平成30年度資源管理研修会
2018/12/27
9:30-12:30

CPUE標準化と時空間解析

西嶋　翔太
（中央水産研究所 資源研究センター）

謝辞

本研修の実例として、愛知県水産試験場からシャコの、富山県水産試験場からサワラの漁業データを貸していただきました。大変ありがとうございました。

データの使用は参加者のみとし、他の人には配布しないようお願いいたします。

使用するRパッケージ

- AER
- MASS
- mgcv
- arm
- lme4
- CARBayes
- pscl
- speedglm
- TMB（Rtoolsを事前に入れておく必要あり）
- INLA
- VAST

✓ 実演したい人は事前にインストールをお願いします

✓ また、以下のサイトに示されている手順に従って、TMB, INLA, VASTのパッケージのインストールもお願いします。 https://github.com/James-Thorson/VAST

本日の内容

- CPUE標準化とは
- 実例1: シャコ
  - 様々な確率分布を使った一般化線形モデル (GLM)
  - モデル診断とモデル選択
  - 信頼区間
  - 一般化加法モデル (GAM)
- 実例2: サワラ
  - 一般化線形混合モデル (GLMM)
  - 空間自己相関を扱ったモデル1: CAR model
- 実例3: ミナミマグロ
  - zero catchを多く含むときの解析: delta GLM / zero inflated model
  - 空間自己相関を扱ったモデル2: VAST

CPUE(catch per unit effort)

- 単位努力量あたりの漁獲量CPUEとは、努力量を漁獲量で除したもの
- 努力量の単位は、時間・人数・網・隻数など
- 相対的なトレンドを表す資源量指数　CPUE = q × N
- 個体群モデル (VPAなど)のチューニングに使用される
- それ自身で許容漁獲量を決められることもある（2系ルール）

Evaluating methods for setting catch limits in data-limited fisheries

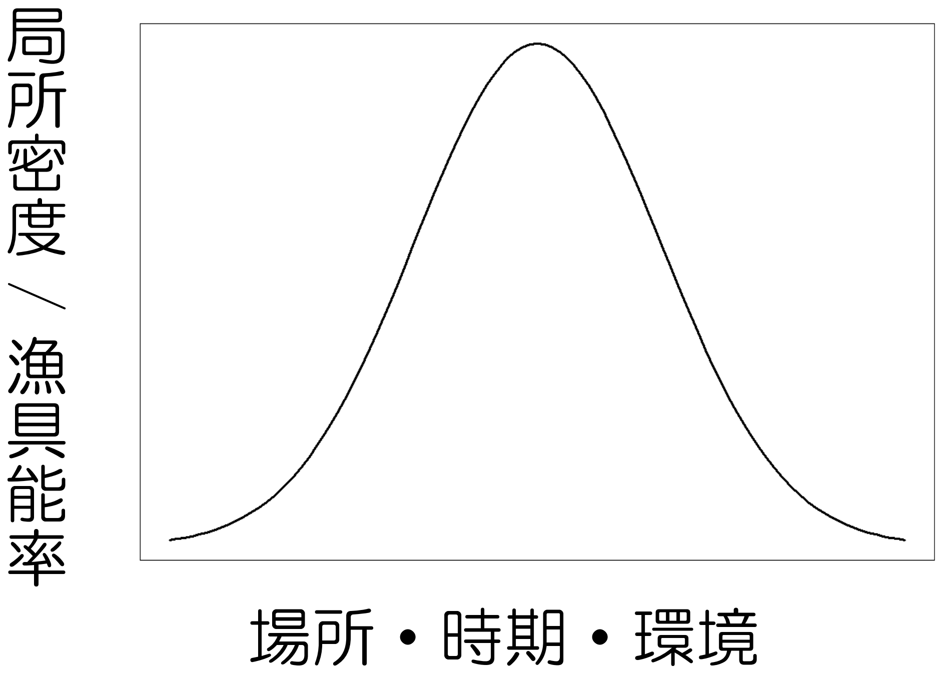
Thomas R. Carruthers[a,*], André E. Punt[b,1], Carl J. Walters[a], Alec MacCall[c,2], Murdoch K. McAllister[a], Edward J. Dick[c,2], Jason Cope[d,3]

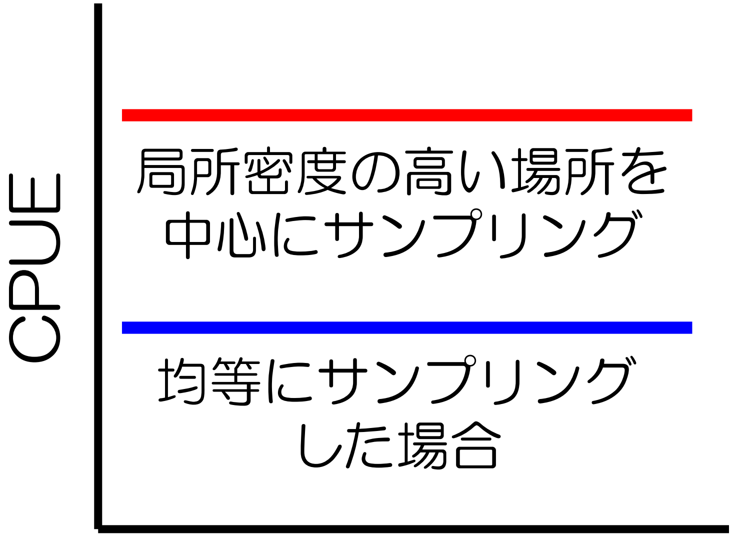
(Carruthers et al. 2014、Fish.Res.)

CPUEデータに生じるバイアス

CPUE = q * N

- 漁具能率（q）も時間や場所によって変わる
- 知りたいのは資源量全体だが、データは局所的な密度を反映



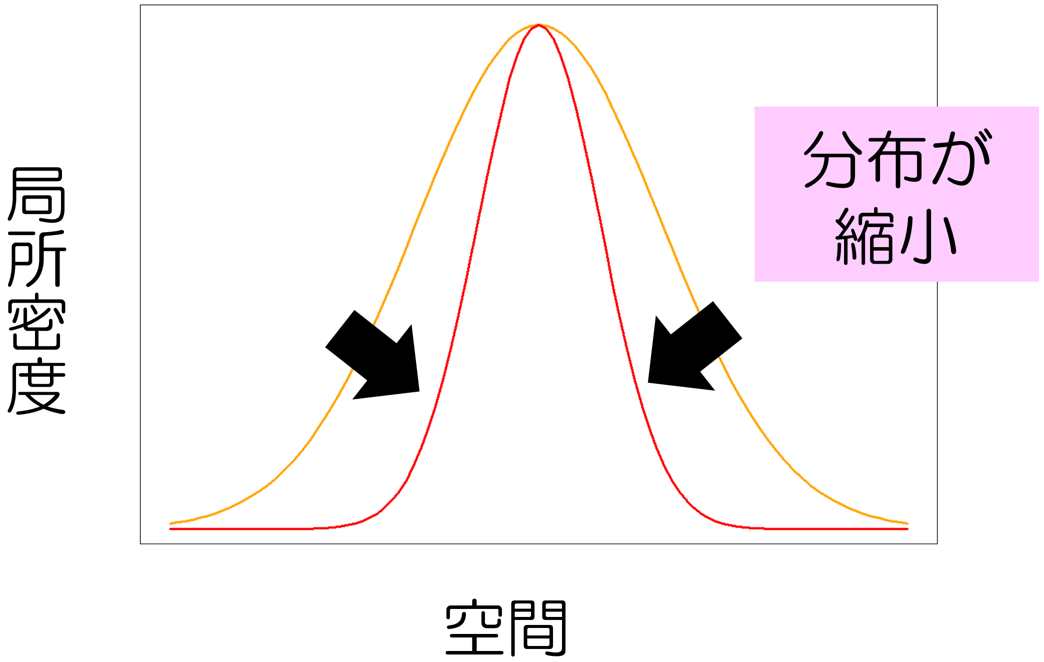


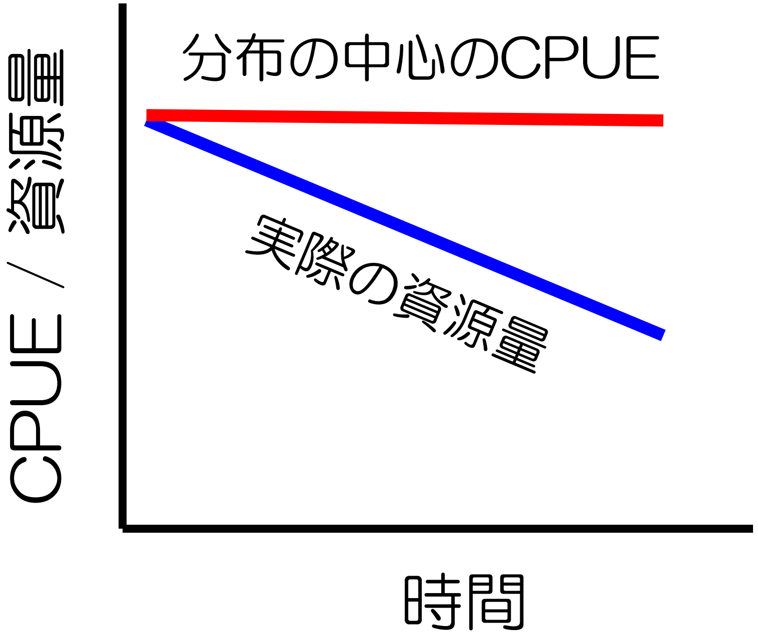
バイアスが生じる（特に漁業データで）

CPUEデータに生じるバイアス

CPUE = q＊N

- 漁具能率（q）も時間や場所によって変わる
- 知りたいのは資源量全体だが、データは局所的な密度を反映

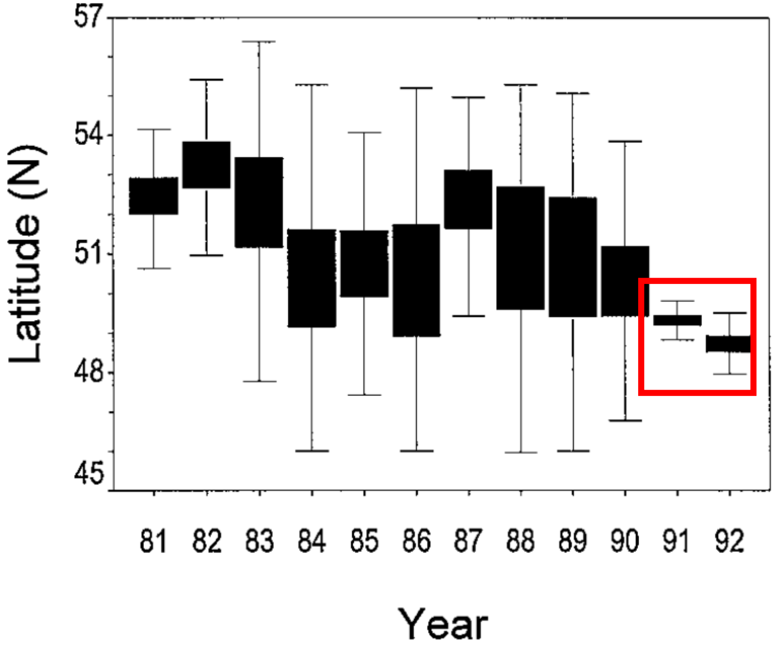




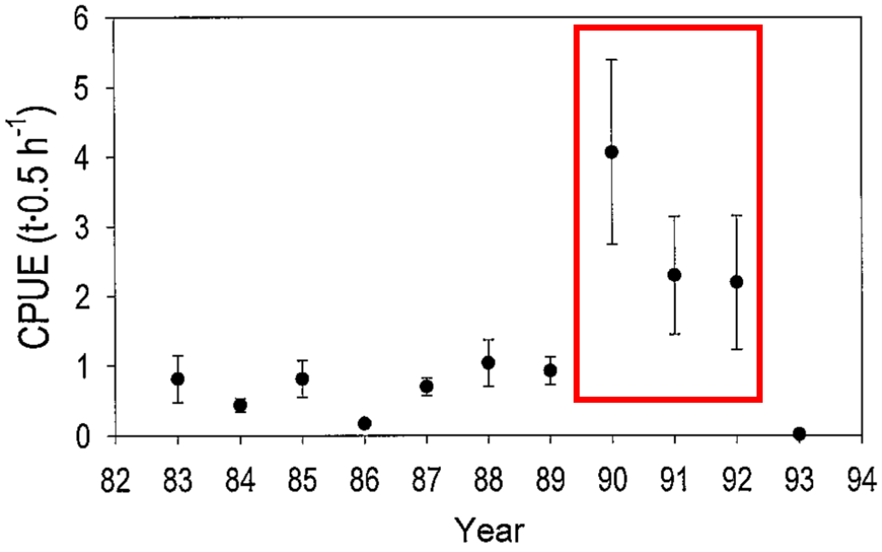
分布変化の影響を受ける

タイセイヨウダラ (Atlantic cod) の例

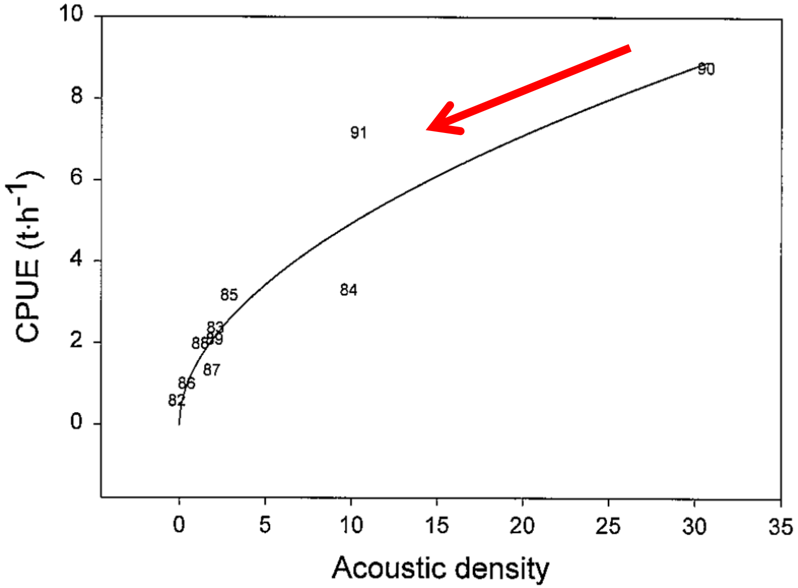
分布の縮小
Hyperaggregation



CPUEの一時的な上昇



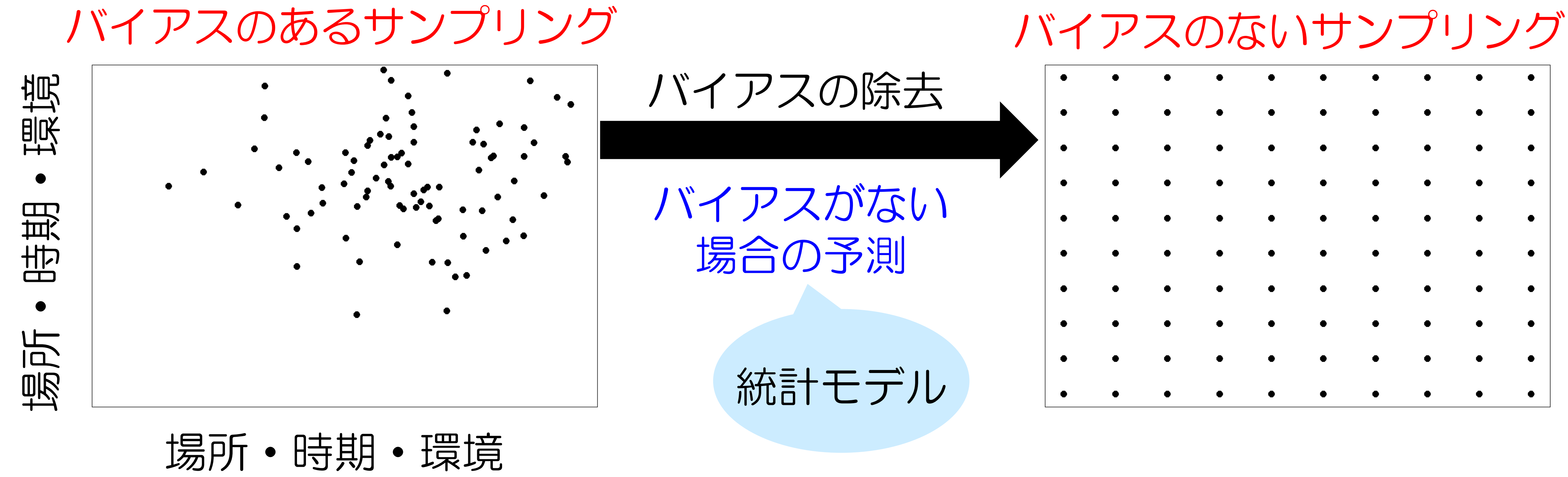
Hyperstability



(Rose & Kulka 1999 CJFAS)

