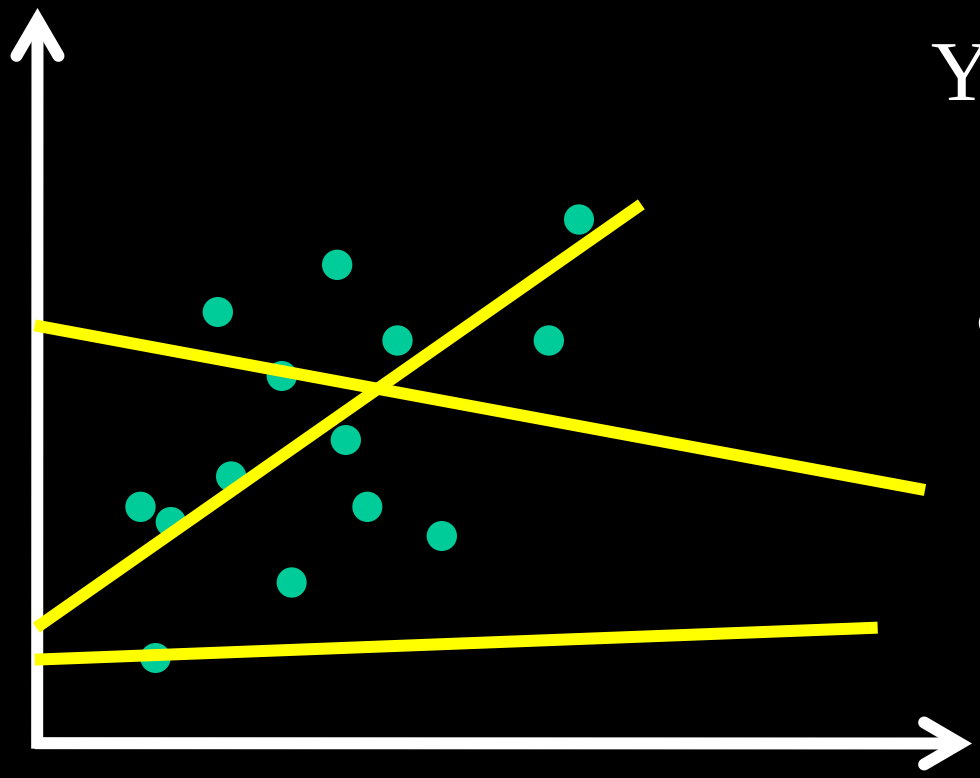
Chapter 4

# Parameterization

横浜国立大学　D1　柴田泰宙

# データに最も当てはまる直線は？



$$Y=a + b*X$$

a, bのような直線(モデル)の形を変える係数を**パラメータ**と呼ぶ

パラメータはどのように決めればよいのか？

# 数理モデルを構築するに当たっては、パラメータや定数が必要

それらを得るための方法

4.1. 生データそのまま

4.2. 生データを変換してから

4.3. 生データを変換して、モデルも試行錯誤

## 4.1. 生データそのまま

最も直接的(Y=Xなモデル)

(当たり前だが)パラメータは推定されるものであることに留意

## 4.2. 生データを変換してから

先行研究に従って変換(事例ではlog(Y)=log(X))

しかし、その変換がいつでも最良とは限らない

# 4.3. 生データを変換して、モデルも試行錯誤

観測値と予測値の差が最小になるように、
パラメータを調整

重みつき残差平方和

$$ModelCost = \sum_i \frac{\left(Modelvalue_i - Observedvalue_i\right)^2}{error_i}$$

(AIC的には・・・)

errorは、i番目のデータの持つ分散としている
(i番目のデータはj個のデータの平均値扱い)

データ多→分散小→Cost大→だから、ズレちゃだめ

# 4.3.1 線形回帰

## 基本形

$$y = a + bx$$

## 指数

$$y = ax^{b}$$

↓

$$\log(y) = \log(a) + b\log(x)$$

↓

$$A = a' + bC$$

## 逆数

$$y = \frac{ax}{b + x}$$

↓

$$\frac{1}{y} = \frac{b}{a}\frac{1}{x} + \frac{1}{a}$$

## 指数2(教科書と違う)

$$y = ax^{2} + bx + c$$

↓

$$y = aZ + bx + c$$

# 4. 3. 2 非線形回帰

非線形の定義は？

**線形でないこと**

線形モデルに比べ、最もModelCostの低い点が
見つけにくいので、初期値を変えて試す必要あり

本当に見つけにくい？

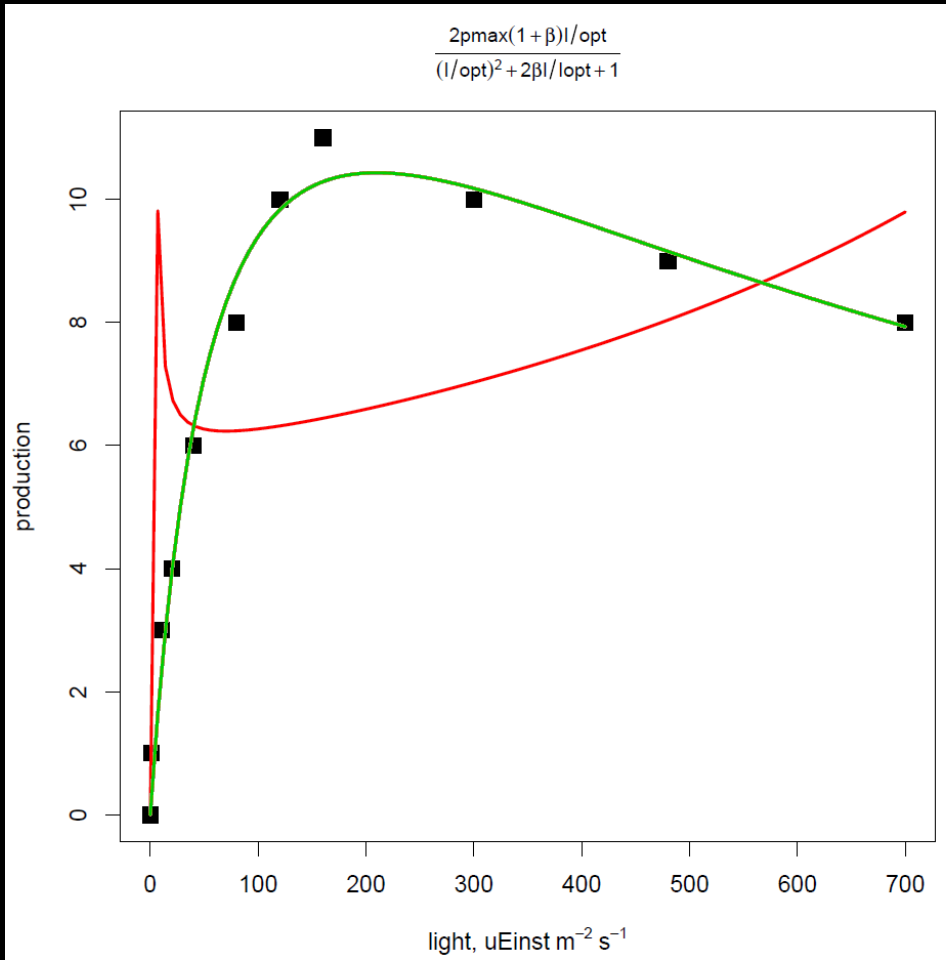
(例は当てはめただけでつまらないので勝手に)初期値を乱数で変えてみた

# 4. 4 ケーススタディ

## 4.4.1 P-Iカーブ
## ←本(p125)と一緒の値

```
[1,] 10.435  1.60  209.632
[2,] 10.435  1.60  209.633
[3,] 10.435  1.60  209.632
   ⋮
[198,] 10.435  1.60  209.633
[199,] 45.229 -1.60 -209.632
[200,]  5.284 12.12  -70.093
[201,] 10.435  1.60  209.632
   ⋮
```

