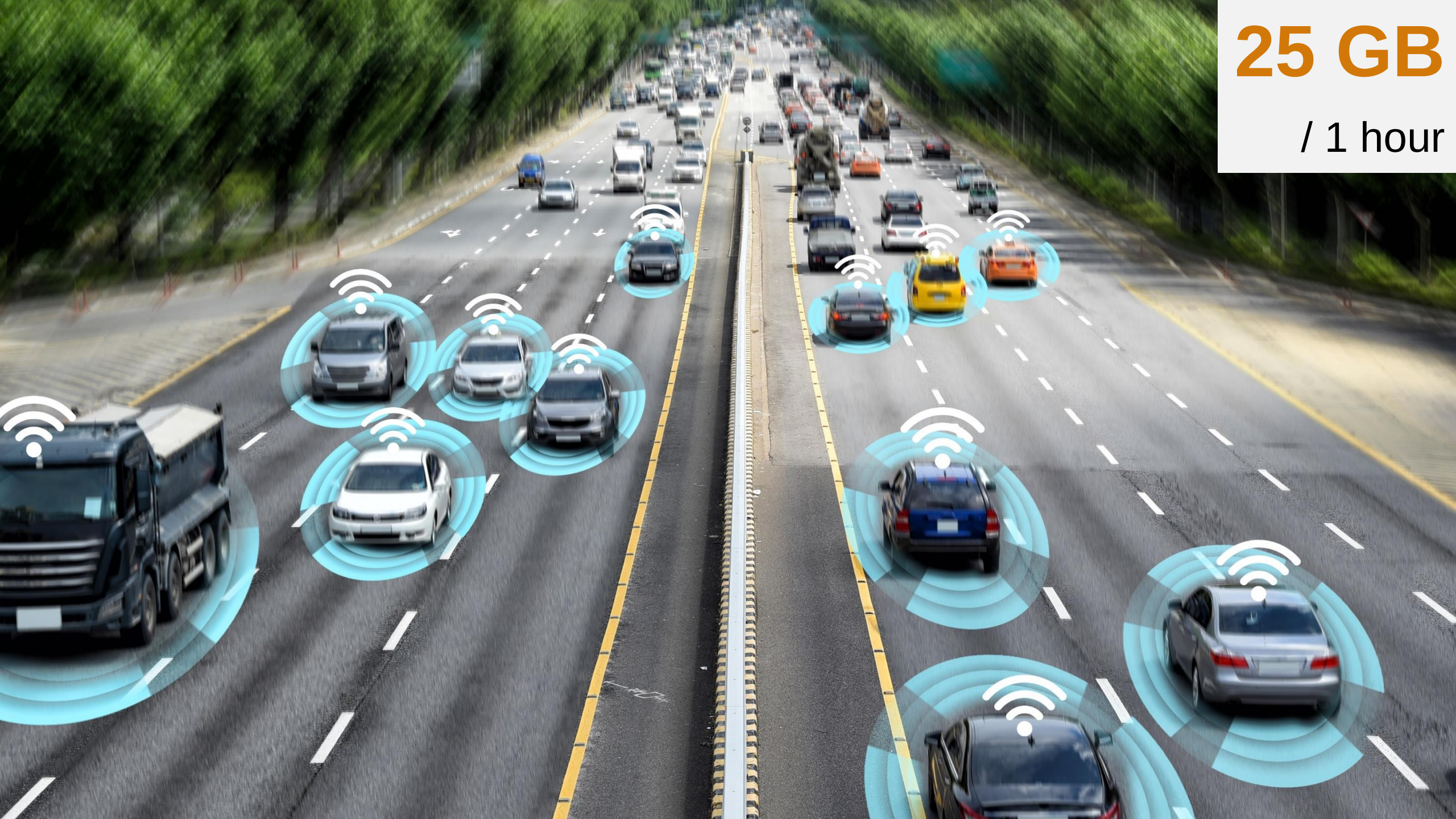
MATLAB による大規模フリートデータ解析

Part 1: デスクトップ編

MathWorks Japan
アプリケーションエンジニアリング部
齊藤　甲次朗

アジェンダ

- はじめに
  - ビッグデータ解析の課題
- フリートデータ解析実践
  - デスクトップでの解析


25 GB
/ 1 hour

フリートデータ解析を含むビッグデータ解析の課題

1. ビッグデータのための新しいツールを学ぶコストが掛かる

2. 大規模な計算に移行するために、プロトタイプで書いたコードの書き直しが必要になる

フリートデータ解析実践

フリートデータ解析実践
使用するデータ

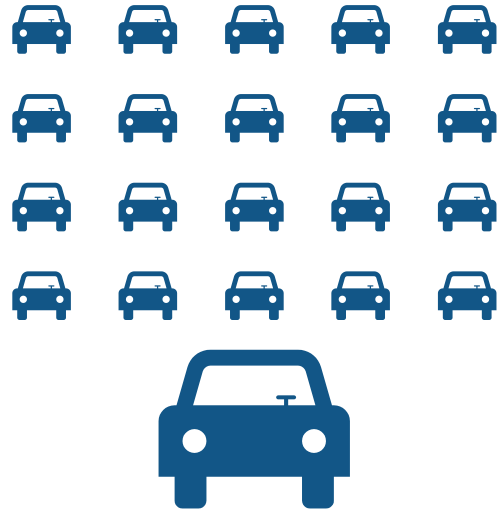
- MathWorksの社員が
  車にOBDドングルを付け走行データを記録

車両： **21**台

トリップ数：**1300**以上

チャネル数：**39**

データ収集期間：約**1.5**年



フリートデータ解析実践
使用するデータ

4G LTE

スマートフォン

OBD II

Bluetooth

MATLAB Production Server™

Request Broker and Program Manager

- データの整形
- ファイル生成

データベース

全てのデータをダウンロードして
MATファイルに変換

フリートデータ解析のワークフロー

