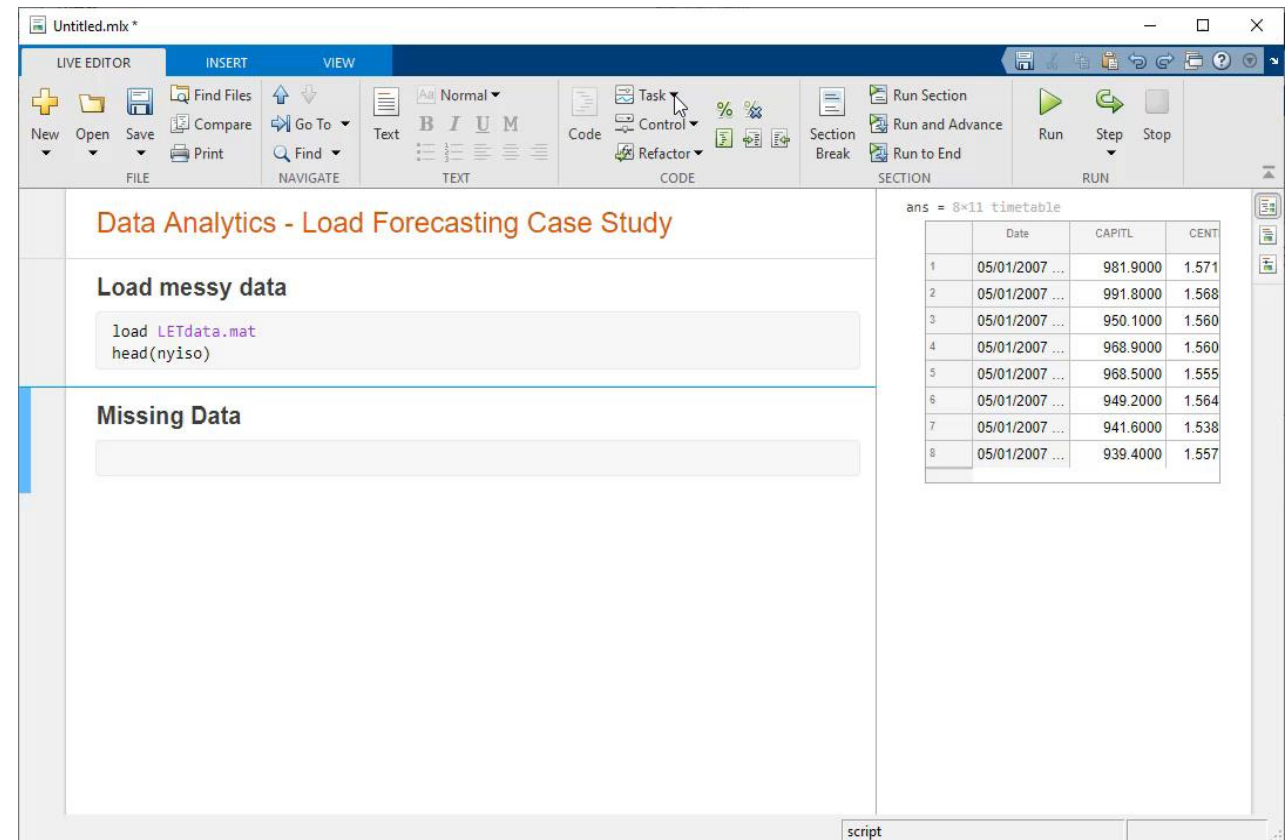
シンボリック
数学

SYMBOLIC MATH



ライブタスク

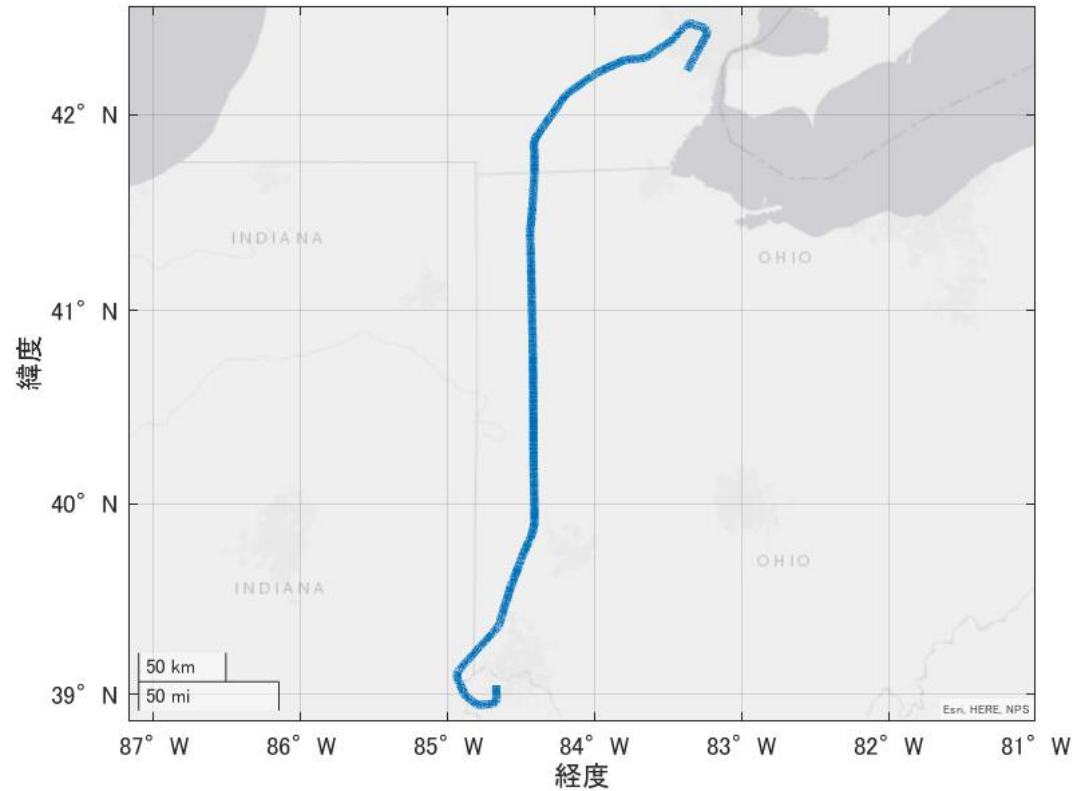
- 半GUI操作で試行錯誤が簡単に
- 対応するMATLABコードを自動生成



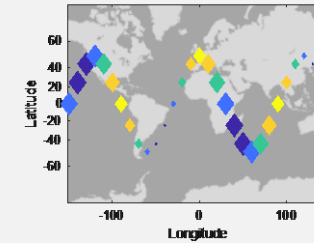
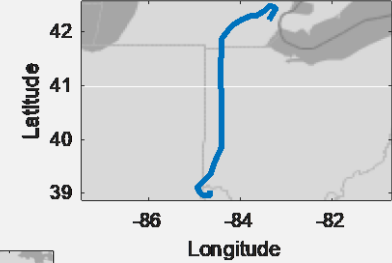
可視化 – 地図を利用

geoplot: 地理座標に軌跡をプロット

geoplot(lat,lon)

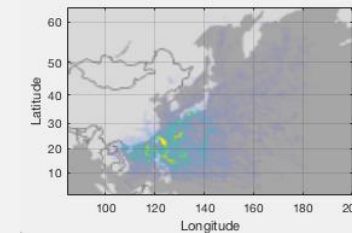
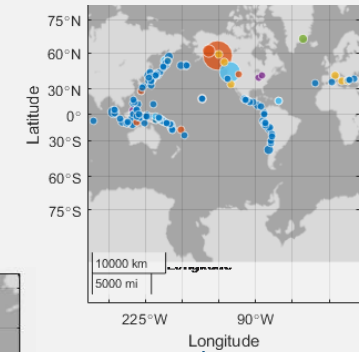


