

## 「データ解析」(下平英寿)

## 講義資料3

## 一変量の統計学

- 目標：一変量(スカラ)の統計手法を理解する。
  1. MASSライブラリの利用
  2. 平均,分散,メディアンなど,主な要約統計量を理解する。
  3. ハズレ値が統計量に及ぼす影響について理解する。
  4. ヒストグラム,カーネル密度推定など,密度関数の推定法を学ぶ。

## 1 MASSライブラリ

```
> search()
[1] ".GlobalEnv"      "package:methods" "package:stats"   "package:graphics"
[5] "package:utils"   "Autoloads"       "package:base"
> library(MASS)
> search()
[1] ".GlobalEnv"      "package:MASS"    "package:methods" "package:stats"
[5] "package:graphics" "package:utils"   "Autoloads"       "package:base"
> library(help=MASS)
Information on Package 'MASS'
Description:
Package: MASS
Description: The main library and the datasets
Title: Main Package of Venables and Ripley's MASS
Bundle: VR
Priority: recommended
Version: 7.2-2
Date: 2004-05-23
Depends: R (>= 1.9.0), graphics, stats, lattice, nlme, survival
Author: S original by Venables & Ripley. R port by Brian Ripley
       <ripley@stats.ox.ac.uk>, following earlier work by Kurt Hornik
       and Albrecht Gebhardt.
Maintainer: Brian Ripley <ripley@stats.ox.ac.uk>
BundleDescription: Functions and datasets to support Venables and
                   Ripley, 'Modern Applied Statistics with S' (4th edition).
License: GPL (version 2 or later) See file LICENCE.
URL: http://www.stats.ox.ac.uk/pub/MASS4/
Packaged: Sun May 23 19:28:43 2004; riple
Built: R 1.9.1; i686-pc-linux-gnu; 2004-08-12 15:49:40; unix
Index:
Aids2           Australian AIDS Survival Data
Animals         Brain and Body Weights for 28 Species
Boston          Housing Values in Suburbs of Boston
Cars93          Data from 93 Cars on Sale in the USA in 1993
Cushings        Diagnostic Tests on Patients with Cushing's Syndrome
DDT             DDT in Kale
GAGurine        Level of GAG in Urine of Children
Insurance       Numbers of Car Insurance claims
Melanoma        Survival from Malignant Melanoma
Null            Null Spaces of Matrices
OME             Tests of Auditory Perception in Children with OME
Pima.tr         Diabetes in Pima Indian Women
Rabbit          Blood Pressure in Rabbits
Rubber          Accelerated Testing of Tyre Rubber
SP500           Returns of the Standard and Poors 500
Sitka           Growth Curves for Sitka Spruce Trees in 1988
Sitka89         Growth Curves for Sitka Spruce Trees in 1989
Skye            AFM Compositions of Aphyric Skye Lavas
Traffic         Effect of Swedish Speed Limits on Accidents
```

|                      |  |
|----------------------|--|
| UScereal             | Nutritional and Marketing Information on US Cereals    |
| UScrime              | The Effect of Punishment Regimes on Crime Rates        |
| VA                   | Veteran's Administration Lung Cancer Trial             |
| abbey                | Determinations of Nickel Content                       |
| accdeaths            | Accidental Deaths in the US 1973-1978                  |
| addterm              | Try All One-Term Additions to a Model                  |
| anorexia             | Anorexia Data on Weight Change                         |
| anova.negbin         | Likelihood Ratio Tests for Negative Binomial GLMs      |
| area                 | Adaptive Numerical Integration                         |
| austres-MASS         | Quarterly Time Series of the Number of                 |
| Australian Residents |  |
| bacteria             | Presence of Bacteria after Drug Treatments             |
| bandwidth.nrd        | Bandwidth for density() via Normal Reference           |
|                      | Distribution   |
| bcv                  | Biased Cross-Validation for Bandwidth Selection        |
| beav1                | Body Temperature Series of Beaver 1                    |
| beav2                | Body Temperature Series of Beaver 2                    |
| biopsy               | Biopsy Data on Breast Cancer Patients                  |
| birthwt              | Risk Factors Associated with Low Infant Birth Weight   |
| boxcox               | Box-Cox Transformations for Linear Models              |
| cabbages             | Data from a cabbage field trial                        |
| caith                | Colours of Eyes and Hair of People in Caithness        |
| cats                 | Anatomical Data from Domestic Cats                     |
| cement               | Heat Evolved by Setting Cements                        |
| chem                 | Copper in Wholemeal Flour                              |
| con2tr               | Convert Lists to Data Frames for use by Trellis        |
| confint-MASS         | Confidence Intervals for Model Parameters              |
| contr.sdif           | Successive Differences contrast coding                 |
| coop                 | Co-operative Trial in Analytical Chemistry             |
| corresp              | Simple Correspondence Analysis                         |
| cov.rob              | Resistant Estimation of Multivariate Location and      |
|                      | Scatter  |
| cov.trob             | Covariance Estimation for Multivariate t Distribution  |
| cpus                 | Performance of Computer CPUs                           |
| crabs                | Morphological Measurements on Leptograpsus Crabs       |
| deaths               | Monthly Deaths from Lung Diseases in the UK            |
| denumerate           | Transform an Allowable Formula for 'loglm'             |
|                      | into one for 'terms'                                   |
| dose.p               | Predict Doses for Binomial Assay model                 |
| drivers              | Deaths of Car Drivers in Great Britain 1969-84         |
| dropterm             | Try All One-Term Deletions from a Model                |
| eagles               | Foraging Ecology of Bald Eagles                        |
| epil                 | Seizure Counts for Epileptics                          |
| eqscplot             | Plots with Geometrically Equal Scales                  |
| farms                | Ecological Factors in Farm Management                  |
| fgl                  | Measurements of Forensic Glass Fragments               |
| fitdistr             | Maximum-likelihood Fitting of Univariate Distributions |
| forbes               | Forbes' Data on Boiling Points in the Alps             |
| fractions            | Rational Approximation                                 |
| galaxies             | Velocities for 82 Galaxies                             |
| gamma.dispersion     | Calculate the MLE of the Gamma Dispersion Parameter    |
| in a GLM Fit         |  |
| gamma.shape          | Estimate the Shape Parameter of the Gamma Distribution |
| in a GLM Fit         |  |
| gehan                | Remission Times of Leukaemia Patients                  |
| genotype             | Rat Genotype Data                                      |
| geyser               | Old Faithful Geyser Data                               |
| gilgais              | Line Transect of Soil in Gilgai Territory              |
| ginv                 | Generalized Inverse of a Matrix                        |
| glm.convert          | Change a Negative Binomial fit to a GLM fit            |
| glm.nb               | Fit a Negative Binomial Generalized Linear Model       |
| glmmPQL              | Fit Generalized Linear Mixed Models via PQL            |
| hills                | Record Times in Scottish Hill Races                    |
| hist.scott           | Plot a Histogram with Automatic Bin Width Selection    |
| housing              | Frequency Table from a Copenhagen Housing Conditions   |
| Survey               |  |
| huber                | Huber M-estimator of Location with MAD Scale           |
| hubers               | Huber Proposal 2 Robust Estimator of Location          |
|                      | and/or Scale   |
| wtloss               | Weight Loss Data from an Obese Patient                 |
| .... 以下略             |  |
| > help(huber)        |  |

```

huber                                package:MASS                                R Documentation
Huber M-estimator of Location with MAD Scale
Description:
  Finds the Huber M-estimator of location with MAD scale.
Usage:
  huber(y, k = 1.5, tol = 1e-06)
Arguments:
  y: vector of data values
  k: Winsorizes at 'k' standard deviations
  tol: convergence tolerance
Value:
  list of location and scale parameters
  mu: location estimate
  s: MAD scale estimate
References:
  Huber, P. J. (1981) Robust Statistics. Wiley.
  Venables, W. N. and Ripley, B. D. (2002) Modern Applied
  Statistics with S. Fourth edition. Springer.
See Also:
  'hubers', 'mad'
Examples:
  huber(chem)

```

## 2 データの「中心」,「バラツキ」,そして「ハズレ値」

### 2.1 chem データセットを見る

```

> chem
 [1] 2.90 3.10 3.40 3.40 3.70 3.70 2.80 2.50 2.40 2.40 2.70 2.20
[13] 5.28 3.37 3.03 3.03 28.95 3.77 3.40 2.20 3.50 3.60 3.70 3.70
> help(chem)
chem                                package:MASS                                R Documentation
Copper in Wholemeal Flour
Description:
  A numeric vector of 24 determinations of copper in wholemeal
  flour, in parts per million.
Usage:
  data(chem)
Source:
  Analytical Methods Committee (1989) Robust statistics - how not to
  reject outliers. The Analyst *114*, 1693-1702, 1989
References:
  Venables, W. N. and Ripley, B. D. (2002) Modern Applied
  Statistics with S. Fourth edition. Springer.

