

統計モデリング入門

実は分かっていないけど
仕事で一番使う”線形回帰“を
懇切丁寧に

MathWorks Japan

吉野 紘和, Special thanks to 王 曉星

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Agenda

理論編 (1)

- 信頼区間と予測区間
 - イントロ + 線形回帰とは
 - モデル推定手法
 - 信頼区間と予測区間の違い

理論編 (2)

- モデルの選択方法は?
 - 決定係数 と AIC
 - SE, t値, p値
- 脆い線形回帰
 - 多重共線性
 - データの標準化

実践編

- 理論編の総復習
- 交差相関にご用心

理論編 (1)

- 信頼区間と予測区間
 - イントロ + 線形回帰とは
 - モデル推定手法
 - 信頼区間と予測区間の違い

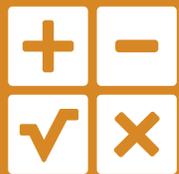
理論編 (1)

- 信頼区間と予測区間
 - イントロ + 線形回帰とは
 - モデル推定手法
 - 信頼区間と予測区間の違い

線形回帰分析のワークフロー

正規化/標準化

- 4 zscore
- 4 normalize
- 4 LiveTask 活用



データのチェック

- 多重共線性
 - 3 corrcoef
 - 3 collintest
 - 3 corrplot
- 相互相関
 - 5 xcorr
 - 5 crosscorr

モデル作成

- 1 ウィルキンソンの表記法
- 1 stepwise
- 1 正則化



モデル選択

- 2 t値, p値
- 2 決定係数
- 2 情報量基準, AIC
- 2 標準誤差 (SE)



スプレッドシートデータから燃費を推論するモデルを構築

MPG = Miles Per Gallon (燃費: km/Litter のようなもの)

	A	B	C	D	E	F	G
1	Weight	Acceleration	MPG	Horsepower	Cylinders	Displacement	Model_Year
2	3504	12	18	130	8	307	70
3	3693	11.5	15	165	8	350	70
4	3436	11	18	150	8	318	70
5	3433	12	16	150	8	304	70
6	3449	10.5	17	140	8	302	70
7	4341	10	15	198	8	429	70
8	4354	9	14	220	8	454	70
9	4312	8.5	14	215	8	440	70
10	4425	10	14	225	8	455	70
11	3850	8.5	15	190	8	390	70
12	3090	17.5		115	4	133	70
13	4142	11.5		165	8	350	70
14	4034	11		153	8	351	70
15	4166	10.5		175	8	383	70
16	3850	11		175	8	360	70
17	3563	10		170	8	383	70
18	3609	8		160	8	340	70
19	3353	8		140	8	302	70
20	3761	9.5		150	8	400	70
21	3086	10		225	8	455	70
22	2372	15		95	4	113	70
23	2833	15.5		95	6	198	70
24	2774	15.5		97	6	199	70
25	2587	16		85	6	200	70
26	2130	14.5		88	4	97	70
27	1835	20.5		46	4	97	70

欠損値

MPG を他の6つの予測子から推論

- Weight: 重量
- Acceleration: 加速度
- Horsepower: 馬力
- Cylinders: 気筒数
- Displacement: 排気量
- Model_Year: 車体年式



開発環境へのデータのインポート

- データインポートアプリ
 - マウス操作でファイルからデータをインポート
 - MATLABコード自動生成
 - 関数化
 - スクリプト化



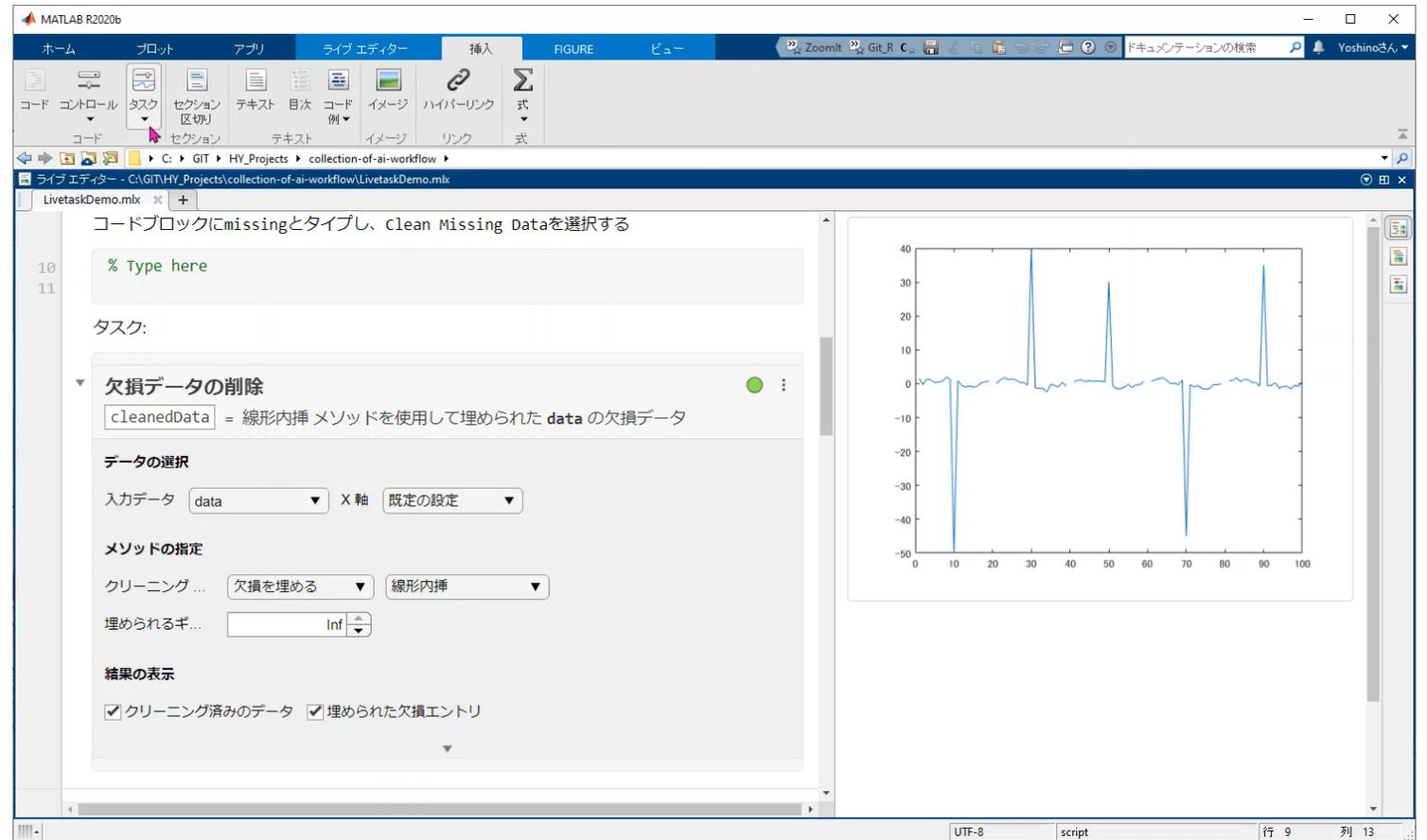
Weight	Acceleration	MPG	Horsepower	Cylinders	Displacement	Model_Year
3504	12	18	130	8	307	70
3693	11.5000	15	165	8	350	70
3436	11	18	150	8	318	70
3433	12	16	150	8	304	70
3449	10.5000	17	140	8	302	70
4341	10	15	198	8	429	70
4354	9	14	220	8	454	70
4312	8.5000	14	215	8	440	70
4425	10	14	225	8	455	70
3850	8.5000	15	190	8	390	70
3090	17.5000		115	4	133	70
4142	11.5000		165	8	350	70
4034	11		153	8	351	70
4166	10.5000		175	8	383	70
3850	11		175	8	360	70
3563	10	15	170	8	383	70
3609	8	14	160	8	340	70
3353	8		140	8	302	70
3761	9.5000	15	150	8	400	70

データインポートアプリ

前処理における試行錯誤を高速化する“LiveTask”

前処理パラメータの変更を即反映

	グループ別に計算 R2021b グループごとに要約、変換、またはフィルター処理を行います
	データの平滑化 ノイズを含むデータを平滑化します
	データの正規化 R2021b データのセンタリングとスケールリングを行います
	トレンドの除去 多項式のトレンドをデータから除去します
	変化点の検出 データ内の急激な変化を検出します
	外れ値データの削除 外れ値の検出、埋め込み、または削除を行います
	局所的極値の検出 局所的な最大値と局所的な最小値を検出します
	欠損データの削除 欠損データの検出、埋め込み、または削除を行います



MATLAB R2020b

ホーム プロット アプリ ライブ エディター 挿入 FIGURE ビュー

コード コントロール タスク セクション区切り テキスト 目次 コード例 イメージ ハイパーリンク 式

コード コード セクション テキスト イメージ リンク

ライブ エディター - C:\GIT\HY_Projects\collection-of-ai-workflow\collection-of-ai-workflow

LivetaskDemo.mlx

コードブロックにmissingとタイプし、Clean Missing Dataを選択する

```
10 % Type here
11
```

タスク:

▼ 欠損データの削除

cleanedData = 線形内挿 メソッドを使用して埋められた data の欠損データ

データの選択

入力データ data X 軸 既定の設定

メソッドの指定

クリーニング... 欠損を埋める 線形内挿

埋められるギ... Inf

結果の表示

クリーニング済みのデータ 埋められた欠損エントリ

40
30
20
10
0
-10
-20
-30
-40
-50

0 10 20 30 40 50 60 70 80 90 100

UTF-8 script 行 9 列 13

- データサイエンスに必要な“タスク”を半GUIで実現
- MATLABコードも自動生成

27の Live Task がコーディングの可能性を広げる

前処理パラメータの変更を即反映 & 作業の再利用のためのコードも提供



R2021b

DATA AND VISUALIZATION	
	Create Plot
DATA PREPROCESSING	
	Clean Missing Data
	Clean Outlier Data
	Find Change Points
	Find Local Extrema
	Remove Trends
	Smooth Data

TABLES AND TIMETABLES	
	Join Tables
	Retime Timetable
	Stack Table Variables
	Synchronize Timetables
	Unstack Table Variables
OPTIMIZATION	
	Optimize

CONTROL SYSTEM DESIGN AND ANALYSIS	
	Convert Model Rate
	Reduce Model Order
	Tune PID Controller
PREDICTIVE MAINTENANCE	
	Estimate Approximate Entropy
	Estimate Correlation Dimension
	Estimate Lyapunov Exponent
	Extract Spectral Features
	Reconstruct Phase Space

SYSTEM IDENTIFICATION	
	Estimate Process Model
	Estimate State-Space Model
SIGNAL PROCESSING AND COMMUNICATIONS	
	Extract Audio Features
SYMBOLIC MATH	
	Simplify Symbolic Expression
	Solve Symbolic Equation
IMAGE ACQUISITION	
	Acquire Webcam Image

Engineering と Science を加速する

```

# weather.py
import csv
import datetime
import json
import urllib.request

BASE_URL = "https://api.openweathermap.org/data/2.5/{city}({units}:{Metric})"
PARAMS_JSON = { "current": True, "units": "temp", "lang": "japanese",
                "speed": "m/s", "humidity": "no", "pressure": "no" }

def read_horizon(city):
    """Read example data from a horizon file"""
    with open('backuptele.csv', 'w') as csvfile:
        reader = csv.DictReader(csvfile)
        for s in ['reader']:
            city = city
            data = dict(s)
            data['city'] = city
            if len(data) > 0:
                for k in ('temp', 'pressure', 'humidity', 'temp_min',
                        'temp_max', 'speed', 'sea', 'sun', 'lon'):
                    data[k] = float(data[k])
            return data
    return None

def get_current_weather(city, country, apiKey, **kwargs):
    """Get current weather in specific location
    """

```

本質的な課題

非本質的作業:

プログラミング、ライブラリ間の繋ぎ合わせ、
ツール間の整合性、実装、バージョン管理、知財...

MATLAB®
& SIMULINK®

本質的な課題 + α

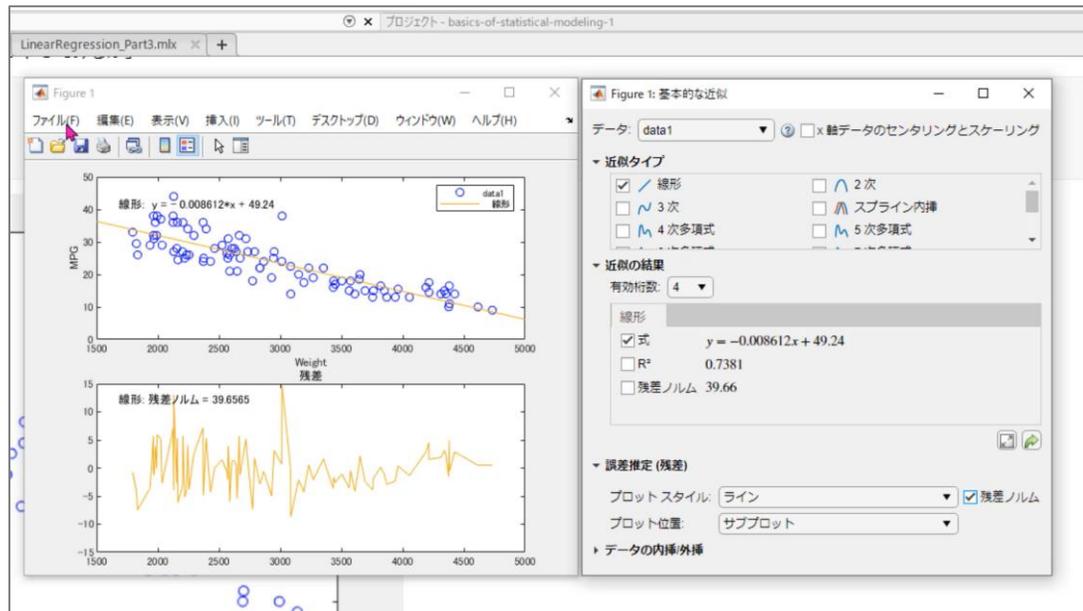
最小限の
非本質的作業

Accelerating the pace of Engineering and Science

表計算ソフトレベルの単回帰は”秒”で片づける

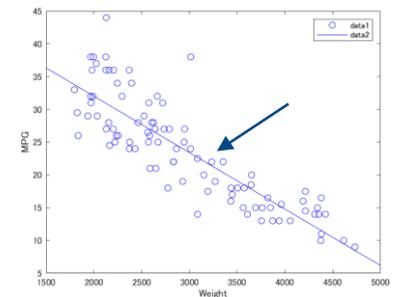
$$\text{MPG} = \beta_0 + \beta_1 \cdot \text{Weight} + \epsilon$$

- 単回帰
 - 説明変数が1つのもの
 - 燃費を車両重量で回帰



- **MATLAB plot 関数**

- 多項式、スプライン等の基本関数近似
- 残差プロットあり
- MATLAB コード自動生成
- 最小2乗線を最速で引く
 - refline



plot 関数 > ツール > 基本的な近似

理論編 (1)

- 信頼区間と予測区間
 - イントロ + 線形回帰とは
 - モデル推定手法
 - 信頼区間と予測区間の違い

線形回帰モデルとは

応答 (目的) 変数が予測子 (説明変数) の線形結合

$$y_1 = \beta_{1,0} + \beta_{1,1}x_{1,1} + \beta_{1,2}x_{1,2} + \cdots + \beta_{1,M}x_{1,M} + \epsilon_1$$

$$y_2 = \beta_{2,0} + \beta_{2,1}x_{2,1} + \beta_{2,2}x_{2,2} + \cdots + \beta_{2,M}x_{2,M} + \epsilon_2$$

⋮

$$y_N = \beta_{N,0} + \beta_{N,1}x_{N,1} + \beta_{N,2}x_{N,2} + \cdots + \beta_{N,M}x_{N,M} + \epsilon_N$$

- N 個 (次元) の観測 \mathbf{y}
- $M + 1$ 次元の係数 β
- 正規分布に従う
 N 次元ノイズ ϵ



$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

応答変数

回帰係数
ベクトル

ノイズ

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,M} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \cdots & x_{N,M} \end{bmatrix}$$

説明変数行列

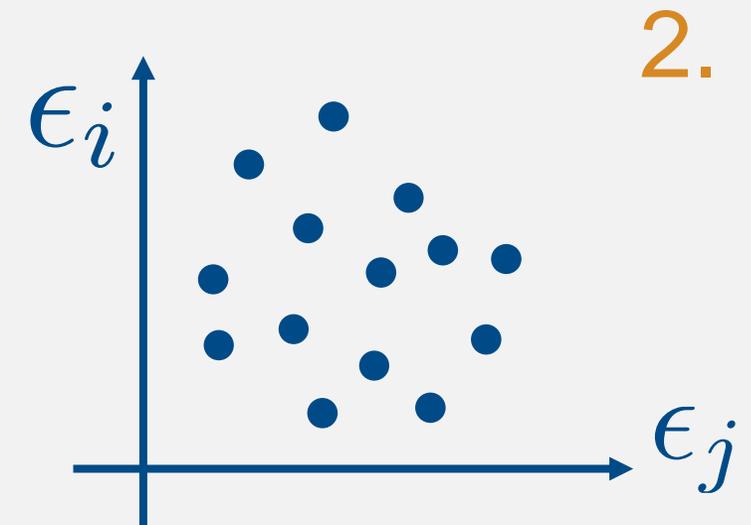
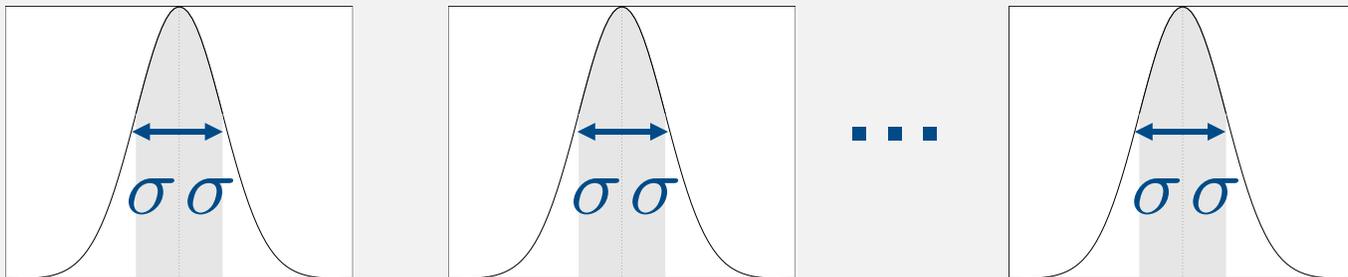
where, $\mathbf{y} \in \mathcal{R}^N$, $\beta \in \mathcal{R}^{M+1}$, and $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$

線形回帰モデルとは

$$\epsilon \sim N(\mathbf{0}, \sigma^2 I)$$

1. 各々の成分が正規分布に従う
2. 各々の成分の相関はゼロ

1. ϵ_1 ϵ_2 ... ϵ_N



線形回帰モデルを作成 ウィルキンソンの表記法

燃費を車両重量で(単)回帰

$$\text{MPG} = \beta_0 + \beta_1 \cdot \text{Weight} + \epsilon$$

$$= \begin{bmatrix} 1 & \text{Weight} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \epsilon$$

→ MPG ~ 1 + Weight

ウィルキンソンの表記法

線形回帰モデル
オブジェクト

```
linear mdl = fitlm(tbl, "MPG~1+Weight")
```

データ モデル指定

線形回帰モデル:
MPG ~ 1 + Weight

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	49.238	1.6504	29.834	9.0258e-49
Weight	-0.0086118	0.00053775	-16.014	3.2405e-28

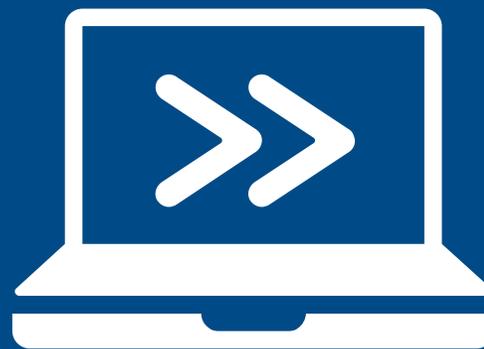
観測数: 93、誤差の自由度: 91
平方根平均二乗誤差: 4.16
決定係数: 0.738、自由度調整済み決定係数: 0.735
F 統計量 - 定数モデルとの比較: 256、p 値は 3.24e-28 です

ウィルキンソンの表記法まとめ

モデル内の予測子項	ウィルキンソンの表記法
切片	1
切片なし	-1
X1	x1
X1, X2	x1 + x2
X1, X2, X1X2	x1*x2 または x1 + x2 + x1:x2
X1X2	x1:x2
X1, X1 ²	x1^2
X1 ²	x1^2 - x1

