演習2
中身を理解しよう

実行の流れ: まとめ

必要なファイル
- forecast.ss
- test1.ctl
- test1.dat
- starter.ss
- ss3.exe

コマンドプロンプトで
> ss3.exe

結果のファイル
- ss3.par
- ss3.std
- Report.sso
- CompReport.sso
- …

Rで
> allplot.ss()

Rによる可視化
- out.pdf

これから個々のファイルの中身を
説明してきます

実行に必要なファイル

- ss3.exe
  - 実行ファイル（プログラム）
  - Windows (64bit， 32bit), MAC, Linux (64bit, 32bit)

- starter.ss ⊶ データファイル・設定ファイル名の指定等
- data1.ss ⊶ データ
- control1.ss ⊶ 資源評価の設定
- forecast.ss ⊶ 将来予測の設定

注意事項

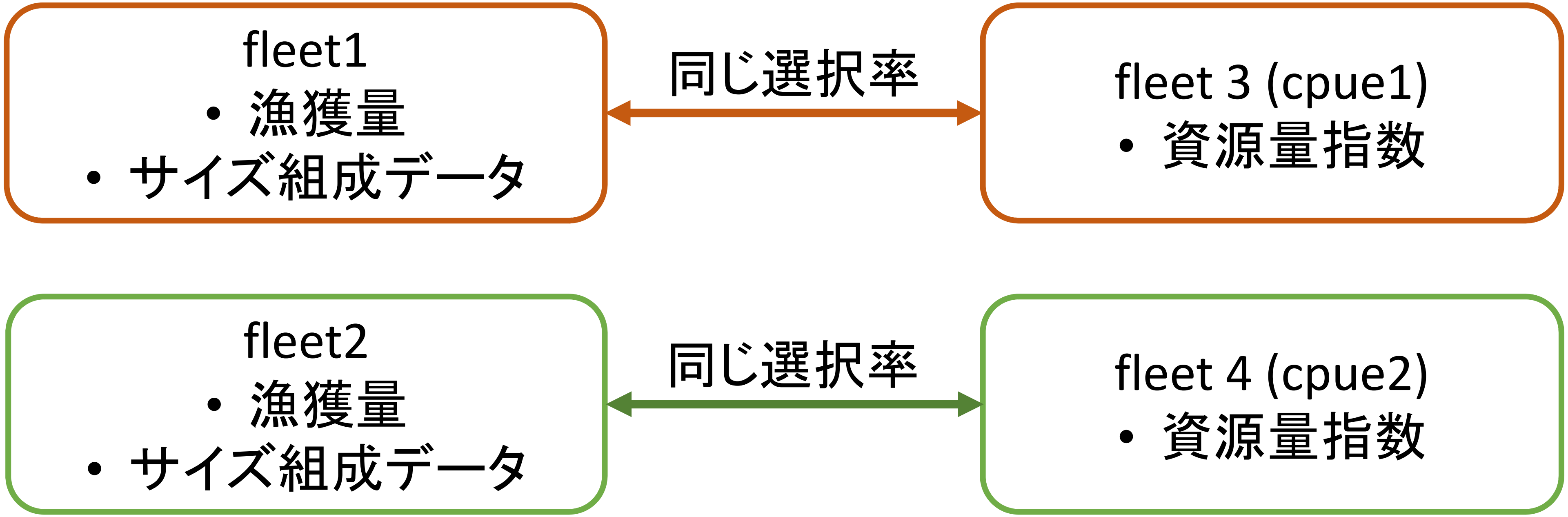
- コメントアウトは　#
- 数字のあとにコメントアウトする場合は，#の前に「必ず」スペースを入れてください
- これらは，テキストエディタ（例; Mery）で編集します