```

```

# run0019.R
# chem データを見る

library(MASS) # MASS ライブラリのロード
print(length(chem)) # データサイズ
plot(chem) # 縦軸 = データ, 横軸 = 番号
dev.copy2eps(file="chem-sp.eps")
boxplot(chem) # 箱ヒゲ図 (Box プロット)
dev.copy2eps(file="chem-bp.eps")
truehist(chem,nbins=20) # ヒストグラム (20 分割)
rug(chem) # 「ラグ」をプロット (下部にデータを示す線分)
lines(density(chem),col=2) # 密度関数をカーネル法で推定しプロット

```

```

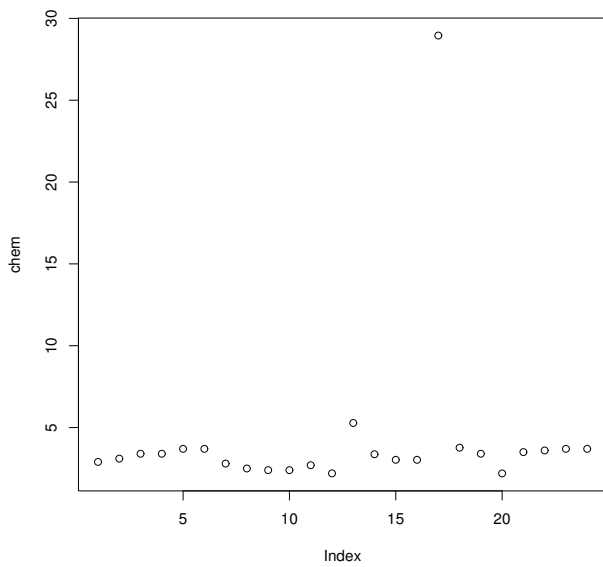
dev.copy2eps(file="chem-h.eps")
chem2 <- chem[chem < 5] # 5未満の要素だけ取り出して再描画
print(length(chem2))
truehist(chem2,nbins=8)
rug(chem2)
lines(density(chem2),col=2)
dev.copy2eps(file="chem-h2.eps")

```

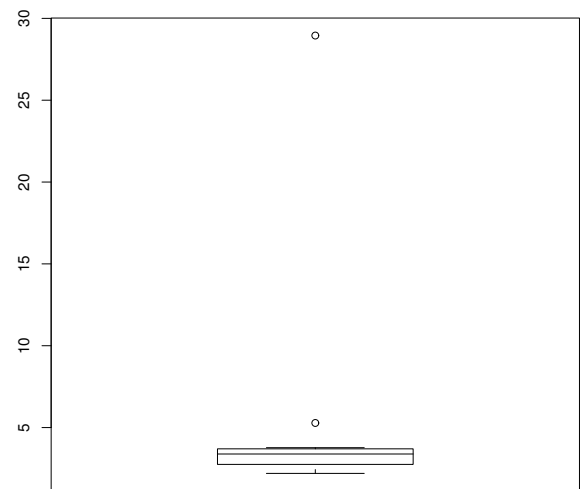
```
> source("run0019.R")
```

```
[1] 24
```

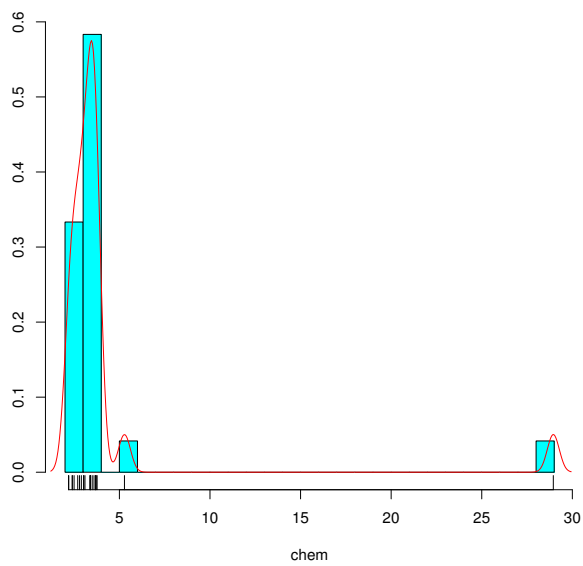
```
[1] 22
```



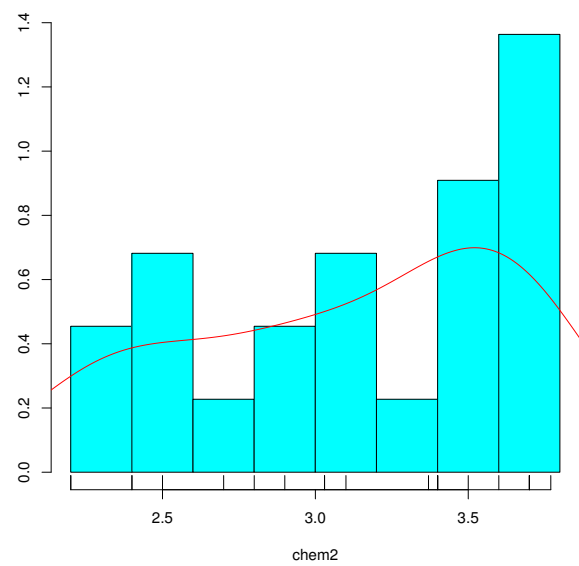
chem-sp



chem-bp



chem-h



chem-h2

## 2.2 平均，標準偏差

- サイズ  $n$  のデータ

$$x_1, \dots, x_n$$

- (標本) 平均  $\bar{x}$

$$\bar{x} = \frac{x_1 + \dots + x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i$$

- 不偏標本分散  $S_x^2$

$$S_x^2 = \frac{(x_1 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{n - 1} = \frac{1}{n - 1} \sum_{i=1}^n (x_i - \bar{x})^2$$

ただし分母の  $n - 1$  を  $n$  で置き換えたものは、(不偏ではない) 標本分散。実際には、「不偏」や「標本」を省いて書いてしまうことが多いので注意。

- 標本標準偏差  $S_x$  .

$$S_x = \sqrt{S_x^2}$$

```
# run0020.R
# chem データの平均，標準偏差

print(mean(chem)) # 平均
print(sum(chem)/length(chem)) # 定義に戻って計算
mymean <- function(x) sum(x)/length(x) # 関数定義
print(mymean(chem)) # 関数呼び出し

print(sqrt(var(chem))) # 標準偏差
mysd <- function(x) sqrt(sum((x-mymean(x))^2)/(length(x)-1)) # 関数定義
print(mysd(chem)) # 関数呼び出し
```

```
> source("run0020.R")
```

```
[1] 4.280417
```

```
[1] 4.280417
```

```
[1] 4.280417
```

```
[1] 5.297396
```

```
[1] 5.297396
```

## 2.3 メディアン，四分位偏差

- chem データセットには「ハズレ値」(outlier) とみなせるような大きな値が含まれる (5.28 と 28.95) .
- この 2 個の「ハズレ値」を取り除いたデータセットが chem2 .

- 平均や標準偏差などの統計量はハズレ値に弱い。つまり、測定のミスなどによって少数のハズレ値が混入するだけで、統計量の値が大きく変化してしまう。
- ハズレ値に強い統計量としては、メディアンや四分位偏差が知られている。つまり、平均 ⇒ メディアン、標準偏差 ⇒ 四分位偏差と置き換えると、少数のハズレ値によって統計量が大きく変化することは無くなる。
- ハズレ値に強いということは、裏を返せば、データの微妙な変化を捉えることができず感度が悪いことを意味する。またハズレ値に強い統計量は、一般にデータの一部を捨てるので、推定の効率が悪い（標準誤差が大きい）。このように、ハズレ値に強いということと推定効率はトレードオフの関係にある。

```
# run0021.R
# chem データの平均 (mean), 標準偏差 (sd), メディアン (median), 四分位偏差 (iqr)

myfournum <- function(x) {
  m1 <- mean(x) # 平均
  s1 <- sqrt(var(x)) # 標準偏差
  m2 <- median(x) # メディアン
  s2 <- IQR(x)/1.3489795 # 四分位偏差 (Interquartile Range) を調整したもの

  list(mean=m1,sd=s1,meadian=m2,iqr=s2)
}

print(unlist(myfournum(chem)))
print(unlist(myfournum(chem2)))
```

```
> source("run0021.R")
      mean      sd  meadian      iqr
4.2804167 5.2973960 3.3850000 0.6857035
      mean      sd  meadian      iqr
3.1136364 0.5299375 3.2350000 0.6301059
```

- chem の median と iqr は、chem2 の mean と sd にかなり近い値を与えている。つまり、meadian と iqr は、自動的にハズレ値を取り除いてから mean と sd を計算することに、相当する（大体の話）。
- このようにハズレ値を取り除くことが、そもそも適切か不適切かは、意見の分かれるところである。

```
> x <- rnorm(10000,mean=5,sd=1)
> unlist(myfournum(x))
      mean      sd  meadian      iqr
4.9955791 0.9984380 4.9979840 0.9974582
```

```
> myfournum(x) # unlist をつけないと, こうなる
```

```
$mean
```

```
[1] 4.995579
```

```
$sd
```

```
[1] 0.998438
```

```
$median
```

```
[1] 4.997984
```

```
$iqr
```

```
[1] 0.9974582
```

- メディアンと四分位偏差は, 分位点 (quantile) またはパーセンタイル (percentile) から計算される (quantile と percentile は, ほぼ同義語として使われている)
- $p$ -分位点 ( $100p$  パーセンタイル) は, データ  $x = (x_1, \dots, x_n)$  を小さい順にソートしたとき,  $1 + (n - 1)p$  番目の数値に相当する. これが整数値でない場合には前後の値から線形補間などする. 大雑把に言えば,  $np$  番目に小さい値である. これを計算する関数は `quantile(x,p)` である.  $p$  にベクトルも可能. `help(quantile)` を参照.

```
> quantile(chem,c(0,0.25,0.5,0.75,1))
```

```
 0%    25%    50%    75%   100%
2.200 2.775 3.385 3.700 28.950
```

- メディアン=50 パーセンタイル, つまりちょうど真ん中. 左右対称にデータが分布していれば, 平均値に等しい.
- 四分位偏差=75 パーセンタイル - 25 パーセンタイル. `IQR(x)` は  $x$  の四分位偏差を与える. `IQR(x) = quantile(x,3/4) - quantile(x,1/4)`. ただし `run0021.R` で計算している `irq` は, これを  $2 * \text{qnorm}(3/4) = 1.3490$  で割っている. これは, もし  $x$  が正規分布に従うとき, (十分に  $n$  が大きければ,) 標準偏差に等しい値を与えるためである. `help(IQR)` を参照.
- 箱ヒゲ図 (Box プロット) の箱の中心=メディアン, 箱の両端 (hinges)=ほぼ 25%と 75%のパーセンタイル点.

```
IQR package:stats
```

```
R Documentation
```

```
The Interquartile Range
```

```
Description:
```

```
  computes interquartile range of the 'x' values.
```

```
Usage:
```

```
  IQR(x, na.rm = FALSE)
```

```
Arguments:
```

```
  x: a numeric vector.
```

```
  na.rm: logical. Should missing values be removed?
```

```
Details:
```

```
  Note that this function computes the quartiles using the
  'quantile' function rather than following Tukey's recommendations,
```

i.e., 'IQR(x) = quantile(x,3/4) - quantile(x,1/4)'.

