文字列関数 ― 文字列の切り出し・結合・検索・置換

切り出し
substr(対象変数名，始点，終点)

始点，終点で，要素の何文字目から何文字目までを切り出すかを指定する

結合
paste(文字列の指定，sep="区切りの指定")

sep="," でカンマ区切り，sep="" で区切りなし．sep= を省略すると半角スペース区切りになる．

検索
```
grep("検索する文字列", 検索対象, fixed = FALSE)
grepl("検索する文字列", 検索対象, fixed = FALSE)
regexpr("検索する文字列", 検索対象, fixed = FALSE)
gregexpr("検索する文字列", 検索対象, fixed = FALSE)
```

grep：　検索する文字列があったところの要素番号を返す．
grepl：　検索する文字列があるかどうか，すべての要素について TRUE，FALSE で返す．
regexpr：検索する文字列があれば，初出のもののみそれが何文字目かの値，なければ-1を返す．
gregexpr：検索する文字列が複数あればそれがそれぞれ何文字目かの値，なければ-1を返す．

置換
```
sub("検索する文字列", "置き換える文字列", 検索対象, fixed = FALSE)
gsub("検索する文字列", "置き換える文字列", 検索対象, fixed = FALSE)
```

sub：置換したい文字列があれば，初出のもののみ置換する．
gsub：置換したい文字列が複数あれば，それがそれらすべてを置換する．

【注】
fixed=TRUE とすると完全一致のみ評価．カタカナや特殊記号を含む文字列を評価する場合，fixed=FALSE か fixed=TRUE の指定をしておかないとエラーになることがある．

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("文字列_データ.csv", header=TRUE, sep=",")
> d1
  id         course
1  1           行動
2  2           臨床
3  3     行動コース
4  4     臨床コース
5  5     コース行動
6  6     コース臨床
7  7 行動行動コース
8  8 臨床臨床コース
>
```

	A	B
1	id	course
2	1	行動
3	2	臨床
4	3	行動コース
5	4	臨床コース
6	5	コース行動
7	6	コース臨床
8	7	行動行動コース
9	8	臨床臨床コース
10		

```
> # 文字列の切り出し
> d1$course1 <- substr(d1$course,1,3)
> d1
  id         course course1
1  1           行動    行動
2  2           臨床    臨床
3  3     行動コース  行動コ
4  4     臨床コース  臨床コ
5  5     コース行動  コース
6  6     コース臨床  コース
7  7 行動行動コース  行動行
8  8 臨床臨床コース  臨床臨
```

```
> # 文字列の結合
> d1$course2 <- paste("心理", d1$course)
> d1$course3 <- paste("心理", d1$course, sep="")
> d1
  id         course course1             course2            course3
1  1           行動    行動           心理 行動           心理行動
2  2           臨床    臨床           心理 臨床           心理臨床
3  3     行動コース  行動コ      心理 行動コース     心理行動コース
4  4     臨床コース  臨床コ      心理 臨床コース     心理臨床コース
5  5     コース行動  コース      心理 コース行動     心理コース行動
6  6     コース臨床  コース      心理 コース臨床     心理コース臨床
7  7 行動行動コース  行動行 心理 行動行動コース 心理行動行動コース
8  8 臨床臨床コース  臨床臨 心理 臨床臨床コース 心理臨床臨床
>
>
> # course2 は半角スペース区切り，course3 は区切りなしで，"心理" が結合されている.


> # 文字列の検索

> grep("行動", d1$course, fixed = FALSE)                    # "行動"を含む行番号を表示
[1] 1 3 5 7

> grepl("行動", d1$course, fixed = FALSE)
[1]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE        # 各行が"行動"を含むかどうかを表示

> regexpr("行動", d1$course, fixed = FALSE)
[1]  1 -1  1 -1  4 -1  1 -1                                # "行動"を含めばそれが何文字目か,
attr(,"match.length")                                      # 含まなければ-1を表示
[1]  2 -1  2 -1  2 -1  2 -1

> gregexpr("行動", d1$course, fixed = FALSE)
[[1]]                                                      # "行動"を複数含めばそれぞれ何文字目か,
[1] 1                                                      # 含まなければ-1を表示
attr(,"match.length")
[1] 2