CPUEの「標準化」

- サンプリングバイアスを除去した場合のCPUEを予測し、年トレンドを取り出すこと



- 「予測力の高い」統計モデルが必要

統計モデルの「予測力」

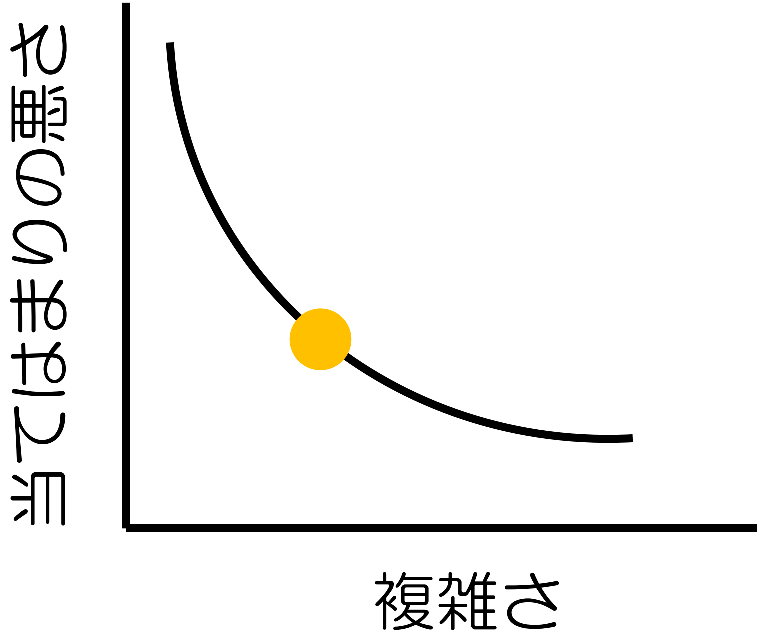
- 予測力の高いモデルとは、節約的かつデータへの当てはまりが良いモデル

トレードオフにおいてバランスのいいポイントを探す

- 予測力を測る指標：

当てはまりの悪さ ＋ モデルの複雑さ

AIC = -2*Log(L) + 2*n*p



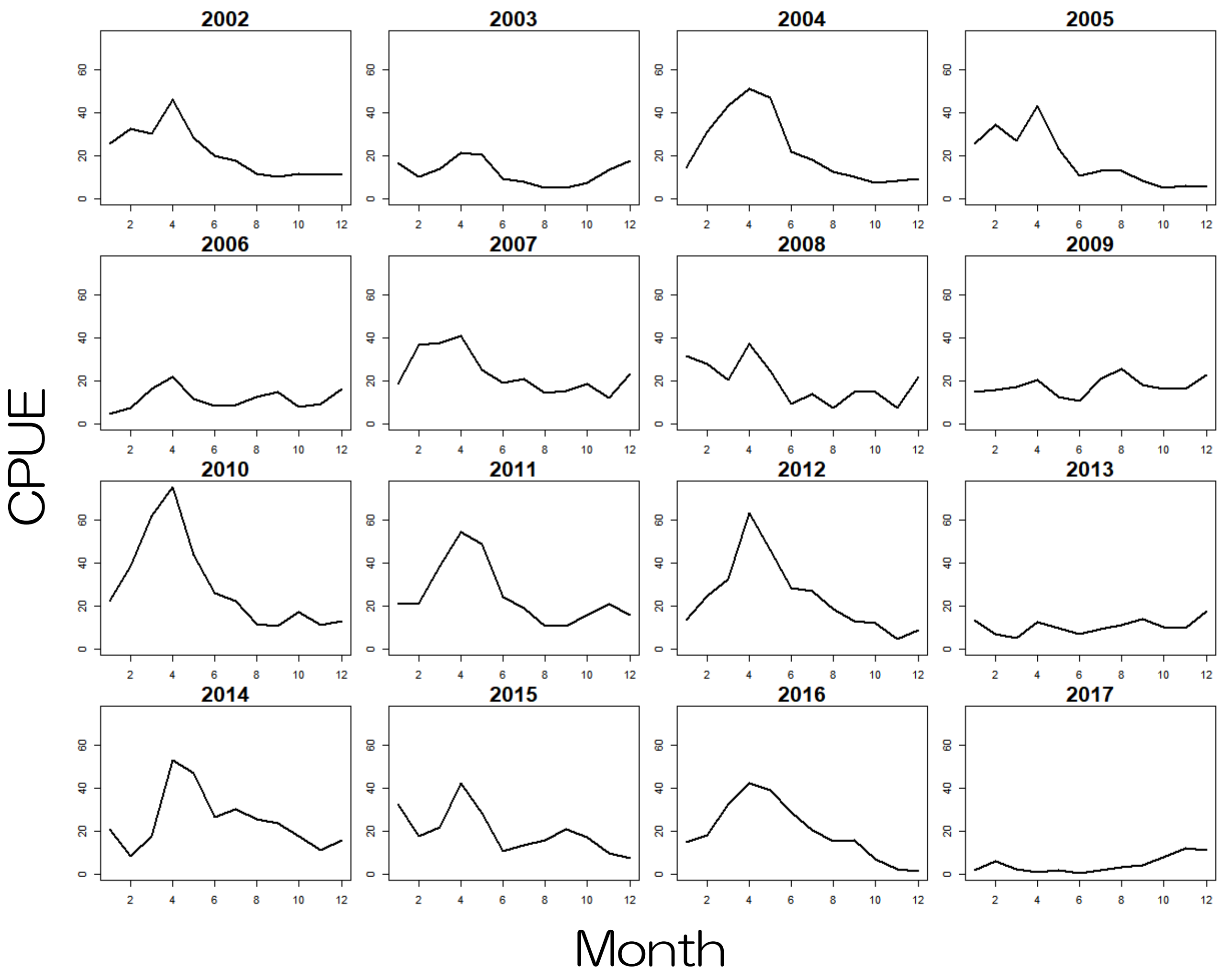
本日の内容

- 実例1: シャコ
  - 様々な確率分布を使った一般化線形モデル (GLM)
  - モデル診断とモデル選択
  - 信頼区間
  - 一般化加法モデル (GAM)
- 実例2: サワラ
  - 一般化線形混合モデル (GLMM)
  - 空間自己相関を扱ったモデル1: CAR model
- 実例3: ミナミマグロ
  - zero catchを多く含むときの解析: delta GLM / zero inflated model
  - 空間自己相関を扱ったモデル2: VAST

実例１：シャコの月別CPUEデータ

```
> head(dat)
  Year Month Catch Effort     CPUE
1 2002     1  8404    327 25.70031
2 2002     2 10945    334 32.76946
3 2002     3 14456    479 30.17954
4 2002     4 31887    693 46.01299
5 2002     5 21873    768 28.48047
6 2002     6 18028    910 19.81099
```

使用するデータ
- 2002～17年の月別データ
- 小型底引き網の漁獲量（kg）
- 努力量（延べ隻数）



一般化線形モデル（GLM）：正規分布

```
norm <- glm(CPUE ~ Year + Month, family = gaussian(), data = dat)
```

目的変数　説明変数　確率分布（今は正規分布）　データ

係数表の見方

```
> as.data.frame(coef(norm))
             coef(norm)
(Intercept)   20.510672
Year2003      -8.988538
Year2004       1.471489
...
Year2017     -16.904601
Month2         2.952152
Month3         7.907778
...
Month12       -4.651303
```

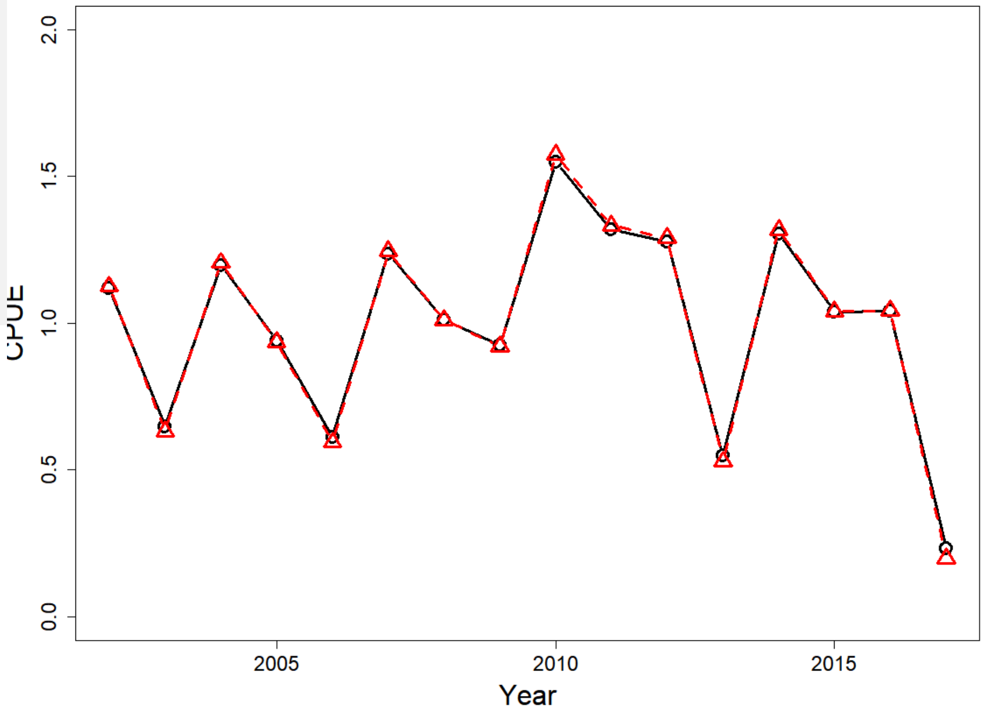
(Intercept) → ２００２年1月の回帰係数（最初のカテゴリ）

2003～2017年（Year2003～Year2017）：２００２年と比較したときの回帰係数

2月～12月（Month2～Month12）：1月と比較したときの回帰係数

年トレンドの導出（交互作用なし）

```
standardCPUE <- c()
for (i in 1:nyear) {
  if (i==1) standardCPUE[i] <- norm$coefficients[1]
#1年目は切片の係数
  else standardCPUE[i] <-
      norm$coefficients[1] + norm$coefficients[i]
# 2年目以降は切片+回帰係数
}
standardCPUE <- standardCPUE/mean(standardCPUE)
#scaled by mean
```



※ただし、この手法は年との交互作用項を含まないモデルしか使えない

より簡単な年トレンドの導出手法

```
norm <- glm(CPUE ~ 0 + Year + Month, family = gaussian(), data =dat)
```

0を足すか1を引く
と、切片がなくなる

2002年から始まる！
（各年の係数の絶対値）

```
Coefficients:
         Estimate Std. Error t value Pr(>|t|)
Year2002   20.511      3.237   6.336 2.16e-09 ***
Year2003   11.522      3.237   3.559 0.000486 ***
Year2004   21.982      3.237   6.790 1.93e-10 ***
```

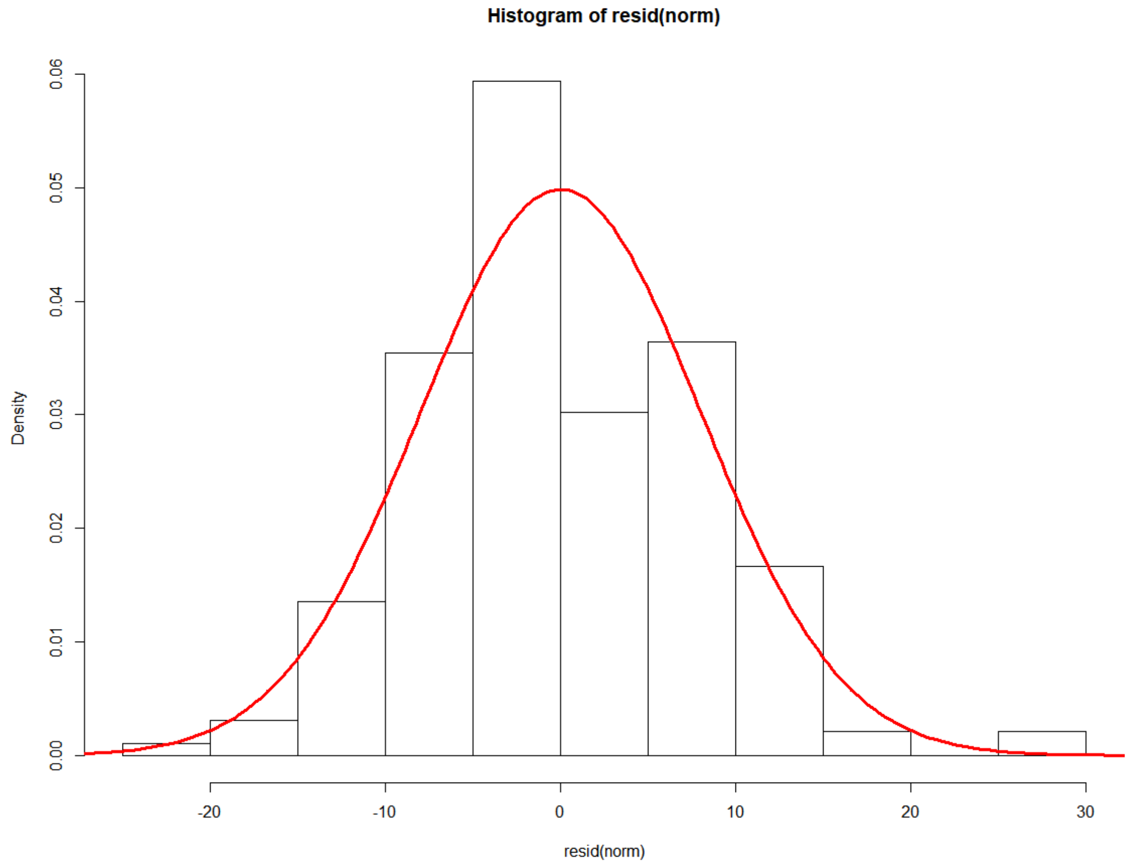
```
standardCPUE <- norm$coefficients[1:nyear] #係数をそのまま取ればいい
standardCPUE <- standardCPUE/mean(standardCPUE)
```

※ただし、この手法は年との交互作用項を
含まないモデルしか使えない

モデルの診断：正規性のチェック

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon,$$

$$\underbrace{\varepsilon \sim \mathrm{Normal}(0, \sigma^2)}_{\text{残差が正規分布に従う}}$$



