# データフレーム読み込みのパラメータ

## ?read.table の出力

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
  dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
  row.names, col.names, as.is = !stringsAsFactors,
  na.strings = "NA", colClasses = NA, nrows = -1,
  skip = 0, check.names = TRUE, fill = !blank.lines.skip,
  strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "#", allowEscapes = FALSE,
  flush = FALSE,stringsAsFactors = default.stringsAsFactors(),
  fileEncoding = "", encoding = "unknown", text,
  skipNul = FALSE)
```

## 重要なパラメータ

header: FALSE | TRUE (**ヘッダあり**)
sep: ""(**スペース，タブ区切り**) | "," (CSV)
fileEncoding: Shift-JIS | UTF-8 (**既定値はプラットフォームによる**)

# １変量データの分布を調べる

## ファイルデータのようすを知る

いま， BMIdata.txt というファイルがあったとして，それがどのようになっているかをまず確かめることが必要。まず，ファイルを直接エディタやページャで見てみるのが素朴でよいやり方。
次に，読み込みのパラメータを適当に仮定してデータフレーム DT に読み込んで調べる。次はパラメータなし，つまりデフォルトのパラメータのみで読み込んでいる。

```
DT <- read.table("BMIdata.txt")
```

- DT を直接表示させる（たいていは長すぎて？？？ となる）
- head(DT) でヘッダと最初の数個のデータを調べる（よい方法）
- names(DT) で変量に与えられている名前を知る
- str(DT) でデータフレームの構造を知る
- 以上をもとに，最適なデータフレームとして読み込む

## パラメータなしで読み込んでチェックしていく

```
> DT <- read.table("BMIdata.txt")
> DT   # 打ち出してみる
...
21   Jirou   M  191.5   76.4
22     Tei   M  178.5   75.3
23    Yumi   F  155.6   54.3
24    Miki   F  164.2   63.2
25   Sacho   F  158.3   52.3
26 Taichi   M  171.4   84.4
27 Ichiro   M  191.5   76.4
28   Nobuo   M  178.5   75.3
```

**ターミナルいっぱいになってしまって，末尾しか見れない。これではようすがわからない。**

## head 関数でデータの冒頭を調べる

```
> head(DT)
      V1   V2     V3     V4
1   Name  Sex Height Weight
2   Yuri    F  155.6   54.3
3   Miwa    F  164.2   63.2
4   Saki    F  158.3   52.3
5 Taiki    M  171.4   84.4
6 Tarou    M  191.5   76.4
```

- **元々ヘッダ** (Name, Sex, ...) **があるのに，**V1〜V4 **というヘッダが付加されている**
  → header = TRUE **を使うべき**
- **セパレータはうまく機能しているので，スペース・タブ区切りになっている**

## names と str 関数でも調べる

```
> names(DT)
[1] "V1" "V2" "V3" "V4"
> str(DT)
'data.frame': 28 obs. of  4 variables:
 $ V1: Factor w/ 19 levels "Aki","Daiki",..: 9 19 8 12 14 15 5 1
 $ V2: Factor w/ 3 levels "F","M","Sex": 3 1 1 1 2 2 2 1 1 1 ...
 $ V3: Factor w/ 7 levels "155.6","158.3",..: 7 1 3 2 4 6 5 1 3
 $ V4: Factor w/ 7 levels "52.3","54.3",..: 7 2 3 1 6 5 4 2 3 1
>
```

- names **の結果**
  **データセットの名前がおかしい**
- str **の結果**
  $<$ **数値データの種類が** Factor **になっているのは，最初の** Height, Weight **がデータに取り込まれてしまったから**

## 正しいパラメータで読み込み

```
> DT <- read.table("BMIdata.txt",header=TRUE)
> names(DT)
[1] "Name"   "Sex"    "Height" "Weight"
> str(DT)
'data.frame': 27 obs. of  4 variables:
 $ Name  : Factor w/ 18 levels "Aki","Daiki",..: 18 8 11 13 14
 $ Sex   : Factor w/ 2 levels "F","M": 1 1 1 2 2 2 1 1 1 2 ...
 $ Height: num  156 164 158 171 192 ...
 $ Weight: num  54.3 63.2 52.3 84.4 76.4 75.3 54.3 63.2 52.3 ...
```

**上記の問題が解決されて，適切にデータが読み込まれている。特に** str **の結果に注目。**

- Factor **因子変数，カテゴリカル変数，名義変数**
- num **実数（整数と浮動小数点の区別はない）**

## データファイルのエンコーディングの問題

CSV ファイルを表計算ソフトからテキストデータにエクスポートした場合

- MS-Office Excel から — Shift-JIS
- LibreOffice Calc から — UTF-8

プラットフォームによる日本語のエンコーディングの既定値

- Windows 版 — Shift-JIS
- Mac/Linux 版 — UTF-8

読み込むテキストのエンコーディングを誤った時のエラーメッセージ（一部省略）

```
# UTF-8 のテキストを Shift-JIS として読み込んだとき
   警告メッセージ:
   入力コネクション 'Score.csv' に不正な入力がありました
   incomplete final line found by readTableHeader on 'Score.csv'
# Shift-JIS のテキストを UTF-8 として読み込んだとき
   '<f1>' に不正なマルチバイト文字があります
   追加情報:  警告メッセージ:
   入力コネクション 'Score2.csv' に不正な入力がありました
   incomplete final line found by readTableHeader on 'Score2.csv'
```

# 1 変量データの分布を調べる

**代表的な確率分布**

**一様分布** ある区間内のどこも等しい確率で発生する現象の分布（落下する雨粒の位置の分布）

**正規分布** 多数のランダムデータの重なりで生じる普遍的な分布

**ポアソン分布** 単発のバラバラな現象によって生じる分布（事故の発生数）

**t-分布（スチューデント分布）** 正規分布に従う少数のランダムデータの重なりで生じる分布（数個〜10 個のサンプルサイズ）

与えられたデータがどの分布に従っているかを知ることは，解析の最初のステップとして重要。

**分布を確認する２つの方法**

**統計的検定** ある分布を仮定して，それに合致するかどうかを検定する

**可視化による検証** データの分布を可視化することによって，直感的に判断する

# データの分布は正規分布か？

## データファイル読み込み

**ファイル** DataA.csv **のデータが正規分布しているかどうかを知りたい。**