データへのアクセス

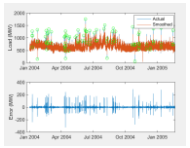
ファイル

データベース

センサー

データの前処理

異常・欠損データの扱い



データ削減/変換

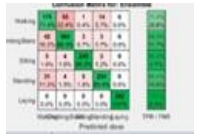
特徴抽出

予測モデルの開発

モデルの作成(機械学習)

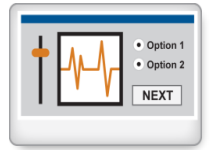
パラメータ最適化

モデルの検証



システムへの統合

デスクトップアプリケーション



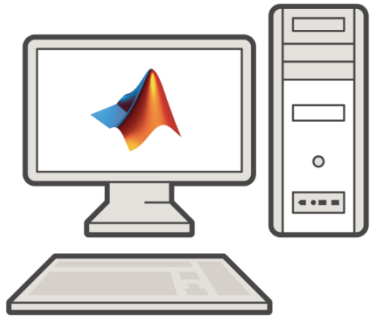
エンタープライズシステム

MATLAB Excel .NET .exe C/C++ Java .dll

組込デバイスとハードウェア

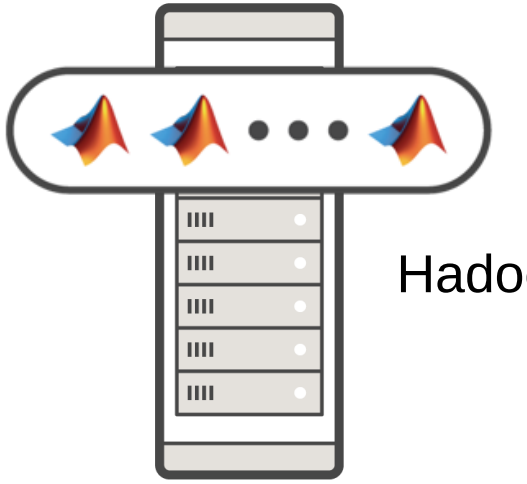
ビッグデータの扱い
フリートデータ解析　サマリー

ステップ1



デスクトップPCでの解析

ステップ2

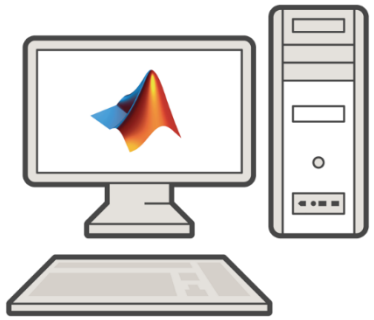


Hadoop® / Spark™

Hadoopクラスターでの解析

フリートデータ解析実践
デスクトップでの解析

ステップ1



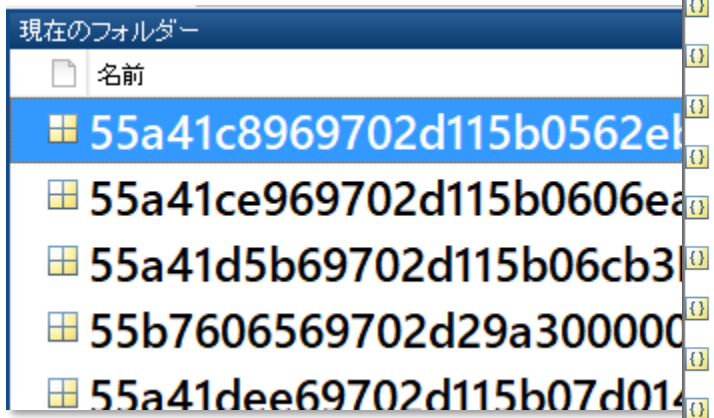
フリートデータ解析アルゴリズムを検討するために、
まずはデスクトップで**試行錯誤**