Shapiro-Wilk検定

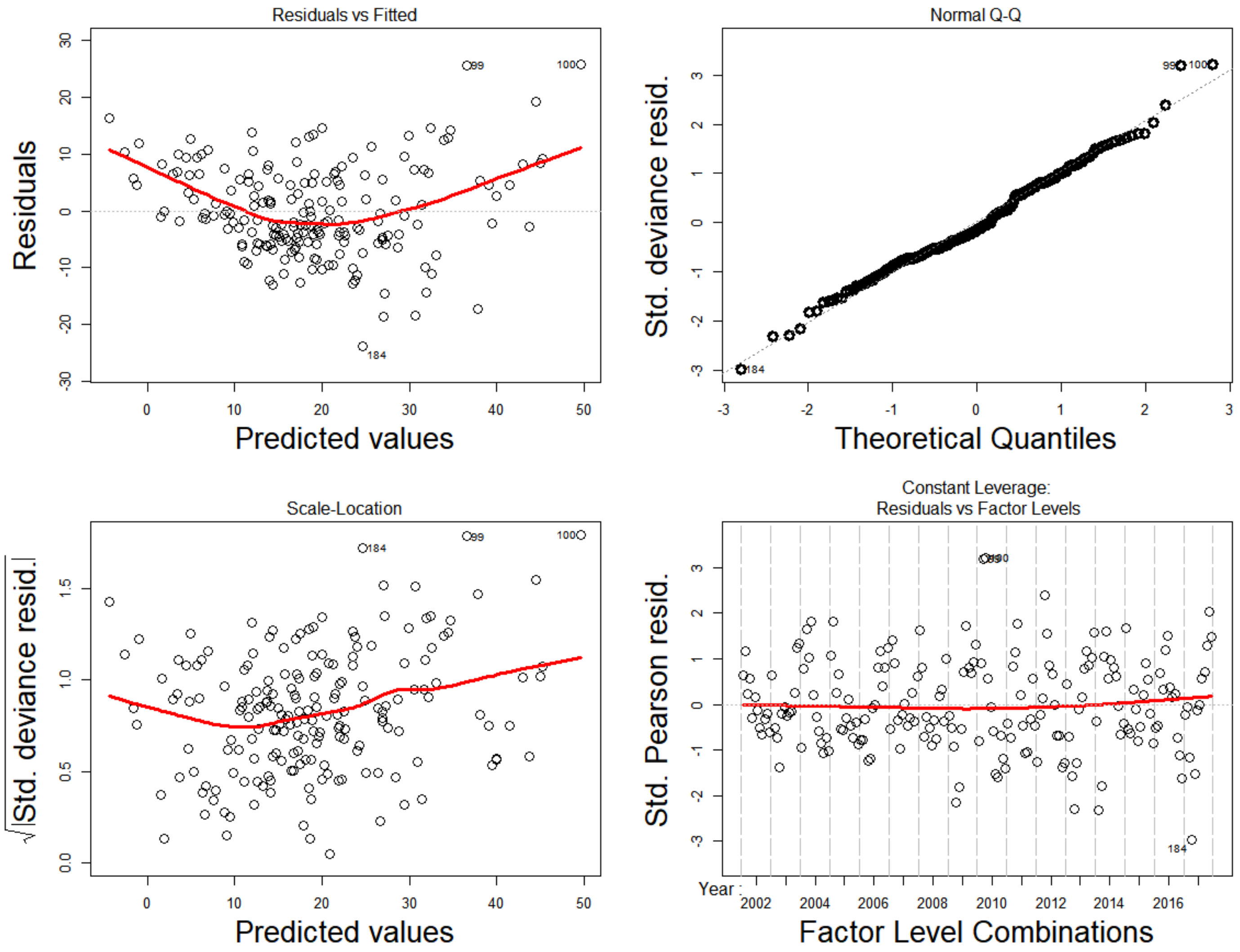
帰無仮説：標本が正規正規分布に従う

```
> shapiro.test(resid(norm))

        Shapiro-Wilk normality test

data:  resid(norm)
W = 0.9885, p-value = 0.1235
```

モデルの診断：QQプロット等

`plot(glm object)`
で診断結果がプロットできる

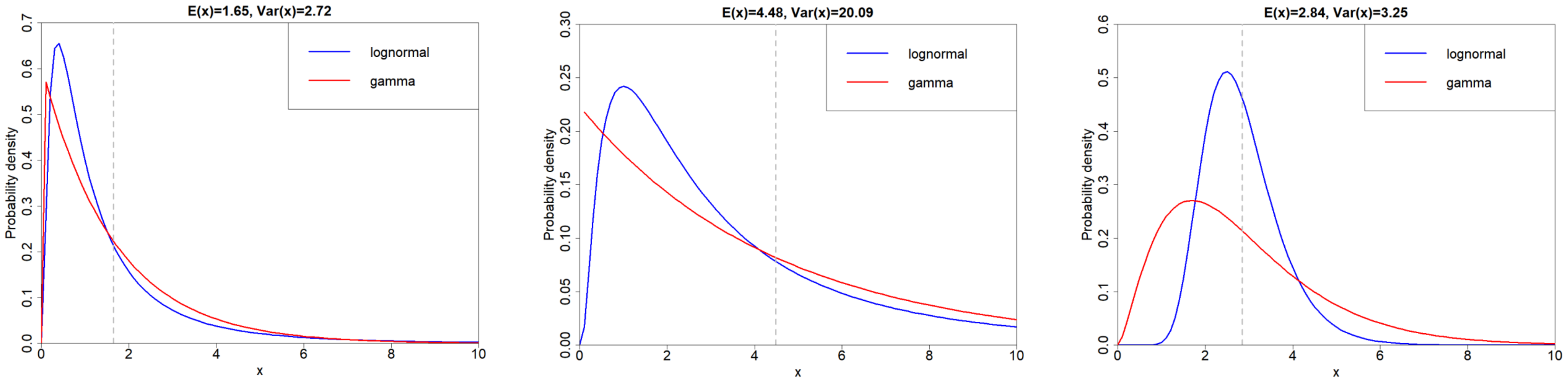
QQプロット（右上）：
正規分布に従っていれば直線上に点が載る



0よりも大きい連続変数を扱える分布

対数正規分布　対数をとると正規分布に従う分布 $\log(y) \sim \mathrm{Normal}(\mu, \sigma^2)$

ガンマ分布　指数分布から導出、対数正規分布よりも極端な値をとりやすい



GLM：対数正規分布

```
lognorm <- glm(log(CPUE) ~ 0 + Year + Month, family = gaussian(), data=dat)
```

↓

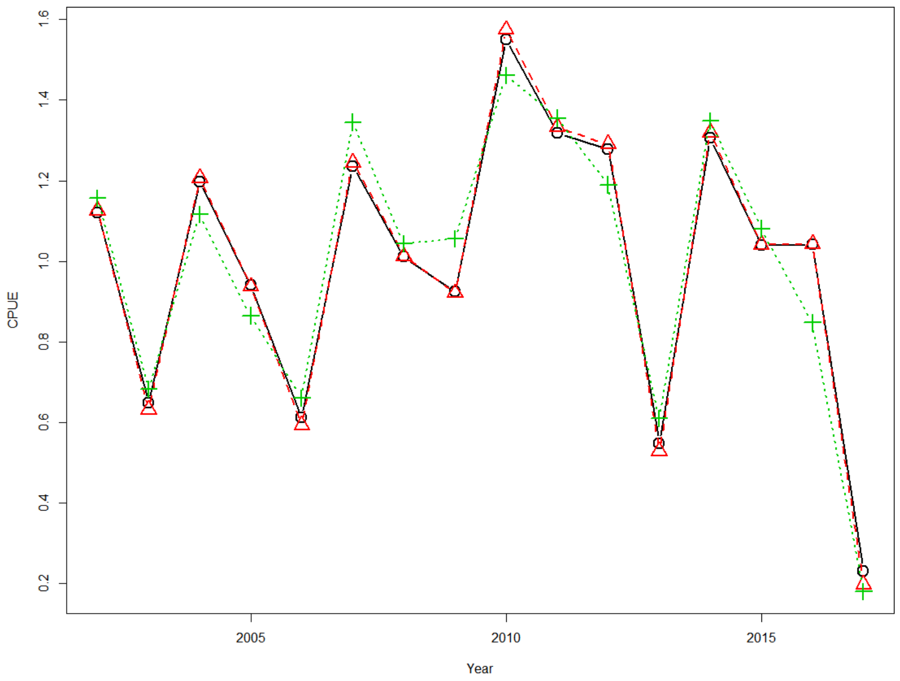
対数をとったものを
目的変数にする

CPUEの幾何平均を予測

```
standardCPUE <- exp(lognorm$coefficients[1:nyear])
```

↓

係数のexpをとった
ものが標準化CPUE

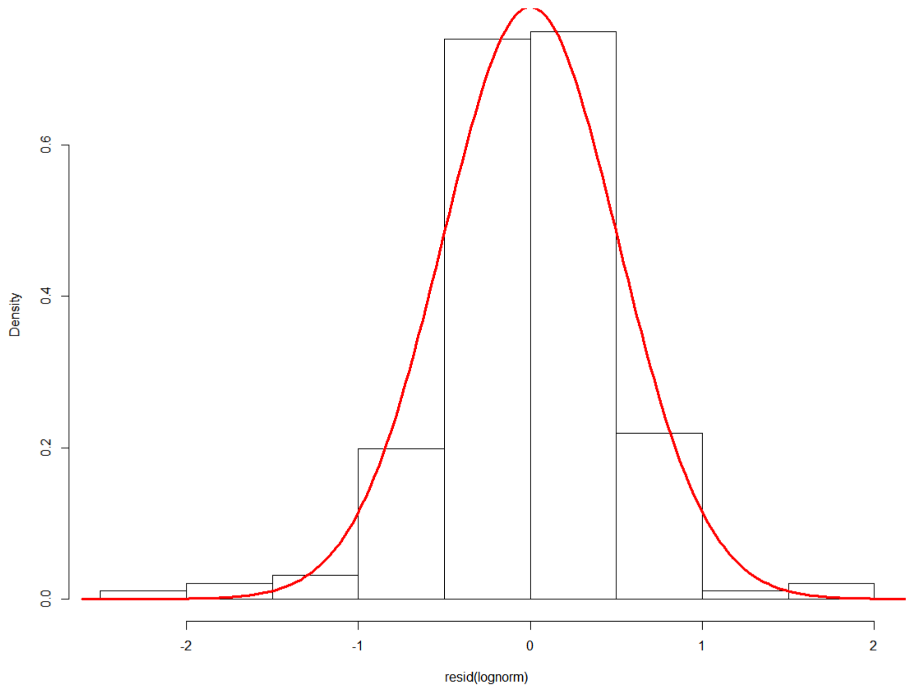


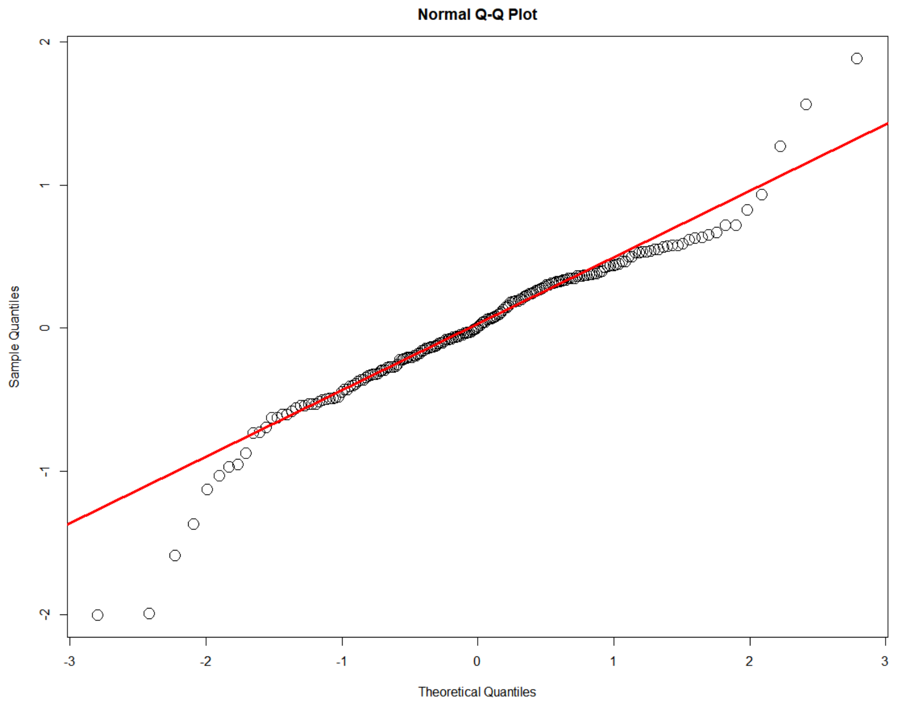
モデル診断

```
> shapiro.test(resid(lognorm))

        Shapiro-Wilk normality test

data:  resid(lognorm)
W = 0.95123, p-value = 3.817e-06
```

↓

有意：正規分布に従っている
とは言えない





GLM：ガンマ分布

```
gamma <- glm(CPUE ~ 0 + Year + Month, family = Gamma("log"), data = dat)
```

↓ 目的変数はCPUEのまま

↓ ガンマ分布を指定し、リンク関数をlogに

$$\mathrm{E}(\mathrm{CPUE}) = \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots)$$

CPUEの期待値が線形予測子のexpで表されるということ

```
standardCPUE <- exp(gamma$coefficients[1:nyear])
```

↓ 係数のexpをとったものが標準化CPUE

（対数正規分布と同じ）

正規分布以外の場合の診断

逸脱度 (deviance)　D = -2*log-likelihood

当てはまりの悪さを表し、負の対数尤度の2倍で表される

残差逸脱度 (residual deviance)　D_{resid} = D − 最小のDeviance

Devianceと完全にデータを予測できた場合のdeviance (min(D))との差

逸脱残差 (deviance residual)　R = sign(y-mu)*$\sqrt{D_{resid}}$

残差逸脱度の平方根をとって、残差の符号をかけたもの

➡ これを一般化された残差として、正規分布と同様のチェックができる（はず）

resid(glm object)はディフォルトで逸脱残差を計算してくれる

ホームワーク①

- ガンマ分布を使った解析結果で、ヒストグラムやQQ plotを描いたり、Shapiro-Wilk検定を実行し、モデルの診断をしてみよう
- 正規分布や対数正規分布のコードにおいて、glm objectを入れ替えればできるはずです

AICによるモデル選択

```
lognorm <- glm(log(CPUE) ~ 0 + Year + Month, family = gaussian(), data=dat)
```

目的変数が違うので単純には比較できない

```
gamma <- glm(CPUE ~ 0 + Year + Month, family = Gamma("log"), data = dat)
```

正規分布の対数尤度

$$\log L_{norm} = -\frac{n}{2} - \frac{1}{2\sigma^2}\sum_{k=1}^{n}(\log x_k - \mu_k)^2$$

変換

対数正規分布の対数尤度

$$\log L_{lognorm} = \log L_{norm} - \sum_{k=1}^{n} \log x_k$$

```
> AIC(gamma) # gamma
[1] 1359.531
> as.numeric(-2*(logLik(lognorm)-sum(log(dat$CPUE))) +
+              2*(norm$rank+1)) #AIC for lognormal
[1] 1377.399
```

目的変数の合計を引く

パラメータ数

GLM：ポワソン分布～非負整数の解析

シャコのCatchデータは漁獲重量なので、本来整数ではなく連続変数とみなすべきですが、ここでは漁獲尾数とみなしてポワソン分布の解析をしてみます

CPUEではなくCatchを使う
（整数データ）
↓

Offset項に
log(effort)を指定
↓

ポワソン分布で
log link関数
↓

```
pois <- glm(Catch ~ Year + Month + offset(log(Effort)), family = poisson("log"),
            data = dat)
```

Offset項　　回帰係数を1と固定してパラメータ推定

$$\mathrm{E(Catch)} = \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \log(\mathrm{Effort}))$$

$$\Rightarrow \mathrm{E}\left(\frac{\mathrm{Catch}}{\mathrm{Effort}}\right) = \mathrm{E(CPUE)} = \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots)$$