For normally  $N(m,1)$  distributed  $X$ , the expected value of 'IQR(X)' is '2\*qnorm(3/4) = 1.3490', i.e., for a normal-consistent estimate of the standard deviation, use 'IQR(x) / 1.349'.

References:

Tukey, J. W. (1977). *Exploratory Data Analysis.* Reading: Addison-Wesley.

See Also:

'fivenum', 'mad' which is more robust, 'range', 'quantile'.

Examples:

```
IQR(rivers)
```

## 2.4 刈り込み平均

- 平均よりハズレ値に強い「中心」の推定量は、メディアンだけではなく、他にもいろいろある。
- オリンピックの採点。5人の採点結果から、最高点と最低点を除いた3人の平均値。
- より一般的には、刈り込み平均 (trimmed mean)。100 $\alpha$ パーセントの刈り込み平均では、100 $\alpha$ パーセンタイルと100(1 -  $\alpha$ )パーセンタイルの間のデータの平均。つまり両側であわせて200 $\alpha$ パーセントのデータを捨ててから平均を取る。(0  $\leq$   $\alpha$   $\leq$  0.5とする。)
- $\alpha = 0.5$ とすれば、メディアンに一致。
- `mean(x, trim=a)` は刈り込み平均 ( $a=\alpha$ ) を計算する。

```
> mean(chem) # 0%の刈り込み平均
```

```
[1] 4.280417
```

```
> mean(chem, trim=0.1) # 10%の刈り込み平均
```

```
[1] 3.205
```

```
> median(chem) # 50%の刈り込み平均
```

```
[1] 3.385
```

```
> sapply(c(0,0.1,0.2,0.3,0.4,0.5),function(a) mean(chem, trim=a))
```

```
[1] 4.280417 3.205000 3.239375 3.273000 3.283333 3.385000
```

## 2.5 MAD

- 標準偏差よりハズレ値に強い「バラツキ」の推定量は、四分位偏差だけでなく、他にもいろいろある。
- まず  $x = (x_1, \dots, x_n)$  のメディアンを  $\text{median}_i(x_i)$  と書く。
- バラツキの推定量である MAD (Median Absolute Deviation) は、

$$\text{MAD} = \text{median}_i(|x_i - \text{median}_j(x_j)|) / 0.6745$$

- つまりデータのメディアンを中心として、データのズレの絶対値をメディアンで測る。最後に0.6745で割るのは、正規分布の場合の標準偏差に一致させるため。



```
> mad(chem) # MAD
[1] 0.526323
> myfournum(chem)$iqr # 四分位偏差
[1] 0.6857035
```

## 2.6 頑健統計量について

- ハズレ値に強い統計量を頑健統計量（ロバスト統計量）という。たとえば「中心」: メディアン, 刈り込み平均。「バラツキ」: 四分位偏差, MAD.
- これ以外にもさまざまな手法が提案されている。一般理論は1980年代に整備された。 $M$ -推定量というクラスが理論の中心。(Mは「MLEと類似の」というところから来ている。MLEというのは後述する最尤推定量の意味)
- $M$ -推定量の例として, MASSライブラリの `huber` 関数と `hubers` 関数は, どちらも中心とバラツキを同時に計算する。ただし `huber` の返すバラツキは MAD。詳細は `help(huber)`, `help(hubers)` を参照。
- いろいろ提案されていて, 性能の理論的な比較もあるが, 結局どれが良いのか, 意見の分かれるところであろう(「場合による」ということ。)

```
> unlist(huber(chem)) # Huber 法 . mu=中心, s=バラツキ
      mu      s
3.206724 0.526323
> unlist(hubers(chem)) # Huber 法その2 . mu=中心, s=バラツキ
      mu      s
3.205498 0.673652
```

## 2.7 最尤推定量

- 最尤推定量 (maximum likelihood estimator の頭文字をとって MLE と呼ばれる) は, 最尤法 (最大尤度法) によって得られる推定量。
- 最尤法は非常に一般的な概念で, 統計学において中心的な役割を果たしている。
- まずデータの従う確率モデルを一つ指定し, それが真実であると仮定する。たとえば,  $x_1, \dots, x_n$  が平均  $\mu$ , 標準偏差  $\sigma$  の正規分布  $N(\mu, \sigma^2)$  に従うと仮定する。
- 仮定したモデルから観測したデータが得られる確率 (密度) を計算する。

$$L(\theta) = \prod_{i=1}^n f(x_i|\theta)$$

$$f(x_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

ただし,  $\theta = (\mu, \sigma)$  は未知パラメタである。ここではデータ  $x$  はすでに観測されていて固定しているので,  $L(\theta)$  はパラメタ  $\theta$  の関数とみなす。このとき  $L(\theta)$  を尤度 (likelihood) と呼ぶ。

- $L(\theta)$  を最大にするような  $\theta$  を探索し、これを  $\hat{\theta}$  と置く。つまり、

$$\max_{\theta \in \Theta} L(\theta) = L(\hat{\theta})$$

である。  $\Theta$  は  $\theta$  が取る値の集合。正規分布の場合なら、

$$\Theta = \{(\mu, \sigma) : -\infty < \mu < \infty, 0 < \sigma < \infty\}$$

このようにして求めた  $\hat{\theta}$  が、最尤推定量である。

- 対数尤度 (log-likelihood) とは、尤度の対数。つまり、 $\ell(\theta) = \log L(\theta)$  のこと。

$$\ell(\theta) = \sum_{i=1}^n \log f(x_i | \theta)$$

$L(\theta)$  よりも、 $\ell(\theta)$  のほうが取り扱いが容易なことが多い。正規分布の場合なら、

$$\ell(\theta) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$$

- 数値的最適化には `optim` 関数が見える。これは標準で最小化を行うので、最大化を実行するにはオプションで `control=list(fnscale=-1)` を指定する。 $\ell(\theta)$  の最大化を行う代わりに、 $-\ell(\theta)$  の最小化を行うことにすれば、この `control` オプションは不要である。`optim` は内部で反復計算を実行しているので、その初期値を与える必要がある。`optim` 内部で採用している最適化アルゴリズムは数種類あり、それを `method` オプションで指定できる。`method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN")`。アルゴリズムの説明などは `help(optim)` を参照。

```
# run0022.R
# 最尤推定量 (正規分布) .

x <- chem # chem データセット
lik <- function(th) sum(log(dnorm(x,mean=th[1],sd=th[2]))) # 対数尤度

th <- list(mu=seq(2,6,length=100),
           sigma=seq(0.1,10,length=100)) # プロットする範囲
z <- matrix(apply(expand.grid(th),1,lik),
            length(th[[1]])) # すべての格子点で対数尤度を計算
image(th$mu,th$sigma,exp(z)) # 尤度の2次元プロット (赤 = 低い, 白 = 高い)
contour(th$mu,th$sigma,exp(z),add=T) # 等高線の表示
dev.copy2eps(file="run0022-cp1.eps")
z2 <- pmax(z,-100) # プロットのレンジを調整
image(th$mu,th$sigma,z2) # 対数尤度の2次元プロット (赤 = 低い, 白 = 高い)
contour(th$mu,th$sigma,z2,add=T) # 等高線の表示
dev.copy2eps(file="run0022-cp2.eps")
```

```

opt <- optim(c(4,5),lik,control=list(fnscale=-1)) # 数値的最適化
print(opt$par) # 最尤推定量の表示
truehist(x,nbins=20) # ヒストグラム
rug(x)
x0 <- seq(min(x),max(x),length=300)
lines(x0,dnorm(x0,mean=opt$par[1],sd=opt$par[2]),col=2) # 密度関数
dev.copy2eps(file="run0022-h.eps")

print(fitdistr(x,"normal")) # これでも良い.

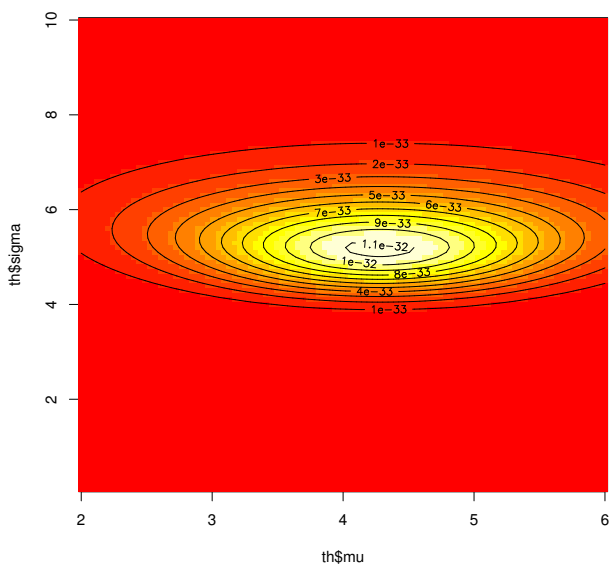
print(mean(x)) # 平均
print(sqrt(sum( (x-mean(x))^2 )/length(x))) # (不偏でない) 標本標準偏差