今後のクラスターへの**スケールアウトを意識**してコードを書く

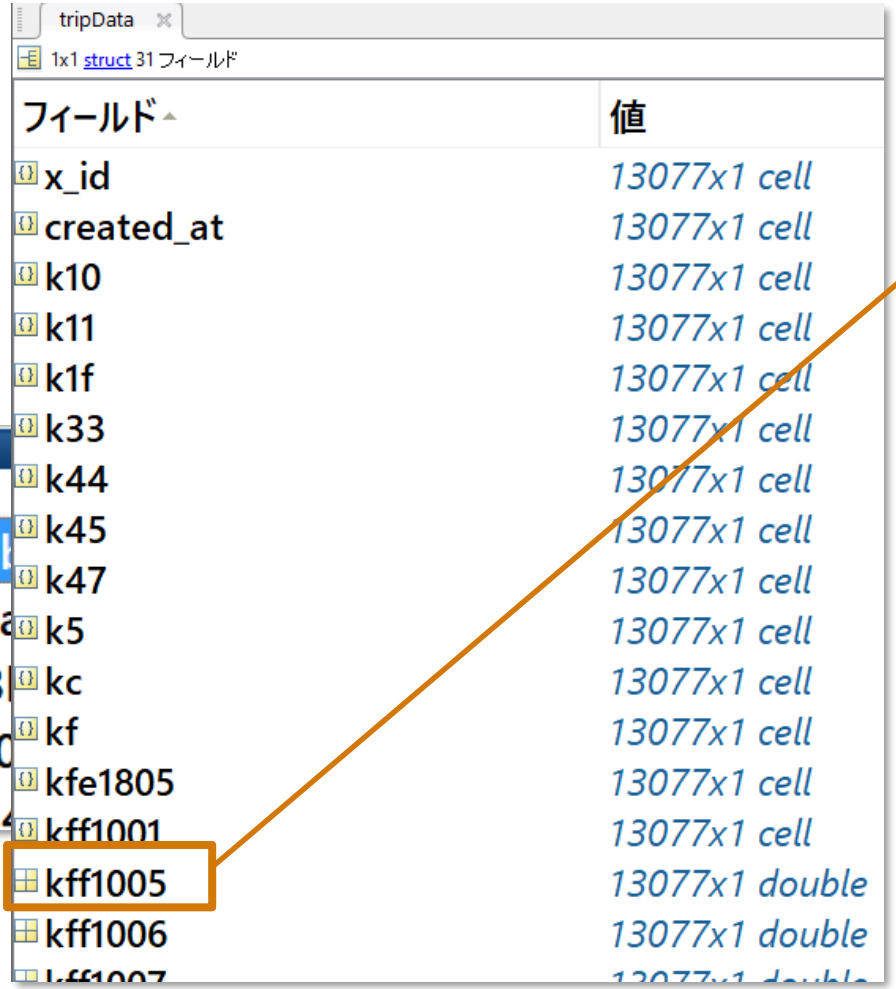
フリートデータ解析実践
データへのアクセス

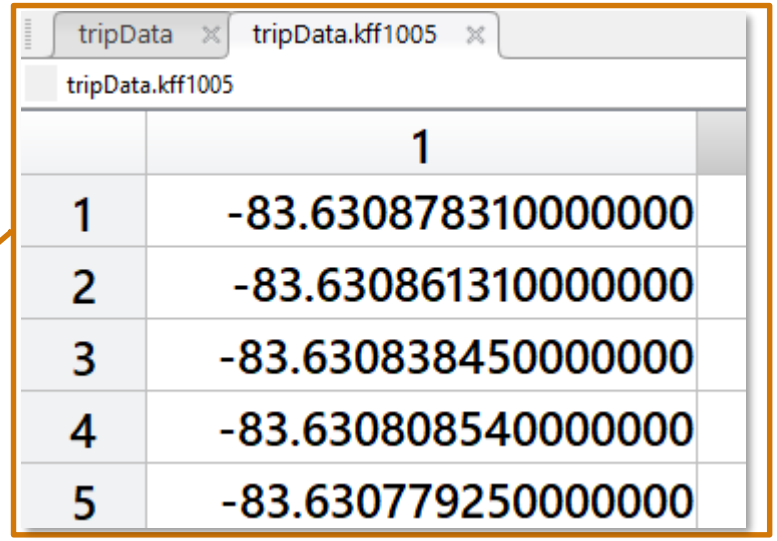
生データを見てみる

1ファイル



MATLAB上でファイルを
ダブルクリック

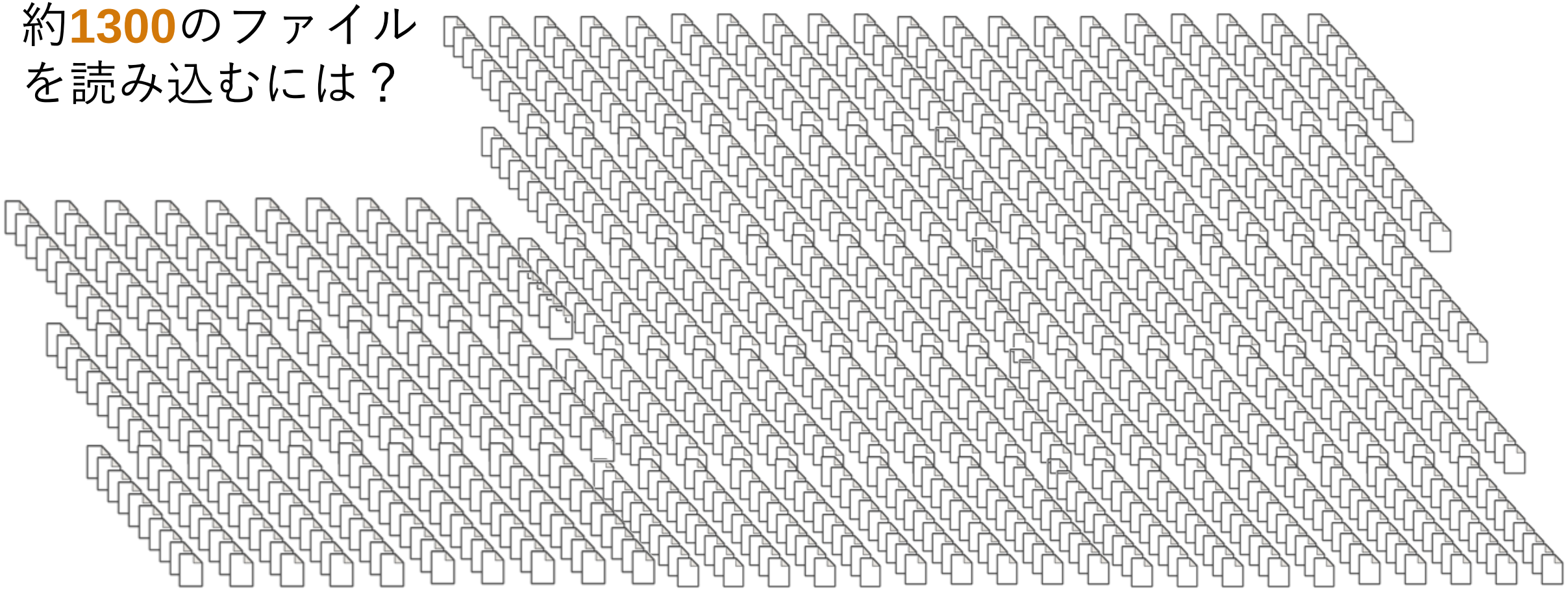




経度

フリートデータ解析実践
データへのアクセス

約**1300**のファイル
を読み込むには？



フリートデータ解析実践
データへのアクセス

datastore: データ、ファイルの集合体を読み取るオブジェクト
　特に**機械学習**や**ディープラーニング**で使用

対象データ	データストアの種類
表形式のテキストファイル (CSVなど)	TabularTextDatastore
Excel®形式のスプレッドシート(XLSXなど)	SpreadsheetDatastore
画像	ImageDatastore
リレーショナルデータベースのデータ	DatabaseDatastore
カスタム形式のファイル	FileDatastore
MDF形式のファイル	mdfDatastore

など

https://jp.mathworks.com/help/matlab/import_export/what-is-a-datastore.html

フリートデータ解析実践
データへのアクセス

トリップIDとVIN(車両識別番号)リストを読み取り

```
tripTable = readtable('tripTable.csv');
```

ワイルドカードで指定可能

個々のトリップファイルの読み取り

