

最適化ツール**Dakota**の
計算結果を**R**で分析する

OpenCAE勉強会@岐阜
SH

Agenda

- ① 最適化ツールは何に使うのか(最適化ツール概要)？
- ② CAEの商用最適化ツールの機能概要
- ③ オープンソース Dakotaについて
 - Dakotaと各ツールとの連携方法概要
 - Dakota 新GUIツール
- ④ Rと R-studio について
- ⑤ Dakotaの計算結果をR自己組織化マップ
KOHONENを使って分析

最適化ツールは何に使うの？

最適化手法/最適化アルゴリズムとは？

→ **何かの目標値を最大化または最小化するためのやり方、手順**

ー最適化手法の例

制御工学：変分法、最大原理、動的計画法

応用数学：

線形計画法 LP (Simplex法, 内点法)

非線形最適化法 NLP

ー制約条件無し：最急降下法, Newton法, 共役勾配法 (CG法)

ー制約条件有り：ペナルティ関数法, 逐次2次計画法 (SQP法)

経営工学：整数計画法, ミニマックス問題, ゲーム理論

**情報工学：ニューラルネットワーク (NN), 進化戦略 (ES),
遺伝的アルゴリズム (GA), 遺伝的プログラミング (GP)
人工生命, シミュレーテッドアニーリング (SA), PSO**

最適化ツールは何に使うの？

- 最適化の手法は腐るほどあり、またどんどんあたらしい手法が提案されていますが、大きく分けて、目的値や変数を連続量(実数連続関数)として扱う方法(古典的な方法とか数理計画法とか呼ばれている)と、離散値(整数,2進数など)として扱う方法(一種の組み合わせ最適化問題, 例)遺伝的アルゴリズムGAなど)があります。
- 特徴として以下のような特徴があります

	特徴	具体手法
古典手法 (数理計画法)	計算量: 少ない 勾配情報が必要 局所最適解に収束 データ: 連続関数(実数)	SQP(逐次2次計画法) ラグランジュ乗数法 ペナルティ法
最近の手法 (ヒューリスティックアプローチ)	計算量: 多い 勾配情報不要 大域最適解に収束できる かもしれない? データ: 離散値(整数)	GA(遺伝的アルゴリズム) SA(焼きなまし法) PSO(粒子群最適化)

(参考) 最適化で用いられる用語

ー最適化：構造、寸法、状態、方法、過程などを最も目的に良く合致させること

ー設計変数（説明変数）：最適化を行う際に変化可能な変数（例：寸法、材料、荷重）

ー目的関数（評価関数）：目標の達成度合いを示す関数（例：軽量化（質量）、最短化（時間）、応力最小化（応力））

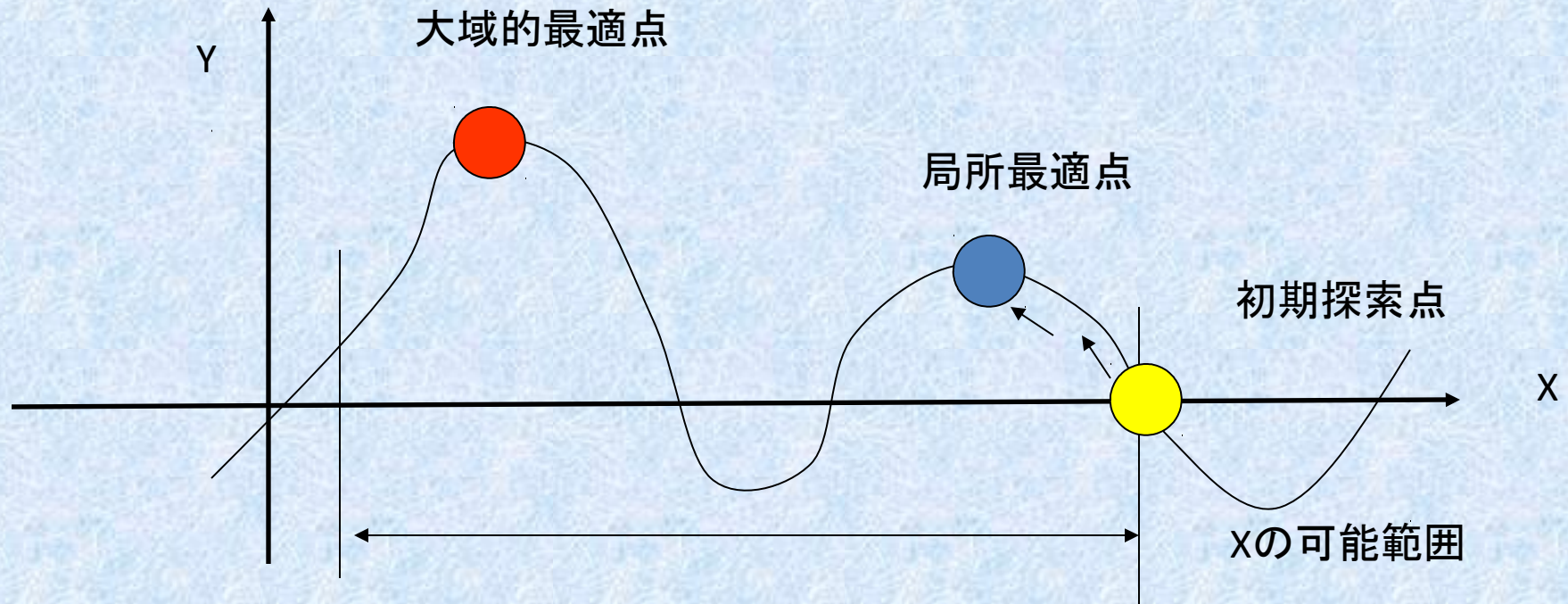
ー制約条件：最適化問題に課せられる条件（例：最大応力○ ○ 以下、金額、時間）

ー大域的最適化：解の全空間での最適化

ー局所最適化：ローカルな最適化を行う

ー単一目的最適化と多目的最適化

(参考) 大域的 (Global) 最適化と局所 (Local) 最適



— 局所最適化：ある部分での局所的な最適化

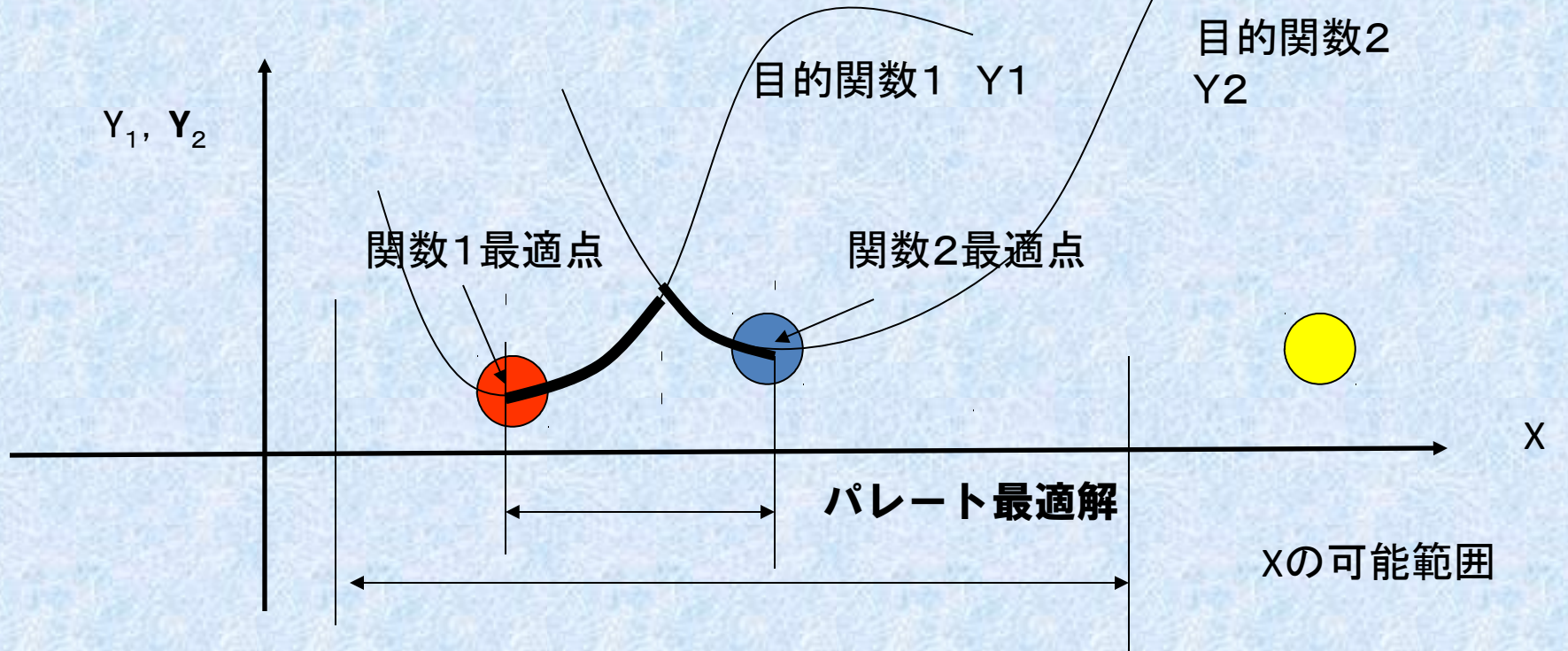
数理計画法などの古典的手法が確立

— 大域的最適化：解の全空間での最適化

⇒ 数理計画法などの古典的な局所的最適解を求める
アルゴリズムでは、大域的な最適解を得ることは難しい。

⇒ GA, SA などの手法が用いられる。

(参考) 単一目的最適化と多目的最適化



— 単一の目的関数を持つ場合には唯一の最適解が存在する場合でも、多目的最適化問題では、2つ以上の目的関数が単一の最適解を持つことはあまり無い。

⇒ 多目的最適化問題では、他の部分より良いという意味で最適解の集合を求め、これらをパレート最適解と呼ぶ。

多目的最適化アルゴリズム

⇒ スカラー化, 優先順位法 などの手法が用いられる。

シミュレーション最適化手法

-解析分野から見た最適化手法の分類-

具体的な解析手法：

-パラメータ（寸法）最適化問題（有限個のパラメータ）：

- 1) 近似モデルを用いない
 - ・ GAなどの評価関数の勾配を用いないアルゴリズム
- + FEMプログラムの自動反復による最適化（汎用的）
- 2) 近似モデルを用いる
 - ・ 応答曲面法（RSM）、品質工学（タグチメソッド）
 - ・ 感度解析、確率有限要素法、テラ級数近似（線形問題）

-形状最適化問題：

- ・ Basis Vector法（Nastranなどのプログラムに装備）
ベシスベクトル法は、設計が考えられる基本形状候補（ベシスベクトル）の組み合わせにより最適形状を求める手法（力法（Azegami et al., 1994）システム応力の一様化）

-位相最適化問題

密度法・均質化法（菊池 他, Quint）平均コンプライアンスの最小化
[→類似手法：FSD（全応力設計法）構成部材応力を許容値以下]