# 中略

[[6]]
[1] -1
attr(,"match.length")
[1] -1

[[7]]
[1] 1 3
attr(,"match.length")
[1] 2 2

[[8]]
[1] -1
attr(,"match.length")
[1] -1
>


> # 文字列の置換

> sub("行動", "教心",d1$course, fixed = FALSE)
[1] "教心"           "臨床"           "教心コース"     "臨床コース"
[5] "コース教心"     "コース臨床"     "教心行動コース" "臨床臨床コース"

# 同一要素内の１つめの"行動"は"教心"に置換されるが，それ以降のものは置換されない.


> gsub("行動", "教心",d1$course, fixed = FALSE)
[1] "教心"           "臨床"           "教心コース"     "臨床コース"
[5] "コース教心"     "コース臨床"     "教心教心コース" "臨床臨床コース"
>
# 同一要素内のすべての"行動"が"教心"に置換される.
```

演算記号・算術関数

```
> x <- 5
> y <- 2
>

> # 和
> x + y
[1] 7
>

> # 差
> x - y
[1] 3
>

> # 積
> x * y
[1] 10
>

> # 除算
> x / y
[1] 2.5
>

> # 商
> x %/% y
[1] 2
>

> # 余り
> x %% y
[1] 1
>

> # べき乗
> x^y
[1] 25
>

> # 平方根
> sqrt(x)
[1] 2.236068
>

> # 切り上げ，切り捨て，四捨五入
> w <- 2.506
> ceiling(w)
[1] 3
> floor(w)
[1] 2
> trunc(w)
[1] 2
> round(w,2)
[1] 2.51
>
> v <- -2.506
> ceiling(v)
[1] -2
> floor(v)
[1] -3
> trunc(v)
[1] -2
> round(v,2)
[1] -2.51
>
```

```
> # 指数
> exp(x)
[1] 148.4132
>

> # 自然対数
> log(x)
[1] 1.609438
>

> # 三角関数
> sin(x)
[1] -0.9589243
> cos(x)
[1] 0.2836622
> tan(x)
[1] -3.380515
> u <- 0.5
> asin(u)
[1] 0.5235988
> acos(u)
[1] 1.047198
> atan(u)
[1] 0.4636476
>
>

> # 最大値・最小値
> a <- c(1,3,2,5,4)
> max(a)
[1] 5
> min(a)
[1] 1
>

> # 最大値・最小値の要素番号
> which.max(a)
[1] 4
> which.min(a)
[1] 1
>

> # 異なるベクトルの同じ位置にある要素の最大値・最小値
> b <- c(2,4,4,4,3)
> pmax(a,b)
[1] 2 4 4 5 4
> pmin(a,b)
[1] 1 3 2 4 3
>
>
```

集合関数

```
> x <- c("a","c","e")
> y <- c("a","b","c","d")
>
> # 和集合
> (z <- union(x,y))
[1] "a" "c" "e" "b" "d"
>
> # 積集合
> (z <- intersect(x,y))
[1] "a" "c"
>
> # xの要素のうち，yに含まれるものを除いた集合
> (z <- setdiff(x,y))
[1] "e"
>
> # xの各要素はyの要素であるか
> (z <- is.element(x,y))
[1]  TRUE  TRUE FALSE

> (z <- x %in% y)
[1]  TRUE  TRUE FALSE
>
> # 集合として等しいか
> (z <- setequal(x,y))
[1] FALSE
>

> w <- c("a","c","e","a","c","e")
> (z <- setequal(x,w))
[1] TRUE
>
>
```

比較演算子

```
> x <- 5
> y <- 5
> z <- 5.00000001
>
> w <- NA
>
> t <- c(1,1,1)
> u <- c(2,2,2)
> v <- c(1,2,3)
>
>