```
> DA <- read.table("DataA.csv",sep=",",header=T)
> head(DA)
  X32.25      # 最初の行のデータをヘッダとみなしている
1  34.27
2  34.41
...
> DA <- read.table("DataA.csv",sep=",",header=F) # やり直し
> head(DA)
     V1       # 今度はオーケー
1 32.25
2 34.27
```

DA **というデータフレームは** V1 **という名前のベクターデータ１つだけをもつ。**

## Shapiro-Wilk 検定で正規分布から逸脱していないか調べる

**ベクター** DT$V1 **に対してシャピロ—ウィルク検定を実行する。**

```
> summary(DA$V1)    # まずデータの概略を知っておく
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  32.25   49.39   55.78   55.59   62.19   79.60
> shapiro.test(DA$V1)

Shapiro-Wilk normality test
data:  DA$V1
W = 0.99452, p-value = 0.6782
```

## 結果の意味

W: **検定統計量** p: p **値**
W **は確率変数で，データの正規性を反映する分布に従う。**p **は** W **より外側（分布の端）の面積。**
**上の結果では** p **が小さくないので，データが正規分布しているという帰無仮説は棄却できない。つまり正規分布していることを否定できない。→とりあえず正規分布でやれそう。**

## lattice パッケージによるグラフの描画

R には基本的な統計処理と描画の機能が備わっているが，この世界には膨大な計算手法や多彩なビジュアル表現が存在するので，機能を追加するために多数のパッケージが提供されている。以下では，データの分布の正規性をチェックするために，高度なグラフィックス機能を備えた lattice ライブラリを利用する。
新規にライブラリをインストールするには，次のようにコンソールから入力する。この作業は一度やっておけばよい。

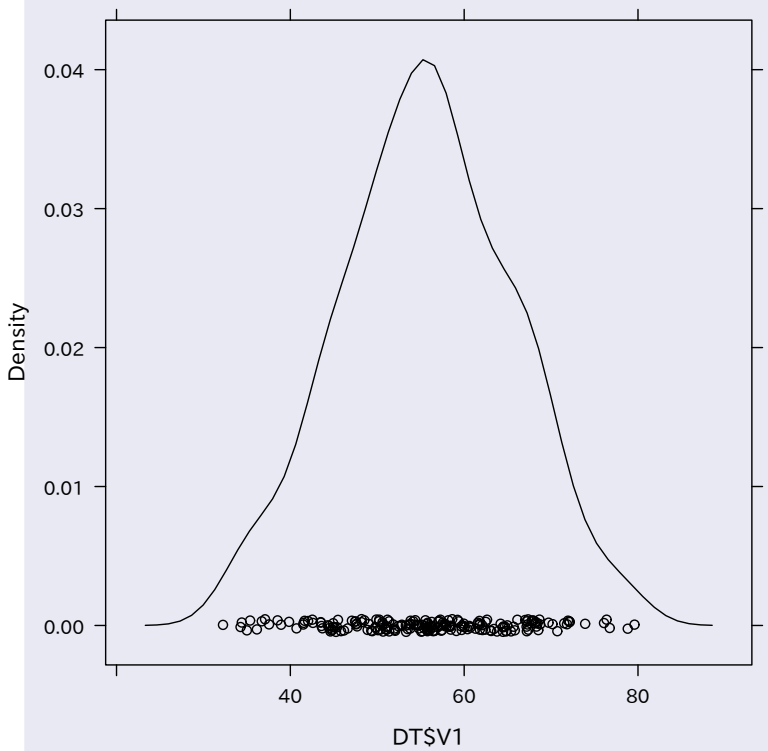
```
> install.packages("lattice")
Installing package into ...
# この後，ライブラリをどこのミラーサーバからダウンロードするよう
# に尋ねられるので，日本のサーバを指定すると。処理が行われる。
```

## ちょっとした注意

install.packages() の引数（ライブラリ名）にはダブルクオーテーションをつけるが，後述の library() では library(lattice) のように引用符は付けない。

# 可視化による正規性のチェック (1)

