



# 統計解析ツールと数理計画法 パッケージ NUOPT の関係

(株)数理システム

# 数理計画問題とは

■ 変数 -- 決定したい値及び選択肢

$$x_i$$

■ 目的関数 -- 最小化・最大化したい関数

$$\sum_i x_i$$

■ 制約式 -- 条件等を数式化したもの

$$\sum_i a_{ij} x_i \leq b_j$$

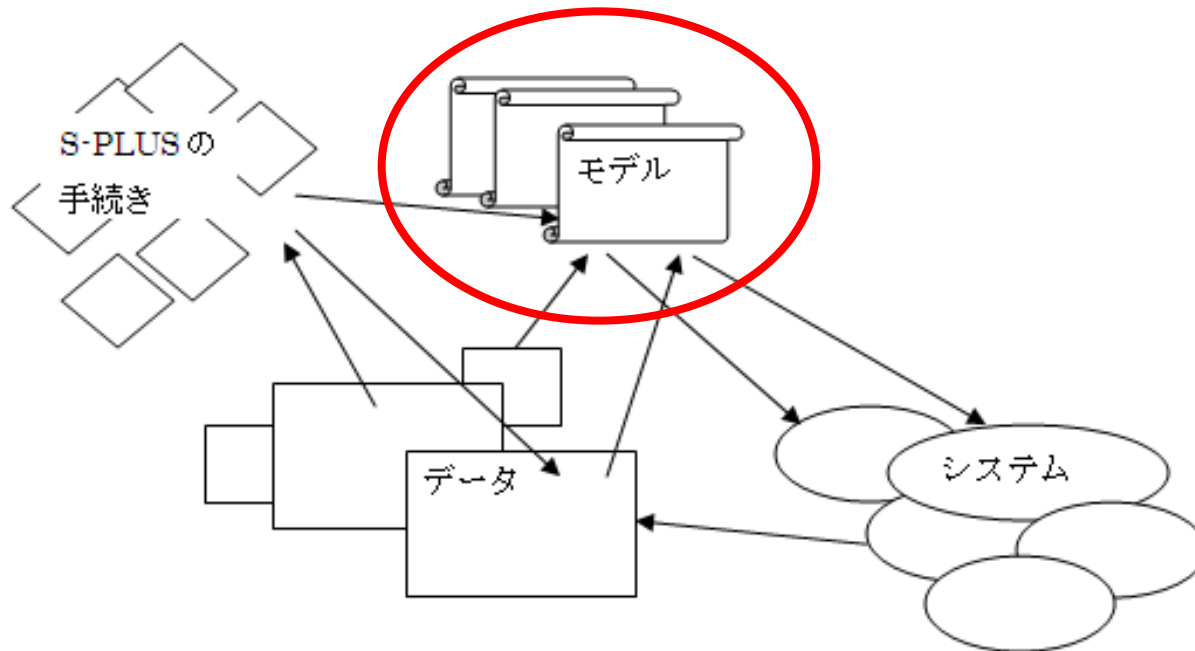
# 数理計画問題の種類

- 線形計画問題
- 二次計画問題
- 非線形計画問題
- 半正定値計画問題
- 混合整数計画問題
- 制約充足問題

**NUOPT は全てに対応**

# S-PLUS/R (統計パッケージ) と NUOPT (数理計画法パッケージ) の大きな違い

- NUOPTは
  - 数理モデルの記述が**必須**⇒**導入コスト**



# S+NUOPT / R 関係機能では

- 典型的なデータとモデル付き  
⇒ 典型的なモデルがすぐに使える

例題	内容	モデル名	適合するデータの例	章番号
重回帰	データに特化	Nlsfit	freeny.x/y	2.1
	汎用	Nlsfit.gen	freeny.x/y,stack.x/loss	2.7.1
	パラメータの線形制約付き	Nlsfit.eq	freeny.x/y,stack.x/loss	2.8
	パラメータの選択付き	Nlsfit.int	freeny.x/y,stack.x/loss	2.9
キャッシュフローマッチング	資金調達の問題(条件式の応用)	Cashflow	Cashflow.flow/bf	3.1
半正定値計画導入	最小固有値の取得	MinLambda	var(air)	3.2
集合の分割	1指標2分割	Half	state.x77	3.3.1
	多指標2分割	Half2	state.x77	3.3.2
	1指標多分割	Partition	state.x77	3.3.3
ポートフォリオ最適化	マルコビッツモデル基本	Marko	R.60x4	5.1
	分散	MinVar	R.8000x5	5.2.1
	絶対偏差	MinMad	R.8000x5	5.2.2
	絶対偏差(abs())を使ったモデル	MinMadNL	R.8000x5	
	1次の下方部分積率	MinLPM1	R.8000x5	5.2.3
	CVaR	MinCVaR	R.8000x5	5.2.4
	コンパクト分解	MinVar	R.60x1000	5.3
	Maximum Drawdown	MinMaxDD	R.521x95	
	Sharpe Ratio最大化	Sharpe	R.60x200	
	Sharpe Ratio最大化(QP)	Sharpe.qp	R.60x200	
離散最適化とポートフォリオ	端株処理	RoundLot	RoundLot.unit	5.4
	銘柄のグルーピング	Basket	Basket.flow/fhigh/fbar/W	
半正定値計画の応用	ロバスト最適化	Robust	Robust.sigU/sigL/mu	6.2
	相関行列の成形	Cormat	Cormat.A	6.1
非線形回帰	イールドカーブの推定	Yield	Yield.telem/term/price	7.1
	格付け推移行列の推定	Rating	Rating.Q0	7.2
	ロジスティック回帰	LogReg	LogReg.X/t/test.X/test.t	

