R入門

基礎生物学研究所
ゲノムインフォマティクストレーニングコース
内山　郁夫　**(uchiyama@nibb.ac.jp)**

Rとは

- ベル研究所で開発された統計処理言語「S」を基に、フリーソフトとして開発された**統計解析環境・プログラミング言語**
- **コマンドラインインターフェイス**が基本。対話的なコマンド実行による解析のほか、プログラム（スクリプト）を書いて一括処理を行うことも可能
- **ベクトル・行列演算**が簡便かつ効率的に行える
- 独自の**関数**を作成することによって機能拡張が可能
- 作成した関数等を**パッケージ**単位でまとめることにより、機能拡張が容易に行える。様々なパッケージが公開されており、これらを導入することによって、最先端の統計手法を用いることができる。

スカラー演算

```
> 1+3
[1] 4
> 1+3*5
[1] 16
> a <- 1+3*5
> a
[1] 16
> a > 10
[1] TRUE
```

通常の通り、*（かけ算）は+（足し算）より優先する

結果を変数 a に代入する

a とだけタイプすると、変数aの内容が出力される

論理演算は、論理値TRUEまたはFALSEを返す

変数と代入

- 計算結果を変数に代入することにより、再利用できる
  - 変数名にはアルファベット、数字、'.'（dot）、'_' (underscore)が利用できる。ただし，先頭はアルファベット。
  - 大文字と小文字は区別される。
- 代入を表す演算子には、<- = -> の3通りがある。

  以下の3つはいずれも a に 4 が代入される。

```
> a <- 1 + 3
> a = 1 + 3
> 1 + 3 -> a
```

ヒストリー（履歴）

- コンソール上で、上下の矢印キー（↑↓）によて、コマンドの履歴を前後にたどることができる。
- 左右の矢印キー（←→）によって、カーソルを左右に動かせる。これによってコマンドの編集ができる。
- 以下のコントロールキーを使った操作も可能
  - Control+P 履歴を前に移動（↑と同じ）
  - Control+N 履歴を後ろに移動（↓と同じ）
  - Control+B カーソルを左に移動（←と同じ）
  - Control+F カーソルを右に移動（→と同じ）
  - Control+A カーソルを行の先頭に移動
  - Control+E カーソルを行の最後に移動
- GUIからヒストリーパネルを使って履歴をたどることも可能

ヒストリーパネルの表示・非表示

ベクトル

```
> a <- c(1, 3, 7, 4, 6)
> a
 [1] 1 3 7 4 6
> b <- 3:7
> b
 [1] 3 4 5 6 7
> length(a)
 [1] 5
> b[3]
[1] 5
```

ベクトルは関数 c を用いて作成する

3から7までの連続した整数

ベクトルaの長さ（要素数）

bの3番目の要素

ベクトル演算

a=c(1,3,7,4 6); b=c(3,4,5,6,7) と設定されている

```
> 1 + a
[1] 2 4 8 5 7
> 2 * a
[1]  2  6 14 8 12
> a + b
[1]  4  7 12 10 13
> a * b
[1]  3  12 35 24 42
> a > 3
[1] FALSE FALSE TRUE  TRUE  TRUE
> a < b
[1] TRUE TRUE FALSE TRUE TRUE
```

aの各要素に1を加える

aの各要素を2倍する

aとbの要素ごとの和をとる

aとbの要素ごとの積をとる

aの要素ごとに3より大きいか比較する

aとbを要素ごとに大小を比較する

基本統計量の計算

a=c(1,3,7,4,6)と設定されている

```
> length(a)
> sum(a)
> mean(a)
> median(a)
> var(a)
> sd(a)
```

aの長さ(要素数)
aの要素の合計値
aの要素の平均値
aの要素の中央値
aの要素の不偏分散
aの要素の標準偏差

問題） mean(a)をsum(a)とlength(a)を使って計算してみよう。

ベクトル要素の抽出

a=c(1,3,7,4,6)と設定されている

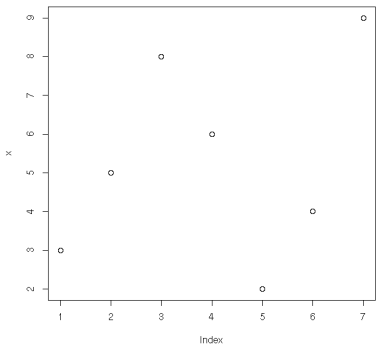
```
> a[2:4]              # 2番目から4番目までの要素
[1] 3 7 4
> a[ c(3,5,2) ]       # 3, 5, 2番目の要素
[1] 7 6 3
> a[ c(T,T,F,F,T) ] # T(TRUE)である要素だけ出力
[1] 1 3 6
> a[ a > 3 ]          # a>3がTRUEである要素だけ出力
[1] 7 4 6
```