シミュレーション最適化手法の概要

構造解析分野から見た最適化手法の分類

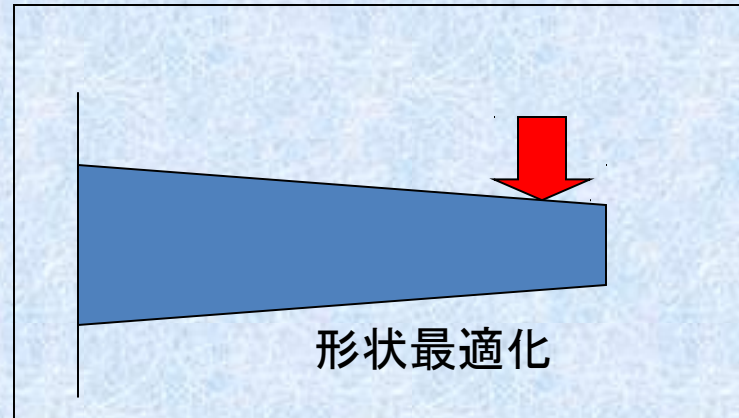
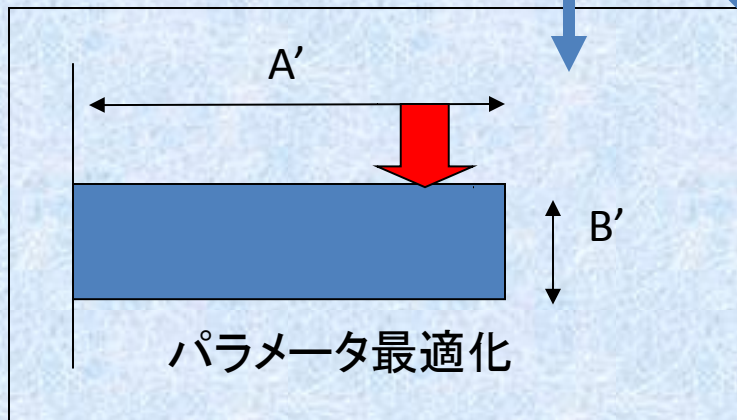
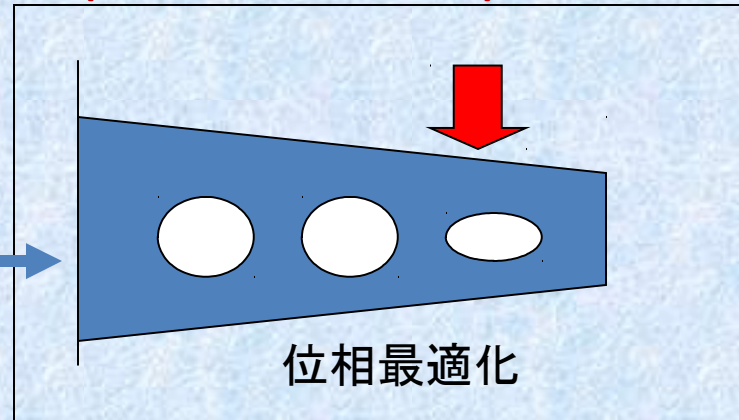
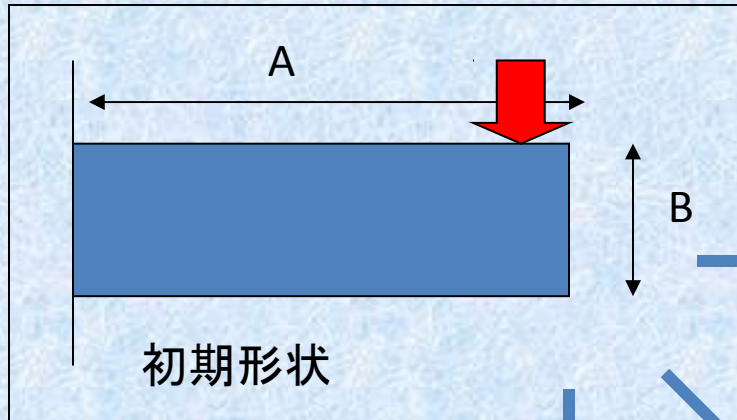
形態による分類：

- ーパラメータ（寸法）最適化問題：寸法、材料物性などパラメータを最適化
- ー形状最適化問題：位相を維持しつつ形状（物体境界位置）を最適化
- ー位相最適化問題：位相幾何学形状を最適化

ノンパラメトリック最適化

Adjoint法はこっち（設計変数が無限）

DAKOTAやOpenMDAOはパラメータ最適化
(設計変数が有限個)のソフトである



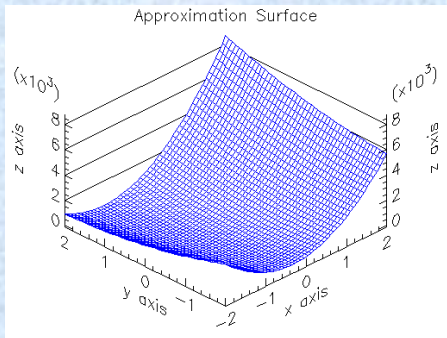
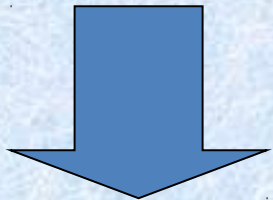
シミュレーション最適化手法の概要

最近の流行

- ・ 近似モデル (応答曲面法=RESPONSE SURFACE METHOD) を用いる手法が多く用いられる。

一理由: シミュレーションでは、ある設計条件が与えられた場合に1つの解 (評価値, 応力や変位や温度 その他) を返すが、設計条件 (設計変数) が変化した場合の応答関数は数学一般における関数の最適化 (最大化、最小化) 問題のようにあらかじめ数式で与えられているわけではない。

CAEの場合1つの設計変数に対する応答点を計算するのに膨大な計算時間を要する



多項式モデル

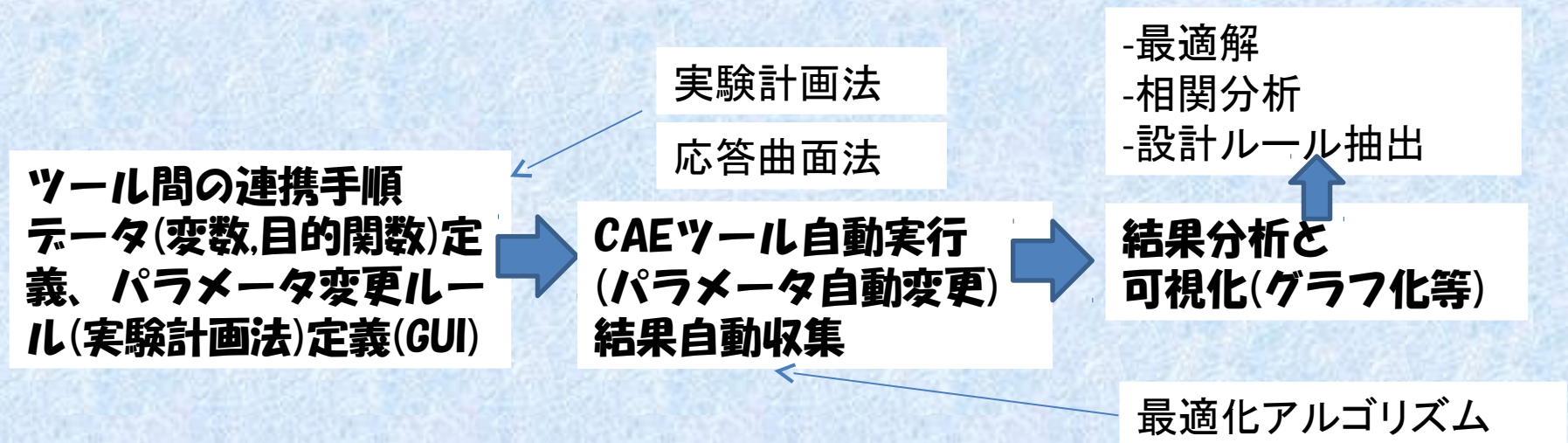
$$Y = \sum a_{ij} x_i x_j + \sum b_i x_i + c$$

幾つかの解析結果から、設計変数変化に対する応答を近似する
近似式 (近似モデル: 例えば2次関数程度) を作成する事ができれば、
その近似式の最適解を求める事は 従来の最適化技術で容易に実現できる。

$$Y = f(x_1, x_2, x_3, \dots, x_n) \rightarrow \max \text{ or } \min$$

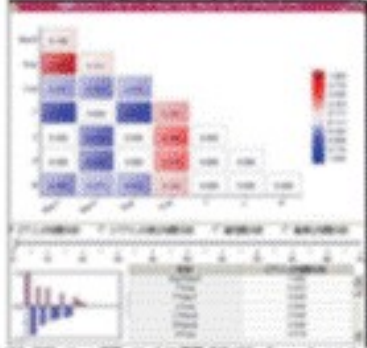
CAE商用最適化ツール (PIDOツール)

- 商用の最適化ツールは最適設計支援ツール(以下PIDOツール: Process Integration & Design Optimization)と呼ばれて基本機能はCAEソフトの自動実行・ツール間統合・最適化アルゴリズム・近似モデル作成(応答曲面)、結果データ分析(可視化)などが行える。
- 代表的なソフトにmodeFrontier(IDAJ社), Isight(ダッソー社), Optimus(サイバーネット社)などがある。



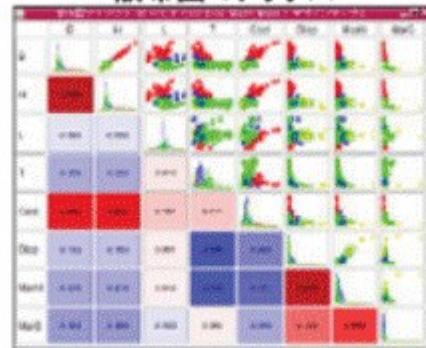
市販CAE最適化ツールの例 (modeFrontierの例)

相関マトリクス



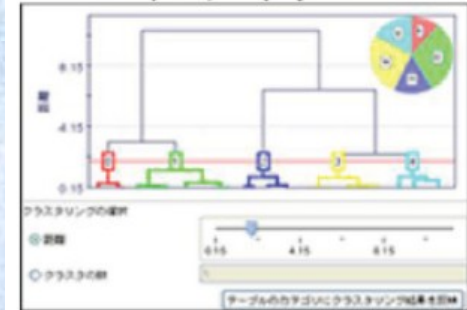
各2変数間の4種の相関係数をマトリクス表示
相関係数値の大きい2変数をランキング表示

散布図マトリクス



散布図において色によるカテゴリ表示が可能
拡大縮小表示により詳細を把握

デンドログラム



クラスタ散布図



直感的でわかりやすい日本語GUI



豊富なダイレクトインターフェースを
標準搭載



分散実行環境構築が容易



実験データを活用した仮想最適化



豊富な応答曲面手法



一般的な実験計画法から独自手法まで



他社の追随を許さない
最先端の探索アルゴリズム



多目的ロバスト設計による同時満足化

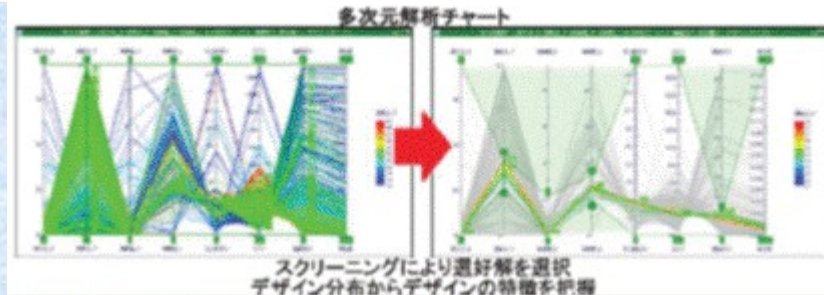


50以上の結果処理機能、
多変量解析機能



設計の原理原則抽出支援機能(CAP)

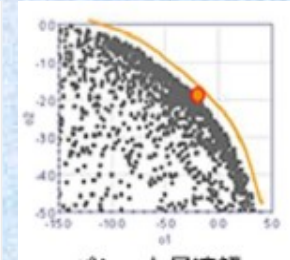
基本モジュール / Runtimeモジュール
Advance / Standard / Basic



クラスタリングSOM



クラスタ多次元解析チャート



パレート最適解

➡ Isight, Optimus などと同様機能の他
- **多目的最適化機能**(MOGA等, パレート最適解), **結果分析**(マイニング機能)

市販ツールとOSSツール比較

	商用ソフト (modeFrontier, Isight, OPTIMUS)	OSS Dakota	OSS OpenMDAO	備考
サンプリング 実験計画法	○	○	○	OSSでも結構いける 直交表(OA), ラテン超方格子(LH)、CCD etc.
最適化アルゴリズム	○	○	○	OSSでも結構いける 数理計画, GAなど
近似手法	○	○	○	OSSでも結構いける 多項式近似, Krigingなど
GUIによるツール連携	○	△	×	OpenMDAOは昔GUIがあったが消滅 DakotaはGUI設定画面が復活
モデル形状の 自動変更	△(他ツール)	×	×	モデル形状の自動変更は商用ツールで も無理 → 別の形状作成編集ツール要
結果分析	○	×	×	OSSでは結果分析が苦手