初期値が悪いと確かに
良くないようだ



# 4.4.2 線形モデル vs 非線形モデル

P126の(4. 10)式を線形モデルと非線形モデル、両方でパラメータ推定します((4.10)式の導出はAppendix 1)

非線形モデルのモデル式

$$C(x) = C_0 e^{-\sqrt{\frac{\lambda}{D_b}}x}$$

線形モデルのモデル式

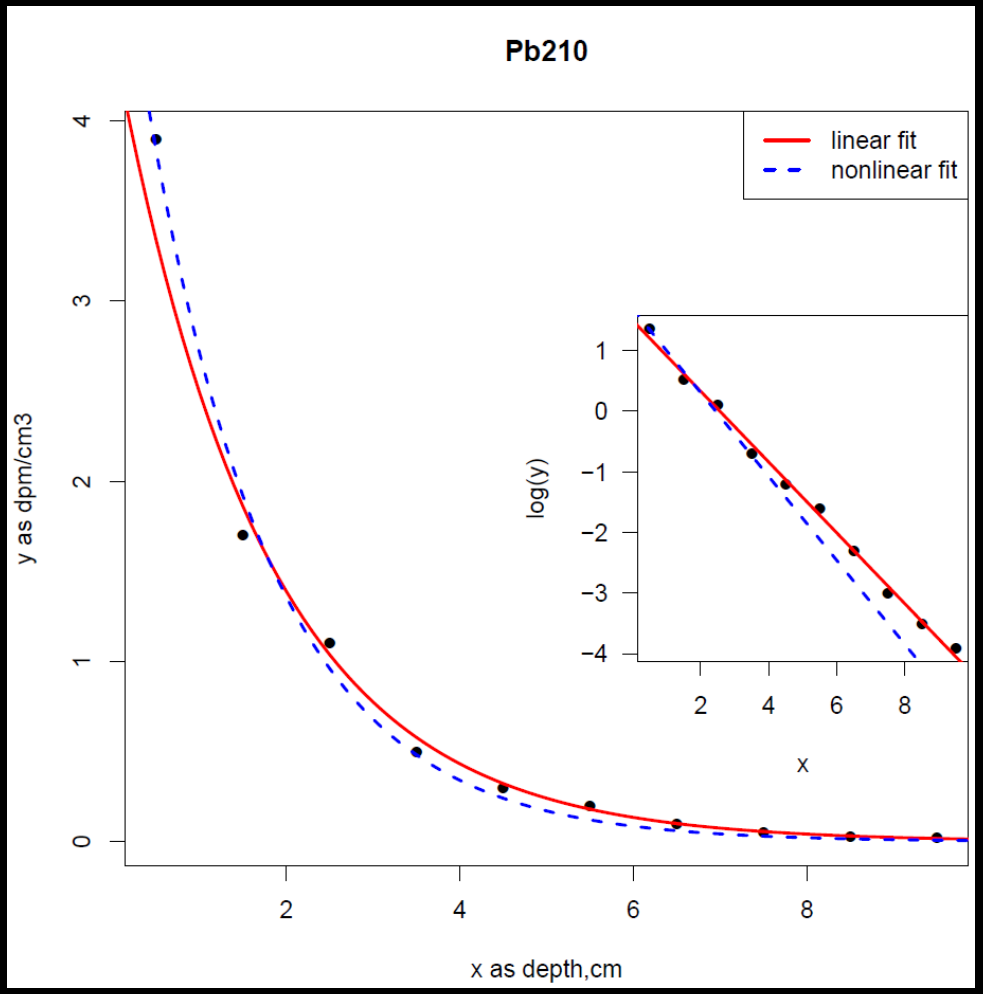
$$\log(C(x)) = a + bx$$

$$a = \log(C_0)$$

ただし $$b = -\sqrt{\frac{\lambda}{D_b}}$$ と置く

# 残差平方和は非線形回帰の方が小さかった

(教科書はx軸とy軸が逆転してて非常に見にくいので、勝手に変えました.Appendix2)




線形モデル(赤)

sum(LL$residuals^2)

[1] 0.1073133

非線形モデル(青)

sum(summary(fit)$residuals^2)

[1] 0.08434943

大きなxと小さなxで
当てはまりに差

# 4.4.3 疑似乱数によるパラメータ探索

手順

1. たくさんのパラメータ候補を発生させる
2. それらのパラメータを使ったModelCostを計算
3. パラメータをいくつか選んでその平均値を計算
4. もう一つ別のパラメータを選んで、それを軸に対称にあるパラメータを選択
5. そのパラメータのModelCostが一番悪いModelCostより小さければ採択

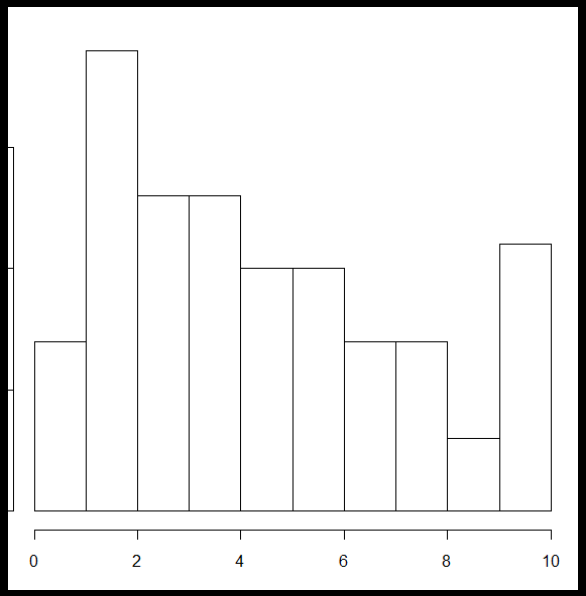
# 例(p133のやつはコードを打てばできるので)

Y = a + bXのパラメータaとbを推定してみる(Appendix3)

1. たくさんのパラメータ候補を発生させる

```
a <- 5
b <- 3

a2 <- runif(100, 0, 10)
b2 <- runif(100, 0, 10)
```




2. それらのパラメータを使ったModelCostを計算

この場合、100個のModelCostが計算される

3. パラメータをいくつか選んでその平均値を計算

```
a_mean <- mean(sample(a2, 3))
```

4. もう一つ別のパラメータを選んで、それを軸に対称にあるパラメータを選択(例えば、a_meanが5だったとして、a_mirrorが3だとすると、2*5-3=7となり、3を中心に等しい距離(2)だけ移動したことになる。局所解にトラップされないエ夫だと思われる)