> # 比較演算子
>
> # 等しいか
> x == y
[1] TRUE

> identical(x,y)
[1] TRUE

> identical(x,z)
[1] FALSE
>

> # ほとんど等しいか
> all.equal(x,z)
[1] TRUE
>

> # 等しくないか
> x != y
[1] FALSE
>

> # 以上か
> x >= y
[1] TRUE
>

> # 超か
> x > y
[1] FALSE
>

> # 以下か
> x <= y
[1] TRUE
>

> # 未満か
> x < y
[1] FALSE
>
>
> # 欠測値か
> is.na(w)
[1] TRUE
>
>
>
```

論理演算子

```
> x <- 5
> y <- 5
> z <- 5.00000001
>
> w <- NA
>
> t <- c(1,1,1)
> u <- c(2,2,2)
> v <- c(1,2,3)
>
>

> # 論理演算子
>
> # かつ・または(スカラー)
> (x==5) && (y==4)
[1] FALSE

> (x==5) || (y==4)
[1] TRUE
>


> # かつ・または(ベクトル)
> # if文でベクトルを評価した場合は，先頭要素の真偽で判断される
> (t==v) & (u==v)
[1] FALSE FALSE FALSE

> (t==v) | (u==v)
[1]  TRUE  TRUE FALSE
>


> # でない
> !(x==4)
[1] TRUE

> !(u==v)
[1]  TRUE FALSE  TRUE
>
```

ベクトルの生成

```
> # ベクトルの生成
> (x <- c(1,4,3))
[1] 1 4 3


> # 値が1ずつ増えていくベクトル
> (x <- c(1:9))
[1] 1 2 3 4 5 6 7 8 9


> (x <- seq(1,9))
[1] 1 2 3 4 5 6 7 8 9
>


> # ベクトルを逆順にする
> (y <- rev(x))
[1] 9 8 7 6 5 4 3 2 1


> # 指定した値ずつ増えていくベクトル
> (x <- seq(1,9,by=2))
[1] 1 3 5 7 9
>


> # 同一ベクトルを指定個つなげる
> (x <- rep(c("a","b","c"),2))
[1] "a" "b" "c" "a" "b" "c"
>


> # 各要素を指定回繰り返す
> (x <- rep(c("a","b","c"),each=2))
[1] "a" "a" "b" "b" "c" "c"
>


> # 重複要素のあるベクトルから素な要素だけを抽出する
> unique(x)
[1] "a" "b" "c"
>


> # ベクトル1の各要素が，ベクトル2の何番目の要素に対応するかを返す
> charmatch(c(1:6), c(1,4,5))
[1]  1 NA NA  2  3 NA
>


> # ベクトル1の各要素が，ベクトル2のいずれかの要素であるか否かを返す
> c(1:6) %in% c(1,4,5)
[1]  TRUE FALSE FALSE  TRUE  TRUE FALSE
>
>
```

行列演算

```
> # 行列の生成
> # データフレームを行列に変える
> x <- as.matrix(d1)

> # 要素を入力して行列を作成する
> (x <- matrix(c(
+         1,2,
+         3,4,
+         5,6), 3, 2, byrow=T))
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
>
> (y <- matrix(c(
+         1,1,
+         0,1,
+         1,0), 3,2, byrow=T))
     [,1] [,2]
[1,]    1    1
[2,]    0    1
[3,]    1    0

> # 単位行列
> diag(3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
>


> # 行列の転置
> t(x)
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
>

> # 行列の足し算
> (x+y)
     [,1] [,2]
[1,]    2    3
[2,]    3    5
[3,]    6    6
>

> # 行列の引き算
> (x-y)
     [,1] [,2]
[1,]    0    1
[2,]    3    3
[3,]    4    6
>

> # 要素の2乗
> x^2
     [,1] [,2]
[1,]    1    4
[2,]    9   16
[3,]   25   36


> # 行列のかけ算
> (z <- t(x) %*% y)
     [,1] [,2]
[1,]    6    4
[2,]    8    6
```

```
> (w <- x %*% t(y))
     [,1] [,2] [,3]
[1,]    3    2    1
[2,]    7    4    3
[3,]   11    6    5
>