過分散 (overdispersion)

- ポワソン分布は平均と分散が等しく、分散に関するパラメータが存在しません
- 個体差がなく、各個体が独立にふるまっていることを仮定しています
- 生態学や水産資源学のデータではこの仮定が不成立なことが多く、分散が平均よりも大きくなることが多いです（過分散）

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 832515  on 191  degrees of freedom
Residual deviance: 239035  on 165  degrees of freedom
```

← この比が1に近くないといけない

```
> dispersiontest(pois) #significantly overdispersed

	Overdispersion test

data:  pois
z = 9.6924, p-value < 2.2e-16
```

← library(AER)内にある過分散の検定

有意 ⇒ 帰無仮説「過分散でない」とはいえない

GLM：負の二項分布～過分散への対応

- 過分散の問題に対処可能な分布として、負の二項分布があります
- 負の二項分布は、ポワソン分布における平均がガンマ分布で「ばらつく」ことを考えています（negative binomial = poisson + gamma）
- 確率pで成功するベルヌーイ試行において、成功回数がk回になるまでに必要な試行回数として定義されまし（そのため「負の」二項分布と呼ばれる）

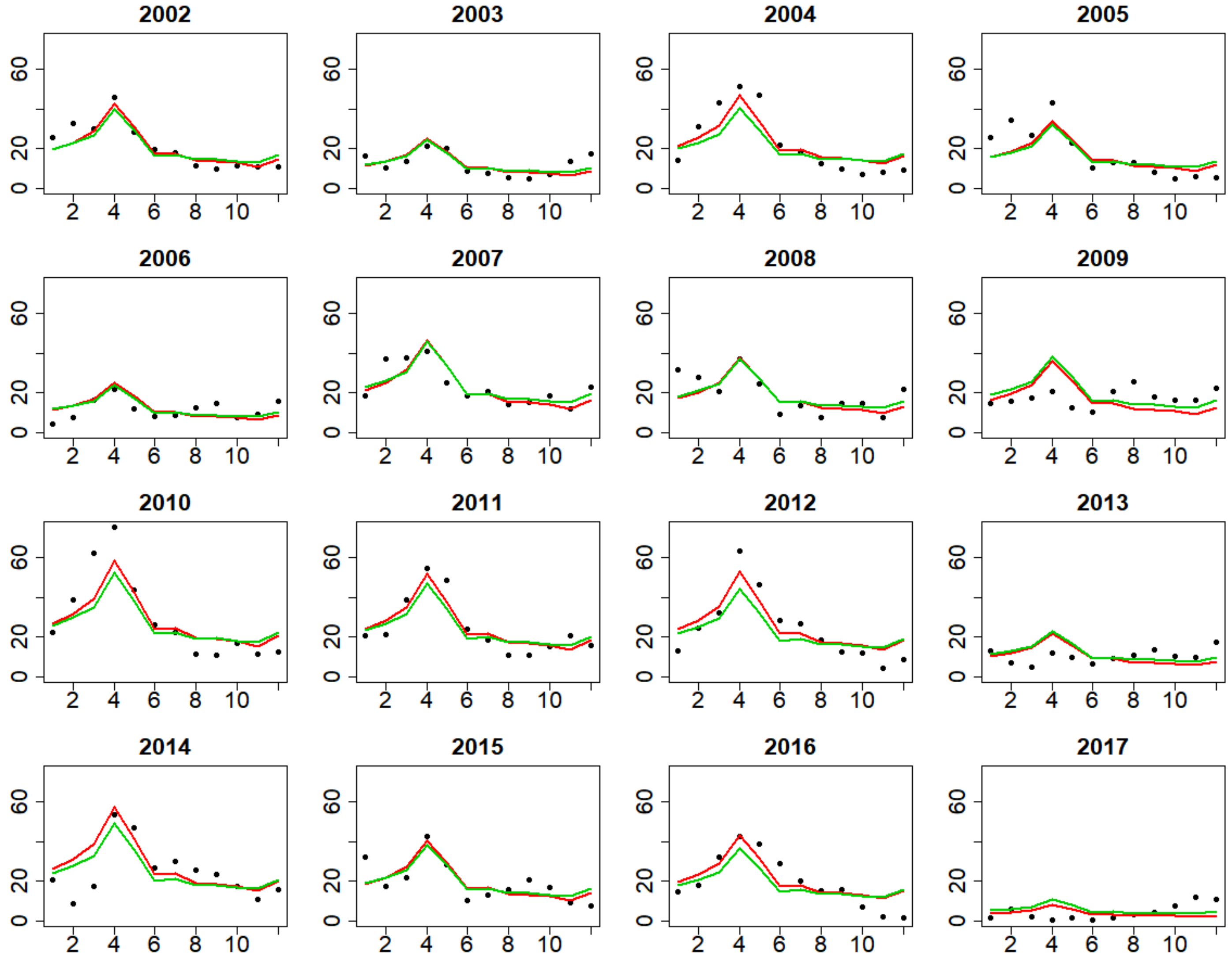
```
library(MASS)
nb <- glm.nb(Catch ~ Year + Month + offset(log(Effort)), data = dat)
```

library(MASS) に入っているglm.nbという関数を使います

```
> AIC(pois,nb)
     df          AIC
pois 27 241167.036
nb   28   3774.602
```

- 負の二項分布の方がAICが断然低い！
- 負の二項分布の方が自由度が1つ増えていることに注意

当てはまりと予測値の比較



赤：ポワソン

緑：負の二項分布

交互作用効果：月の影響が年で変わる

- ある変数の影響が別の変数によって変わることを「交互作用効果」と言います

```
interact0 <- glm(log(CPUE) ~ Year*Month, data = dat)
```

*を使って年と月の交互作用を表す

```
                  Estimate    Std. Error t value Pr(>|t|)
(Intercept)        3.246503          NA       NA        NA
Year2003          -0.437131
...
Year2017          -2.623973
Month2             0.242994
...
Month12           -0.858927
Year2003:Month2   -0.722569
…
Year2017:Month12   2.624844          NA       NA        NA
```

2002年1月

2003~17年の1月（切片との差）

2002年2~12月（切片との差）

2003~17年の2~12月（切片との差）

推定できてない！

データ数192に対して誤差の分散を入れて193個のパラメータを推定しているから

典型的なoverfitting（過剰適合）

月をカテゴリカル変数からを連続変数に

```
dat$Month <- as.numeric(dat$Month)
interact <- glm(log(CPUE) ~ Year * Month, data = dat)
```

2002年0月

2003～17年の0月（切片との差）

2002年の月効果（傾き）

2003～17年の月効果（2002年傾きとの差）

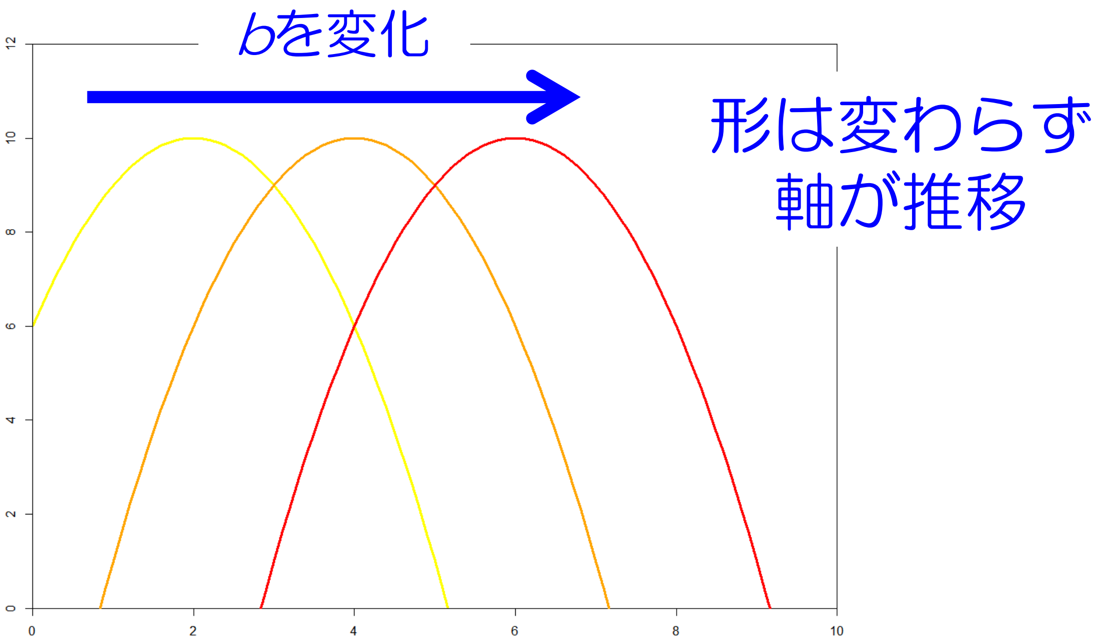
```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.758058   0.304753  12.331  < 2e-16 ***
Year2003       -1.068334   0.430985  -2.479 0.014219 *
...
Year2017       -3.730573   0.430985  -8.656 4.98e-15 ***
Month          -0.127228   0.041408  -3.073 0.002495 **
Year2003:Month  0.083186   0.058559   1.421 0.157396
...
Year2017:Month  0.288261   0.058559   4.923 2.11e-06 ***
```

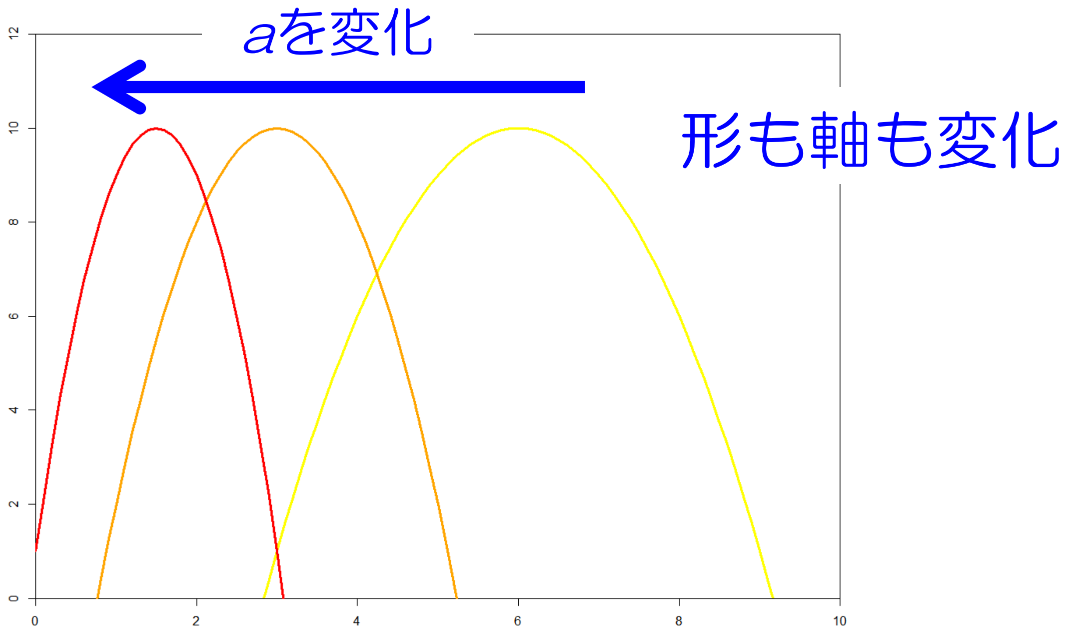
推定できている

連続変数の二乗項

- 連続変数の二乗項を入れると上（下）に凸の関数が作れ、非線形性を考慮できる
- 環境変数（水温など）に対する反応を表すのに便利

$$f(x) = ax^2 + bx + c$$





二乗項と交互作用効果

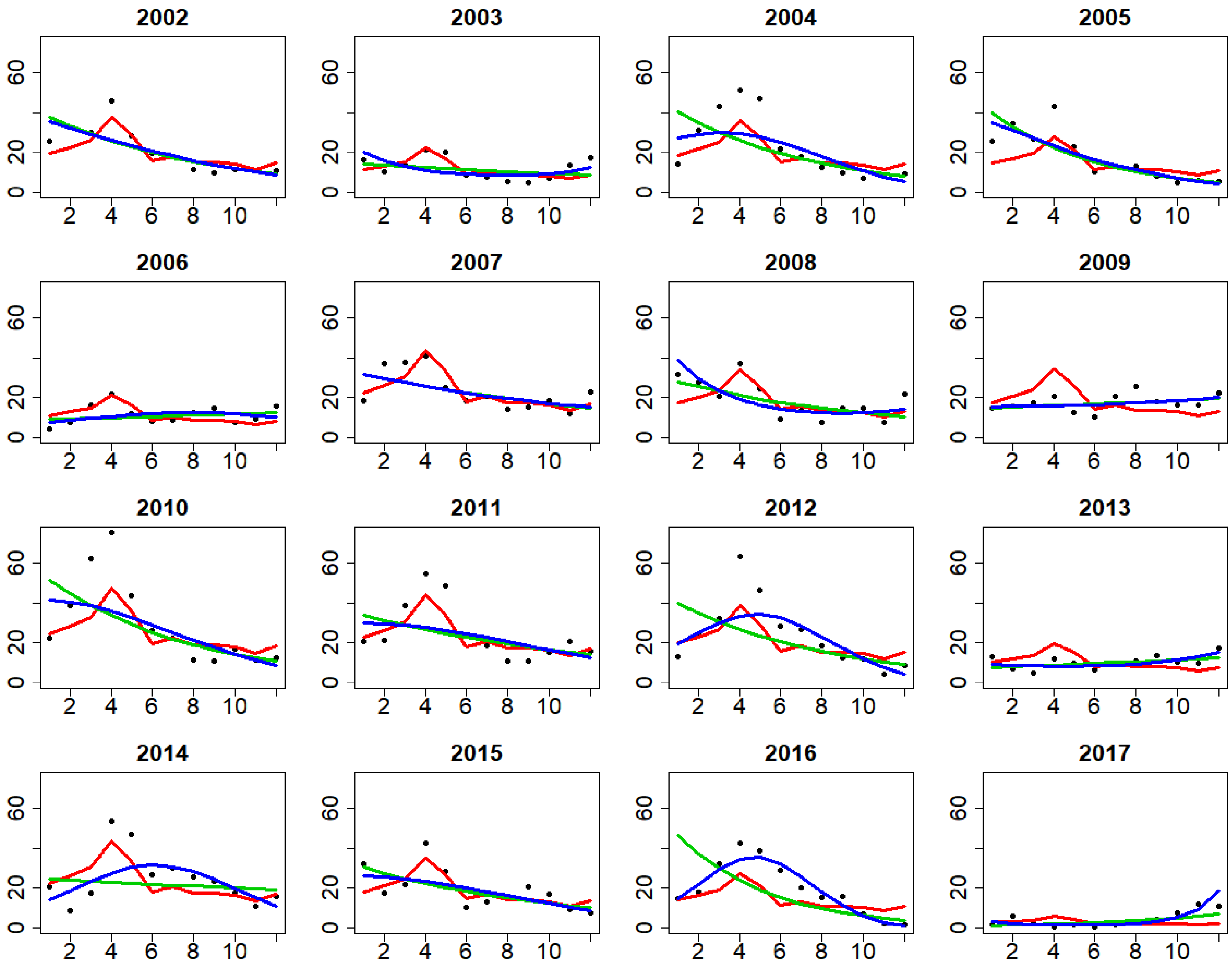
二乗項