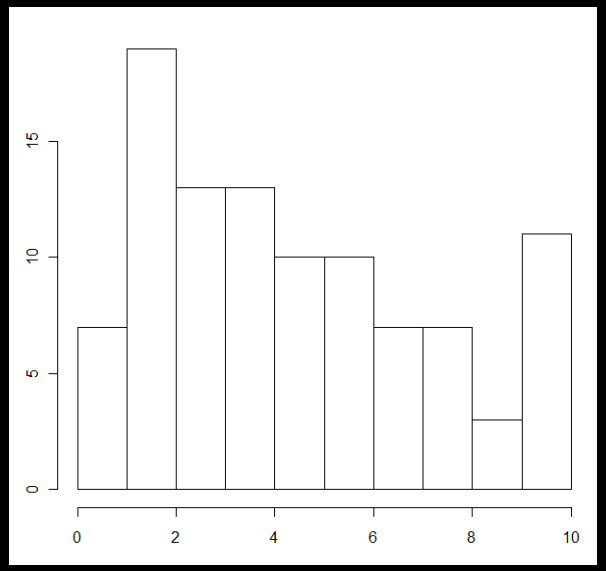
```
a_mirror <- sample(a2, 1)
```

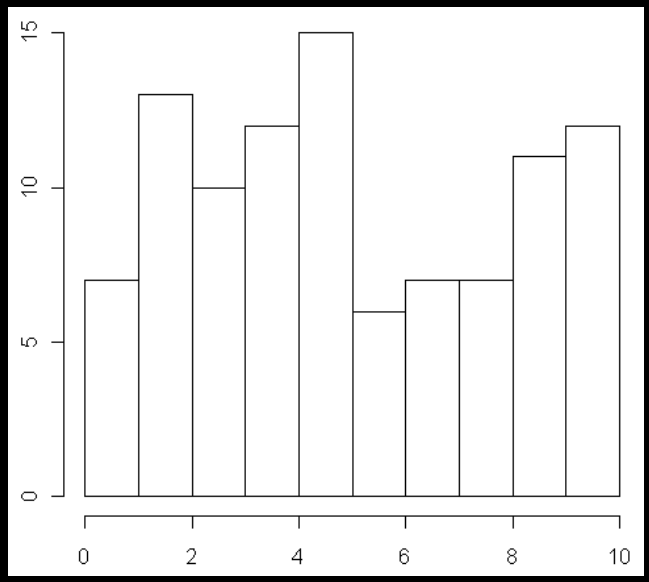
```
new_a <- 2*a_mean - a_mirror
```

5. そのパラメータのModelCostが一番悪いModelCostより小さければ採択

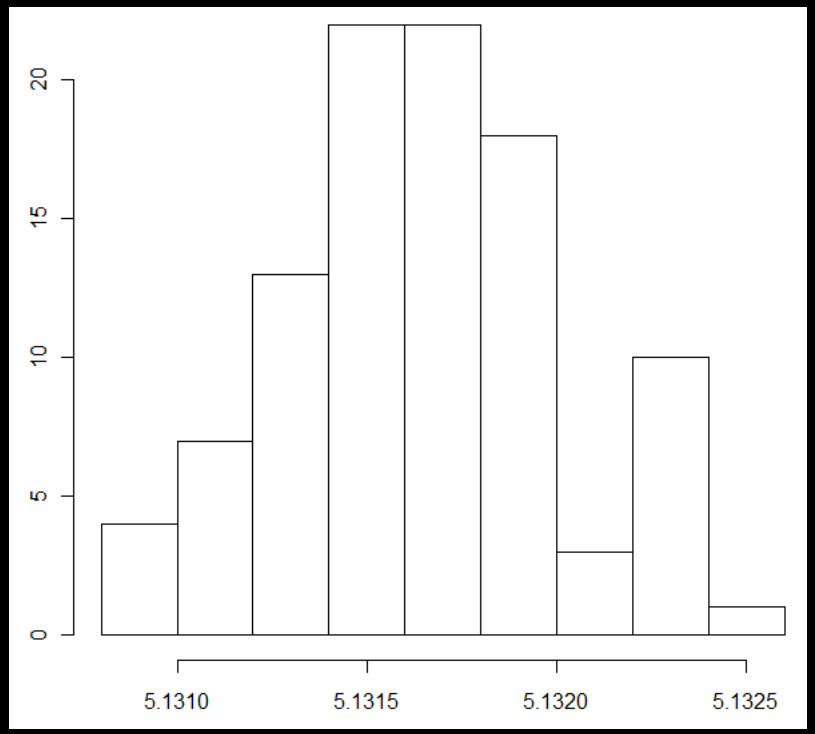
最適解だけ求めようとしていることに留意

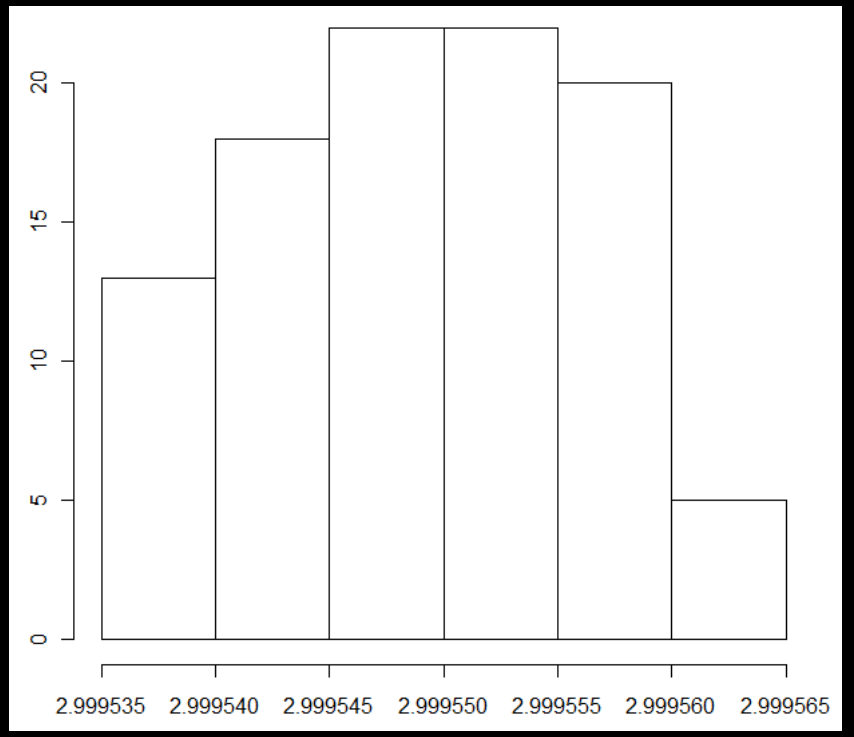
Before




After

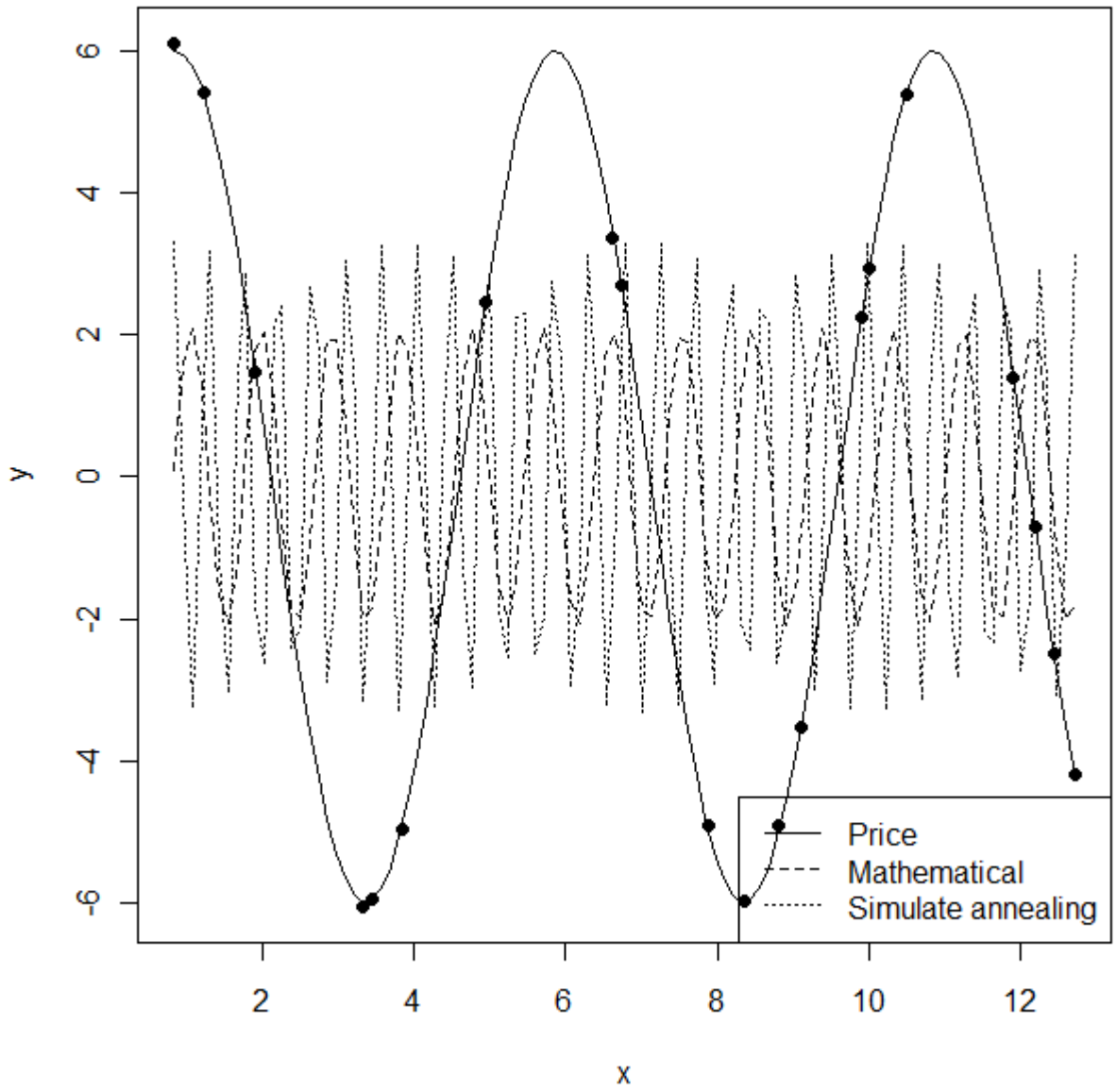



真の値5

真の値3

教科書のような図に
毎度はならない




複雑なモデル
でも大丈夫

ただし

得られるの
は点推定値

# 4.4.4 先ほどのアルゴリズムを使って、微分方程式のパラメータ推定

http://geos.ees.hokudai.ac.jp/noriki/Trap.html

セジメントトラップ

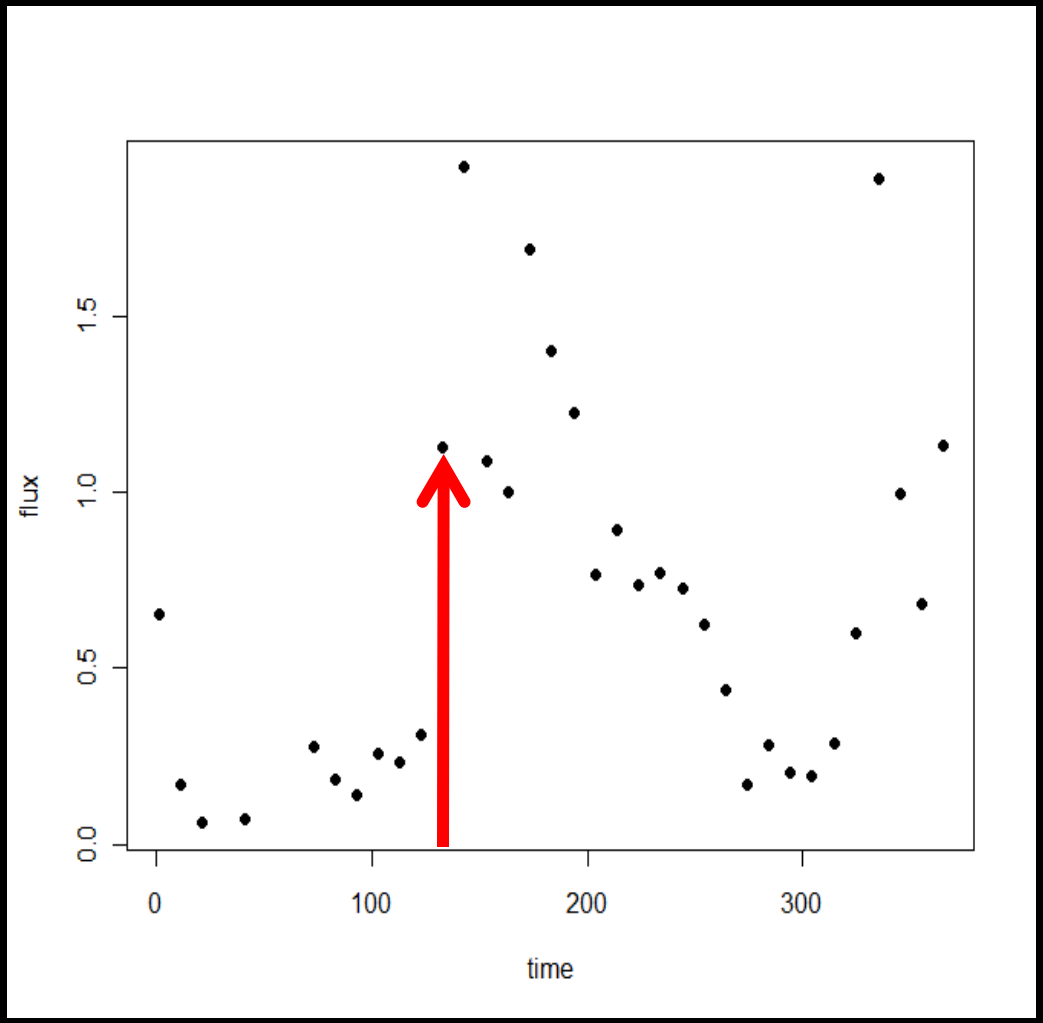
海水中を沈降する粒子を集める装置

今回は、有機炭素がどのように無機炭素に石灰化していくのかモデリングしてみる

<手持ちのデータは3つ>

- 有機・無機炭素量(*flux*)
- 酸素消費量(*oxycon*)
- 上記の時間軸