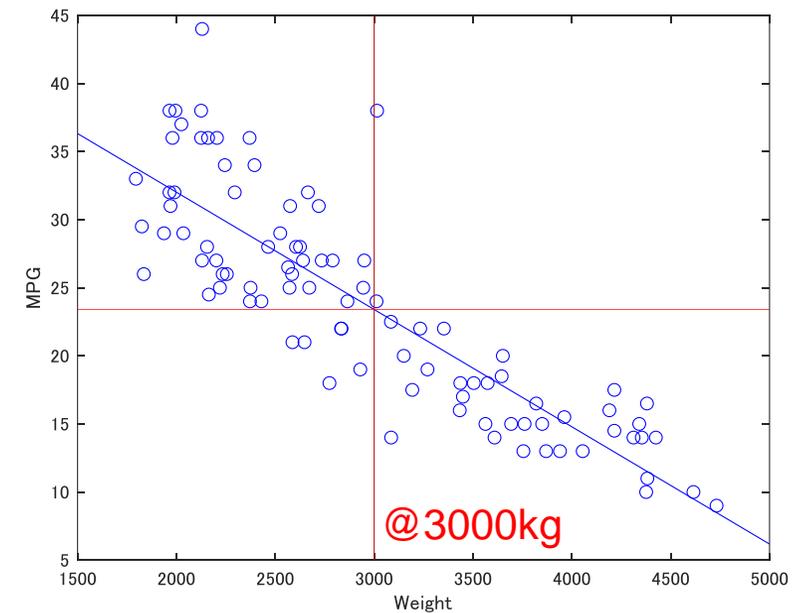
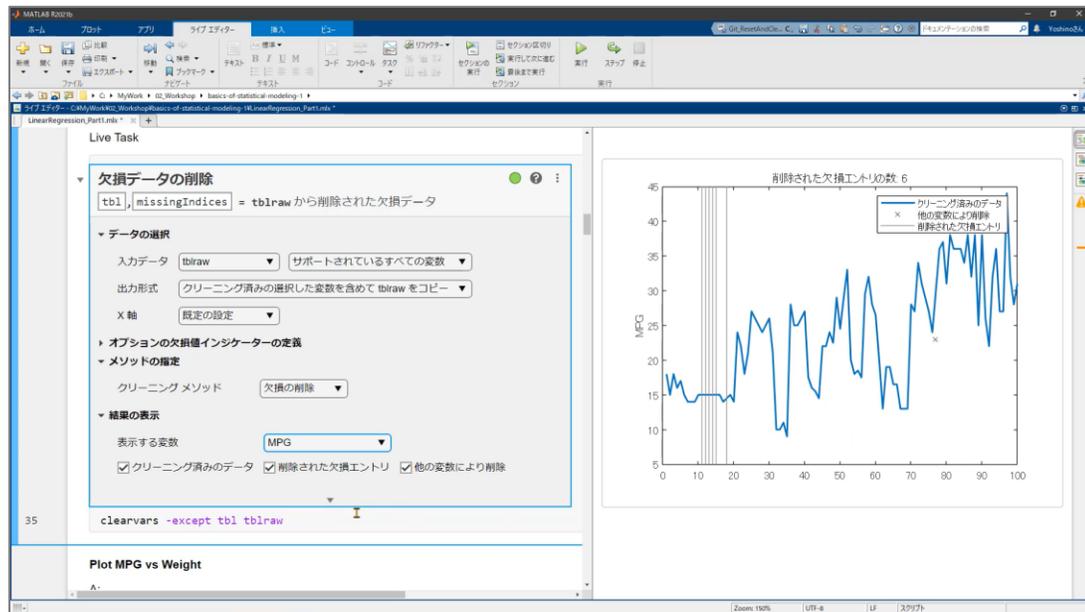
[詳細はこちら](#)

デモンストレーション



Demo.1: 従来の“線形回帰”の作業

1. ライブタスクを使った前処理
2. refline
3. 線形回帰モデルの作成
4. 信頼区間のプロット
5. ある値における推論

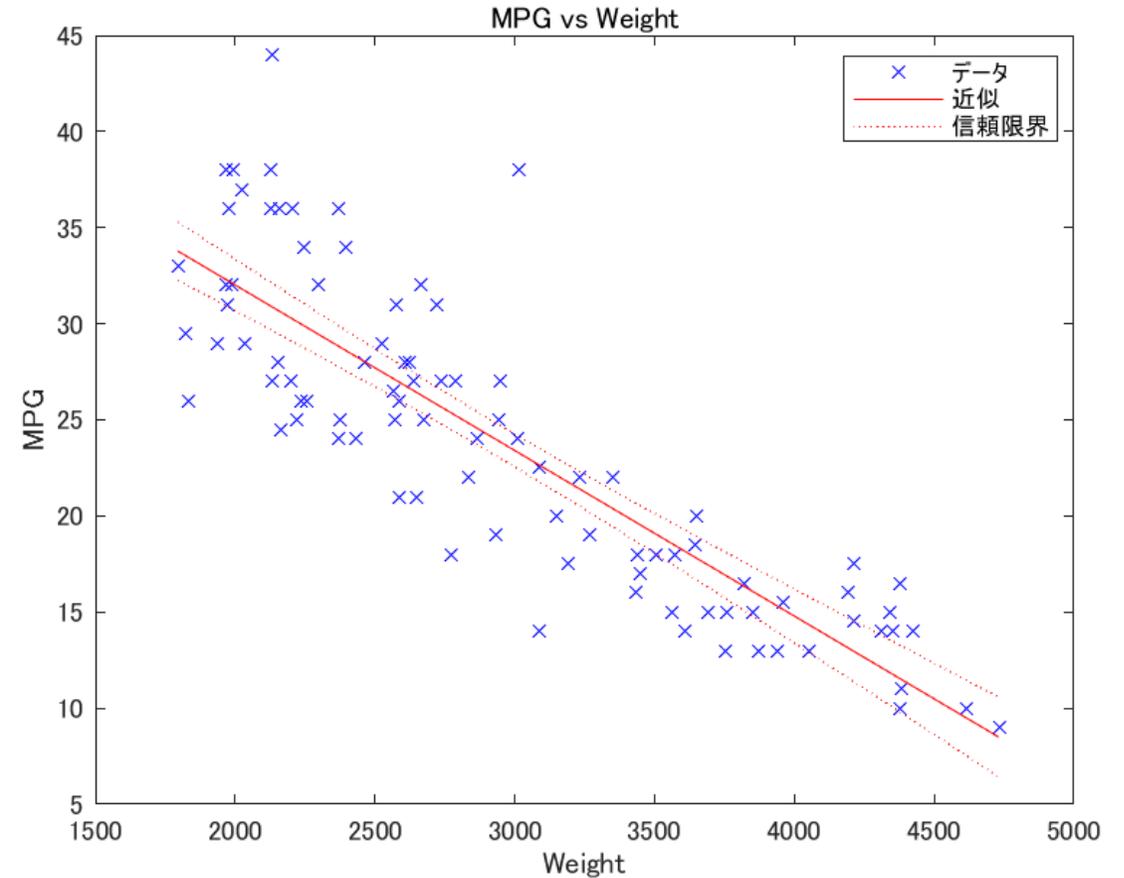


線形回帰モデルから分析をする

- `plot(LM_model)`
表示される赤い線3本 – 信頼区間
- `feval(LM_model, x)`
でモデルを x において評価



Q: 信頼区間とは何でしょうか？



理論編 (1)

- 信頼区間と予測区間
 - イントロ + 線形回帰とは
 - モデル推定手法
 - 信頼区間と予測区間の違い

線形回帰モデルの推定

- 最小二乗法

$$\mathbf{y} \approx X\boldsymbol{\beta}$$

二乗誤差

$$J \equiv \|\mathbf{y} - X\boldsymbol{\beta}\|^2 \text{ を最小化}$$

- 最尤推定法

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

平均 分散共分散行列

$$\mathbf{y} \sim \text{N}(\mathbf{y} | X\boldsymbol{\beta}, \sigma^2 I)$$

$$\text{尤度 } L(\boldsymbol{\beta}) \equiv \text{P}(\mathbf{y} | \boldsymbol{\beta}) \text{ を最大化}$$

$$\rightarrow \hat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \mathbf{y}$$

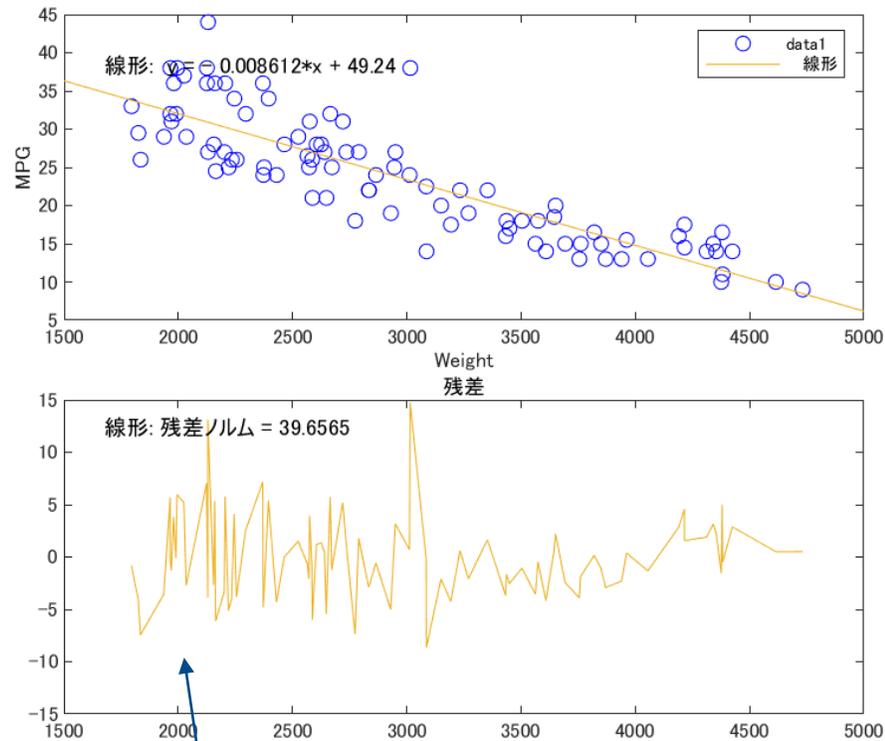
最小二乗法 = 最尤推定法 + “ノイズ~正規分布”

線形回帰モデルの推定イメージ



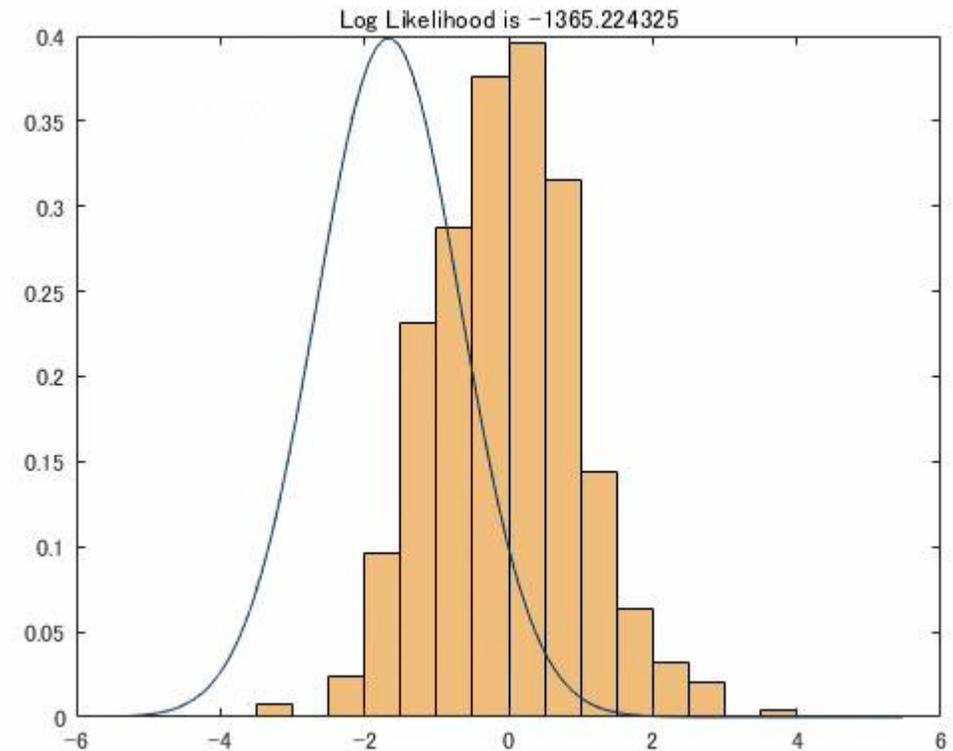
統計学に基づく
多くの情報を得られる

- 最小二乗法 $J \equiv \| \mathbf{y} - X\boldsymbol{\beta} \|^2$



残差を最小化する
パラメータを探索

- 最尤推定法 $L(\boldsymbol{\beta}) \equiv P(\mathbf{y}|\boldsymbol{\beta})$



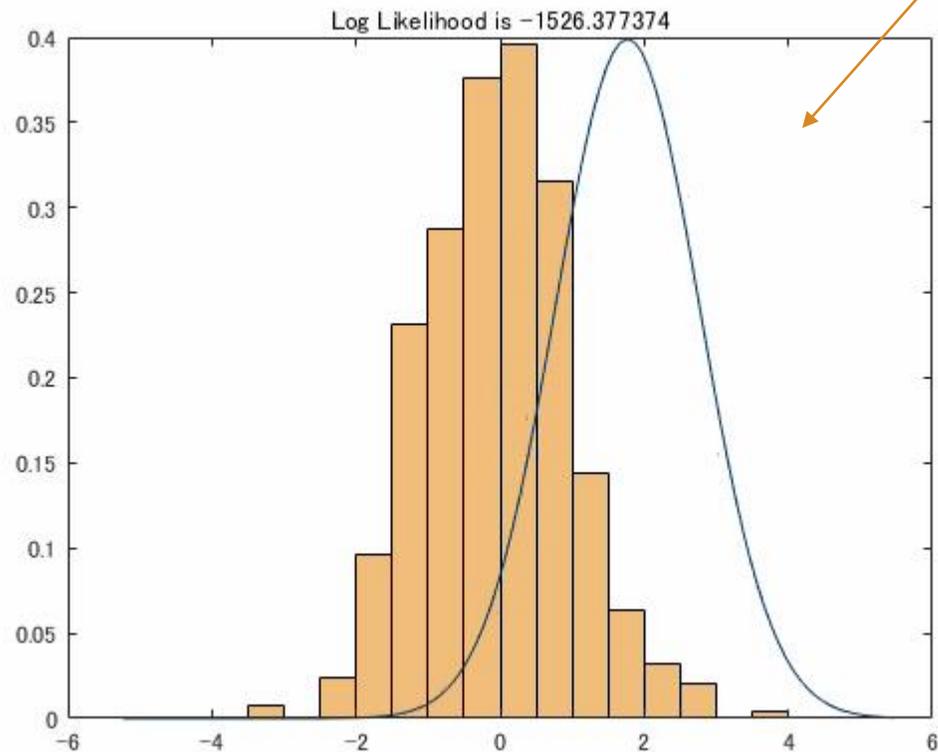
パラメータが与えられた時に手持ちの
データが得られる確率 (尤度) を最大化

線形回帰モデルの推定イメージ (Cont.)

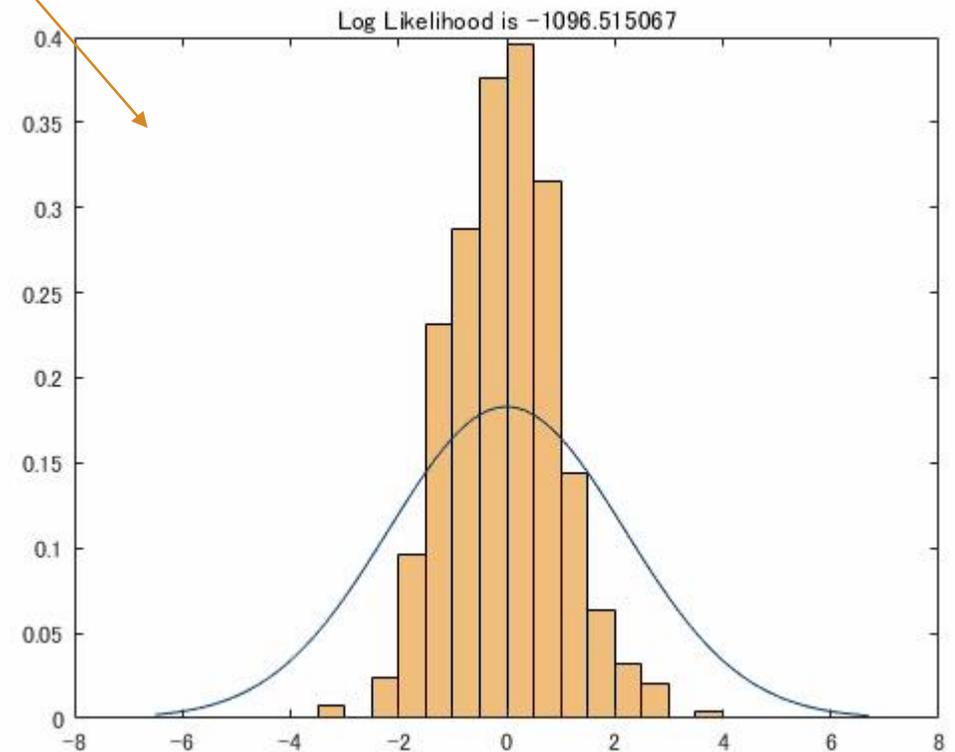
線形回帰モデル内のパラメータを推定する

$$\mathbf{y} \sim \mathcal{N}(\mathbf{y} | X\boldsymbol{\beta}, \sigma^2 I)$$

統計学に基づく
多くの情報を得られる



平均値



分散値

理論編 (1)

- 信頼区間と予測区間
 - イントロ + 線形回帰とは
 - モデル推定手法
 - 信頼区間と予測区間の違い

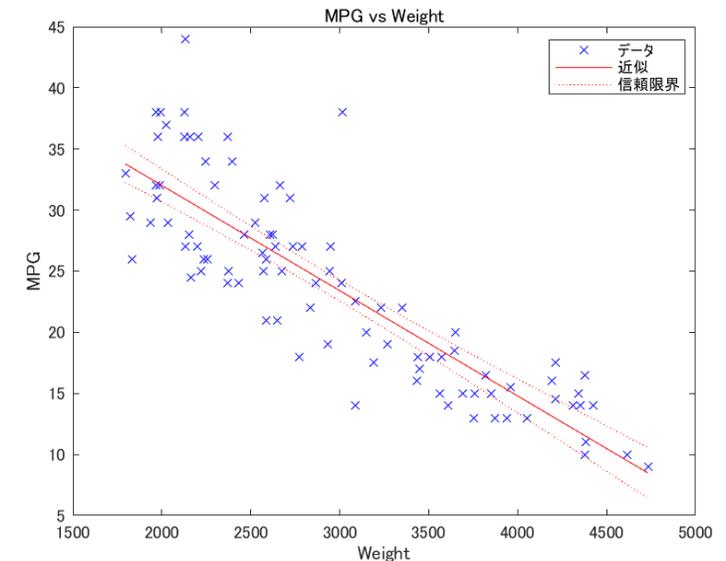
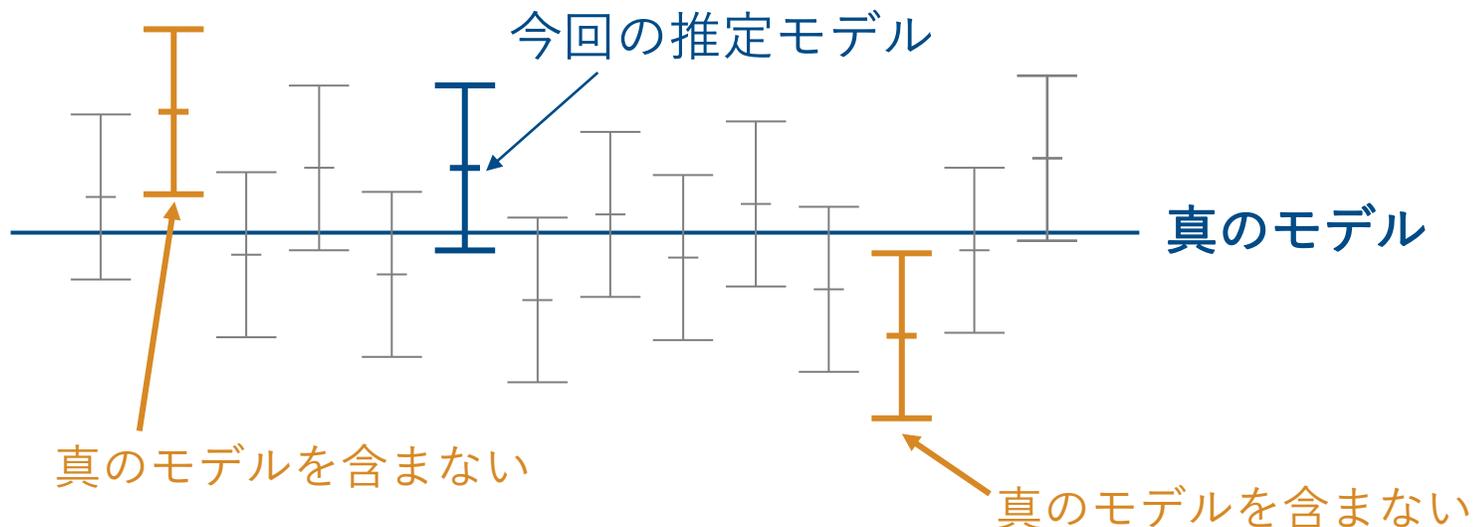
信頼区間と予測区間