```

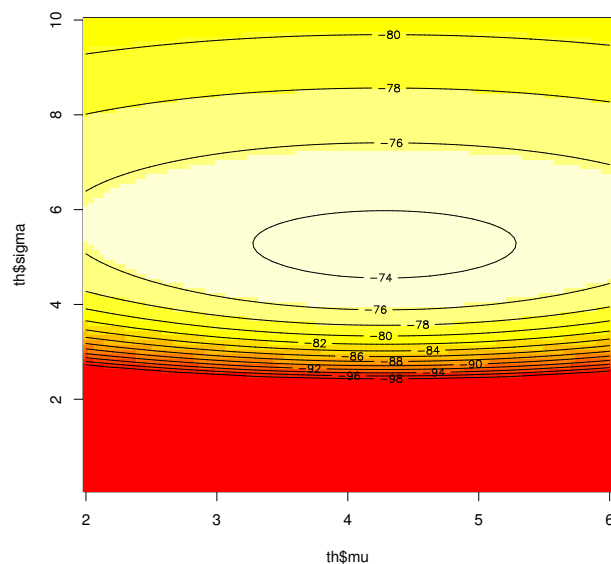
```

> source("run0022.R")
[1] 4.279753 5.185585
      mean      sd
4.2804167 5.1858594
(1.0585591) (0.7485143)
[1] 4.280417
[1] 5.185859

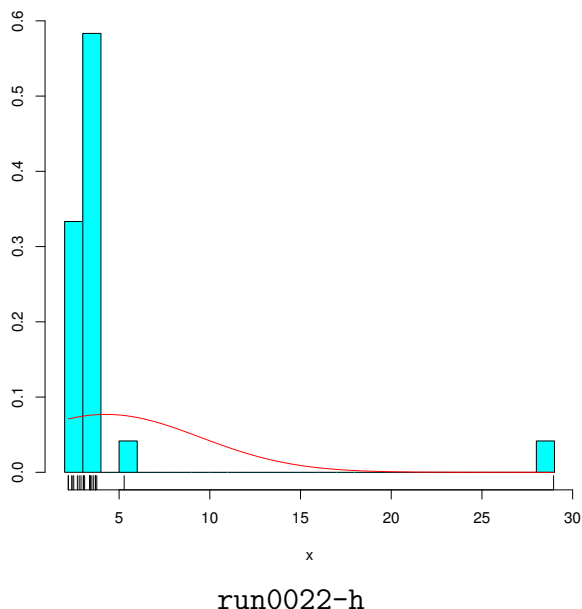
```



run0022-cp1



run0022-cp2



- 簡単な確率モデルの最尤推定量は，数値的最適化の必要はなく，解析的に解が求まる．正規分布の場合なら，

$$\frac{\partial \ell(\theta)}{\partial \mu} = 0, \quad \frac{\partial \ell(\theta)}{\partial (\sigma^2)} = 0$$

を解くと， $\hat{\theta} = (\hat{\mu}, \hat{\sigma})$  として，

$$\hat{\mu} = \frac{x_1 + \dots + x_n}{n}, \quad \hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2}$$

結局， $\hat{\mu} = \bar{x}$  は平均， $\hat{\sigma} = \sqrt{\frac{n-1}{n}} S_x$  (不偏でない) 標本標準偏差になる．

- 従って，正規分布を仮定した最尤推定量は，ハズレ値に弱く，ロバストでない．前節で説明した  $M$ -推定量は，実は，最尤推定量を一般化してロバストにしたものである．しかし最尤推定量でも，次の示すようにロバストにすることも可能である．

## 2.8 最尤推定量 (その2)

- 以上では説明のために正規分布を用いたが，たとえば  $t$ -分布をモデルとして仮定しても良い．自由度  $m$  の  $t$ -分布の密度関数は

$$f(x_i|\theta) = \frac{\Gamma((m+1)/2)}{\sqrt{m\pi\sigma^2}\Gamma(m/2)} \left(1 + \frac{(x_i - \mu)^2}{m\sigma^2}\right)^{-(m+1)/2}$$

である． $t$ -分布は正規分布より，分布のスソが重く，ハズレ値に強い推定量が得られる．自由度パラメタ  $m$  を小さくするとスソが重くなり， $m$  を大きくするとスソが軽くなる． $m = 1$  がコーシー分布， $m \rightarrow \infty$  が正規分布である．分布の中心は  $\mu$  である． $\sigma$  はバラツキの程度を表す．分散は  $V(X) = \sigma^2 n / (n - 2)$  ただし  $n > 2$  である．

```
# t-分布の密度関数
```

```
x <- seq(-7,7,length=400) # プロットする範囲
```

```
deg <- c(1,2,5,30,100) # 自由度
```

```
col <- 2:6 # 色
```

```
plot(x,dnorm(x),type="l") # 縦軸 = 確率密度
```

```
for(i in 1:length(deg)) lines(x,dt(x,df=deg[i]),col=col[i])
```

```
legend(2,0.4,paste("deg=",c(deg,Inf)),col=c(col,1),lty=1,bty="n")
```

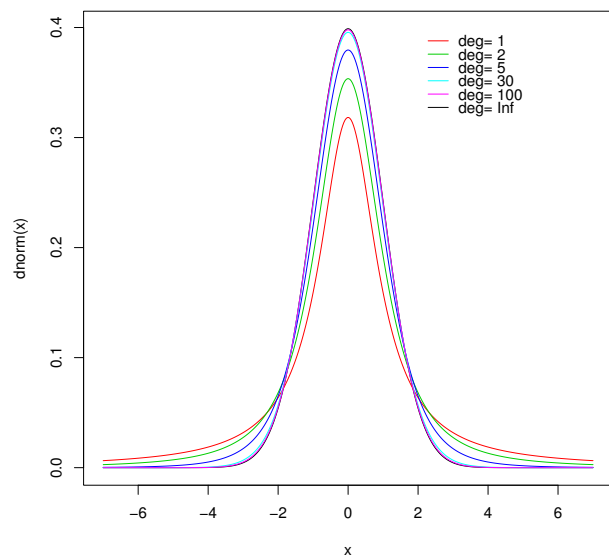
```
dev.copy2eps(file="run0026-dt1.eps")
```

```
plot(x,dnorm(x),type="l",log="y") # 縦軸 = 確率密度の対数
```

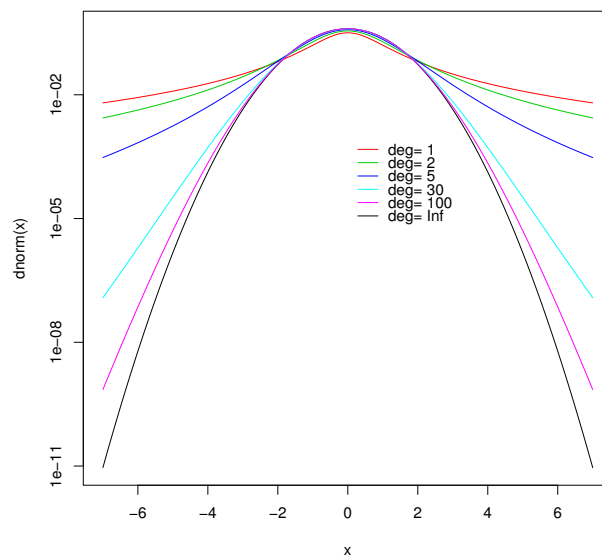
```
for(i in 1:length(deg)) lines(x,dt(x,df=deg[i]),col=col[i])
```

```
legend(0,1e-3,paste("deg=",c(deg,Inf)),col=c(col,1),lty=1,bty="n")
```

```
dev.copy2eps(file="run0026-dt2.eps")
```



run0026-dt1



run0026-dt2

```
# run0023.R
```

```
# 最尤推定量 (t-分布) .
```

```
x <- chem # chem データセット
```

```
deg <- 2 # 自由度は2にしておく .
```

```
mydt <- function(x,mean,sd,df)
```

```
  dt((x-mean)/sd,df=df)/sd # 標準の dt は mean=0, sd=1 に相当
```

```
lik <- function(th) sum(log(mydt(x,th[1],th[2],deg))) # 対数尤度
```

```
th <- list(mu=seq(2,6,length=100),
```

```
          sigma=seq(0.1,3,length=100)) # プロットする範囲
```

```
z <- matrix(apply(expand.grid(th),1,lik),
```

```

length(th[[1]])) # すべての格子点で対数尤度を計算
image(th$mu,th$sigma,exp(z)) # 尤度の2次元プロット(赤=低い,白=高い)
contour(th$mu,th$sigma,exp(z),add=T) # 等高線の表示
dev.copy2eps(file="run0023-cp1.eps")
z2 <- pmax(z,-100) # プロットのレンジを調整
image(th$mu,th$sigma,z2) # 対数尤度の2次元プロット(赤=低い,白=高い)
contour(th$mu,th$sigma,z2,add=T) # 等高線の表示
dev.copy2eps(file="run0023-cp2.eps")

opt <- optim(c(3,1),lik,control=list(fnscale=-1)) # 数値的最適化
print(opt$par) # 最尤推定量の表示
truehist(x,nbins=20) # ヒストグラム
x0 <- seq(min(x),max(x),length=300)
lines(x0,mydt(x0,mean=opt$par[1],sd=opt$par[2],df=deg),col=2) # 密度関数
dev.copy2eps(file="run0023-h.eps")

print(fitdistr(x,"t",df=deg)) # これでも良い.
print(fitdistr(x,"t")) # 自由度も推定できるが,数値的に不安定になる.
print(sapply(c(1,2,5,10,100,1000),
             function(deg) fitdistr(x,"t",df=deg)$estimate))

```

```
> source("run0023.R")
```

```
[1] 3.2169429 0.5051544
```

```
      m      s
```

```
3.2169663 0.5051234
```

```
(0.1400725) (0.1100541)
```

```
      m      s      df
```

```
3.2484136 0.4553490 1.3670314
```

```
(0.1475079) (0.1190068) (0.4974041)
```

```
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
```

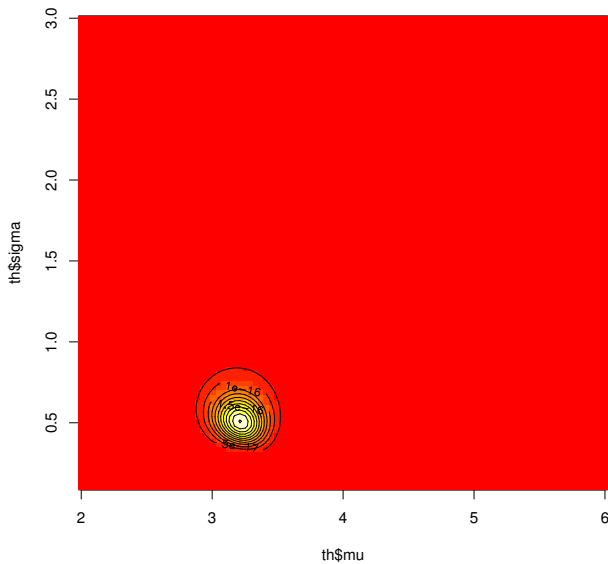
```
m 3.285039 3.2169663 3.1853242 3.2001124 4.048658 4.256082
```

```
s 0.412430 0.5051234 0.6421415 0.8326225 4.619992 5.131446
```