> # クロネッカー積
> x %x% y
      [,1] [,2] [,3] [,4]
 [1,]    1    1    2    2
 [2,]    0    1    0    2
 [3,]    1    0    2    0
 [4,]    3    3    4    4
 [5,]    0    3    0    4
 [6,]    3    0    4    0
 [7,]    5    5    6    6
 [8,]    0    5    0    6
 [9,]    5    0    6    0


> # 対角行列
> diag(c(1,2,3))
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
>

> # 対角要素の取り出し
> diag(w)
[1] 3 4 5
>

> # 対角要素だけを残した行列の作成
> diag(diag(w))
     [,1] [,2] [,3]
[1,]    3    0    0
[2,]    0    4    0
[3,]    0    0    5
>

> # 三角行列
> u <- v <- w
> u[lower.tri(u)] <- 0
> v[upper.tri(v)] <- 0
> u
     [,1] [,2] [,3]
[1,]    3    2    1
[2,]    0    4    3
[3,]    0    0    5
> v
     [,1] [,2] [,3]
[1,]    3    0    0
[2,]    7    4    0
[3,]   11    6    5
>

> # 行列式
> det(z)
[1] 4

> round(det(w),4)
[1] 0
>

> # 逆行列
> (invz <- solve(z))
     [,1] [,2]
[1,]  1.5 -1.0
[2,] -2.0  1.5
```

```
> # 一般逆行列
> library(MASS)
> (ginv(w))
             [,1]           [,2]         [,3]
[1,] -0.08333333  8.257284e-16  0.08333333
[2,]  1.16666667  3.333333e-01 -0.50000000
[3,] -1.25000000 -3.333333e-01  0.58333333
>

> # 固有値分解
> eigen(z)
$values
[1] 11.6568542  0.3431458

$vectors
          [,1]       [,2]
[1,] 0.5773503 -0.5773503
[2,] 0.8164966  0.8164966

>
> # 特異値分解
> svd(x)
$d
[1] 9.5255181 0.5143006

$u
           [,1]       [,2]
[1,] -0.2298477  0.8834610
[2,] -0.5247448  0.2407825
[3,] -0.8196419 -0.4018960

$v
           [,1]       [,2]
[1,] -0.6196295 -0.7848945
[2,] -0.7848945  0.6196295

>
> # QR分解
> qr(x)
$qr
           [,1]       [,2]
[1,] -5.9160798 -7.4373574
[2,]  0.5070926  0.8280787
[3,]  0.8451543  0.9935832

$rank
[1] 2

$qraux
[1] 1.169031 1.113104

$pivot
[1] 1 2

attr(,"class")
[1] "qr"
>

> # コレスキー分解
> chol(z)
         [,1]     [,2]
[1,] 2.44949 1.632993
[2,] 0.00000 1.825742

> chol(w)
         [,1]     [,2]      [,3]
[1,] 1.732051 1.154701 0.5773503
[2,] 0.000000 1.632993 1.4288690
[3,] 0.000000 0.000000 1.6201852
```

制御コマンド

評価する条件式に NA が入るとエラーになってしまうので，is.na(変数名) などとして，NAであればTRUE，NAでなければFALSEを返すように，まずしておくとよい．

```
for(ループ変数 in 値を変化させる範囲){
 ループ変数の値の変化に伴い実行される式の群
}
```

```
if (条件式1) {
 条件式1がTRUEの時に実行される式の群
}
 else if (条件式2) {
 条件式1がFALSEで，条件式2がTRUEの時に実行される式の群
}
…
 else{ 以上の条件式がすべてFALSEのときに実行される式の群
}
```

実行される式が1つであれば{}で囲わなくてもよい．
else if 以下は省略することができる．
if ((条件式1a) & (条件式1b)) とすれば，条件式1a **および** 条件式1bの**両方**がTRUEの場合，となる
if ((条件式1a) | (条件式1b)) とすれば，条件式1a **または** 条件式1bの**一方**がTRUEの場合，となる

```
ifelse(ある変数についての条件式，条件式がTRUEの時の当該変数の値，条件式がFALSEの時の当該変数の値)
```

```
while(条件式){
 実行される式の群
}
```

式を実行する前に条件式の評価を行い，条件式がTRUEである間，式を実行する．

条件式が常にTRUEであれば，いつまでも関数が動いてしまうので注意．