信頼区間の幅の源

$$y = \mathbf{x}^T \beta$$

■ 信頼区間 / Confidence Interval (95%)

- 同じ手順でモデル構築を100回繰り返した際、**95**個のモデルの信頼区間に真の値が含まれる (毎回異なる信頼区間が作られることに注意)
- 再度データを取得した際に収まる範囲を示しているわけではない



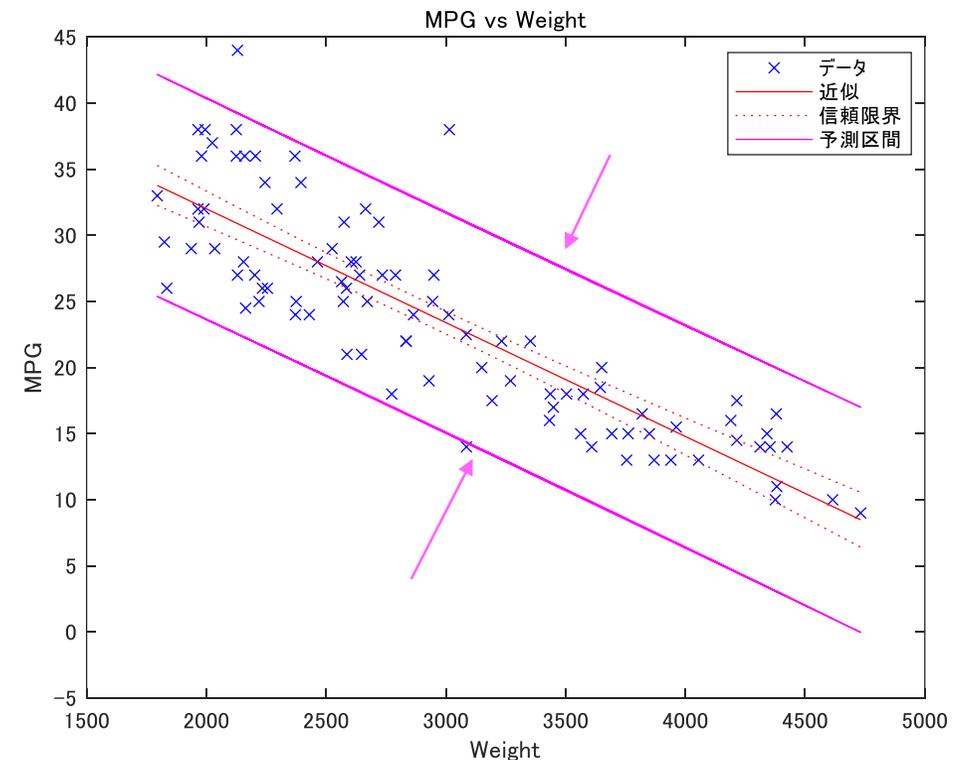
信頼区間と予測区間 (Cont.)

■ 予測区間 / Prediction Interval

- もう1度データを取得した場合、測定ノイズ分の不確かさも考慮して、データがどのくらいの範囲に収まるかを表す。
- モデルの不確かさ + 測定の不確かさ
- 区間がより広くなる

予測区間の幅の源

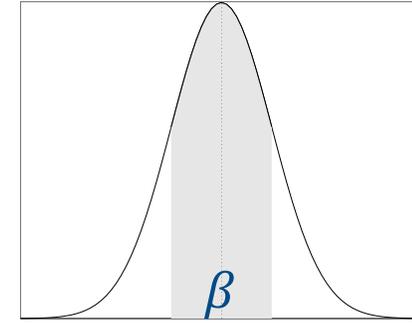
$$y = \mathbf{x}^T \beta + \epsilon$$



信頼区間と予測区間 (Cont.)

パラメータの推定値の解析解

$$\begin{cases} \mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \\ \hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \end{cases}$$



$$\hat{\boldsymbol{\beta}} \sim \mathbf{N}(\hat{\boldsymbol{\beta}} | \boldsymbol{\beta}, \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1})$$

パラメータの推定値は正規分布に従う確率変数

確率変数 = あるルールの下でばらつく数

信頼区間: (1) のばらつき

$$(1) \quad y = \mathbf{x}^\top \boldsymbol{\beta}$$

予測区間: (2) のばらつき

$$(2) \quad y = \mathbf{x}^\top \boldsymbol{\beta} + \boldsymbol{\epsilon}$$

信頼区間と予測区間 (補足)

分散の不変推定量の確率密度分布

$$\nu_e = N - \text{rank}(X)$$

$$\hat{\sigma}^2 = \frac{1}{\nu_e} \|\mathbf{y} - X\hat{\beta}\|^2, \quad \mathbf{E}[\hat{\sigma}^2] = \sigma^2 \quad \text{不変推定量}$$

$$\frac{\|\mathbf{y} - X\hat{\beta}\|^2}{\sigma^2} \sim \chi^2(\nu_e) \Rightarrow \nu_e \frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2(\nu_e)$$

信頼区間と予測区間 (補足)

信頼区間

$$E[\hat{y}] = \mathbf{x}^\top \boldsymbol{\beta} \quad \text{Note that } \hat{y} = \mathbf{x}^\top \hat{\boldsymbol{\beta}}$$

$$V[\hat{y}] = \sigma^2 \mathbf{x}^\top (X^\top X)^{-1} \mathbf{x}$$

t分布に従うため
ここから信頼区間が評価できる

$$\Rightarrow T = \frac{\hat{y} - \mathbf{x}^\top \boldsymbol{\beta}}{\hat{\sigma}^2 \sqrt{\mathbf{x}^\top (X^\top X)^{-1} \mathbf{x}}} \sim t(\nu_e)$$

信頼区間と予測区間 (補足)

予測区間

$$\begin{cases} y = \mathbf{x}^\top \beta + \epsilon \\ \hat{y} = \mathbf{x}^\top \hat{\beta} \\ e \equiv y - \hat{y}, e \sim N(e|0, V[e]) \end{cases}$$

where,

$$\begin{aligned} V[e] &= V[y] + V[\mathbf{x}^\top \hat{\beta}] \\ &= \sigma^2 (1 + \mathbf{x}^\top (X^\top X)^{-1} \mathbf{x}) \end{aligned}$$

t分布に従うため
ここから信頼区間が評価できる

$$\Rightarrow T = \frac{\frac{e}{\sqrt{V[e]}}}{\sqrt{\frac{\nu_e \hat{\sigma}^2}{\sigma^2} \frac{1}{\nu_e}}} = \frac{y - \mathbf{x}^\top \hat{\beta}}{\hat{\sigma} \sqrt{1 + \mathbf{x}^\top (X^\top X)^{-1} \mathbf{x}}} \sim t(\nu_e)$$

信頼区間と予測区間 (補足)

”幅が広がる”直感的理解

$$\hat{y} = \mathbf{x}^\top \hat{\beta}$$

信頼区間

$$y = \mathbf{x}^\top \beta$$

評価したい
信頼区間

$$T = \frac{\mathbf{x}^\top \hat{\beta} - y}{\hat{\sigma} \sqrt{\mathbf{x}^\top (X^\top X)^{-1} \mathbf{x}}} \sim t(\nu_e)$$

予測区間

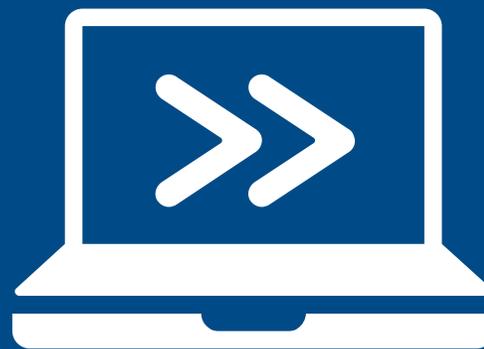
$$y = \mathbf{x}^\top \beta + \epsilon$$

評価したい
予測区間

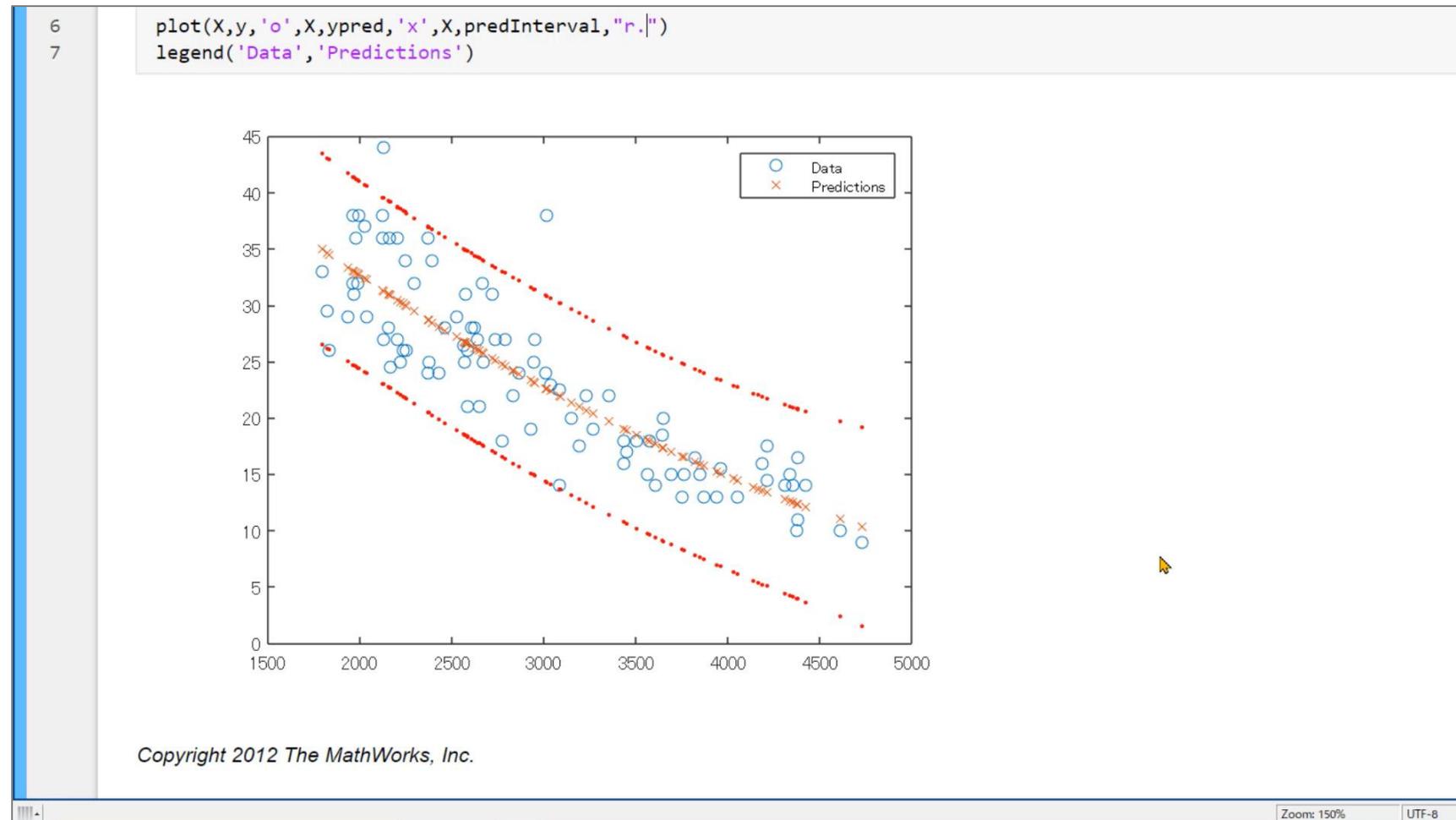
$$T = \frac{y - \mathbf{x}^\top \hat{\beta}}{\hat{\sigma} \sqrt{1 + \mathbf{x}^\top (X^\top X)^{-1} \mathbf{x}}} \sim t(\nu_e)$$

デモンストレーション

linear_md1

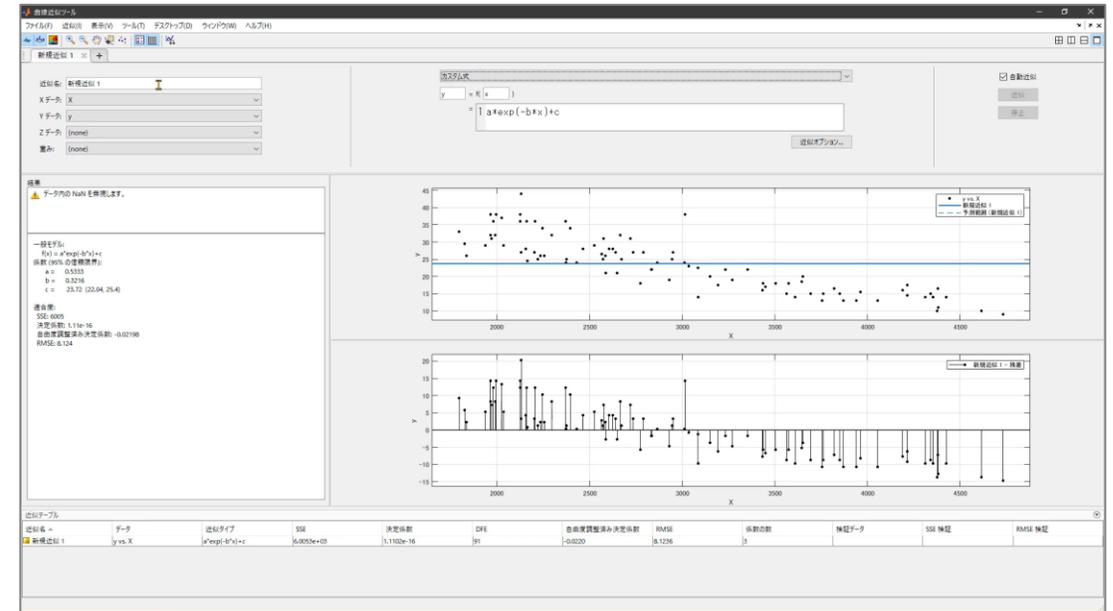
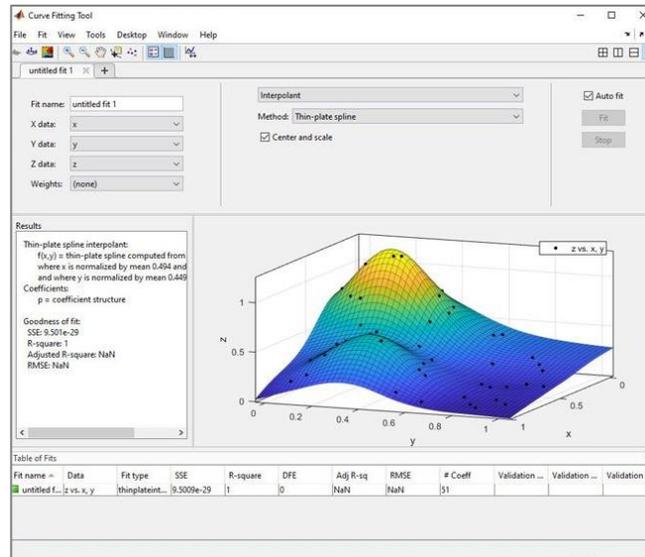


Demo.2: 予測区間を手に入れる



Demo2': 予測区間を手に入れる

- 曲線近似アプリ
 - 曲線、曲面の近似
 - 残差表示
 - 各種統計量も計算
 - MATLAB コード自動生成



曲線近似アプリ

理論編 (1) まとめ

- LiveTask で効率的なデータの前処理
- plot の基本機能で関数近似機能 & コード生成が可能
- fitlm を使ってウィルキンソンの表記で線形回帰モデルを作成
- 信頼区間と予測区間の違いに要注意
- モデルから次のデータ予測する場合は、予測区間を使いましょう

理論編 (2)

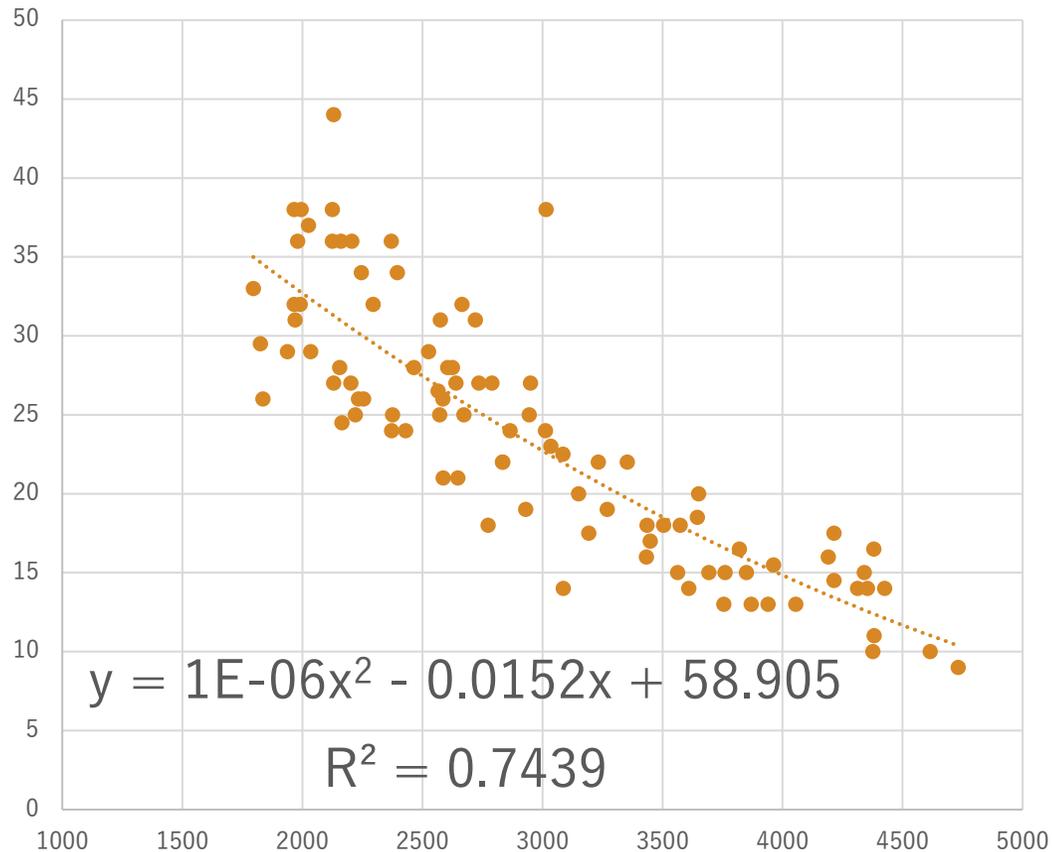
- モデルの選択方法は？
 - 決定係数 と AIC
 - SE, t値, p値
- 脆い線形回帰
 - 多重共線性
 - データの標準化

理論編 (2)

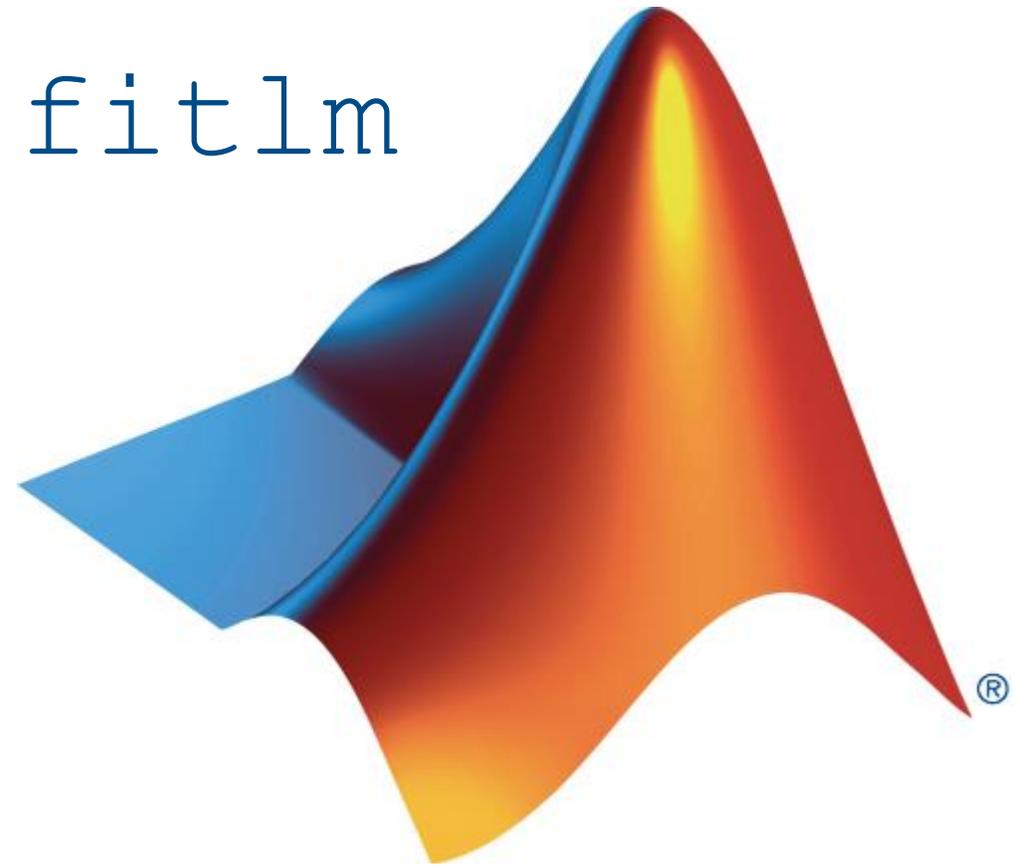
- モデルの選択方法は？
 - 決定係数 と AIC
 - SE, t値, p値
- 脆い線形回帰
 - 多重共線性
 - データの標準化

表計算ソフトではこんなことができるけど

2次関数でのデータ近似



`fitlm`



線形回帰モデルとは (定義再掲)

応答 (目的) 変数が予測子 (説明変数) の線形結合

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i \\ &= [1 \ x_i \ x_i^2] \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \epsilon_i \end{aligned} \quad \Rightarrow \quad \mathbf{y} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_M & x_M^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \boldsymbol{\epsilon}$$

これも線形回帰モデル!