全体的な設定のファイル：starter.ss

- データファイルや設定ファイルの名前をここで指定
- このファイルの名前は変更できない

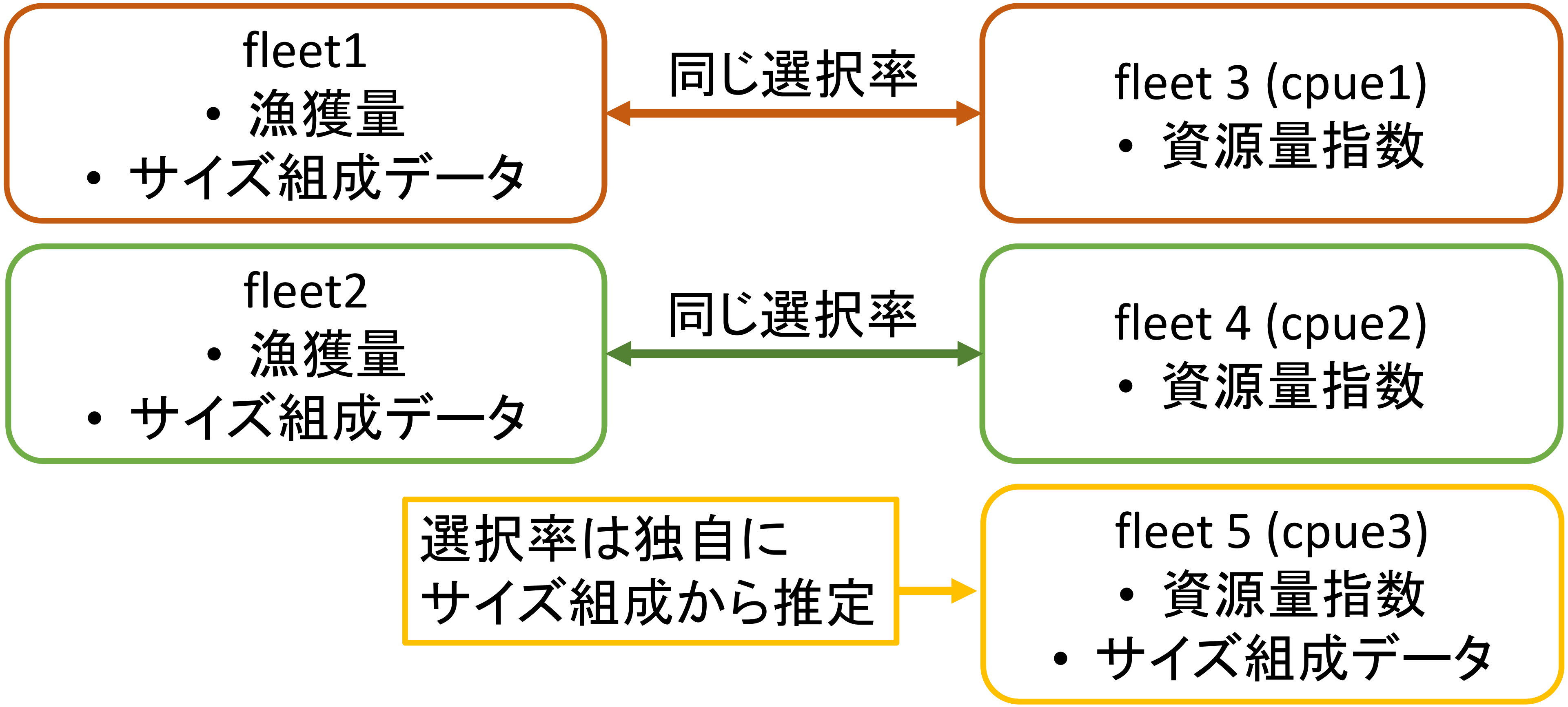
データファイル：data1.ss

- 選択率が異なる2種類の漁業 (fleet) があると仮定



漁業の定義のバリエーション

- 漁獲しない調査データから資源量指数を得る場合



コントロールファイル：control1.ss

- 生物パラメータ
- 親子関係
- 初期の漁獲率
- 漁獲効率
- 選択率の設定
- ラムダ・分散のコントロール

将来予測ファイル：forecast.ss

- こちらも，starter.ssと同様に，ファイル名の変更はできない
- あんまり解説はしません（明日，竹内さん）
- 現在のところ，沿岸資源でやられているような stochastic simulationはできない
- かわりに，MCMCを使う

トピック① パラメータ推定の方法

- Controlファイルでは，推定するパラメータ1つにつき, ↓のように設定します

#_生物パラメータ（自然死亡・成熟・成長）

#_LO HI INIT PRIOR PR_type SD PHASE env-var use_dev dev_minyr dev_maxyr dev_stddev Bloc

0.1 0.6 0.25 0.25 0 1000 -3 0 0 0 0 0 0.5 0 0 # M

5 100 20 20 0 1000 -3 0 0 0 0 0 0.5 0 0 # Aminのときの体長(Lmin)

30 200 100 100 0 1000 -4 0 0 0 0 0 0.5 0 0 # Amaxのときの体長(Lmax)

0.01 0.65 0.15 0.15 0 1000 -4 0 0 0 0 0 0.5 0 0 # VB式のK

- 表で整理すると，，，

#_LO	HI	INIT	PRIOR	PR_type	SD	PHASE	env-var	use_dev	dev_minyr	dev_maxyr	dev_stddev	Block	Block_Fxn		
0.1	0.6	0.25	0.25	0	1000	-3	0	0	0	0	0.5	0	0	#	M
5	100	20	20	0	1000	-3	0	0	0	0	0.5	0	0	#	Aminのときの体長(Lmin)
30	200	100	100	0	1000	-4	0	0	0	0	0.5	0	0	#	Amaxのときの体長(Lmax)
0.01	0.65	0.15	0.15	0	1000	-4	0	0	0	0	0.5	0	0	#	VB式のK