```
> summary(DA$V1)    # 事前に分布の大略をつかんでおく
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  32.25   49.39   55.78   55.59   62.19   79.60
> library(lattice) # lattice パッケージを読み込む
> densityplot(DA$V1) # データから予想される確率密度をプロット
```

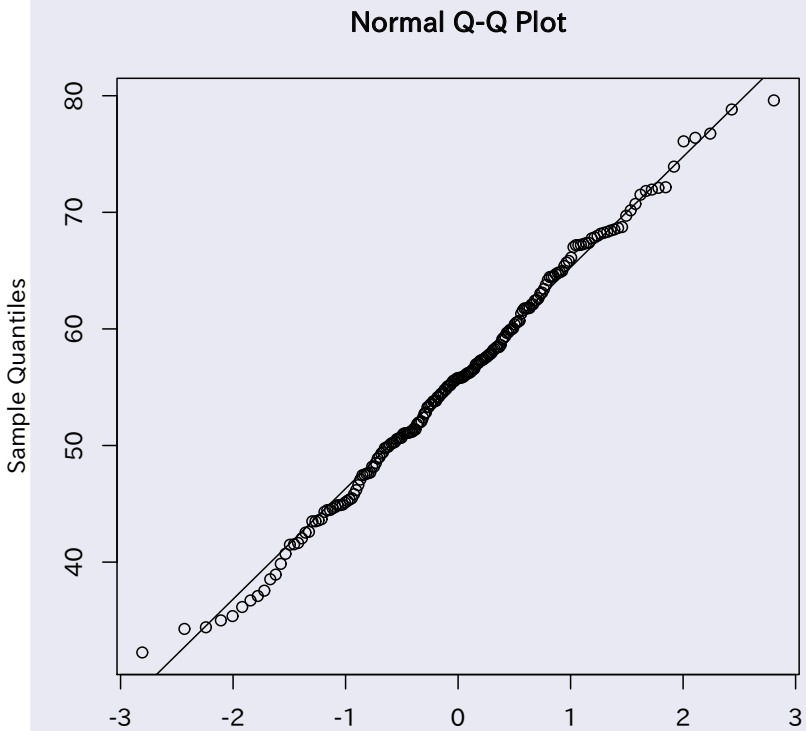



出力される図。まあいいかな。

## 可視化による正規性のチェック (2)

**Q-Q プロット：横軸に正規分布から得られるパーセンタイル，縦軸にデータのパーセンタイルを取る。**
**注意：qqnorm 関数は通常のプロット機能でサポートされていて，lattice ライブラリは不要。**

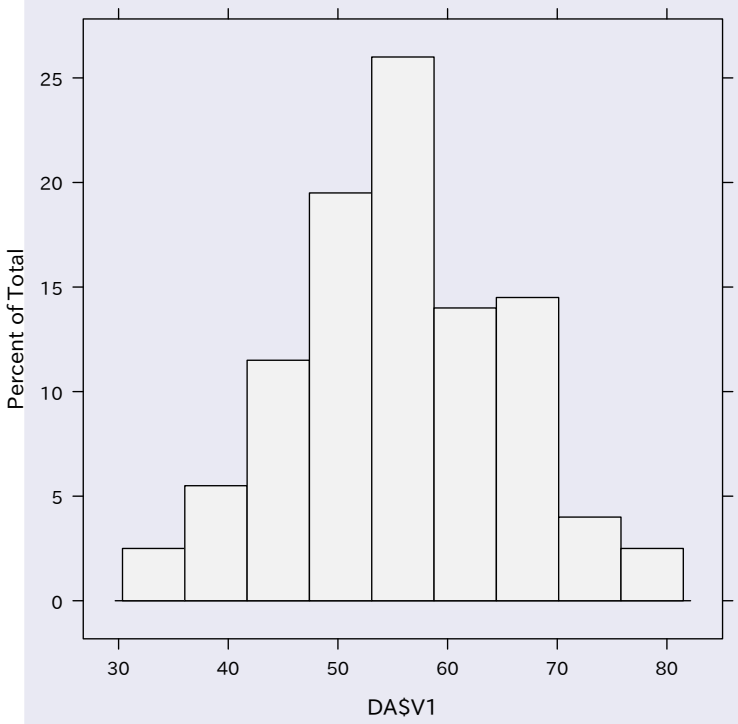
```
> qqnorm(DA$V1)    # Q-Q プロットを描く
> qqline(DA$V1)    # 縦軸にも正規分布を使って得られる直線を重ね描き
```

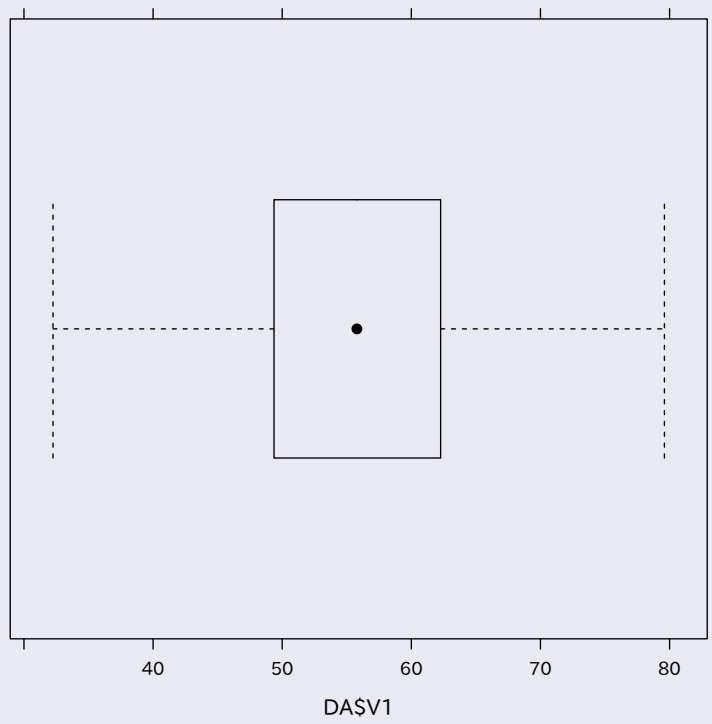



**Q-Q プロットもよい直線性を示している。**

## 分布を可視化する他のプロット

```
> histogram(DA$V1)   # ヒストグラム
> bwplot(DA$V1)      # 箱ひげ図 Box-Whisker plot, Box Plot
```




**ヒストグラムの区間と階級幅は自動的に設定される。オプションで変更可能。通常の箱ひげ図は五数要約を使って描かれる。**

## 複数のデータを描画して比較する

**２つのデータセットの分布を比較したいことはよくある。**
**いま，２組のデータがヘッダ付きで書き込まれているタブ区切りファイル** DataAB.txt **があり，冒頭の数行は次のように** A, B **２タイプのデータの値が混在している。このフォーマットのデータは** lattice **ライブラリで扱いやすい。**