# 単位時間ごとのfluxの変化




矢印が変化量そのもの

$$\frac{dC}{dt} = flux$$

無機化した分を抜く

$$\frac{dC}{dt} = flux - k \cdot C$$

実は過少推定かも

$$\frac{dC}{dt} = mult \cdot flux - k \cdot C$$

# パラメータ推定の流れ

1. 観測値と微分方程式からの予測値の差を最小にするようにパラメータk, multを推定する.fluxの値はoffsetのように強制的に与えた

$$\left(k \cdot \mathrm{C} - \mathrm{oxycon}\right)^2$$

2. 得られたパラメータで、もう一度微分方程式に当てはめる

3. 予測値をプロットしてみて、実際のものと合っているか確認

# ちょっと説明:Rで微分方程式

- ode関数(Ordinary differential equation)を使用

<用法>

ode(初期値、時間、微分方程式、パラメータ)

例:個体群増加モデルを解いてみる(Appendix 4)

$$\frac{dN}{dt} = rN\frac{(K-N)}{K}$$

ロジスティックモデル

Nは個体数、rは増加率、Kは環境収容力

```
Growth_model <- function (t, state, pars) {
  with (as.list(c(state, pars)), {
dN <- r * N *(K - N)/K
return(list(dN, r * N *(K - N)/K))})
}
```