```
dataDir = fullfile('..', 'LogFiles', '*.mat');
```

fileDatastoreを使ったカスタム読み込み関数での読み取り

```
readFcn = @(filename) readTrip(filename, tripTable);
fds = fileDatastore(dataDir, 'ReadFcn', readFcn, 'UniformRead', true
disp(fds)
```

datastoreの作成

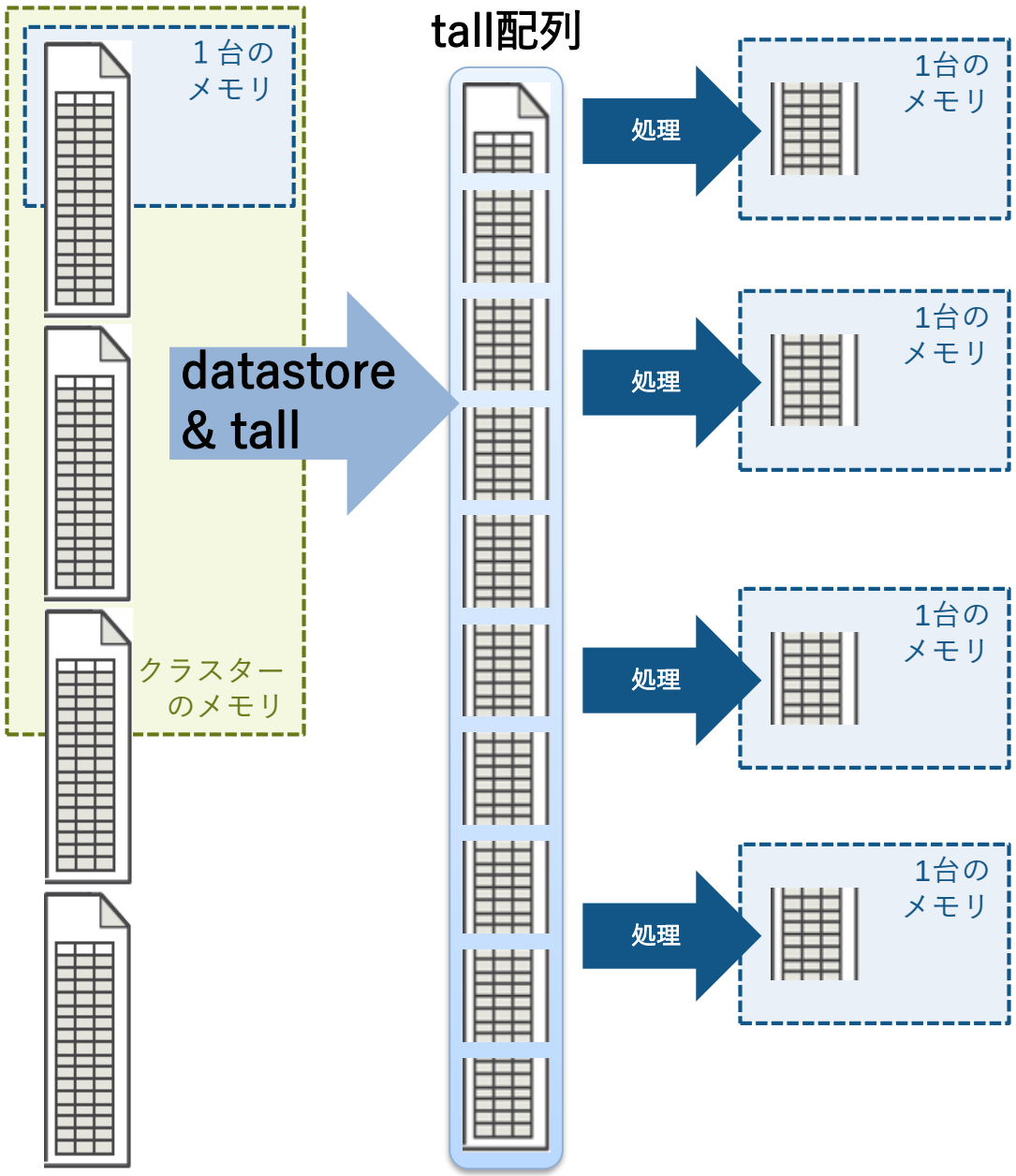
```
  FileDatastore のプロパティ:

                    Files: {
                           ' ...\FleetDataAnalytics\LogFiles\55a3fd0069702d5
                           ' ...\FleetDataAnalytics\LogFiles\55a3fd0169702d5
                           ' ...\FleetDataAnalytics\LogFiles\55a3fe3569702d5
                            ... and 1371 more
                           }
              UniformRead: 1
                  ReadFcn: @(filename)readTrip(filename,tripTable)
    AlternateFileSystemRoots: {}
```