```
X       Type
43.69   A
42.04   A
39.85   A
67.98   B
58.00   B
```

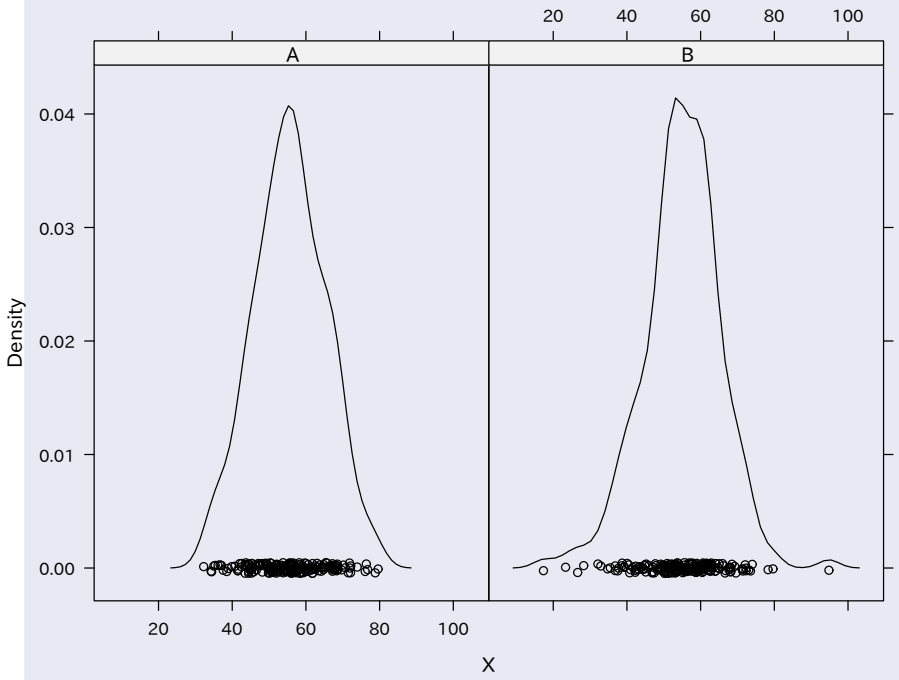
## データフレームに読み込んでようすをみる

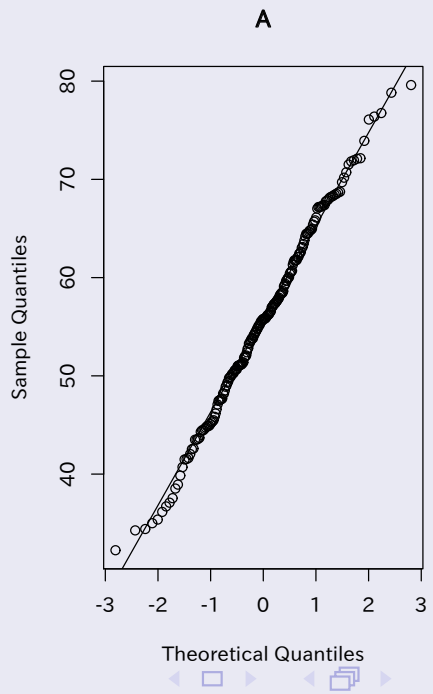
```
> DT <- read.table("DataAB.txt",header=T)

> str(DT)
'data.frame': 400 obs. of  2 variables:
 $ X    : num  43.7 42 39.9 68 58 ...
 $ Type: Factor w/ 2 levels "A","B": 1 1 1 2 2 2 1 2 2 1 ...
```

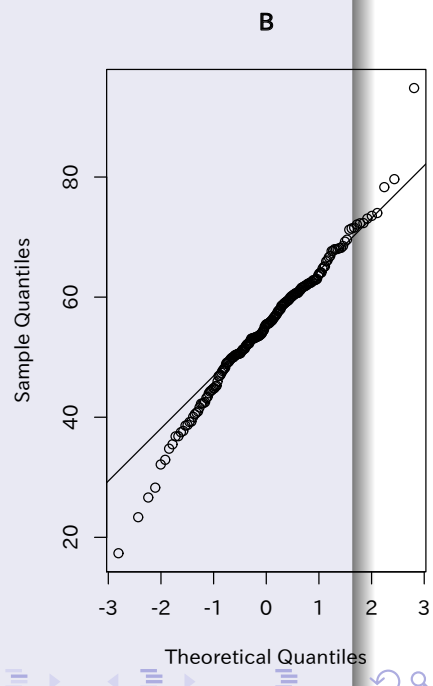
## 2 つの密度関数を並べて描画

densityplot 関数は lattice ライブラリに含まれていて，複数のファクター (カテゴリカル変数) をもつデータは自動的に別のグラフになる。
次の例では Type が A, B からなるので，2 つのグラフになる。
|の左の ~ X は，チルダの後ろの X を独立変数として密度を描く意味。

