

R のコマンドのまとめ 今野 良彦 (2015.10.07)

ホームページ

<http://www.okada.jp.org/RWiki/?RjpWiki>
の検索などが役に立つ .

四則演算

| | | | | | |
|---|-----|---|-----|----|-----|
| + | 足し算 | - | 引き算 | * | 掛け算 |
| / | 割り算 | ^ | 累乗 | %% | 整数商 |

初等関数等

| | | | |
|---------|------|---------|-------|
| sqrt() | √ | abs() | 絶対値 |
| exp() | 指数関数 | log() | 自然対数 |
| log10() | 常用対数 | log2() | 底が 2 |
| sin() | sin | cos() | cos |
| tan() | tan | gamma() | ガンマ関数 |

四捨五入・切捨て

| | |
|-------------|-------------------|
| ceiling() | 引数以上の最小整数 |
| floor() | ガウス記号 |
| round() | 四捨五入 |
| signif(x,a) | x を 有効桁で a 桁まで丸める |

ヘルプをみる

```
help(solve)
?solve
```

オブジェクトの表示

```
x <- "one"
print(x)          # "" ありで出力
[1] "one"
print(x, quote=F) # "" なしで出力
[1] one
```

出力をファイルに送る

作業ディレクトリの中に output.txt が出来る

```
sink("output.txt")
```

出力をファイルに送るのを解除

```
sink()
```

ベクトルの作成 (1)

```
c(1.0, 2.0, 3.0, 4.0, 5.0)
[1] 1 2 3 4 5 # 長さ 5 のベクトルを作成
( x <- c(1.0, 2.0, 3.0, 4.0, 5.0) )
[1] 1 2 3 4 5 # x という名のオブジェクトにベクトル
length(x)    # ベクトルの長さを調べる
[1] 5
```

ベクトルの作成 (2)

1 から 5 まで 1 ずつ増加するベクトルを作る

```
1:5
```

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

数列 (1 2 3) を長さ 8 になるまで反復生成

```
rep(1:3, length=8)
```

```
[1] 1 2 3 1 2 3 1 2
```

1 から 10 までを等分割した長さ 5 のベクトルを作る

```
seq(1, 10, length=5)
```

```
[1] 1.00 3.25 5.50 7.75 10.00
```

ベクトルの作成 (3)

a から b (a<b) までの交差が 1 の数列を生成

```
a:b
```

a, b 間を n 等分する等差数列を生成

```
seq(a, b, length = n)
```

a から b まで c ずつ増加するベクトルを生成

```
seq(a, b, by = c)
```

a から b まで 1 ずつ増加する数列を c 個生成

```
rep(a:b, times = c)
```

データの修正)

最後に追加

```
x<-c(10,20,30,40,50,60)
```

```
newx<-append(x,c(70,80))
```

3 番目と 4 番目にベクトルを追加

```
newxx<-c(x[1:3],c(33,36,39),x[4:length(x)])
```

```
newxxx<-append(x,c(33,36,39),after=3)
```

3 番目のデータの削除

```
delx<-x[-3]
```

3 番目のデータの変更

```
x[3]<-100
```

ベクトルの演算

ベクトルの足し算

```
c(1, 2, 3) + c(4, 5, 6)
```

```
[1] 5 7 9
```

スカラー倍

```
3*c(1.0, 2.0, 3.0, 4.0, 5.0)
```

全ての要素から 1 を引く

```
c(1.0, 2.0, 3.0, 4.0, 5.0) - 1
```

```
[1] 0 1 2 3
```

ベクトルの関数

| | | | |
|---------|--------------|----------|--------|
| sum() | 総和 | mean() | 平均 |
| var() | 不偏分散 | median() | 中央値 |
| cor() | 相関係数 | max() | 最大値 |
| min() | 最小値 | sd() | 標準偏差 |
| sort() | 昇順整列 | rev() | 要素を逆順 |
| range() | 範囲 | rank() | 各要素の順位 |
| order() | 整列した各要素の元の位置 | | |

ベクトルの要素

3 番目の要素を取り出す

```
x <- c(1, 2, 3, 4, 5)
x[3]
[1] 3
```

5 番目の要素を 0 に変更する

```
x <- c(1, 2, 3, 4, 5)
x
[1] 1 2 3 4 5
x[5] <- 0
x
[1] 1 2 3 4 0
```

行列の作成

行列 $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ の作成

```
matrix(1:6, nrow=2, ncol=3)
# matrix(1:6, 2, 3) と略記しても良い
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
```

行列の要素の指定

```
( x <- matrix(1:6, nrow=2, ncol=3) )
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
x[1, 2]
[1] 3
```