カスタムの読込関数

```
function tOut = readTrip(fpath,tripTable)
%% 個々のトリップファイルを読み取り、整形する読込関数
% Copyright 2018 The MathWorks, Inc.

% [入力]
% fpath: 生データファイルへのパス
% tripTable: トリップIDとVINの関係が記載されているテーブル

% [出力]
% tOut: 整形されたタイムテーブル

%% ファイルを読み込み生データを抽出する
% .matファイルメモリーに読み込む
data = load(fpath);

% ファイルから構造体データを取り出す
data = struct2table(data.tripData,'AsArray',false);
```

フリートデータ解析実践
データへのアクセス　`tall`

- メモリに収まる小さな塊にデータを自動的に分割
- データアクセスを最適化して実行
- 並列演算もサポート



フリートデータ解析実践
データへのアクセス　tall

tall配列の作成

```
tt = tall(fds);
```

```
'local' プロファイルを使用して並列プール (parpool) を起動中.
2 ワーカーに接続されます。
```

tt

Mx4 tall timetable (未評価)

```
val =
  M×4 tall timetable
    time     trip_id     VIN     ChannelName     ChannelValue
    ____     _______     ___     ___________     ____________
     ?          ?         ?          ?               ?
     ?          ?         ?          ?               ?
     ?          ?         ?          ?               ?
     :          :         :          :               :
     :          :         :          :               :
```

関連ツール	tallでできること	
MATLAB	tall処理	ステップ1
+ Parallel Computing Toolbox™	ローカルマシンでの並列tall処理	ステップ1
+ MATLAB Parallel Server™	クラスターでの並列tall処理	ステップ2
+ *Apache™ Hadoop / Apache Spark* (サードパーティ)	Hadoop/Sparkクラスター上での並列tall処理	ステップ2

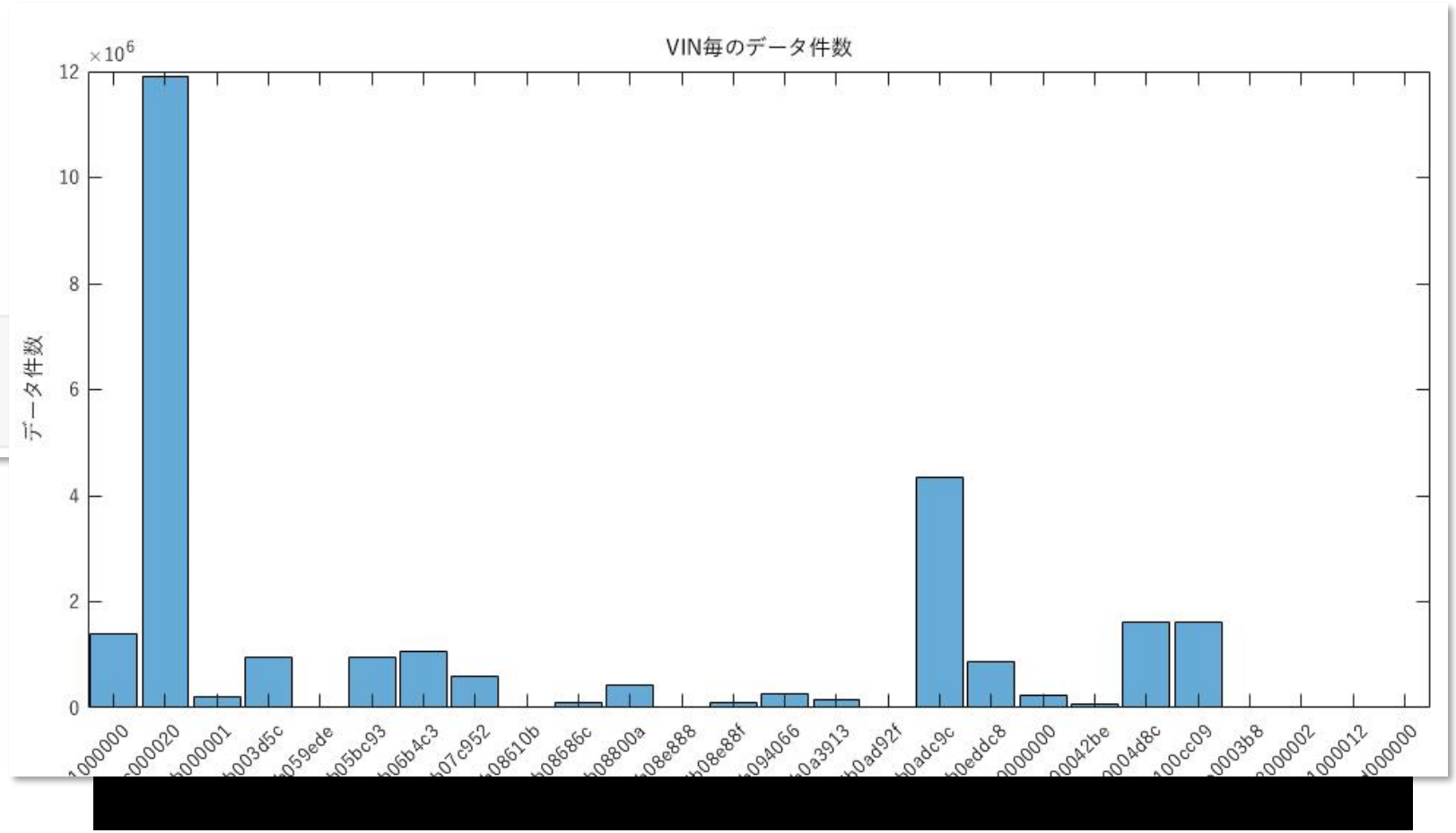
※MATLAB Distributed Computing Server ™は、R2019aからMATLAB Parallel Server ™に名称が変わりました。