```
> library(lattice)
> densityplot(~ X|Type, data=DT) # 密度関数の描画
> densityplot(~ DT$X|DT$Type)    # これでも構わない
```



## Q-Q プロット (lattice ライブラリ不要)

前ページ右の Q-Q プロットを描く qqnorm 関数は基本描画機能でサポートされていて，自動的に 2 枚の図面を作らない。そのため par() 関数で描画のフレーム 2 枚を設定してから，別々に qqnorm() を使って描画する。
このとき，図のタイトルは自動的に付かないので，qqnorm() の引数で設定する。

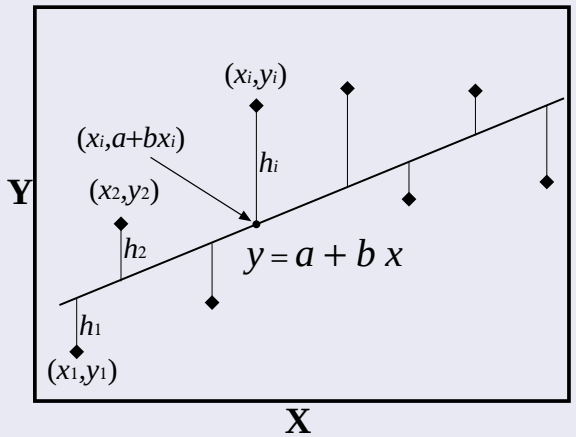
```
> par(mfrow=c(1,2))
> XA <- DT$X[DT$Type=="A"] # Type が A の要素だけを選択
> XB <- DT$X[DT$Type=="B"] # Type が A の要素だけを選択
> qqnorm(XA,main="A")      # メインのタイトルを "A"に
> qqline(XA)
> qqnorm(XB,main="B")      # メインのタイトルを "B"に
> qqline(XB)
```

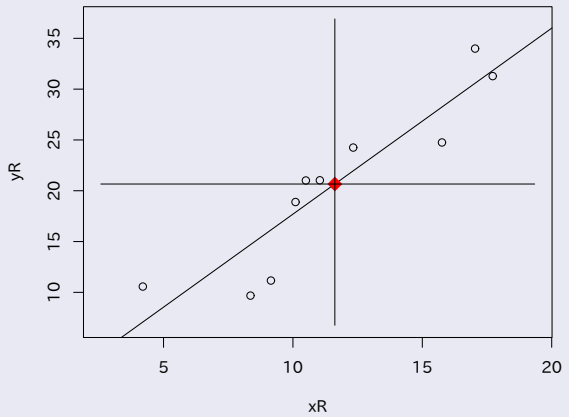
## lattice を使えば qqmath() で

なお，lattice でサポートしている類似の関数として qqmath() があり，これを使えば前ページの densityplot() と同じように描画できる。

# 多変量データの相関を調べる

## 2 変数の最小二乗法の原理




**$h_i$ の２乗和が最小になるように，微分を使って $a, b$ を決めてやる（$ax + b$ の形が多いが，ここでは逆にしてある）．ここで，$s_x^2$ は $x_1, x_2, \ldots, x_n$ の分散，$s_{xy}$ は $x$, $y$ の共分散。**

$$b = \frac{s_{xy}}{s_x^2}, \quad a = \bar{y} - b\bar{x}$$

## 重回帰分析の数学的原理

**モデル式**

$$y = a + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$$

**次のようにして，分散と共分散を使って係数 $b_1, b_2, \ldots$ を求めることができる．**

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = \begin{bmatrix} s_{x_1 x_1} & s_{x_1 x_2} & \cdots & s_{x_1 x_p} \\ s_{x_2 x_1} & s_{x_2 x_2} & \cdots & s_{x_2 x_p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{x_p x_1} & s_{x_p x_2} & \cdots & s_{x_p x_p} \end{bmatrix}^{-1} = \begin{bmatrix} s_{x_1 y} \\ s_{x_2 y} \\ \vdots \\ s_{x_p y} \end{bmatrix}$$

## R による単回帰分析

**二次元データについて分析を試みる．**

| X | Y |
|---|---|
| 11.04 | 21.03 |
| 15.76 | 24.75 |
| 17.72 | 31.28 |
| 9.15 | 11.16 |
| 10.1 | 18.89 |
| 12.33 | 24.25 |
| 4.2 | 10.57 |
| 17.04 | 33.99 |
| 10.5 | 21.01 |
| 8.36 | 9.68 |

```
DT <- read.table("xy10.dat",header=TRUE)
result = lm(Y ~ X, data = DT) # 線形回帰を実行
summary(result) # 結果の数値を出力
plot(Y ~ X, data = DT) # 散布図出力
abline(result)   # 回帰直線を出力
```

## 処理の意味

```
result = lm(Y ~ X, data = DT)
```

**線形モデル** (linear model) **への回帰の計算を実行し，結果を** `result` **というオブジェクトに代入．**
`lm` **の引数，** `Y ~ X, data = DT` **は，** `DT` **オブジェクトの** `X` **を説明変数，** `Y` **を目的変数とする意味．**

```
summary(result)
```

`result` **の内容，つまり線形回帰の結果のデータを出力．**

```
plot(Y ~ X, data = DT)
```

**データフレーム** `DT` **の** `X` **に対して** `Y` **をプロットして散布図を描く．**