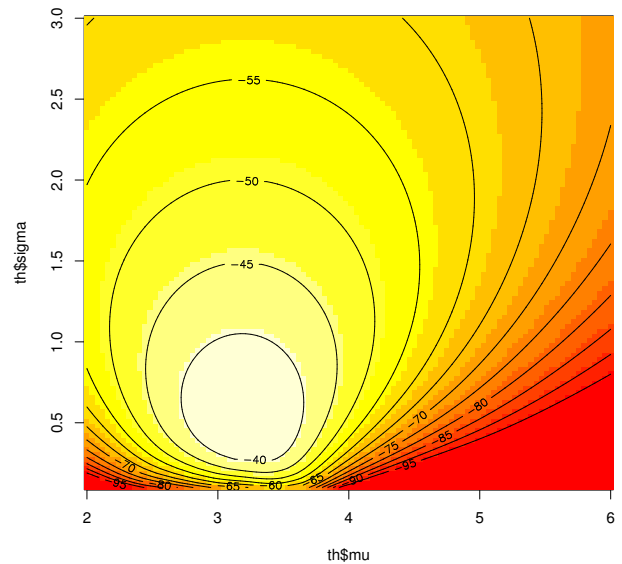
```
Warning message:
```

```
NaNs produced in: dt(x, df, log)
```

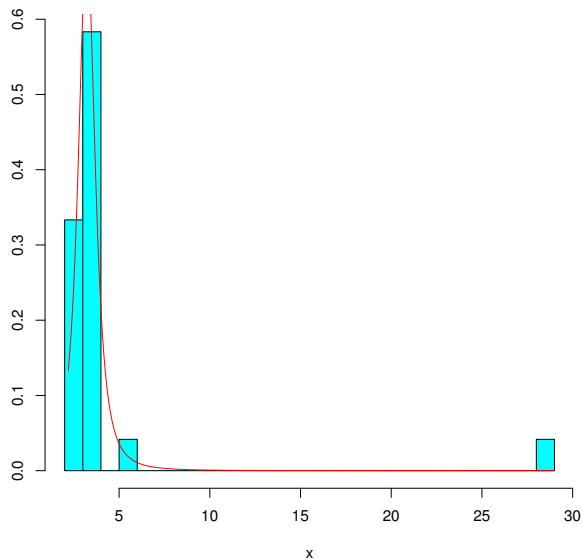
最後の Warning は `fitdistr(x,"t")` が出したもの.



run0023-cp1



run0023-cp2



run0023-h

- 正規分布に従うデータに  $t$ -分布を当てはめる場合を考える .
- このとき ,  $t$ -分布の  $\sigma$  をそのまま用いると , 正規分布の標準偏差を過小評価してしまう . 従って , これを補正する係数を求めておく .  $n$  が十分に大きい場合を想定して  $n \rightarrow \infty$  とする .
- $t$ -分布の密度関数 ( $\mu = 0, \sigma > 0$ ) の対数に正規分布 (平均 0, 分散 1) の密度関数をかけて積分する

$$\ell(\sigma) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \log \left\{ \frac{\Gamma((m+1)/2)}{\sqrt{m\pi\sigma^2}\Gamma(m/2)} \left(1 + \frac{x^2}{m\sigma^2}\right)^{-(m+1)/2} \right\} dx$$

これを  $\sigma$  で微分すると ,

$$\frac{d\ell(\sigma)}{d\sigma} = \frac{1+m}{s} \left\{ \frac{m}{m+1} + \frac{\Phi(\sqrt{m}\sigma) - 1}{\phi(\sqrt{m}s)/(\sqrt{m}s)} \right\}$$

ただし  $\phi(x)$  と  $\Phi(x)$  は  $N(0,1)$  の密度関数と分布関数．従って， $d\ell(\sigma)/d\sigma = 0$  を数値的に解いた値を  $\tilde{\sigma}_m$  とする．たとえば， $\tilde{\sigma}_5 = 0.85662$

- データに自由度  $m$  の  $t$ -分布を当てはめて得られた  $\hat{\sigma}$  を  $\tilde{\sigma}_m$  で割れば，補正済みの標準偏差になる．
- 研究者の方へ：この補正項の導出に関する文献をご存知の方は，下平までお知らせ願えれば幸いです．

```
# run0040.R
# t-分布の補正
tsdcorrection <- function(m) { # 補正項の計算
  dlik <- function(x) m/(1+m) + (pnorm(x)-1)/(dnorm(x)/x)
  f <- uniroot(dlik,c(0,sqrt(m)))
  f$root/sqrt(m)
}
m0 <- 1:10 ; names(m0) <- m0 # 1から10まで自由度を変えて
print(sapply(m0,tsdcorrection)) # 補正項を表示
x <- rnorm(10000) # 乱数N(0,1)を10000個つくりデータとする．
print(sd(x)) # 標準偏差
s <- sapply(m0,function(m)
             fitdistr(x,"t",df=m)$estimate["s"]/tsdcorrection(m)) # 補正済み
print(s)
```

```
> source("run0040.R")
```

```
      1      2      3      4      5      6      7      8
0.6120098 0.7326011 0.7935125 0.8310038 0.8566214 0.8753183 0.8896034 0.9008930
      9     10
0.9100508 0.9176348
[1] 1.006540
      1.s      2.s      3.s      4.s      5.s      6.s      7.s      8.s
1.001654 1.003168 1.004288 1.004766 1.005143 1.005333 1.005699 1.005746
      9.s     10.s
1.005665 1.005914
```

### 3 密度推定

- 中心，バラツキの統計量は，データを要約する統計量として有用であった．頑健統計量を用いれば，多少のハズレ値に対して自動的に対応できた．
- ところが，中心，バラツキといった概念は，分布の形状が一つ山（+多少のハズレ値）の場合は良いが，分布の形状が二つ山になると，要約統計量として不十分である．



- やはり、データの従う確率密度関数を推定して分布の形状を直接表現することが有用である。つまり、「まずデータを見る」ということ。

### 3.1 geysers データセットを見る

```
geyser                package:MASS                R Documentation
Old Faithful Geysers Data
Description:
  A version of the eruptions data from the 'Old Faithful' geysers in
  Yellowstone National Park, Wyoming. This version comes from
  Azzalini and Bowman (1990) and is of continuous measurement from
  August 1 to August 15, 1985.
  Some nocturnal duration measurements were coded as 2, 3 or 4
  minutes, having originally been described as 'short', 'medium' or
  'long'.
Usage:
  data(geyser)
Format:
  A data frame with 299 observations on 2 variables.
  'duration' numeric Eruption time in mins
  'waiting'   numeric Waiting time to next eruption
References:
  Azzalini, A. and Bowman, A. W. (1990) A look at some data on the
  Old Faithful geysers. Applied Statistics *39*, 357-365.
  Venables, W. N. and Ripley, B. D. (2002) Modern Applied
  Statistics with S. Fourth edition. Springer.
See Also:
  'faithful'
```

```
# run0024.R
# geysers データを見る

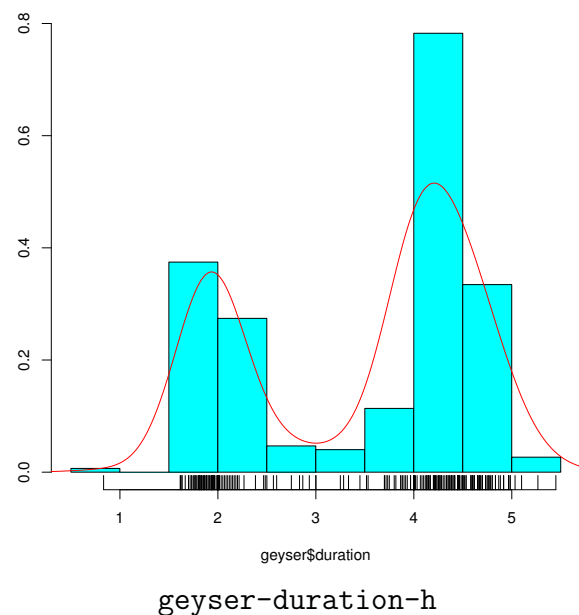
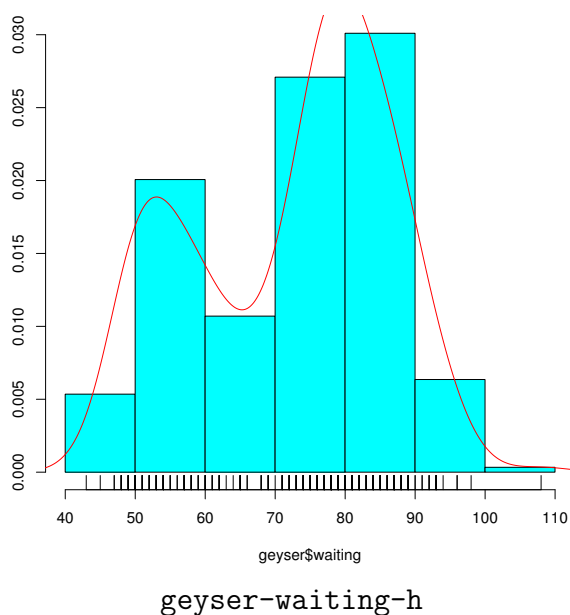
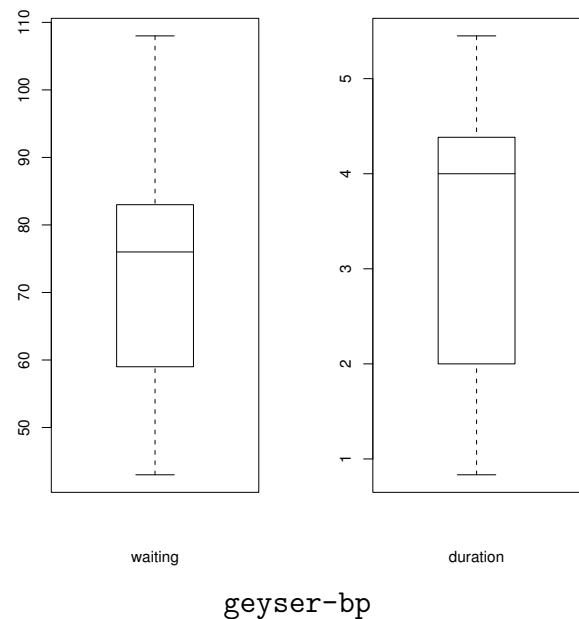
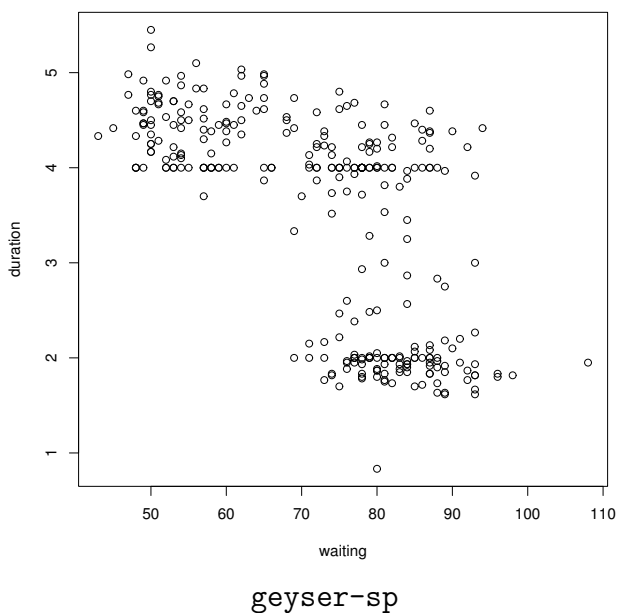
library(MASS) # MASS ライブラリのロード
print(names(geyser)) # 変数の名前
print(dim(geyser)) # データサイズ
plot(geyser)
dev.copy2eps(file="geyser-sp.eps")
par(mfrow=c(1,2))
boxplot(geyser$waiting,xlab="waiting") # 箱ヒゲ図 (Box プロット)
boxplot(geyser$duration,xlab="duration") # 箱ヒゲ図 (Box プロット)
dev.copy2eps(file="geyser-bp.eps")
par(mfrow=c(1,1))

print(unlist(myfournum(geyser$waiting))) # 平均等
truehist(geyser$waiting) # ヒストグラム
rug(geyser$waiting) # 「ラグ」をプロット (下部にデータを示す線分)
lines(density(geyser$waiting),col=2) # 密度関数をカーネル法で推定しプロット
dev.copy2eps(file="geyser-waiting-h.eps")

print(unlist(myfournum(geyser$duration))) # 平均等
truehist(geyser$duration) # ヒストグラム
```

```
rug(geyser$duration) # 「ラグ」をプロット（下部にデータを示す線分）
lines(density(geyser$duration),col=2) # 密度関数をカーネル法で推定しプロット
dev.copy2eps(file="geyser-duration-h.eps")
```

```
> source("run0024.R")
[1] "waiting" "duration"
[1] 299 2
      mean      sd meadian    iqr
72.31438 13.89032 76.00000 17.79123
      mean      sd meadian    iqr
3.460814 1.147904 4.000000 1.766768
```