```
repeat{
 実行される式の群
 if(条件式) break
}
```

式を実行した後に条件式の評価を行い，条件式がTRUEになれば終了する．
if(条件式) break を入れるのを忘れない．

```
break
```

for，while，repeat などのループから強制的に抜け出すコマンド．

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("制御_データ.csv", header=TRUE, sep=",")
> d1
   x1 x2 x3
1  NA  3  2
2   3  3  2
3   1  3  2
4   3  3  2
```

```
5   3  3  3
6   3  3  2
7   2  3  3
8   2  3  2
9   3  3  2
10  2  3  2
11  1  3  2
12  3  3  2
13  1  3  3
14  2  3  2
15 NA NA  2
>
```

	A	B	C
1	x1	x2	x3
2		3	2
3	3	3	2
4	1	3	2
5	3	3	2
6	3	3	3
7	3	3	2
8	2	3	3
9	2	3	2
10	3	3	2
11	2	3	2
12	1	3	2
13	3	3	2
14	1	3	3
15	2	3	2
16			2
17			

```
> # d1の行数
> nr <- nrow(d1)
>

> # d1の変数名
> cnames1 <- colnames(d1)
>

> # d1の行数，2列，要素の値0の行列を生成
> d2 <- matrix(c(0),nr,2)
>

> # d1とd2を結合
> d1 <- data.frame(d1,d2)
> colnames(d1) <- c(cnames1, "s1","m1")
> d1
   x1 x2 x3 s1 m1
1  NA  3  2  0  0
2   3  3  2  0  0
3   1  3  2  0  0
4   3  3  2  0  0
5   3  3  3  0  0
6   3  3  2  0  0
7   2  3  3  0  0
8   2  3  2  0  0
9   3  3  2  0  0
10  2  3  2  0  0
11  1  3  2  0  0
12  3  3  2  0  0
13  1  3  3  0  0
14  2  3  2  0  0
15 NA NA  2  0  0
>
```

```
> #for, if
> for (i in 1 : nr){                          # iを1からnrまで変化させる
+  if (is.na(d1$x1[i])!=TRUE){                 # x1が欠測でなければ，
+   if (d1$x1[i]==1) d1$s1[i] <- 0.5            # x1が1ならs1を0.5とする
+    else if (d1$x1[i]==2) d1$s1[i] <- 1        # x1が2ならs1を1とする
+    else d1$s1[i] <- 0                         # x1がそれ以外なら，s1を0とする
+  }
+  else {
+   d1$s1[i] <- 0                               # x1が欠測値なら，s1を0とし，
+   d1$m1[i] <- 1                               # さらに，m1を10とする
+  }
+ }
> d1
   x1 x2 x3  s1 m1
1  NA  3  2 0.0  1
2   3  3  2 0.0  0
3   1  3  2 0.5  0
4   3  3  2 0.0  0
5   3  3  3 0.0  0
6   3  3  2 0.0  0
7   2  3  3 1.0  0
8   2  3  2 1.0  0
9   3  3  2 0.0  0
```

```
10  2  3  2 1.0  0
11  1  3  2 0.5  0
12  3  3  2 0.0  0
13  1  3  3 0.5  0
14  2  3  2 1.0  0
15 NA NA  2 0.0  1
>
>

> # ifelse
> ifelse(d1$x3==2, 1, 0)                          # x3が2なら１，それ以外なら0を表示
 [1] 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1
>


> # while
> j <- 1                                  # はじめにjの値を1と指定
> while(j <= 10){                         # jが10以下である間は次を繰り返す
+  print(d1[j,])                            # d1の第j行を表示
+  j <- j+1                                 # jの値を1だけ増加させる
+ }
  x1 x2 x3 s1 m1
1 NA  3  2  0  1
  x1 x2 x3 s1 m1
2  3  3  2  0  0
  x1 x2 x3  s1 m1
3  1  3  2 0.5  0
  x1 x2 x3 s1 m1
4  3  3  2  0  0
  x1 x2 x3 s1 m1
5  3  3  3  0  0
  x1 x2 x3 s1 m1
6  3  3  2  0  0
  x1 x2 x3 s1 m1
7  2  3  3  1  0
  x1 x2 x3 s1 m1
8  2  3  2  1  0
  x1 x2 x3 s1 m1
9  3  3  2  0  0
   x1 x2 x3 s1 m1
10  2  3  2  1  0
>


> # repeat
> k <- 1                                  # はじめにkの値を1と指定
> repeat{
+  print(d1[k,])                            # d1の第k行を表示
+  k <- k+1                                 # kの値を1だけ増加させる
+  if (k > 8) break                         # もしkが8を超えたら終了．そうでなければ上を繰り返す
+ }
  x1 x2 x3 s1 m1
1 NA  3  2  0  1
  x1 x2 x3 s1 m1
2  3  3  2  0  0
  x1 x2 x3  s1 m1
3  1  3  2 0.5  0
  x1 x2 x3 s1 m1
4  3  3  2  0  0
  x1 x2 x3 s1 m1
5  3  3  3  0  0
  x1 x2 x3 s1 m1
6  3  3  2  0  0
  x1 x2 x3 s1 m1
7  2  3  3  1  0
  x1 x2 x3 s1 m1
8  2  3  2  1  0
>
```