# S+NUOPTの構成

## NUOPTによる最適化計算を行う S-PLUS アドオン



### S-PLUS

```
S-PLUS [commands]
Iteration begin
  next.iteration .... 1.4e-007
Iteration end

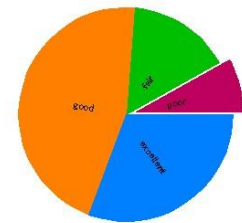
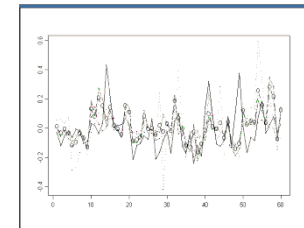
STATUS
VALUE_OF_OBJECTIVE          798.089924
VALUE_OF_CONSTRAINT         0
VALUE_OF_LOWER_BOUND        0
VALUE_OF_UPPER_BOUND        0
ACQUISITION_COST            1.0542767e-007
RESIDUAL                     0.00
LAPSED_TIME(sec.)           0.00

+ 最適化結果の取得
r <- c(current.iter, obj)
c(1) 798
c(2) model/root

+ 問題の確認
obj <- get.object.value(model)
Evaluate the model.model[1, ...] obj
Solving objective: 1.0 "near" obj?
Solving constraint: (2/3)
Solving constraint: (4/5)
Solving constraint: (4/5)
Solving constraint: (4/5)

+ 実行
eval <- get.eval(model)
RMPFF 11.1.0a (RMP/P/SP/STEP module)
with: META-HEURISTICS where "meta"/"framer"
Copyright (C) 1991-2000 Mathemtical Systems Inc.
NUMBER_OF_VARIABLES         1
NUMBER_OF_FUNCTIONS         1
PROBLEM_TYPE                 MINIMIZATION
METHOD                       GRADIENT_DESCENT
Solve process begin[.....] Gradient method
Iteration begin
  next.iteration .... 1.4e-007
Iteration end
STATUS
VALUE_OF_OBJECTIVE          798.089924
VALUE_OF_CONSTRAINT         0
VALUE_OF_LOWER_BOUND        0
VALUE_OF_UPPER_BOUND        0
ACQUISITION_COST            1.0542767e-007
RESIDUAL                     0.00
LAPSED_TIME(sec.)           0.00

+ 実行結果の取得
r <- c(current.iter, obj)
c(1) 798
c(2) model/root
```



### NUOPT



# S+NUOPT 実行例：油田運転計画

1週間にかかる, 油田A,Bの運転コストを最小化したい



油田 A

1日運転すると

- ・コスト 180
- ・重油 6, ガス 4

制約条件・生産量ノルマ

- 重油 12 以上
- ガス 24 以上

1日運転すると

- ・コスト 160
- ・重油 1, ガス 6



油田 B

# S+NUOPT 記述例

- 変数

- $x$ : 油田 A の運転日数
- $y$ : 油田 B の運転日数

- 目的関数(最小化)

- $180x + 160y$

- 制約式

- $6x + y \geq 12$
- $4x + 6y \geq 24$
- $x \geq 0$
- $y \geq 0$

```
oil.model <- function(){  
# 変数  
  x <- Variable()  
  y <- Variable()  
  
# 目的関数  
  obj <- Objective()  
  obj ~ 180*x + 160*y  
  
# 制約式  
  6*x + y >= 12  
  4*x + 6*y >= 24  
  x >= 0  
  y >= 0  
}
```



# S+NUOPT 記述例

数理計画問題 oil.model に対して最適化計算を行う。

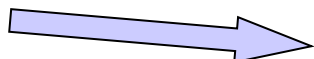
```
module(nuopt)

# 問題の展開・実行
sys <- System(oil.model)
sol <- solve(sys)

# 解情報の取得
r <- current(sys, x)
```