geoplot



geoscatter

geobubble



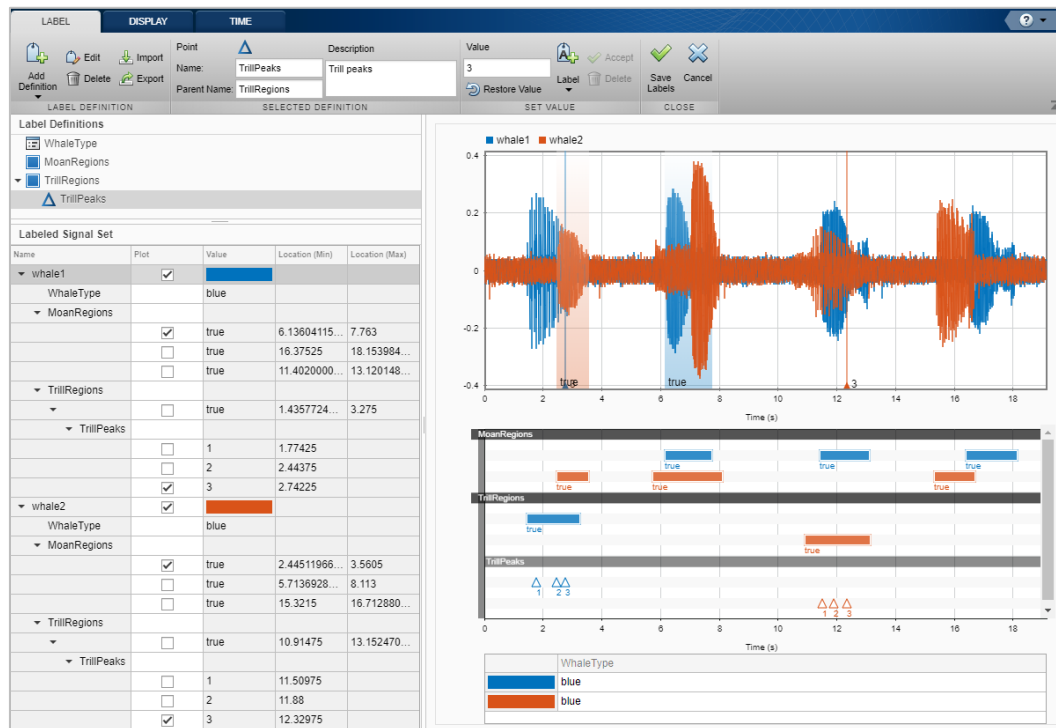
geodensityplot

<https://www.mathworks.com/help/matlab/geographic-plots.html>

機械学習等で特に必要な前処理の高速化支援ツール (本編とは無関係)

ラベリング作業はアプリで高速化

- マウス操作で高速ラベリング
- トラッキングアルゴリズムを使った自動ラベリング機能



Signal Labeler app
(例) イルカの鳴き声の分類

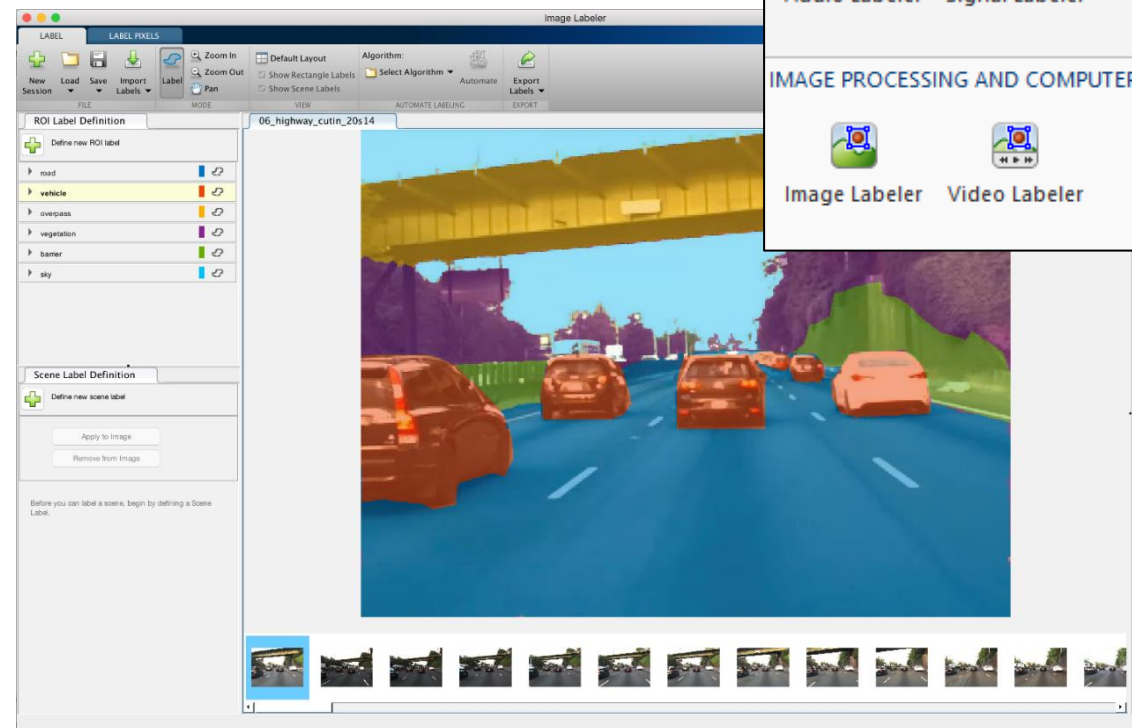


Image Labeler app
(例) セマンティックセグメンテーション

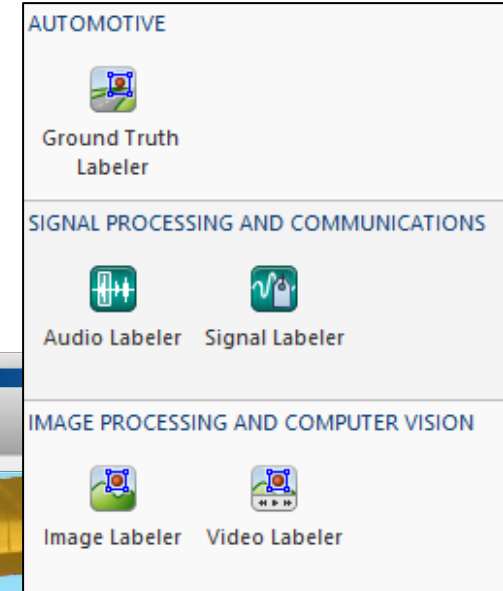
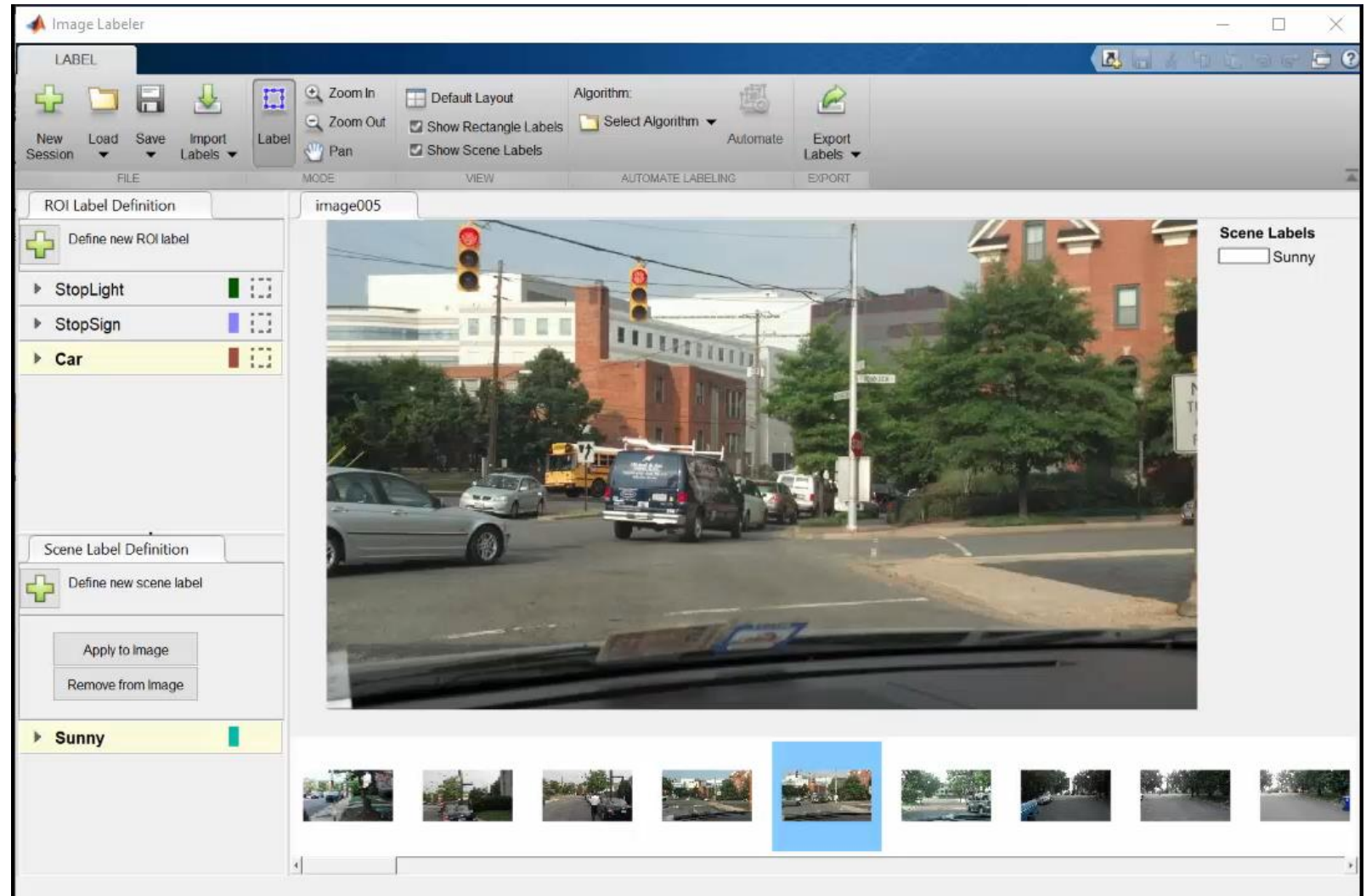


Image Labeler App

- Image Labeler App
 - マウス操作でROI指
 - イメージフレーム毎にラベリング
- トラッキングアルゴリズム
 - イメージフレーム全体に自動ラベリング



テキストデータの前処理もコマンド一行で!