フリートデータ解析実践
ビッグデータの可視化

データ全てを使って可視化
histogram

tall配列の可視化

```
%% どの車両が多いのかプロット
histogram(tt.VIN)
```



tall配列の可視化
https://www.mathworks.com/help/matlab/import_export/tall-data-visualization.html

フリートデータ解析実践
ビッグデータの可視化

データの緯度経度の散らばりを見たい

scatter

地理空間的な分布をプロット

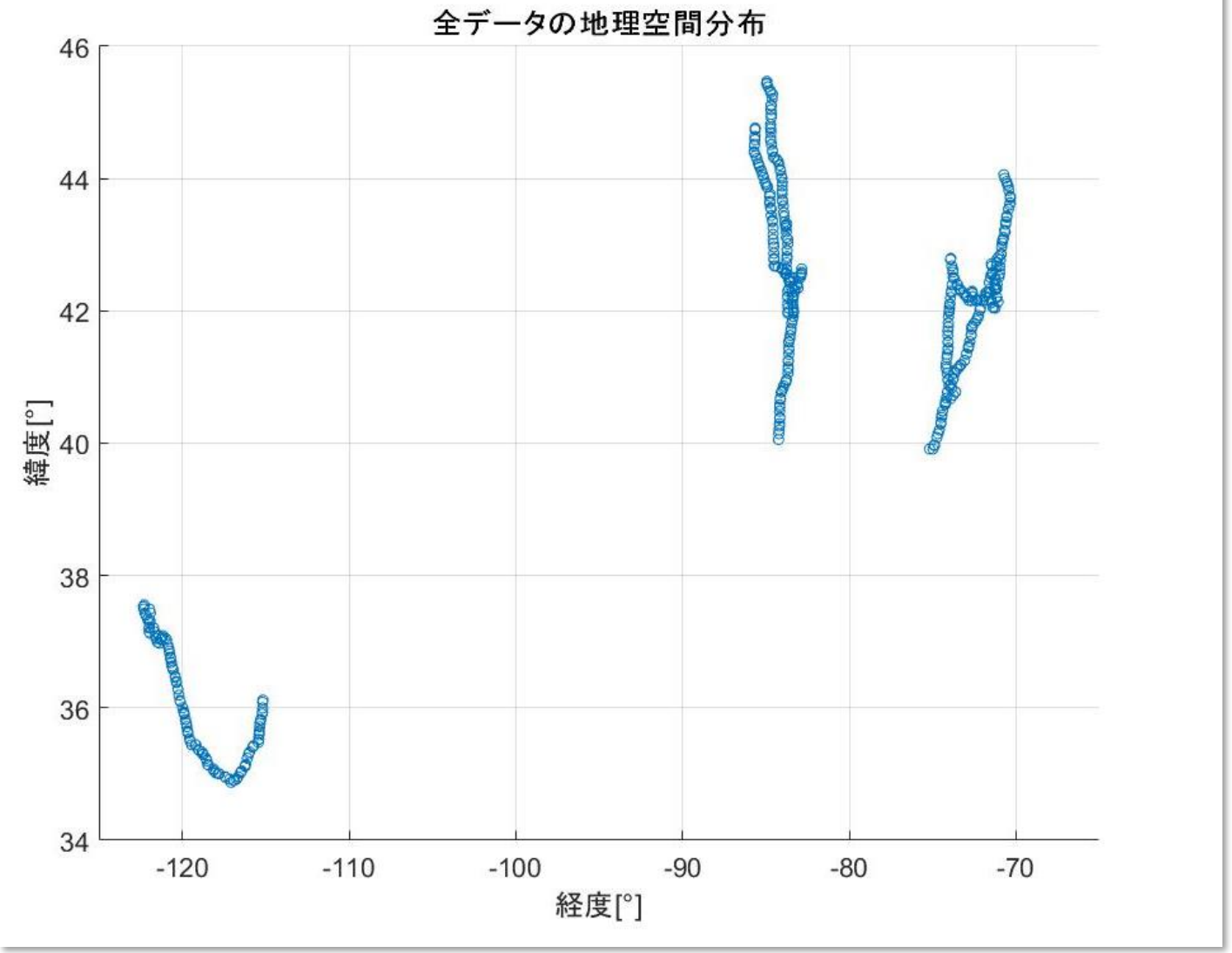
緯度経度だけ取り出し

メンバーを抽出

```
latChannel = 'kff1006';
lonChannel = 'kff1005';
mskLat = ismember(tt.ChannelName, latChannel);
mskLon = ismember(tt.ChannelName, lonChannel);
ttLat = tt(mskLat, :);
ttLon = tt(mskLon, :);
```

scatterプロット

```
scatter(ttLon.ChannelValue, ttLat.ChannelValue)
grid on
xlim([-125, -65])
ylim([34, 46])
xlabel('経度[°]', 'FontSize', 16)
ylabel('緯度[°]', 'FontSize', 16)
title('全データの地理空間分布');
set(gca,'FontSize',16);
```



フリートデータ解析実践
ビッグデータの可視化

地図上にプロットするには
メモリに取り込んでから
geoscatter R2018b

gatherを実行してメモリに取り込み

```
[ttLatGathered, ttLonGathered] = gather(ttLat, ttLon);
```

tall配列をメモリに取り込み