パラメータの
下限・上限・初期値

フェーズ（負の場合：そのパラメータを推定しない，
正の場合：数字の順番で推定される）

#_LO	HI	INIT	PRIOR	PR_type	SD	PHASE	env-var	use_dev	dev_minyr	dev_maxyr	dev_stddev	Block	Block_Fxn		
0.1	0.6	0.25	0.25	0	1000	-3	0	0	0	0	0.5	0	0	#	M
5	100	20	20	0	1000	-3	0	0	0	0	0.5	0	0	#	Aminのときの体長(Lmin)
30	200	100	100	0	1000	-4	0	0	0	0	0.5	0	0	#	Amaxのときの体長(Lmax)
0.01	0.65	0.15	0.15	0	1000	-4	0	0	0	0	0.5	0	0	#	VB式のK

フェーズphaseの概念

- phase>0なら推定する。phase<0なら推定しない（パラメータの固定・推定が簡単にコントロールできる）
- phase>0の場合，phaseが小さいパラメータから順に推定し，だんだん推定するパラメータを増やしていく
- 順番が来ていないパラメータについては，初期値で固定される

- 全体的な挙動をコントロールする大事なパラメータから徐々に推定していくことによって計算が安定する

- 一方，phaseの組み方によって推定結果が変わることも

- 自然死亡係数の場合，phaseが負に設定されていて，初期値が0.25なので，M=0.25で固定していることになる

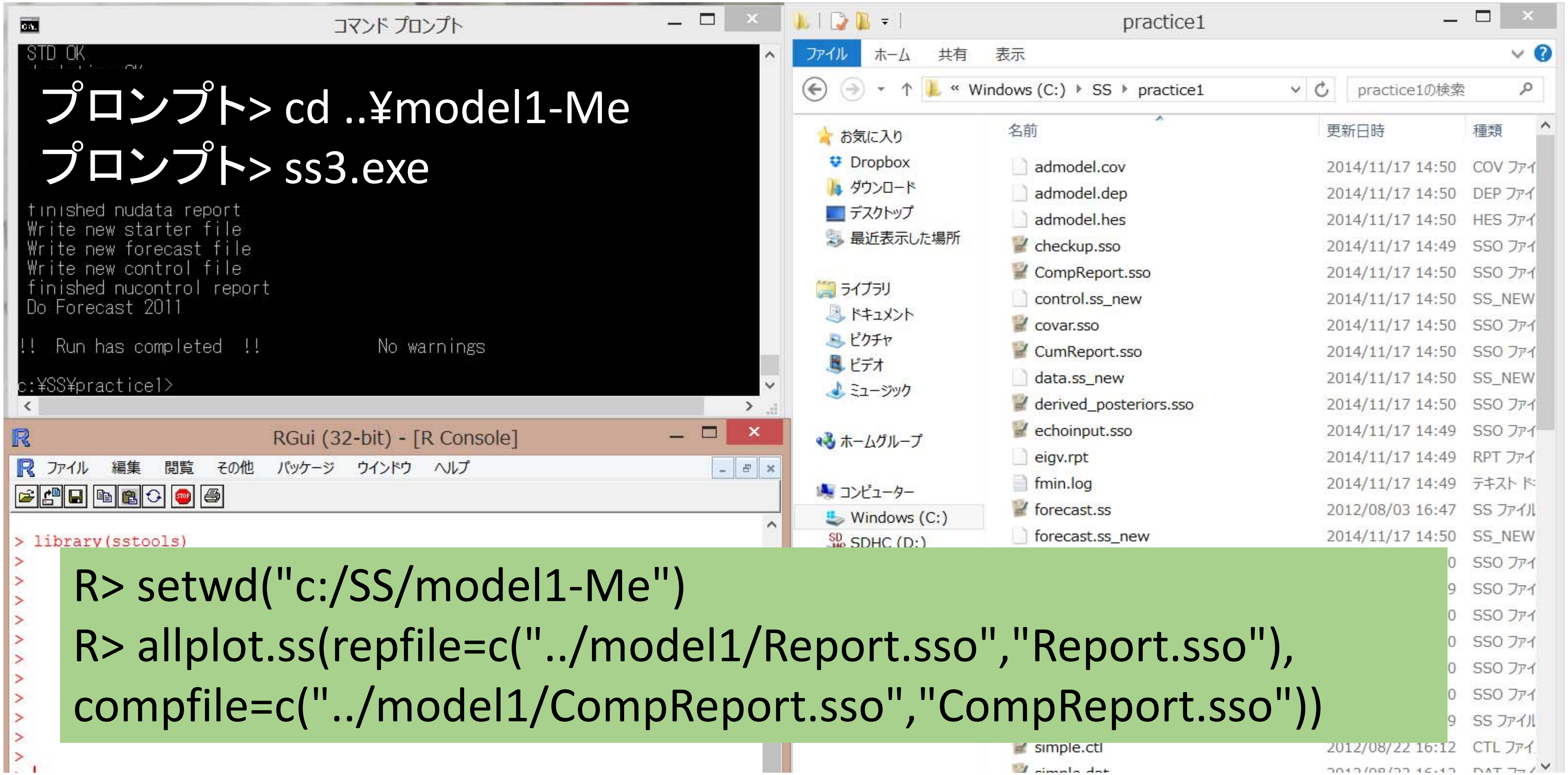
#_LO	HI	INIT	PRIOR	PR_type	SD	PHASE	env-var	use_dev	dev_minyr	dev_maxyr	dev_stddev	Block	Block_Fxn		
0.1	0.6	0.25	0.25	0	1000	-3	0	0	0	0	0.5	0	0	#	M
5	100	20	20	0	1000	-3	0	0	0	0	0.5	0	0	#	Aminのときの体長(Lmin)
30	200	100	100	0	1000	-4	0	0	0	0	0.5	0	0	#	Amaxのときの体長(Lmax)
0.01	0.65	0.15	0.15	0	1000	-4	0	0	0	0	0.5	0	0	#	VB式のK