アクセス

Webから



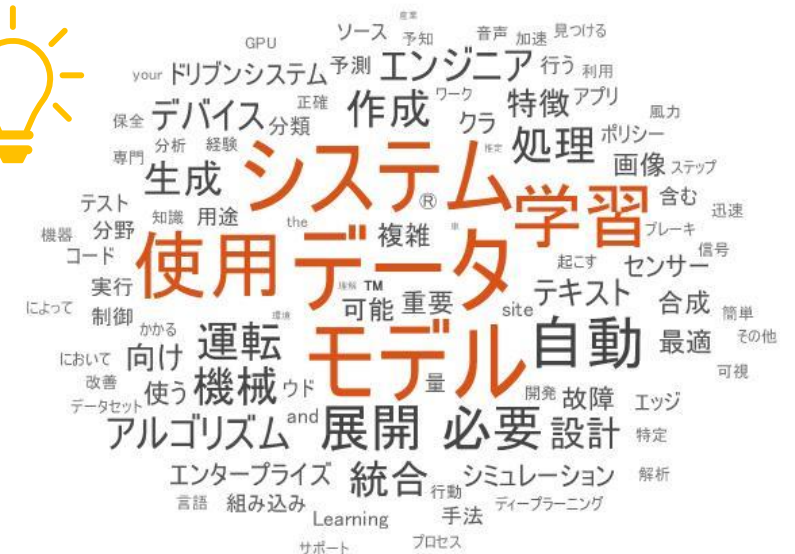
HTMLをパース
HTMLの要素検出
HTMLの構文解析
トークン化
語幹, レンマ化
特定単語削除



htmlTree
findElement
extractHTMLText
tokenizedDocument
normalizeWords
removeWords

探索 & 発見

データ解析 & モデリング



テキストデータ用の前処理関数の一例

- 不要なコンテンツの削除を可能にするフィルター処理関数

➤ トークン化

`tokenizedDocument`

吾輩は猫である。



吾輩 は 猫 で ある 。

➤ ストップワード削除

`removeStopWords`

吾輩 は 猫 で ある 。



吾輩 猫 。

➤ 句読点の削除

`erasePunctuation`

吾輩 猫 。



吾輩 猫

➤ レンマ化

`normalizeWords`

輝き を 増し て いる



輝く を 増す て いる

➤ 小文字化

`lower`

Hello World








hello world

その他、

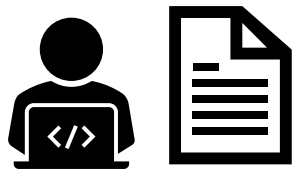
- 数値変換
- 特定の語の削除
- 長い/短い語の削除
- URL/HTMLタグの削除
- スペル修正（英独韓）etc…

ポイントのまとめ

-  GUIでデータインポート操作、等価な関数も1クリックで作成
-  テーブルでデータを集約、`table`.**変数名**でデータアクセス
-  時系列解析に便利な**timetable**
-  データの前処理は**関数1行**, ライブタスク or アプリで実施
-  系列データは**stackedplot**で簡単可視化

①アドホックな解析

- データ読み込み
- データアクセス
- データ結合
- 前処理
- 可視化



アドホック解析における
データの前処理をマスター

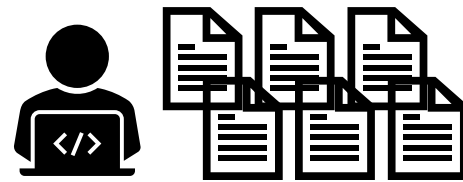
①アドホックな解析

- データ読み込み
- データアクセス
- 前処理
- 可視化



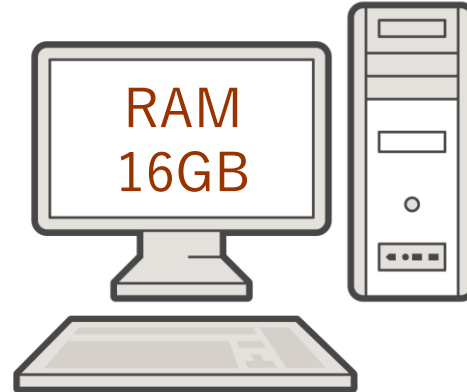
②ビッグデータ解析

- ビッグデータの読み込み
- ビッグデータの解析・深掘り

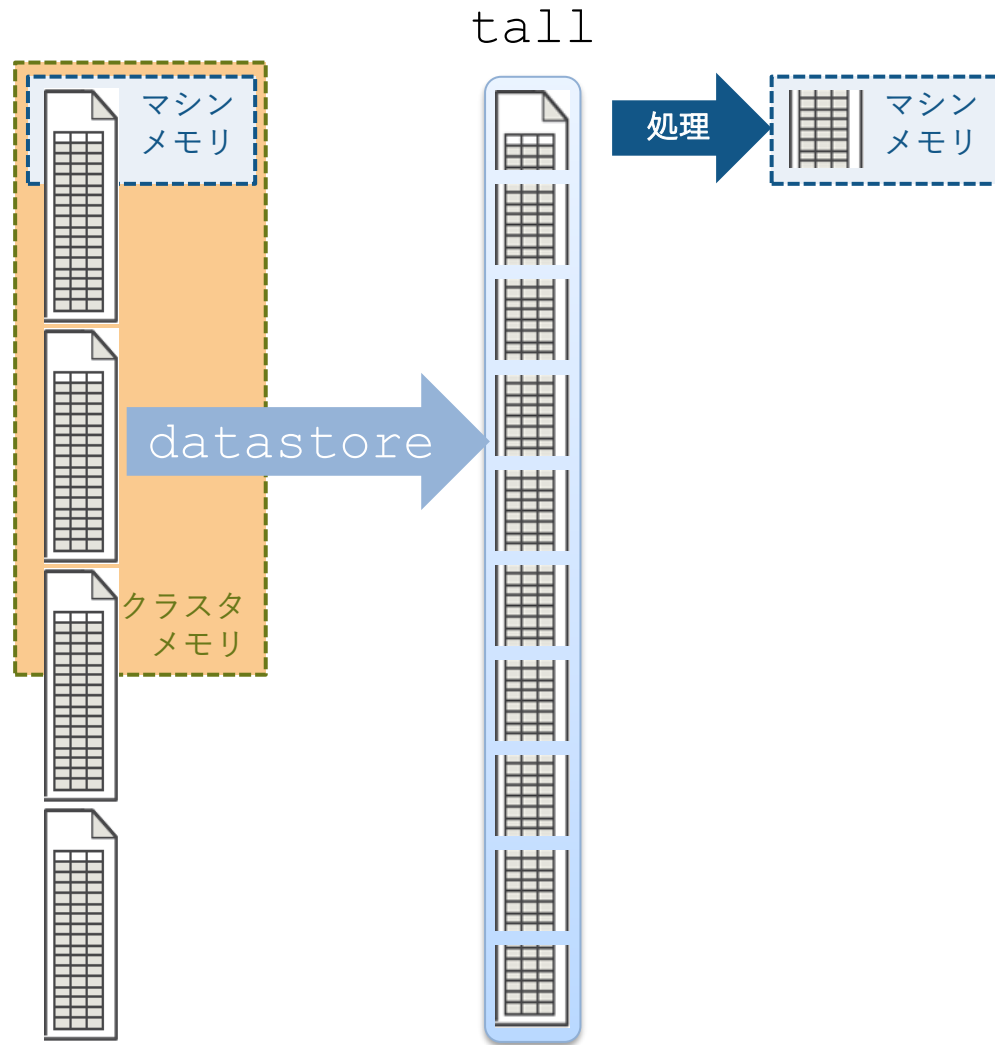


フライトデータ概要

- 35 機
- 180,000 飛行
- サイズ **300 GB**
- 出典:
 - NASA Dash Link: Sample Flight Data
 - <https://c3.nasa.gov/dashlink/projects/85/>