```
> interact2 <- glm(log(CPUE) ~ Year * Month + I(Month^2), data = dat)
> interact3 <- glm(log(CPUE) ~ Year * Month + Year * I(Month^2), data = dat)
> AIC(lognorm, interact, interact2, interact3)
          df      AIC
lognorm   28 342.2260
interact  33 305.9665
interact2 34 303.8725
interact3 49 248.9623
```

二乗項と年の交互作用

一次項と二乗項と両方の
交互作用アリがベスト

当てはまり



赤：交互作用なし

緑：交互作用あり（二乗項なし）

青：交互作用＆二乗項あり（ベストモデル）

あまり当てはまっているようには見えない

月の効果を二乗項で表すのは難しい

ホームワーク②

- 月の二乗項と、年と月の交互作用を加えたモデルにおいて、QQ plotやShapiro-Wilk検定で残差の正規性をチェックしてみよう
- 同様のモデルのガンマ分布版を作って、AICを対数正規分布モデルとAICを比較してみよう

- 正規性が保証され、AICも対数正規分布モデルの方が少し低くなるはず…

年トレンドの導出方法

1. 各説明変数（年と月）の総当たりの組み合わせをもつバランスデータを作成（今回は元のデータと一致する）

CPUE	1月	2月	…	11月	12月	
2002	$\hat{y}_{2002,1}$	$\hat{y}_{2002,2}$	…	$\hat{y}_{2002,11}$	$\hat{y}_{2002,12}$	$\rightarrow \mathrm{E}(\hat{y}_{2002})$
2003	$\hat{y}_{2003,1}$	$\hat{y}_{2003,2}$	…	$\hat{y}_{2003,11}$	$\hat{y}_{2003,12}$	$\rightarrow \mathrm{E}(\hat{y}_{2003})$
…	…	…	…	…	…	
2017	$\hat{y}_{2017,1}$	$\hat{y}_{2017,2}$	…	$\hat{y}_{2017,11}$	$\hat{y}_{2017,12}$	$\rightarrow \mathrm{E}(\hat{y}_{2017})$

2. 推定されたモデル（GLM等）を使って、バランスデータにおけるCPUEを予測する

3. 各年における予測CPUEの平均値（または中央値）を算出

Rでの実行例

1. “expand.grid” を用いてバランスデータの作成

```
new.dat <- expand.grid(Year=unique(dat$Year), Month=unique(dat$Month))
```

総当たりデータを作成
してくれる関数

2. “predict” 関数を使用してCPUEの予測

```
new.dat$pred.cpue <- exp(predict(interact3, newdata=new.dat))
```

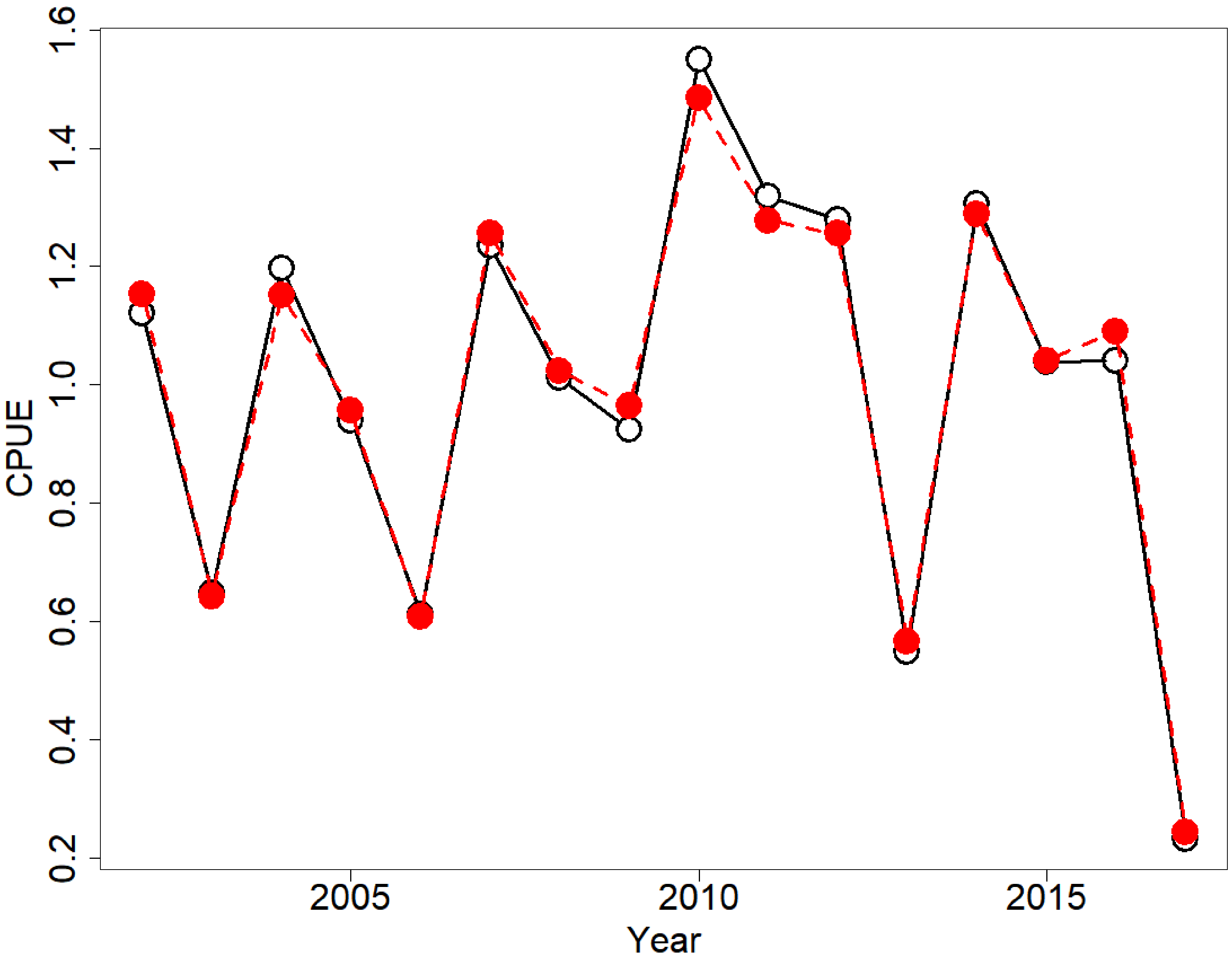
GLM
object

予測用データ
を使用

3. “apply” 関数を使用して平均CPUEの計算

```
standardCPUE <- tapply(new.dat$pred.cpue, new.dat$Year, mean)
standardCPUE <- standardCPUE / mean(standardCPUE)
```

年トレンド



黒：ノミナルCPUE

赤：標準化CPUE

信頼区間の推定：データリサンプリング

- サンプル数が等しくなるようにデータを再抽出し、モデルを当てはめ標準化CPUEを計算する作業を繰り返す

元データ

ID	Year	Month	CPUE
1	2002	1	26
2	2002	2	33
…	…	…	…
192	2017	12	11

リサンプルデータ

ID	Year	Month	CPUE
1	2002	1	26
4	2002	4	46
…	…	…	…
190	2017	10	8

- GLMによる推定
- 年トレンドの算出
- これを繰り返す

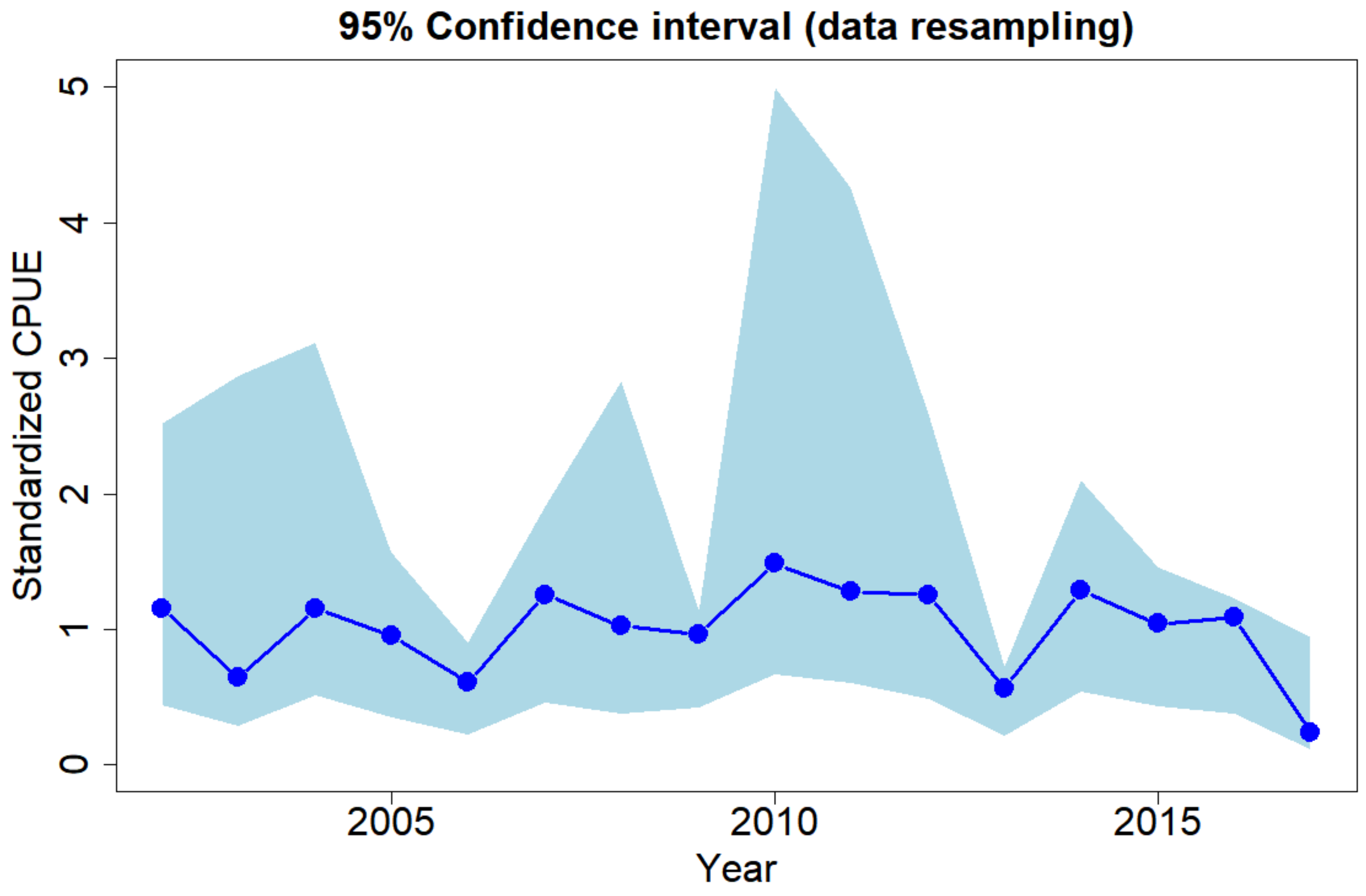
…

- 説明変数も変わる
- パラメータ推定＋説明変数の不確実性を評価

```r
boot.cpue <- sapply(1:nsim, function(i){
  boot.dat <- dat[sample(1:nrow(dat),nrow(dat),replace=TRUE),] #データのリサンプリング
  boot.res <- update(interact3, data=boot.dat) #GLMの更新
  pred.cpue <- exp(predict(boot.res, newdata=new.dat)) #CPUEの予測
  staCPUE <- tapply(pred.cpue, new.dat$Year, mean) #標準化CPUEの導出
  staCPUE / mean(staCPUE)
})
```

信頼区間が広くなりやすい

Rでの実行例



信頼区間の推定：残差のリサンプリング

- 残差をリサンプリングし、予測値から観測値を生成し、モデルを当てはめ標準化CPUEを計算する作業を繰り返す

元データ+予測値+残差

Year	Month	y	$\hat{y}$	ε
2002	1	3.3	3.6	−0.3
2002	2	3.5	3.5	0.0
…	…	…	…	…
2017	12	2.4	2.9	−0.5

残差のみ
resampling

リサンプルデータ

ε	$\hat{y}$	y	Year	Month
0.4	3.6	4.0	2002	1
−0.5	3.5	4.0	2002	2
…	…	…	…	…
0.0	2.9	2.9	2017	12

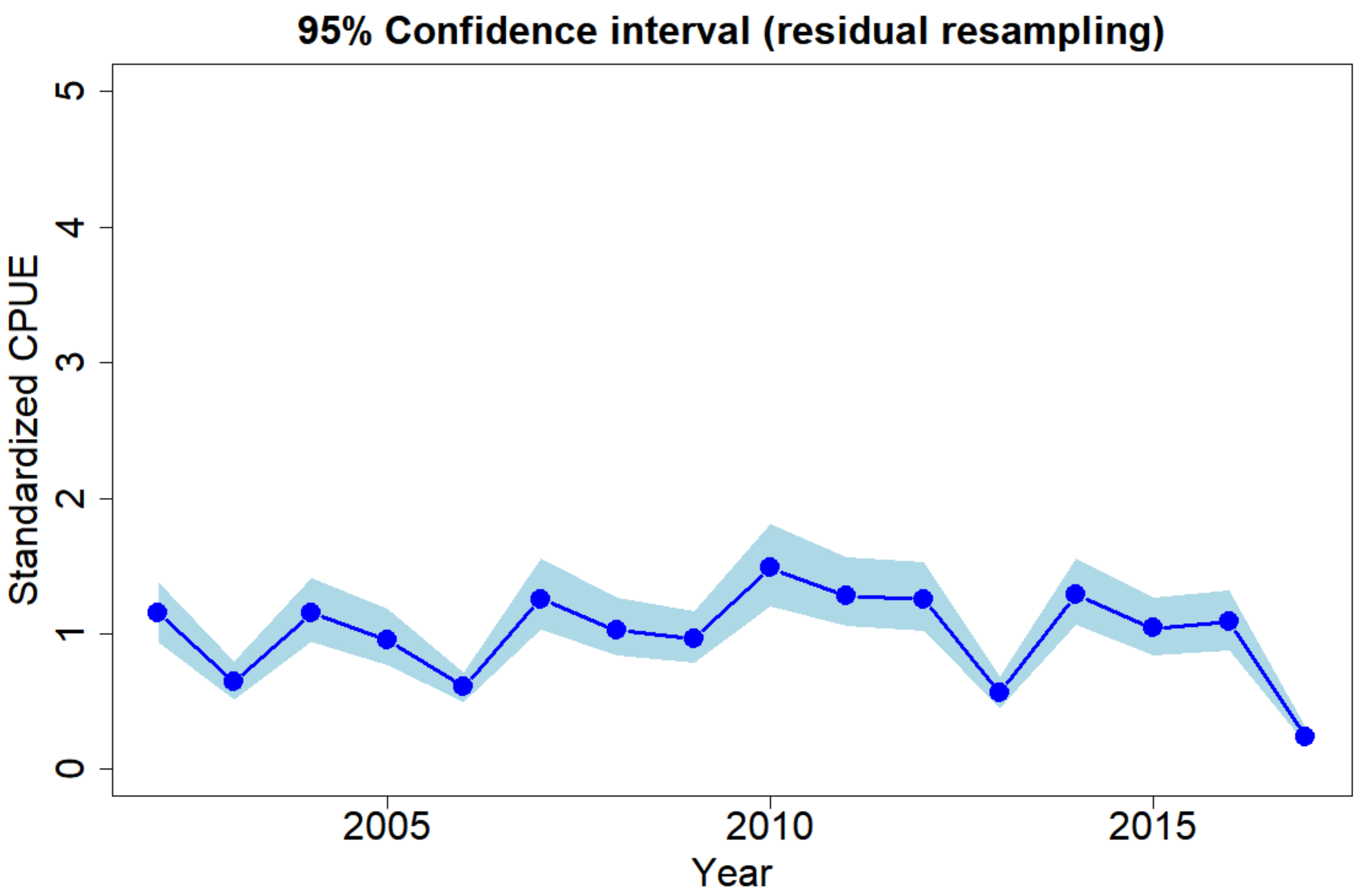
- 説明変数は変わらない
- パラメータ推定の不確実性を評価

- 予測値に残差を足す
- 説明変数を使ってGLM
- これを繰り返す