- waiting も duration も二つ山があるが，その特徴を平均や標準偏差は表現できない．小さいほうの山を「ハズレ値」としては捨てられない．

- 単に平均や標準偏差だけでは不十分．密度関数を推定すれば，分布の形状もわかる．

### 3.2 最尤法による密度推定

- モデルをいろいろ変えて，最尤法を適用して分布を推定する．
- (1) 正規分布，(2)  $t$ -分布，(3) 2個の混合正規分布，(4) 3個の混合正規分布

```
# run0030.R
# 最尤推定量 (duration データ)

x <- geyser$duration # データ
x0 <- seq(min(x),max(x),length=300) # 密度関数を推定する範囲
drawx <- function(y,na,v=NULL) {
  truehist(x) # ヒストグラム
  rug(x) # 「ラグ」をプロット (下部にデータを示す線分)
  lines(x0,y,col=2)
  sapply(v,function(i) abline(v=i,col=3))
  dev.copy2eps(file=paste("run0030-",na,".eps",sep=""))
}

mydt <- function(x,mean,sd,df) # t-分布の密度関数
  dt((x-mean)/sd,df=df)/sd # 標準の dt は mean=0, sd=1 に相当

# 正規分布
m1 <- mean(x); s1 <- sqrt(sum( (x-mean(x))^2 )/length(x))
print(c(m1,s1))
drawx(dnorm(x0,mean=m1,sd=s1),"d1",m1)

# t-分布
deg <- 2; fit <- fitdistr(x,"t",df=deg)
m2 <- fit$estimate["m"]; s2 <- fit$estimate["s"]
print(c(m2,s2,deg))
drawx(mydt(x0,mean=m2,sd=s2,df=deg),"d2",m2)

# 2個の正規分布の混合分布
mydnorm2 <- function(x,th) # 密度関数
  th[5]*dnorm(x,mean=th[1],sd=th[2]) +
  (1-th[5])*dnorm(x,mean=th[3],sd=th[4])
lik <- function(th) sum(log(mydnorm2(x,th))) # 対数尤度
opt <- optim(c(2,1,5,1,0.3),lik,
            control=list(fnscale=-1)) # 数値的最適化
print(opt$par) # 最尤推定量の表示
```

```

drawx(mydnorm2(x0,opt$par),"d3",opt$par[c(1,3)])

# 3個の正規分布の混合分布
mydnorm3 <- function(x,th) # 密度関数
  th[7]*dnorm(x,mean=th[1],sd=th[2]) +
  th[8]*dnorm(x,mean=th[3],sd=th[4])+
  (1-th[7]-th[8])*dnorm(x,mean=th[5],sd=th[6])
lik <- function(th) sum(log(mydnorm3(x,th))) # 対数尤度
opt <- optim(c(2,1,3,1,4,1,0.3,0.3),lik,
            control=list(fnscale=-1)) # 数値的最適化
print(opt$par) # 最尤推定量の表示
drawx(mydnorm3(x0,opt$par),"d4",opt$par[c(1,3,5)])

```

```
> source("run0030.R")
```

```
[1] 3.460814 1.145982
```

```
      m      s
```

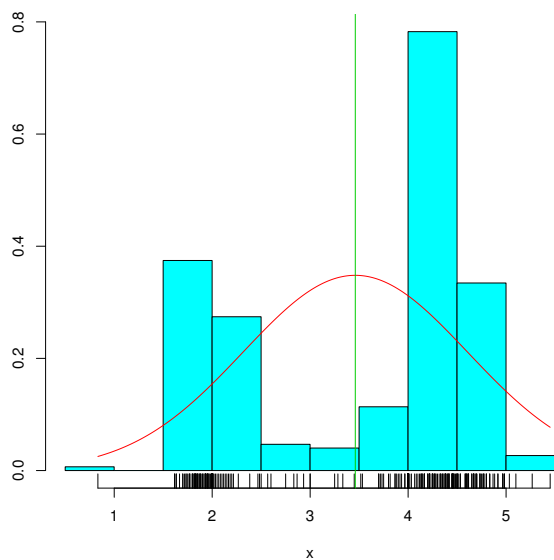
```
3.8493845 0.8806723 2.0000000
```

```
[1] 1.9569665 0.2447000 4.2188958 0.4074268 0.2978477
```

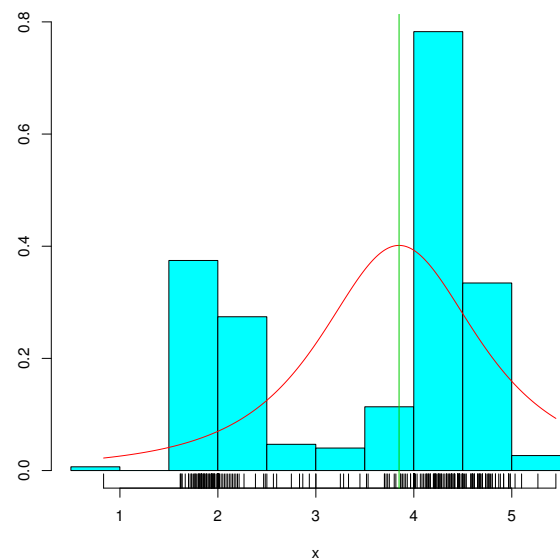
```
[1] 1.91654783 0.13097639 3.59800767 2.12497798 4.26608951 0.37341999 0.30965333
```

```
[8] 0.08658342
```

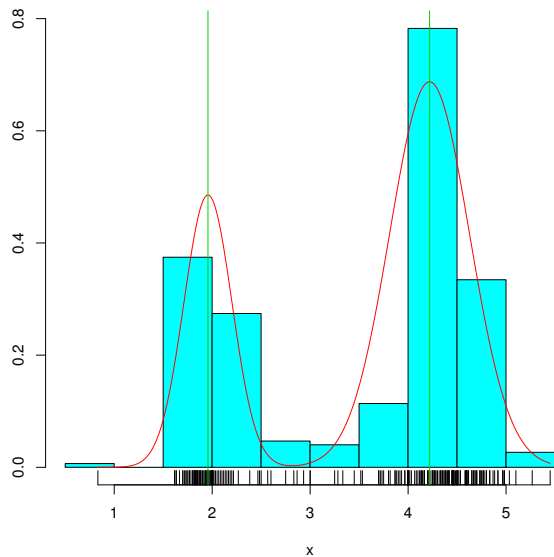
```
There were 26 warnings (use warnings() to see them)
```



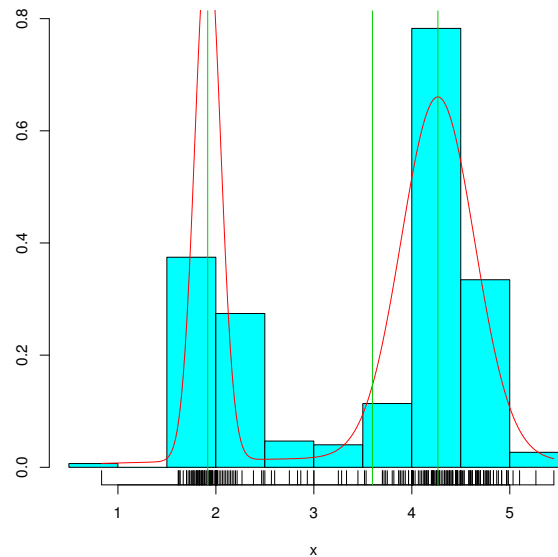
run0030-d1



run0030-d2



run0030-d3



run0030-d4

- 正規分布， $t$ -分布では，山が一つしかないので不自然。(単に平均や標準偏差だけでは，duration データがうまく表現できない.)
- 二つの正規分布を混合すれば，二つの山が表現できる．

$$f(x|\mu_1, \sigma_1, \mu_2, \sigma_2, p_1) = \frac{p_1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_i - \mu_1)^2}{2\sigma_1^2}\right) + \frac{1-p_1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(x_i - \mu_2)^2}{2\sigma_2^2}\right)$$

- 数値的最適化の問題 (不安定) . 山の数を推定する必要があることも問題 .