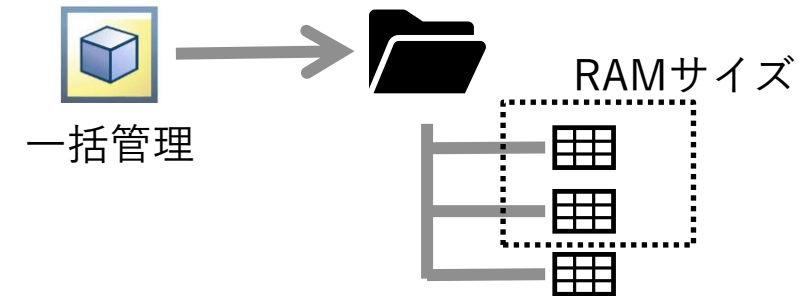
メモリに収まらないデータ取り扱いのためのテクニック



datastore型変数

メモリに収まらないデータの集まりのリポジトリ

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



tall配列

datastoreに格納されたデータを操作する

```
t = tall(ds)
```


datastoreで大きなデータ集合を扱う

必要な項目（センサー等）の選択

```
varnames = {'Time', 'EngineSpeedRPM'}
```

選択項目についてのdatastore型変数の作成

```
ds = datastore('EngineData*.csv', ...  
              'SelectedVariableNames', varnames)
```

“メモリに乘る”サイズのデータの読み込み

```
data = read(ds)
```



利用可能な datastore 一覧

一般	datastore
	spreadsheetDatastore
	tabularTextDatastore
	fileDatastore
データベース	databaseDatastore
画像	imageDatastore
	denoisingImageDatastore
	randomPatchExtractionDatastore
	pixelLabelDatastore
	augmentedImageDatastore
音声	audioDatastore
予知保全	fileEnsembleDatastore
	simulationEnsembleDatastore
Simulink	SimulationDatastore
自動車	mdfDatastore
カスタム	subclass matlab.io.Datastore
変換	transform 存在するdatastoreの変換

“全体”の計算はtallで実行

1個のファイル

データにアクセス

```
data = readtable("EngineData1.csv");
data = table2timetable(data);
```

データの前処理

興味のあるデータの選択

```
data = data(:, "EngineSpeedRPM");
```

欠損データ処理

```
data = fillmissing(data, "linear");
```

統計量計算

```
m = mean(data.EngineSpeedRPM);
s = std(data.EngineSpeedRPM);
```

1,000個のファイル

データにアクセス

```
data = datastore("EngineData*.csv");
data = tall(data);
data = table2timetable(data);
```

データの前処理

興味のあるデータの選択

```
data = data(:, "EngineSpeedRPM");
```

欠損データ処理

```
data = fillmissing(data, "linear");
```

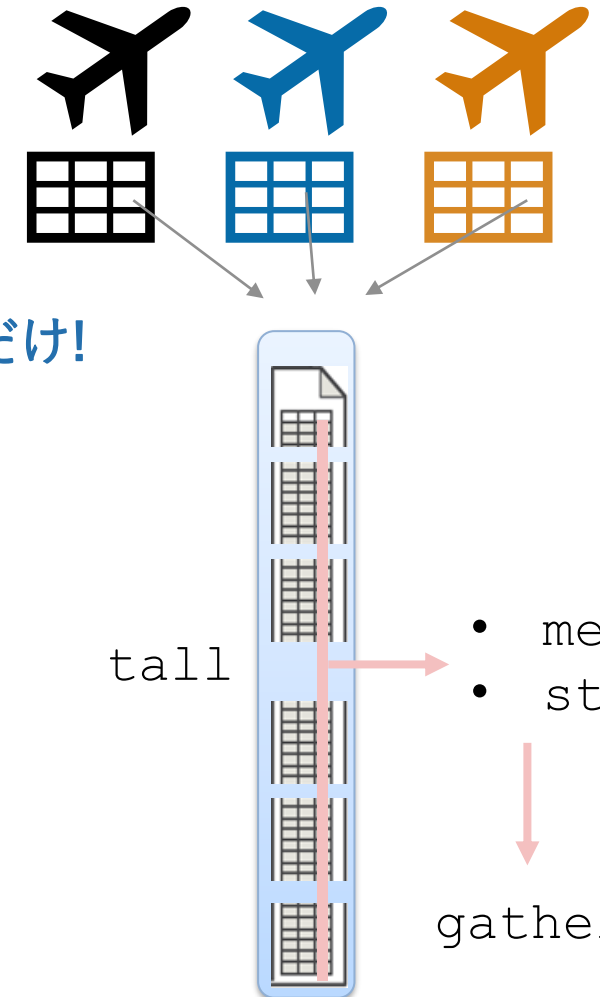
統計量計算

```
m = mean(data.EngineSpeedRPM);
s = std(data.EngineSpeedRPM);
```

- 平均
- 分散

```
[m,s] = gather(m,s);
```

2行付け加えるだけ!



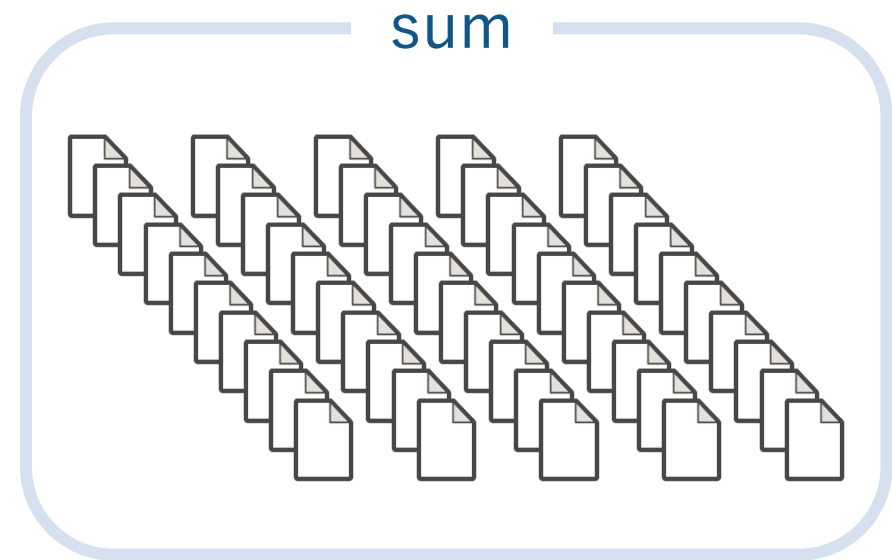
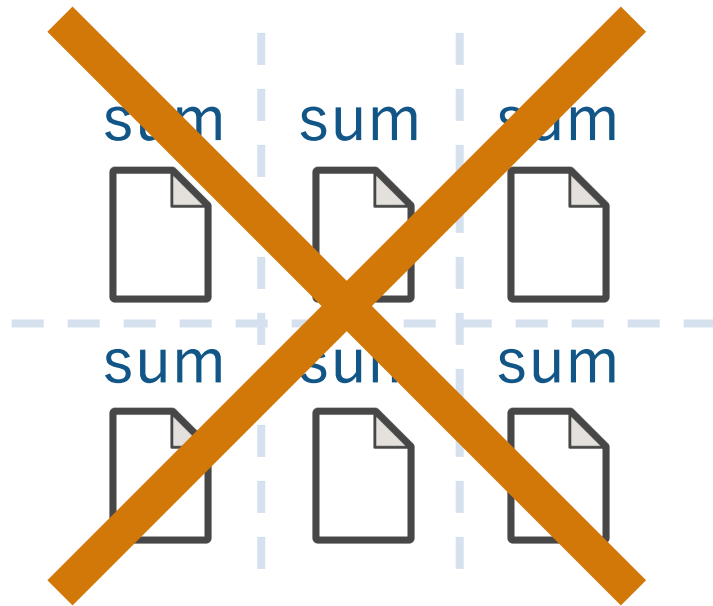
ポイントのまとめ