```
boot.cpue2 <- sapply(1:nsim, function(i){
  boot.resid <- sample(interact3$resid,nrow(dat),replace=TRUE) #残差のリサンプリング
  boot.dat <- dat
  boot.dat$CPUE <- exp(interact3$fitted.values + boot.resid) #予測値に残差を足す
  boot.res <- update(interact3, data=boot.dat) #後は同じ
  pred.cpue <- exp(predict(boot.res, newdata=new.dat))
  staCPUE <- tapply(pred.cpue, new.dat$Year, mean)
  staCPUE / mean(staCPUE)
})
```

（対数）正規分布以外、特に離散変数の場合に難しい

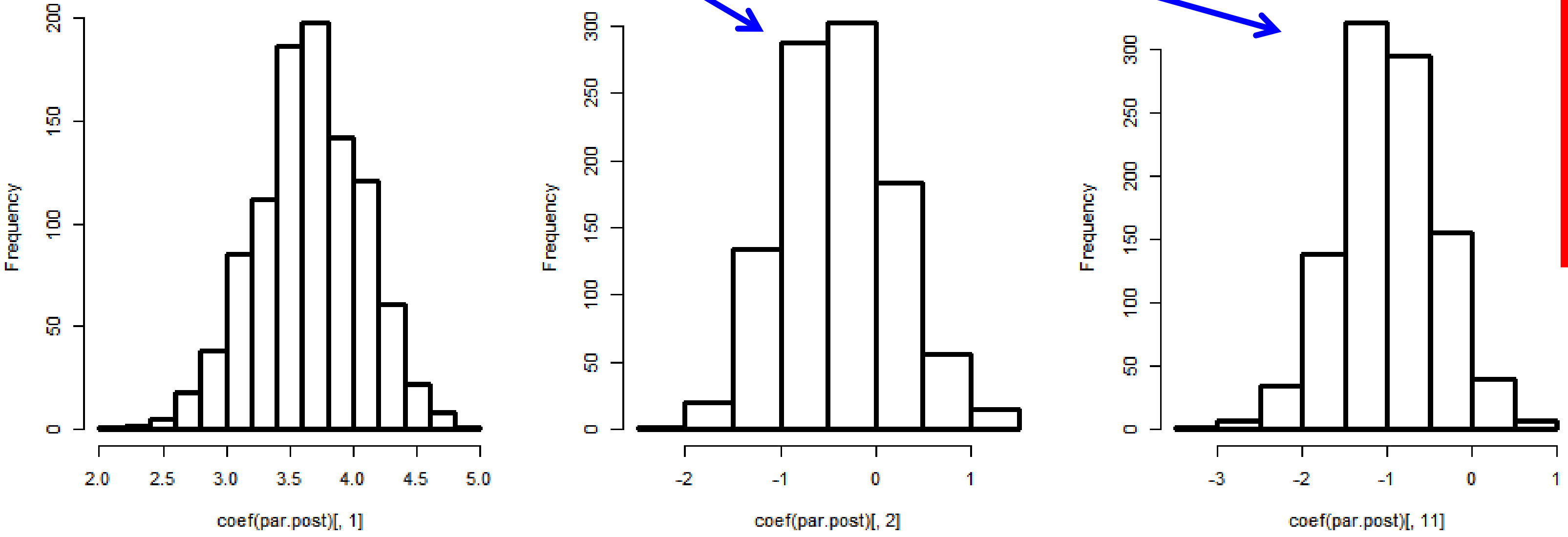
Rでの実行例



信頼区間の推定：事後分布の推定

- （ベイズ推定的に）MCMCを）行い、パラメータの事後分布を得る（package”arm”のsim()を使う）

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon,$$



事後分布から係数セット生成

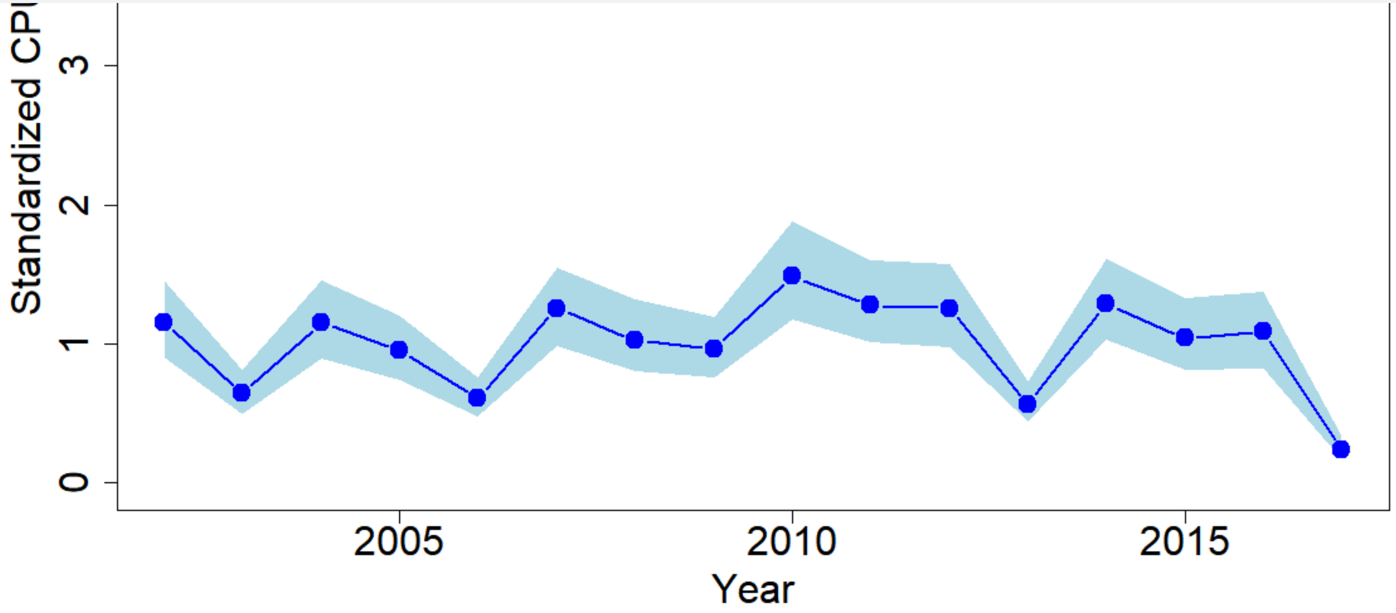
$$\boxed{\begin{aligned} y &= \alpha^{(1)} + \beta_1^{(1)} x_1 + \beta_2^{(1)} x_2 + \cdots \\ y &= \alpha^{(2)} + \beta_1^{(2)} x_1 + \beta_2^{(2)} x_2 + \cdots \\ y &= \alpha^{(3)} + \beta_1^{(3)} x_1 + \beta_2^{(3)} x_2 + \cdots \\ &\vdots \end{aligned}}$$

標準化CPUEを計算

```
library(arm)  #package"arm"の読み込み
par.post <- sim(interact3, n.sims = 1000) #事後分布の計算
boot.coef <- coef(par.post)  #係数の取り出し
new.dat$CPUE <- rep(1,length(new.dat))
sim.dat <- model.matrix(interact3$formula, data = new.dat) %*% t(boot.coef)
#予測値の計算
#model.matrixはカテゴリ変数をダミー変数化したり係数にかける値を算出する関数
sim.dat <- exp(sim.dat)
boot.cpue3 <- apply(sim.dat,2, function(x) {
  staCPUE <- tapply(x, new.dat$Year, mean)
  staCPUE / mean(staCPUE)
  })
```

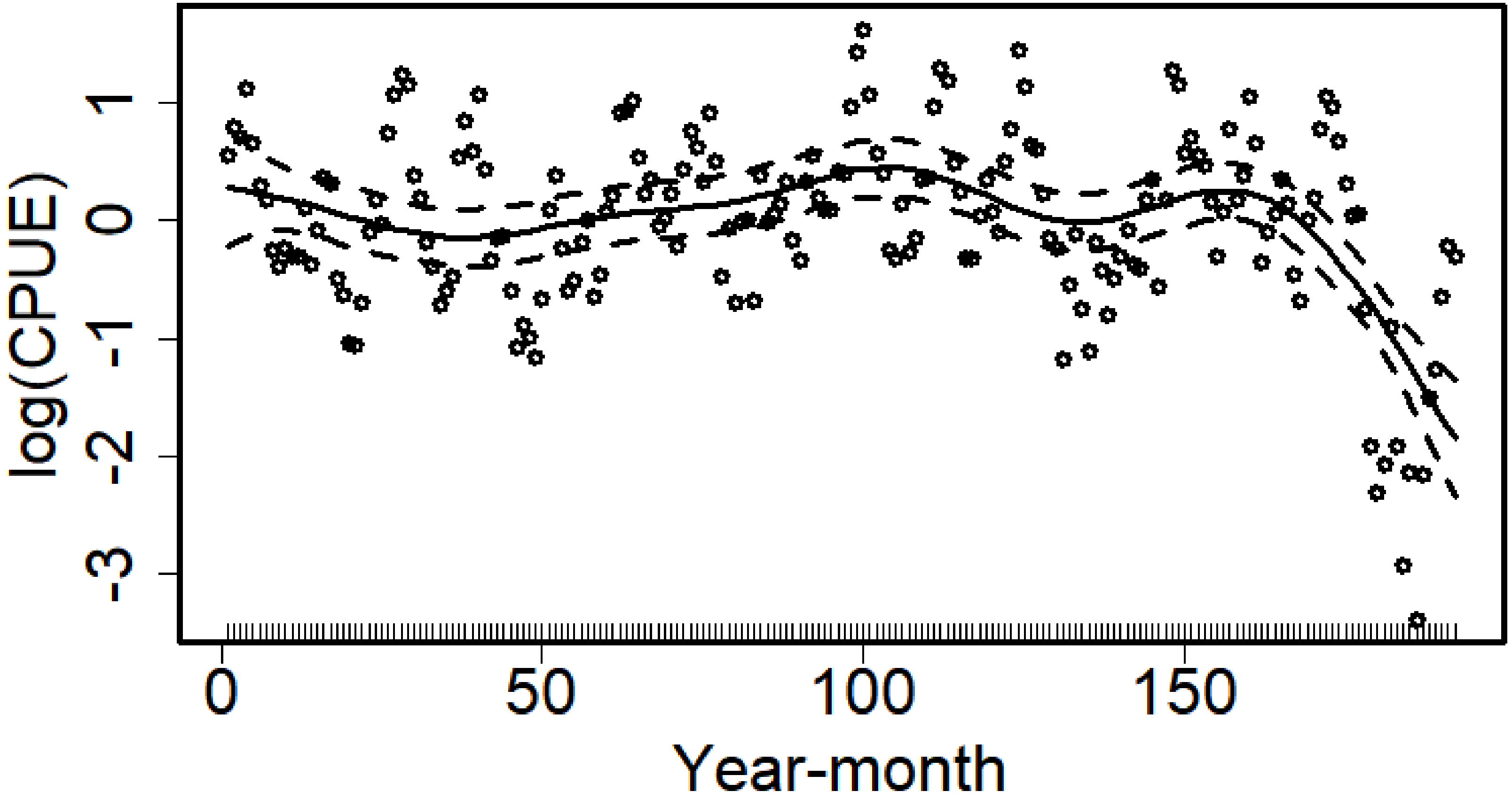
sim()はglmやlmerに対して使える（正規分布以外もOK）

Rでの実行例



一般化加法モデル (GAM)

- 連続変数に対してグネグネした関数を用いてモデル（平滑化）
- 線形ではない式を足し合わせたモデル（加法モデル）



$$y = f(x) + \varepsilon$$

最小

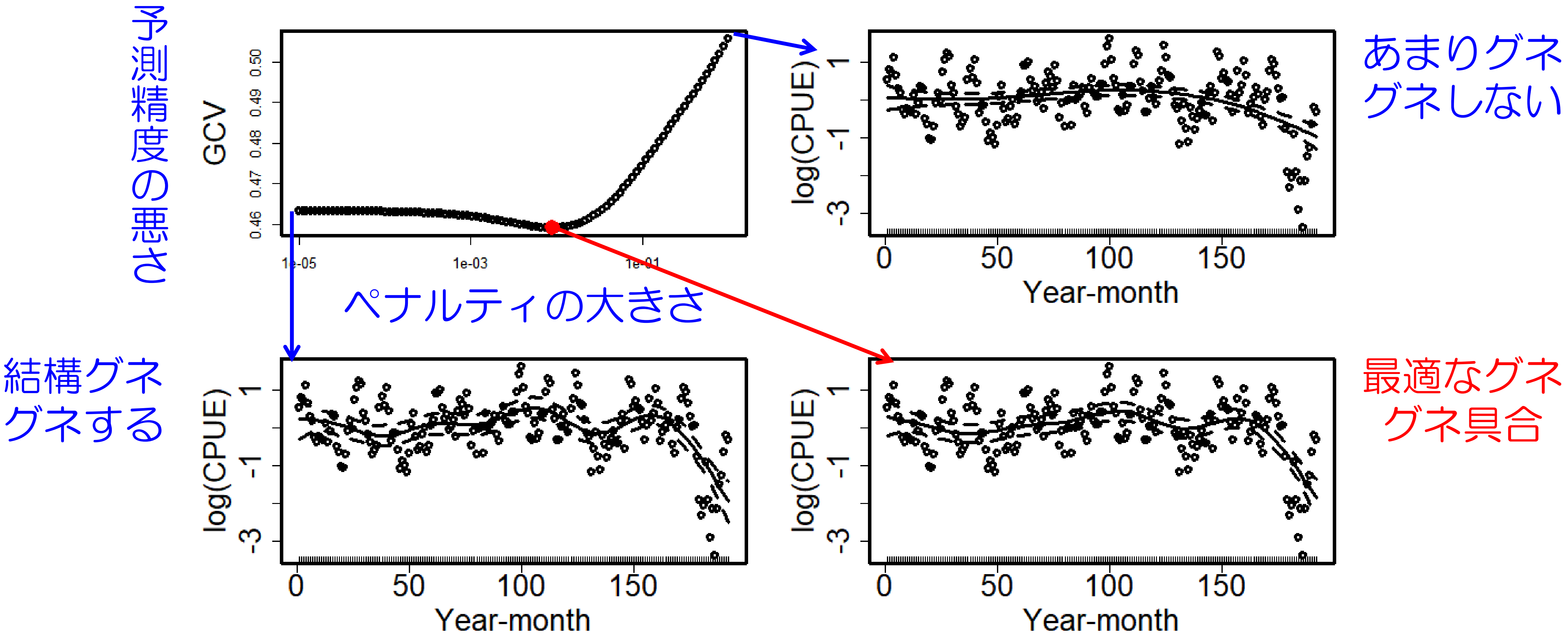
$$-\log(L) + \lambda \int f''(x)dx$$

グネグネ度

交差検証による予測精度を基に決定

Penalized likelihood

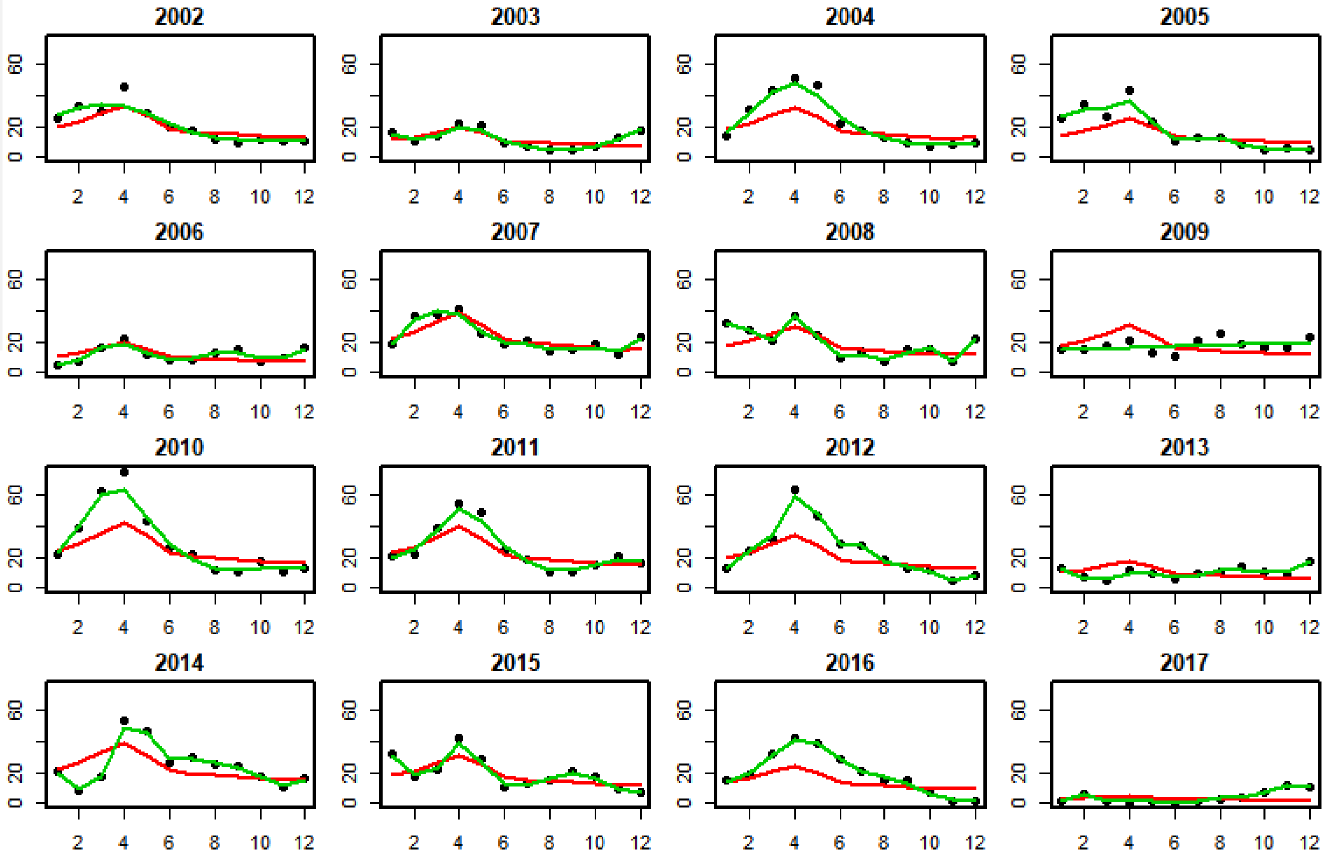
ペナルティの大きさの影響



GAMの解析：交互作用なし・あり