演習：自然死亡係数を推定してみよう

- model1 フォルダをコピーして，model1-Meフォルダを作る
- control1.ssをテキストエディタで開いてフェーズを正の値に(3とか)

```
#_生物パラメータ（自然死亡・成熟・成長）
#_LO HI INIT PRIOR PR_type SD PHASE env-var use_dev dev_minyr dev_maxyr dev
0.1 0.6 0.25 0.25 0 1000 -3 0 0 0 0 0.5 0 0 # M
5 100 20 20 0 1000 -3 0 0 0 0 0.5 0 0 # Aminのときの体長(Lmin)
30 200 100 100 0 1000 -4 0 0 0 0 0.5 0 0 # Amaxのときの体長(Lmax)
0.01 0.65 0.15 0.15 0 1000 -4 0 0 0 0 0.5 0 0 # VB式のK
```

Model1-Meで走らせる&結果の比較

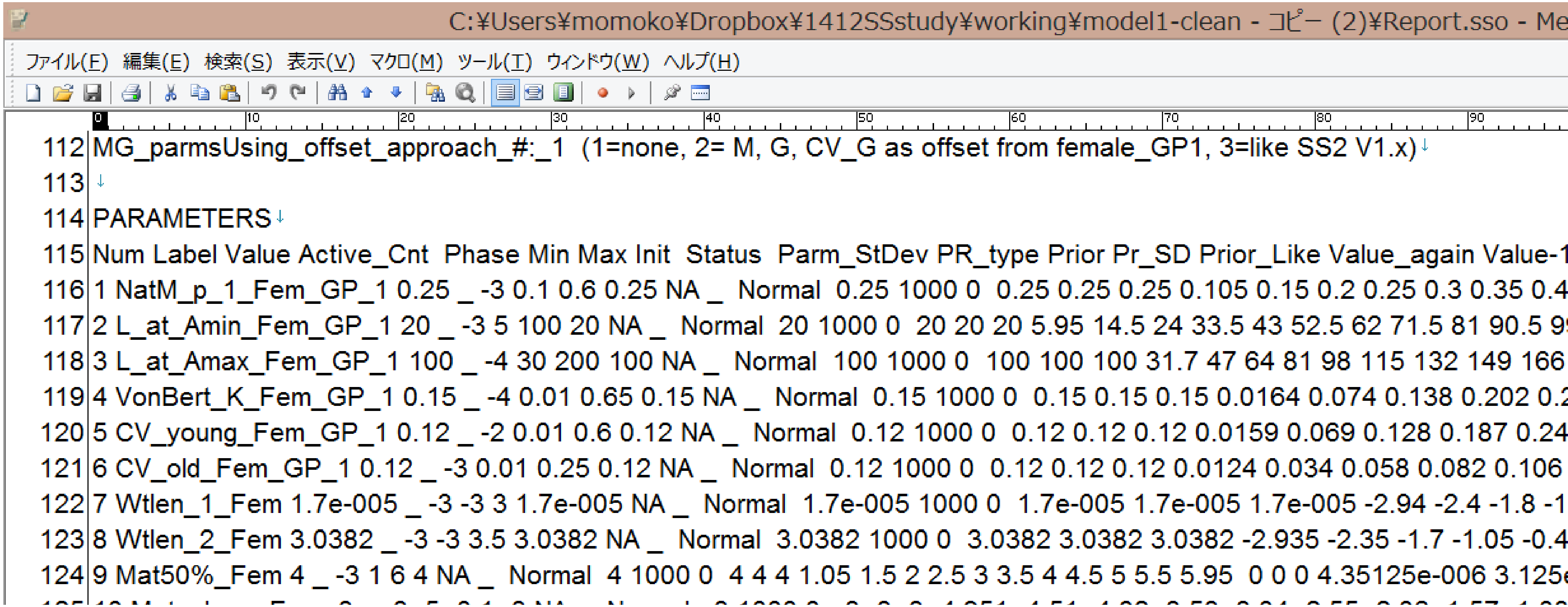



```
プロンプト> cd ..¥model1-Me
プロンプト> ss3.exe
```

```
R> setwd("c:/SS/model1-Me")
R> allplot.ss(repfile=c("../model1/Report.sso","Report.sso"),
compfile=c("../model1/CompReport.sso","CompReport.sso"))
```