datastoreによりRAMサイズ以上の大容量データを一括管理



tailによってビッグデータも従来通りの操作が可能



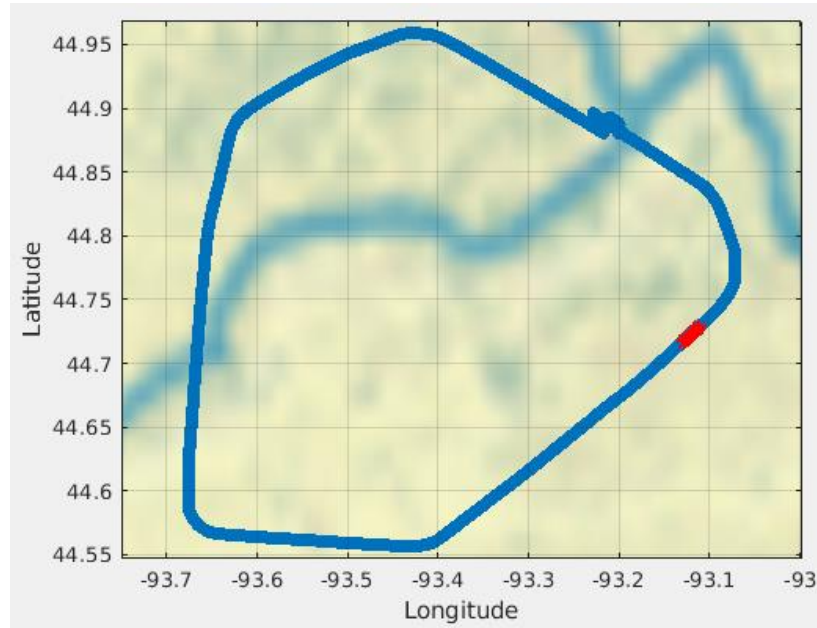
ビッグデータの活用例 – 事象の深堀り

4つの火災センサーを探索



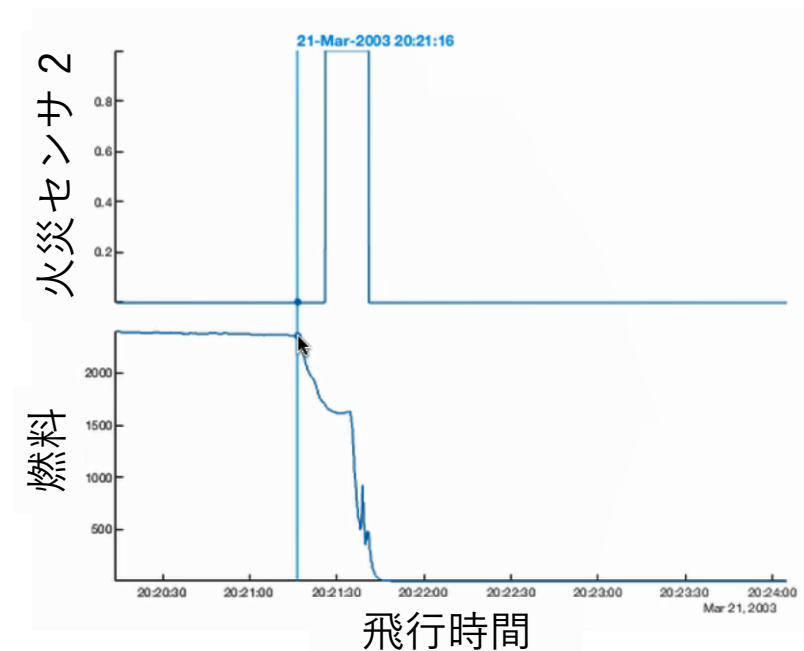
stackedplot

フライト軌跡の確認



geoplot

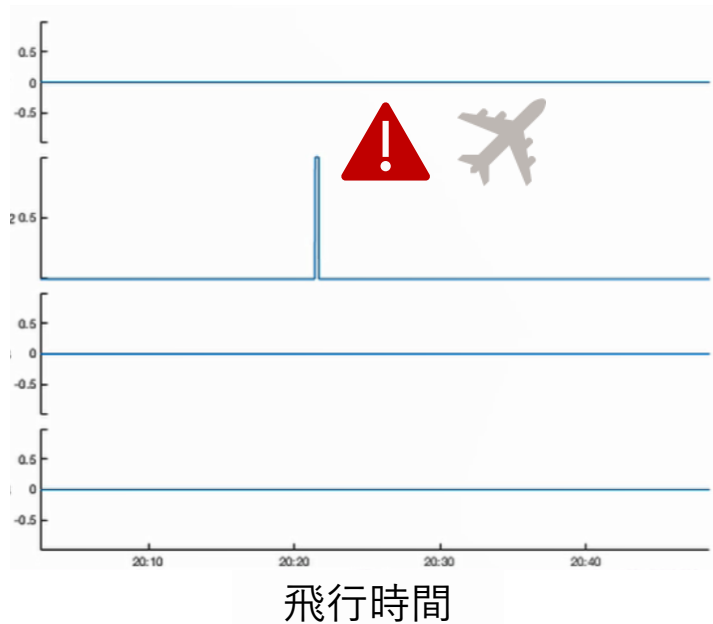
燃料データの確認



stackedplot

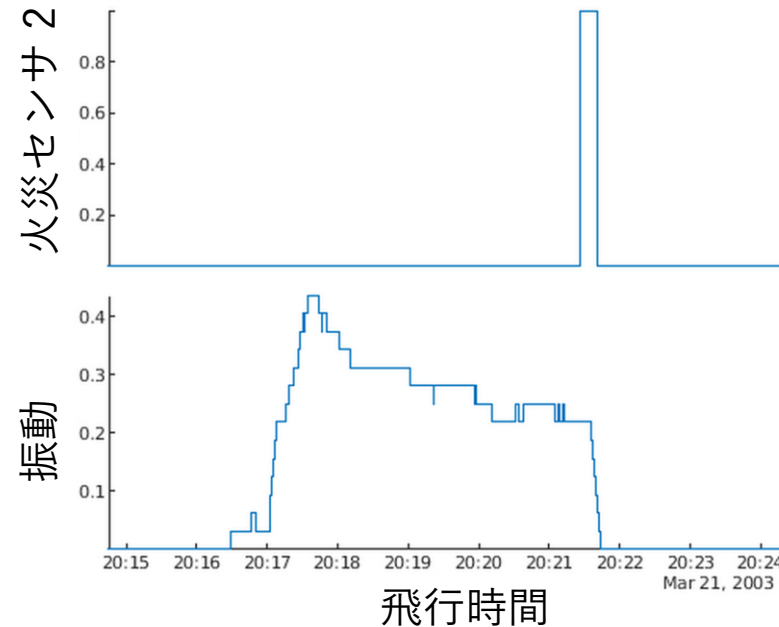
ビッグデータの活用例 – 事象の深堀り

4つの火災センサーを探索



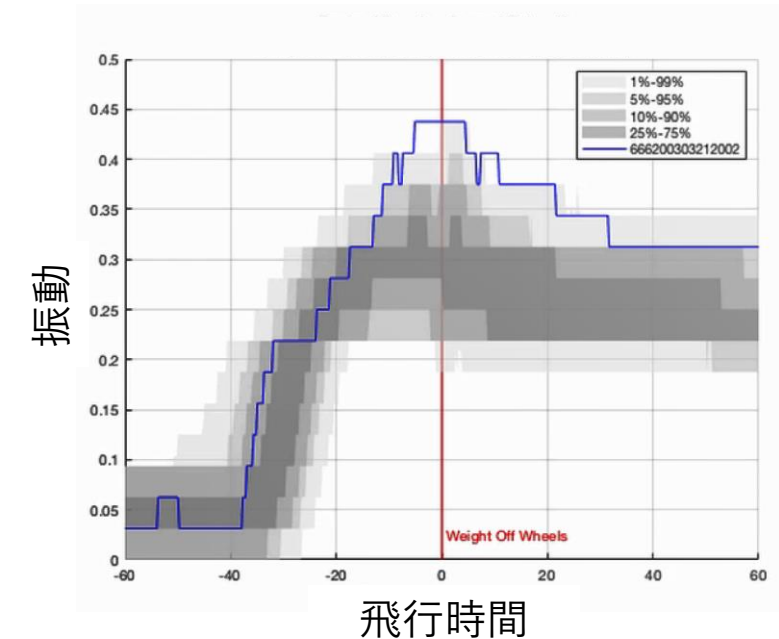
stackedplot

振動データの確認



stackedplot

全フライトとの比較



火災が発生したフライトは
振動レベルが高かった

ビッグデータも通常通り解析を継続可能

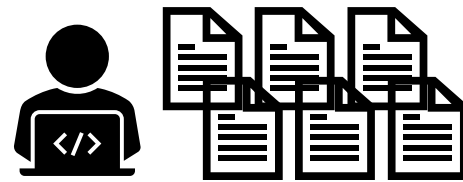
①アドホックな解析

- データ読み込み
- データアクセス
- データ結合