```
> gam <- gam(log(CPUE) ~ Year + s(Month), data = dat) #no interaction
> gam2 <- gam(log(CPUE) ~ Year + s(Month, by=Year), data = dat) # interaction
> AIC(gam,gam2)
            df        AIC
gam   23.36517 337.03791
gam2 119.18574 -46.01879
> c(gam$gcv.ubre,gam2$gcv.ubre)
    GCV.Cp     GCV.Cp
0.34021964 0.09006807
```

年ごとに平滑化



交互作用ありの方がよい（季節変動パターンが年によって変わってる）

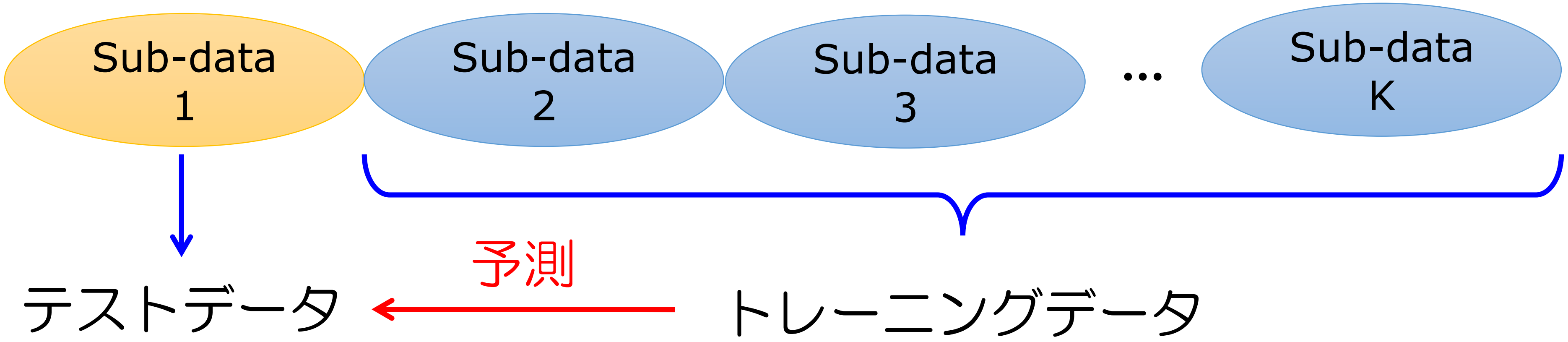
赤：交互作用なし

緑：交互作用あり

交差検証 (cross validation)

- GLMは最尤法で、GAMはペナルティ付き最尤法なのでAIC等では比較できない
- K-分割交差検証によって異なるモデル間の比較が可能である

1. データをK個に分割し、1つのサブデータを除いたデータでパラメータ推定



2. テストデータを予測し、予測精度をRMSE (root mean squared error)などで測る

3. K回繰り返す

GLM vs. GAM

```
> nfolds <- 10 #10-fold cross validation
> dat$id <- sample(1:nfolds, nrow(dat), replace=TRUE)
> rmse <- sapply(1:nfolds, function(i) {
+   train.dat <- subset(dat, id != i)
+   test.dat <- subset(dat, id == i)
+   glm.cv <- predict(update(interact3, data=train.dat),newdata = test.dat)
+   glm.rmse <- sqrt(mean((log(test.dat$CPUE) - glm.cv)^2))
+   gam.cv <- predict(update(gam2, data=train.dat),newdata = test.dat)
+   gam.rmse <- sqrt(mean((log(test.dat$CPUE) - gam.cv)^2))
+   c(glm.rmse,gam.rmse)
+   })
> rmse
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.5843955 0.4890191 0.5507541 0.5466020 0.4347081 0.4318414 0.5496188
[2,] 0.4057898 0.4125000 0.3239788 0.7122763 1.9777294 0.3819786 0.5984412
          [,8]      [,9]     [,10]
[1,] 0.5056577 0.5627170 0.6045362
[2,] 0.2663090 0.4960411 0.4800705
> rowMeans(rmse)
[1] 0.5259850 0.6055114
```

平均したらGLMの方がよい

GAMはすごく悪いときがある

ホームワーク③

- GAMの結果から、年トレンドを算出してみよう
- さらに、残差リサンプリングによって、その信頼区間も求めてみよう

実例２：サワラの定置網の漁獲量データ

- 2006年以降の大型定置網における銘柄：サイズ大の（1統あたりの）漁獲量を予測してみよう

```
> head(dat.l)
      year month day gyosyu catch netnumber depth area
9358  2006     2  16    330    18        37    50    4
41776 2006     2  16    330     3        45    36    8
11893 2006     2  17    330     3        37    50    4
21082 2006     2  17    330     3        53    55   12
37787 2006     2  17    330    10        47    43    9
42164 2006     2  18    330    10        45    36    8
```

- netnumberが漁場を表しており（小さいほど東側）、areaはnetnumberから番号（1～16）を割り当てたもの

一般化線形混合モデル：GLMM

$$\log(catch_{y,m,a}) = \alpha + \beta_y + \gamma_m + \varphi_a + \varepsilon_{y,m,a}$$

年の効果　月の効果　場所の効果

GLM: $\varphi = (\varphi_2, \varphi_3, \ldots, \varphi_{16})$ という15個のパラメータを推定

GLMM: $\varphi_a \sim \mathrm{Normal}(0, \tau^2)$ として、一つのパラメータ（σ_a^2）を推定

- lme4パッケージのlmer()関数を使用

```
library(lme4) #lme4パッケージの読み込み
glmm.res <- lmer(log(catch) ~ year + month + (1|area), data=dat.l)
```

切片がランダム効果で変わるという意味

固定効果とランダム効果

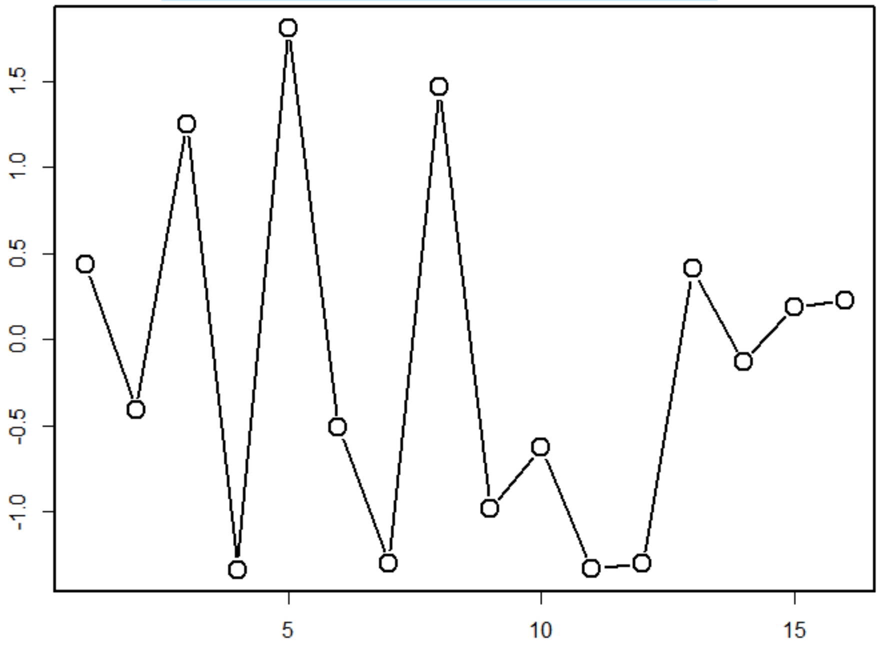
- カテゴリカル変数の一つ一つの水準に関心があるなら固定効果
- 水準による「ばらつき」を考慮したい/関心があるならランダム効果
- 混合モデルは、各水準の効果はランダム効果として推定するので、固定効果として推定するパラメータ数は少なく、節約的
- ただし、尤度の算出に積分が必要なので、計算が大変

$$L_{y,m,a} = \int \underbrace{p\left(y_{y,m,a} \middle| \alpha, \beta_y, \gamma_m, \varphi_a\right)}_{\text{あるδでの条件付き尤度}} \times \underbrace{p(\varphi_a | \sigma_a)}_{\text{そのδの尤度}} d\varphi_a \longrightarrow \text{δが消える}$$

- そのため、複雑なランダム効果の場合、ベイズ推定やTMB（自動微分）が有効

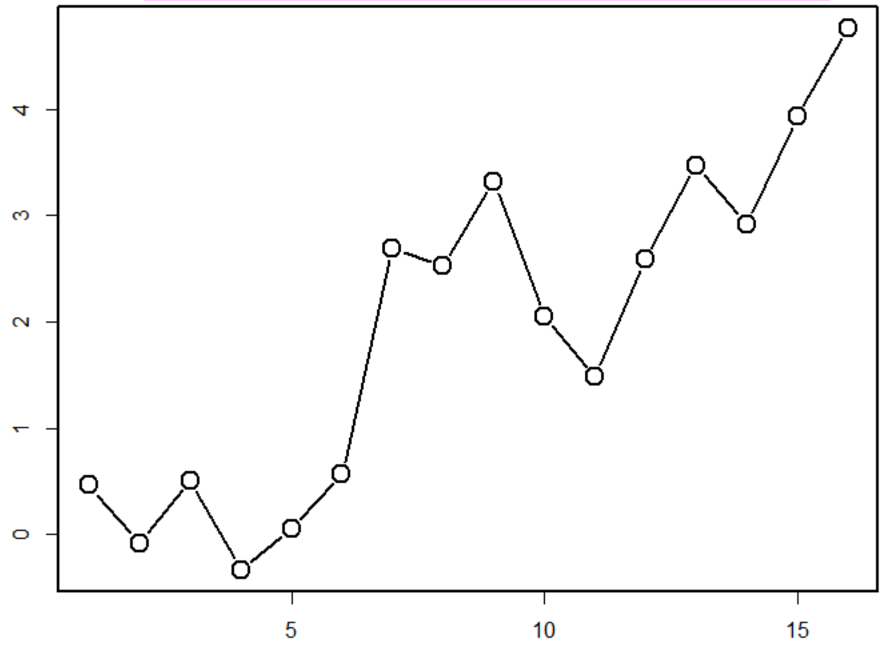
空間自己相関

各地点がランダム
（自己相関なし）



場所

近い地点間は似ている
（自己相関あり）



場所

空間自己相関したモデリングの方が予測性能が上がる可能性がある

条件付き自己回帰モデル (CAR model)

- ある地点のランダム効果が近傍のランダム効果の影響を受ける

地点k以外

自己相関なしのときの分散

$$\phi_k | \boldsymbol{\phi}_{-k}, \mathbf{W}, \tau^2, \rho \sim \mathrm{N}\left(\frac{\rho \sum_{i=1}^{K} w_{ki} \phi_i}{\rho \sum_{i=1}^{K} w_{ki} + 1 - \rho}, \frac{\tau^2}{\rho \sum_{i=1}^{K} w_{ki} + 1 - \rho} \right)$$

地点間の影響の強さを表す行列

$$\mathbf{W} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & 0 \\ 0 & 1 & 0 & \ddots & \vdots \\ \vdots & 0 & \cdots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

近傍間のみ1であとは0

$\rho = 0$: 平均0
$\rho = 1$: 近傍の平均

自己相関係数と近傍セルが多いほど、分散が小さくなる

自己相関係数

$\rho = 0$: 自己相関なし
$\rho = 1$: 自己相関のみ

Rでの解析

- library(CARBayes) で解析できる

```
library(CARBayes)
```

```
n.area <- length(unique(dat.l$area))
W <- matrix(0, ncol = n.area, nrow = n.area)
for (i in 1:(n.area-1)) W[i,i+1] <- W[i+1,i] <- 1
```

各要素が0の行列を作成

対角成分の隣に1を代入