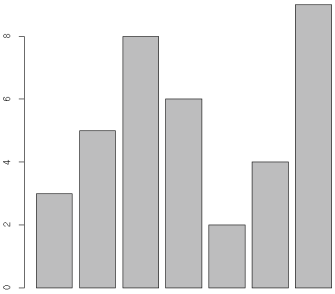
並べかえ（ソート）

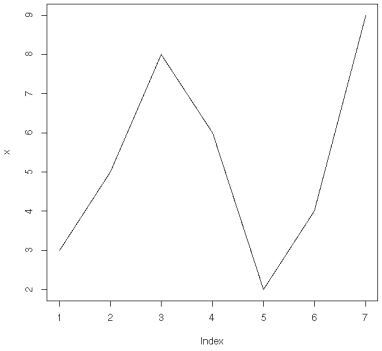
```
> x <- c(3,5,8,6,2,4,9)
> sort(x)                    # 小さい順に並べかえ
[1] 2 3 4 5 6 8 9
> sort(x, decreasing=TRUE) # 大きい順
[1] 9 8 6 5 4 3 2
```

プロットの作成(1)

```
> x
[1] 3 5 8 6 2 4 9
> plot(x)
> plot(x,type="l")
> barplot(x)
```







プロットの作成(2)

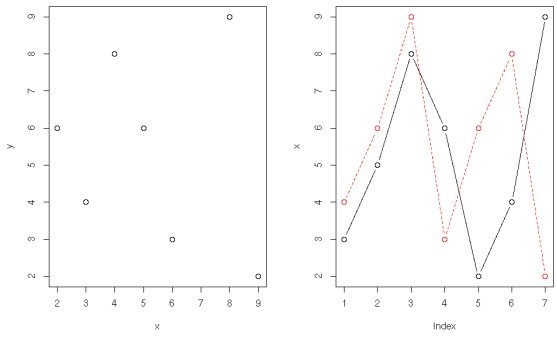
```
# x <- c(3,5,8,6,2,4,9)と設定されている
> y <- c(4,6,9,3,6,8,2)

> par(mfrow=c(1,2))   # 2つのプロットを表示
# x,y をx軸、y軸にとって散布図としてプロット

> plot(x,y)
# x,yを別々にプロット。まずxをプロット。点と線を両方描く。

> plot(x,type="b")
# yを重ねてプロット。linesは枠を描き直さずに線だけを引く。
# lty はline type, col は colorを指定。

> lines(y, type="b", lty=2, col=2)
```



Rにおける基本データ型

- **Rの「原子データ型」として以下のものがある**
  - 数値型 numeric（実数または整数）
  - 論理型 logical（TRUE/FALSE)
  - 文字列型 character("abc"、"123" のように、二重引用符で囲まれた文字列として表す）
- **Rの原子データはベクトルである。スカラー値も長さ1のベクトルである。すなわち、ベクトルは型を持ち、すべての要素は同じ型のデータからなる。**
- **異なる型のデータを集めた構造として「リスト」がある。**
- **mode(x) によって、変数 x の型を調べられる**

演算子

- **算術演算子**

  +（加算）–（減算）*（乗算）/（除算）,

  %/%（整数除算）%%（剰余）^（累乗）

  例）　5/2 (=2.5)　5%/%2 (=2)　5%%2 (=1)

- **論理演算子**

  &（論理積「かつ」）　|（論理和「または」）　! a（aの否定）

- **比較演算子**

  >, >=, <, <=（不等号）==（等しい）!=（等しくない）

これらの演算子はベクトルの要素ごとにはたらく

（例）a=c(1,3,7,4,6)と設定されている

```
> a >= 2 & a < 5
[1] FALSE  TRUE  FALSE  TRUE FALSE
```

ワークスペースとオブジェクト

- コンソール上で `ls()` または `objects()`とすると、変数に保存されたデータ(オブジェクト)のリストを参照できる。

```
>ls()
[1] "a"  "b"  "x"
```

- ワークスペースブラウザ(メニューバー「ワークスペース」から起動)でより詳細な情報を閲覧可能
- `rm(`変数名`)`で、オブジェクトを消去できる。

関数

関数名(引数1, 引数2, ...)