これも線形モデル？

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_m x_i^m + \epsilon_i$$

$$y_i = \beta_0 + \beta_1 x_{i,1}^2 + \beta_2 x_{i,2}^5 + \beta_3 x_{i,1}^2 x_{i,2} + \epsilon_i$$

$$y_i = \beta_0 + \sin \beta_1 x_{i,1} + \log \beta_2 x_{i,1} + \beta_3 x_{i,2} + \epsilon_i$$



Tip: 未知パラメータについての線形結合になっているか確認すればいい

決定係数

モデルの当てはまりの良さの指標

```
>> mdl = fitlm(tbl,"y~1+x")
```

観測数： 93、誤差の自由度： 89

平方根平均二乗誤差： 4.12

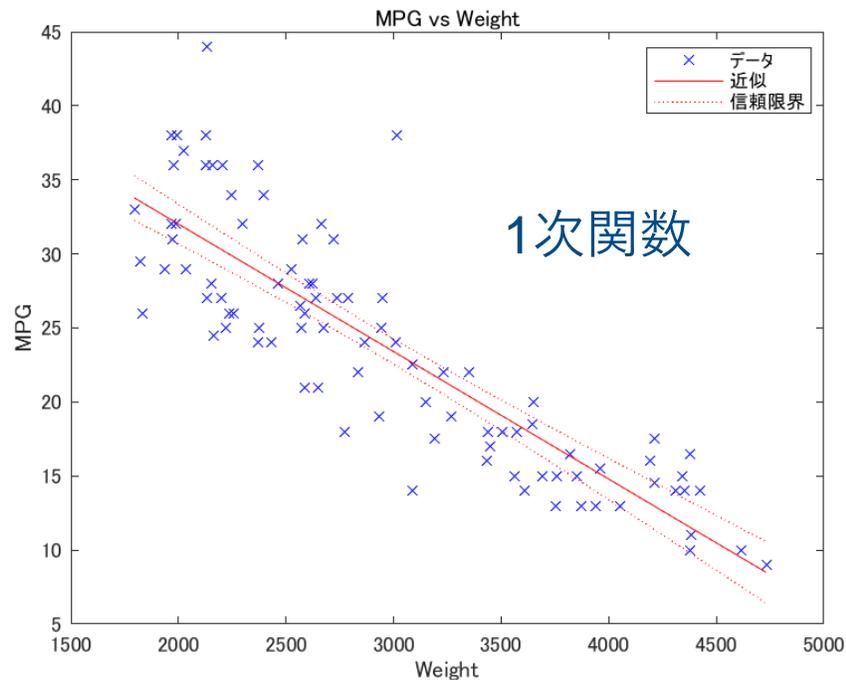
決定係数： 0.748、自由度調整済み決定係数： 0.74

F 統計量 - 定数モデルとの比較： 88.3、p 値は 1.4e-26 です

$$R^2 = 1 - \frac{\text{予測値による変動} \sum (y_i - \hat{y}_i)^2}{\text{総変動} \sum (y_i - \bar{y})^2}$$

1次モデル vs 3次のモデル – 決定係数は高次のモデルが有利?!

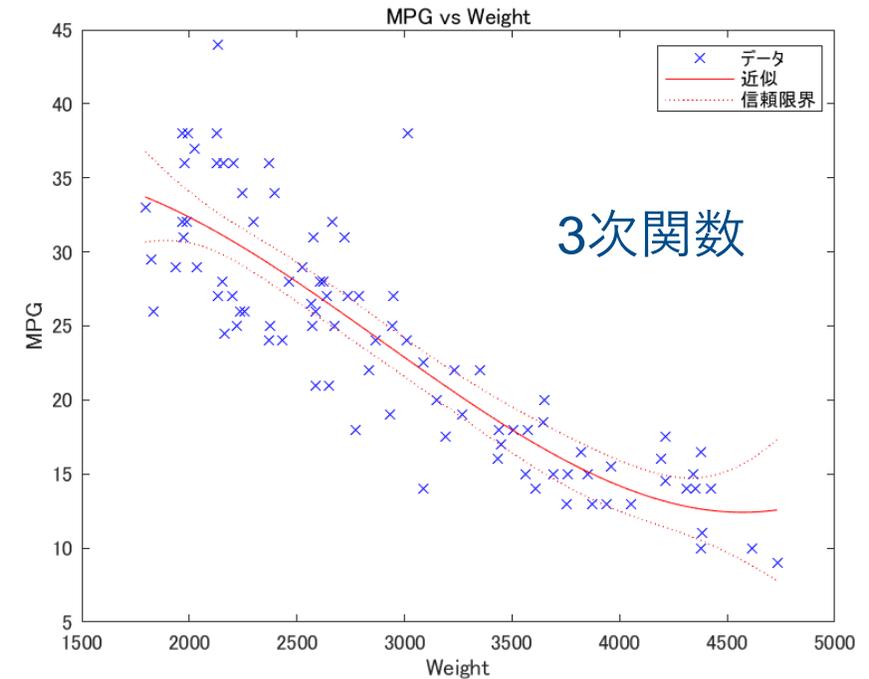
MPG ~ 1 + Weight



- 平方根二乗誤差 (RMSE) = 4.16
- 決定係数 (R^2) = 0.738

MPG ~ 1 + Weight³ *

*MPG ~ 1 + Weight + Weight² + Weight³ と同義



- 平方根二乗誤差 (RMSE) = 4.12
- 決定係数 (R^2) = 0.748

ウィルキンソンの表記法まとめ 2

演算子	意味
+	この項を含める
-	この項を除外する
*	積とすべての低次項を含める
:	積 (相互作用) だけを含める
^	べき乗とすべての低次項を含める ($x^*x^* \dots *x$ と等価)

$$W = \beta_0 + \beta_1 H + \beta_2 H^2$$

$$\Rightarrow W \sim 1 + H + H^2$$

$$\Rightarrow W \sim 1 + H^2$$

どちらも一緒!

[1]: Wilkinson, G. N., and C. E. Rogers. Symbolic description of factorial models for analysis of variance. J. Royal Statistics Society 22, pp. 392–399, 1973.

ゴミを加えても決定係数は大きくなる

MPG~1+Weight

```
linear_md1_1 = fitlm(tbl,"MPG~1+Weight")
```

linear_md1_1 =
線形回帰モデル:
MPG ~ 1 + Weight

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	49.238	1.6504	29.834	9.0258e-49
Weight	-0.0086118	0.00053775	-16.014	3.2405e-28

観測数: 93、誤差の自由度: 91
平方根平均二乗誤差: 4.16
決定係数: 0.738、自由度調整済み決定係数: 0.735
F 統計量と一定のモデルの比較: 256、p 値は 3.24e-28 です

$$R^2 = 0.738$$

MPG~1+Weight+Var1

```
linear_md1_rand = fitlm(tbl,"MPG~1+Weight+Var1")
```

linear_md1_new =
線形回帰モデル:
MPG ~ 1 + Weight + Var1

乱数を加えた

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	50.191	1.7793	28.208	1.8165e-46
Weight	-0.0085679	0.00053595	-15.986	4.9489e-28
Var1	-0.0020317	0.0014615	-1.3902	0.1679

観測数: 93、誤差の自由度: 90
平方根平均二乗誤差: 4.14
決定係数: 0.744、自由度調整済み決定係数: 0.738
F 統計量と一定のモデルの比較: 131、p 値は 2.52e-27 です

$$R^2 = 0.744$$



自由度調整済み決定係数

パラメータ数で調整したモデルの当てはまりの良さ

観測数： 93、誤差の自由度： 89

平方根平均二乗誤差： 4.12

決定係数： 0.748、自由度調整済み決定係数： 0.74

F 統計量 - 定数モデルとの比較： 88.3、p 値は 1.4e-26 です

$$R^2_{adj} = 1 - \frac{\sum (y_i - \hat{y}_i)^2 / (N - M - 1)}{\sum (y_i - \bar{y})^2 / (N - 1)}$$

$\beta \in \mathbb{R}^{M+1}$
 誤差(残差)の自由度

実測値の自由度

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

決定係数と自由度調整済み決定係数の関係

観測数: 93、誤差の自由度: 89

平方根平均二乗誤差: 4.12

決定係数: 0.748、自由度調整済み決定係数: 0.74

F 統計量 - 定数モデルとの比較: 88.3、p 値は 1.4e-26 です

$$R_{adj}^2 = 1 - \frac{(1 - R^2)(\overset{\text{観測数}}{N} - 1)}{N - (M + 1)}$$

誤差(残差)の自由度

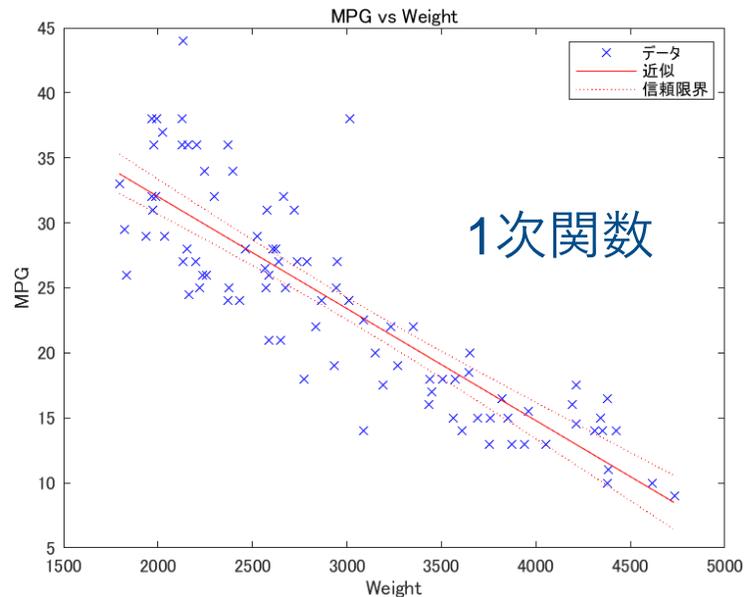
1
2
3
4

```
N = 93;
edf = 89;
R2 = 0.748;
R2adj = 1 - (1-R2)*(N-1)/edf
```

R2adj = 0.7395

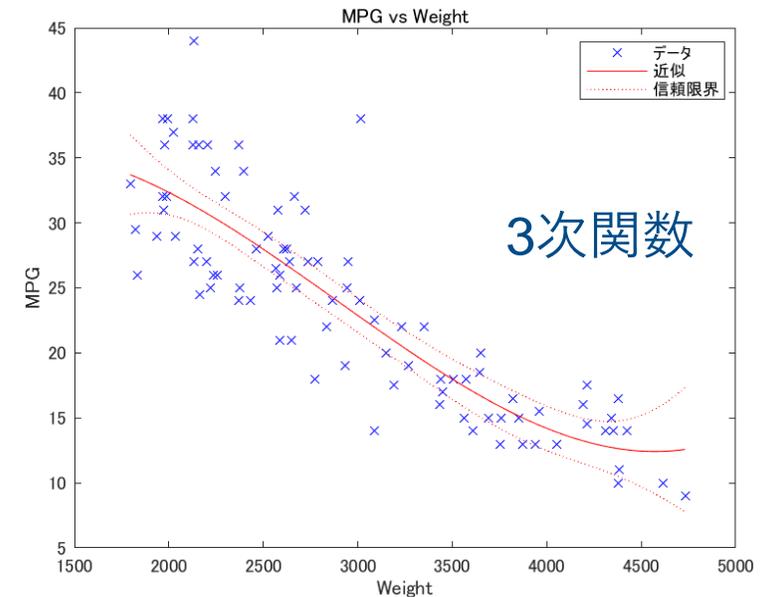
1次モデル vs 3次のモデル – 赤池情報量基準 (AIC)

MPG~1+Weight



- 平方根二乗誤差 (RMSE) = 4.16
- $R^2 = 0.738$
- $R^2_{adj} = 0.735$
- AIC = 530.9183

MPG~1+Weight^3



- 平方根二乗誤差 (RMSE) = 4.12
- $R^2 = \underline{0.748}$
- $R^2_{adj} = \underline{0.74}$
- AIC = 531.1625

モデル選択に一般的に用いられる指標: 情報量基準

- 赤池情報量基準 (AIC): モデルの複雑さと、データのモデル適合度のバランスを取る統計量。データが生成されたモデルと、推定したモデルの擬距離を指標にしているため、データの当てはまりよりも将来予測が上手く行くかどうか注目した基準

$$AIC = -\ln L(\theta) + 2k$$

小さい方が良い!

対数尤度: モデル適合度の指標

自由パラメータ数
(今回は $M+1$)

- (注意) ネストしているモデル間の比較に用いることができる

$$(1) y = \beta_0 + \beta_1 x_1$$

$$(2) y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

ネストの関係

(2) が (1) を含む関係

モデルの選択基準を指定して自動モデル選択

stepwiselm

- `stepwiselm(tbl, "y~x")`
 - 設定した指標を用いて最適モデルを探索
 - SSE, AIC, BIC, Rsquared, AdjRsquared

```

Displacement を追加する AIC 内の変化は 1.8324 です
Weight:Horsepower を追加する AIC 内の変化は -1.8792 です
5. Weight:Horsepower, AIC = 455.2812 を追加中
Acceleration を追加する AIC 内の変化は -5.6575 です
Cylinders を追加する AIC 内の変化は 1.8987 です
Displacement を追加する AIC 内の変化は 1.168 です
6. Acceleration, AIC = 449.6237 を追加中
Cylinders を追加する AIC 内の変化は 1.9116 です
Displacement を追加する AIC 内の変化は 1.9216 です
Weight:Acceleration を追加する AIC 内の変化は 1.2188 です
Acceleration:Horsepower を追加する AIC 内の変化は 1.9591 です
Acceleration:Model_Year を追加する AIC 内の変化は 0.65128 です
Weight:Horsepower を削除する AIC 内の変化は 6.1499 です
Weight:Model_Year を削除する AIC 内の変化は 0.46372 です
Horsepower:Model_Year を削除する AIC 内の変化は 10.5038 です

stepwise_md1 =
線形回帰モデル
MPG ~ 1 + Acceleration + Weight*Horsepower + Weight*Model_Year + Horsepower*Model_Year

推定された係数:

```

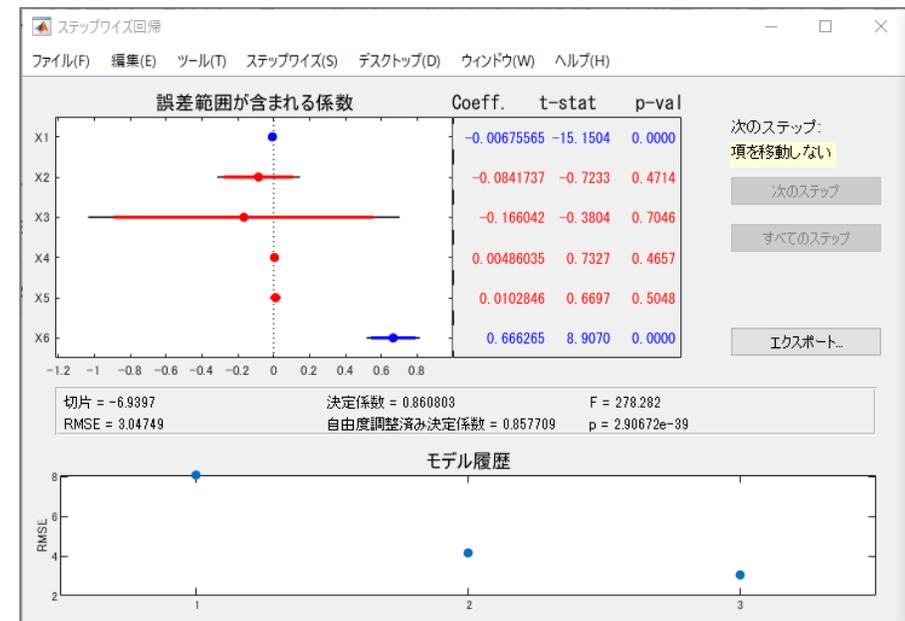
	Estimate	SE	tStat	pValue
(Intercept)	-27.488	30.602	-0.89824	0.37159
Weight	-0.034617	0.018355	-1.8859	0.06272
Acceleration	-0.37453	0.13867	-2.7009	0.0083444
Horsepower	0.97913	0.31805	3.0786	0.0028004
Model_Year	1.1791	0.36573	3.2239	0.0017948
Weight:Horsepower	3.1958e-05	1.1454e-05	2.7902	0.0065025
Weight:Model_Year	0.00036122	0.00023912	1.5106	0.1346
Horsepower:Model_Year	-0.015797	0.0045166	-3.4974	0.00074964

```

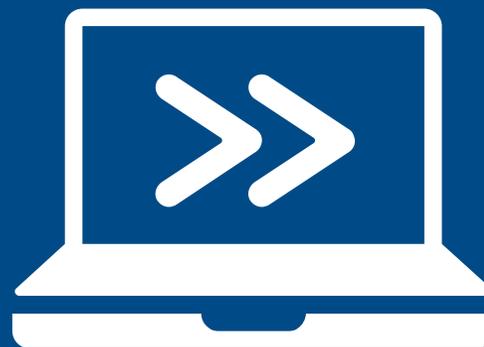
観測数: 93、誤差の自由度: 85
平方根平均二乗誤差: 2.6
決定係数: 0.904、自由度調整済み決定係数: 0.896
F 統計量 - 定数モデルとの比較: 114、p 値は 1.74e-40 です