- 前処理
- 可視化



②ビッグデータ解析

- ビッグデータの読み込み
- ビッグデータの解析・深掘り



ビッグデータの
効率の良い
解析方法をマスター

①アドホックな解析

- データ読み込み
- データアクセス
- 前処理
- 可視化



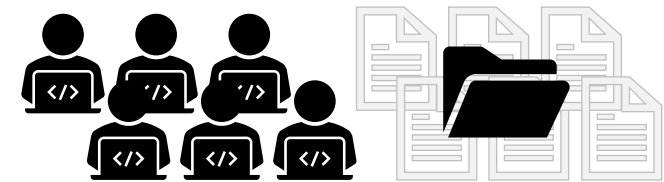
②ビッグデータ解析

- ビッグデータの読み込み
- ビッグデータの解析



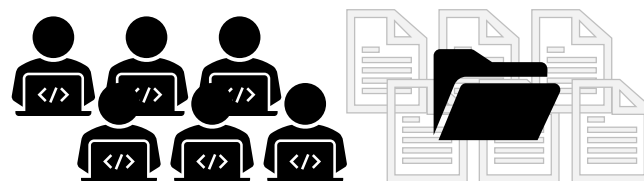
③プロジェクト化

- プロジェクトの作成・共有
- 大人数で解析

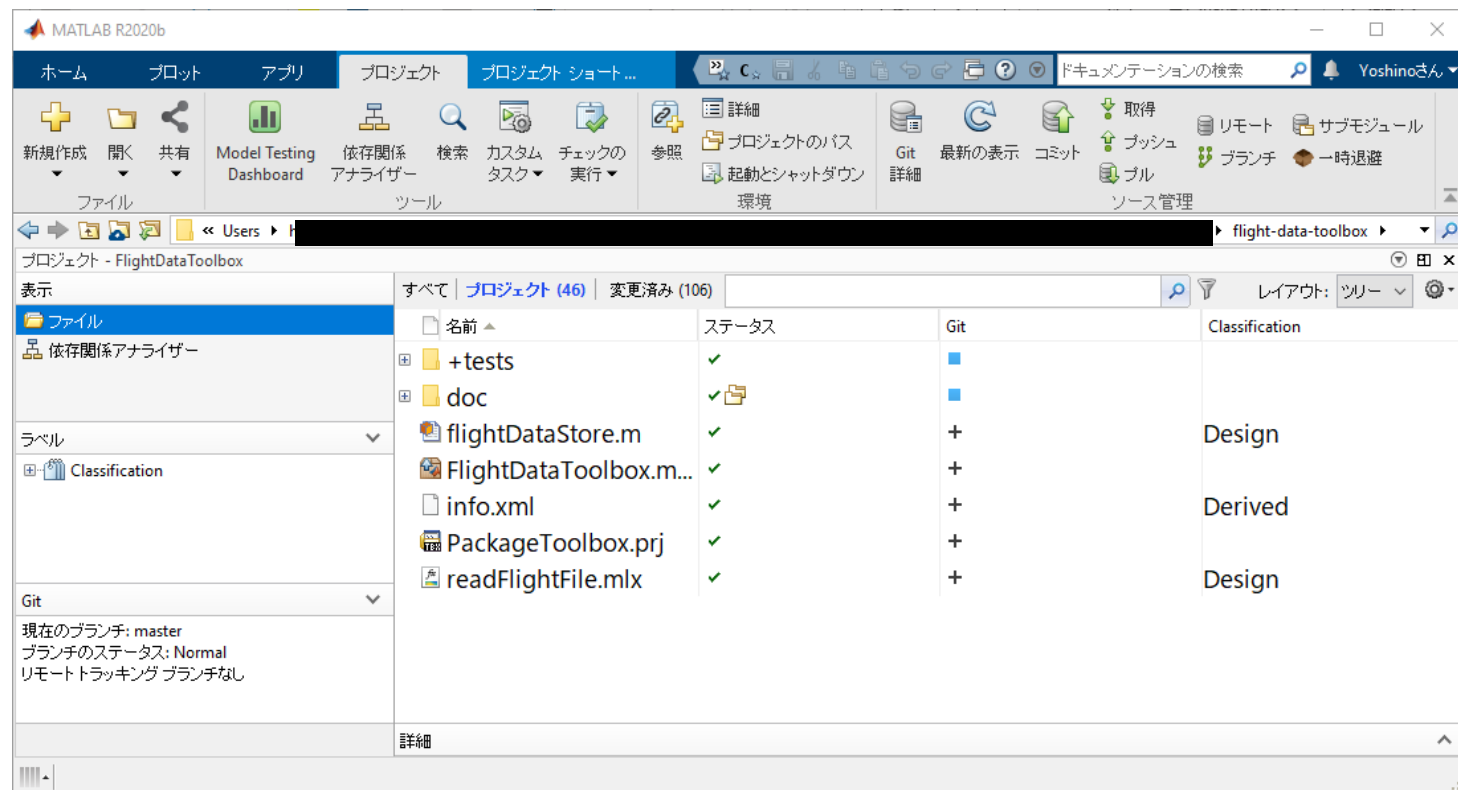


プロジェクト化

共同作業・バージョン管理



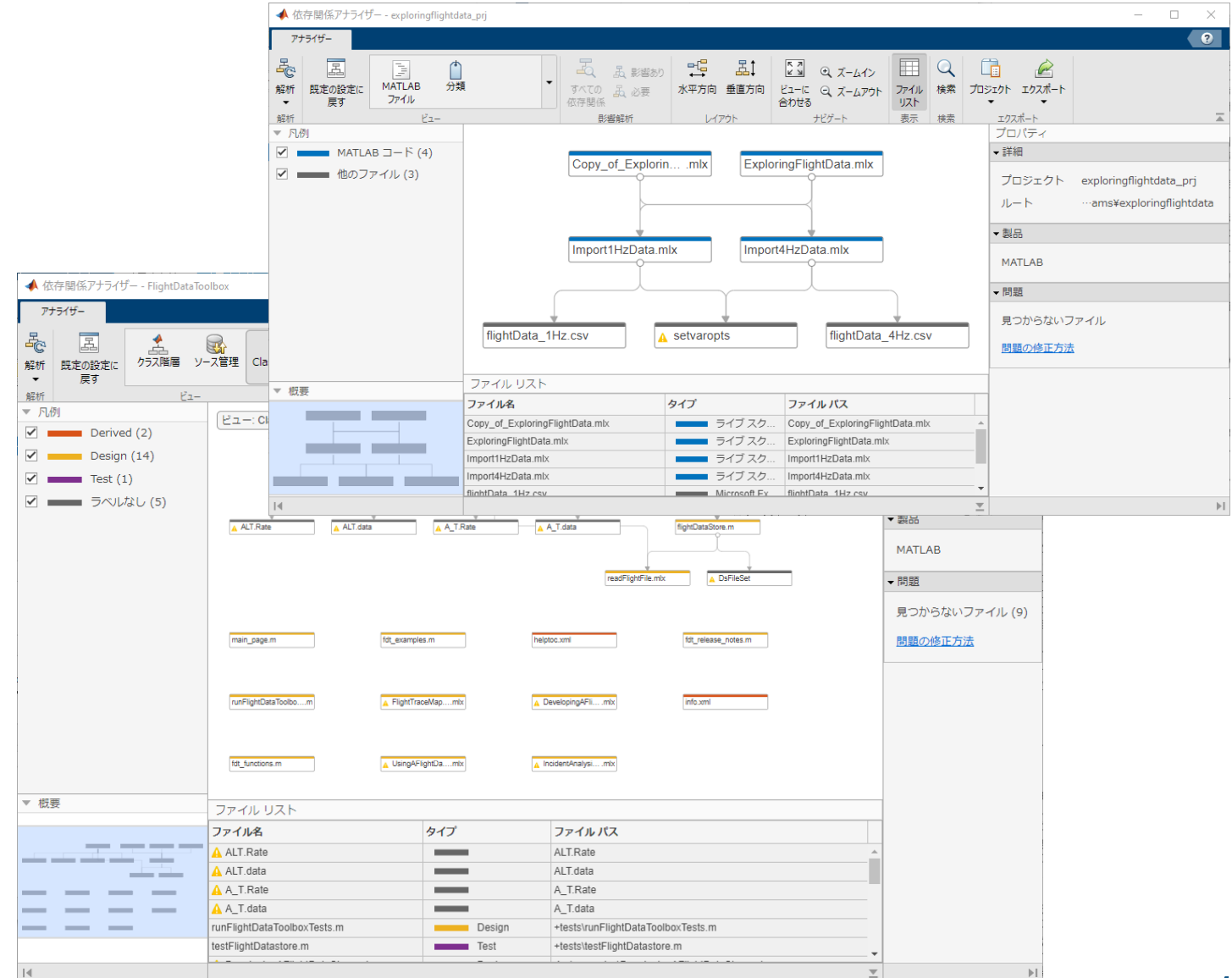
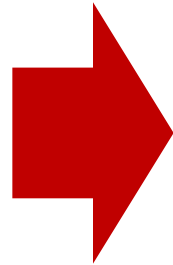
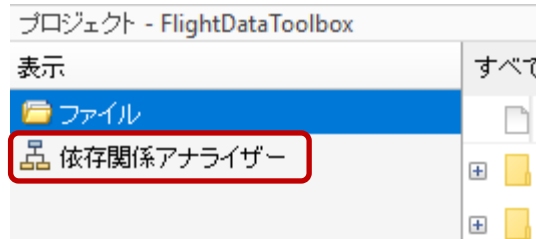
- ファイル、データ、テスト等を1か所で管理
- 作業の整理、チームでの共同作業を円滑にする
- Git™, SVN®を使用したソース管理ツール (GitはMATLABに同梱)
- 依存関係の解析