geoscatterプロット

```
gs = geoscatter(ttLatGathered.ChannelValue, ttLonGathered.ChannelValue);
gax = gca;
gax.FontSize = 16;
gax.Title.String = '全データの地理空間分布';
```

ベースマップの変更

```
name = 'openstreetmap';
url = 'a.tile.openstreetmap.org';
copyright = char(uint8(169));
attribution = copyright + "OpenStreetMap contributors";
displayName = 'Open Street Map';
addCustomBasemap(name,url,'Attribution',attribution,'DisplayName',displayName)
geobasemap('openstreetmap')
```

フリートデータ解析実践
ビッグデータの可視化

全トリップのトリップ時間を調べる

トリップIDでグループ分け

```
fileIdx = findgroups(ttSet.trip_id);
```

それぞれのグループをミニテーブルのcell配列に分割する

```
ttSet = splitapply(@(g) {g}, table(ttSet), fileIdx);
```

タイムスタンプの確認

時刻でソート

```
ttSet = cellfun(@(x) sortrows(x,'time','ascend'), ttSet,'UniformOutput', false);
```

トリップの所要時刻を求める

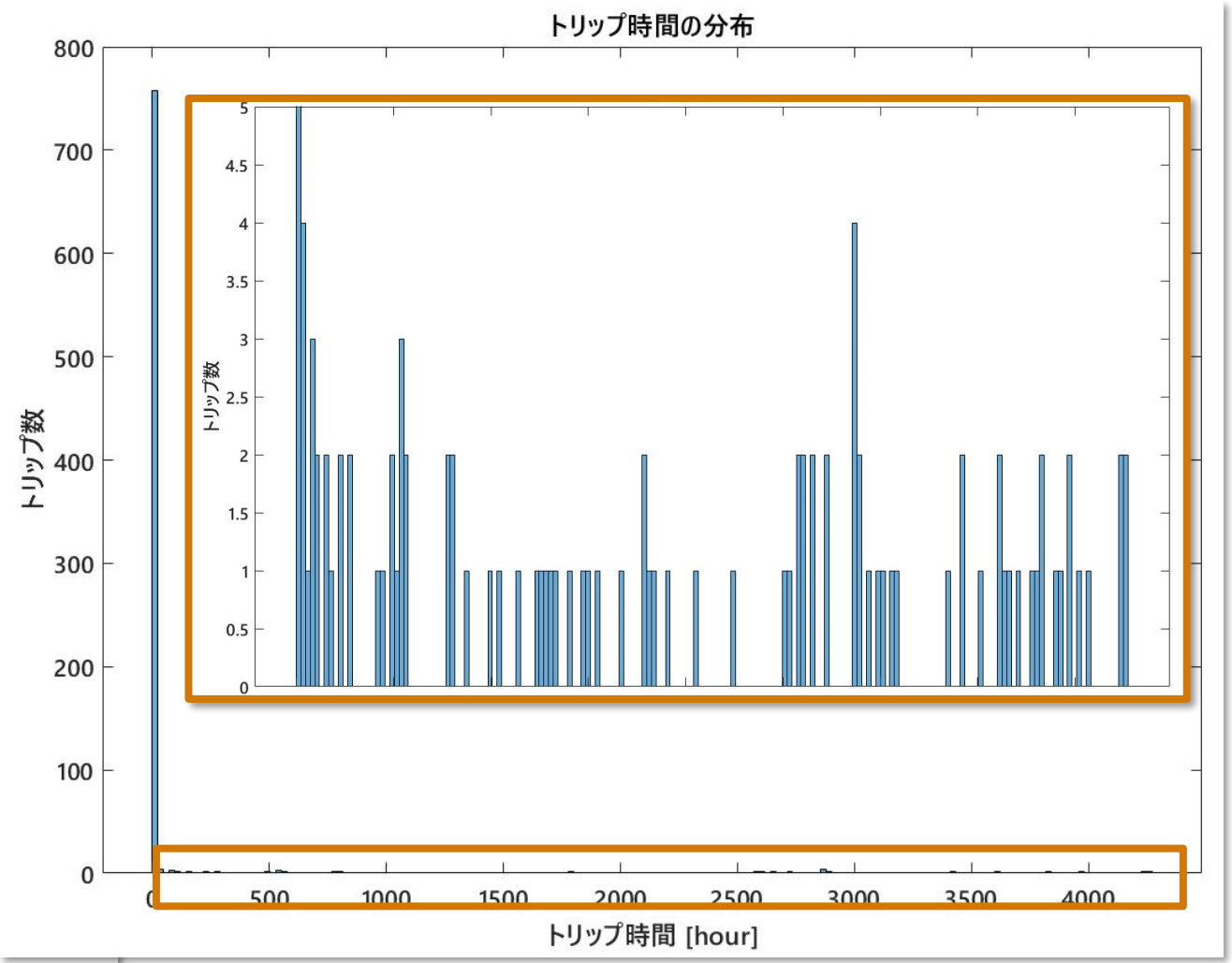
```
ttTripDuration = cellfun(@(x) seconds(x.time(end)-x.time(1)), ttSet,'UniformOutput'
```