```
abline(result)
```

**線形回帰によって得られた直線を引く．**

## 結果の出力

```
Call:
lm(formula = Y ~ X, data = DT)
Residuals:
   Min     1Q Median     3Q    Max
-5.014 -2.754  1.221  2.372  3.491
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.6092     3.4405  -0.177 0.863859
X             1.8305     0.2800   6.538 0.000181 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’
0.1 ‘ ’ 1
Residual standard error: 3.54 on 8 degrees of freedom
Multiple R-squared:  0.8424,Adjusted R-squared:  0.8227
F-statistic: 42.75 on 1 and 8 DF,  p-value: 0.000180
```

## 結果を読む

Residuals: **残差の五数要約**

Coefficients: **回帰直線の情報：**

Intercept **切片**，X **偏回帰係数（傾き）**
Estimate **予測値，**Std. Error **標準誤差，**t value t **値，**Pr(>|t|)，p **値**

(**切片，予測値，**p **値が重要**)

Residual standard error: **残差標準誤差（自由度）**

Multiple R-squared: (**重**) **相関係数の** 2 **乗**

Adjusted R-squared: (**重**) **相関係数の** 2 **乗** (**自由度調整済**)

F-Statistic: **回帰の分散分析の** F **値，自由度，**p **値**

## 重回帰分析

**次のデータが** TestScore.txt **に収められている。**

| Eng | Math | Sci | Art |
|---|---|---|---|
| 84 | 58 | 87 | 47 |
| 84 | 59 | 89 | 54 |
| 86 | 59 | 90 | 50 |
| 87 | 63 | 94 | 55 |
| 83 | 60 | 88 | 51 |
| | | | |
| 83 | 60 | 88 | 50 |
| 84 | 60 | 90 | 54 |
| 82 | 60 | 86 | 50 |
| 82 | 60 | 88 | 52 |
| 85 | 63 | 90 | 53 |

**全** 30 **件**

**説明変数：**Eng, Math, Sci
**目的変数：**Art

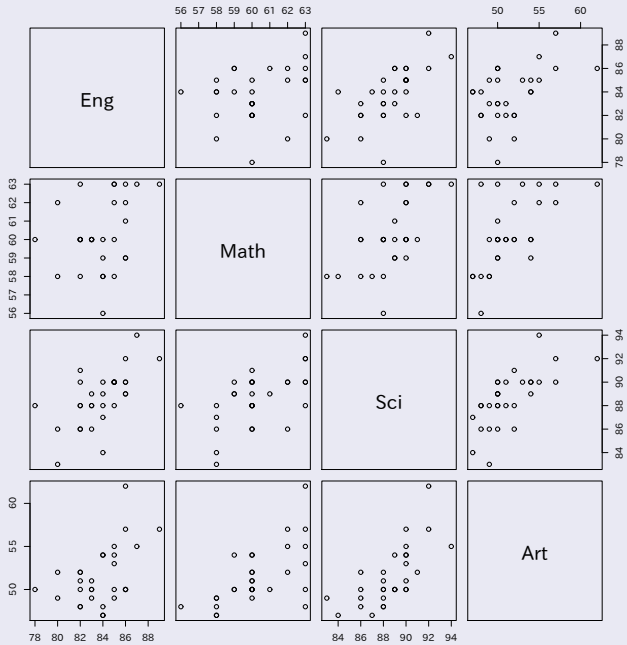
## データ間の相関行列をチェック

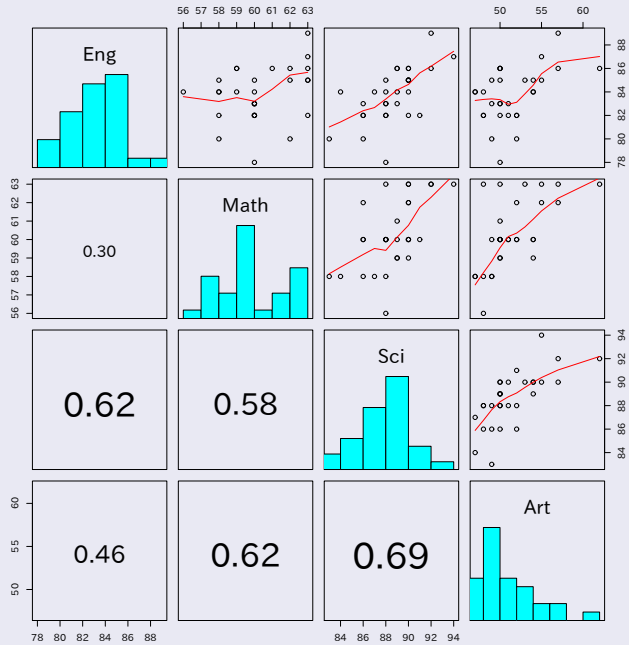
```
Result <- read.table("TextScore.txt",header=T)
cor(Result) ## データ間の相関係数を計算して表示
```

```
             Eng       Math        Sci     Art
Eng    1.0000000 0.3000223 0.6240064 0.4580098
Math   0.3000223 1.0000000 0.5753726 0.6212204
Sci    0.6240064 0.5753726 1.0000000 0.6888909
Art 0.4580098 0.6212204 0.6888909 1.0000000
```

**一般に，説明変数同士に強い相関があることは好ましくない．ここではたかだか 0.6 程度なので問題なし．**

# グラフィカルに確認する




**前例の `cor(Result)` の代わりに，`pairs(Result)` とすると，左のように散布図が表示されて，全体の傾向をつかめる．右のようにさらに細かい情報を表示させることもできる（説明省略）。**

## 結果を出力する

```
Result.fit <- lm(Art ~ Eng + Math + Sci, data = Result) # 説明変数を + でつなぐ
summary(Result.fit)
```

**結果：**