MATLABプロジェクトについて

プロジェクトによる業務一元化

コードの依存関係解析して可視化



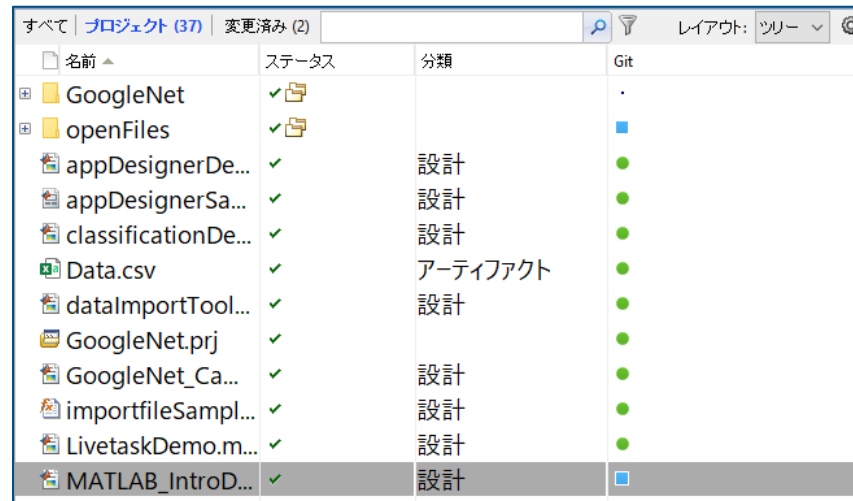
- ファイル、データ間の依存関係を可視化
- クラスの階層なども表示可能 (オブジェクト指向プログラミング)

プロジェクトによる業務一元化

ソースコード管理

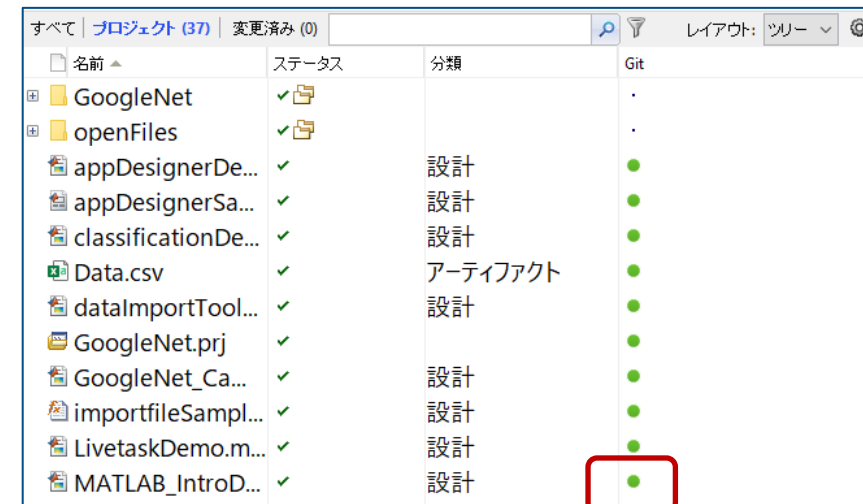
Gitを使ったバージョン管理の例

- Git 対応 (GitHubなどにも使用できる!)
- SVN 対応
- ボタンクリックで操作



名前	ステータス	分類	Git
GoogleNet	✓		•
openFiles	✓		•
appDesignerDe...	✓	設計	•
appDesignerSa...	✓	設計	•
classificationDe...	✓	設計	•
Data.csv	✓	アーティファクト	•
dataImportTool...	✓	設計	•
GoogleNet.prj	✓		•
GoogleNet_Ca...	✓	設計	•
importfileSampl...	✓	設計	•
LivetaaskDemo.m...	✓	設計	•
MATLAB_IntroD...	✓	設計	•

commit



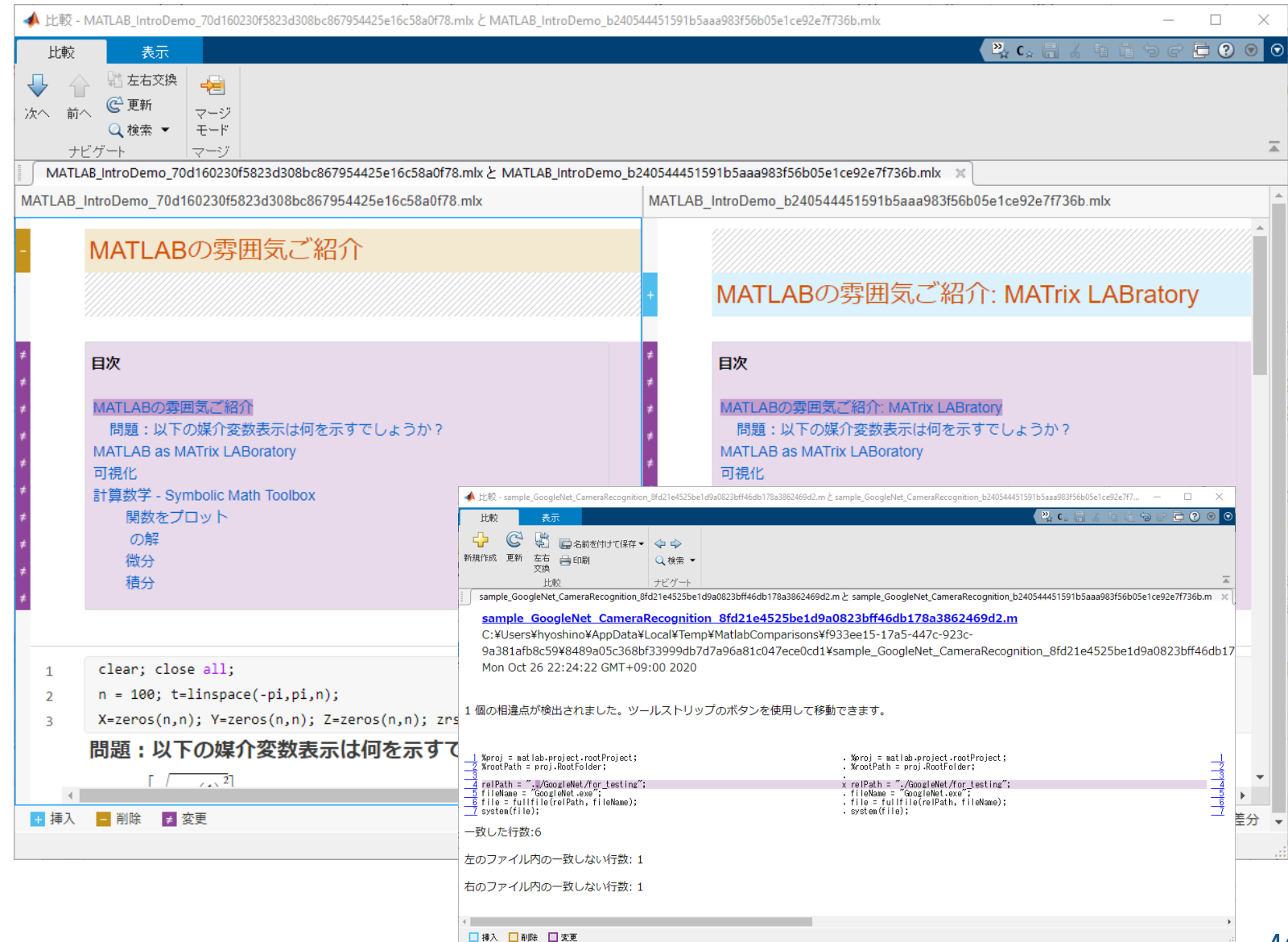
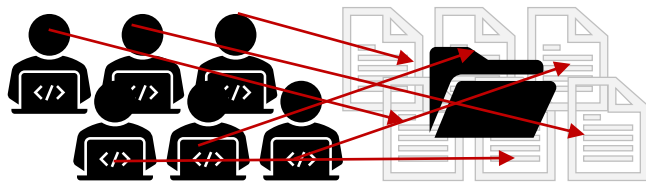
名前	ステータス	分類	Git
GoogleNet	✓		•
openFiles	✓		•
appDesignerDe...	✓	設計	•
appDesignerSa...	✓	設計	•
classificationDe...	✓	設計	•
Data.csv	✓	アーティファクト	•
dataImportTool...	✓	設計	•
GoogleNet.prj	✓		•
GoogleNet_Ca...	✓	設計	•
importfileSampl...	✓	設計	•
LivetaaskDemo.m...	✓	設計	•
MATLAB_IntroD...	✓	設計	•