# Rnuoptの構成

NUOPT による最適化計算を行う R パッケージ

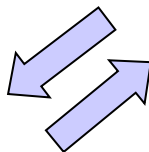


```
RGU - [R Console]
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

Rnuopt 1.12.0 (changeset: 659:0ba326b887e9) for R 2.11.1 on Microsoft Windows
, Copyright (C) 2010 Mathematical Systems Inc. All rights reserved.

次のパッケージを付け加えます: 'Rnuopt'

The following object(s) are masked from 'package:base':
--
> S <- Set()
> i <- Element(set=S)
> S ~ "1 2 3"
> S
{ 1 2 3 }
> i
1
Element was defined by <Set> S
> x <- Variable(index=1)
> x
1 2 3
0
attr(,"indexes")
[1] ""
> |
```



NUOPT



# Rnuropt の実行例

## ポートフォリオ最適化・マルコヴィッツモデル

```
library(Rnuropt)

# 展開・実行
sys <- System(MinVar, R.60x1000)
sol <- solve(sys)

# 解の取得
x <- as.array(current(sys,x))

# 図示
eps <- 1e-2
pie(x[x>eps])
```

# 半正定値ロジスティック回帰

- ロジスティック回帰の拡張
- ロジットを凸二次関数まで拡張

目的関数 
$$\prod_{k \in A} \frac{1}{1 + \exp(z_k)} \prod_{k \in B} \frac{\exp(z_k)}{1 + \exp(z_k)}$$

制約式 
$$z_k = \alpha + \beta^T x_k + \frac{1}{2} x_k^T Q x_k$$

$$Q \succeq 0$$

半正定値制約

# 半正定値ロジスティック回帰

```
LogSemi.model <- function(X.d, t.d)
{
  ...
  Q <- SymmetricMatrix(dprod(i, j))
  Q[i, j, i >= j] ~ q[i, j]

  Q >= 0

  z[l] == Sum(q[i,j]*X[l,i]*X[l,j], i,j)+ Sum(a[i]*X[l,i],i)+a0
  y[l] ~ exp(z[l]) / (1+exp(z[l]))

  mle <- Objective(type="maximize")
  mle ~ Sum(t[l]*log(y[l])+(1-t[l])*log(1-y[l]),l)
  q[i,j,i<j] == q[j,i]
}
```

# S+NUOPT/Rnuoptでは

- 豊富なグラフ機能
- 統計演算

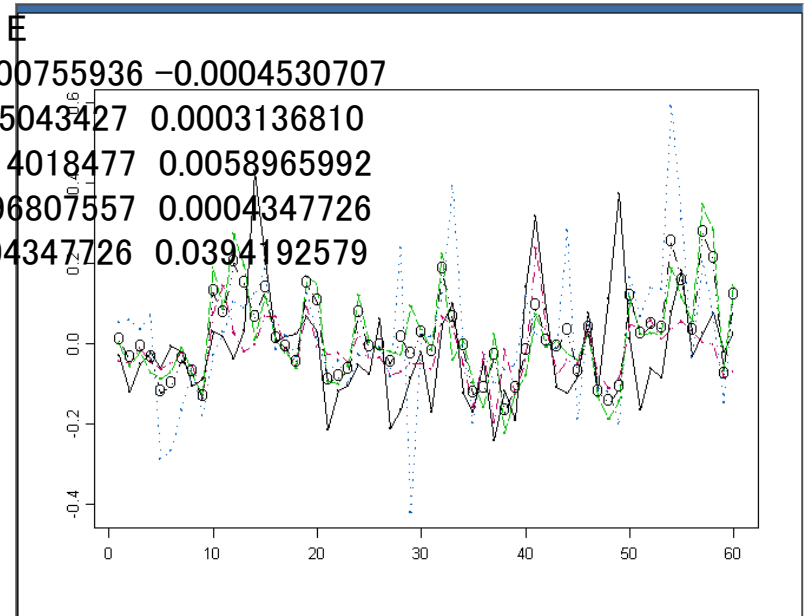
```
> var(R.8000x5)
```

	A	B	C	D	E
A	0.0612951061	0.0038057910	-0.0010892558	-0.0000755936	-0.0004530707
B	0.0038057910	0.1177609216	0.0006516297	0.0005043427	0.0003136810
C	-0.0010892558	0.0006516297	0.0458277805	0.0014018477	0.0058965992
D	-0.0000755936	0.0005043427	0.0014018477	0.0096807557	0.0004347726
E	-0.0004530707	0.0003136810	0.0058965992	0.0004347726	0.0394192579

- データ解析

- データハンドリング

```
> x[x>eps]
```



# S + NUOPT/Rnuoptの可能性

- 金融工学プラットフォーム
  - ポートフォリオモデル
  - 非線形回帰
- アドバンストな統計解析
  - 判別分析
  - 変数選択
  - 集合分割