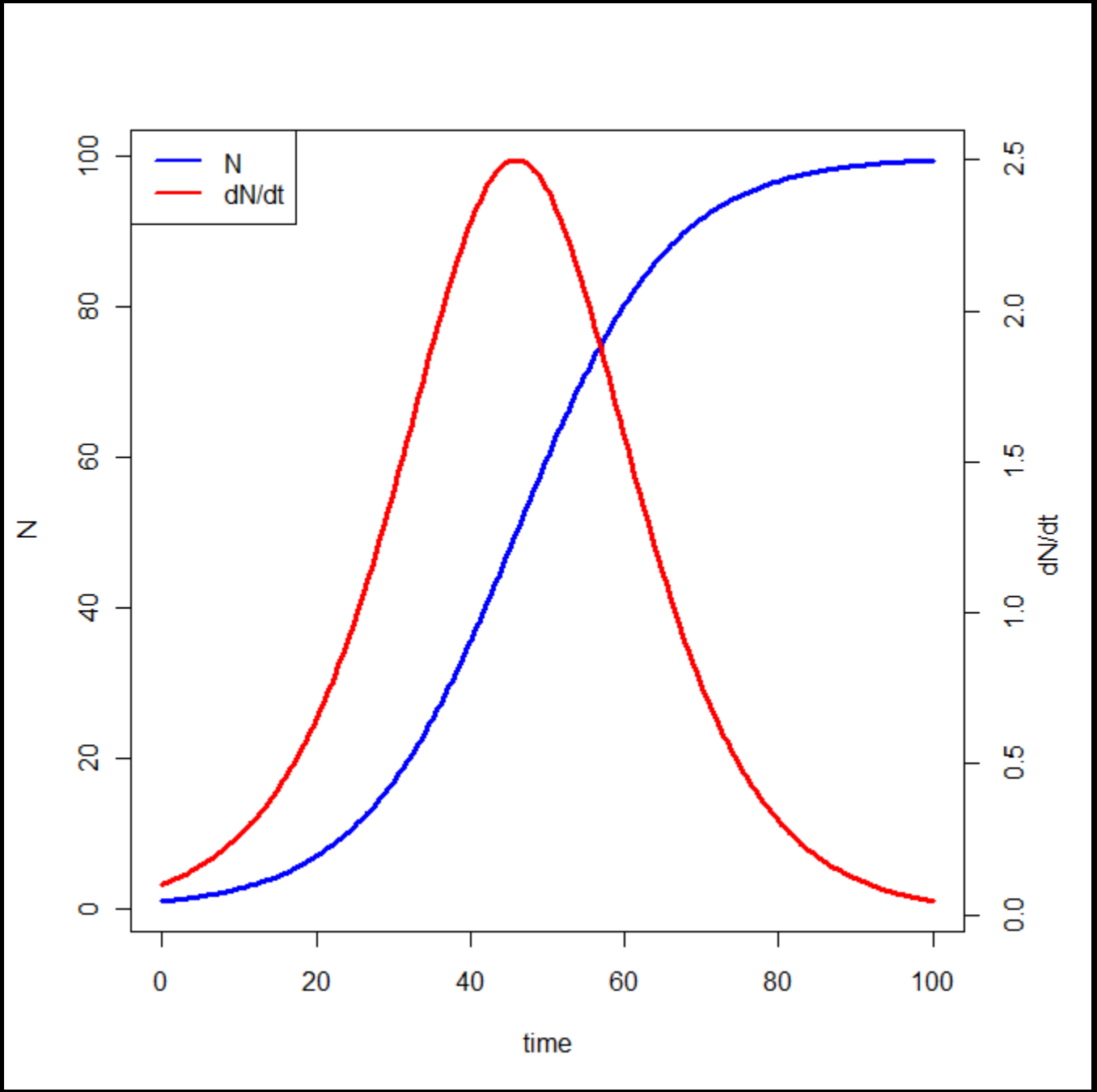
微分方程式作成

```
pars <- c(r=0.3, K=100)
ini <- c(N=1)
times <- seq(0, 100)
```

← 初期値とパラメータ

微分方程式解かせる

```
out <- as.data.frame(ode(func=Growth_model, y=ini,
                         parms=pars, times=times))
```

