データ解析
第十回「ノンパラメトリック推定法（１）」

鈴木　大慈
理学部情報科学科
西八号館 W707 号室
s-taiji@is.titech.ac.jp

休講情報

6/24 は休講

今日の講義内容

- カーネル密度推定
- スプライン回帰

構成

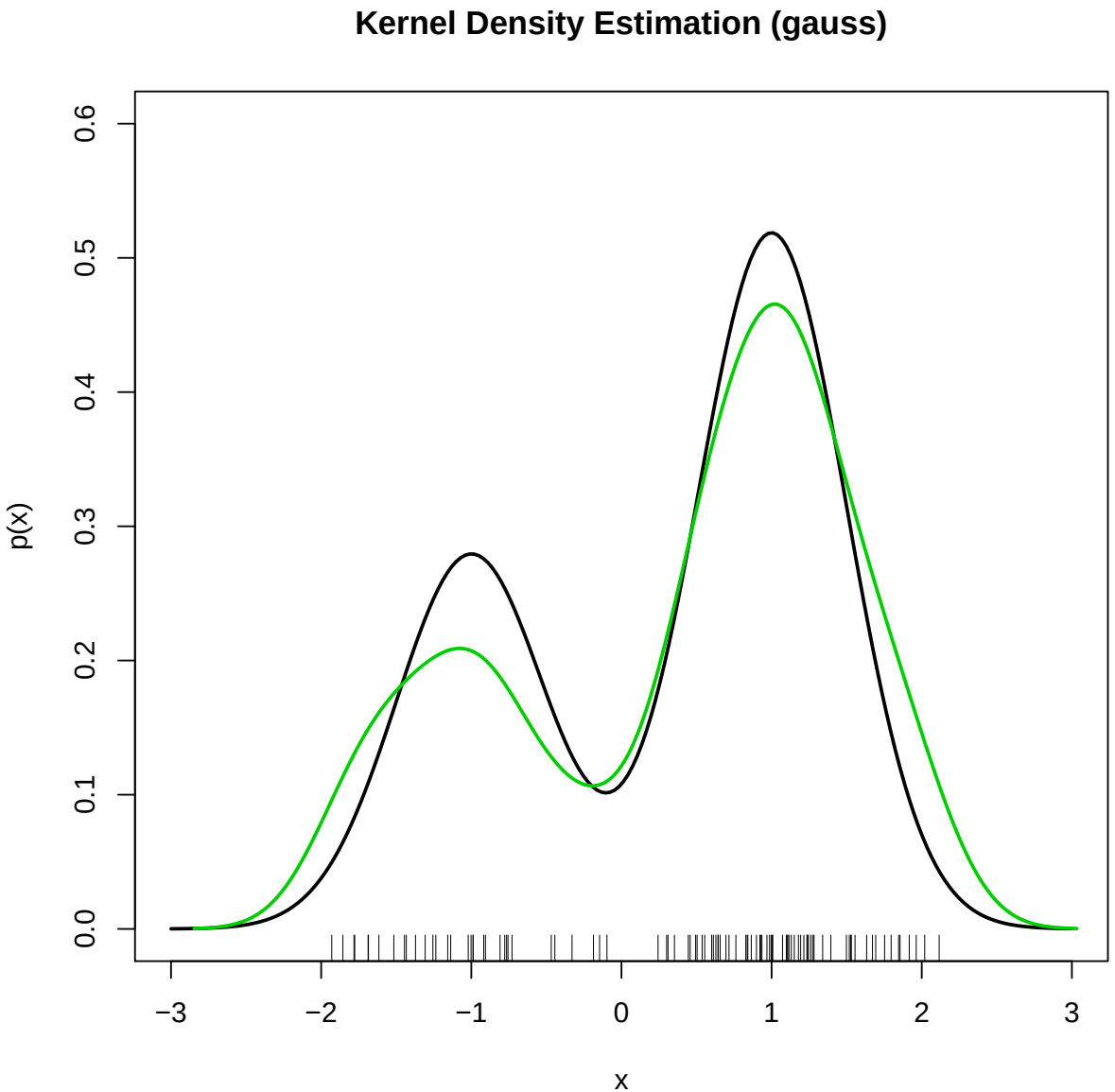
カーネル密度推定の目的

ある分布の密度関数を推定したい.

いままではパラメトリックモデルの上での推定手法を紹介してきた.
(最尤推定，ベイズ推定)
もし分布を適切なパラメトリックモデルで記述できなかったら？

→　**ノンパラメトリック推定**
カーネル密度推定はその代表的な方法.

カーネル密度推定で何が得られる？



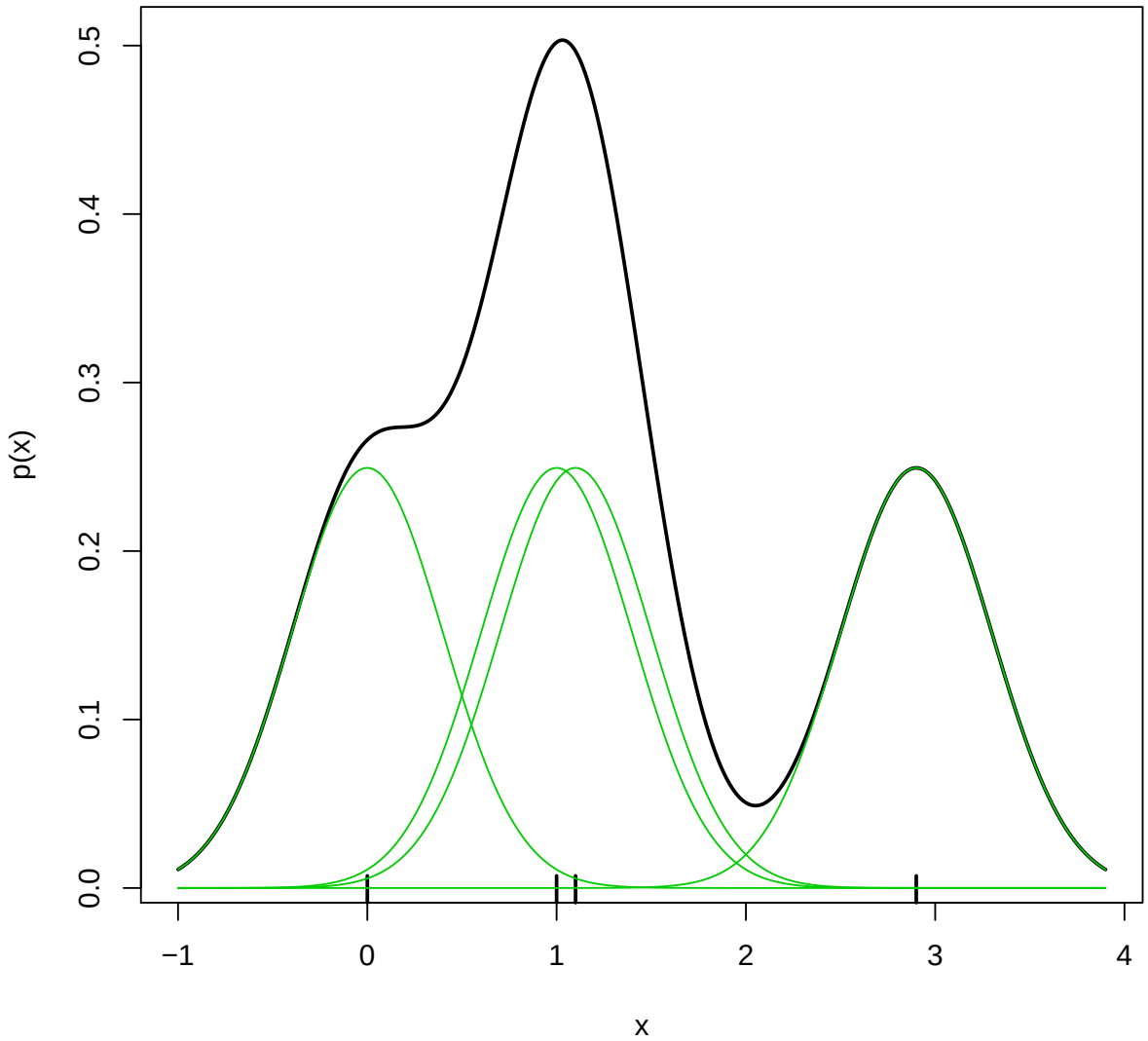
カーネル密度推定量

$\{X_i\}_{i=1}^n$: データ (一次元とする)
カーネル密度推定量: ある**カーネル関数** K を用いて,

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right).$$

1. $h > 0$ のことをバンド幅と呼ぶ. 適切に選択する必要がある.
2. K は次の性質を満たすものとする:

$$\int K(x)\mathrm{d}x = 1, \quad \int xK(x)\mathrm{d}x = 0, \quad \int x^2 K(x) > 0.$$