```
Residuals:
    Min      1Q  Median      3Q     Max
-4.5320 -1.1779 -0.3508  1.4179  6.4489
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -51.8052    19.2769  -2.687   0.0124 *
Eng           0.1165     0.2475   0.471   0.6418
Math          0.6130     0.2873   2.133   0.0425 *
Sci           0.6383     0.2835   2.251   0.0330 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
1
Residual standard error: 2.385 on 26 degrees of freedom
Multiple R-squared:  0.554,Adjusted R-squared:  0.5025
F-statistic: 10.76 on 3 and 26 DF,  p-value: 8.859e-05
```

Math, Sci **の相関はあるが，**Eng **の相関はない．**

# クラスター分析

## 「距離」の近いデータを結ぶ

多数の「もの」に対して数量的なデータを元にした分類を行うことは有益な情報処理の手段である．
ものとものの間の「距離」を使うことによって，近いもの＝類縁という関係を抽出して，グルーピングを行う手法がクラスター分析である．結果を階層的なツリー構造で表せる階層的クラスター分析と，非階層的クラスター分析とがある．
ここでは視覚的にわかりやすい階層的クラスター分析を紹介する．

クラスター分析では，データの間の「近さ」を定義して，
それによって近親関係をまとめていく．

## 各年齢別の平均余命の国別比較

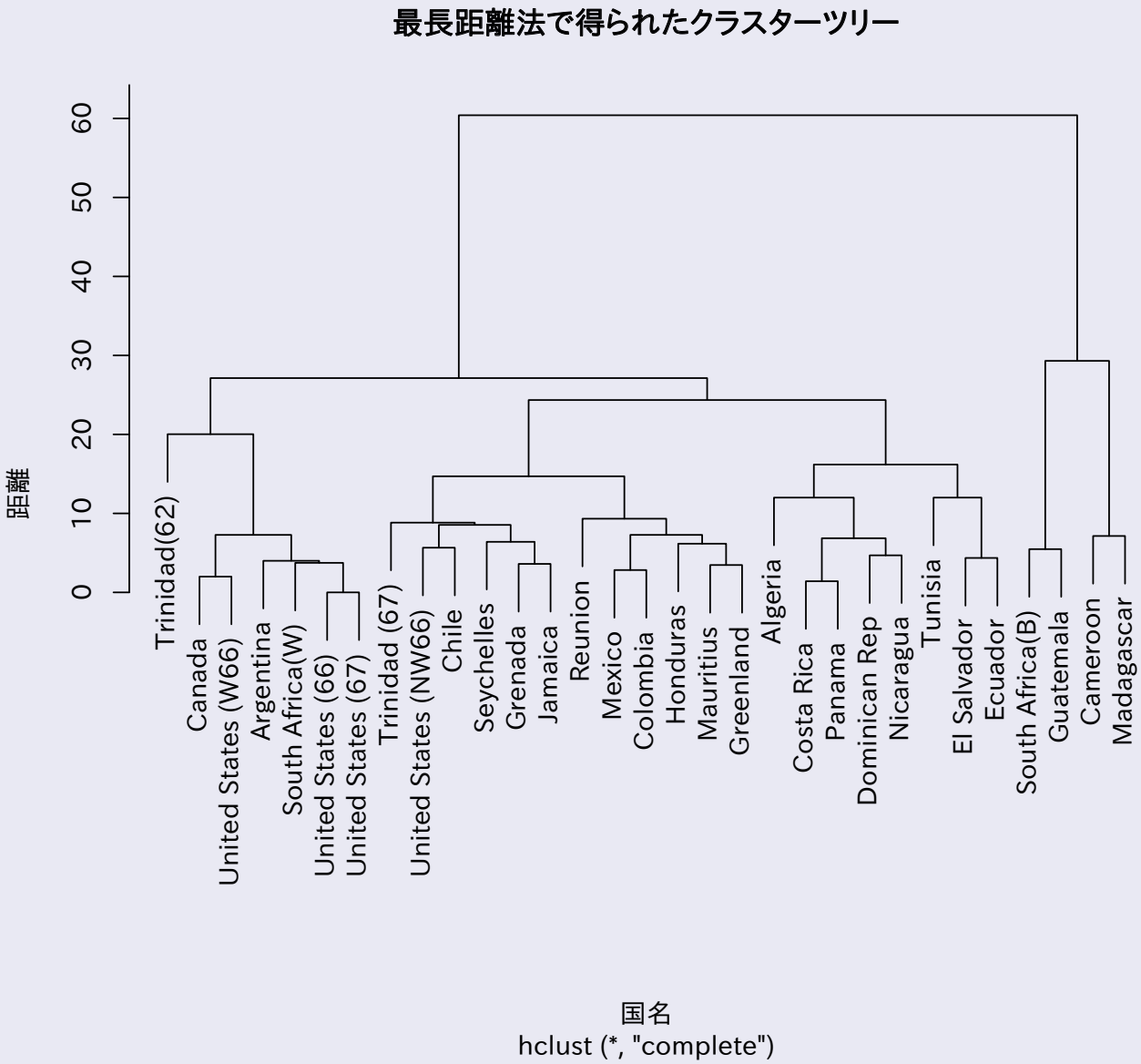
| | m0 | m25 | m50 | m75 | w0 | w25 | w50 | w75 |
|---|---|---|---|---|---|---|---|---|
| Algeria | 63 | 51 | 30 | 13 | 67 | 54 | 34 | 15 |
| Cameroon | 34 | 29 | 13 | 5 | 38 | 32 | 17 | 6 |
| Madagascar | 38 | 30 | 17 | 7 | 38 | 34 | 20 | 7 |
| Mauritius | 59 | 42 | 20 | 6 | 64 | 46 | 25 | 8 |
| Reunion | 56 | 38 | 18 | 7 | 62 | 46 | 25 | 10 |
| Seychelles | 62 | 44 | 24 | 7 | 69 | 50 | 28 | 14 |
| South Africa(B) | 50 | 39 | 20 | 7 | 55 | 43 | 23 | 8 |
| South Africa(W) | 65 | 44 | 22 | 7 | 72 | 50 | 27 | 9 |
| Tunisia | 56 | 46 | 24 | 11 | 63 | 54 | 33 | 19 |
| Canada | 69 | 47 | 24 | 8 | 75 | 53 | 29 | 10 |
| Costa Rica | 65 | 48 | 26 | 9 | 68 | 50 | 27 | 10 |
| Dominican Rep | 64 | 50 | 28 | 11 | 66 | 51 | 29 | 11 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| Ecuador | 57 | 46 | 28 | 9 | 60 | 49 | 28 | 11 |

**全 31 ヵ国．** mxx, wxx **は** xx **歳の男性 (女性) の平均余命**

## R による処理