```

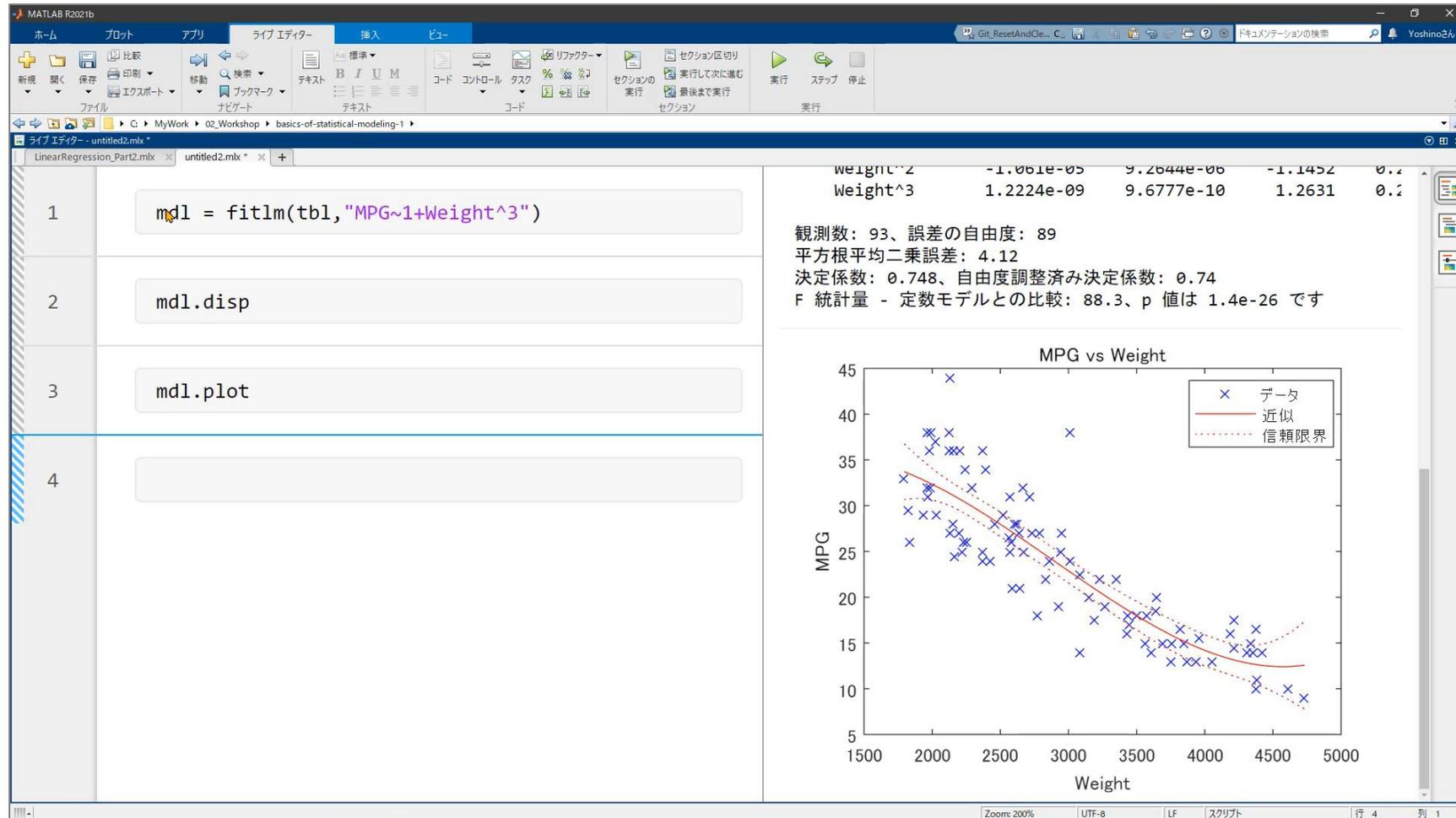
- `stepwise(X, y)`
 - GUI 機能
 - SSE (F 統計量) が基準
 - コードの自動生成対応



デモンストレーション



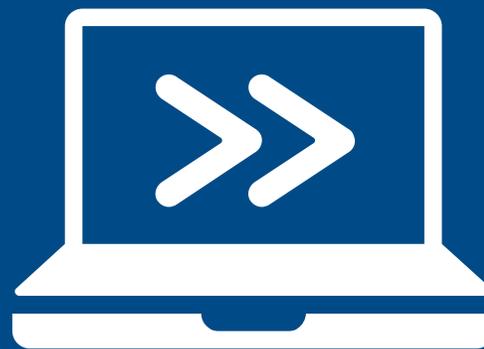
Demo.3: 情報量基準、対数尤度などの確認方法



理論編 (2)

- モデルの選択方法は？
 - 決定係数 と AIC
 - SE, t値, p値
- 脆い線形回帰
 - 多重共線性
 - データの標準化

デモンストレーション



Demo.4: 重回帰分析

```

34 mcol_horp = fitlm(tbl, "MPG~1 + Weight + Model_Year + Horsepower");
35 disp(mcol_horp);

```

線形回帰モデル:
 $MPG \sim 1 + \text{Weight} + \text{Horsepower} + \text{Model_Year}$

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	39.863	1.5289	26.072	3.6414e-43
Weight	-0.0057673	0.00085191	-6.7698	1.384e-09
Horsepower	-0.018408	0.016641	-1.1062	0.27166
Model_Year_76	1.3281	0.92714	1.4325	0.15554
Model_Year_82	7.6755	0.94371	8.1333	2.4757e-12

観測数: 93、誤差の自由度: 88
 平方根平均二乗誤差: 2.88
 決定係数: 0.878、自由度調整済み決定係数: 0.873
 F 統計量 - 定数モデルとの比較: 159、p 値は 2.19e-39 です

A:

どちらの方が良いモデル? @重回帰分析

MPG~1+Weight+Model_Year

Horsepower を含まないモデル

線形回帰モデル:

MPG ~ 1 + Weight + Model_Year

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	39.897	1.5305	26.068	1.9419e-43
Weight	-0.0065843	0.00042509	-15.489	5.3727e-27
Model_Year_76	1.9477	0.73977	2.6329	0.0099823
Model_Year_82	8.1301	0.85057	9.5585	2.6198e-15

MPG~1+Weight+Model_Year+Horsepower

Horsepower を含むモデル

線形回帰モデル:

MPG ~ 1 + Weight + Horsepower + Model_Year

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	39.863	1.5289	26.072	3.6414e-43
Weight	-0.0057673	0.00085191	-6.7698	1.384e-09
Horsepower	-0.018408	0.016641	-1.1062	0.27166
Model_Year_76	1.3281	0.92714	1.4325	0.15554
Model_Year_82	7.6755	0.94371	8.1333	2.4757e-12

SE, tStat, pValue 結果を正しく解釈せよ

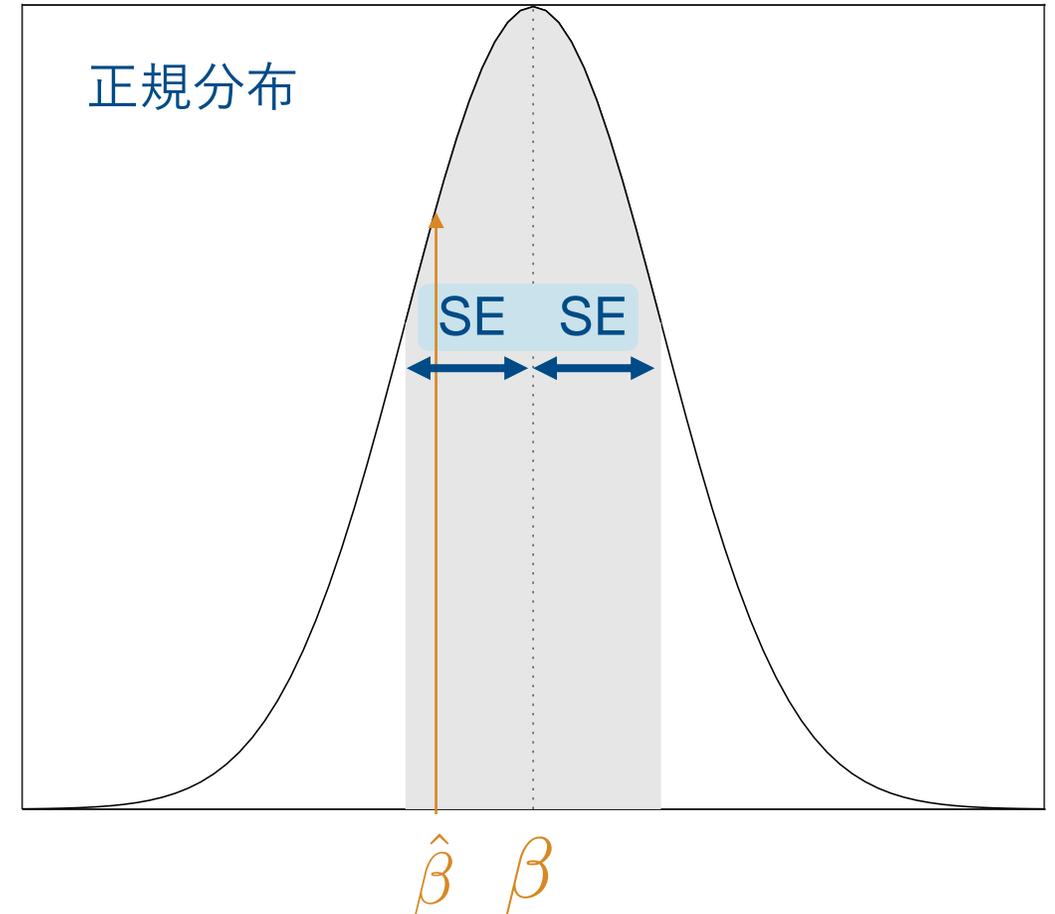
各種統計量の見方 – SE (Standard Error, 標準誤差)

$$\hat{\beta} \sim N(\hat{\beta} | \beta, \sigma^2 (X^T X)^{-1})$$

SE: 共分散行列の各要素の平方根

推定された係数:

	$\hat{\beta}$ Estimate	SE SE
(Intercept)	39.897	1.5305
Weight	-0.0065843	0.00042509
Model_Year_76	1.9477	0.73977
Model_Year_82	8.1301	0.85057



SE が小さければ、推定パラメータが真値に近いかな? と考える

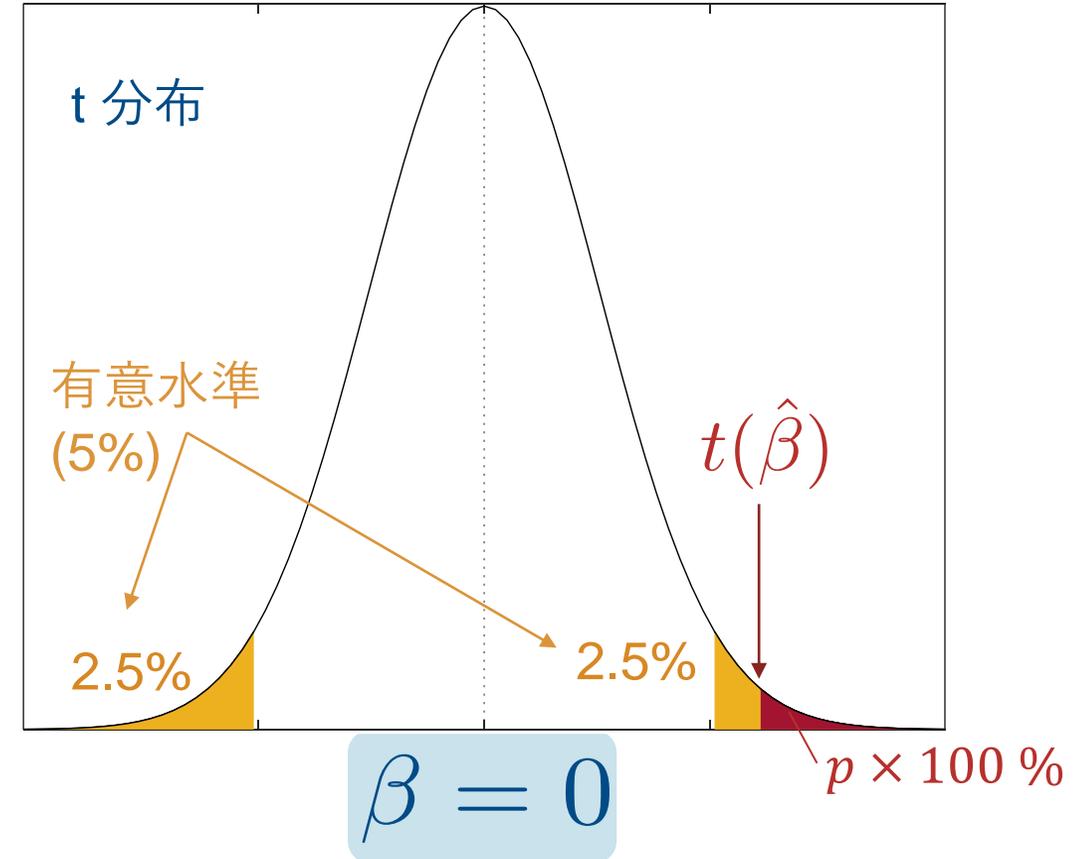
各種統計量の見方 – tStat, pValue (t値, p値)

$$\hat{\beta} \sim N(\hat{\beta} | \beta, \sigma^2 (X^T X)^{-1})$$

- パラメータをゼロとして検定 (帰無仮説)
- 分散をその不変推定量で代用するため、現実的には t 分布を使って検定する (t 検定)

推定された係数:

	$t(\hat{\beta})$	tStat	p	pValue
(Intercept)		26.068		1.9419e-43
Weight		-15.489		5.3727e-27
Model_Year_76		2.6329		0.0099823
Model_Year_82		9.5585		2.6198e-15



有意水準より pValue が小さい (tStat の絶対値が大きい) 場合は、パラメータが有意である

よく分からなかったけど結果だけ持っていきたい方に

※厳密性に欠けます

1. SE:

✓ 真の β に近いかも度, モデルの自信あり度

2. t値:

✓ 対応する β が意味あるか(ゼロじゃない) 度

3. p値:

✓ 対応する β が意味あるか (ゼロじゃない) 度 ← おすすめ

理論編 (2)

- モデルの選択方法は？
 - 決定係数 と AIC
 - SE, t値, p値
- 脆い線形回帰
 - 多重共線性
 - データの標準化

脆い線形回帰モデル

モデルが悪化?!

MPG~1+Weight+Model_Year

Horsepower を含まないモデル ✓ AIC = 464.8753

線形回帰モデル:

MPG ~ 1 + Weight + Model_Year

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	39.897	1.5305	26.068	1.9419e-43
Weight	-0.0065843	0.00042509	-15.489	5.3727e-27
Model_Year_76	1.9477	0.73977	2.6329	0.0099823
Model_Year_82	8.1301	0.85057	9.5585	2.6198e-15

MPG~1+Weight+Model_Year+Horsepower

Horsepower を含むモデル AIC = 465.9510

線形回帰モデル:

MPG ~ 1 + Weight + Horsepower + Model_Year

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	39.863	1.5289	26.072	3.6414e-43
Weight	-0.0057673	0.00085191	-6.7698	1.384e-09
Horsepower	-0.018408	0.016641	-1.1062	0.27166
Model_Year_76	1.3281	0.92714	1.4325	0.15554
Model_Year_82	7.6755	0.94371	8.1333	2.4757e-12

全体的に“悪化”した根本的理由は？

脆い線形回帰モデル (Cont.)

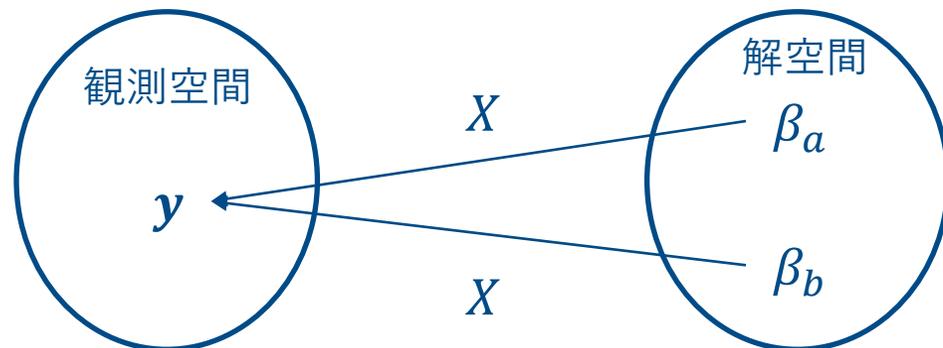
多重共線性

- 一方の定数倍
- 相関が高い
- **多重共線性**があるという

この2つが”すごく似ている”とする

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,i} & \cdots & x_{1,j} & \cdots & x_{1,M} \\ 1 & x_{2,1} & & x_{2,i} & & x_{2,j} & & \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \\ 1 & x_{N,1} & & x_{N,i} & & x_{N,j} & \cdots & x_{N,M} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_M \end{bmatrix}$$

β_a, β_b に決めかねる



不安定な解

求める解が

- 一意に決まらない
- フラフラする
- ノイズに負ける

脆い線形回帰モデル (Cont.)

多重共線性

$$\begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 3 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

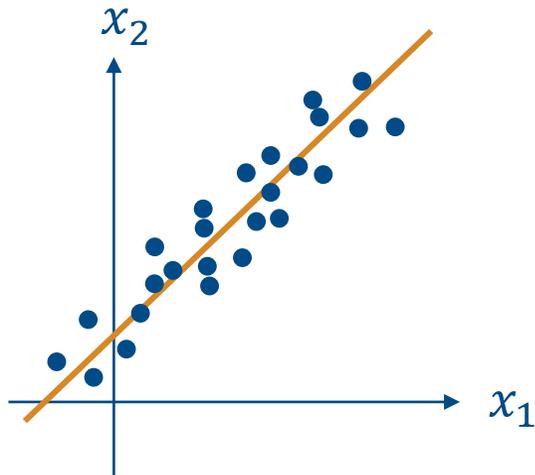


例えば

$$\hat{\beta} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

不安定な解

特徴量間での高い相関



多重共線性を持つ
(マルチコ)

X

Multi-collinearity



- モデル推定が不安定になる
- SE (標準誤差) 等に数値として発現
- 増幅されたノイズがモデルに影響

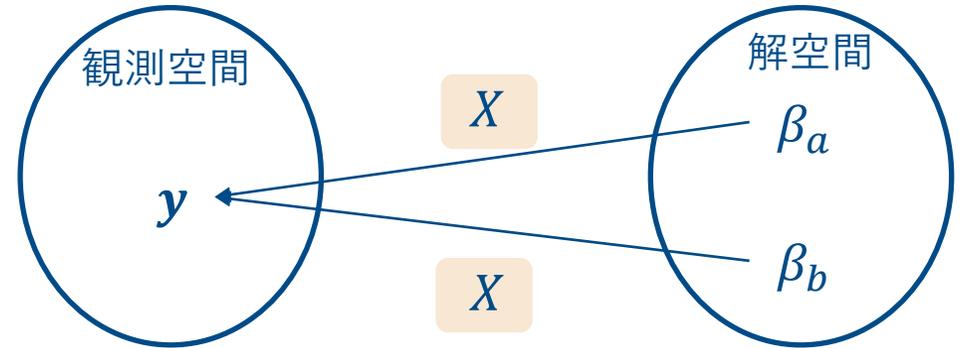
脆い線形回帰モデル (基礎習得者向け)

増幅するノイズ

特異値分解 (SVD) を実行

$$X_{N \times (M+1)} = U \begin{bmatrix} \gamma_0 & 0 & \cdots & 0 \\ 0 & \gamma_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_M \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} V^T$$

where, U consists of the orthonormal eigenvectors of XX^T ,
and V consists of the orthonormal eigenvectors of $X^T X$.



• $\gamma \approx 0$ が必要

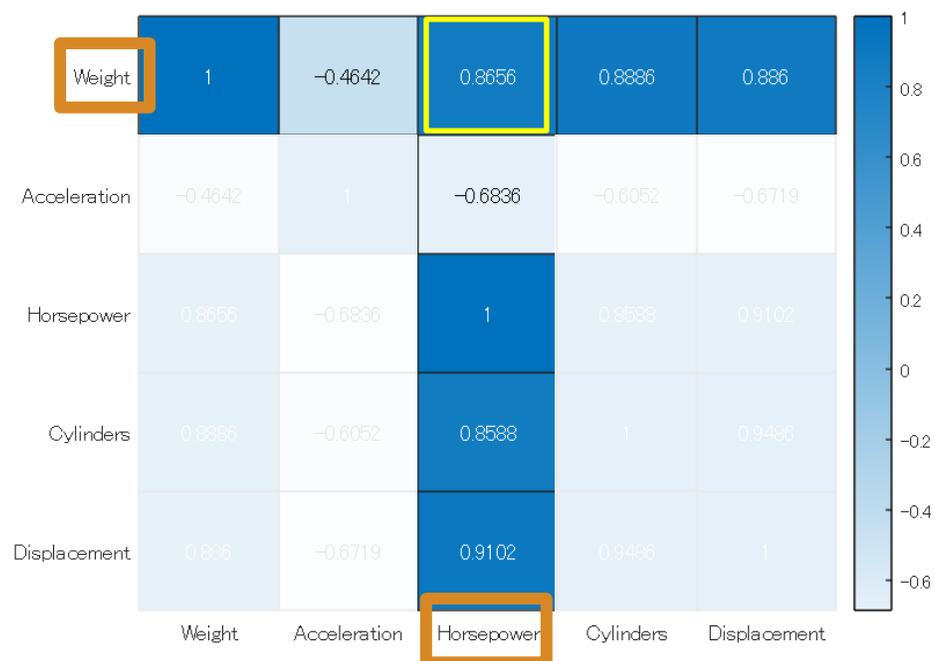
$$\begin{aligned} \hat{\beta} &= (X^T X)^{-1} X^T y \\ &= \beta + \sum_{i=0}^M \frac{1}{\gamma_i} \mathbf{v}_i \mathbf{u}_i^T \epsilon \end{aligned}$$

推定値に増幅したノイズがのってくる
(真の値から大幅にズレる可能性がある)

脆い線形回帰モデル (Cont.)