hourに変換

```
numOfHours = cell2mat(ttTripDuration) / 60 / 60;
```

histogramのプロット

```
fig = figure;
ax = axes('Parent', fig);
histogram(ax, numOfHours, 'BinWidth', 24)
title(ax, 'トリップ時間の分布')
xlabel(ax, 'トリップ時間 [hour]','FontName','Yu Gothic UI Semibold');
ylabel(ax, 'トリップ数','FontName','Yu Gothic UI Semibold');
box(ax,'on');
set(ax,'FontName','Yu Gothic UI Semibold','FontSize',16);
```



フリートデータ解析実践
ビッグデータの可視化

不自然なトリップ時間を詳しく見てみる

```
numOfHoursGathered = gather(numOfHours);
```

最大のトリップ時間のデータを抽出

```
[maxNumOfHours, idx] = max(numOfHoursGathered);
ttOfMaxTripHours = ttSet(idx, :);
ttOfMaxTripHours = gather(ttOfMaxTripHours);
```

ttOfMaxTripHours{1, 1}

	time	1 trip_id	2 VIN	3 ChannelName	4 ChannelValue
3432	2015/01/15 22:44:21	55a41a...	55a3...	kff1006	42.474901140...
3433	2015/01/15 22:44:22	55a41a...	55a3...	kff1005	-83.62906784...
3434	2015/01/15 22:44:22	55a41a...	55a3...	kff1006	42.474908820...
3435	2015/01/15 22:44:23	55a41a...	55a3...	kff1005	-83.629061190...
3436	2015/01/15 22:44:23	55a41a...	55a3...	kff1006	42.474917500...
3437	2015/07/12 17:49:20	55a41a...	55a3...	kff1005	-84.985817060...
3438	2015/07/12 17:49:20	55a41a...	55a3...	kff1006	45.435388420...
3439	2015/07/12 17:49:21	55a41a...	55a3...	kff1005	-84.98563572...
	5/07/12 17:49:21	55a41a...	55a3...	kff1006	45.435394190...
3441	2015/07/12 19:00:38	55a41a...	55a3...	kff1005	-84.68864308...
3442	2015/07/12 19:00:38	55a41a...	55a3...	kff1006	44.878871850...

半年ほど間隔が空いている

フリートデータ解析実践
ビッグデータの前処理

- 前処理

タイムスタンプ間隔が24hourより大きいデータを消去

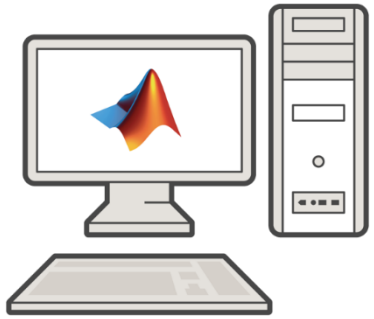
```
thd = hours(24);
ttSet = RejectSpuriousTimestamps(ttSet, thd);
```

```
function cellOut = RejectSpuriousTimestamps(cellIn,thd)
%% データ内の不自然なタイムスタンプを消去する
% Copyright 2018 The MathWorks, Inc.

%% トリップデータのタイムスタンプをソート
cellIn = cellfun(@(x) sortrows(x,'time','ascend'), cellIn,'UniformOutput', false);

% 不自然なタイムスタンプの行を消去する関数ハンドルの定義
removeSpuriousTimeFcn = @(T)DeleteRows(T,thd);

% 不自然なタイムスタンプ行の消去
cellOut = cellfun(removeSpuriousTimeFcn, cellIn, 'UniformOutput', false);
end

%% タイムスタンプの行を消去する関数
function out = DeleteRows(in,thd)
dt = vertcat(0,diff(in.time));
msk = dt > thd;

if sum(msk) == 0
    out = in;
else
    idx = find(msk,1);
    % 行の消去 (事前に時刻でソートしておくこと)
    in(idx:end,:) = [];
    out = in;
end
end
```

フリートデータ解析実践
クラスターへのスケールアウト

ステップ2

デスクトップの限界



- 処理時間
- データコピーの手間
- ディスク容量

Hadoop/Spark

Part 2に続く