- 関数は、一般に複数の引数を入力としてとり、何らかの計算を行って一つのオブジェクトを返す(戻り値)。
- 引数には、必須のものと省略可能なものがある。後者は省略すると「デフォルト値」が使用される。
- 引数は、順番によって指定する方法と、「名前＝値」の形式で指定する方法がある。通例、必須の引数は前者、選択可能な引数は後者で指定する。

例1) ベクトル x を小さい順(increasing order)でソートする

```
sort(x)
```

例2) ベクトル x を大きい順 (decreasing order)でソートする

```
sort(x, decreasing=TRUE)
```

マニュアルの表示

- **help(関数名)または ?関数名　でマニュアルを表示する**

```
> help(median)
```

median　　package:stats　　R Documentation

Median Value

Description:

Compute the sample median.

Usage:

median(x, na.rm = FALSE)

Arguments:

x: an object for which a method has been defined, or a numeric vector containing the values whose median is to be computed.

na.rm: a logical value indicating whether 'NA' values should be stripped before the computation proceeds.

Details:

This is a generic function for which methods can be written. However, the default method makes use of 'sort' and 'mean' from package 'base' both of which are generic, and so the default method will work for most classes (e.g. '"Date"') for which a median is a reasonable concept.

Value:

The default method returns a length-one object of the same type as 'x', except when 'x' is integer of even length, when the result will be double.

If there are no values or if 'na.rm = FALSE' and there are 'NA' values the result is 'NA' of the same type as 'x' (or more generally the result of 'x[FALSE][NA]').

References:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New S Language_. Wadsworth & Brooks/Cole.

See Also:

'quantile' for general quantiles.

Examples:

```
median(1:4)# = 2.5 [even number]
median(c(1:3,100,1000))# = 3 [odd, robust]
```

Description: 関数の簡単な説明

Usage: 関数の呼び出し方

median(x, na.rm = FALSE)

x: 必須の引数
na.rm: 省略可能な引数
デフォルト値はFALSE

Arguments: 各引数の詳しい説明

Details: 関数の動作の詳しい説明

Values: 戻り値の説明

Reference:方法に関する文献

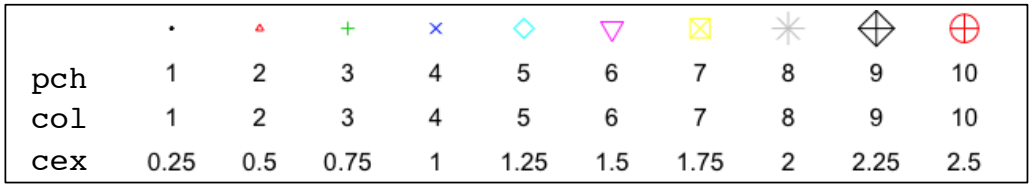
See Also: 関連するコマンド

Examples: 実行例

（参考） plotのオプション

- `main="title"`　グラフのタイトル
- `xlab(ylab)="label"`　x軸(y軸)のラベル
- `log="xy"`　対数軸の指定
- `xlim(ylim)=c(0,100)`　x軸(y軸)の値の範囲の設定
- `type="l"`　`"p"`(点), `"l"`(線), `"b"`(両方), `"n"`(枠だけ) など
- `lty=1`　プロットする線の種類
- `pch=1`　プロットする点の種類(文字)
- `col=2`　プロットする点および線の色
- `cex=0.8`　プロットする文字の大きさ

ベクトルとして指定することにより、点ごとに色や種類を変更することもできる。例)`pch=c(1,2,1,3,2)`



✧ plotのオプションは,、`help(plot)`だけでは調べられない。一部のオプションは`help(plot.default)`, `plot(par)`を参照する必要がある。
✧ 既存のグラフ上に線を重ねる場合は`lines`, 点を重ねる場合は`points`を使う。
✧ 凡例をつけたいときは`legend`関数を使う。