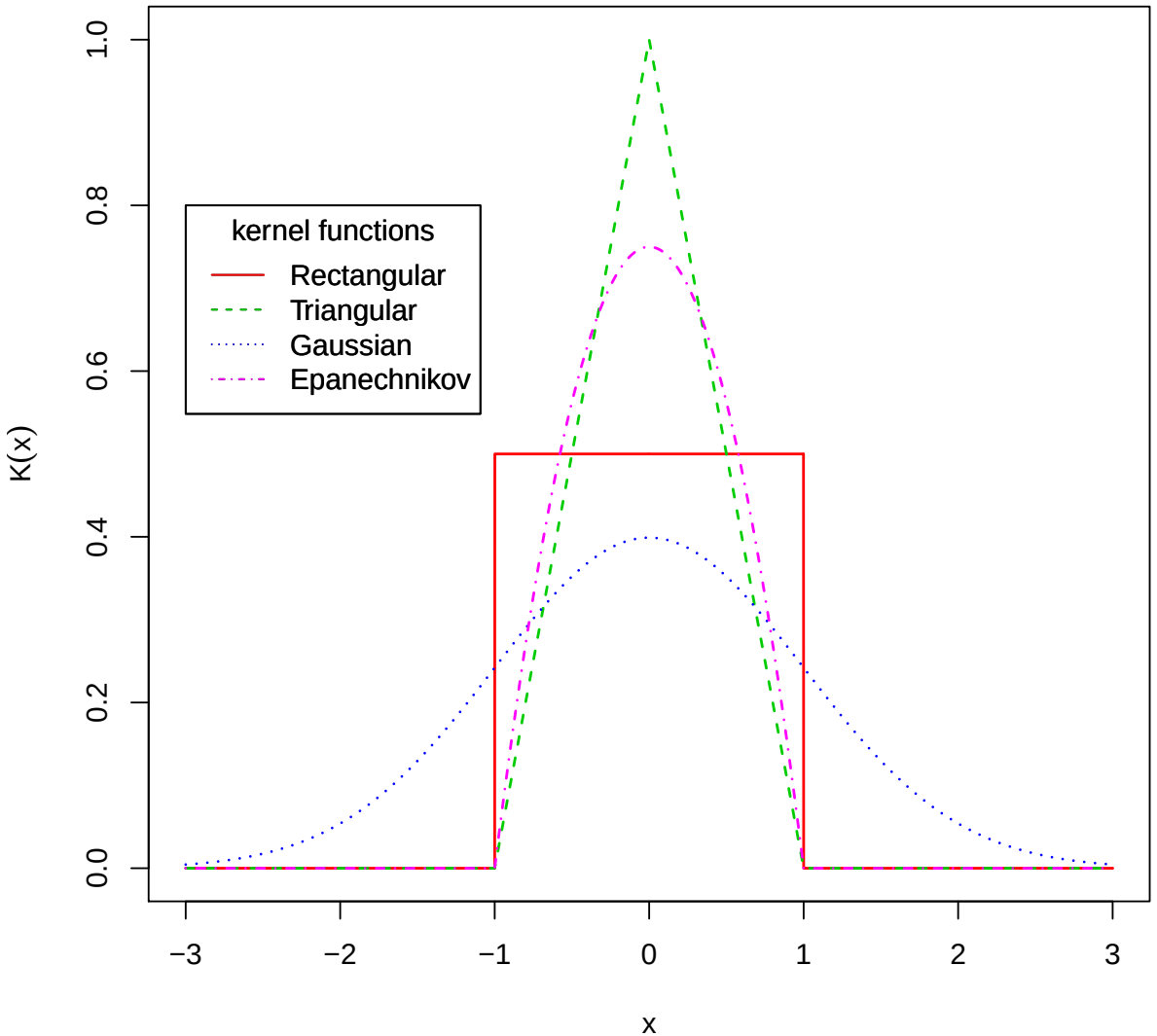

p(x)
0.0
0.1
0.2
0.3
0.4
0.5
−1
0
1
2
3
4
x

カーネルの種類

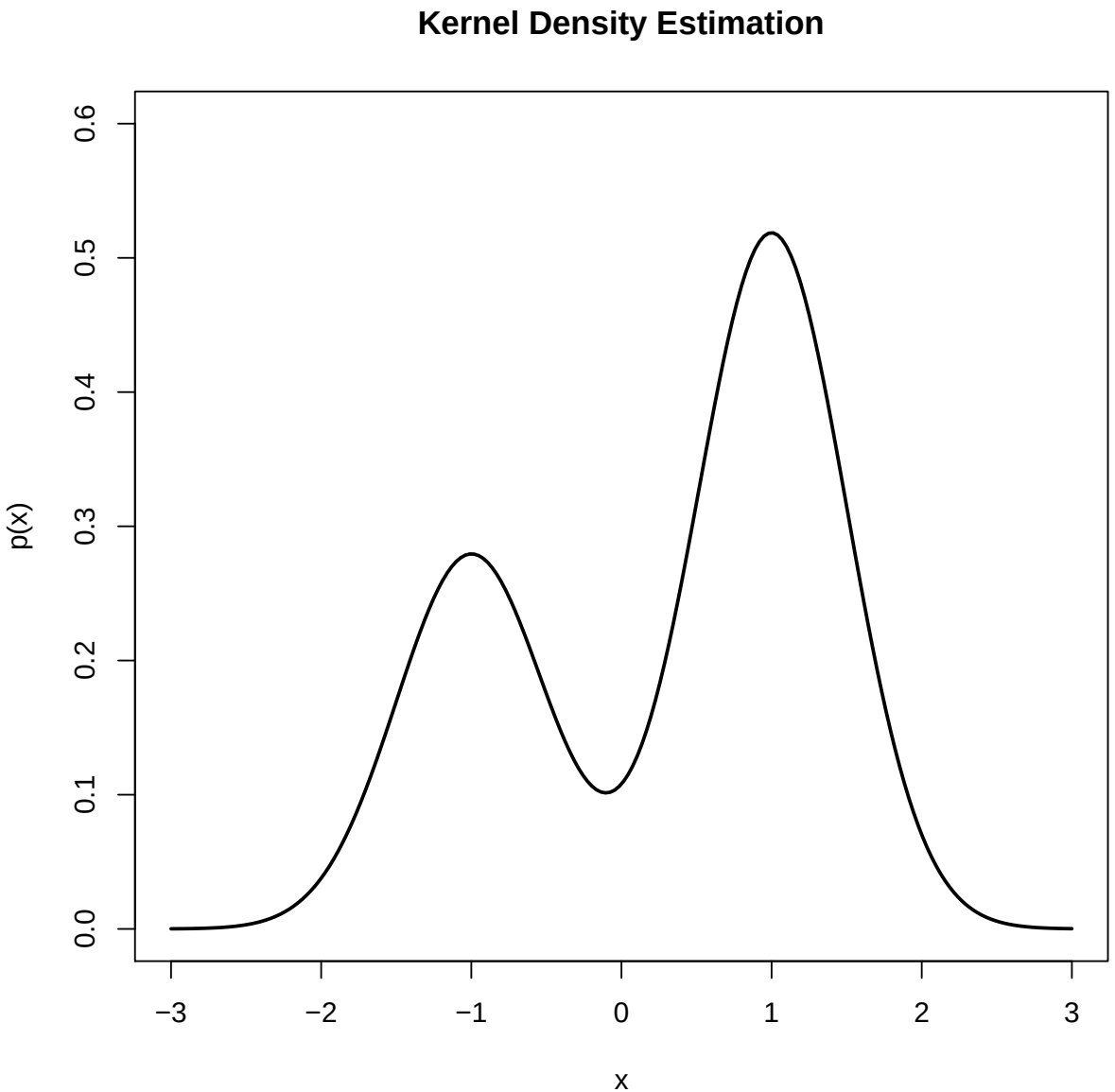
次のようなカーネル関数がよく用いられる.

1. Gaussian:
$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$
2. Rectangular:
$$K(x) = \begin{cases} \frac{1}{2} & (|x| \leq 1), \\ 0 & (\text{otherwise}). \end{cases}$$
3. Triangular:
$$K(x) = \begin{cases} |x| & (|x| \leq 1), \\ 0 & (\text{otherwise}). \end{cases}$$
4. Epanechnikov:
$$K(x) = \begin{cases} \frac{3}{4}(1 - x^2) & (|x| \leq 1), \\ 0 & (\text{otherwise}). \end{cases}$$

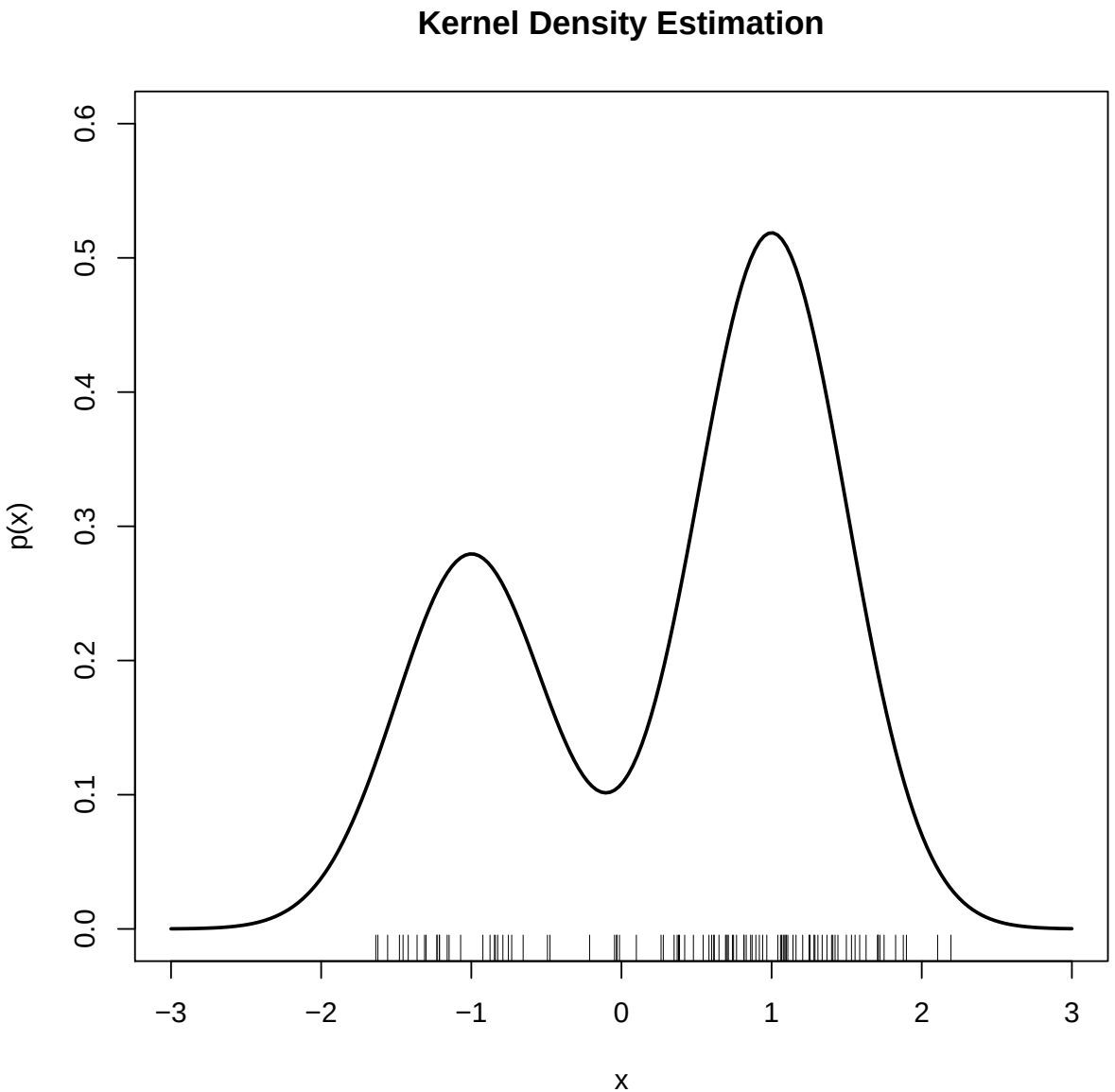
カーネルの種類


kernel functions
Rectangular
Triangular
Gaussian
Epanechnikov
K(x)
x
0.0
0.2
0.4
0.6
0.8
1.0
−3
−2
−1
0
1
2
3