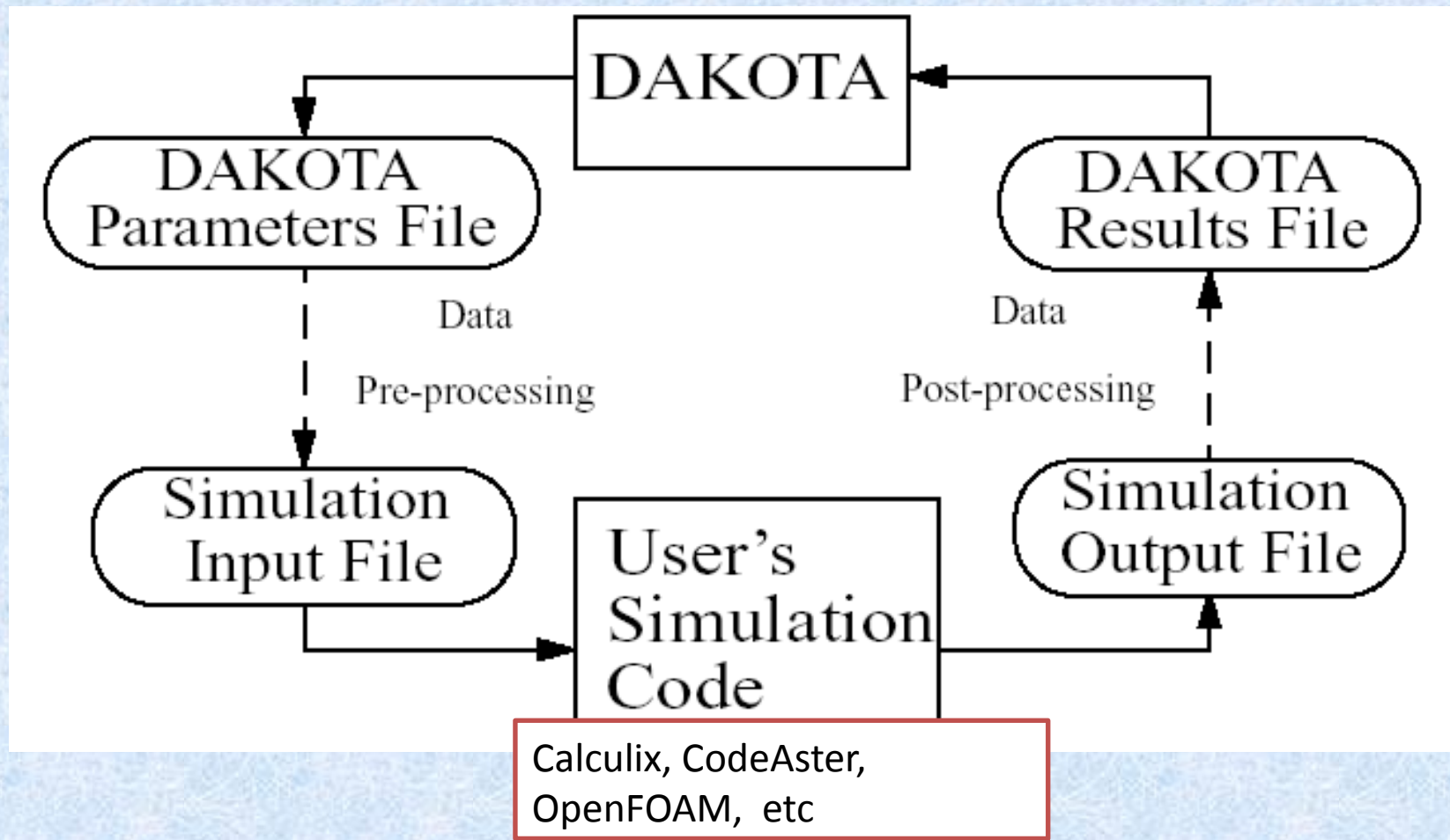
今回はDakotaの結果分析にRを使って行う方法を賢答する。

Dakotaとは①？

自動的に解析に入力するパラメータを変更したパラメータSTUDYや目的となる指標（目的関数）を与えると自動的にそのような目的の値になるようなパラメータを見つけてくれるツールです（最適化ツール）

- ParaViewと同じ Sandia National Laboで開発されている

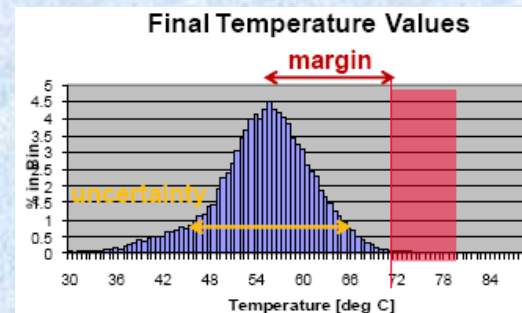
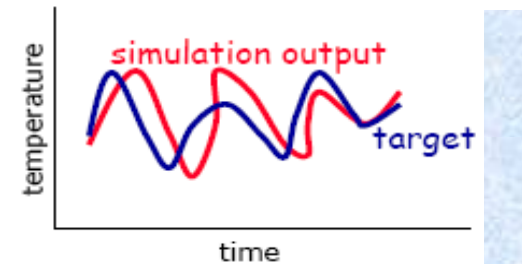
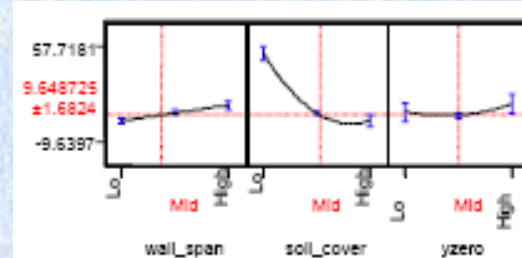
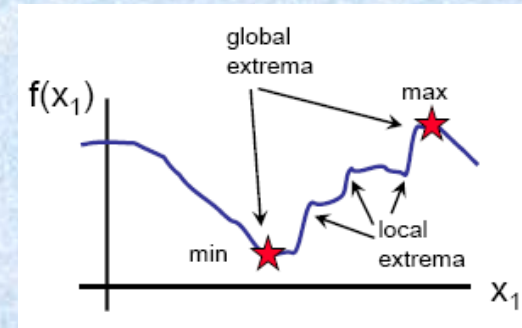
- <http://dakota.sandia.gov/>



Dakotaとは②？

- DAKOTAは具体的には何に使えるのか？

- Optimization 設計最適化:
最適寸法や最適パラメータ決定
- Sensitivity Analysis 感度解析:
入力パラメータ変動に対する出力感度
- Calibration パラメータ同定:
未知パラメータ決定 (物性値同定など)
- Uncertainty Quantification 信頼度:
入力パラメータが変動した場合の応答
変動の確率分布推定

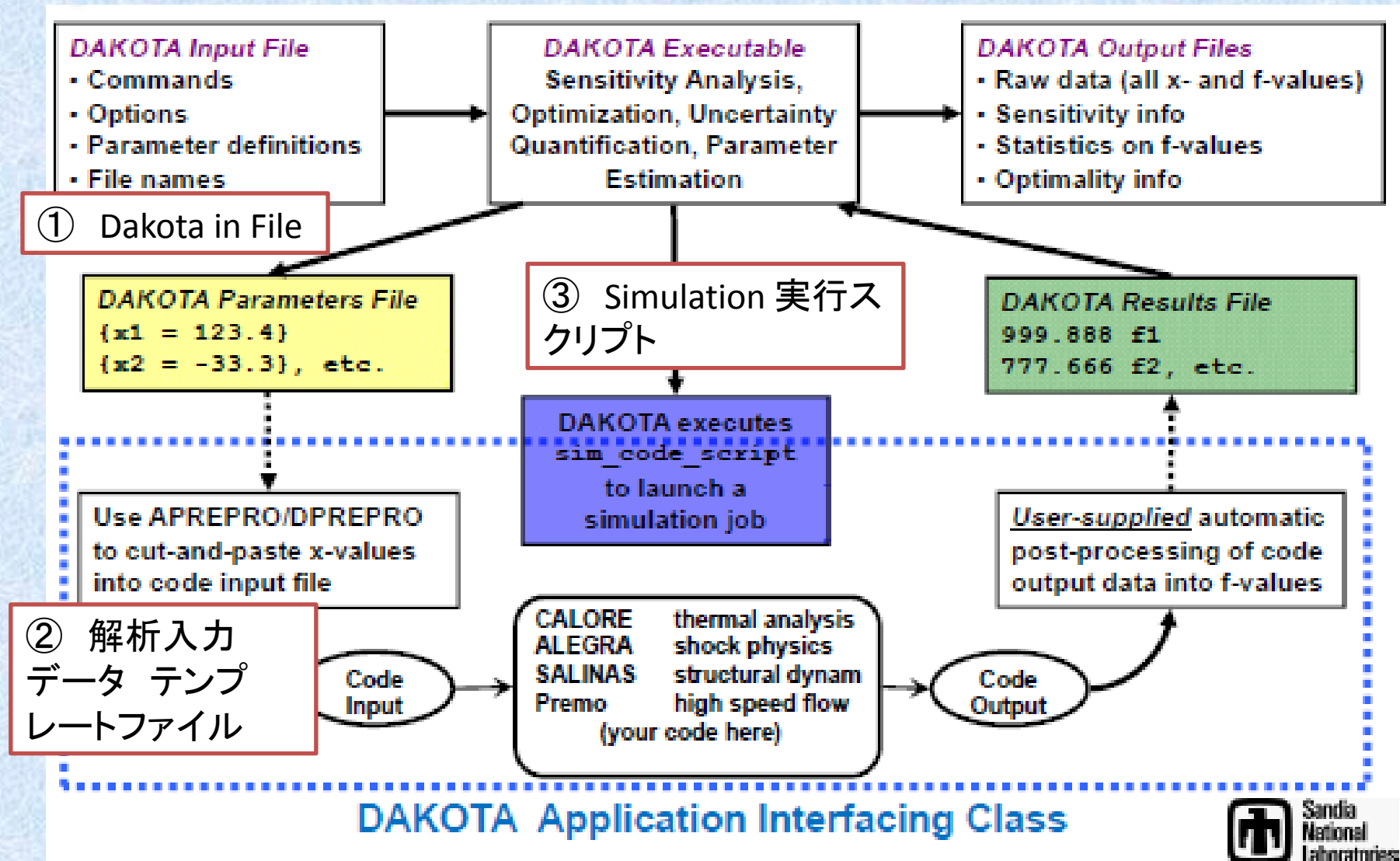


Dakota と各ツールとの連携①

- Dakotaは任意のプログラムや解析ツールと連成する場合、スクリプトなど外部インターフェースを使って自動実行させる。
- 幾つかのスクリプトやbinary 連携ツールがexamples Directory 以下に提供されているのでこれを使って連携実行させる compiled interface 以下にはmatlab 以外はつかえそうもなくscript_interfacesの下のサンプルを利用することにした(V5.3からMatlab以外にOSS scalibの直接interface が準備された)。
- ABAQUS, NASTRANなど解析ツールの他、Python script, Excel Visual Basic script と 連携する例のsample がある。

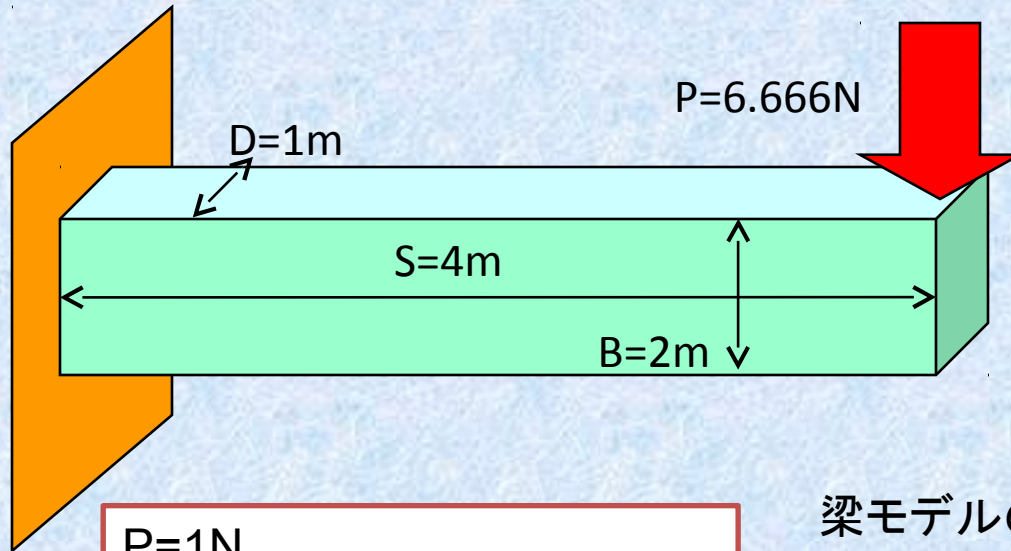
Dakota と各ツールとの連携②

- Dakotaと解析ツールは下記のように連携ユーザは以下3つファイルを準備する必要あり。



Dakota と各ツールとの連携③

- 片持ちはりの反り計算の例題でABAQUSのSample が入っていたので、これをOS S有限要素解析ソフト Calculix で動くように書き換えて実行してみる
- 前提:Linux 環境 (Ubuntu) Calculix linux版 install 済み
- Calculix V2.1 を使用



設計変数: S, B, P

S=2-6m, B=1-3m, P=4-10N

目的関数: 梁の最大応力
 $\sigma \rightarrow$ 最小化

2次元平面応力要素を利用

P=1N
S= 4m
B= 2m
D= 1m
E=30e3Pa
I= bt³/12

梁モデルの最大反り理論解

$$\delta = \frac{WL^3}{3EI}$$

M=PS (モーメント)