コマンドを生成して実行

```
eval(parse(text=paste(コマンド文を生成する指定, sep="")))
```

for文で，iやjの値を変数の指定に用いて，コマンドを生成して実行するのに用いることも多い．

```
> x <- 2
> y <- 4
>
> paste(x, y, sep="")
[1] "24"

> paste(x+y, sep="")
[1] "6"

> paste(x,"+",y, sep="")
[1] "2+4"

> paste("x+y", sep="")
[1] "x+y"
>


> eval(parse(text=paste(x, y, sep="")))
[1] 24

> eval(parse(text=paste(x+y, sep="")))
[1] 6

> eval(parse(text=paste(x,"+",y, sep="")))
[1] 6

> eval(parse(text=paste("x+y", sep="")))
[1] 6
>
>
>
```

一括分析・総当たりの分析

分析に含める変数名を書いたベクトルの作成
ベクトル名 <- c(変数名の列)

一括分析・総当たりの分析
```
for(i in ベクトル名){
 オブジェクト名 <- eval(parse(text=paste(分析関数や変数の設定, sep="")))
 print(変数, quote = FALSE)
 print(オブジェクト名)
}
```

対応のある変数の総当たりの分析の場合は，for文を二重にして，iとjで回す.

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("総当たりの分析_データ.csv", header=TRUE, sep=",")
> head(d1)
  id class x1 x2 x3
1  1     a  3  3  2
2  2     a  3  3  2
3  3     a  3  3  2
4  4     a  3  3  2
5  5     b  3  3  3
6  6     a  3  3  2
>

> d1 <- d1[,c(-1)]
>
> # 記述統計量
> dtmp <- d1[d1$class=="a", colnames(d1) %in% c("class")==F]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp),2)
> colnames(ktmp) <- c("N","Mean","SD",colnames(ctmp))
> ktmp
    N Mean   SD   x1   x2   x3
x1 43 2.53 0.55 1.00 0.37 0.55
x2 43 2.84 0.43 0.37 1.00 0.20
x3 43 2.16 0.57 0.55 0.20 1.00
>
> dtmp <- d1[d1$class=="b", colnames(d1) %in% c("class")==F]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp),2)
> colnames(ktmp) <- c("N","Mean","SD",colnames(ctmp))
> ktmp
    N Mean   SD   x1   x2   x3
x1 43 2.21 0.67 1.00 0.61 0.34
x2 43 2.44 0.70 0.61 1.00 0.40
x3 43 2.19 0.55 0.34 0.40 1.00
>
```

	A	B	C	D	E
1	id	class	x1	x2	x3
2	1	a	3	3	2
3	2	a	3	3	2
4	3	a	3	3	2
5	4	a	3	3	2
6	5	b	3	3	3
7	6	a	3	3	2
8	7	a	2	3	3
9	8	a	2	3	2
10	9	a	3	2	2
11	10	b	3	3	3
12	11	b	3	3	2
13	12	b	2	2	2
14	13	a	3	3	2
15	14	a	3	3	2
16	15	a	2	3	2
17	16	a	3	3	2
18	17	a	3	3	2
19	18	a	3	3	3
20	19	a	2	3	2
21	20	a	2	3	2