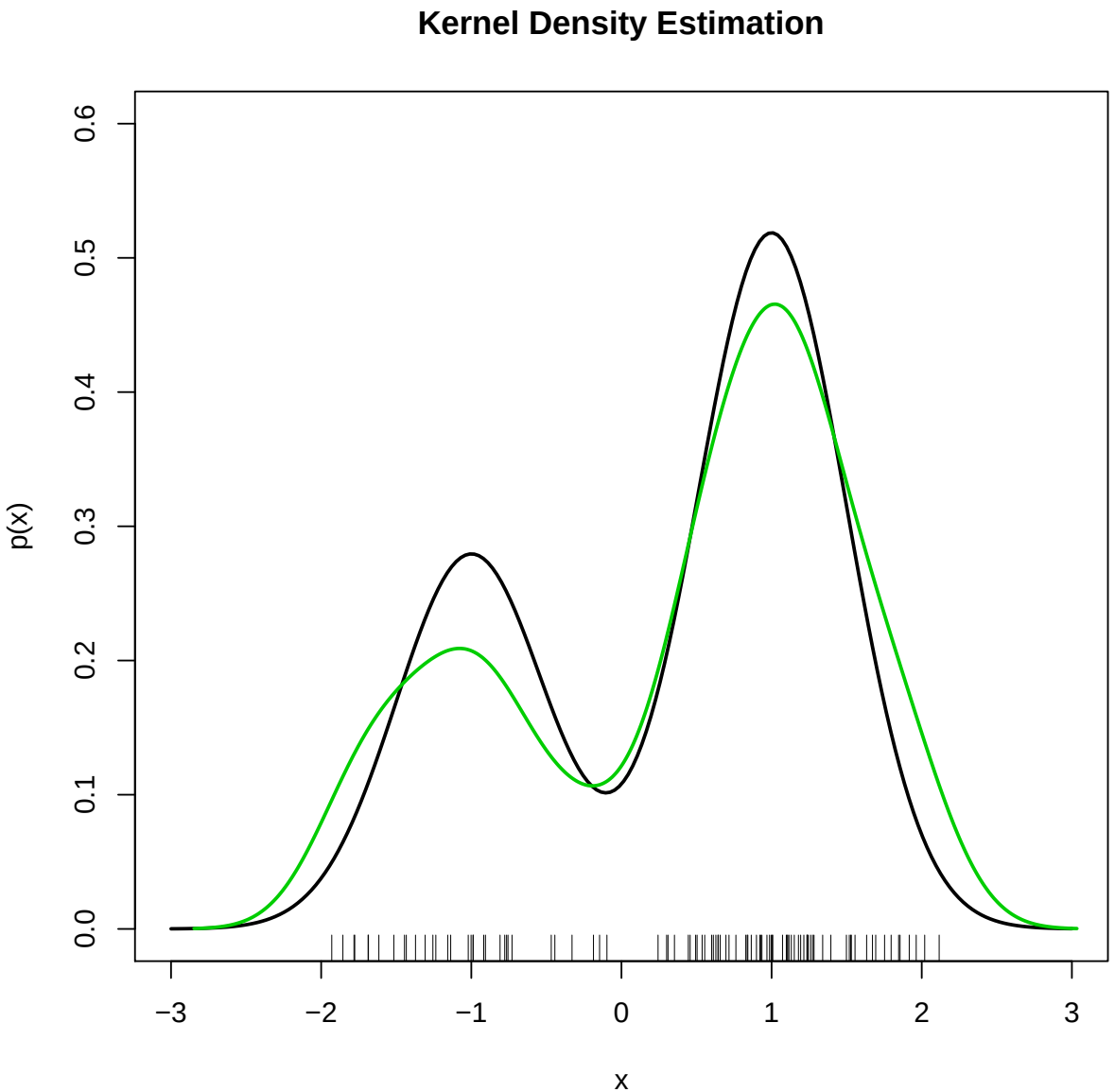
カーネル密度推定量の様子



カーネル密度推定量の様子

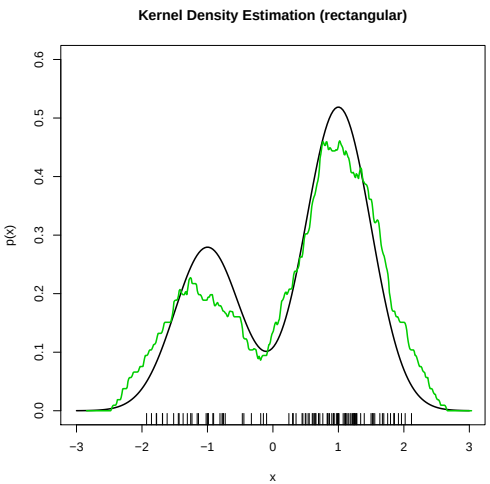


カーネル密度推定量の様子

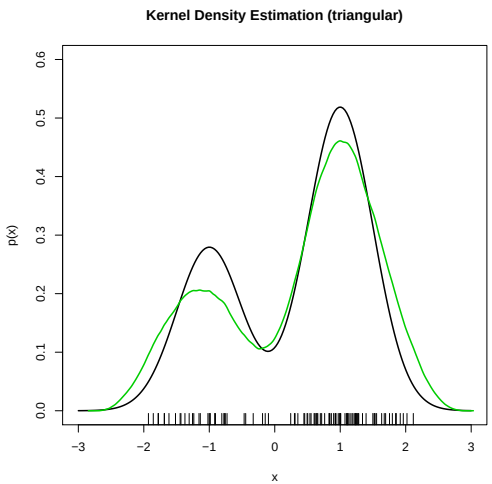


カーネル関数の種類と推定結果

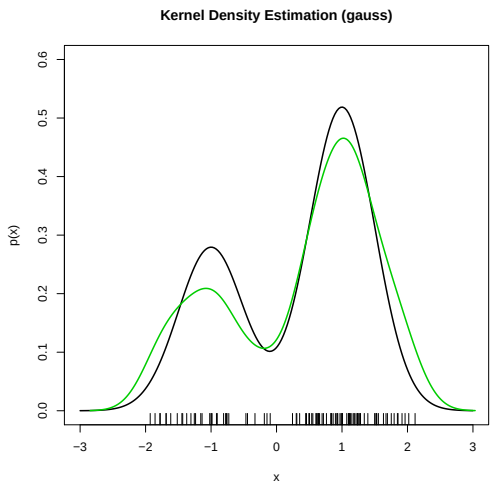
$n = 100$



rectangular



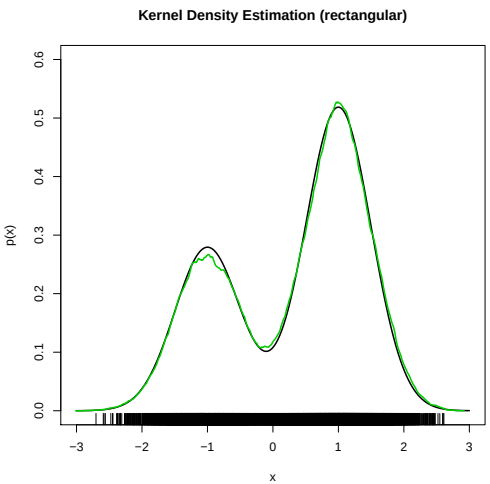
triangular



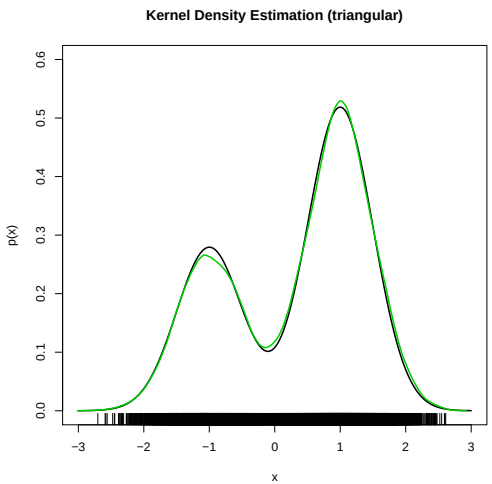
gauss

カーネル関数の種類と推定結果

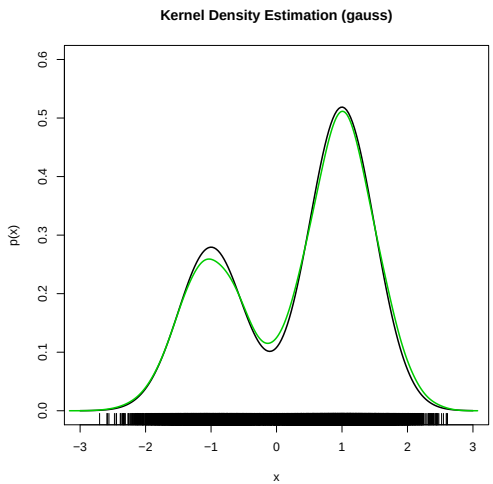
$n = 10000$



rectangular



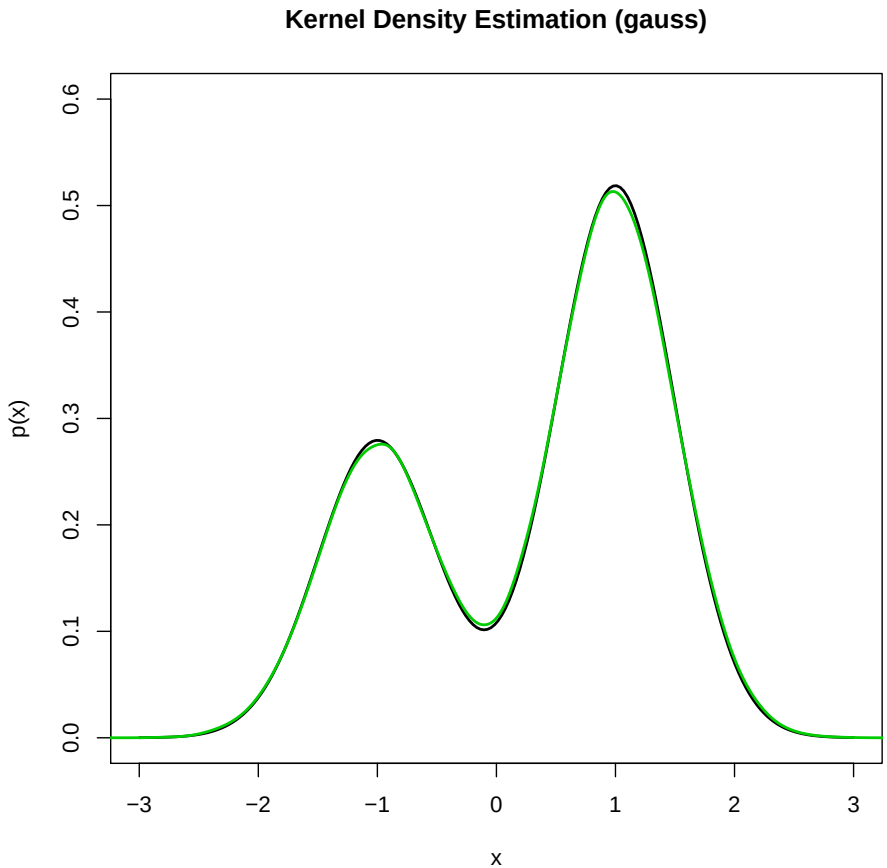
triangular



gauss

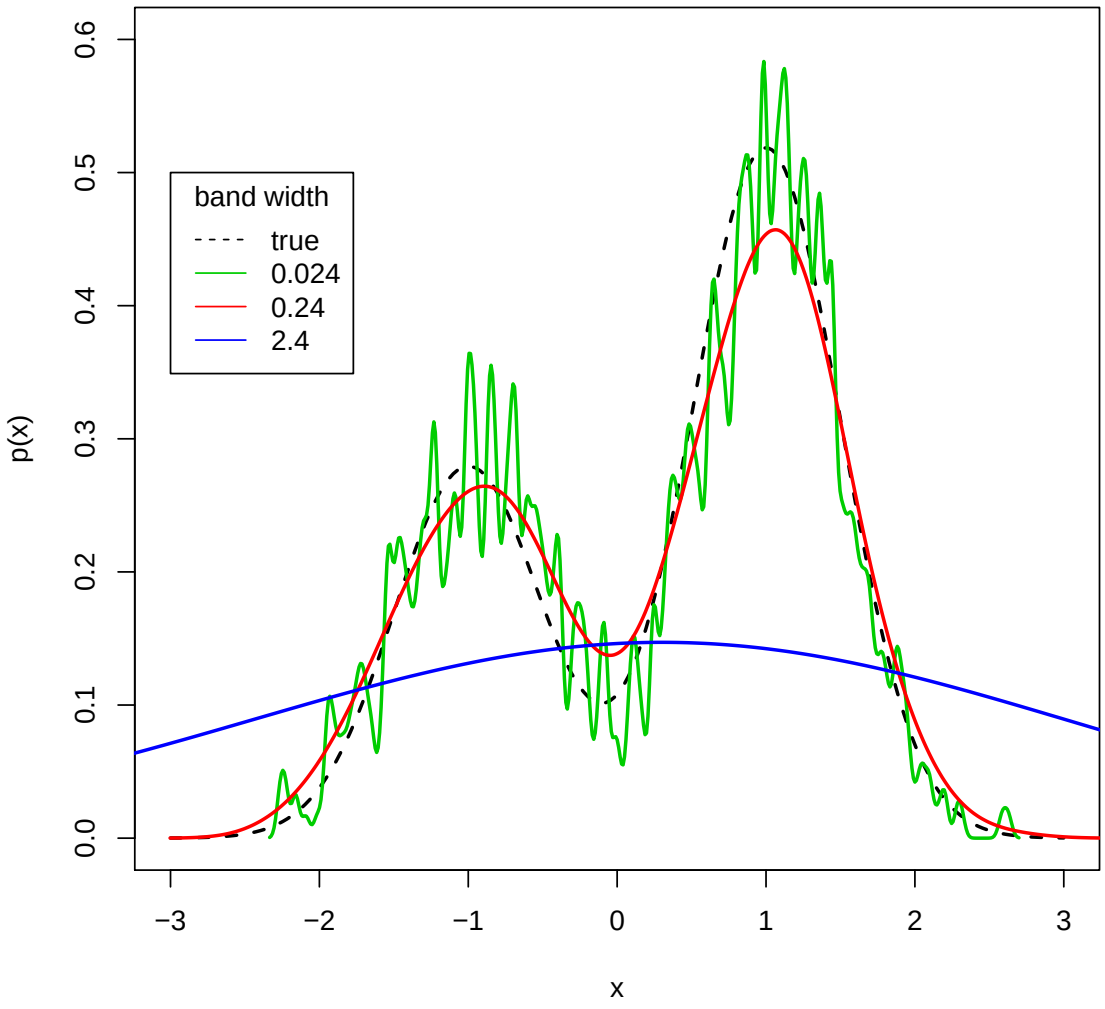
カーネル関数の種類と推定結果

$n = 100000$



gauss

カーネル密度推定量とバンド幅



バンド幅は適切に選ぶ必要がある.

最小二乗クロスバリデーション

最小二乗クロスバリデーション: LSCV, Least Squares Cross Validation.

真の密度との二乗距離 ($\hat{p}_h$ をバンド幅 h のカーネル密度推定量とする)