```
car.bayes <- S.CARmultilevel(formula = log(catch) ~ year + month, data = dat.l,
                             family = "gaussian", #glmのように式を指定
                             W = W, ind.area = as.numeric(dat.l$area),
                             #近傍間の重みの関係を表す行列と各データの地点を指定
                             burnin=2000, n.sample=5000, thin=3)
```

MCMCサンプリングを開始してから捨てるステップ数

MCMCサンプリングを行うステップ数

３つおきにサンプリング

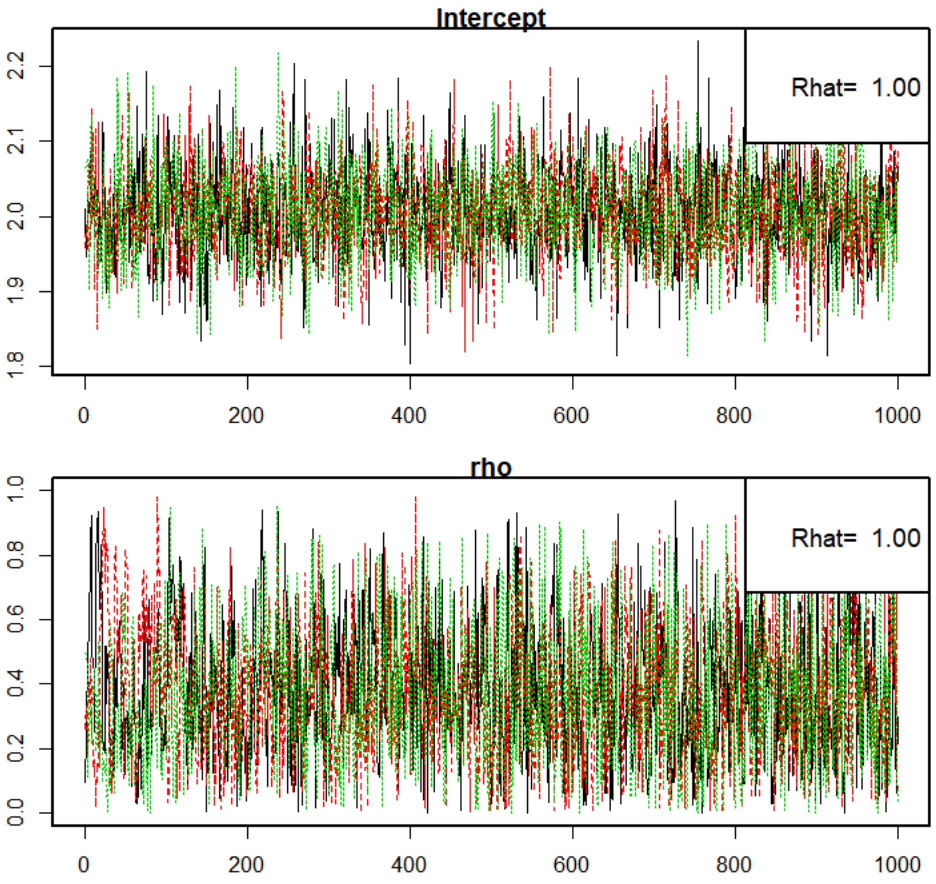
推定パラメータの収束の判断手法・基準

- 複数回、MCMCサンプリングを実行し、サンプリング列内の分散と列間の分散を比較する

$$\hat{R} = \sqrt{\frac{n-1}{n} + \frac{B}{nW}}$$

列間の分散

列内の分散



- $\hat{R} < 1.1$なら収束していると一応みなせる（必要条件）

CAR modelのモデル選択

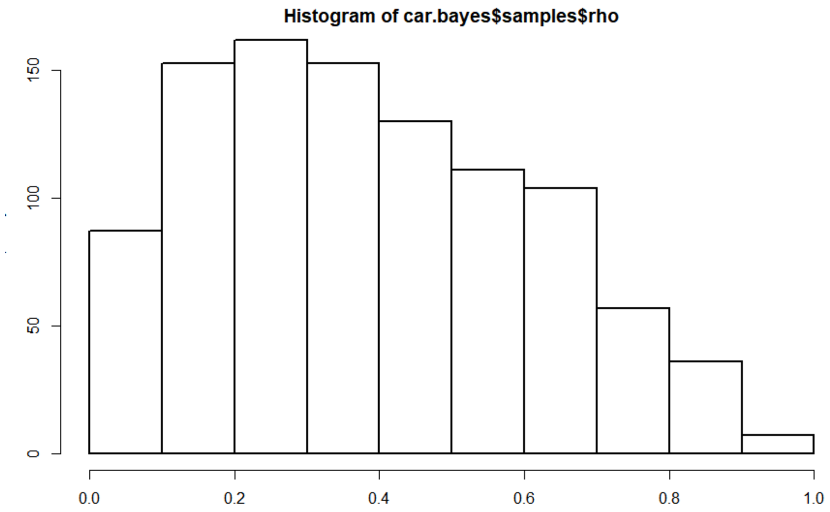
- $\rho = 0$ or 1の場合も解析できる

```
car.bayes.rho0 <- S.CARmultilevel(formula = log(catch) ~ year + month, data = dat.l,
                                  W = W, ind.area = as.numeric(dat.l$area),rho = 0,
                                  family = "gaussian", burnin=2000, n.sample=5000,
thin=3)
```

- WAIC（widely-applicable information criteria: ベイズ推定にも使える情報量基準）によって予測性能を評価

```
rho.est.WAIC    rho.0.WAIC    rho.1.WAIC
    43022.25      43020.92      43022.17
```

➡ 自己相関なしの方がWAIC低い



空間自己相関を年ごとに推定する

- 各地点のランダム効果は年によって変わっているかもしれない
- つまり、分布が年々変化している可能性がある
- 行列においてその年の近傍セルのみweightを1とすればよい

$$
\mathbf{W2} = \begin{array}{c} \begin{array}{cccc} \text{2006} & \text{07} & \dots & \text{18} \end{array} \\ \begin{pmatrix} \mathbf{W} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{W} \end{pmatrix} \end{array}
\qquad
\mathbf{W} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \ddots & 0 \\ 0 & 1 & 0 & \ddots & \vdots \\ \vdots & 0 & \cdots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}
$$

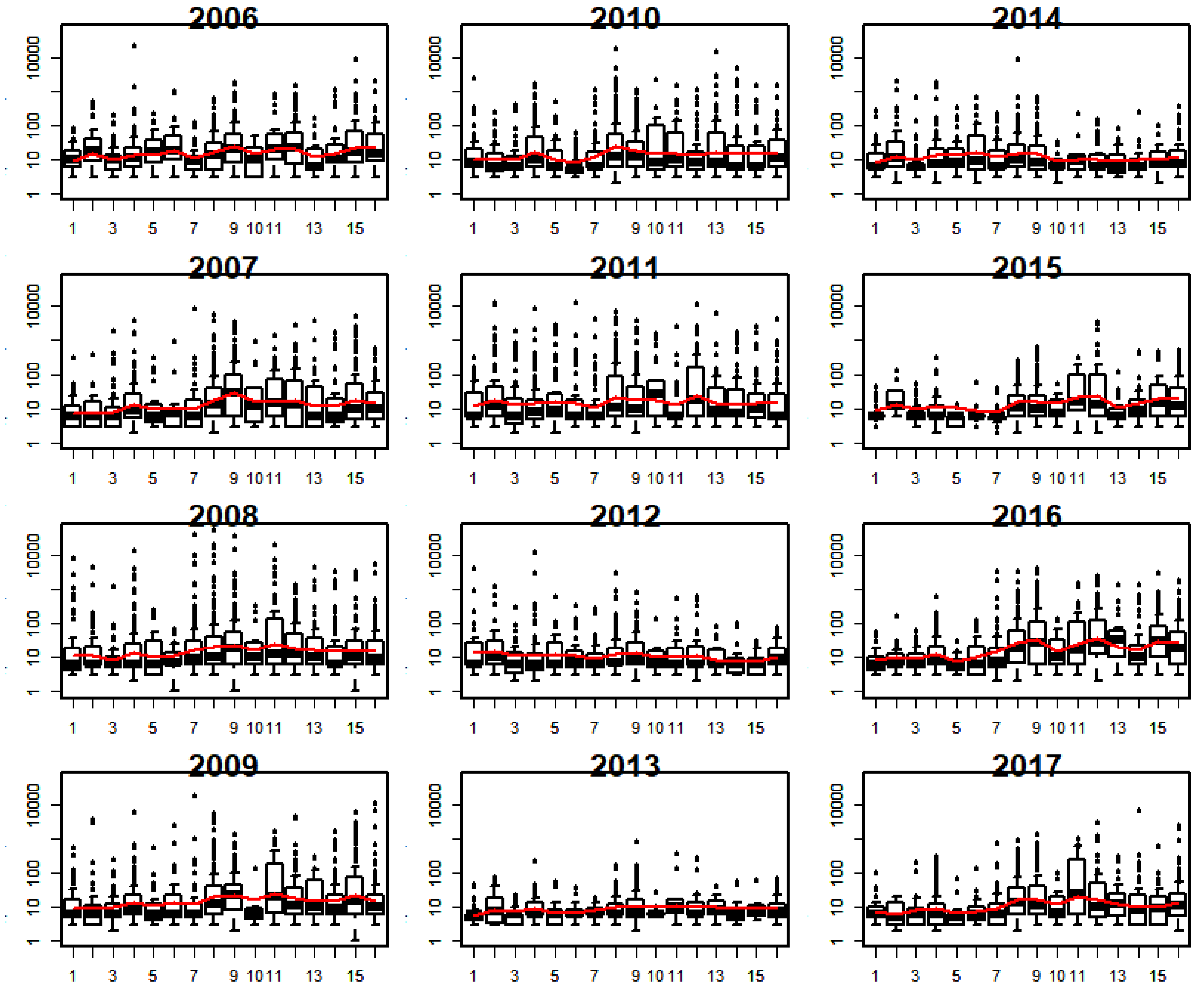
行列の行列

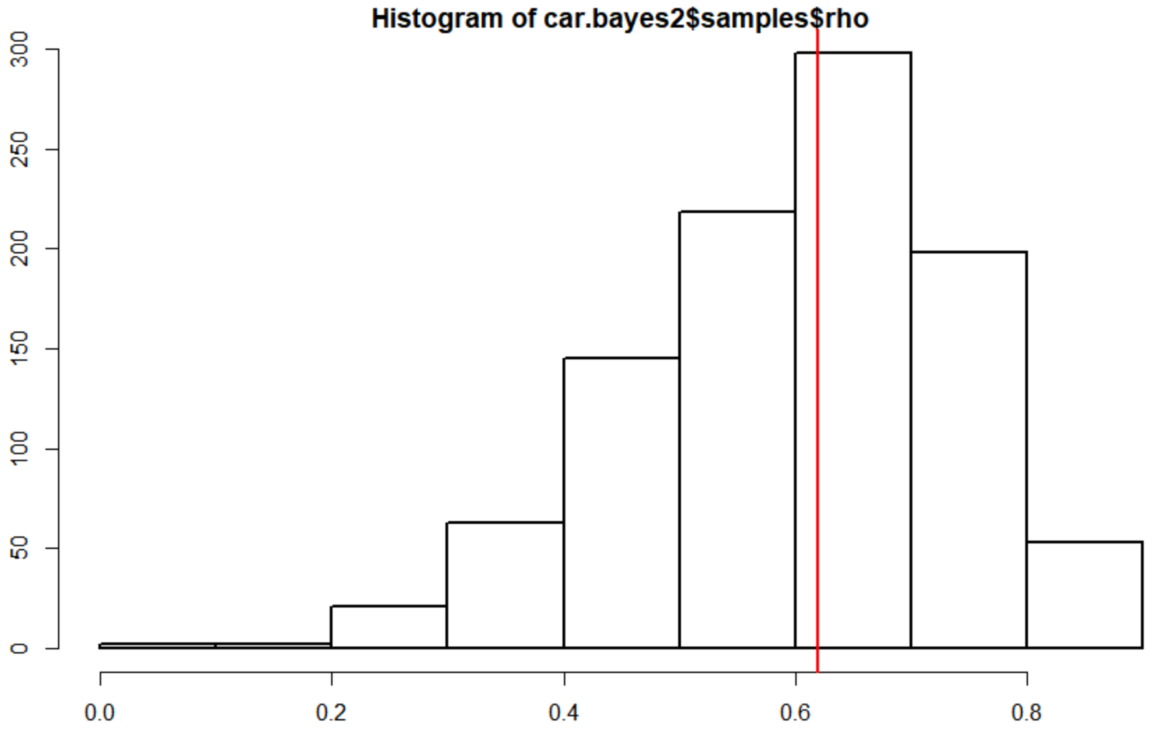
- 対角要素（同一年間の関係）に行列Wを代入の行列
- それ以外（異なる年間の関係）には空行列を代入

解析結果

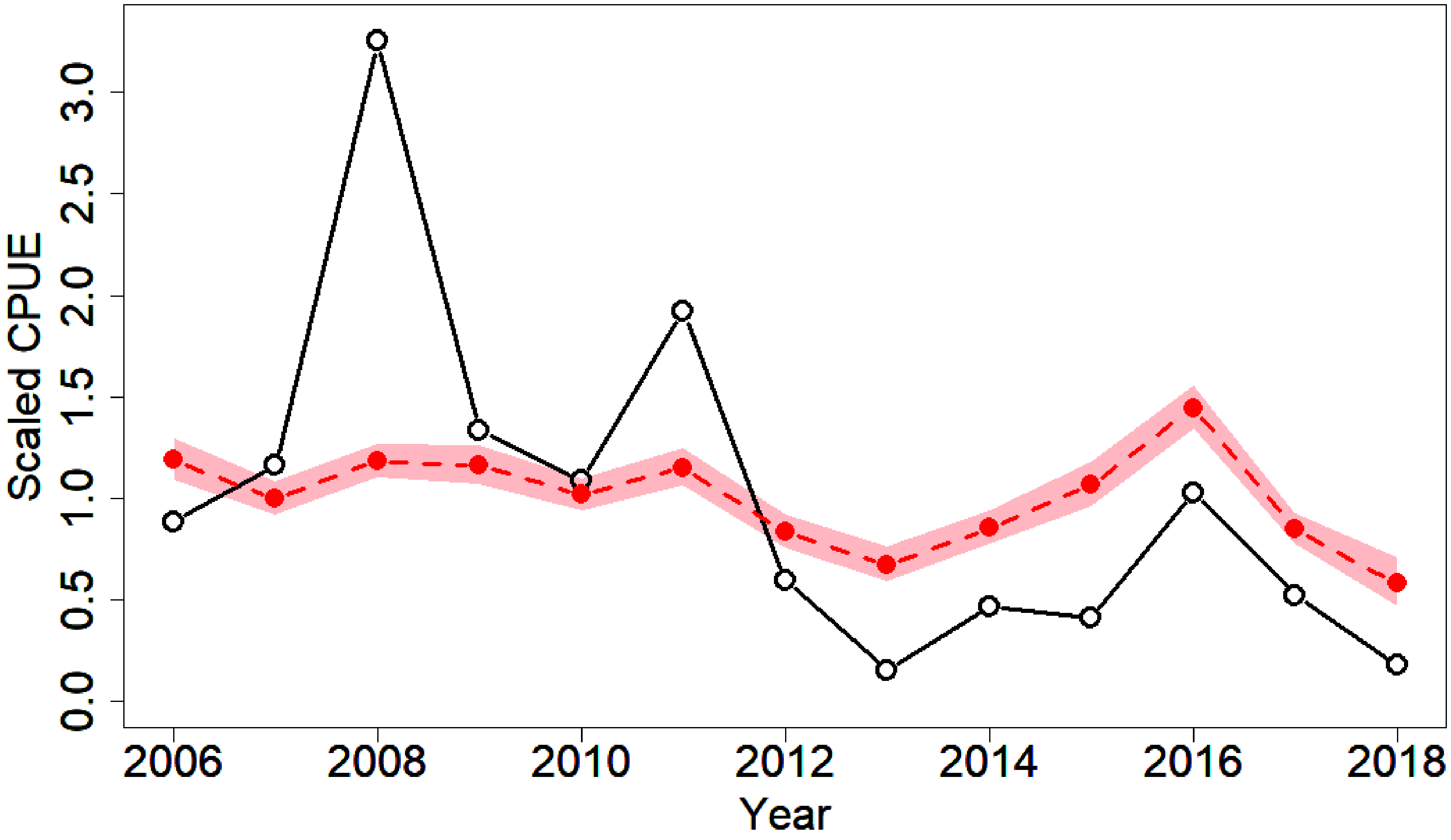
	WAIC
rho.est	43022.25
rho.0	43020.92
rho.est2	42880.50
rho.est2.rho0	42881.24

一応、WAIC最小





年トレンド



黒：ノミナルCPUE

赤：標準化CPUE

ノミナルに比べて
安定？

ホームワーク④

サワラの小型のデータでCar modelのベイズ推定を行い、標準化CPUEを計算してみよう

実例３：ミナミマグロの延縄CPUE

準備中…