プロジェクトによる業務一元化

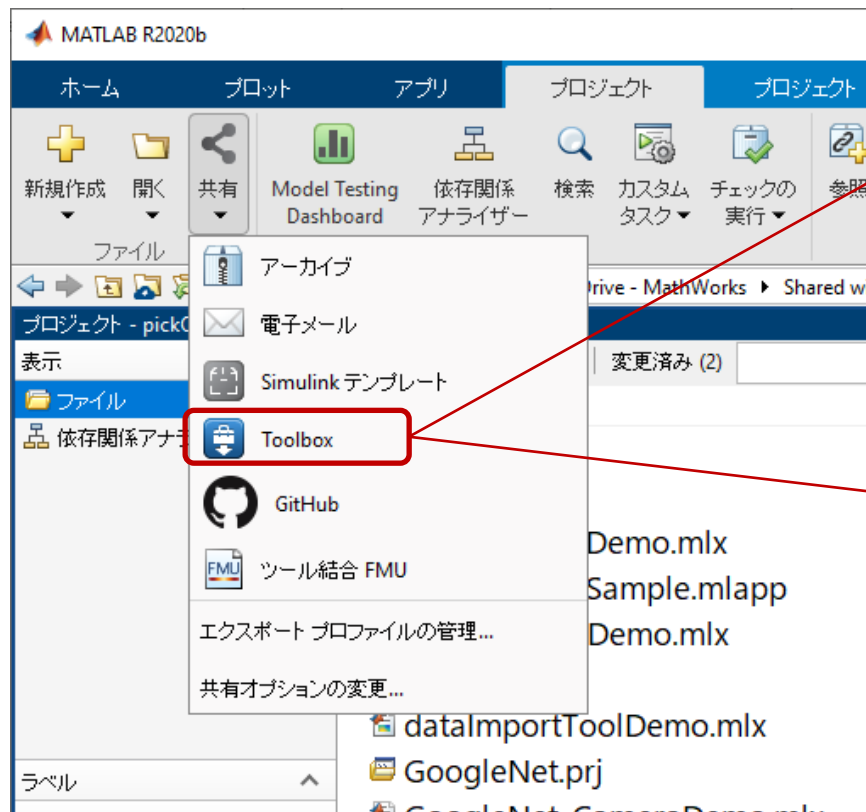
ソースコード管理 (Cont.)

- Revisionとの相違
 - 挿入
 - 削除
 - 変更
- 複数のRevision間を比較可能
- 複数人での作業の効率が向上

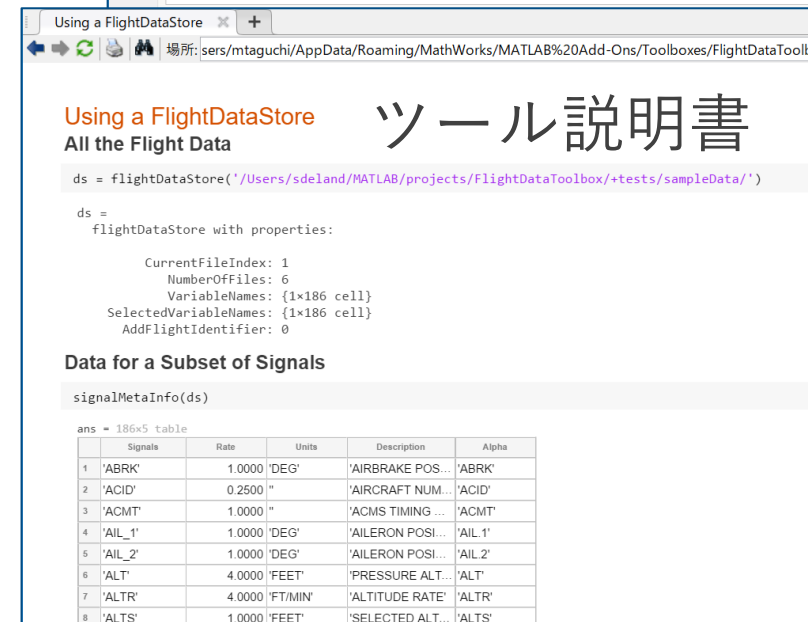
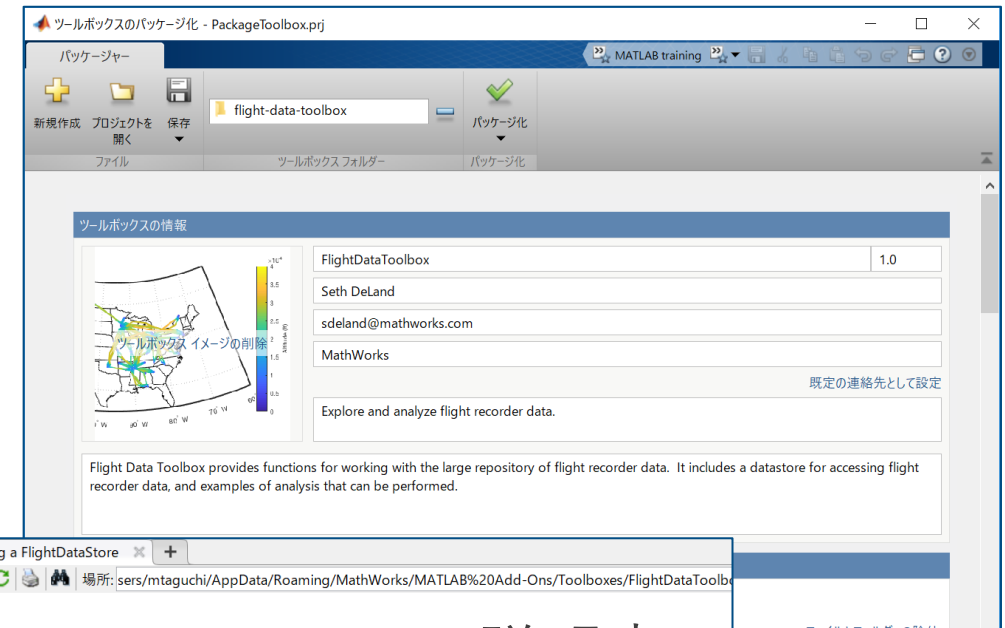


検証済みツールとして共有





プロジェクト終了 →
ツールボックス化して共有



ツールボックス



ポイントのまとめ

-  MATLABプロジェクトによる簡単配布・作業標準化
-  プロジェクト可視化機能による簡単解読
-  ソース管理機能連携により複数人での作業を円滑化
-  ツールボックス化して配布、さらなる協業を加速

①アドホックな解析

- データ読み込み
- データアクセス
- 前処理
- 可視化



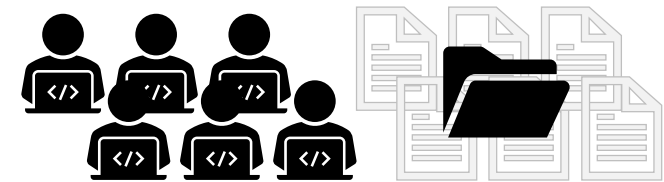
②ビッグデータ解析

- ビッグデータの読み込み
- ビッグデータの解析






③プロジェクト化

- プロジェクトの作成・共有
- 大人数で解析



まとめ

-  データ解析の鍵を握る前処理をMATLABで効率よく実施可能
-  大容量データ解析に対応、ビッグデータ活用の橋渡し
-  プロジェクト化により組織横断プロジェクトへ拡大

本日より紹介したツール – MATLAB R2021a



更に学びたい方へ



MATLAB 入門



機械学習入門



ディープラーニング入門

- 無料の入門コース
- 受講にライセンス不要
- Webブラウザで操作



Stateflow 入門



Simulink 入門

<https://jp.mathworks.com/services/training.html>

機械学習における前処理の効率化

ebookにアクセス

MATLAB®の高水準のツール・可視化・ドメイン固有ツールとアプリ・ライブエディタータスクを使用して、機械学習アルゴリズムに使用される表形式データおよび時系列データの前処理を高速化する方法を学びます。これらのツールを前処理スクリプトと一緒に使用することで、機械学習モデルの精度を迅速に評価することができます。

ebookには次の内容が含まれています。

- ✓ データの調査
- ✓ 一般的な前処理
- ✓ MATLABアプリによる前処理

ebookにアクセス

[リンクはこちら](#)





© 2021 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.