$\sigma = M/Z = PS/Z$
 $= PS/(DB^2/6)$
(最大応力)

Dakota と各ツールとの連携④

- ① dakota_abaq_opt.in の中身を書き換える

```
strategy,  
    single_method graphics  
method,  
    dot_mmfd  
variables,  
    continuous_design = 3  
    cdv_initial_point = 4.0 2.0 6.66667  
    cdv_lower_bounds = 2.0 1.0 4.0  
    cdv_upper_bounds = 6.0 3.0 10.0  
    cdv_descriptor = 'S' 'B' 'P'  
interface,  
    application system #asynch_evaluation_concurrency = 5  
    analysis_driver = 'abaq_driver'  
    parameters_file = 'params.in'  
    results_file = 'results.out'  
    aprepro  
responses,  
    num_objective_functions = 1  
    numerical_gradients  
    method_source dakota  
    interval_type  
    fd_step_size  
no_hessians
```

今はこのモジュールなし。とりあえず下記入れ替え
method,
 conmin_frcg
 max_iterations = 100
 convergence_tolerance = 1e-4

Application は現在認識しない
ので 消す system だけ残す

Dakotaに今入っていない
Internet から Source
Download しinstall
[aprepro-2.01.tar.gz](#)
(他のVersionはmake に失敗)

Deprepro(dakotaに入っ
ている Perl script file)にて置
き換えることが可能

Dakota と各ツールとの連携⑤

- ② fe.inp.app の書き換え (fe.inp.app は Calculix (abaqus) の解析入力ファイルのテンプレート)

```
{ECHO(OFF)}
{include(params.in)}
{ECHO(ON)}
*HEADING
*NODE
1, 0.0, 0.0
2, {S/4}, 0.0
3, {S/2}, 0.0
4, 0.0, {B/2}
5, {S/4}, {B/2}
6, {S/2}, {B/2}
7, 0.0, {B}
8, {S/4}, {B}
9, {S/2}, {B}

*ELEMENT,TYPE=CPS8,ELSET=BEAM
1, 1, 3, 9, 7, 2, 6, 8, 4
**ELEMENT,TYPE=CPS4,ELSET=BEAM
** 1, 1, 2, 5, 4
** 2, 2, 3, 6, 5
** 3, 4, 5, 8, 7
** 4, 5, 6, 9, 8
*SOLID SECTION,ELSET=BEAM,MATERIAL=AL2024T3
1.0
*MATERIAL,NAME=AL2024T3
*ELASTIC
30.000000e3, 0.3
*BOUNDARY
1, 2, , 0.0
3, 1, , 0.0
6, 1, , 0.0
9, 1, , 0.0
*STEP
*STATIC
*CLOAD
** 9, 2, 2.333333,
9, 2, {P/2},
*EL PRINT, POSITION AVERAGED AT NODES, ELSET=BEAM
S
*END STEP
```

S, B, P は Dakota と APREPRO で変更する
Parameter {S}, {B}, {P} のように
中括弧でパラメータを指定すると
Aprepro(or Deprepro) がパラメータを自
動で置き換える

Calculix V2.1 では CPS4 が無いので CPS8
に変更した

Calculix では ELSET 指定が必要

Dakota と各ツールとの連携⑥

- ③ Abaqus_driver(解析自動実行スクリプト) を編集

```
#!/bin/csh -f
```

```
# $argv[1] is params.in FROM Dakota
```

```
# $argv[2] is results.out returned to Dakota
```

```
# Workdir setup for running in parallel (file_tag option turned on)
```

```
# In this simple case, all templatedir would contain is fe.inp.app
```

```
#set num = `echo $argv[1] | cut -c 11-`
```

```
#cp -r templatedir workdir.$num
```

```
#mv $argv[1] workdir.$num/params.in
```

```
#cd workdir.$num
```

```
# Pre-processing
```

```
aprepro --nowarning -q fe.inp.app fe.inp
```

```
# Run ABAQUS
```

```
#rm -f *.dat *.sta
```

```
/usr/bin/ccx 2.1 -i fe
```

```
####>>&! abaqus.out
```

```
# Post-processing
```

```
grep ' 1 27' fe.dat | head -n 1 | awk '{print $3}' > $argv[2]
```

```
# Results file move :and workdir cleanup for running in parallel
```

```
#mv $argv[2] ../.
```

```
#cd ..
```

```
#rm -rf workdir.$num
```

このScriptはcsh 向けに書かれているため、bsh 系は書式を少し変える必要あり

このコマンドでaprepro が各パラメータ{S}などを実際の数字に置き換えたファイルを作成する

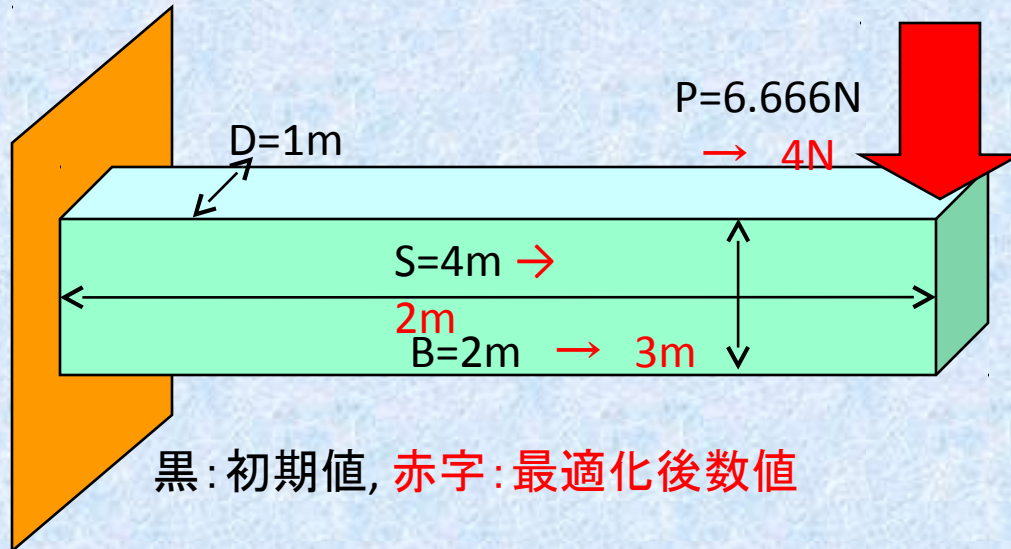
ここをCalculix 用に編集

ここをCalculix 用に編集

Dakota と各ツールとの連携⑦

グラフ作成出力

最適化計算結果



Dakota テキスト出力結果

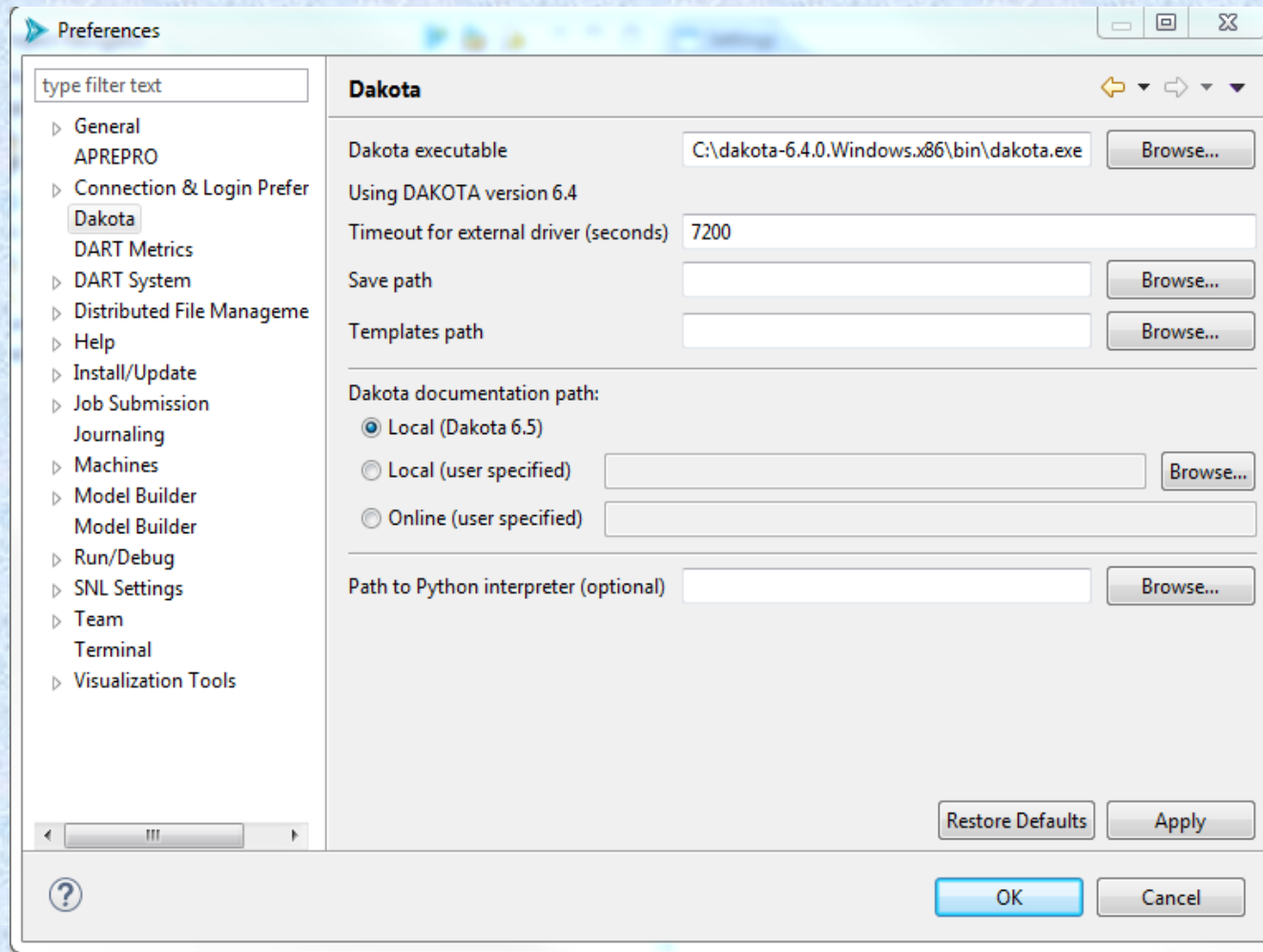
```
FINAL OPTIMIZATION INFORMATION   OBJ =  0.674360
DECISION VARIABLES (X-VECTOR)   1)  0.20000E+01 (
```

%eval_id	S	B	P	obj_fn
1	4	2	6.66667	6.8376
2	3.977324	2.090318	6.653404	6.1956
3	3.886618	2.451589	6.600339	4.2731
4	3.763126	2.94344	6.528094	2.6934
5	3.741134	3	6.517772	2.5533
6	3.65915	3	6.479296	2.4655
7	3.331218	3	6.32539	2.123
8	2	3	5.70062	0.96106
9	2	3	4.560425	0.76884
10	2	3	4	0.67436
11	2	3	4	0.67436
12	2	3	4	0.67436
13	2	3	4	0.67436

梁の長さを短く 厚さ(高さ)を大きく、荷重を小さくすれば応力が最小になるという、常識的な結果が得られる。

最適化ソフト**DAKOTA**のバージョンアップについて

- 最近のバージョンアップ6.5にともなって、GUIが復活した模様



無料のデータ結果可視化・統計分析ツール

- 結果のグラフ化だけならParaViewでそこそこ商用最適化ツールのような表示が可能
- もう少し難しい統計分析(クラスタリングやデンドログラム、主成分分析 など)は統計分析用の専門ソフト(R または Pythonの統計分析関係ライブラリ群)を使うと商用ソフトとほぼ同等のことができる。
- 統計解析、データマイニング関連のフリーのライブラリ(Python、R関連など)の充実度は最近すさまじく、商用ソフトを凌駕していると思われる