### 3.3 ヒストグラム

- ヒストグラムは密度推定的一种 .
- 分割数をいくつにするかが重要 . 分割数を減らすと細かい形状が見えなくなるし，分割数を増やすと (各 bin に入るデータ数が減るので) 推定精度が悪くなる .
- hist や truehist 関数では，あるアルゴリズムに従って自動的に分割数を決めている . hist では breaks オプション，truehist では nbins オプションで分割数を指定する . 自動決定のアルゴリズムも数種類用意されており，そのアルゴリズム名を文字列で指定することも可能 . デフォルトは breaks = "Sturges" と nbins = "Scott" .

```
# run0025.R
# duration のヒストグラム

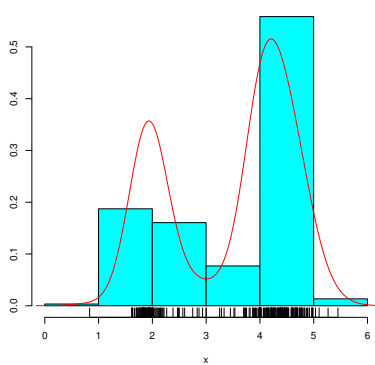
drawhist <- function(x,nb,name) {
  truehist(x,nbins=nb)
  rug(x)
  lines(density(x),col=2)
```

```

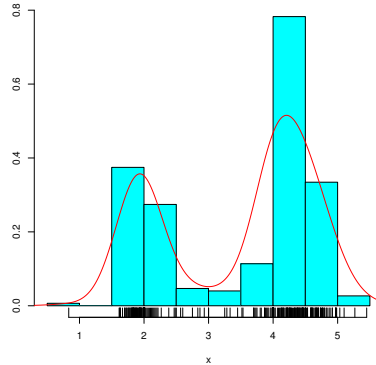
dev.copy2eps(file=paste(name,nb,".eps",sep=""))
}

bins <- c(5,10,20,40,80,1000) # 分割数
for(i in bins) drawhist(geyser$duration,i,"run0025-h")

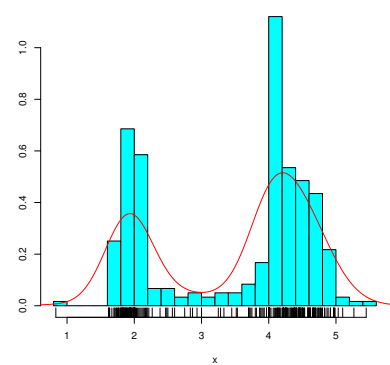
```



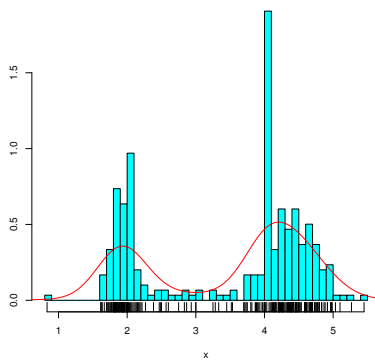
run0025-h5



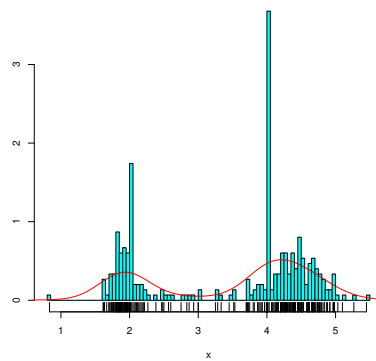
run0025-h10



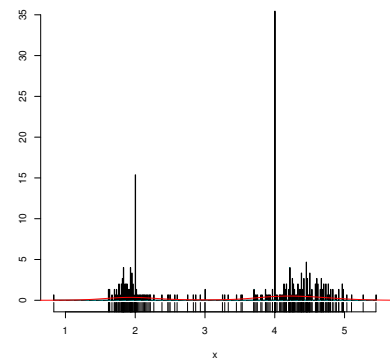
run0025-h20



run0025-h40



run0025-h80



run0025-h1000

### 3.4 カーネル密度推定

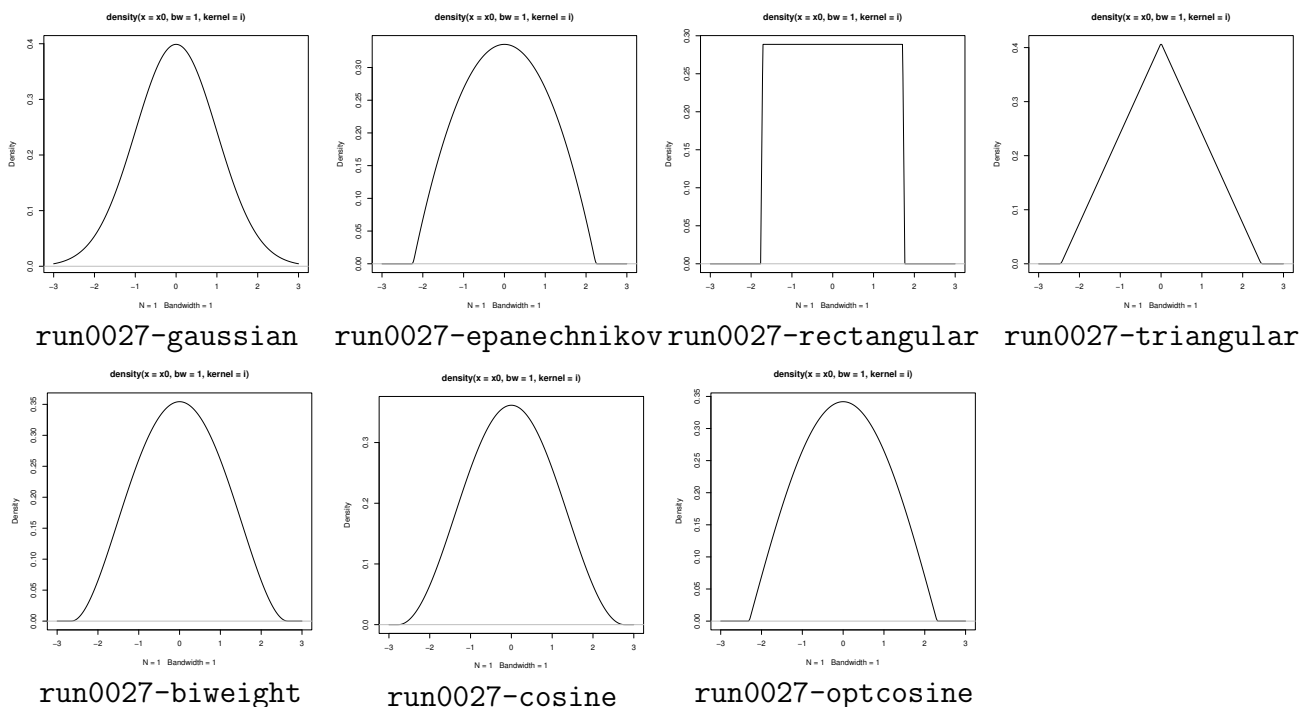
- カーネル関数  $K(x)$  を重み関数として、データをスムージングする。バンド幅パラメータ  $h_x$  は平均をとる範囲を指定する。

$$\hat{f}(x) = \frac{1}{nh_x} \sum_{i=1}^n K\left(\frac{x_i - x}{h_x}\right)$$

- $K(x)$  は原点を中心とする対称な関数で、分散が1に規格化されている。したがって、各  $x$  を中心として  $\pm 2h_x$  程度の幅の領域の  $x_i$  だけを取り出して平均を取ることに相当する。
- density 関数の実装されているカーネル関数： $K(x)$  のデフォルトは平均0, 分散1の正規分布の密度関数 (gaussian)。 $K(x)$  はすべて密度関数の一種であり、その標準偏差は1に規格化されている。以下に示す7種類が kernel オプションで指定できる。
- デフォルトでは  $h_x$  は自動的に調整されるアルゴリズムが呼び出されるがユーザが bw オプションで数値を指定することも可能。adjust=1 オプションで倍率を指定すれば、実際には bw\*adjust のバンド幅が使われる。