```
## データフレームに読みこむ
life <- read.table("LifeExp.txt",header=T)
## 行の名前（表の左端）を country に代入
country <- row.names(life)
## 各国のデータの間のユークリッド距離を計算
dist <- dist(life)
postscript("LifeExp.ps",horizontal=F, width=7,
           height=7,onefile=TRUE)
## 最長距離法でクラスター分析
plot(hclust(dist, method = "complete"),
     labels = country, # データごとのラベルを国名にする
     xlab = "国名", ylab = "距離", main = "最長距離法")
```

# 得られた結果


最長距離法で得られたクラスターツリー
距離
0
10
20
30
40
50
60
Trinidad(62)
Canada
United States (W66)
Argentina
South Africa(W)
United States (66)
United States (67)
Trinidad (67)
United States (NW66)
Chile
Seychelles
Grenada
Jamaica
Reunion
Mexico
Colombia
Honduras
Mauritius
Greenland
Algeria
Costa Rica
Panama
Dominican Rep
Nicaragua
Tunisia
El Salvador
Ecuador
South Africa(B)
Guatemala
Cameroon
Madagascar
国名
hclust (*, "complete")

# 判別分析

## どちらの群れに属するか？

**下の** TibetScull.txt **というファイルがある．これはチベットの２箇所の遺跡で発掘された頭蓋骨のデータである．**Type **のカラムで示されているように，これらは２群に分類されている．**

| | Length | Breadth | Height | Fheight | Fbreadth | Type |
|---|---|---|---|---|---|---|
| "1" | 190.5 | 152.5 | 145 | 73.5 | 136.5 | "1" |
| "2" | 172.5 | 132 | 125.5 | 63 | 121 | "1" |
| "3" | 167 | 130 | 125.5 | 69.5 | 119.5 | "1" |
| "4" | 169.5 | 150.5 | 133.5 | 64.5 | 128 | "1" |
| "5" | 175 | 138.5 | 126 | 77.5 | 135.5 | "1" |
| ... | ... | ... | ... | ... | ... | ... |
| "19" | 179.5 | 135 | 128.5 | 74 | 132 | "2" |
| "20" | 191 | 140.5 | 140.5 | 72.5 | 131.5 | "2" |
| "21" | 184.5 | 141.5 | 134.5 | 76.5 | 141.5 | "2" |
| ... | ... | ... | ... | ... | ... | ... |
| "31" | 197 | 131.5 | 135 | 80.5 | 139 | "2" |
| "32" | 182.5 | 131 | 135 | 68.5 | 136 | "2" |

新しく発掘された頭蓋骨２個がある．どちらに属するかを知りたい．

| | Length | Breadth | Height | Fheight | Fbreadth |
|---|---|---|---|---|---|
| A | 171.0 | 140.5 | 127.0 | 69.5 | 137.0 |
| B | 179.0 | 132.0 | 140.0 | 72.0 | 138.5 |

次のスクリプトで判別分析がなされる．

```
library(MASS)  # 多変量解析のライブラリ MASS をロード
DT <- read.table("TibetScull.txt",header=T)
dis <- lda(Type ~ Length + Breadth + Height +
       Fheight + Fbreadth, data = DT,
       prior = c(0.5,0.5))  # 判別の基準を設定
## 新しい頭蓋骨２個のデータを入力．Type がないことに注意
newscull <-  read.table("NewScull.txt",header=T)
predict(dis, newdata = newscull) # 予測を行う
```

## 結果

**結果を見ると，** A, B **が タイプ** 1 **である確率はそれぞれ** 0.755, 0.174 **となっている．**

```
$class
[1] 1 2
Levels: 1 2

$posterior
           1         2
A 0.7545066 0.2454934
B 0.1741016 0.8258984
```

# 分割表（クロス集計表）から独立性の検定

## 目的

**次のようなアンケート集計結果があったとしよう (ダミーデータで根拠ありません)．ファイル名は** Musicchoice.txt **とする．**

| | 赤 | 青 | 緑 |
|---|---|---|---|
| **クラシック** | 39 | 45 | 21 |
| **邦楽ポップス** | 83 | 68 | 47 |
| **洋楽ポップス** | 53 | 51 | 65 |
| **歌謡曲** | 41 | 32 | 55 |

**次の２行で，独立性の** $\chi^2$ **検定ができる．**

```
MData <- read.table("MusicChoice.txt",header=T)
## ファイル読み込んでデータフレーム MData に代入
## カイ二乗検定を行う
chisq.test(MData)
```

## 結果

**結果は次の通り．**

```
Pearson's Chi-squared test
data:  MData
X-squared = 25.8888, df = 6, p-value = 0.0002335
```

**結果を解釈すると次のようになる．**

> **音楽のジャンルと色の好みが独立であるという仮説は，$p = 0.00024$ となり，棄却される．つまり関係がある．**

**$\chi^2$ 値の 25.9 と下表の自由度 $(\mathrm{df}) = 6$ におけるカイ二乗分布のパーセント点を比較して，「危険率 0.5%で棄却される」としてもよい．**

# 参考書

- **舟尾暢男**『The R Tips **第 2 版 データ解析環境 R の基本技・グラフィックス活用集**』**オーム社**, 2009 (**電子版あり**)
- **青木繁伸**『R **による統計解析**』**オーム社**, 2009 (**電子版あり**)
- A. **ジュール他**『R **初心者のための** ABC』**丸善** * ,2012
- B. **エヴェリット**『R **と** S-PLUS **による多変量解析**』**丸善** *,2012
- **シリーズ**『R **で学ぶデータサイエンス**』**金明哲編，共立出版** 2010〜
- 『RjpWiki』 http://www.okadajp.org/RWiki/ **オープンソースの統計解析システムである** R **に関する情報交換を目的とした** Wiki
- "The R Project for Statistical Computing", https://www.r-project.org/ R **の開発とドキュメントに関する本家サイト。世界中に多数ミラーされている。**

* **丸善はシュプリンガー・ジャパンから版権を譲り受けて，黄色い**