```
> # 一括分析 ー 対応のない t 検定を指定したすべての変数について行う
> # 一括分析に用いる変数名
> var.names <- c("x1","x2","x3")

> # 対応のない t 検定を一括して行う
for(i in v.names){
 tmp <- eval(parse(text=paste("t.test(d1$",i, "~d1$class, paired=FALSE)",sep="")))
 print(i, quote = FALSE)
 print(tmp)
}
```

```
[1] x1

        Welch Two Sample t-test

data:  d1$x1 by d1$class
t = 2.4531, df = 80.715, p-value = 0.01632
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.06148781 0.58967498
sample estimates:
mean in group a mean in group b
       2.534884        2.209302

[1] x2

        Welch Two Sample t-test

data:  d1$x2 by d1$class
t = 3.1491, df = 69.974, p-value = 0.002409
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1449566 0.6457410
sample estimates:
mean in group a mean in group b
       2.837209        2.441860

[1] x3

        Welch Two Sample t-test

data:  d1$x3 by d1$class
t = -0.1925, df = 83.781, p-value = 0.8479
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2635701  0.2170585
sample estimates:
mean in group a mean in group b
       2.162791        2.186047
```

```
> # 総当たりの分析 ー 対応のある t 検定を指定した変数においてすべての組合せについて行う
>
> # 記述統計量
> dtmp <- d1[, colnames(d1) %in% c("class")==F]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp),2)
> colnames(ktmp) <- c("N","Mean","SD",colnames(ctmp))
> ktmp
    N Mean   SD   x1   x2   x3
x1 86 2.37 0.63 1.00 0.56 0.41
x2 86 2.64 0.61 0.56 1.00 0.29
x3 86 2.17 0.56 0.41 0.29 1.00
>


> # 総当たりの分析に用いる変数名
> v.names <- c("x1","x2","x3")
>
> # 対応のある t 検定を総当たりで行う
> p <- length(v.names)
> for(i in 1:(p-1)){
+  for(j in (i+1):p){
+   tmp <- eval(parse(text=paste("t.test(d1$",v.names[i], ",d1$",v.names[j], ",paired=FALSE)",sep
="")))
```

```
+   tcomp <- paste(v.names[i], "and", v.names[j])
+   print(tcomp, quote = FALSE)
+   print(tmp)
+  }
+ }

[1] x1 and x2

        Welch Two Sample t-test

data:  d1$x1 and d1$x2
t = -2.8163, df = 169.799, p-value = 0.005433
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.45489807 -0.07998565
sample estimates:
mean of x mean of y
 2.372093  2.639535

[1] x1 and x3

        Welch Two Sample t-test

data:  d1$x1 and d1$x3
t = 2.1733, df = 167.282, p-value = 0.03117
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.01810312 0.37724572
sample estimates:
mean of x mean of y
 2.372093  2.174419

[1] x2 and x3

        Welch Two Sample t-test

data:  d1$x2 and d1$x3
t = 5.2122, df = 168.527, p-value = 5.404e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2889529 0.6412796
sample estimates:
mean of x mean of y
 2.639535  2.174419

>
```

多変量正規乱数の発生

母集団値を指定した標本の発生

```
library(MASS)
Z <- mvrnorm(n, M, S, tol = 1e-6, empirical = FALSE)
```

標本値を指定した標本の発生