- 密度関数を推定してプロットするには、 $x$  を少しずつ変化させながら多数の点（たとえば 500 個）の値で  $\hat{f}(x)$  を評価するので、計算がすこし大変。
- このような計算を一般に「畳み込み」(convolution) という。畳み込みには高速フーリエ変換 (FFT) を利用した効率の良い計算法が知られており、これを density 関数では実装している。あらかじめ  $x_i$  を離散化しデフォルトで 512 個の代表点にまとめる。この離散化したデータの FFT と、カーネル関数の FFT を単純に掛け算して、それから逆 FFT を実行すると、512 個の代表点すべての  $\hat{f}(x)$  の値が一度に求まる。詳細は、`help(density)` 参照。また、`> density [enter]` とすれば、density オブジェクトの定義が見れるので、やっていることが大体想像できるだろう。

```
# run0027.R
# カーネル関数のプロット
x0 <- c(0) # 原点に1個だけ。これに density を適用すればよい。
krn <- c("gaussian", "epanechnikov", "rectangular", "triangular",
        "biweight", "cosine", "optcosine") # カーネル関数
for(i in krn) {
  plot(density(x0,bw=1,kernel=i))
  dev.copy2eps(file=paste("run0027-",i,".eps",sep=""))
}
```



```
# run0028.R
# カーネル関数とバンド幅を変更する

x <- geysers$duration

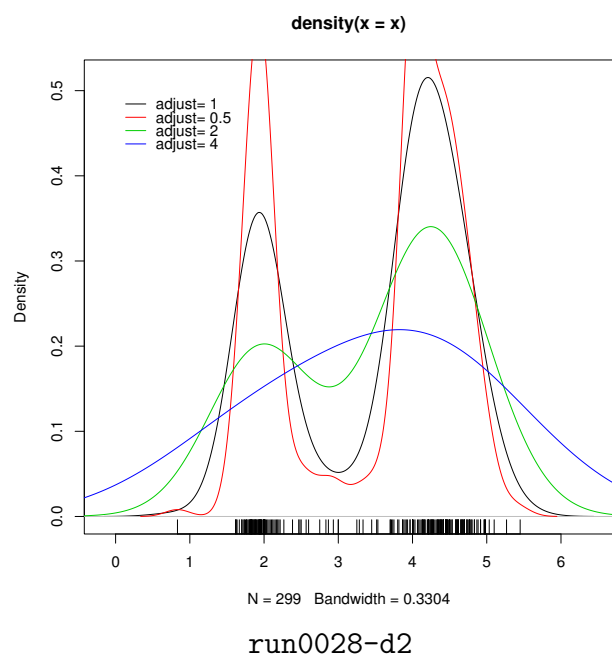
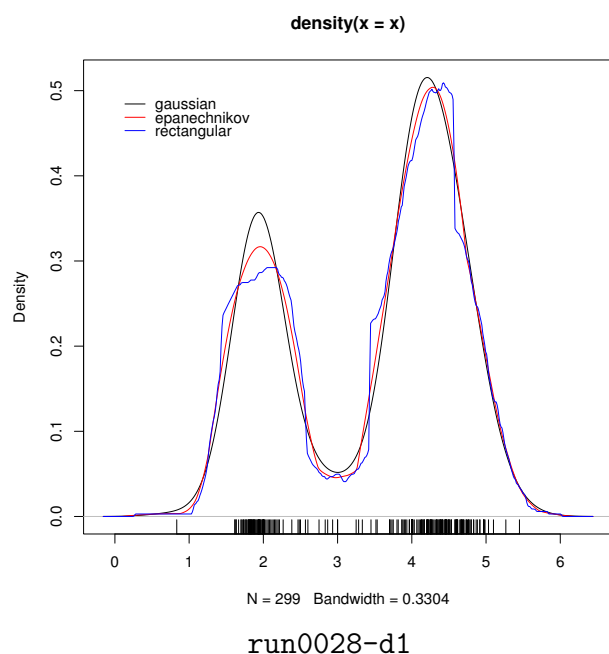
plot(density(x))
```

```

rug(x)
krn <- c("gaussian","epanechnikov","rectangular") # カーネル関数の変更
lines(density(x,kernel=krn[2]),col=2)
lines(density(x,kernel=krn[3]),col=4)
legend(0,0.5,krn,col=c(1,2,4),lty=1,bty="n")
dev.copy2eps(file="run0028-d1.eps")

plot(density(x))
rug(x)
adj=c(1,0.5,2,4) # バンド幅の倍率の変更
for(i in 2:4) lines(density(x,adjust=adj[i]),col=i)
legend(0,0.5,paste("adjust=",adj),col=1:4,lty=1,bty="n")
dev.copy2eps(file="run0028-d2.eps")

```



- カーネル関数を変えても，結果は大して変わらない．ただし， $K(x)$  の両端は連続に値が 0 になるものを使うべき．“rectangular” は連続でないので， $\hat{f}(x)$  がギザギザする．
- バンド幅の変更の影響は大きい．ヒストグラムの分割数の選択とまったく同じ問題である．

### 3.5 2次元の密度推定

- カーネル密度推定を2次元（多次元）に拡張するのは容易．
- データは  $(x_1, y_1), \dots, (x_n, y_n)$  とする．
- 点  $(x, y)$  における密度関数の推定は，

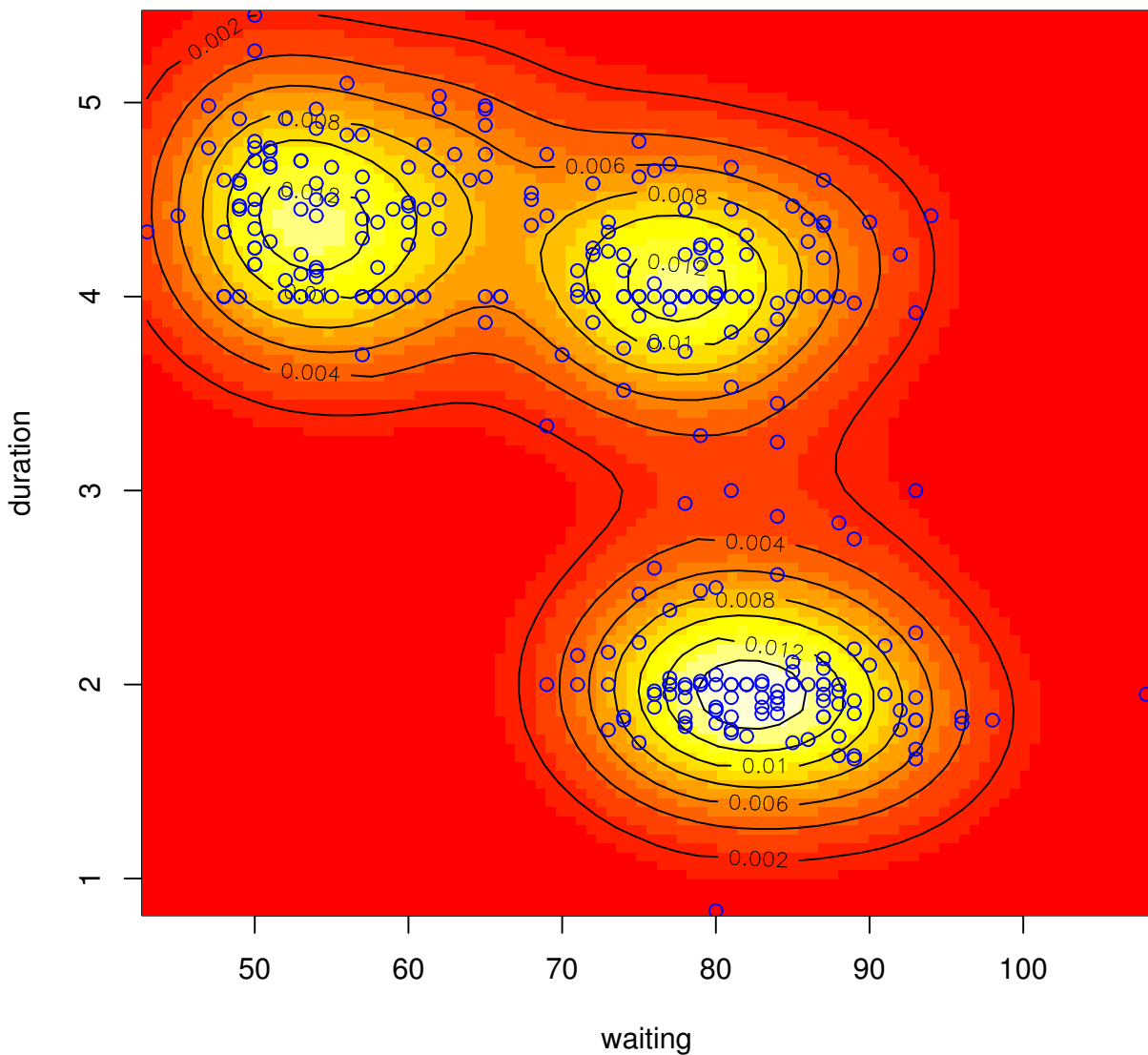
$$\hat{f}(x, y) = \frac{1}{nh_x h_y} \sum_{i=1}^n K\left(\frac{x_i - x}{h_x}\right) K\left(\frac{y_i - y}{h_y}\right)$$



- バンド幅  $h_x$  と  $h_y$  の指定に関しては，1次元のときと同様の問題がある．
- kde2d 関数が2次元版の密度推定である．

```
# run0029.R
# 2次元の密度推定 (help(kde2d) の例題を参照)

x <- geyser$waiting; y <- geyser$duration
f <- kde2d(x,y,n=100) # 100*100 の格子点で密度推定
image(f,xlab="waiting",ylab="duration") # 密度を色で表示
contour(f,add=T) # 等高線
points(x,y,col=4) # データ点
dev.copy2eps(file="run0029-d1.eps")
```



run0029-d1

## 4 課題

### 4.1 課題 3-1

データセット `dat0002` から変量を二つ選び、それぞれについて、以下を実行せよ。

- ヒストグラムの描画し、ラグを描き込む。
- カーネル密度推定を行い、上図に重ね描きする。
- 平均、メディアン、刈り込み平均 ( $\alpha = 0.1$ )、 $t$ -分布の最尤推定 ( $df = 5$ ) から得られる  $m$  を示し、それらの違いについてコメントする。
- 標本標準偏差、四分位偏差 (`run0021.R` で計算している `irq`)、MAD、 $t$ -分布の最尤推定 (自由度=5) から得られる  $s$  を補正したもの (`run0040.R` で計算しているように `tsdcorrection(5)` で割る) を示し、それらの違いについてコメントする。

### 4.2 課題 3-2

上で選んだ二変量について、`run0029.R` を参考にして 2 次元の密度推定を実行せよ。