

# RjpWiki アーカイブス

## 【Rの基本パッケージ中の平滑化関数一覧 (06.06.25)】

### 1 核関数による平滑化

#### 1.1 核関数を用いた平滑化

Nadaraya-Watson による核関数を用いた回帰平滑化を行う。

```
ksmooth(x, y, kernel = c("box", "normal"), bandwidth = 0.5,  
        range.x = range(x), n.points = max(100, length(x)), x.points)
```

### 2 多項式の局所的当てはめによる平滑化

#### 2.1 散布図平滑化

LOWESS 平滑化による計算を実行する。lowess89 は平滑結果の座標である x と y を成分に持つリストを返す。平滑結果は lines() 関数で元の散布図プロットに描き加えることができる。

```
lowess(x, y = NULL, f = 2/3, iter=3, delta = 0.01 * diff(range(xy$x[o])))
```

#### 2.2 散布図と Loess 平滑化曲線の同時プロット

散布図を描き、それに Loess による平滑化曲線を上描きする。

```
scatter.smooth(x, y, span = 2/3, degree = 1,  
              family = c("symmetric", "gaussian"),  
              xlab = deparse(substitute(x)), ylab = deparse(substitute(y)),  
              ylim = range(y, prediction$y), evaluation = 50, ...)  
loess.smooth(x, y, span = 2/3, degree = 1,  
            family = c("symmetric", "gaussian"), evaluation = 50, ...)
```

### 3 スプライン関数当てはめによる平滑化

#### 3.1 スプライン関数当てはめによる予測

スプライン関数当てはめによる新しい点での予測を行う。予測は元のデータ範囲外では直線になる。

```
## クラス "smooth.spline" に対する S3 メソッド
predict(object, x, deriv = 0, ...)
```

#### 3.2 スプライン関数による平滑化

与えられたデータに3次の平滑化スプライン関数を当てはめる。

```
smooth.spline(x, y = NULL, w = NULL, df, spar = NULL,
              cv = FALSE, all.knots = FALSE, nknots = NULL,
              df.offset = 0, penalty = 1, control.spar = list())
```

#### 3.3 散布図と Loess 平滑化曲線の同時プロット

散布図を描き、それに Loess による平滑化曲線を上描きする。

```
scatter.smooth(x, y, span = 2/3, degree = 1,
              family = c("symmetric", "gaussian"),
              xlab = deparse(substitute(x)), ylab = deparse(substitute(y)),
              ylim = range(y, prediction$y), evaluation = 50, ...)
loess.smooth(x, y, span = 2/3, degree = 1,
            family = c("symmetric", "gaussian"), evaluation = 50, ...)
```

## 4 移動直線平滑化、Friedman の supersmoother 法

### 4.1 Friedman の SuperSmoother

Friedman の「super smoother」を用いて (x,y) 値を平滑化する。

```
supsmu(x, y, wt, span = "cv", periodic = FALSE, bass = 0)
```

## 5 移動中央値による平滑化

### 5.1 移動中央値平滑化

奇数スパンの移動中央値 (running median) を計算する。これは考えられる限り「最も頑健」な散布図平滑化である。効率化 (そして歴史的な理由から)、同じ結果を与える二つの方法を選ぶことができる。

```
runmed(x, k, endrule = c("median", "keep", "constant"),
       algorithm = NULL, print.level = 0)
```

### 5.2 Tukey の移動中央値平滑化

Tukey の移動中央値平滑化 (running median) を行う。

```
smooth(x, kind = c("3RS3R", "3RSS", "3RSR", "3R", "3", "S"),
       twiceit = FALSE,
       endrule = "Tukey", do.ends = FALSE)
```

### 5.3 移動中央値に対する端点平滑化

最端点で順次小さくなる中央値と Tukey の端点規則を使用し、ベクトル  $y$  の端点を平滑化する。

```
smoothEnds(y, k = 3)
```