実際の計算結果の確認

- 細かい数字はReport.ssoに出力されているのでそれを確認




```
MG_parmsUsing_offset_approach_#:_1  (1=none, 2= M, G, CV_G as offset from female_GP1, 3=like SS2 V1.x)

PARAMETERS
Num Label Value Active_Cnt  Phase Min Max Init  Status  Parm_StDev PR_type Prior Pr_SD Prior_Like Value_again Value-1
1 NatM_p_1_Fem_GP_1 0.25 _ -3 0.1 0.6 0.25 NA _  Normal  0.25 1000 0  0.25 0.25 0.25 0.105 0.15 0.2 0.25 0.3 0.35 0.4
2 L_at_Amin_Fem_GP_1 20 _ -3 5 100 20 NA _  Normal  20 1000 0  20 20 20 5.95 14.5 24 33.5 43 52.5 62 71.5 81 90.5 9
3 L_at_Amax_Fem_GP_1 100 _ -4 30 200 100 NA _  Normal  100 1000 0  100 100 100 31.7 47 64 81 98 115 132 149 166
4 VonBert_K_Fem_GP_1 0.15 _ -4 0.01 0.65 0.15 NA _  Normal  0.15 1000 0  0.15 0.15 0.15 0.0164 0.074 0.138 0.202 0.2
5 CV_young_Fem_GP_1 0.12 _ -2 0.01 0.6 0.12 NA _  Normal  0.12 1000 0  0.12 0.12 0.12 0.0159 0.069 0.128 0.187 0.24
6 CV_old_Fem_GP_1 0.12 _ -3 0.01 0.25 0.12 NA _  Normal  0.12 1000 0  0.12 0.12 0.12 0.0124 0.034 0.058 0.082 0.106
7 Wtlen_1_Fem 1.7e-005 _ -3 -3 3 1.7e-005 NA _  Normal  1.7e-005 1000 0  1.7e-005 1.7e-005 1.7e-005 -2.94 -2.4 -1.8 -1
8 Wtlen_2_Fem 3.0382 _ -3 -3 3.5 3.0382 NA _  Normal  3.0382 1000 0  3.0382 3.0382 3.0382 -2.935 -2.35 -1.7 -1.05 -0.4
9 Mat50%_Fem 4 _ -3 1 6 4 NA _  Normal  4 1000 0  4 4 4 1.05 1.5 2 2.5 3 3.5 4 4.5 5 5.5 5.95  0 0 0 4.35125e-006 3.125e
```

トピック② 様々な選択率の設定

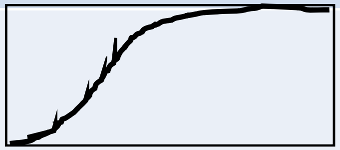
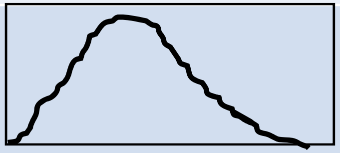
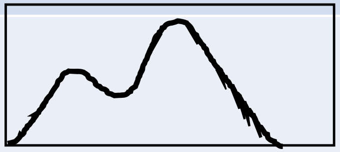
- コントロールファイル

```
#_体長に対する選択率の設定
#_Pattern ___ Male Special
0 0 0 0 # F1
0 0 0 0 # F2
0 0 0 0 # CPUE1
0 0 0 0 # CPUE2
```

漁業が4つなので，4列分の設定

```
#_年齢に対する選択率の設定
#_Pattern ___ Male Special
20 0 0 0 # F1
20 0 0 0 # F2
15 0 0 1 # CPUE1
15 0 0 2 # CPUE2
```

• よく使われる選択率のパターン

体長に対する選択率（推定パラメータの数）	年齢に対する選択率（推定パラメータの数）	説明
0 (0)	10 (0)	全ての体長（年齢）で選択率を1とする。パラメータは必要ない
1 (2)	12 (2)	ロジスティック 
24 (6)	20 (6)	Double-normal いくつか同じような形のものがあるがこれが今のところのおすすめ。logisticも兼ねられる。 
27 (3+2N)		キュービックスプライン。 ノードの数(N)の設定等難しい。 
15	15	他の漁業と同じ選択率を用いる（ミラー）（この場合，specialのところに，ミラーする漁業の番号を入れる）

- コントロールファイル

#_体長に対する選択率の設定
#_Pattern ___ Male Special
0 0 0 0 # F1
0 0 0 0 # F2
0 0 0 0 # CPUE1
0 0 0 0 # CPUE2