$$\begin{aligned}&\int (\hat{p}_h(x) - p(x))^2 \mathrm{d}x \\ &= \underbrace{\int \hat{p}_h(x)^2 \mathrm{d}x - 2\int \hat{p}_h(x)p(x)\mathrm{d}x}_{=:J(h)} + \int p^2(x)\mathrm{d}x.\end{aligned}$$

$J(h)$ を最小化すれば良い．しかし $p(x)$ による積分がわからない→サンプルで代用.
ただし，手元にあるサンプルと $\hat{p}_h$ は相関がある のでクロスバリデーションする.

$\hat{p}_{h,(-i)}$: i 番目のサンプル X_i を抜いて推定した密度関数.

$$\widehat{J}(h) = \int \hat{p}_h(x)^2 \mathrm{d}x - \frac{2}{n}\sum_{i=1}^{n} \hat{p}_{h,(-i)}(X_i).$$

$\widehat{J}(h)$ を最小にする h を採用すればよい.

ヒューリスティクス

Silverman の方法

1. サンプル標準偏差: $\hat{s} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2}$.
2. サンプル分位点 $q(0.25)$: サンプルの 0.25 分位点.

$$\hat{\sigma} = \min\left\{\hat{s}, \frac{\hat{q}(0.75) - \hat{q}(0.25)}{1.34}\right\}$$

として，次のようにする:

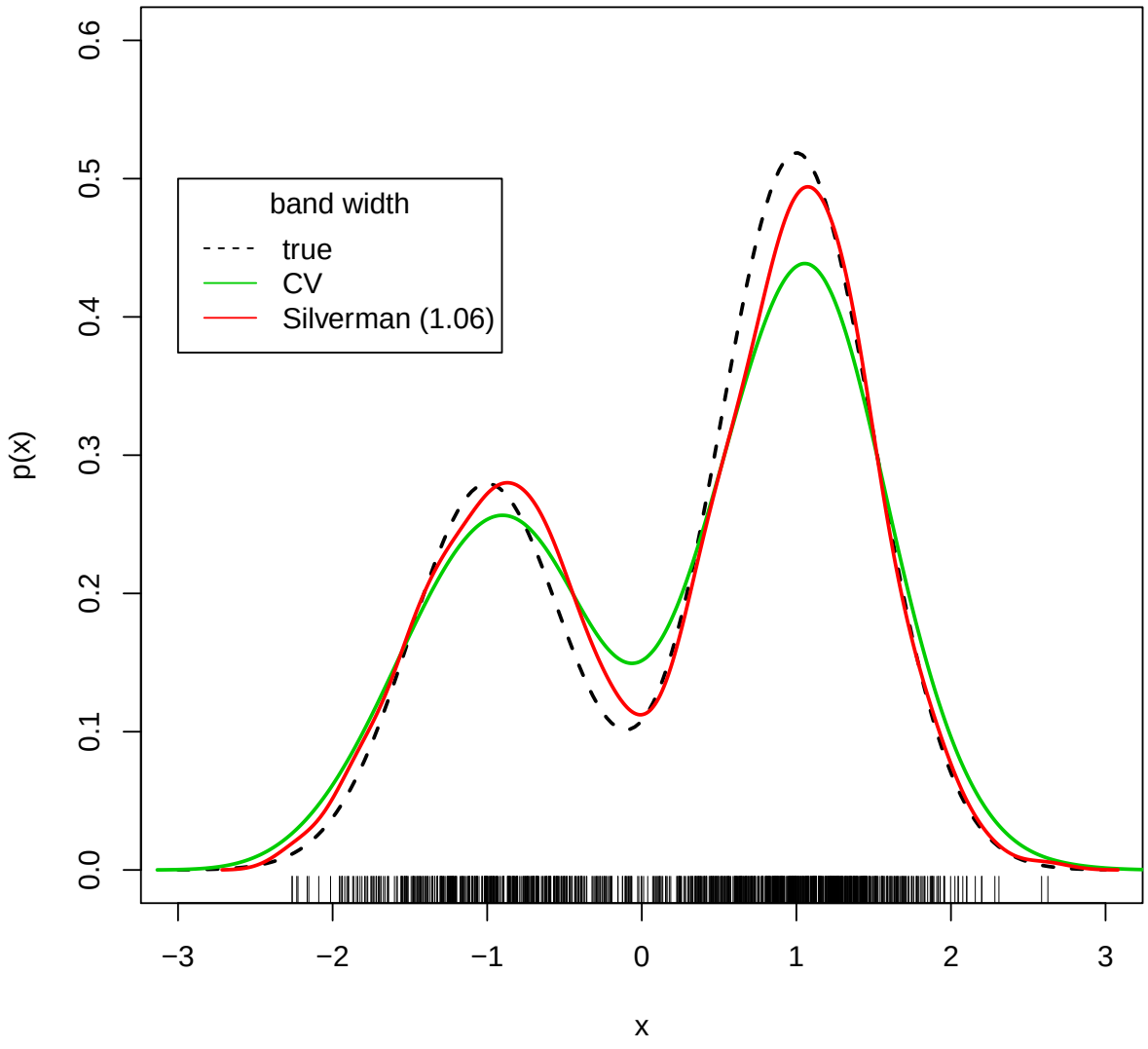
$$\hat{h} = \frac{1.06\hat{\sigma}}{n^{1/5}}.$$