説明変数間の相関係数を確認

```
corrNames = ["Weight", "Acceleration", "Horsepower", "Cylinders", "Displacement"];
corrTbl = corrcoef(table2array(tbl(:,corrNames)));
heatmap(corrTbl, "XData", corrNames, "YData", corrNames);
```



$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,i} & \cdots & x_{1,j} & \cdots & x_{1,M} \\ 1 & x_{2,1} & & x_{2,i} & & x_{2,j} & & x_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & & x_{N,i} & & x_{N,j} & \cdots & x_{N,M} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_M \end{bmatrix}$$

Weight
Horsepower

↓
↓

- **Weight vs Horsepower**
 - 相関係数 = **0.87**
- `corrcoef(array)`
 - 相関係数を計算
- `Heatmap()`
 - ヒートマップを作製
- `corrplot()`
 - 相関係数の計算 + ヒートマップ作製

脆い線形回帰モデル (Cont.)

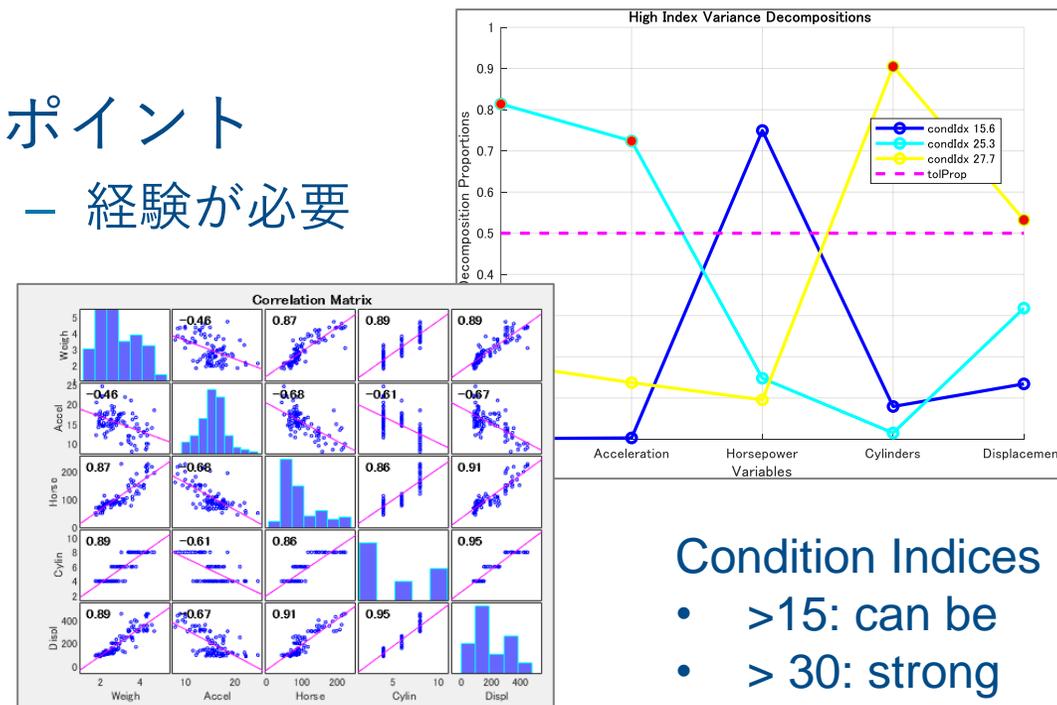
多重共線性対策案

- 対策1: 正則化を取り入れたモデル
 - lasso (Elastic Net)
 - ridge
 - ロバストなモデルを構築できる
 - 所謂、過学習対策
- 注意
 - 正則化パラメータを決める必要あり
 - 不偏推定量は得られない

$$\min_{\beta, \lambda} J(\beta, \lambda) = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_p$$

- 対策2: 変数間相関を予め除去
 - corrplot 関数: 相関係数プロット
 - Collintest 関数:
Belsley Collinearity diagnostics

- ポイント
 - 経験が必要



Condition Indices

- >15: can be
- > 30: strong

理論編 (2)

- モデルの選択方法は？
 - 決定係数 と AIC
 - SE, t値, p値
- 脆い線形回帰
 - 多重共線性
 - データの標準化

データの標準化

MPG~1+Weight+Cylinders
[kg]

	Estimate	SE	tStat	pValue
(Intercept)	48.912	1.5801	30.954	9.2075e-50
Weight	-0.0055097	0.0011201	-4.9187	3.9038e-06
Cylinders	-1.6037	0.51458	-3.1165	0.0024566

MPG~1+Weight+Cylinders
[ton]

	Estimate	SE	tStat	pValue
(Intercept)	48.912	1.5801	30.954	9.2075e-50
Weight	-5.5097	1.1201	-4.9187	3.9038e-06
Cylinders	-1.6037	0.51458	-3.1165	0.0024566

- 数値のスケールが異なると、SE や推定値にも影響が出る
 - 数値を事前に標準化しておくのがお作法
 - zscore, normalize 関数
 - LiveTask から “データの正規化”

$$x'_i = \frac{x_i - \bar{x}}{\sigma_x}$$

```

54 % データの正規化
55 [newTable,centerValue,scaleValue] = normalize(tbl,"norm",...
56 "DataVariables",["Weight","Acceleration","mpg","horsepower","cylinders",...
57 "Displacement"]);
58
59 % 結果の表示
60 clf
61 tiledlayout(2,1);
62 nexttile
63 plot(tbl.Acceleration,"Color",[77 190 238]/255,"Displayname","入力データ")
64 legend
65 ylabel("Acceleration")
    
```

理論編 (2) まとめ

- モデルを評価するための代表的な指標を紹介
 - 決定係数, 自由度調整済み決定係数
 - 情報量基準, e.g., AIC, BIC ...
 - SE, p値, t値 ...
- 説明変数の多重共線性を排除してからモデルを作成しよう
- データの標準化はまず行うべきお作法
 - LiveTask がお勧め!

実践編

- 理論編の総復習
- 交差相関にご用心

実践編

- 理論編の総復習
- 交差相関にご用心

住宅販売数の予測モデルを作成

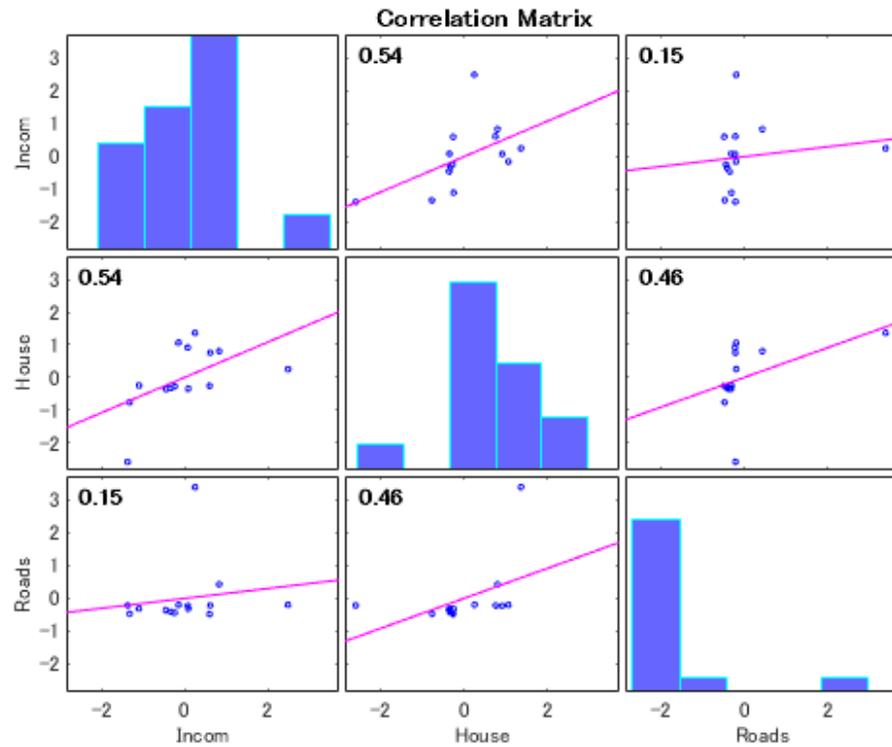
時系列

応答	説明変数 (予測子)			
1 Sales	2 Income	3 Households	4 Roads	
0.8606	0.0788	0.9184	-0.2227	
0.9006	0.2536	1.3684	3.3858	
0.7745	0.8354	0.8049	0.4317	
-2.2062	0.6187	0.7566	-0.2146	
1.8292	-1.3739	-2.5923	-0.2100	
-1.0815	2.4944	0.2531	-0.1950	
-0.2747	-1.1013	-0.2442	-0.3117	
0.2654	0.6030	-0.2602	-0.4792	
0.2819	-0.2372	-0.2758	-0.4376	
-0.9629	-0.1482	1.0667	-0.1964	
-0.5302	-1.3278	-0.7633	-0.4702	
0.1575	-0.3342	-0.3301	-0.4136	
0.0526	0.0880	-0.3442	-0.3136	
0.0668	-0.4495	-0.3580	-0.3529	

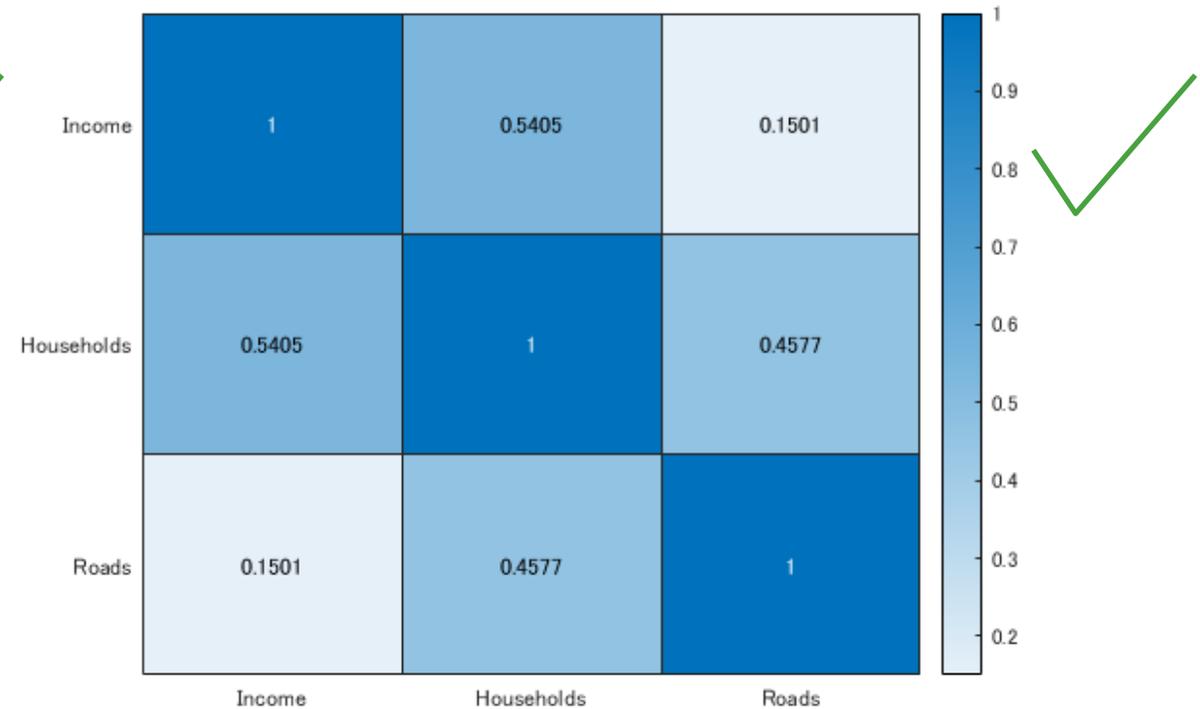
- Sales (販売数) を3つの予測子から推論
 - Income: 所得
 - Households: 世帯数
 - Roads: 道路数
- 各数値は標準化済み ✓



説明変数間の相関性 (多重共線性) のチェック



corrplot



corrcoef + heatmap

モデル作成 & 指標確認

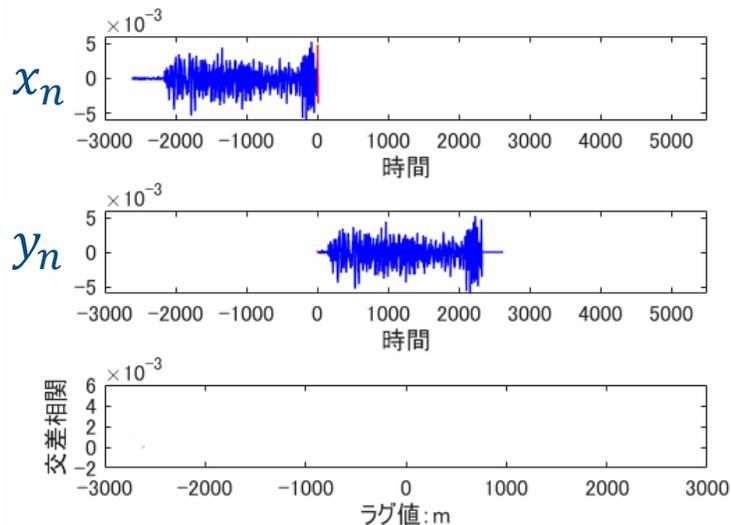
```
mdl_carSales = fitlm(carSales, "Sales ~ 1 + Income + Households + Roads");  
disp(mdl_carSales);|
```

実践編

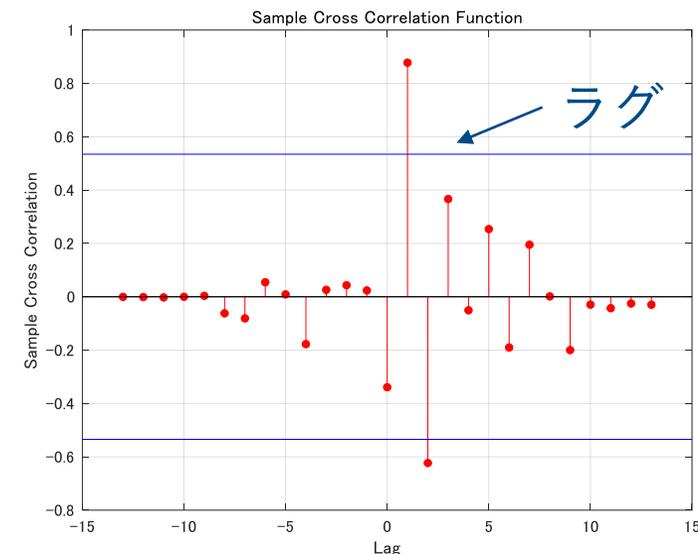
- 理論編の総復習
- 交差相関にご用心

交差相関とは

時間軸に沿った信号間の相関を発見



$$C_{xy}(m) = \begin{cases} \sum_n x_{n+m} \cdot y_n & (m \geq 0) \\ \sum_n y_{n-m} \cdot x_n & (m < 0) \end{cases}$$



- 時間軸に沿った信号間のズレ(相関)を発見
 - xcorr: 交差相関を計算
 - crosscorr: 交差相関を計算 & その検定を実行
- データをラグ分シフトして解析を続行

ラグの分だけデータをshift

	A	B	C	D		A	B	C	D
1	Sales	Income	Households	Roads	1	Sales	Income	Households	Roads
2	0.860560798	0.078759524	0.1382879	-0.222715356	2	0.860560798	0.253592925	1.368388319	3.385845988
3	0.900638973	0.253592925	1.36838319	3.385845988	3	0.900638973	0.835405619	0.804945445	0.431673945
4	0.774502444	0.835405619	0.804945445	0.431673945	4	0.774502444	0.618719709	0.756644208	-0.214608857
5	-2.206173515	0.618719709	0.756644208	-0.214608857	5	-2.206173515	-1.373893071	-2.592343508	-0.209960624
6	1.829182823	-1.373893071	-2.592343508	-0.209960624	6	1.829182823	2.494431487	0.253100993	-0.195005029
7	-1.081502618	2.494431487	0.253100993	-0.195005029	7	-1.081502618	-1.101263613	-0.244224046	-0.311700038
8	-0.27467244	-1.101263613	-0.244224046	-0.311700038	8	-0.27467244	0.603016929	-0.260184652	-0.479246334
9	0.265392656	0.603016929	-0.260184652	-0.479246334	9	0.265392656	-0.237177312	-0.275786593	-0.437570505
10	0.28186344	-0.237177312	-0.275786593	-0.437570505	10	0.28186344	-0.14816907	1.066673748	-0.196391862
11	-0.962916198	-0.14816907	1.066673748	-0.196391862	11	-0.962916198	-1.327765981	-0.763290719	-0.47023091
12	-0.530185878	-1.327765981	-0.763290719	-0.47023091	12	-0.530185878	-0.334171109	-0.330112776	-0.41363013
13	-0.157541592	-0.334171109	-0.330112776	-0.41363013	13	-0.157541592	0.087969938	-0.344190832	-0.313586952
		0.087969938		0.313586952			-0.449455		-0.3528732

データのシフト前後のモデルを比較

シフト無し

```
mdl_carSales = fitlm(carSales,...
    "Sales ~ 1 + Income + Households + Roads")
```

```
mdl_carSales =
線形回帰モデル:
Sales ~ 1 + Income + Households + Roads
```

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	-1.2474e-17	0.2348	-5.3127e-17	1
Income	-0.12104	0.2921	-0.41437	0.68735
Households	-0.55874	0.32481	-1.7202	0.11613
Roads	0.55914	0.27641	2.0229	0.070641

観測数: 14、誤差の自由度: 10
 平方根平均二乗誤差: 0.879
 決定係数: 0.406、自由度調整済み決定係数: 0.228
 F 統計量と一定のモデルの比較: 2.28、p 値は 0.142 です

- $R^2 = 0.406$, SE 大

シフト有り

```
mdl_carSales_Shift = fitlm(carSales_Shift,...
    "Sales ~ 1 + Income + Households + Roads")
```

```
mdl_carSales_Shift =
線形回帰モデル:
Sales ~ 1 + Income + Households + Roads
```

推定された係数:

	Estimate	SE	tStat	pValue
(Intercept)	0.042522	0.09708	0.43801	0.6717
Income	0.62424	0.11766	5.3056	0.00049024
Households	0.4753	0.13857	3.43	0.0075085
Roads	-0.0015437	0.11277	-0.013689	0.98938

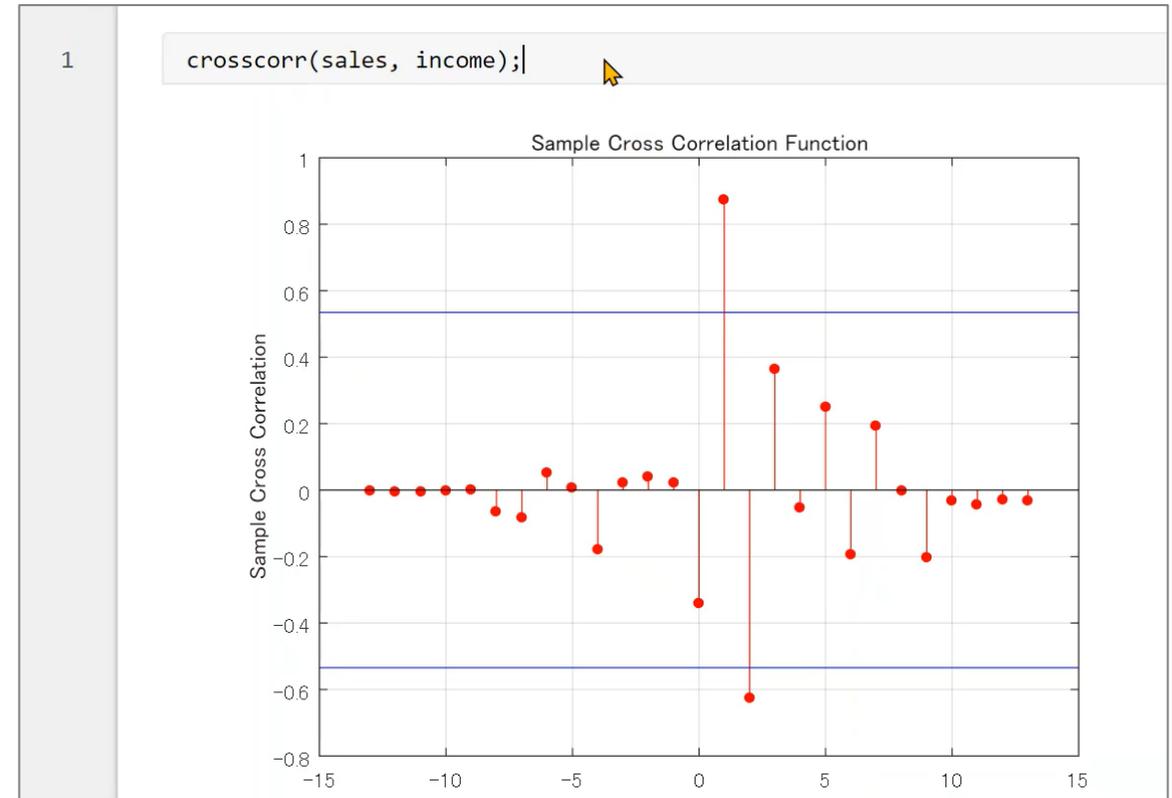
観測数: 13、誤差の自由度: 9
 平方根平均二乗誤差: 0.348
 決定係数: 0.916、自由度調整済み決定係数: 0.888
 F 統計量と一定のモデルの比較: 32.8、p 値は 3.57e-05 です

不要!?