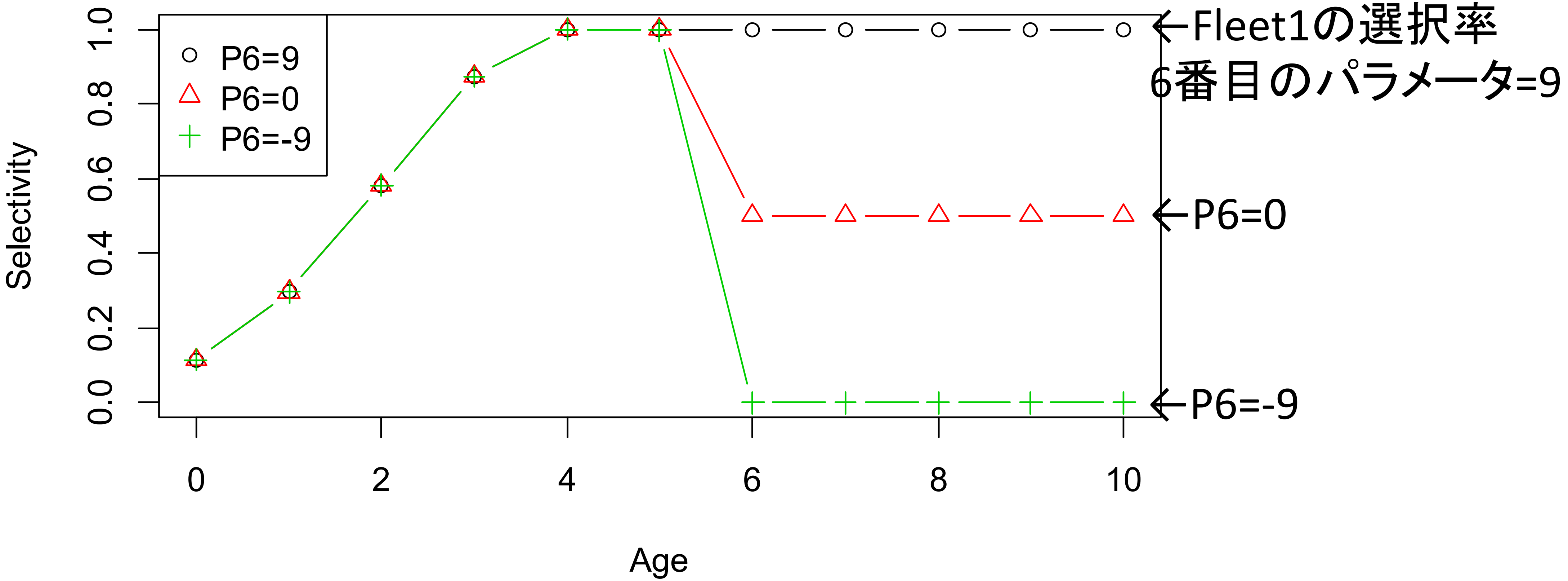
#_年齢に対する選択率の設定
#_Pattern ___ Male Special
20 0 0 0 # F1
20 0 0 0 # F2
15 0 0 1 # CPUE1
15 0 0 2 # CPUE2

F1とF2はdouble normal

CPUE1はF1を，CPUE2はF2をミラー

演習: fleet1の選択率の設定を変えてみる

6番目のパラメータを推定したらどうなるか？　(model1-P6)



トピック③　分散の調整（variance adjustment）とλの設定

- パラメータの推定方法→統合尤度の最大化

- 統合尤度の式＝各尤度の和

$$\text{統合尤度} = \lambda_2 L_{CPUE1} + \lambda_2 L_{CPUE2} + \lambda_3 L_{length1} + \lambda_4 L_{length2}$$

資源量指数の尤度

$$\mathcal{L}_{1f} = 0.5 \sum_t \left(\frac{\ln(G_{tf}) - \ln(\hat{G}_{tf})}{\sigma_{1tf}} \right)^2$$

データファイルで入力したCPUEのCV

体長組成の尤度

$$\mathcal{L}_{4f} = \sum_t \sum_m \sum_{\gamma=1}^{A_\gamma} n_{1tf\gamma m} \sum_{l=1}^{A_l} p_{1tfl\gamma m} \ln(p_{1tfl\gamma m} / \hat{p}_{1tfl\gamma m})$$

データファイルで入力した体長組成の
サンプルサイズ

λ，CPUEのCV，サンプルサイズを調整することで，データの重みづけを変えることができる

- コントロールファイルの最後の部分では，λ，CPUEのCV，サンプルサイズの調整ができます

```
1 #_分散（入力したCVやサンプルサイズ）を調整するための設定
#_fleet: 1 2 3 4
 0 0 0 0 #_add_to_survey_CV（資源量指数のCVに足す値）
 0 0 0 0 #_add_to_discard_stddev
 0 0 0 0 #_add_to_bodywt_CV
 1 1 1 1 #_mult_by_lencomp_N（体長組成のサンプルサイズに乗じる値）
 0 0 0 0 #_mult_by_agecomp_N
 0 0 0 0 #_mult_by_size-at-age_N
```