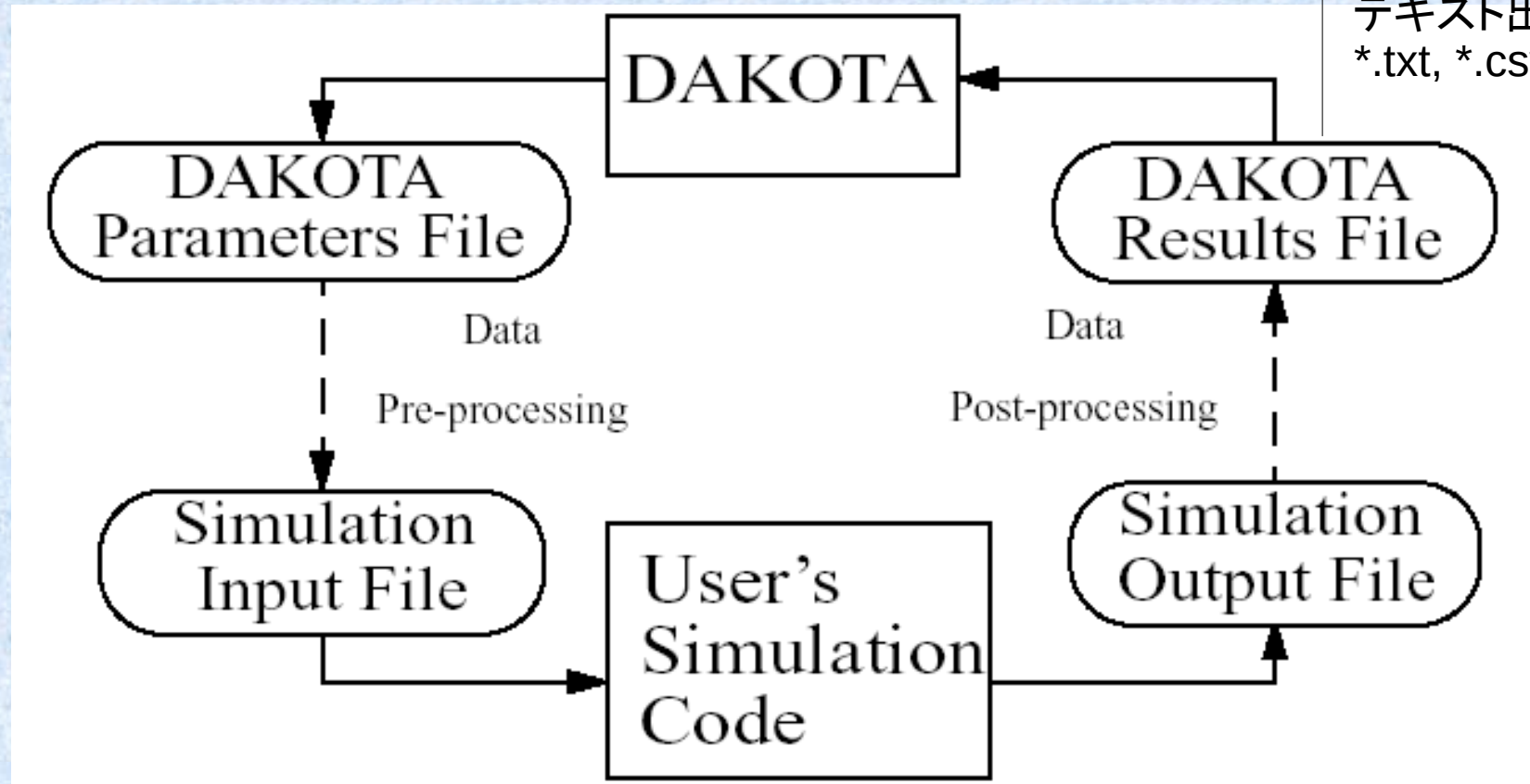
商用の最適化ソフトの売りの一つは大量データの結果分析機能充実であったが、フリーの分析ツールを組み合わせることで、結果分析も問題なくできる。

Dakotaと統計分析ツールの連携方法

- Dakotaで計算を実行させると各変数の出力がテキストデータで出力されるのでこれをRで読み込んで分析する
- うまく読み込みできない場合は事前にLibreOfficeかExcelで成形すると良い

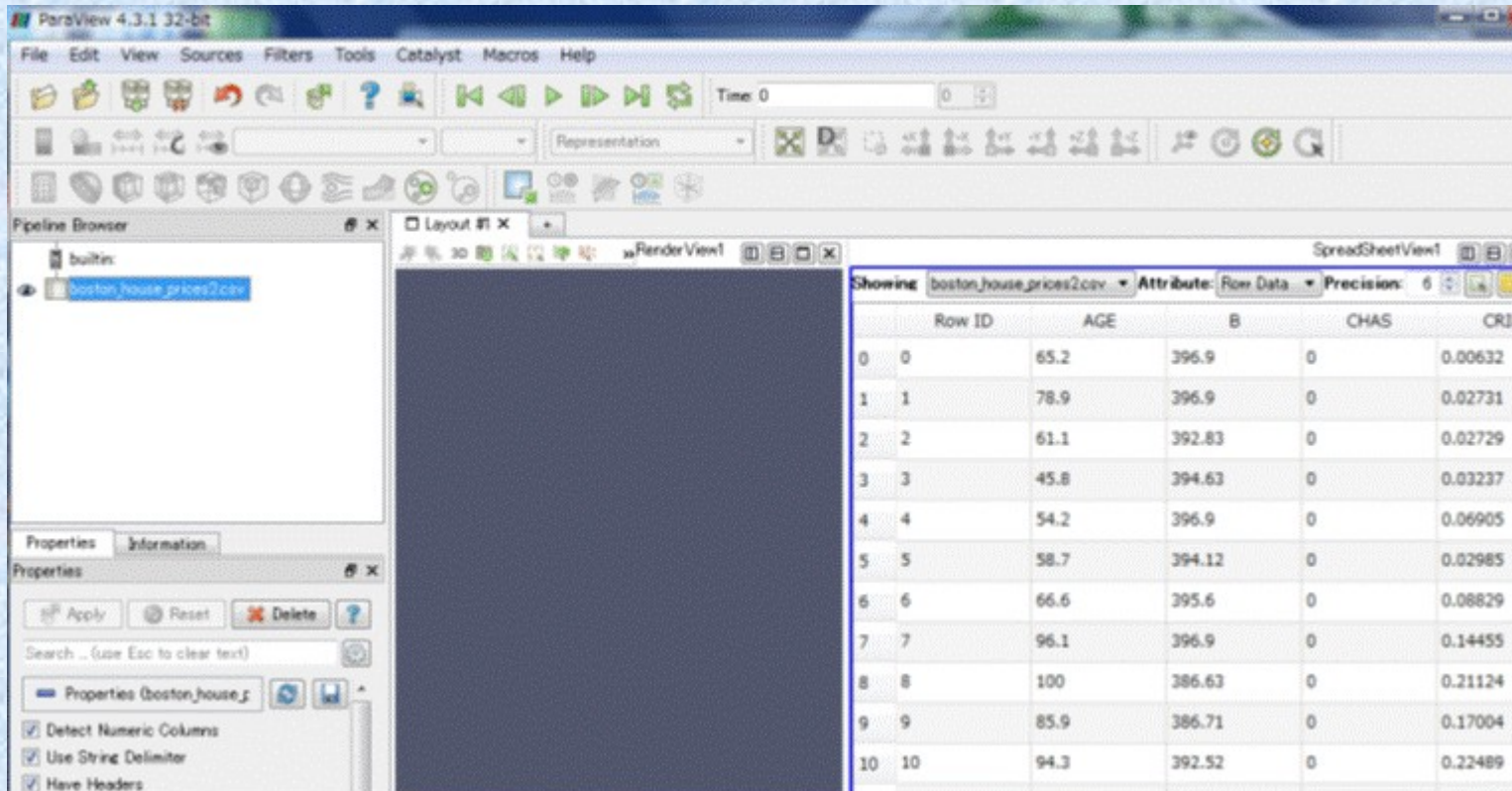
Rなどで分析

テキスト出力
*.txt, *.csv



無料のデータ結果可視化・統計分析ツール ParaViewでの可視化例①

- ParaViewでCSV形式で出力したデータを読み込む。最初はテーブルがそのままこんな感じで表示される

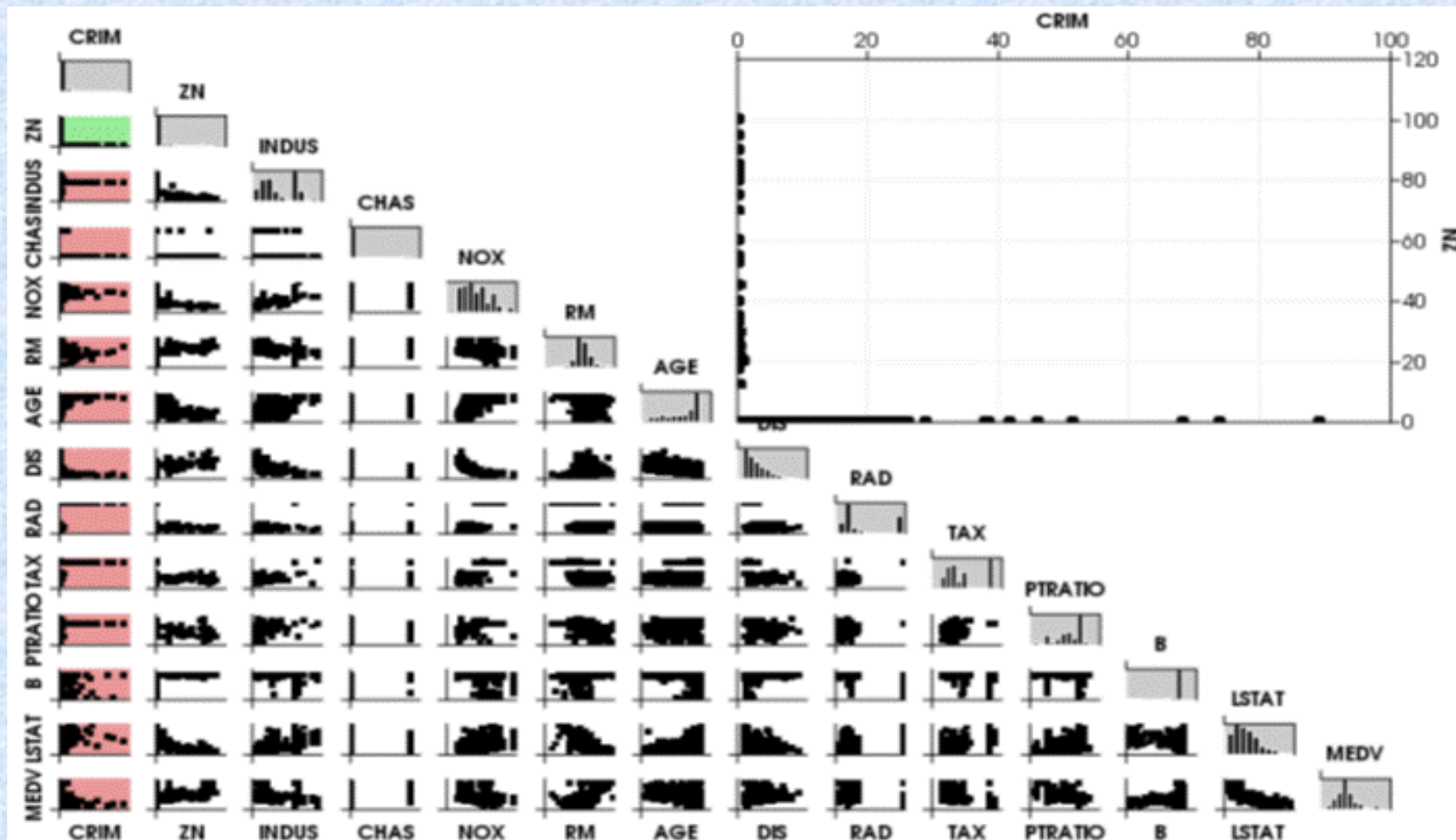


The screenshot shows the ParaView 4.3.1 32-bit interface. The Pipeline Browser on the left lists 'builtin' and 'boston_house_prices2.csv'. The SpreadsheetView1 on the right displays a table with 11 rows and 6 columns. The columns are labeled 'Row ID', 'AGE', 'B', 'CHAS', and 'CRIM'. The data is as follows:

Row ID	AGE	B	CHAS	CRIM
0	65.2	396.9	0	0.00632
1	78.9	396.9	0	0.02731
2	61.1	392.83	0	0.02729
3	45.8	394.63	0	0.03237
4	54.2	396.9	0	0.06905
5	58.7	394.12	0	0.02985
6	66.6	395.6	0	0.08829
7	96.1	396.9	0	0.14455
8	100	386.63	0	0.21124
9	85.9	386.71	0	0.17004
10	94.3	392.52	0	0.22489