1.06 は人によって別の定数に置き換えたりする (1.06 を Scott のルール，0.9 を Silverman のルールと言う).

特に理論的根拠はないが，二乗誤差を漸近展開すると，

$$\hat{h} = \left[\frac{C_K}{n(\int f''(x)\mathrm{d}x)}\right]^{1/5},$$

(ただし $C_K = \frac{\int K(x)^2\mathrm{d}x}{(\int x^2 K(x)\mathrm{d}x)^2}$) が漸近的に最小二乗誤差を与えることが示せて，上のヒューリスティクスは未知の値 $\int f''(x)\mathrm{d}x$ に当たりを付ける経験則とみなせる.



ヒューリスティクスも良い推定結果を出している.

多変量カーネル密度推定

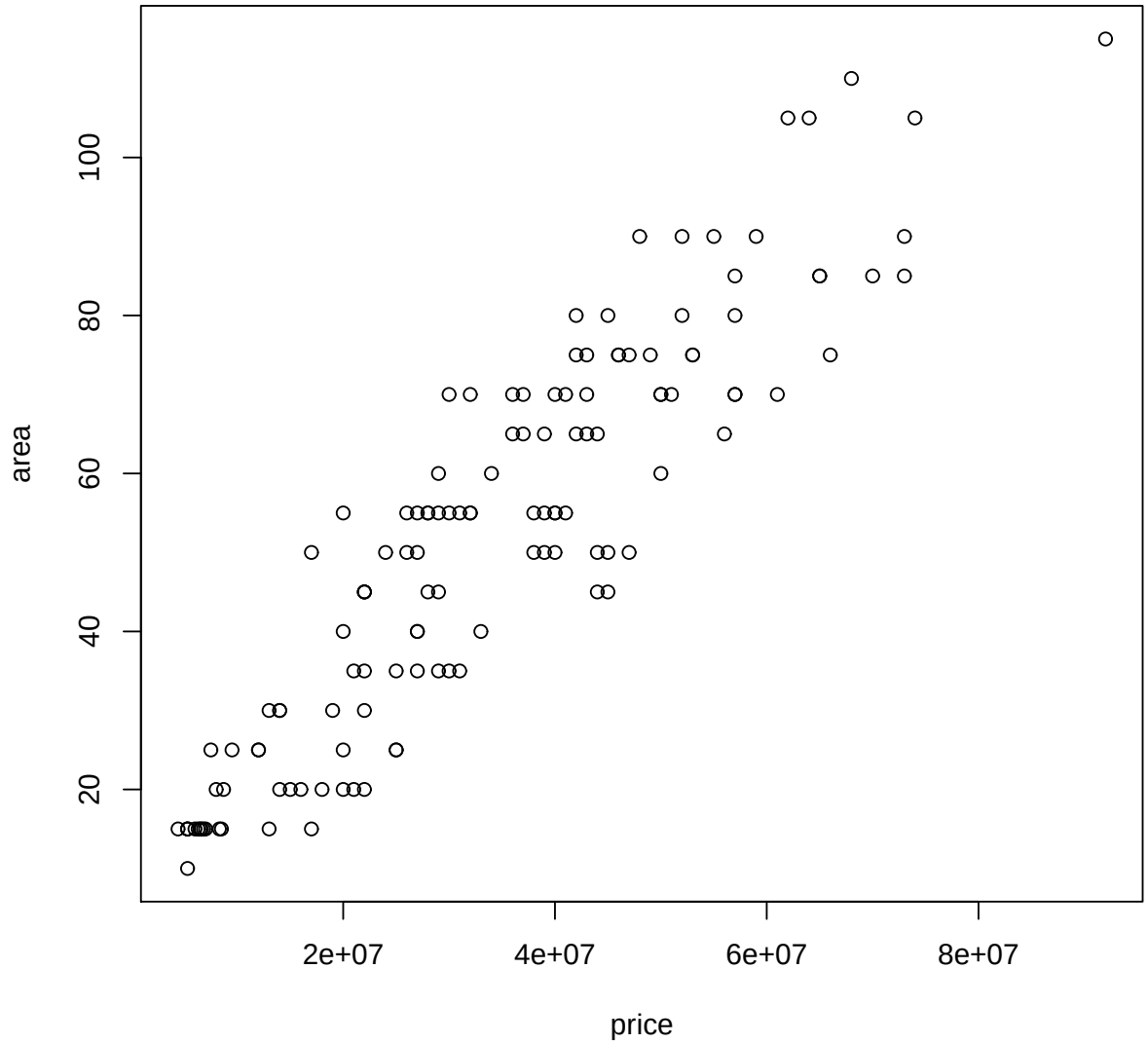
カーネル密度推定は多変量へ拡張できる．$\frac{1}{h}K(\frac{x-X_i}{h})$ の代わりに，

$$\prod_{j=1}^{d} \frac{1}{h_j} K\left(\frac{x - X_i^{(j)}}{h_j}\right)$$

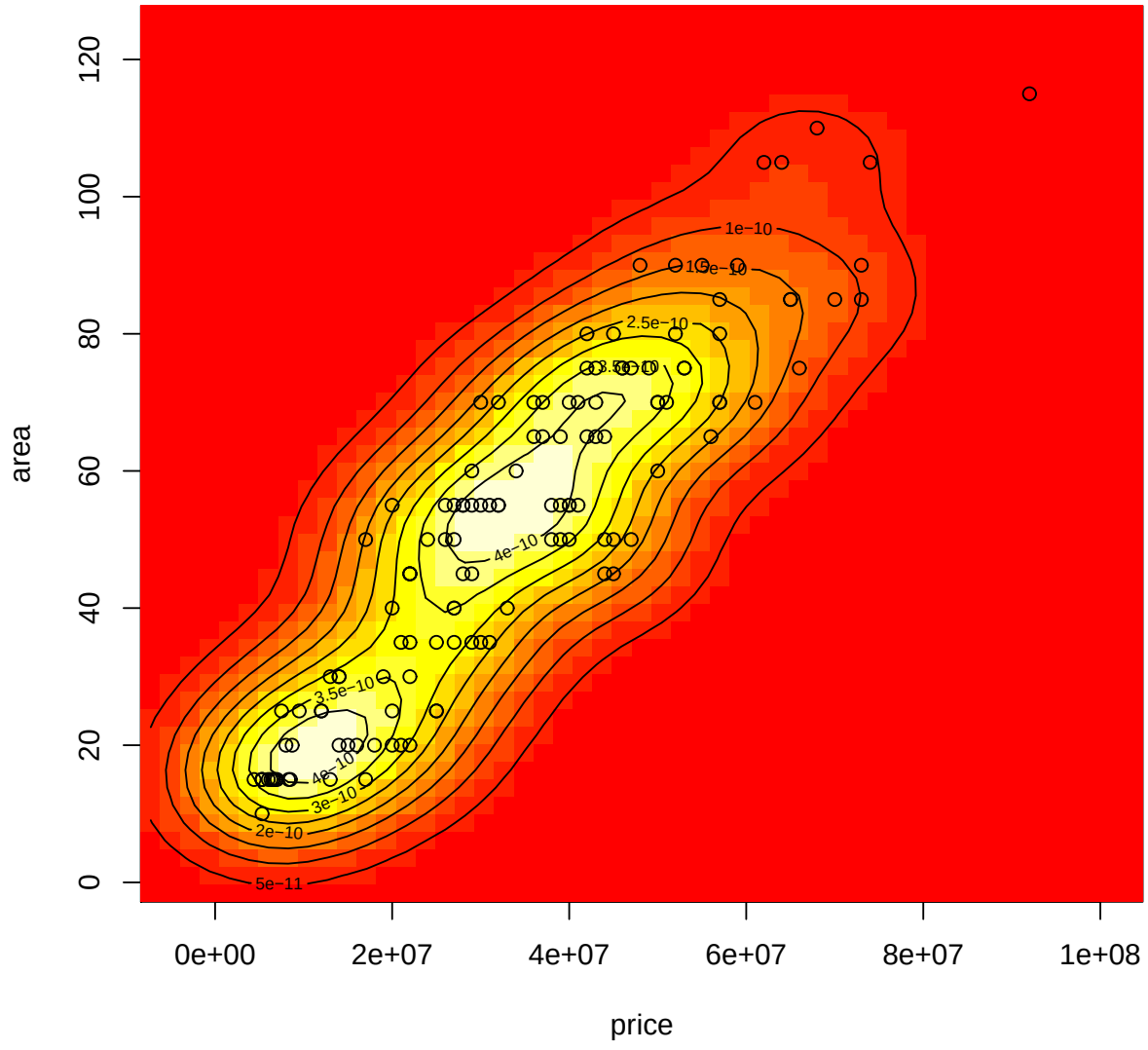
のようにする．なお，$X_i^{(j)}$ はサンプル X_i の j 番目の座標．

```
library(MASS)
d=kde2d(x,y,c(bandwidth.nrd(x),bandwidth.nrd(y)),n=80)
image(d,xlab="latitude",ylab="longitude")
```