- $R^2 = 0.916$, SE 小
- Roads の p値 98% >> 不要

実践編 まとめ

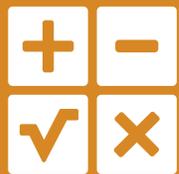
- 線形回帰モデルは
時系列モデリングではない
 - ラグがあると情報を拾うことができない
 - 交差相関を活用し、データ間のラグを事前に除去してモデルを作成する
- 交差相関
 - `crosscorr` 関数を実行すると、相関の検定も実行してくれる



線形回帰分析のまとめ

正規化/標準化

- zscore
- normalize
- LiveTask 活用



データのチェック

- 多重共線性
 - corrcoef
 - collintest
 - corrplot
- 相互相関
 - xcorr
 - crosscorr

モデル作成

- ウィルキンソンの表記法
- stepwise
- 正則化

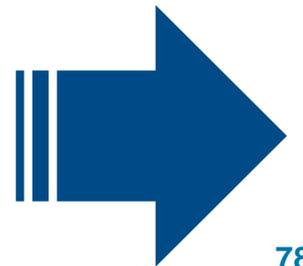


モデル選択

- t値, p値
- 決定係数
- 情報量基準, AIC
- 標準誤差 (SE)



モデルに命を吹き込んで、ビジネスに貢献する



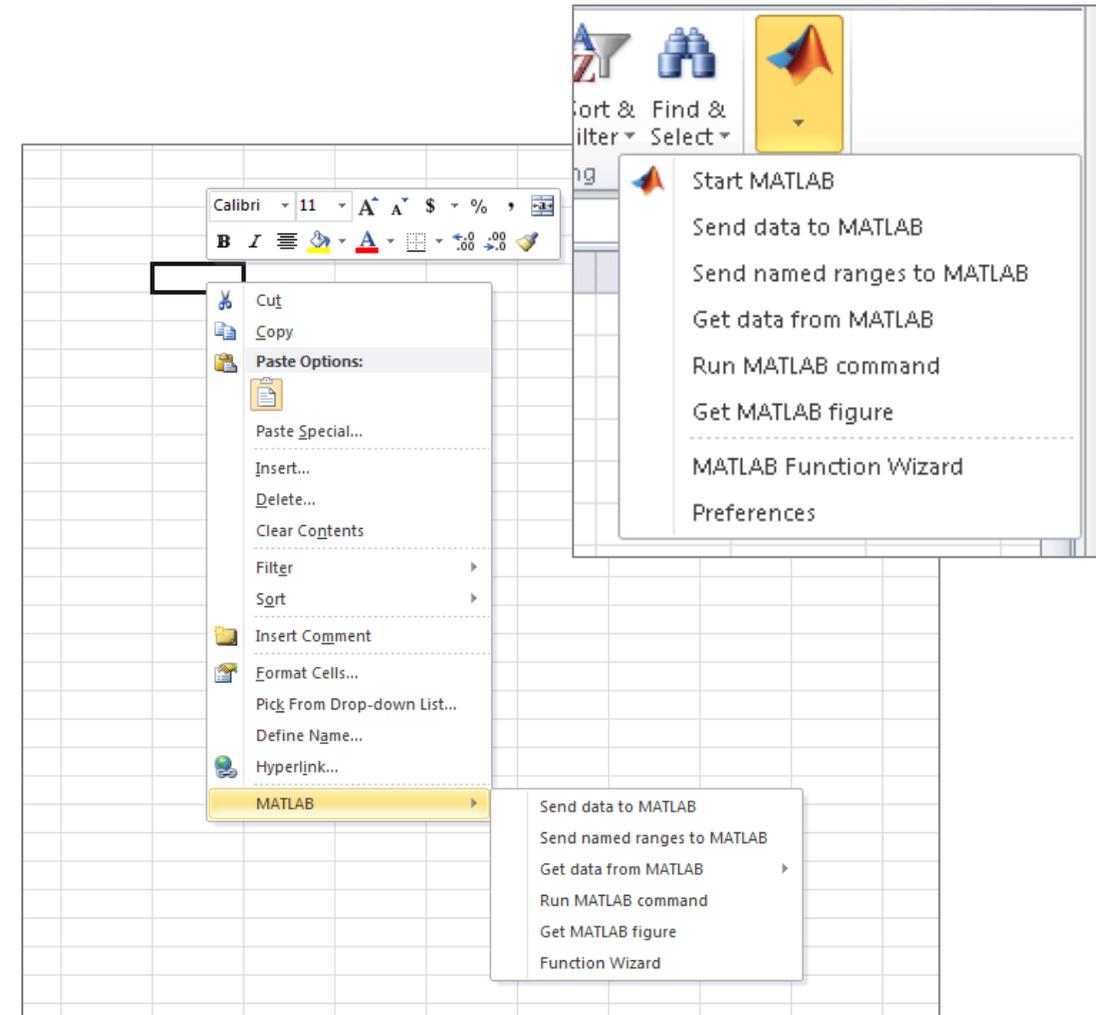
Excel 連携

Excel での MATLAB の使用 - [Spereadsheet Link™](#)

- Excel アドイン
 - MATLAB グループメニューから MATLAB 機能にアクセス

- MATLAB 関数の利用
 - Excel から直接 or VBA マクロをを介して MATLAB の組み込み関数や、カスタム関数を実行可能
 - E.g., =MLPutMatrix("A",C10)

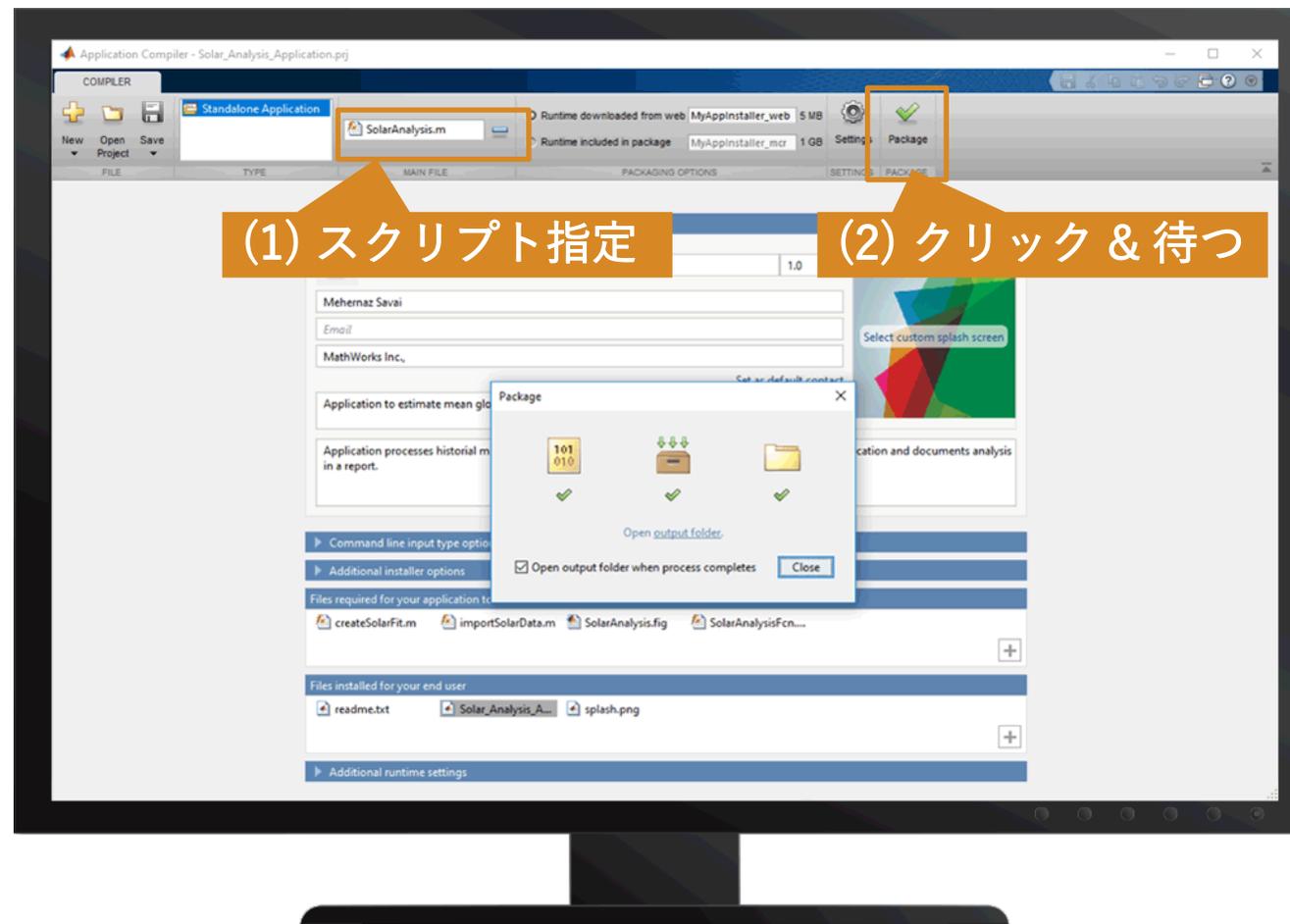
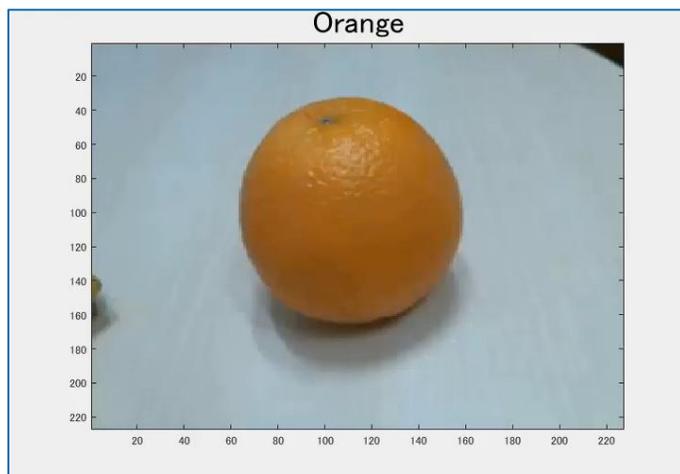
- Excel と MATLAB 間でデータ交換
 - MATLAB グループメニュー、VBA マクロ、Spreadsheet Link ツールバーを使用して、双方向にデータを転送



2クリックで *.exe作成 & 配布

- MATLABプログラムからの
 - スタンドオンアプリ作成
 - Webアプリ作成
 - Excelアドイン作成
- 暗号化されたコード
 - 配布は無料

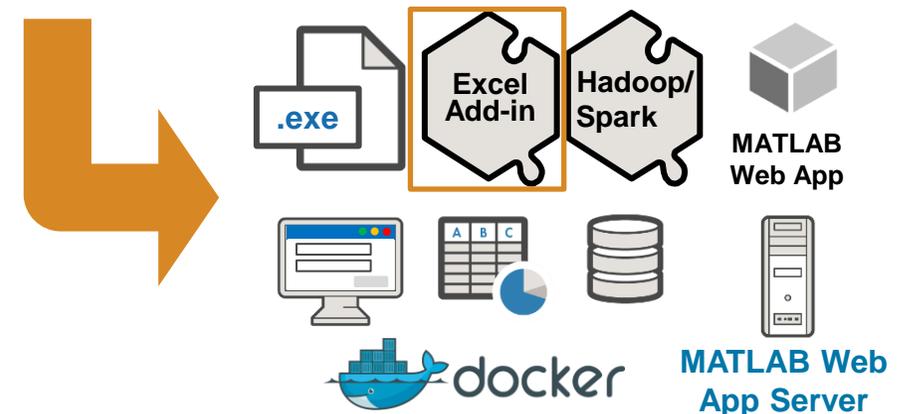
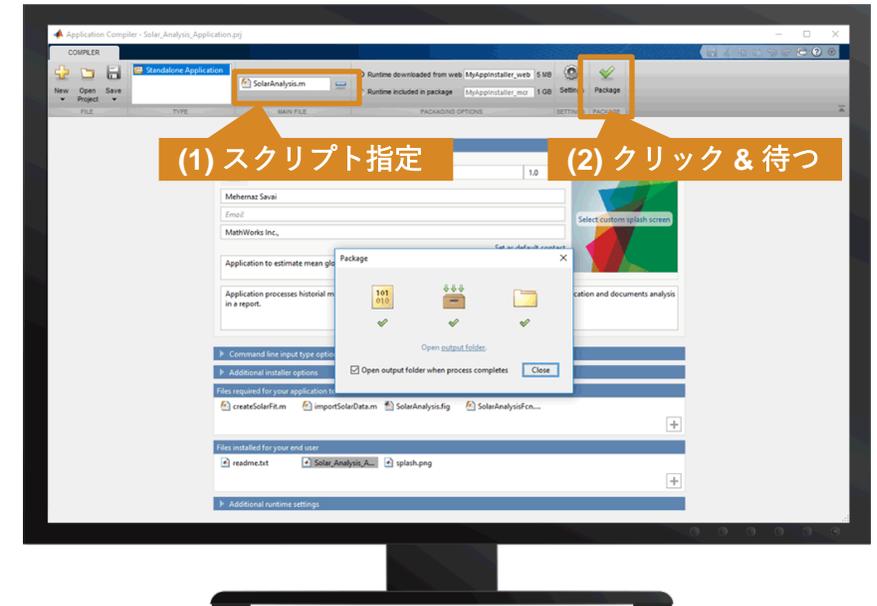
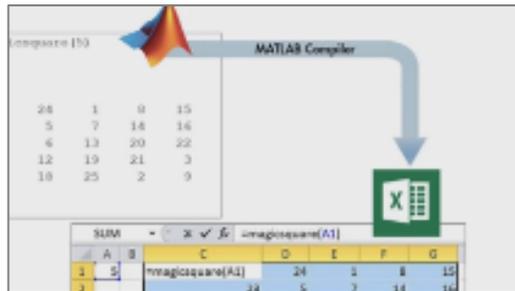
“10行でできるディープラーニング”



Excel ユーザーにも高度な計算を提供

Excel Add-in として MATLAB コードをパッケージ化

- Runtime を用いた幅広い展開性
 - スタンドアロン / Web アプリ化
 - MapReduce 及び Spark ビッグデータアプリケーション化
 - **Microsoft® Excel® アドインのパッケージ化**
 - Docker イメージのパッケージ化
- Microsoft Excel アドイン
 - Excel 用のカスタム関数を MATLAB プログラムから作成
 - 既存の Excel 関数と同じようにカスタム関数にアクセス

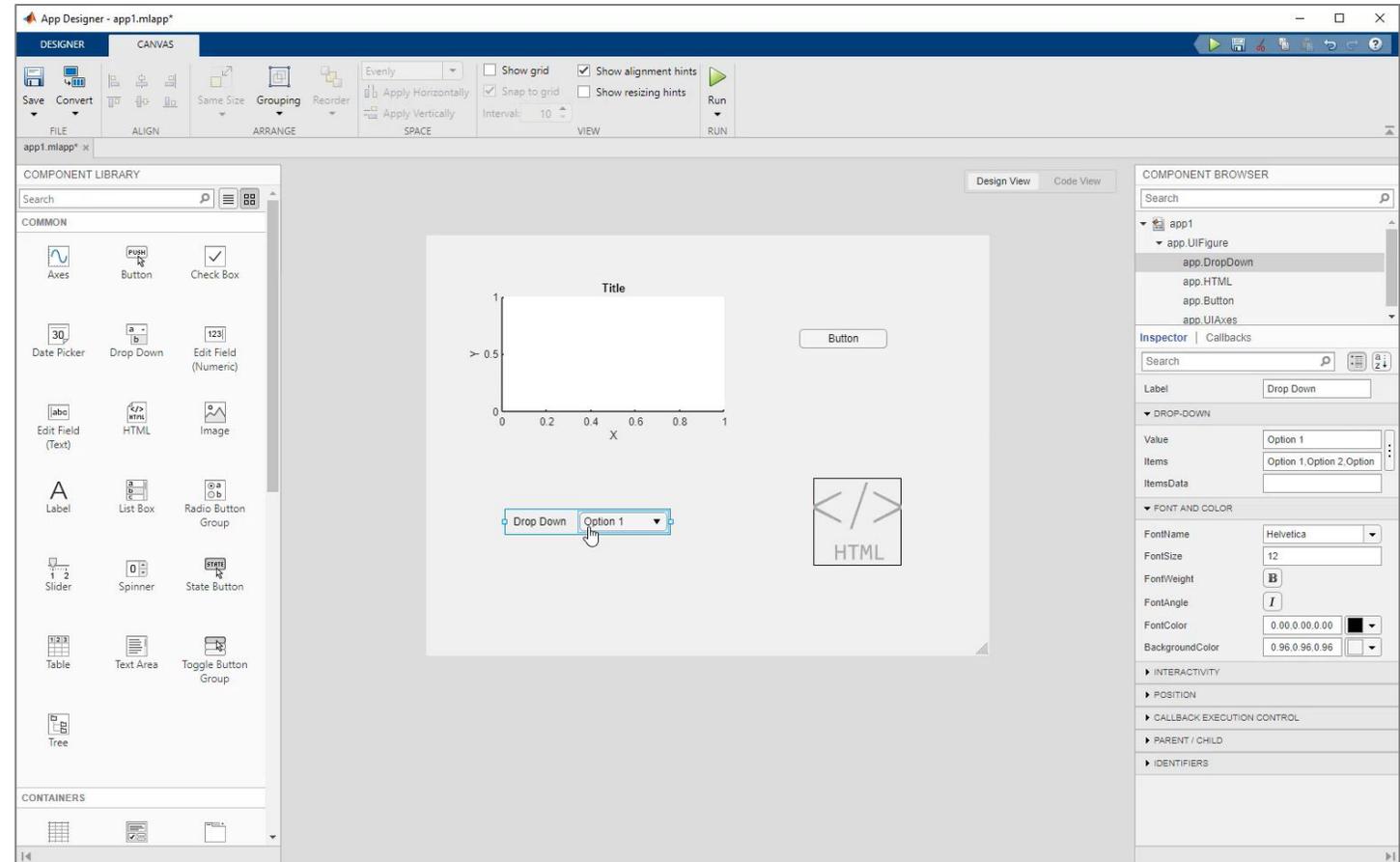


エンジニア・研究者のための GUI 作成アプリ

- 外部へのアプリ配布の方法
 - スタンドアロンアプリ
 - Webアプリ

共有	実行	
MATLAB アプリ MATLAB ユーザーとアプリを共有するためのアプリ インストール ファイルを作成します		
Web アプリ		MATLAB Compiler™ MATLAB Compiler を使用して配布用 Web アプリを作成します
スタンドアロンのデスクトップ アプリ		MATLAB Compiler™ MATLAB Compiler を使用してスタンドアロンのデスクトップ アプリケーションを作成します