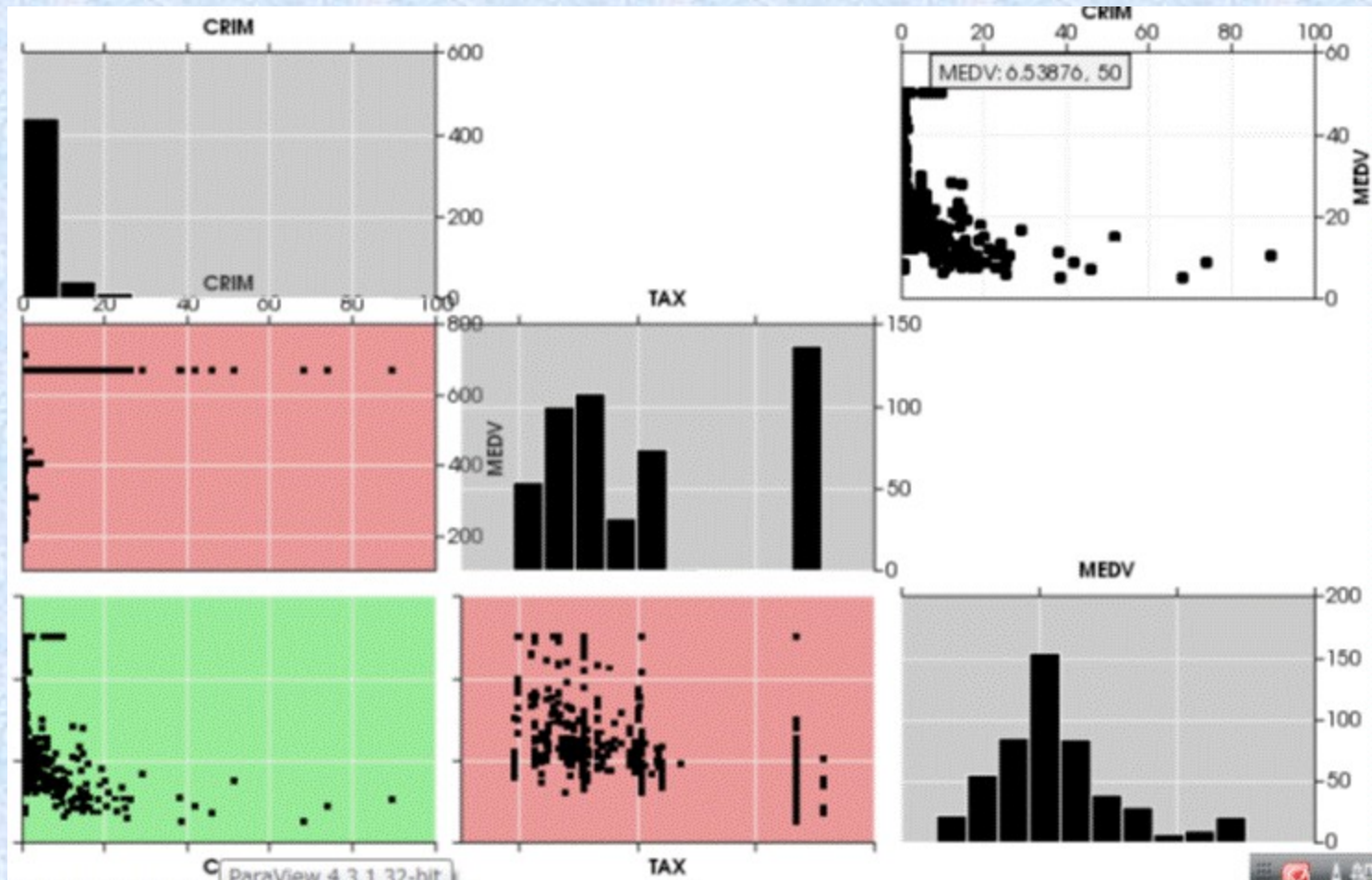
無料のデータ結果可視化・統計分析ツール ParaViewでの可視化例②

- 新規Layoutを選択し、Plot Matrix View を選択すると散布図マトリックスが表示される。



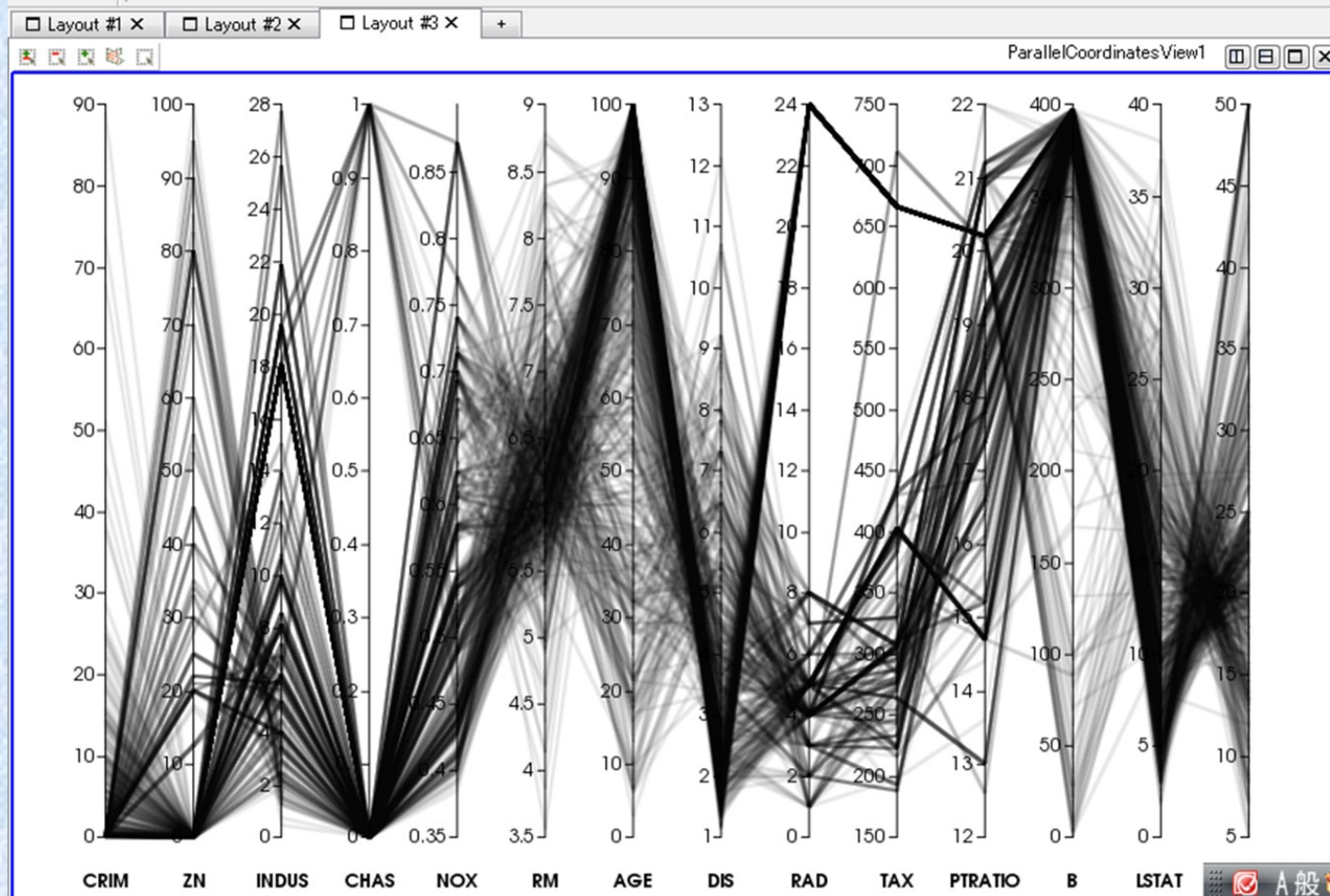
無料のデータ結果可視化・統計分析ツール ParaViewでの可視化例③

散布図マトリックスの変数をしぼりこんだ例



無料のデータ結果可視化・統計分析ツール ParaViewでの可視化例④

新規レイアウトにてParallel Coordinate Viewを選択する。modeFrontier の多次元レーダチャートのようなPlotを描くことが可能



統計解析ソフト**R(R-studio)**によるデータ分析

- ・最近**R**を使っの統計教育(マイニングなど)が盛んであり、**R, R-studio**をインストールしたので**R**を使ったデータ分析

- ・おまけに**1D-CAE**(常微分方程式)の解き方について調べた。

R, R-studioとは？

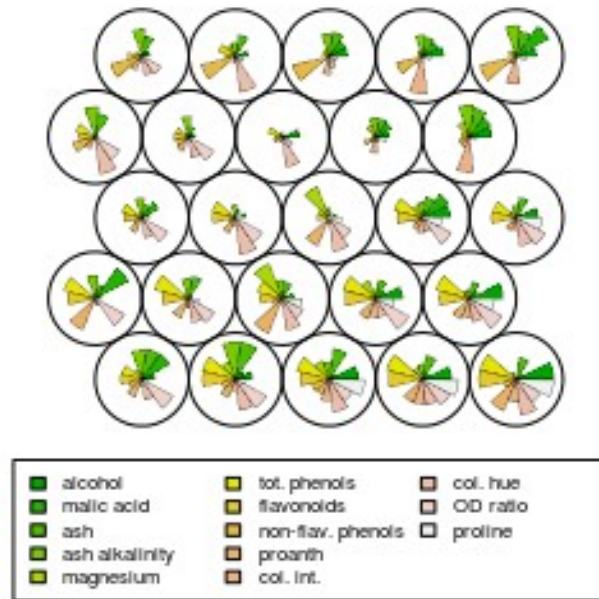
- オープンソースの統計解析向けプログラミング言語
- 商用の統計分析ソフトで**S**というのがあり、そのオープンソース版みたいな関係
- **MATLAB**に対する**Octave, Scilab**みたいな感じのソフト
- **R(R-base)**は**GUI**が全くないので、使いにくい。**R**に**GUI**の開発環境を追加したものが**R-studio**, **RをExcel** みたいに使えるソフトには**R-commander**があるが、こちらは逆にコマンドがつかいづらい。
- **R**の標準では行列(マトリックス)の計算は可能だが、微分方程式を解く機能は無い。**R**で**1D-CAE**を行うためには、追加パッケージをインストールする
- 最近のデータマイニングや人工知能(**NN**)の流行で日本語の関連書籍がものすごく多い。放送大学の授業にも**R**の授業がある
- **OpenCAE**岐阜の**HP**に秋月氏の**R**による**1D-CAE**の資料が公開されている
- データ分析に**R**のパッケージ**Kohonen**自己組織化**MAP(SOM)**を利用する

Rによる統計分析の例

- インストール方法・Rコマンド画面から以下を入力

```
install.packages("kohonen", dependencies = TRUE)
```

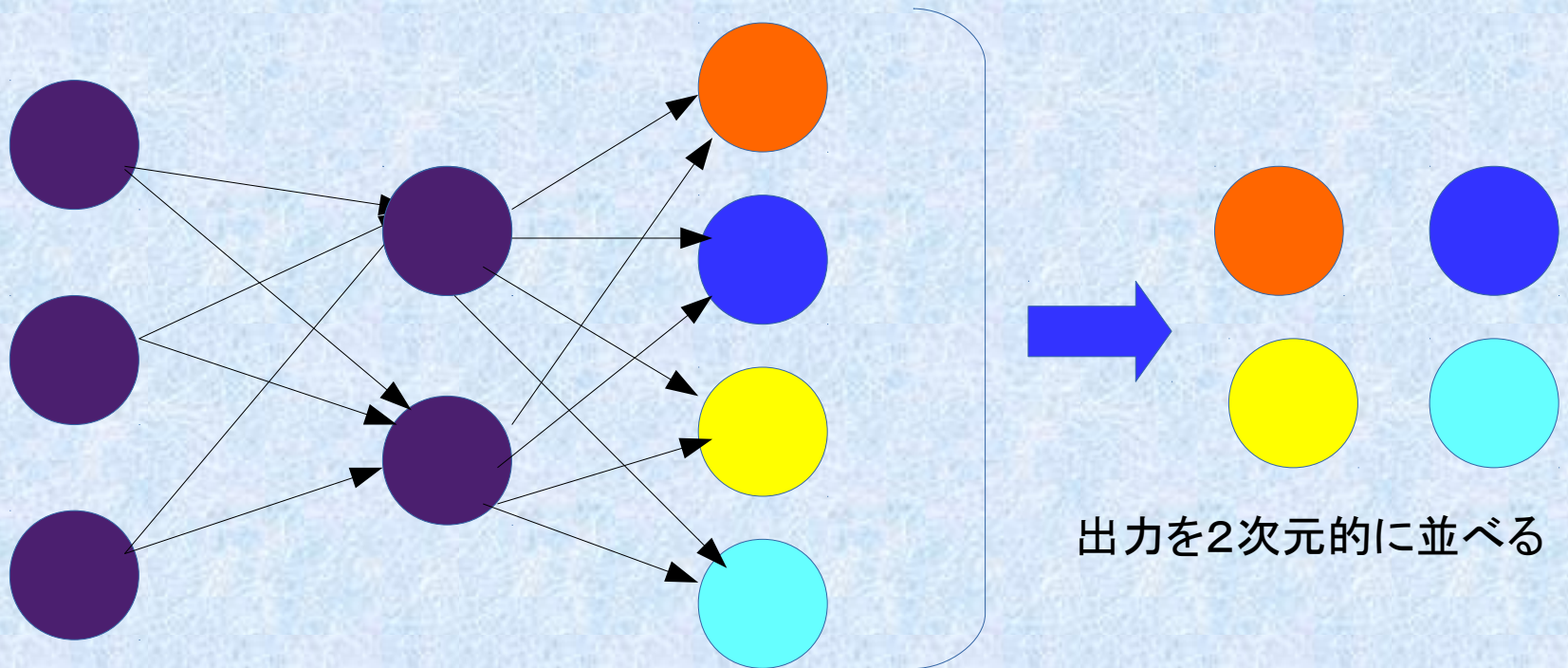
- クラスタ分析、決定木、自己組織化マップ(SOM)などの商用シミュレーション最適化ソフトの後処理でよく利用される