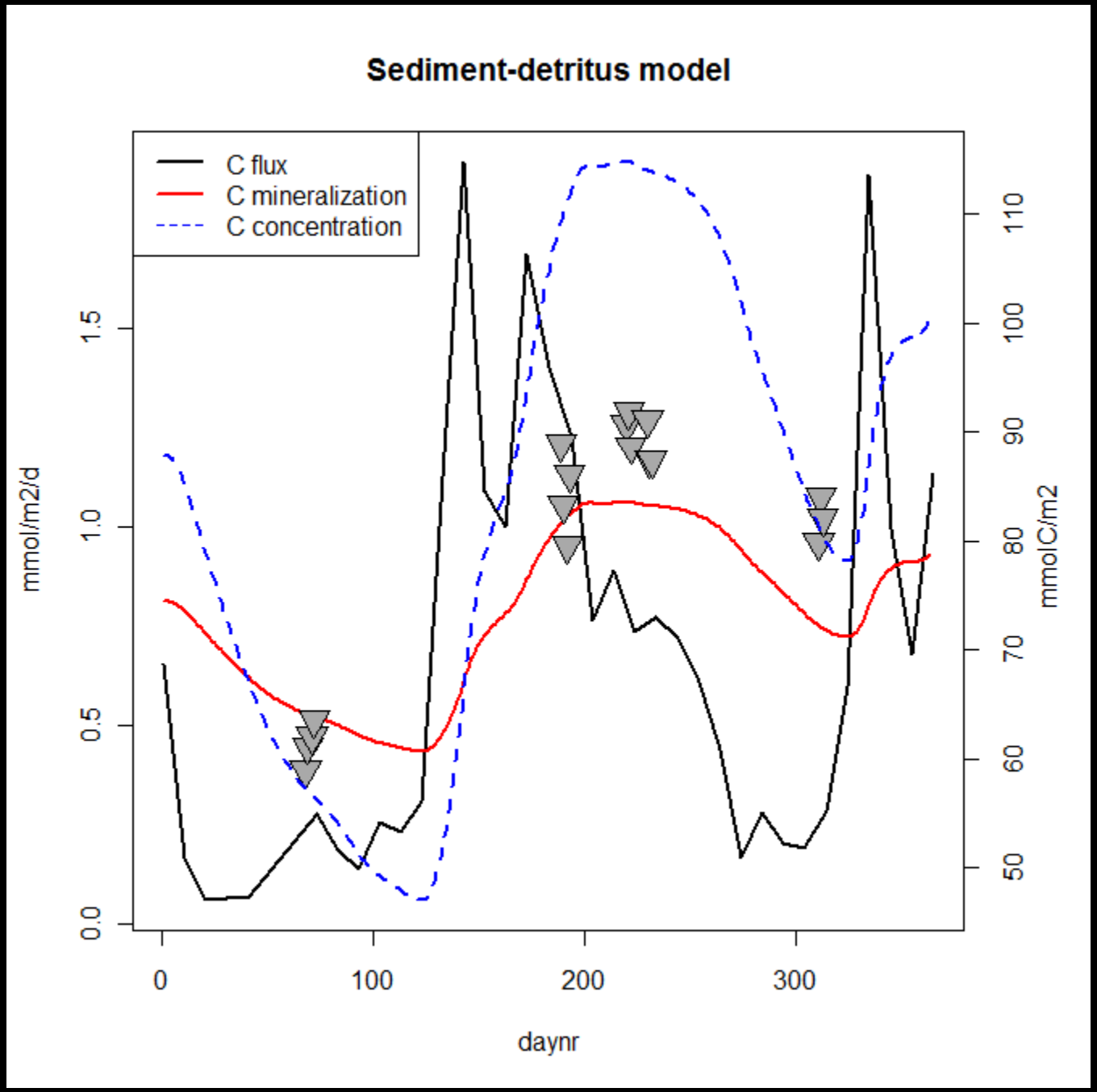

N
dN/dt
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
time
dN/dt

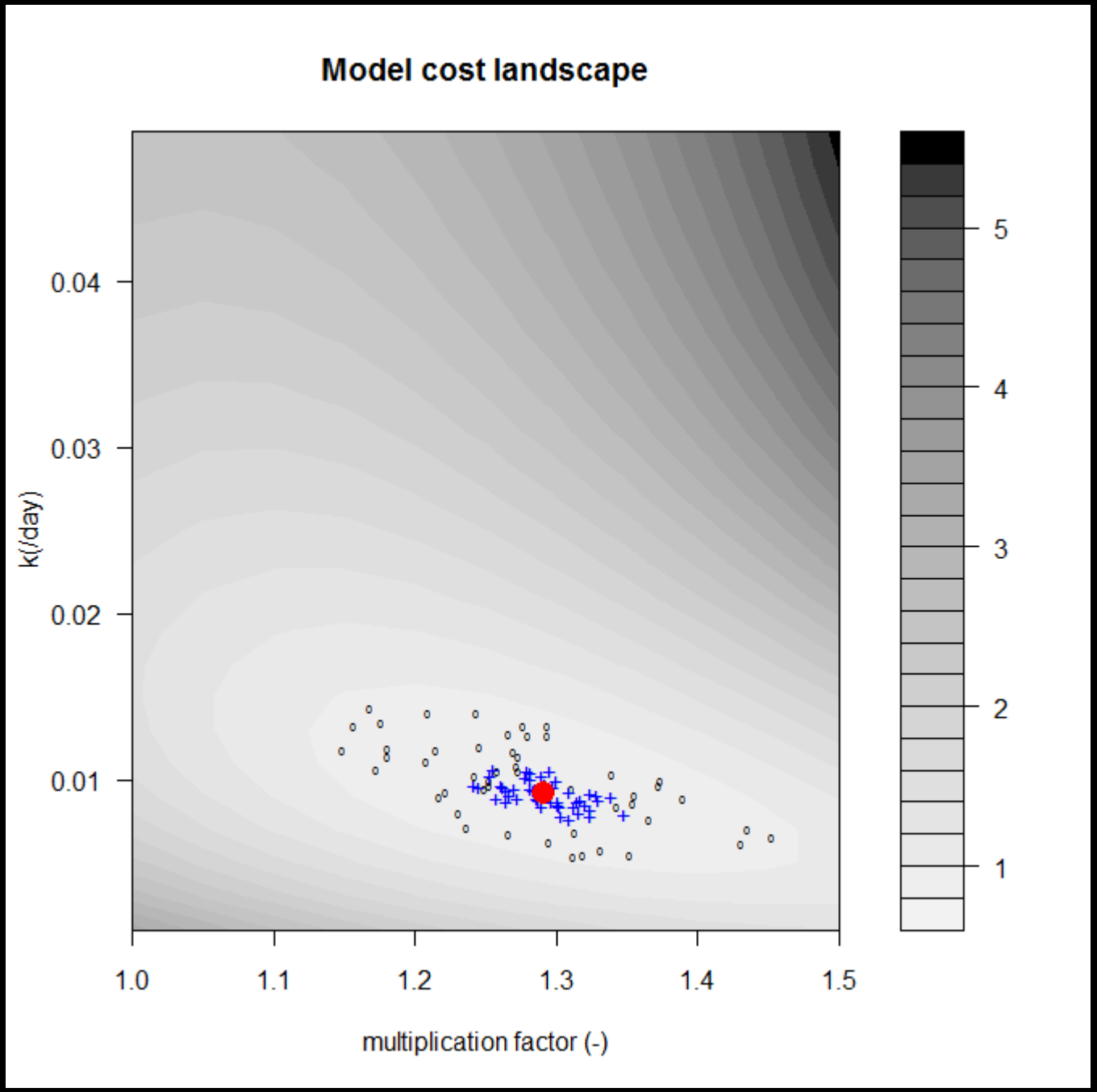
```
minmod <- function(t, Carbon, parameters)
 { with(as.list(c(Carbon, parameters)),
   {
    minrate <- k*Carbon
    Depo <- approx(Flux[, 1], Flux[, 2], xout=t)$y
    dCarbon <- mult*Depo - minrate
   list(dCarbon, minrate)
  })
 }
```

$$\frac{dC}{dt} = mult \cdot flux - k \cdot C$$

```
   costt <- sum((minrate- oxcon$cons)^2)
```

時刻tでのfluxを与えている


Sediment-detritus model
C flux
C mineralization
C concentration
mmol/m2/d
mmolC/m2
daynr
0.0
0.5
1.0
1.5
50
60
70
80
90
100
110
0
100
200
300




コスト分布

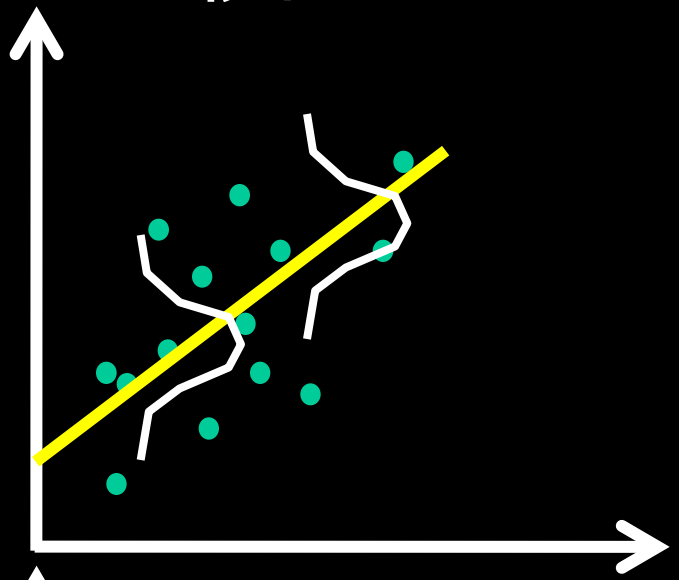
コストがどのくらい小さくなったらやめるか

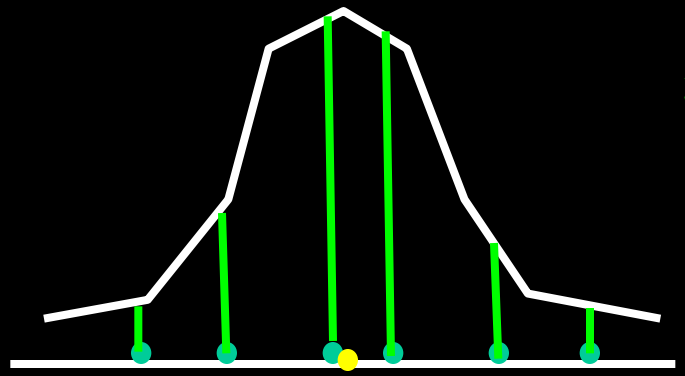
# 第四章のまとめ

- 線形回帰と非線形回帰を紹介した
- ミラー関数(Price 関数)でパラメータ推定することの有効性の示唆
- 非線形回帰の初期値依存の様子を示した
- コストをあらかじめ計算して、良さそうなパラメータからスタートする重要性の示唆