- コンポーネント配置でアプリ作成
 - コンポーネントライブラリ
 - コールバック関数



App Designer

成果物をウェブアプリとして展開

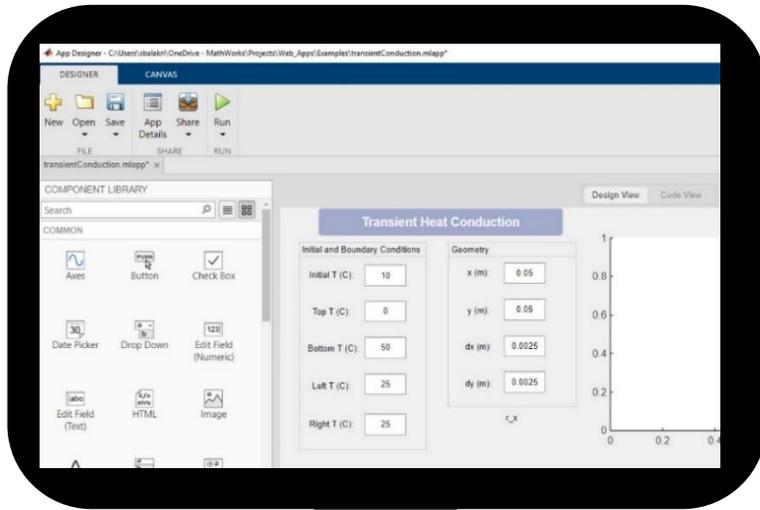
アプリを作成



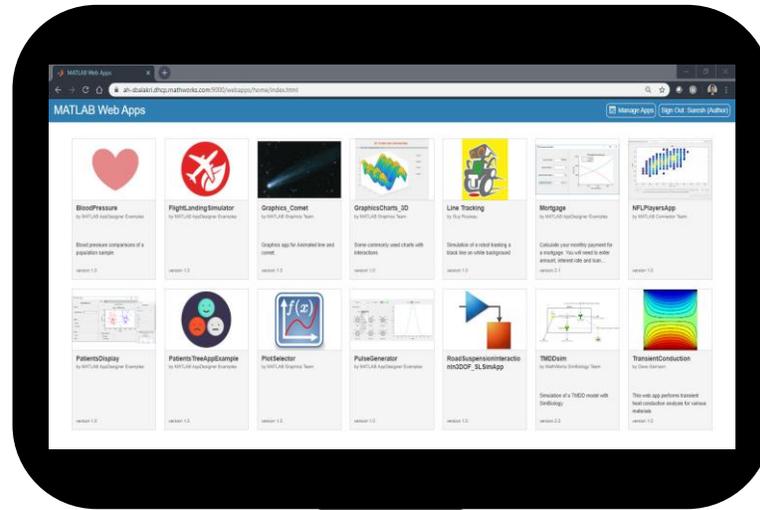
ホストし、アプリを共有



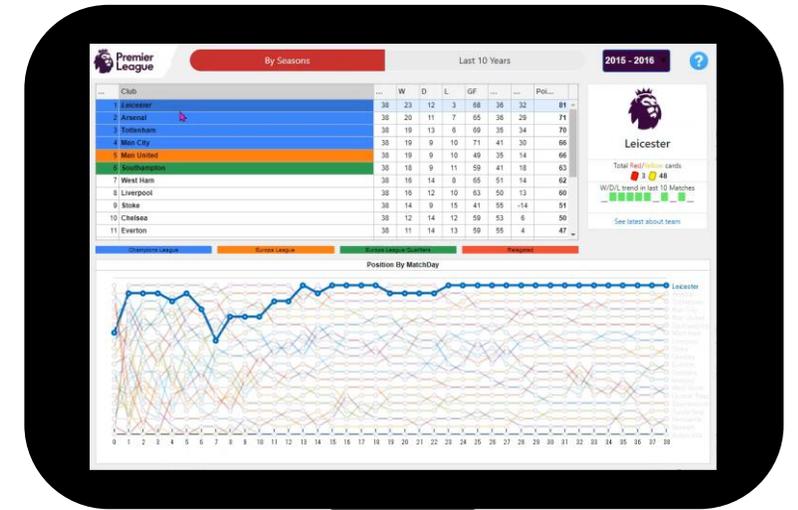
アクセスし、アプリ実行



App Designer



MATLAB Web App Server



デモで使用したツール

MATLAB

- heatmap
- LiveTask - normalize
- corrcoef
- xcorr

Statistics and Machine Learning Toolbox

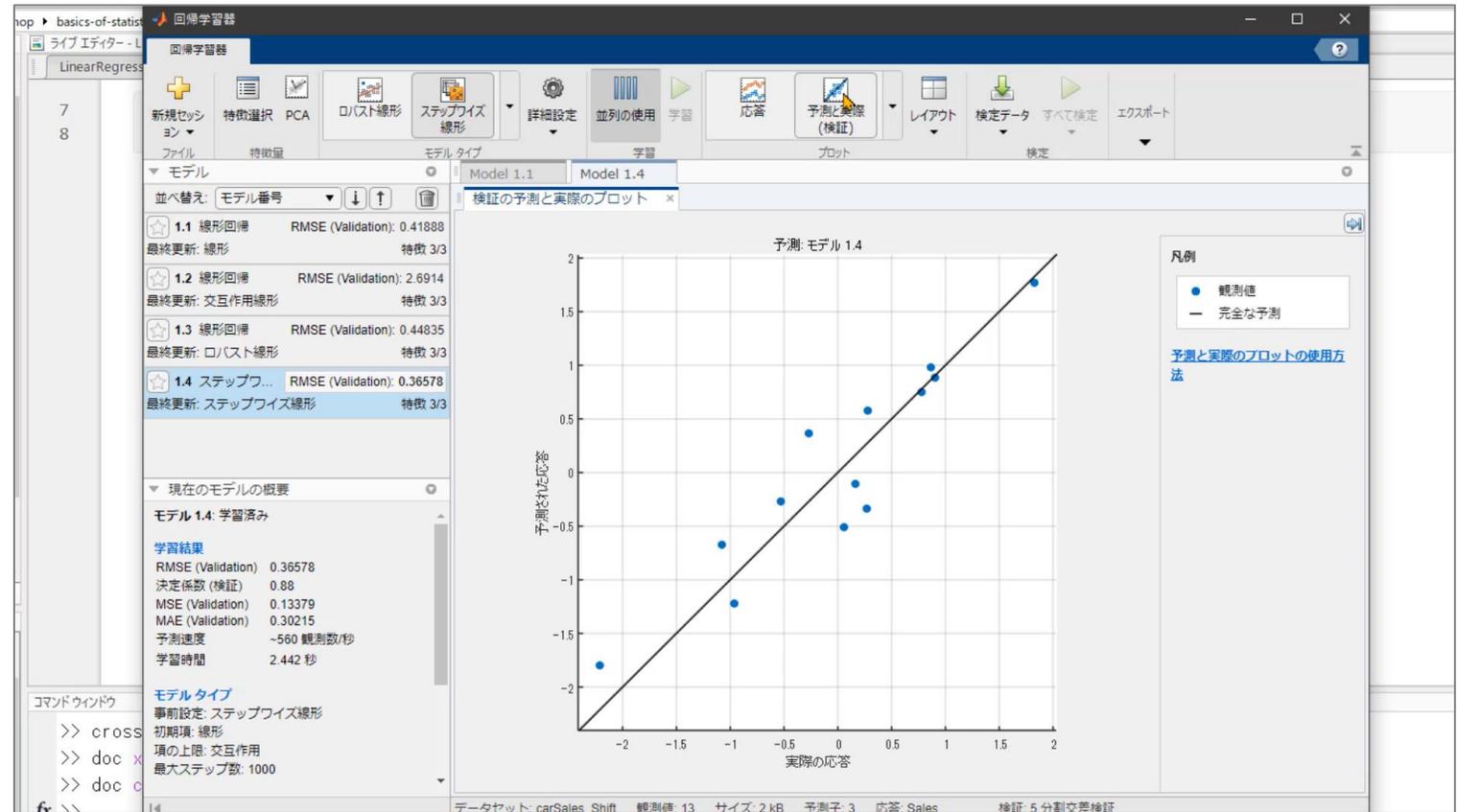
- zscore
- fitlm, stepwise
- etc...

Econometrics Toolbox

- corrplot
- collintest
- corrplot

Curve Fitting Toolbox

- 曲線近似アプリ



回帰学習器アプリ

- 複数モデルの比較
- ハイパーパラメータ最適化
- 並列計算にも対応
- コード生成

無償トレーニングコースでスキルをランプアップ

[詳細はこちら](#)

- ウェブブラウザで動作
- 使い方 & 理論の入門



MATLAB 入門

15 個のモジュール | 2 時間 | 英語

最短で MATLAB の基礎を学びましょう。



Simulink 入門

14 個のモジュール | 2 時間 | 英語

最短で Simulink の基礎を学びましょう。



Circuit Simulation Onramp

7 個のモジュール | 2 時間 | 英語

Simscape で電気回路をシミュレーションするための基礎を学びます。



機械学習入門

6 個のモジュール | 2 時間 | 英語

分類問題のための実用的な機械学習手法の基礎を学びます。



ディープラーニング入門

5 個のモジュール | 2 時間 | 英語

ディープラーニング手法を使用した画像認識を行う方法を学びましょう



強化学習入門

5 個のモジュール | 3 時間 | 英語

強化学習ベースのコントローラを設計するための基礎を学びます。



画像処理入門

6 個のモジュール | 2 時間 | 英語

MATLAB で実用的な画像処理の基本を学びます。



信号処理入門

7 個のモジュール | 1 時間 | 英語

スペクトル解析のための実践に即した信号処理方法を対話形式で説明します。



Wireless Communications Onramp

6 個のモジュール | 1 時間 | 英語

Learn the basics of simulating a wireless communications link in MATLAB.



Simscape 入門

9 個のモジュール | 1.5 時間 | 英語

Simscape で物理システムをシミュレーションするための基礎を学びます。



Stateflow 入門

12 個のモジュール | 2 時間 | 英語

Stateflow でステートマシンを作成、編集、およびシミュレーションするための基礎を学びます。



Simulink による制御設計入門

7 個のモジュール | 1 時間 | 英語

Simulink で基礎的なフィードバック制御系の設計方法を学びます。



最適化入門

5 個のモジュール | 1 時間 | 英語

MATLAB で最適化問題を解くための基礎を、問題解決型のアプローチで学びます。

データ解析のプロになる基礎トレーニングコース (有償)

MATLAB による統計解析

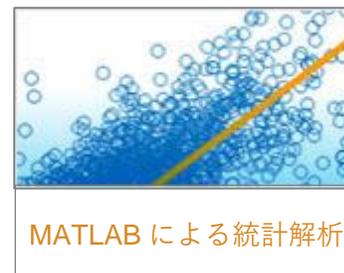
次回 5/19, 20

[トレーニングの詳細](#)

この2日間コースの受講により、MATLAB と Statistics and Machine Learning Toolbox を使用して統計解析を行うために必要な関数の使い方を体系的に幅広く習得できます。基本的かつ重要度が高いと考えられる一連の統計的手法 (分布近似、仮説検定、分散分析、回帰、次元削減など) を題材としています。例題と演習ではデータのインポートと整理を行った後、各手法を実行して MATLAB と Statistics and Machine Learning Toolbox で提供される機能の使い方を学べます。

- データのインポートと整理
- データの調査
- **分布**
- **仮説検定**
- 分散分析
- **回帰**
- 多次元データの取り扱い
- 乱数とシミュレーション

[自己学習コースもあり*](#)



*要 包括契約+Online Training Suite

AI 4days

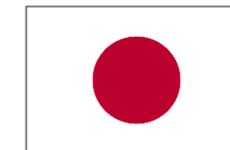
不定期開催

次回 6/16, 17

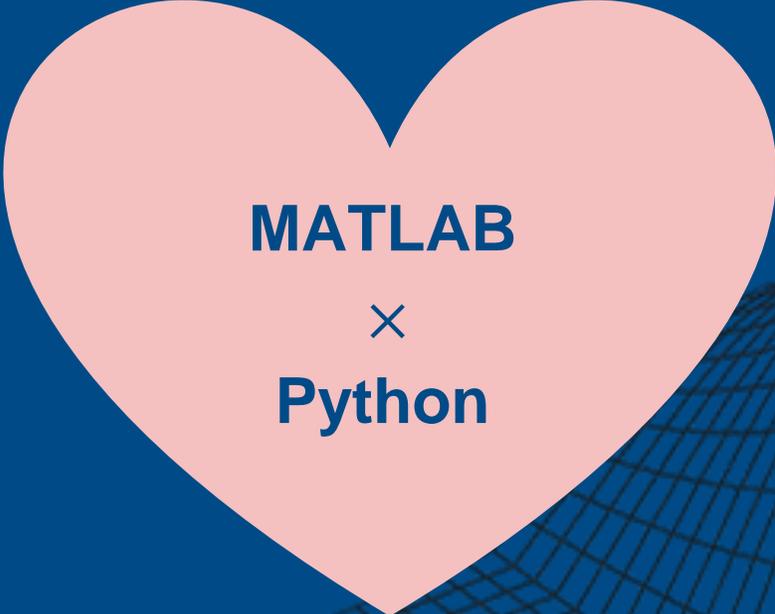
- **機械学習** アルゴリズム・理論入門 (2 days)
 - 代表的なアルゴリズムと理論の座学
 - AI 仮想プロジェクト実習

次回 6/23, 24

- **回帰分析** – 統計/機械学習/深層学習 (2 days)
 - 回帰分析の理論の座学
 - 仮想プロジェクト実習

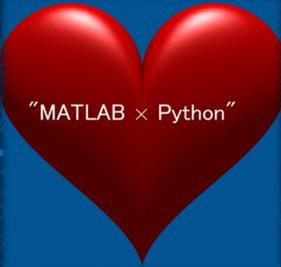


MathWorks Japan
限定トレーニング



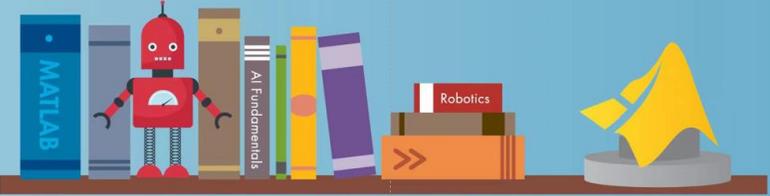
MathWorks

MATLAB × Python
良いとこ取りの二刀流がもたらす
最新かつ高効率なデータ解析・機械学習ワークフロー



MathWorks
吉野 紘和 (JP), Yann Debray (FR)

YouTube



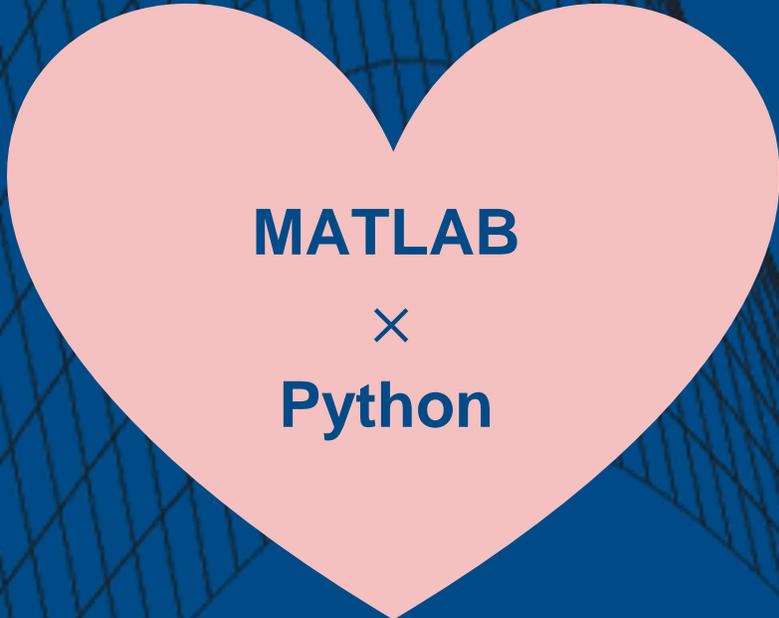
MATLAB × Python

Python ユーザーのための
最適 AI ワークフロー構築

MathWorks Japan
吉野 紘和

虎の巻

YouTube



MATLAB Answers

ユーザーによるQ&Aコミュニティサイト。
お困りごとがあれば過去の回答が確認でき、ご自身の学習にも使えます

[YouTube: 正しい疑問の解決の方法](#)

The screenshot shows the MATLAB Answers website interface. At the top, there's a search bar and navigation links. Below that, a banner reads 'MATLAB と Simulink について質問して回答をもらおう' with statistics: 251,198 answered questions, 135,238 accepted answers, and 295,694 contributing members. The main content area displays search results for the query 'if文の作り方を教えてください。' (How to write an if statement?). The first result is from user 'masaki yamate' (2017-01-20), answered by 'Tohru Kikawada' (2017-01-25). Other results include questions about CSV file creation and plotting points.

• MATLAB Answersの特徴

- 技術的な内容に関するQ&Aコミュニティサイト
- 過去十数万件のQ&Aの検索/閲覧及び投稿が可能
- MathWorksスタッフも回答



• MATLAB Answersのメリット

- ご自身でドキュメントを調べるより
時間の短縮になる
- 過去の回答から学ぶことが可能
- 他ユーザー視点からの回答の参照が可能

2月の関連セミナー告知

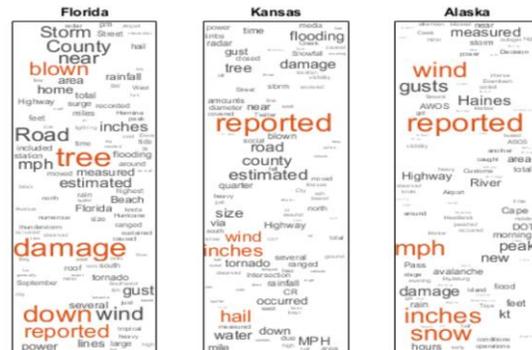
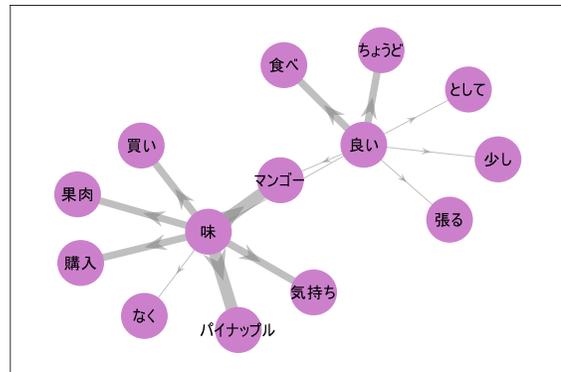
➤ 2月16日(水)14:00-
初心者歓迎！シナリオ毎に学ぶMATLAB
テキストマイニング最前線



詳細確認 &
登録は [こちら](#)

ハイライト

- アンケート解析
- 大規模テキスト
- 将来予測
- GPT-2, BERT
- 成功事例



➤ 2月22日(火)14:00-
現場から学ぶ！データサイエンスからみた
予知保全手法の体系化

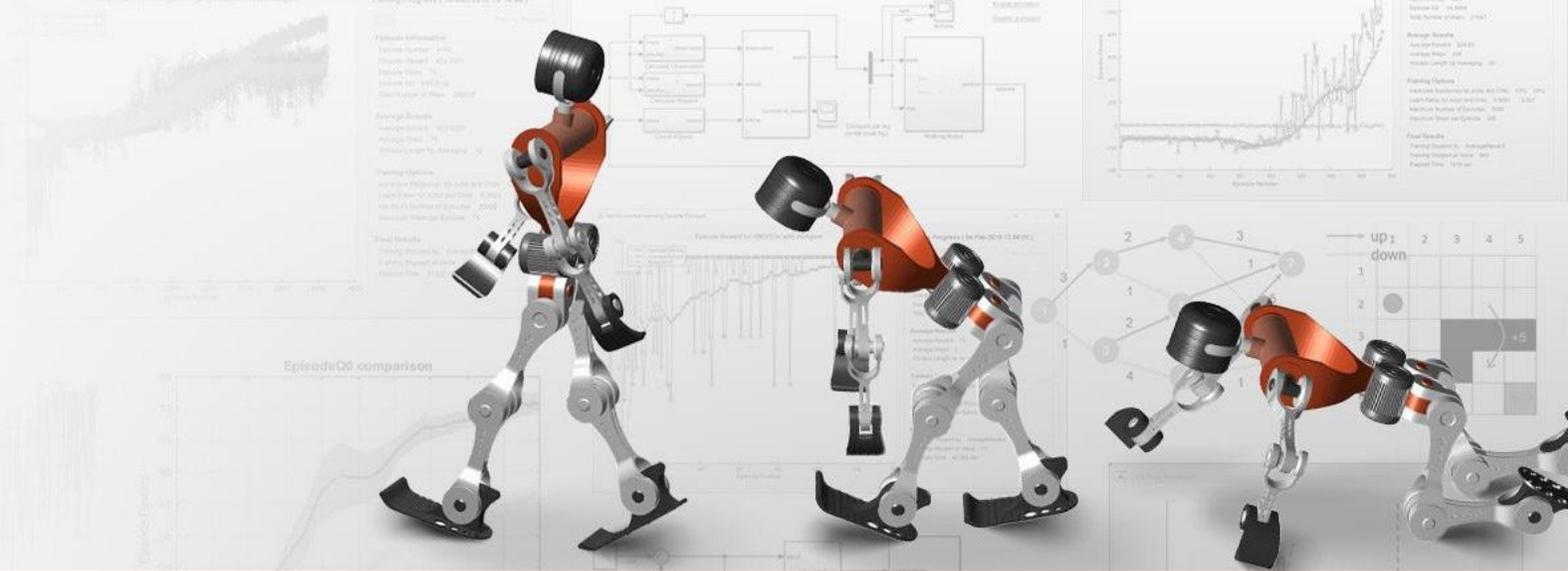
詳細確認 &
登録は [こちら](#)

ハイライト

- 予知保全のための統計手法俯瞰
- 前処理として故障現象のモデル化
- データ解析人材育成のためのヒント



MATLAB® & SIMULINK®



```
>> clear;  
>> disp("ご清聴ありがとうございました!");  
>>  
>> x=[-2:.001:2];  
>> y=(sqrt(cos(x)).*cos(200*x)+ ...  
sqrt(abs(x))-0.7).*(4-x.*x).^0.01;  
>> plot(x,y);
```

fx >> |