自己組織化マップの例

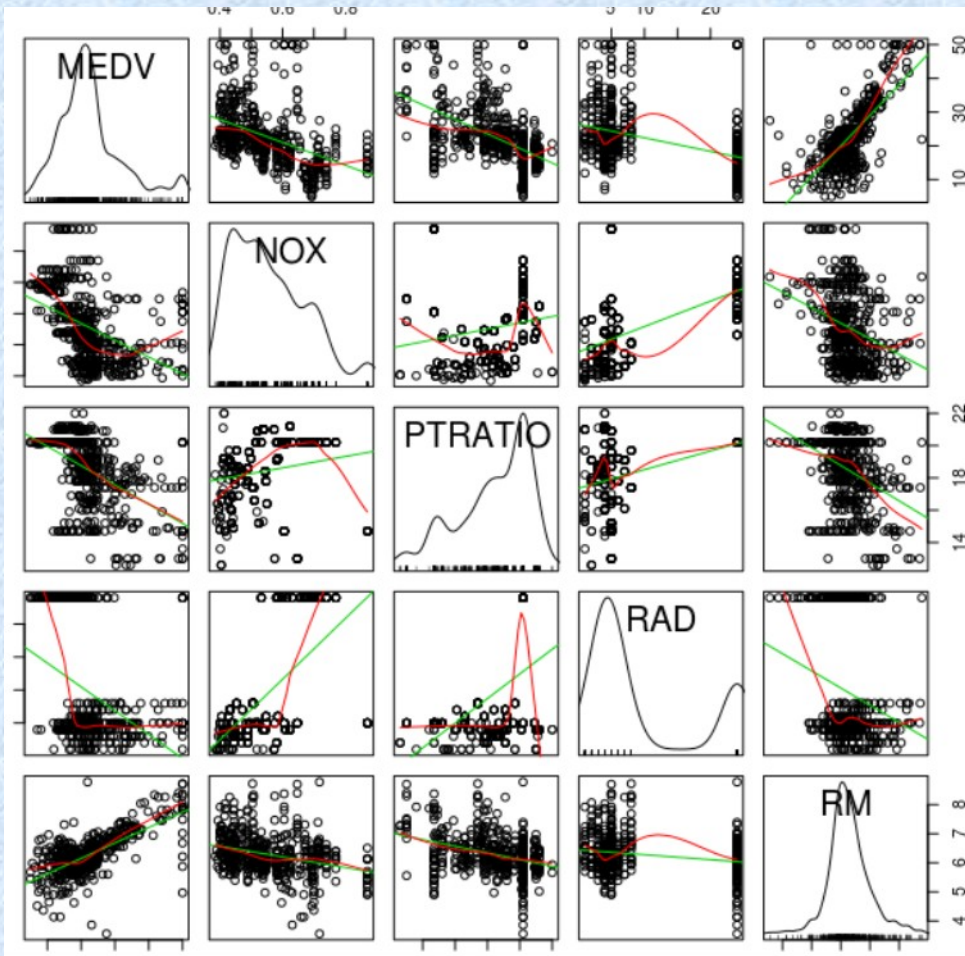
自己組織化マップとは？

- 階層型ニューラルネットの一種
- 教師なし学習: 複数の次元データを2次元マップでなんとなく近いもの通して分類する (クラスタリング手法) の一種

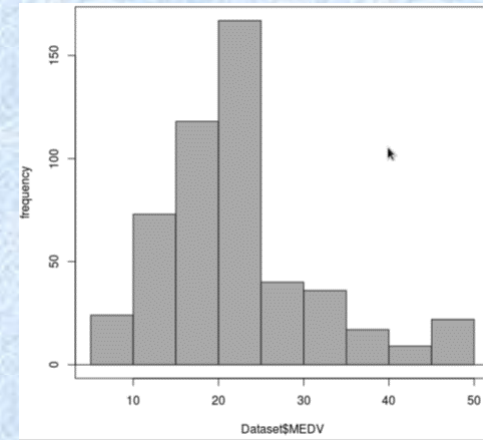


無料のデータ結果可視化・統計分析ツール

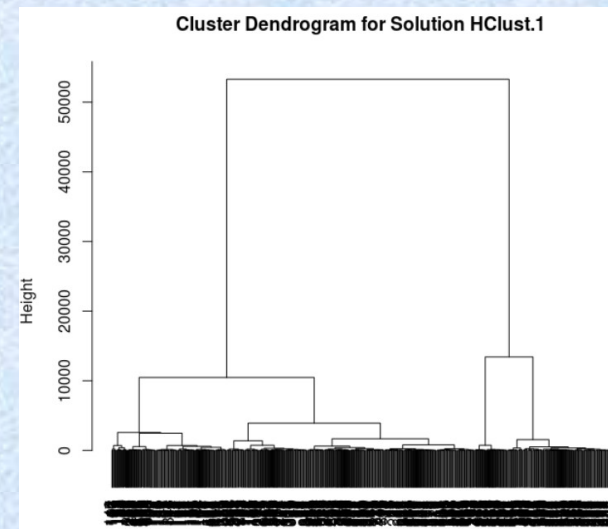
統計解析ソフトRによる分析①



散布図マトリックス



度数分布

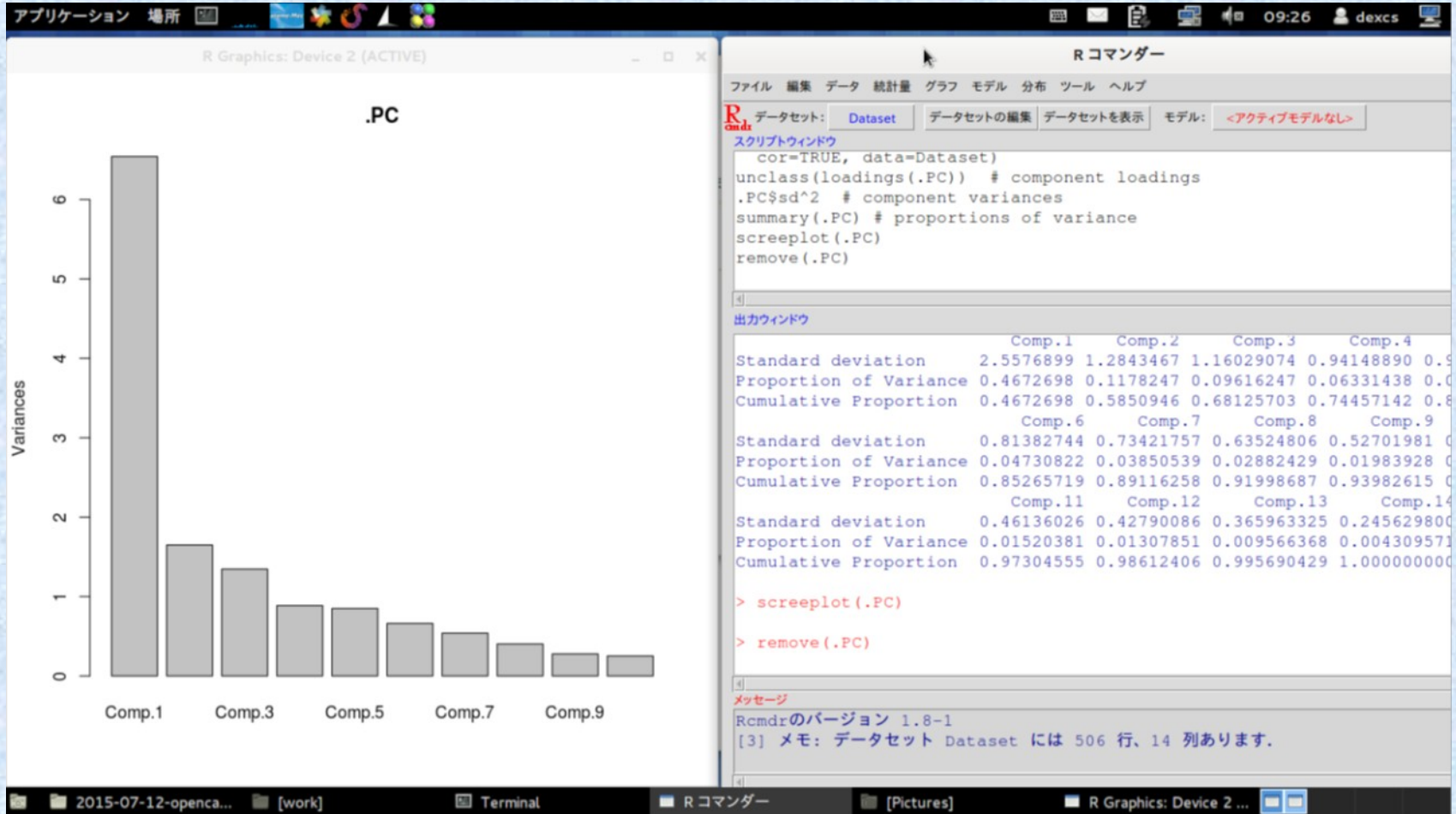


デンドログラム

無料のデータ結果可視化・統計分析ツール

統計解析ソフトRによる分析②

次元削減の例



主成分分析

RのGUI “Rコマンダー”や“EZR”が 出ているのでコマンドを知らなくても問題ない！

Dakotaの計算結果をR自己組織化マップ KOHONENを使って分析①

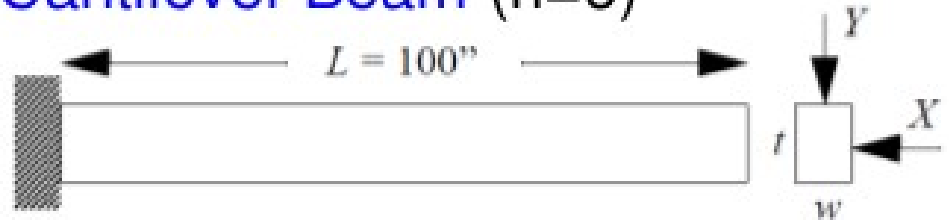
- 実際にDakotaの解析結果を用いて自己組織化マップを作成してみる。
- 簡単な事例としてDakotaの片持ち梁の事例を用いる。チュートリアルの中の“cantilever_optimization.in”のファイルを用いる
- 入力ファイルの中身は次ページ参照：このファイルだけ見ても何をやっているかわからない。
- 詳細はDAKOTAのHPで公開されているプレゼンテーション資料を参照する

Dakotaの計算結果をR自己組織化マップ

KOHONENを使って分析②

- environment
 - tabular_data
 - graphics
 -
- method
 - conmin_mfd
 - convergence_tolerance 1.0e-4
 - constraint_tolerance 1.0e-1
 -
- model
 - single
 -
- variables
 - continuous_design = 2
 - upper_bounds 4.0 4.0
 - initial_point 2.5 2.5
 - lower_bounds 1.0 1.0
 - descriptors 'w' 't'
 - continuous_state = 4
 - initial_state 40000. 29.E+6 500. 1000.
 - descriptors 'R' 'E' 'X' 'Y'
 -
- interface
 - direct
 - analysis_driver = 'mod_cantilever'
 -
- responses
 - objective_functions = 1
 - nonlinear_inequality_constraints = 2
 - descriptors = 'area' 'stress' 'displacement'
 - analytic_gradients
 - no_hessians

Cantilever Beam (n=6)



$$S = \frac{600}{wt^2} Y + \frac{600}{w^2 t} X \leq R$$

$$D = \frac{4L^3}{Ewt} \sqrt{\left(\frac{Y}{t^2}\right)^2 + \left(\frac{X}{w^2}\right)^2} \leq D_0$$

$$w, t, R, E, X, Y: U[1, 10], U[1, 10], \\ N(4E4, 2E3), N(2.9E7, 1.45E6), \\ N(500, 100), N(1E3, 100); D_0 = 2.2535''$$

梁の幅 w と高さ t をパラメータとして S (stress), D (displacement)を最適化(最小化)する例題、このままだとデータ数が少ないので、少しinputファイルを修正する