演習　CPUEのCVを変えてみよう

1 #_分散（入力したCVやサンプルサイズ）を調整するための設定

#_fleet: 1 2 3 4

0 0 0 0 #_add_to_survey_CV（資源量指数のCVに足す値）

0 0 0 0 #_add_to_discard_stddev

0 0 0 0 #_add_to_bodywt_CV

1 1 1 1 #_mult_by_lencomp_N（体長組成のサンプルサイズに乗じる値）

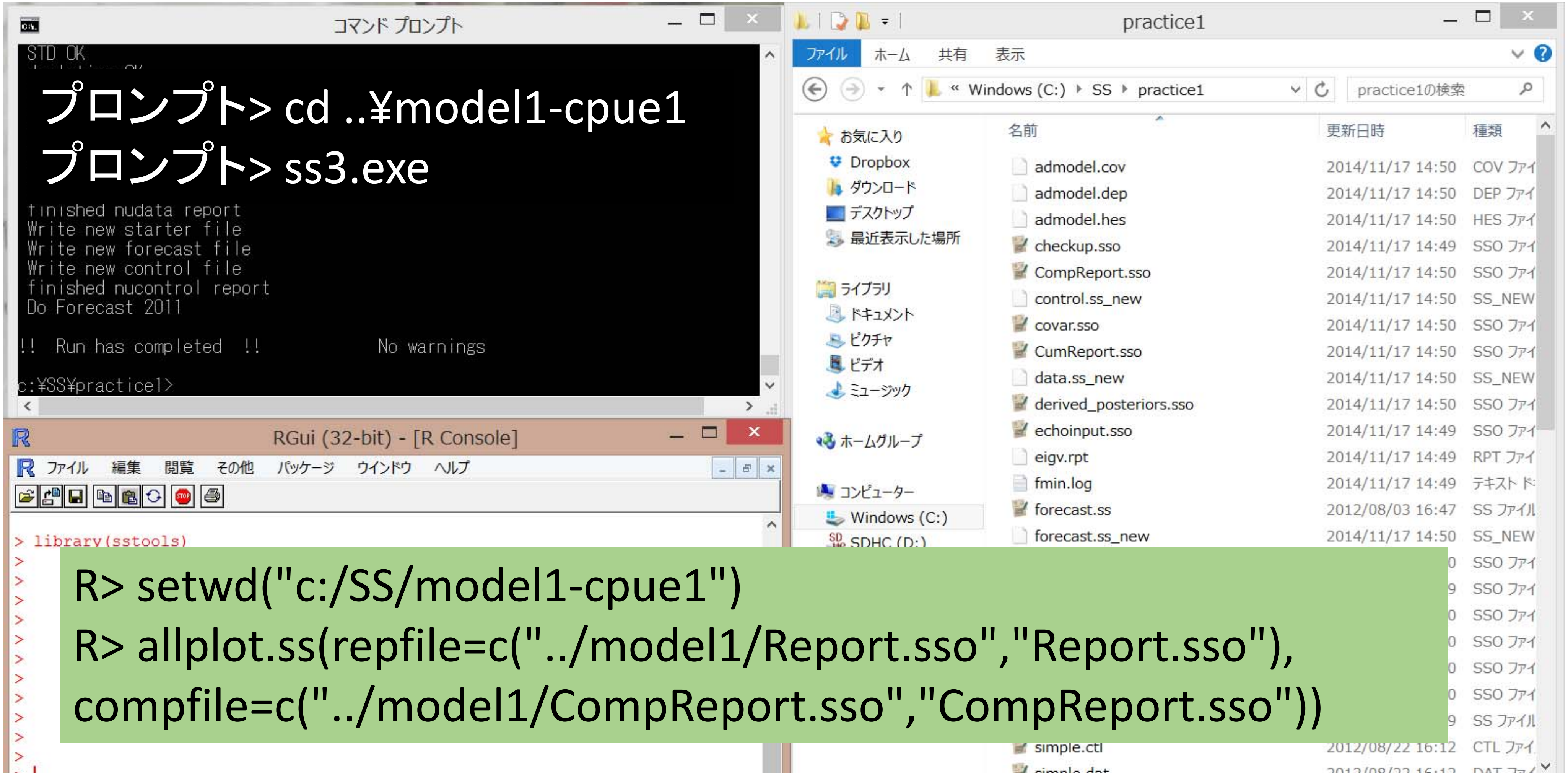
0 0 0 0 #_mult_by_agecomp_N

0 0 0 0 #_mult_by_size-at-age_N

F2のCPUEのCVに0.1を足す

ここを0.1にする

Model1-cpue1で走らせる&結果の比較



- λの調整方法
  - 分散調整の下の部分
  - ラムダを直接変更する

```
0 # 4 #_デフォルトのラムダから変更する数（デフォル
# Like_comp codes:  1=surv; 2=disc; 3=mnwt; 4=lengt
7=sizeage; 8=catch;
# 9=init_equ_catch; 10=recrdev; 11=parm_prior; 12=
14=Morphcomp; 15=Tag-comp; 16=Tag-negbin
#like_comp fleet/survey  phase  value  sizefreq_methc
# 6 1 1 1 1
# 6 2 1 1 1 # what is sizefreq_method?
```

演習　CPUE2を使わない

4 #_ラムダの設定。このあと何列の設定を読むか

Like_comp codes: 1=surv; 2=disc; 3=mnwt; 4=length; 5=age; 6=SizeFreq; 7=sizeage; 8=catch;

9=init_equ_catch; 10=recrdev; 11=parm_prior; 12=parm_dev; 13=CrashPen; 14=Morphcomp; 15=Tag-comp; 16=Tag-negbin

#like_comp fleet/survey phase value sizefreq_method(?)