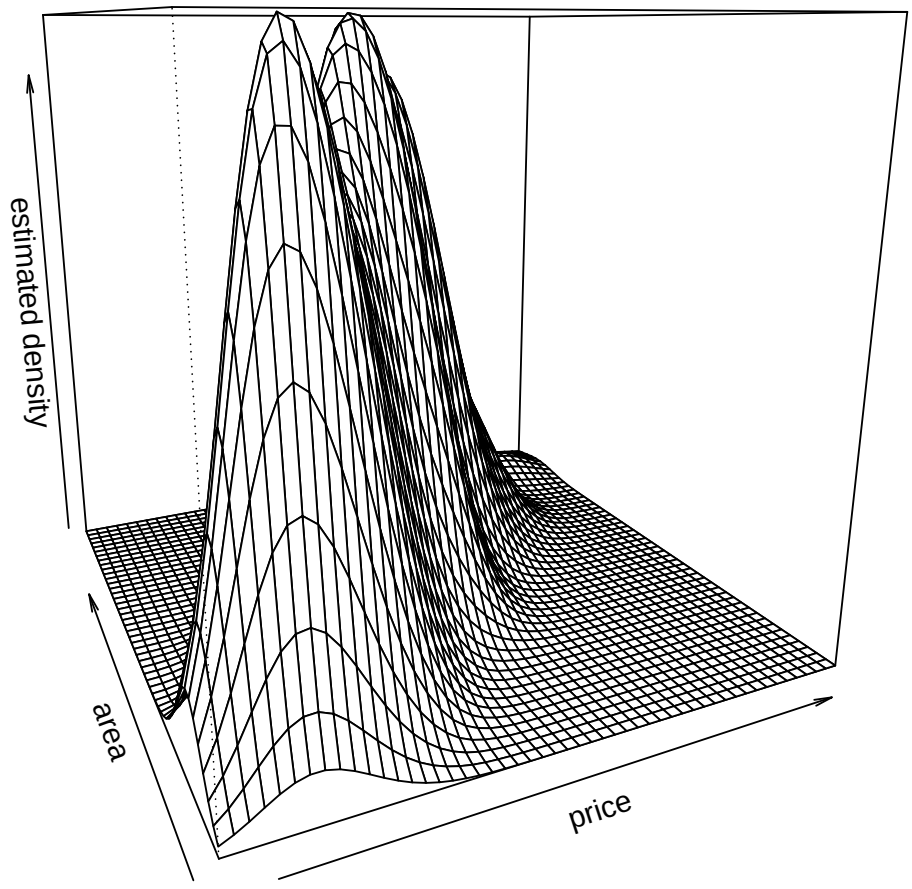
世田谷区中古マンション価格データ



世田谷区中古マンション価格データ



世田谷区中古マンション価格データ



カーネル密度推定を行えるR関数

使い方はスクリプトを参照

1次元のカーネル密度推定

- density(x,bw,kernel):
  - バンド幅: bw = "nrd0"がデフォルト (シルバーマンの方法で0.9を採用), bw="'nrd' で1.06. bw="ucv" で (バイアス修正した) クロスバリデーション, bw="bcv" でクロスバリデーション.
  - カーネル関数: kernel="gaussian" がデフォルト. 他にも"epanechnikov", "rectangular", "triangular", "biweight","cosine", "optcosine"が指定可能.
- bkde: 'KernSmooth' パッケージに入っている.

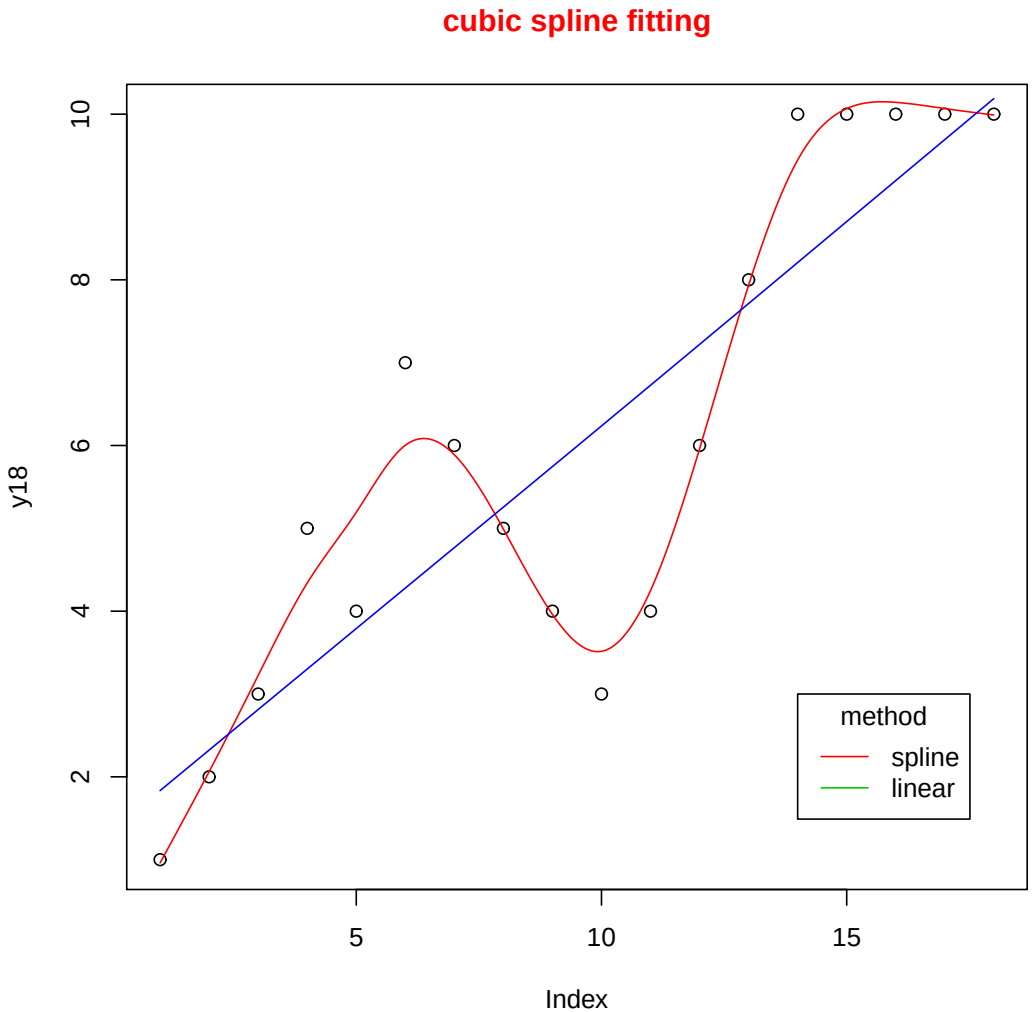
二次元以上

- kde2d: 'MASS' パッケージに入っている. 2次元用.
- bkde2d: 'KernSmooth' パッケージに入っている. 2次元用.
- kde: 'ks' パッケージに入っている. 3次元以上の密度推定も行える.

MASSパケージのbandwidth.nwdはbw.nwdの4倍の値を返す.

構成

ノンパラメトリック回帰



スプライン回帰

ノンパラメトリック回帰の基本的構造:

$$f(x) = \sum_{j=1}^{q} \alpha_j B_j(x).$$

$B_j(x)$ はある非線形な基底関数．ここでは，「**B-スプライン基底**」を考える．

B-スプライン基底

B-スプライン基底関数は局所的な多項式関数である.
例えば，３次 B-スプラインの場合，B_k は次のような局所３次多項式である:

$$B_k(x) = \begin{cases} a_k + b_k x + C_k x^2 + d_k x^3 & (t_k \le x \le t_{k+4}), \\ 0 & (\text{otherwise}). \end{cases}$$

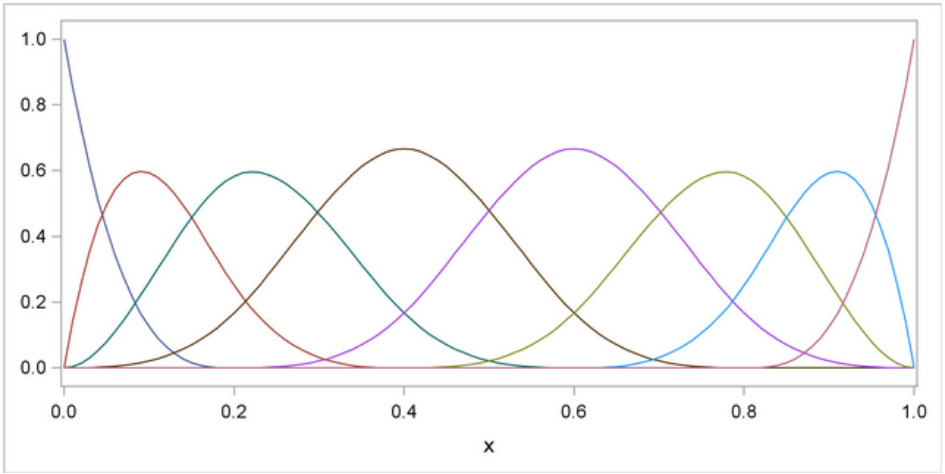
ここで，t_k は節点 (節点) と言う.