Dakotaの計算結果をR自己組織化マップ KOHONENを使って分析③

```
• environment
• tabular_data
• graphics
•
• method
• sampling
• sample_type random
• samples = 100 #s0,#s1
• # conmin_mfd
• # convergence_tolerance 1.0e-4
• # constraint_tolerance 1.0e-1
•
• model
• single
•
• variables
• continuous_design = 4
• upper_bounds 4.0 4.0 500.0 1000.0
• initial_point 2.5 2.5 500.0 1000.0
• lower_bounds 1.0 1.0 100.0 200.0
• descriptors 'w' 't' 'X' 'Y'
• continuous_state = 2
• initial_state 40000. 29.E+6
• descriptors 'R' 'E'
```

In file を左のように一部変更し、dakotaを実行する

サンプル数をランダムサンプリングにて100個

w, t以外にX,Yも変数に

コマンド実行方法

\$dakota -i cantilever_opt-pstudy2.in
実行フォルダに”dakota_tabular.dat”

Dakotaの計算結果をR自己組織化マップ

KOHONENを使って分析④

- “dakota_tabular.dat”というファイルが出来ているのでこれをcsv形式にして、Rに渡す

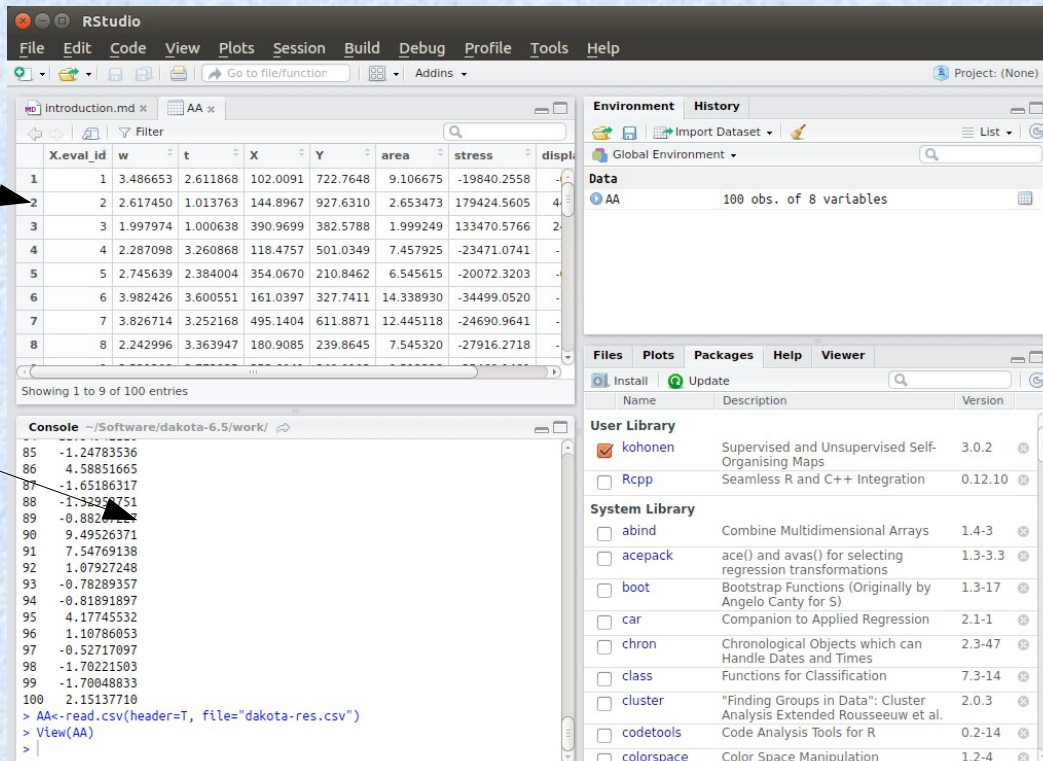
%eval	id	interface	w	t	X	Y	R	E
		area	stress	displacement				
1		NO_ID	3.486652624	2.611867517	102.0090524	722.7647891		
	40000	29000000	9.106674731	-19840.25575	-0.6437638074			
2		NO_ID	2.617449631	1.013762929	144.8966917	927.6310207		
	40000	29000000	2.653473406	179424.5605	44.67848181			
3		NO_ID	1.997974204	1.000637929	390.9698555	382.5788295		
	40000	29000000	1.99924877	133470.5766	24.95975577			
4		NO_ID	2.287097919	3.260868435	118.47566	501.0348817		
	40000	29000000	7.457925414	-23471.07413	-1.286594139			
5		NO_ID	2.745639049	2.384004032	354.0670075	210.846195		
	40000	29000000	6.545614561	-20072.32029	-0.9922878667			
6		NO_ID	3.98242615	3.600551392	161.0397498	327.7410874		
	40000	29000000	14.33893002	-34499.05199	-1.991431904			
7		NO_ID	3.826714145	3.252168185	495.1404353	611.8871022		
	40000	29000000	12.445118	～以下略～				

Dakotaの計算結果をR自己組織化マップ KOHONENを使って分析⑤

- R-studioを起動して、ファイルを読み込む
- コマンド画面から
`library("kohonen", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.2")`
`AA <- read.csv(header=T, file="dakota-res.csv")`

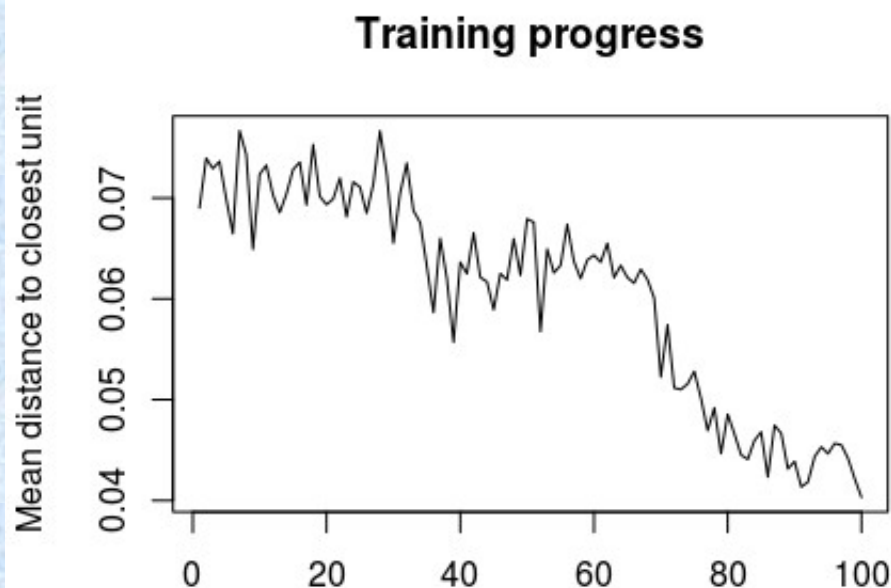
View(AA)
で変数の
中身が表示

コマンド
画面



Dakotaの計算結果をR自己組織化マップ KOHONENを使って分析⑥

- kohonenを使って自己組織化MAPを作成
set.seed(5)
datasom1 <- som(scale(AA),grid=somgrid(5,5))
plot(datasom1, type="changes")



Type =changes に
て収束履歴が表示

Dakotaの計算結果をR自己組織化マップ

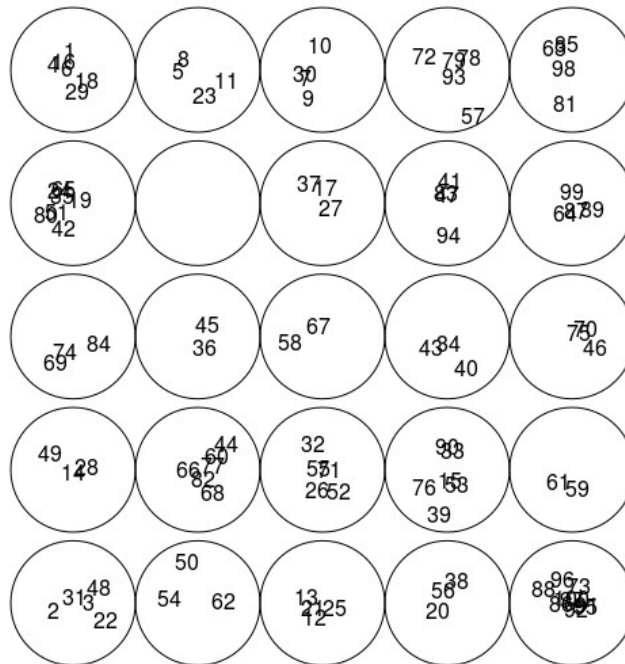
KOHONENを使って分析⑦

- ケース番号にて分類を表示

```
plot(datasom1, type="mapping", labels = AA[,1])
```

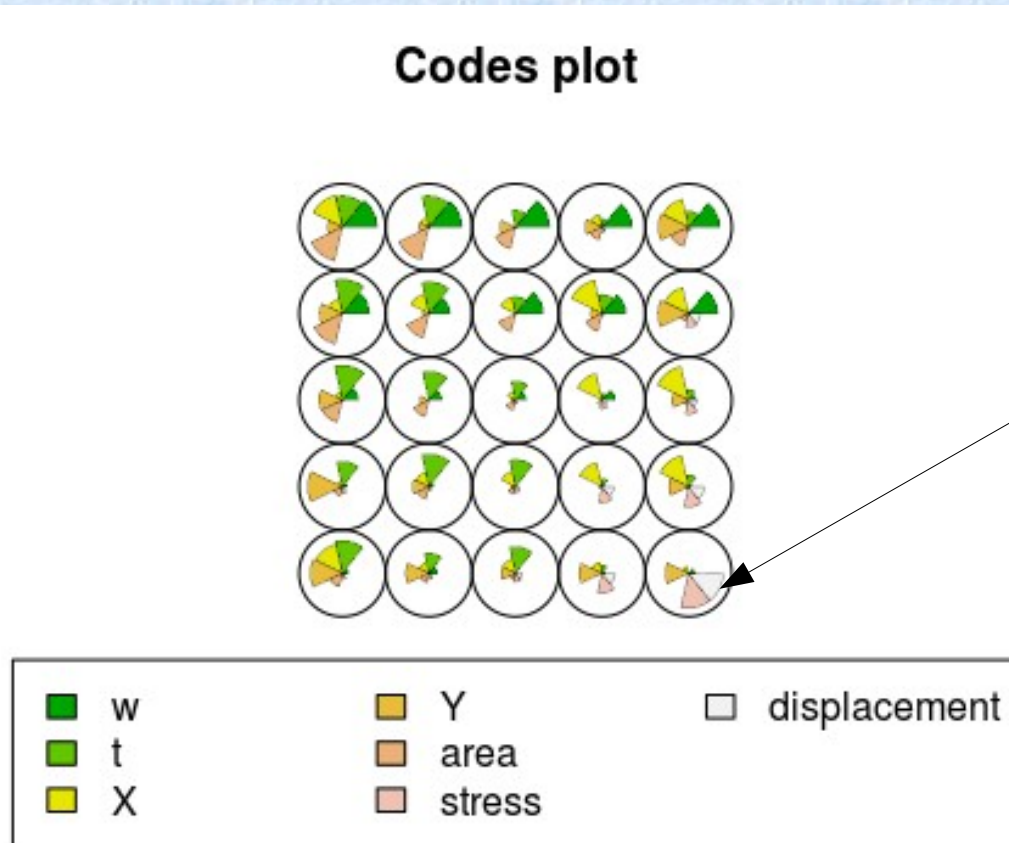
Plot Zoom

Mapping plot



Dakotaの計算結果をR自己組織化マップ KOHONENを使って分析⑧

- このままだと内容がわからないのでコードPlotする
- ```
datasom2 <- som(scale(AA[,2:8]),grid=somgrid(5,5))
plot(datasom2, type="codes")
```



変位が大きい?

分類はされても物理的な解釈まではしてくれないので、ユーザーが理由は考える!

# まとめ

- ・無料で使える最適化ツールDakotaとRを使って最適化、結果分析を行う一連の流れについて調査
- ・オープンソースの統計分析ツールは充実度が高く、両者を組み合わせることで数百万円はする商用CAE最適化ツールと同様のことができそうであることを確認した
- ・関連資料を無料で公開してくれている有志のみなさんに感謝します。



# OpenCAE関連イベント開催告知

5月20日 (土曜日)

オープンCAE勉強会@関東・流体など

6月23日 (金曜日)

オープンCAE学会総会・講習会@東京(ルーテル市ヶ谷)

12月 7日(木)□ 9日(土)

オープンCAE学会シンポジウム@名古屋大学