4 1 1 1 1

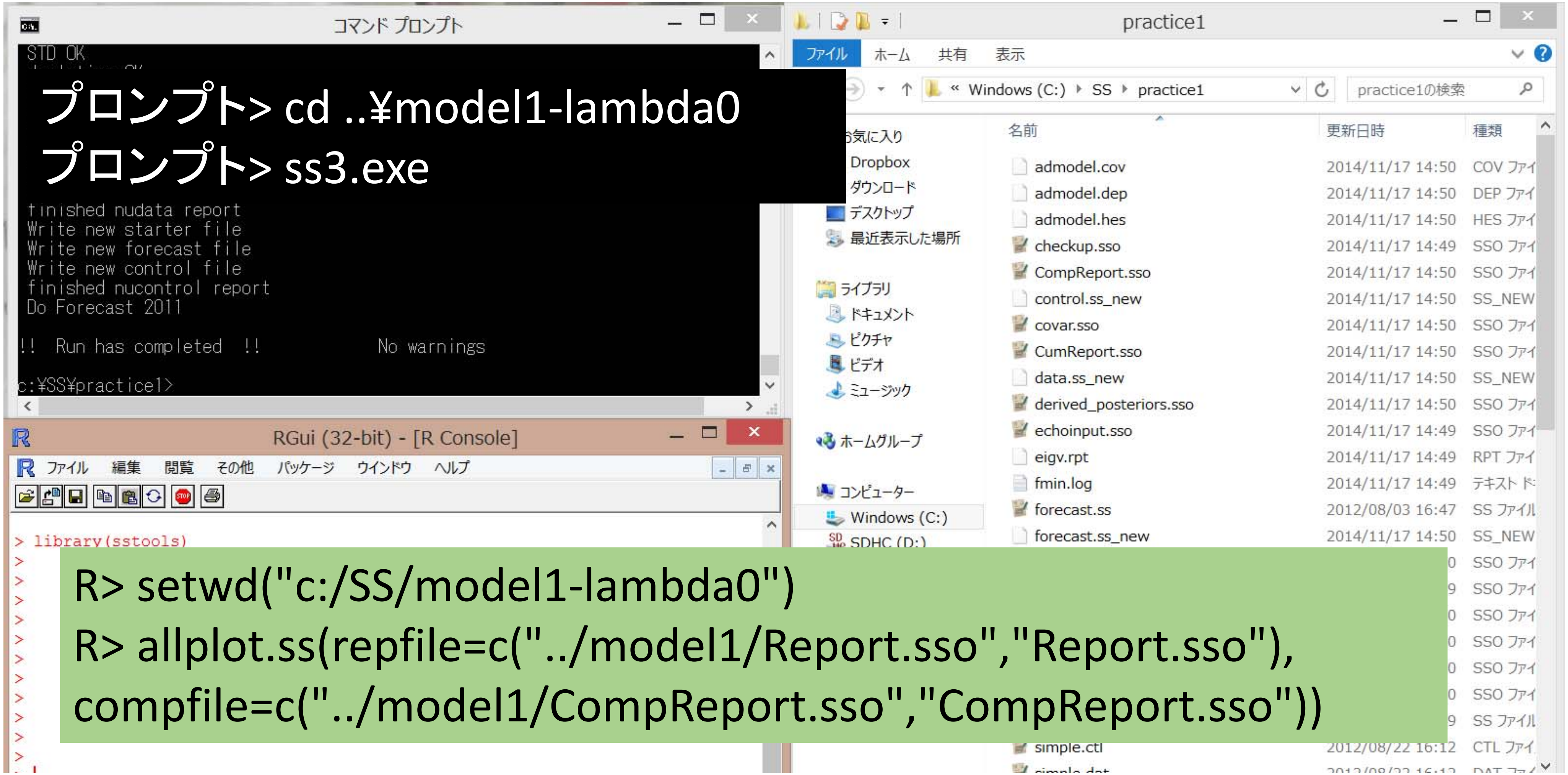
4 2 1 1 1

1 3 1 1 0

1 4 1 1 0

CPUE2のラムダを0に

model1-lambda0で走らせる&結果の比較




```
プロンプト> cd ..¥model1-lambda0
プロンプト> ss3.exe
```

```
R> setwd("c:/SS/model1-lambda0")
R> allplot.ss(repfile=c("../model1/Report.sso","Report.sso"),
compfile=c("../model1/CompReport.sso","CompReport.sso"))
```