- 節点 (節点): $t_1 \le \cdots \le t_{q+1}$

３次スプラインの場合，t_k の候補としてサンプル点 X_i を用いて次のようにしたりする:

$$t_1 \le t_2 \le t_3 \le t_4 = X_1 < \cdots < X_n = t_{n+3} \le t_{n+4} \le t_{n+5} \le t_{n+6}.$$

$((X_1, \ldots, X_n)$ の外側に 6 個余分に節点を適当に設定)



B-スプライン基底の具体的な形

係数 a_k, b_k, c_k, d_k の決め方は本講義の範疇を超えるが，j 次 B-スプライン基底は

$$B_k^{(1)}(x) = \begin{cases} 1 & (t_k \leq x \leq t_{k+1}) \\ 0 & (\text{otherwise}) \end{cases},$$

$$B_k^{(j)}(x) = \frac{x - t_k}{t_{k+j} - t_k} B_k^{(j-1)}(x) + \frac{t_k - x}{t_{k+j+1} - t_{k+1}} B_{k+1}^{(j-1)}(x),$$

なる漸化式で決まる．$j = 3$ の時に 3 次 B-スプライン基底を得る．

要は非線形な関数を局所的に 3 次多項式で近似しましょうということ．

3 次 B-スプラインを 3 次-スプライン，キュービック-スプラインと呼んだりする．

回帰係数の決定と平滑化

以後，3 次スプライン基底考え，各データ点 X_i は節点になっているものとする．
節点が X_i であるような 3 次スプライン基底を $B_i(x)$ $(i = 1, \ldots, n)$ と書き直し，
$B_{n+1}(x) = x$, $B_{n+2}(x) = 1$ とする．
この合計 $n + 2$ 個の基底を用い，次のような関数を構成する:

$$f(x; \alpha) = \sum_{i=1}^{n} \alpha_i B_i(x) + \alpha_{n+1} x + \alpha_{n+2}.$$

$\alpha \in \mathbb{R}^{n+2}$ をデータから推定したい．3 次スプラインでは次のようにして推定する．

$$\min_{\alpha \in \mathbb{R}^{n+2}} \sum_{i=1}^{n} (Y_i - f(X_i; \alpha))^2 + \underbrace{\lambda \int_{X_1}^{X_n} \left(\frac{\mathrm{d}^2 f(x; \alpha)}{\mathrm{d}x^2} \right)^2 \mathrm{d}x}_{\text{正則化項}}.$$

※ 正則化項を加えることで，推定した関数が滑らかになるように調整 →平滑化．
これがなければデータに**過適合**してしまう．
※ $\lambda > 0$ は適切に選ぶ必要がある (クロスバリデーション)．
※ リッジ回帰と対応．

二次関数への展開

二乗損失は次のように展開される:

$$\sum_{i=1}^{n}(Y_i - f(X_i;\alpha))^2 = \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{n+2}\alpha_j B_j(X_i))^2$$

$$= \sum_{j=1}^{n+2}\sum_{j'=1}^{n+2}\alpha_j\alpha_{j'}\sum_{i=1}^{n}(B_j(X_i)B_{j'}(X_i)) - \sum_{j=1}^{n+2}\alpha_j\sum_{i=1}^{n}Y_i B_j(X_i) + Y^\top Y$$

$$=: \alpha^\top \mathbf{B}^\top \mathbf{B}\alpha - Y^\top \mathbf{B}\alpha + Y^\top Y.$$

正則化項は次のように展開される:

$$\int_{X_1}^{X_n}\left(\frac{\mathrm{d}^2 f(X_i;\alpha)}{\mathrm{d}x^2}\right)^2 \mathrm{d}x = \int_{X_1}^{X_n}\left(\sum_{j=1}^{n+2}\alpha_j\frac{\mathrm{d}^2 B_j(x)}{\mathrm{d}x^2}\right)^2 \mathrm{d}x$$

$$= \sum_{j=1}^{n+2}\sum_{j'=1}^{n+2}\alpha_j\alpha_{j'}\int_{X_1}^{X_n}\frac{\mathrm{d}^2 B_j(x)}{\mathrm{d}x^2}\frac{\mathrm{d}^2 B_{j'}(x)}{\mathrm{d}x^2}\mathrm{d}x$$

$$=: \sum_{j=1}^{n+2}\sum_{j'=1}^{n+2}\alpha_j\alpha_{j'}G_{j,j'} = \alpha^\top G\alpha.$$

二次式の最小化

$$\mathbf{B}_{i,j} = B_j(X_i),\ G_{j,j'} = \int_{X_1}^{X_n} \frac{\mathrm{d}^2 B_j(x)}{\mathrm{d}x^2} \frac{\mathrm{d}^2 B_{j'}(x)}{\mathrm{d}x^2} \mathrm{d}x,$$

として,

$$\begin{aligned}
& \min_{\alpha \in \mathbb{R}^{n+2}} \sum_{i=1}^{n} (Y_i - f(X_i; \alpha))^2 + \lambda \int_{X_1}^{X_n} \left(\frac{\mathrm{d}^2 f(x;\alpha)}{\mathrm{d}x^2} \right)^2 \mathrm{d}x \\
\Leftrightarrow \quad & \min_{\alpha \in \mathbb{R}^{n+2}} \alpha^\top \mathbf{B}^\top \mathbf{B} \alpha - Y^\top \mathbf{B} \alpha + \lambda \alpha^\top G \alpha \\
\Rightarrow \quad & \hat{\alpha} = (\mathbf{B}^\top \mathbf{B} + \lambda G)^{-1} \mathbf{B}^\top Y,
\end{aligned}$$

で回帰係数が求まる.

平滑化パラメータの決定: CV, GCV

さて，λ はどう選ばばよいか？→　例のごとくクロスバリデーション (CV).

$$\mathrm{CV} = \frac{\sum_{i=1}^{n}(Y_i - \hat{f}_{(-i)}(X_i))^2}{n}$$

$\hat{f}_{(-i)}(X_i)$ は (X_i, Y_i) を抜いて推定した 3 次スプライン関数.

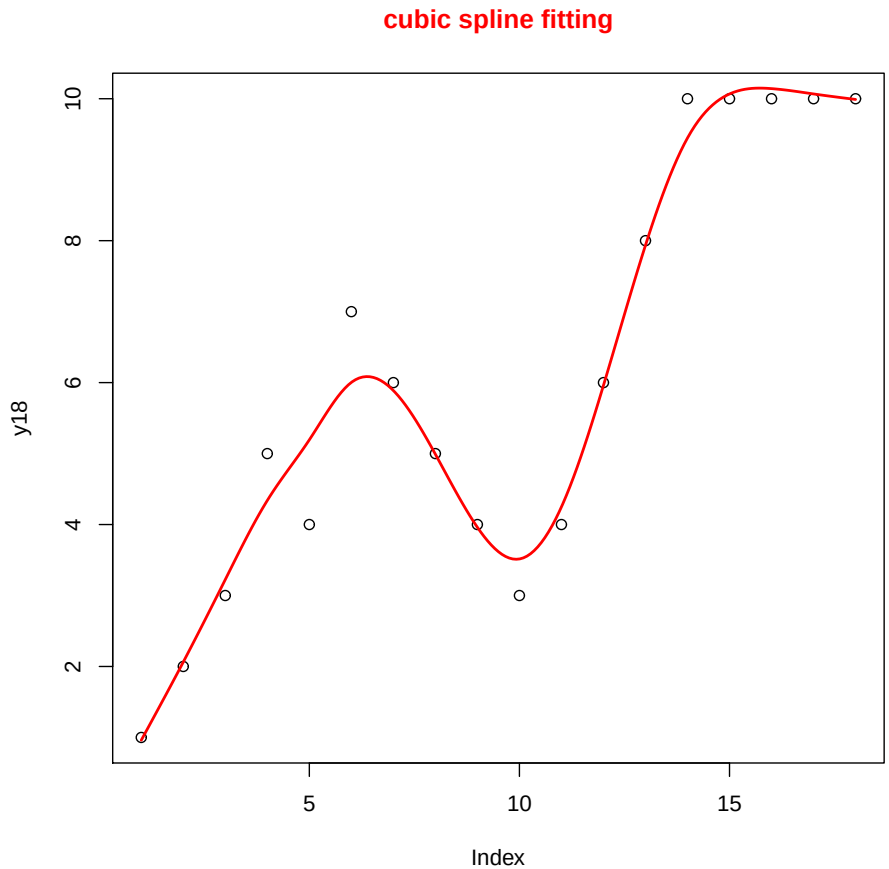
$$\mathrm{GCV} = \frac{\sum_{i=1}^{n}(Y_i - \hat{f}(X_i))^2}{n(1 - \mathrm{Tr}[A(\lambda)]/n)^2},$$

ただし，$A(\lambda) := \mathbf{B}(\mathbf{B}^\top\mathbf{B} + \lambda G)^{-1}\mathbf{B}^\top$. GCV は CV の計算を楽にするために提案されたが，統計的に良い性質があることが知られている.
R (smooth.spline) のデフォルトは GCV.

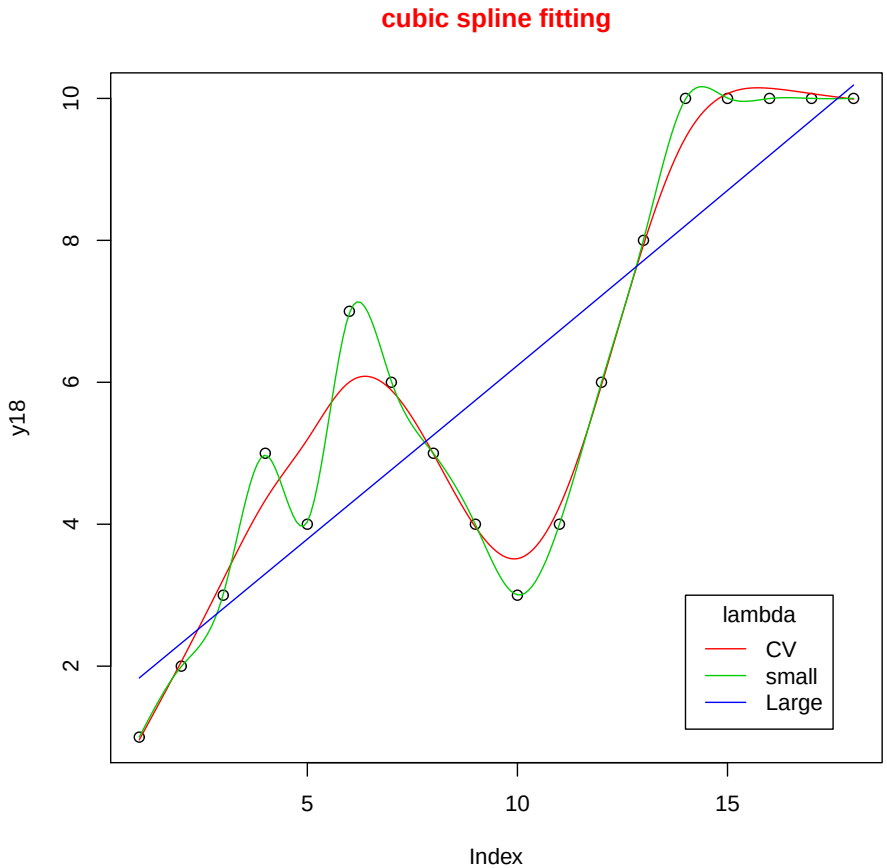
スプラインを実行

```
y18 <- c(1:3, 5, 4, 7:3, 2*(2:5), rep(10, 4))
artdata <- data.frame(x=1:18,y=y18)
s01  <- smooth.spline(artdata)
```

で自動的に λ も GCV で選択.