ベクトルの関数

| | |
|-------------------|----------------------|
| x[2,] | 2 行目を取り出す |
| x[,2] | 2 列目を取り出す |
| x[1, 2] | 1 行 2 列目の成分を取り出す |
| x[c(1,2), 2] | 1,2 行 2 列目の成分を取り出す |
| x[c(1,2), c(1,3)] | 1, 2 行目と 1, 3 列を取り出す |
| x[, c(T,F,T)] | 1, 3 列を取り出す |
| rowSums() | 行の総和 |
| colSums() | 列の総和 |
| rowMeans() | 行の平均 |

行列の計算

行列生成

```
a <- matrix(1:4, 2, 2)
b <- matrix(0:3, 2, 2)
```

和・差・積

```
a + b - a %*% b
```

行列計算用関数

| | |
|-------------------------|--------------------------|
| diag(1, ncol=3, nrow=3) | 3 × 3 の単位行列 |
| t(x) | 行列 x を転置 |
| x[upper.tri(x)] <- 0 | 行列 x の下三角行列化 |
| X[lower.tri(x)] <- 0 | 行列 x の上三角行列化 |
| solve(x) | 行列 x の逆行列 (存在する場合) |
| eigen(x) | 行列 x の固有値と 固有ベクトルを求める |
| det(x) | 行列 x の行列式 |
| colSums() | 列の総和 |
| rowMeans() | 行の平均 |

行列の結合

行ベクトルを与えて行列生成

```
( x <- rbind(c(1,2,3), c(4,5,6)) )
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
```

列ベクトルを与えて行列生成

```
( x <- cbind(c(1,2,3),c(4,5,6)) )
```

自前の関数の作成

```
関数名 <- function( 引数1, ..., 引数n ) {
  <関数本体>
}
```

```
myfunc <- function(x) ifelse(x > 0, 1, -1)
myfunc(1)
[1] 1
myfunc(-2)
[1] -1
```

自分で関数を定義する場合は R に既に用意されている関数と同じ名前の関数を定義しないように注意

比較演算子・比較演算関数

| | | | | | |
|----|----|----|---|----|---|
| == | 等号 | >= | ≥ | != | ≠ |
| > | > | <= | ≤ | < | < |

論理演算子

| | | | |
|---|-----|-------|---------|
| ! | 否定 | && | 条件での論理積 |
| & | 論理積 | | 条件での論理和 |
| | 論理和 | xor() | 排他的論理和 |

```
x <- 7
(x > 2) && (x < 4)
[1] FALSE
if ( (x > 2) || (x < 8) ) print("True.")
[1] "True."
```

繰り返し文

```
for ( ループ変数 in リスト )
x <- 0 for ( i in 1:5 ) {
x <- x - 1
}
x
[1] -5
```

while による繰り返し

```
x <- 0
while (x > -6) {
x <- x - 2
}
x
[1] -6
```

グラフ画面の切り替え

```
par(mfrow=c(1,2))
par(mfrow=c(1,1))
```

曲線の上書き

```
plot(cos, -pi, pi)
curve(sin, add=T)
plot(sin, xlim=c(0,5), ylim=c(-1,1), lty=2)
par(new=T)
plot(cos, xlim=c(0,5), ylim=c(-1,1))
```

ヒストグラムの書き方

```
x<-rnorm(100,0,1)
hist(x,breaks=-4(0:20)*0.4,prob=T)+
curve(dnorm,-4,4,add=T)
x<-rt(100,2)
hist(x,nclass=40)
```

標本分布の確率密度関数

| R での名前 (コマンド) | | |
|---------------|-------|--------------------|
| 分布名 | 名前 | 確率密度関数 |
| 正規分布 | norm | dnorm(x, 平均, 標準偏差) |
| t 分布 | t | dt(x, 自由度) |
| χ^2 分布 | chisq | dchisq(x, 自由度) |
| F 分布 | f | df(x, 自由度, 自由度) |

標本分布の累積分布関数

| R での名前 (コマンド) | | |
|---------------|-------|--------------|
| 分布名 | 名前 | 累積分布関数 |
| 正規分布 | norm | pnorm(x,0,1) |
| t 分布 | t | pt(x,1) |
| χ^2 分布 | chisq | pchisq(x,1) |
| F 分布 | f | pf(x,1,2) |

標本分布の乱数生成

| R での名前 (コマンド) | | |
|---------------|-------|---------------|
| 分布名 | 名前 | 擬似乱数 |
| 正規分布 | norm | rnorm(個数,0,1) |
| t 分布 | t | rt(個数,1) |
| χ^2 分布 | chisq | rchisq(個数,1) |
| F 分布 | f | rf(個数,1,2) |

一変数関数の零点を見付ける

```
uniroot(f, interval, lower = min(interval),
        upper = max(interval), maxiter = 1000)
f<-function(x){exp(x)-2}
uniroot(f,interval=c(0,1))
```