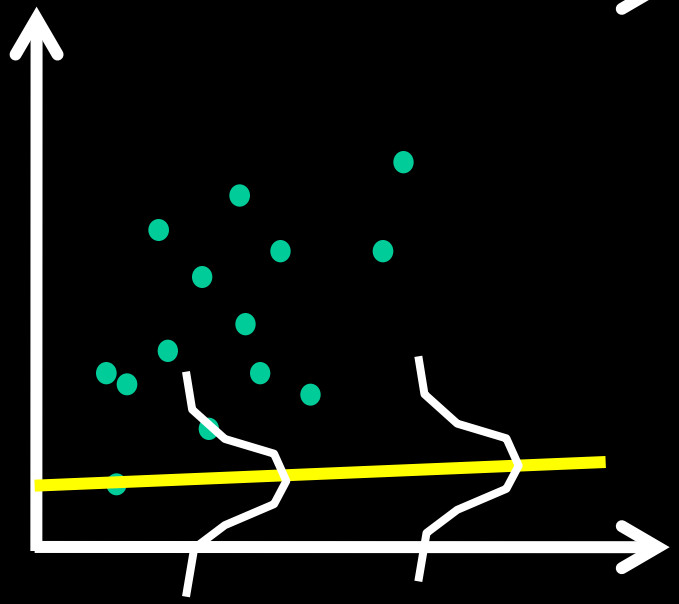
# おまけ

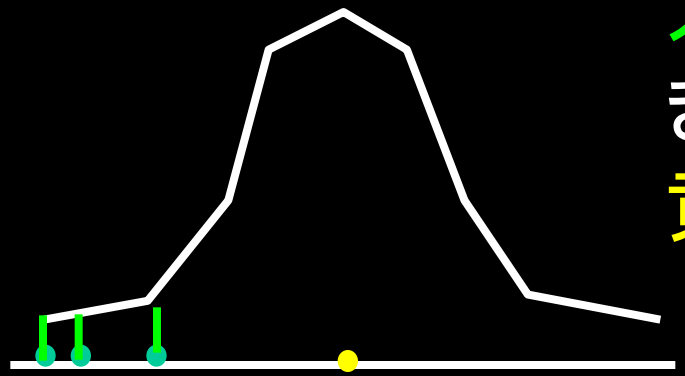
## 統計モデルでは尤度を使ってパラメータ推定





緑の棒線の
長さの合計(積)
＝
尤度





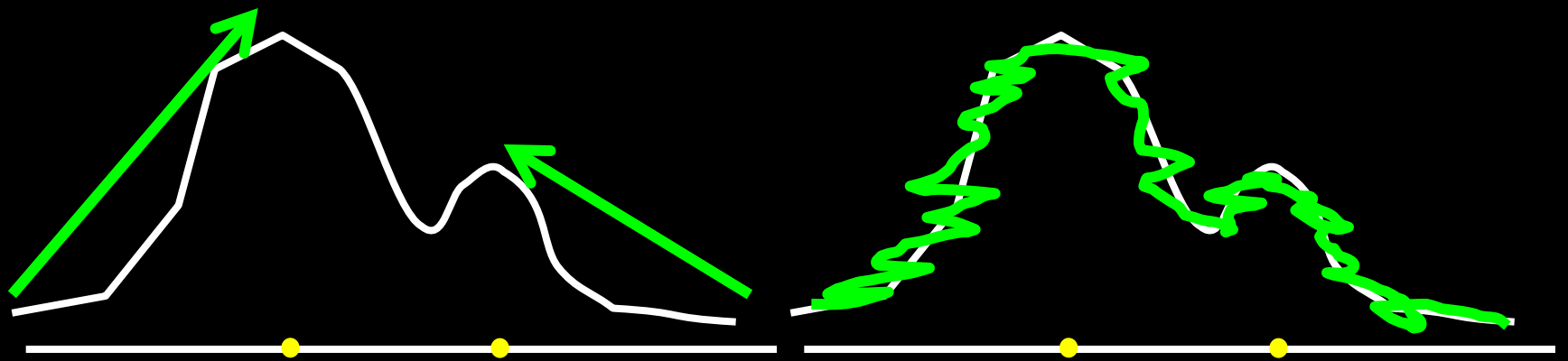
尤度が最も大
きくなるように
黄色の線引く
＝
最尤法

# 最尤法の弱点

たまに、偽ピークにトラップされてしまう。

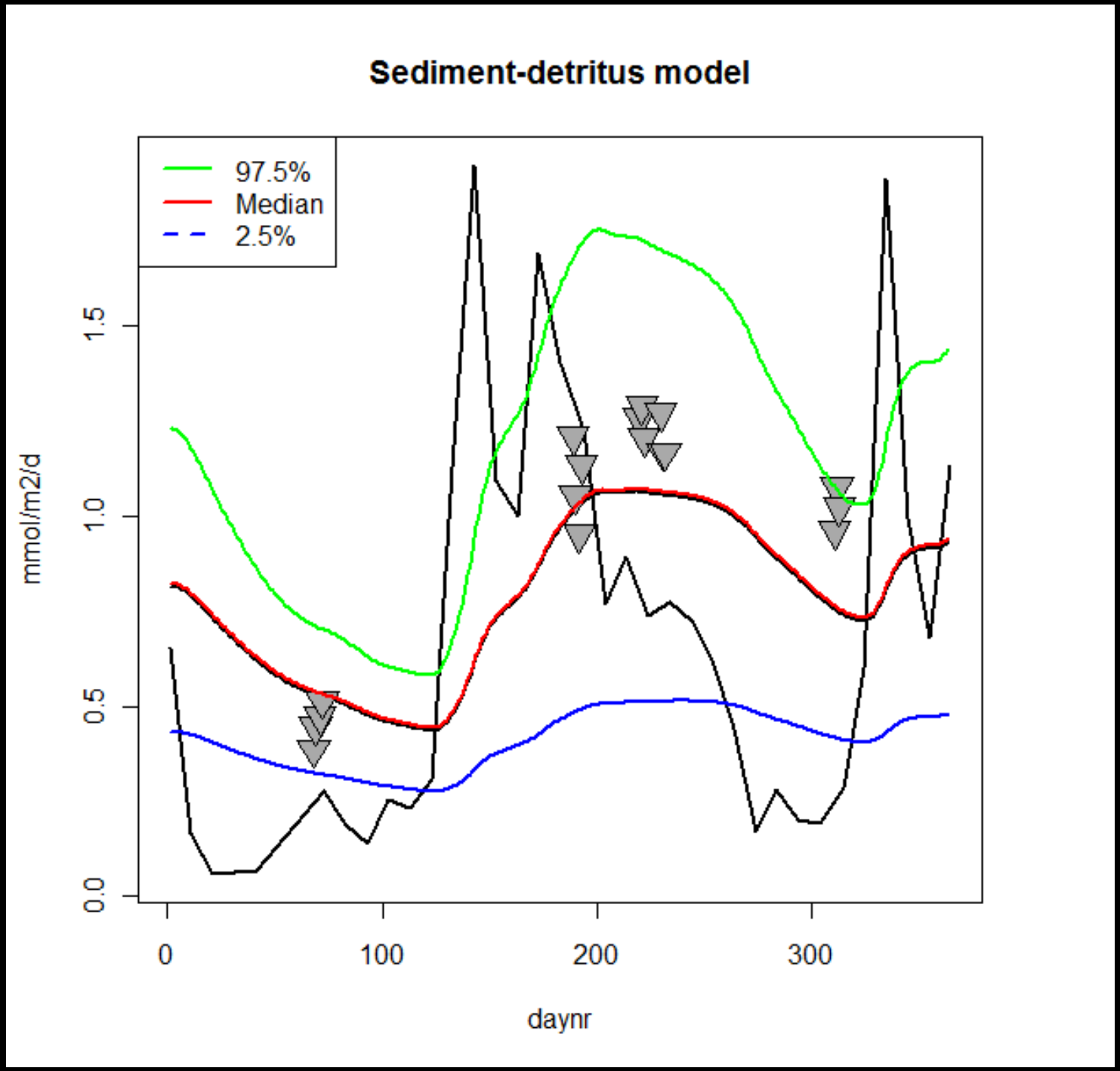
なんで？

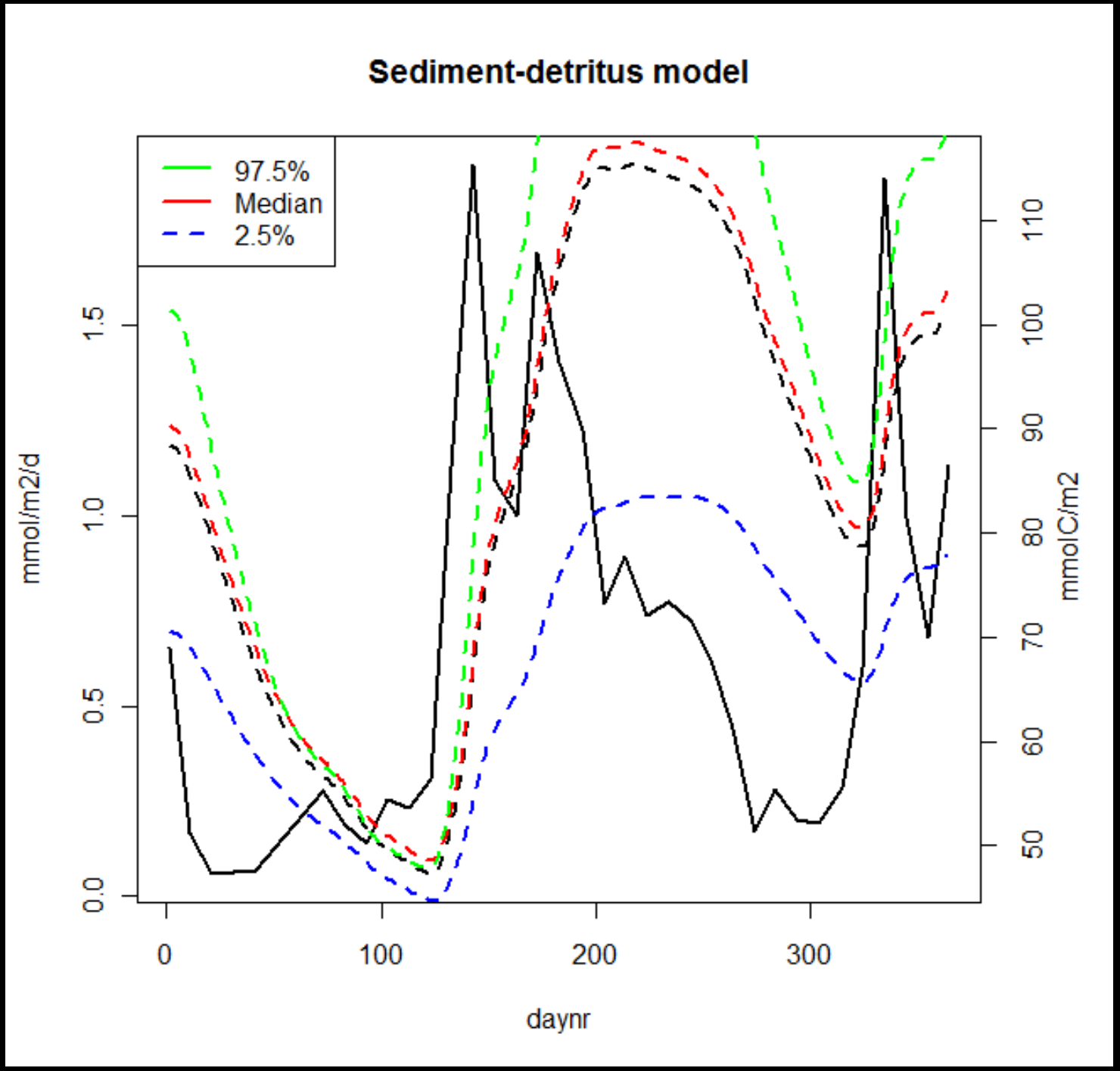
最適解を求めようとするため

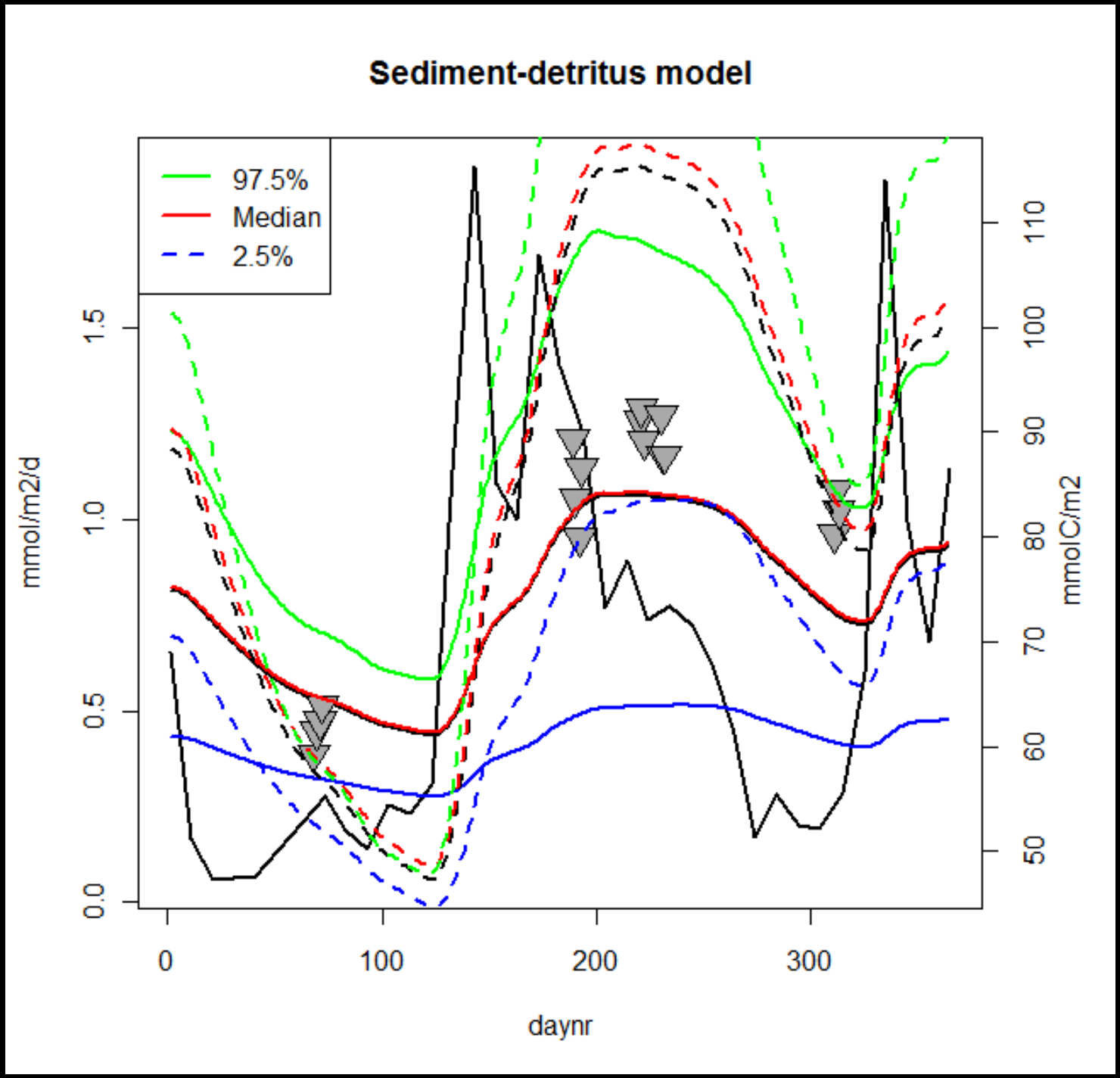


これを回避するためには？

最適解を求めようとせず
パラメータの分布(事後分布)を求めれば良い


Sediment-detritus model
97.5%
Median
2.5%
mmol/m2/d
0.0
0.5
1.0
1.5
0
100
200
300
daynr


Sediment-detritus model
97.5%
Median
2.5%
mmol/m2/d
mmolC/m2
daynr
0.0
0.5
1.0
1.5
50
60
70
80
90
100
110
0
100
200
300


Sediment-detritus model
97.5%
Median
2.5%
mmol/m2/d
mmolC/m2
daynr
0.0
0.5
1.0
1.5
50
60
70
80
90
100
110
0
100
200
300

# おまけのまとめ

最適解を追い求める以外のアルゴリズムでも、
同じ結論になる

誤差を考慮することで、
不確実性を見積もることができる

個人的には、パラメータには誤差があるもの
として扱ってほしいこともある

現実的に無理な場合が多々ありますし