正則化パラメータの影響



正則化パラメータが小さいと過適合，大きすぎると線形回帰になる．
ちょうど良いパラメータは GCV で選ぶ．

```
s02  <- smooth.spline(artdata, spar = 0.02)
s03  <- smooth.spline(artdata, spar = 1)
```

スプラインを実行: gam

同様のことが mgcv パッケージに入っている gam という関数でも可能.

```
gam.s01 <- gam(y~s(x),data=artdata)
```

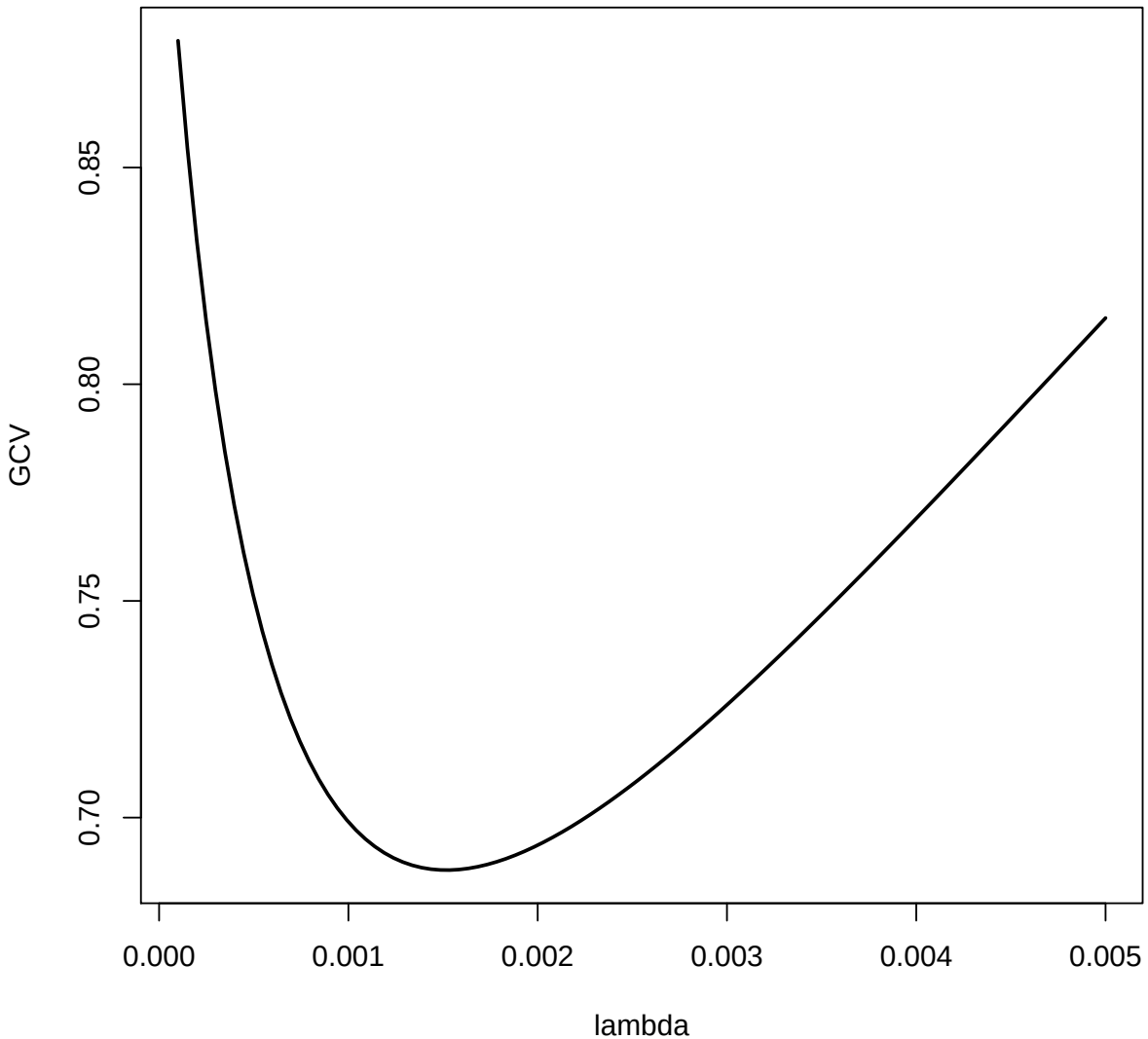
s(x) と書くことでスプラインを使うことを指定. 細かい次数や節点の設定も可能.
これで勝手に GCV で正則化パラメータを選んでフィッティングしてくれる.

gam の GCV 値

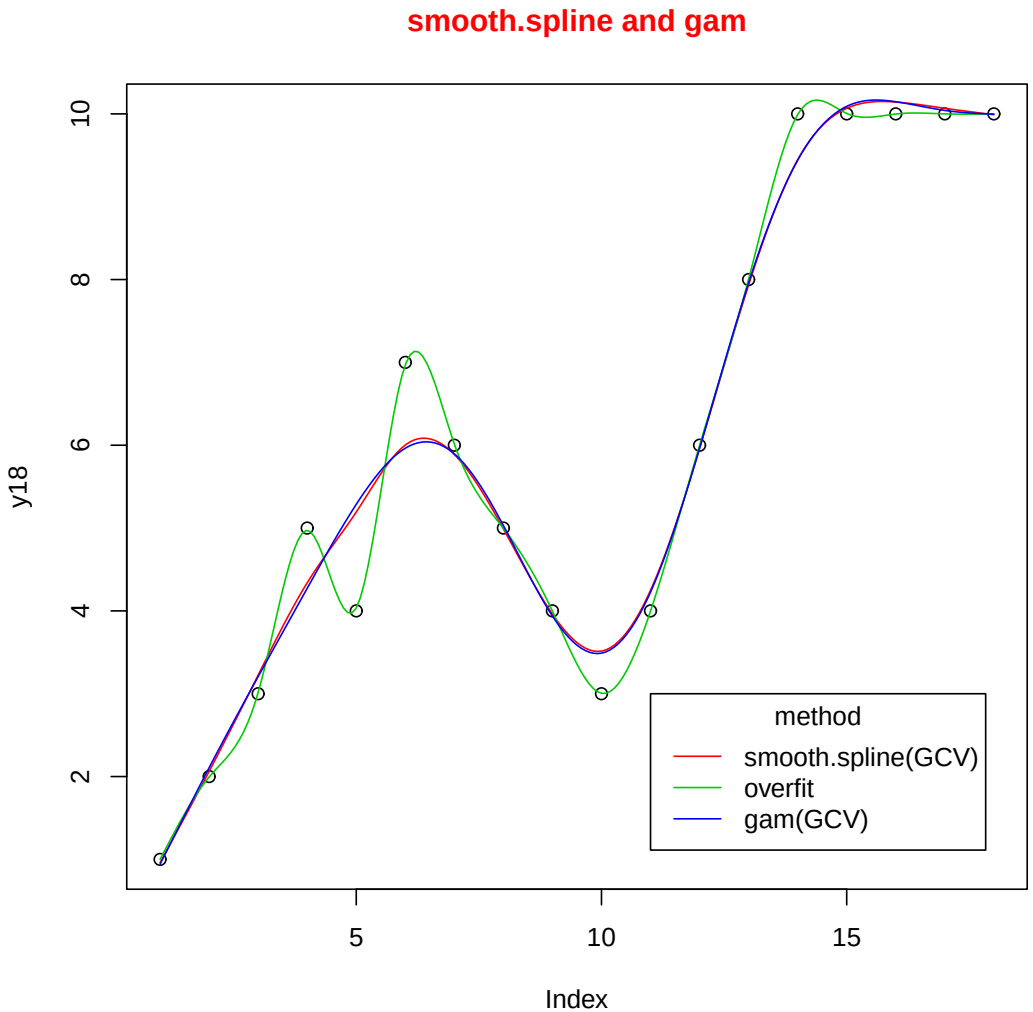
gam を使って，GCV 値を計算してみる．

```
sp<-seq(from=0.0001,to=0.005,length=100)
GCV_art<-numeric()
for(i in 1:length(sp)){
    g.m<-gam(y~s(x),sp=sp[i],data=artdata)
    GCV_art[i]<-g.m$gcv.ubre
}
plot(sp,GCV_art,type="l",lwd=2,xlab="lambda",ylab="GCV")
```

gam の GCV 値



smooth.spline と gam の比較



ほぼ同じ結果

まとめ

2 つのノンパラメトリックな推定手法を紹介した.

1. 密度推定→カーネル密度推定
2. 回帰→ B-スプライン回帰

どちらも，バンド幅や正則化パラメータといったパラメータを CV などを用いてうまく調整する必要があった.

講義情報ページ

http://www.is.titech.ac.jp/~s-taiji/lecture/dataanalysis/dataanalysis.html