```
library(MASS)
Z <- mvrnorm(n, M, S, tol = 1e-6, empirical = TRUE)
```

MASSパッケージは最初からインストールされている.

個体数　　n
平均ベクトル　　M
共分散行列　　S
を指定する.

標準偏差ベクトル s　と　相関係数行列 R　から　共分散行列 S　を作成するのが分かりやすい.

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
>
> # 個体数
> n <- 245
>
> # 変数の数
> p <- 4
>
> # 標本平均ベクトル
> M <- c(15, 20, 18, 16)
>
> # 標本標準偏差ベクトル
> s <- c(5, 6, 5, 7)
>
> # 標本相関係数行列
> R <- matrix(c(
+  1.0,  0.4, -0.3, -0.6,
+  0.4,  1.0,  0.0, -0.3,
+ -0.3,  0.0,  1.0,  0.5,
+ -0.6, -0.3,  0.5,  1.0
+ ), p, p)
>
> # 標本共分散行列の計算
> D <- diag(s)
> S <- D %*% R %*% D
>
```

```
> # 母集団値を指定した２変量標準正規乱数の発生
> library(MASS)
> d1 <- mvrnorm(n, M, S, tol = 1e-6, empirical = FALSE)
> head(d1)
           [,1]     [,2]       [,3]       [,4]
[1,] 16.4792879 31.60532 12.605024  9.637972
[2,] 13.6160765 27.60231 24.699975 19.091026
[3,] 12.8210227 16.07044 15.202283 23.285821
[4,] 18.2472947 23.81371 21.262121 10.259094
[5,] 18.9166243 20.61683  8.419332  5.918010
[6,] -0.3445291 14.12684 23.404541 25.154694
>
```

```
> colnames(d1) <- c("x1","x2","x3","x4")
```

```
> # 記述統計量
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
```

```
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp),2)
> colnames(ktmp) <- c("N","Mean","SD",colnames(ctmp))
> ktmp
  N  Mean   SD    x1    x2    x3    x4
1 245 15.37  4.99  1.00  0.40 -0.30 -0.63
2 245 20.38  6.15  0.40  1.00  0.02 -0.30
3 245 18.20  4.58 -0.30  0.02  1.00  0.44
4 245 15.59  6.53 -0.63 -0.30  0.44  1.00

>
> # write.table(d1, "Z1.csv", row.names=FALSE, sep=",")
>
>
>
>
>
>
> # 標本値を指定した２変量標準正規乱数の発生
> library(MASS)
> d2 <- mvrnorm(n, M, S, tol = 1e-6, empirical = TRUE)
>

> colnames(d2) <- c("x1","x2","x3","x4")
>

> # 記述統計量
> dtmp <- d2
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp),2)
> colnames(ktmp) <- c("N","Mean","SD",colnames(ctmp))
> ktmp
  N  Mean   SD    x1    x2    x3    x4
1 245    15    5   1.0   0.4  -0.3  -0.6
2 245    20    6   0.4   1.0   0.0  -0.3
3 245    18    5  -0.3   0.0   1.0   0.5
4 245    16    7  -0.6  -0.3   0.5   1.0

>
>
> # write.table(d2, "Z2.csv", row.names=FALSE, sep=",")
>
>
```

確率関数

 # 確率関数

dxxxxx ： 確率密度関数
pxxxxx ： 確率分布関数
qxxxxx ： 分位点関数
rxxxxx ： 乱数

 # ベータ分布
dbeta(x, shape1, shape2)， **pbeta**(x, shape1, shape2)， **qbeta**(p, shape1, shape2)， **rbeta**(n, shape1, shape2)

 # 二項分布
dbinon(x, n0, p0)， **pbinon**(x, n0, p0)， **qinon**(p, n0, p0)， **rbinon**(n, n0, p0)

 # カイ２乗分布
dchisq(x, df)， **pchisq**(x, df)， **qchisq**(p, df)， **rchisq**(n, df)

 # 指数分布
dexp(x, rate)， **pexp**(x, rate)， **qexp**(p, rate)， **rexp**(n, rate)

 # ガンマ分布
dgamma(x, shape, rate)， **pgamma**(x, shape, rate)， **qgamma**(p, shape, rate)， **rgamma**(n, shape, rate)

 # 正規分布
dnorm(x, mean, sd)， **pnorm**(x, mean, sd)， **qnorm**(p, mean, sd)， **rnorm**(n, mean, sd)

 # Ｆ分布
df(x, df1, df2)， **pf**(x, df1, df2)， **qf**(p, df1, df2)， **rf**(n, df1, df2)

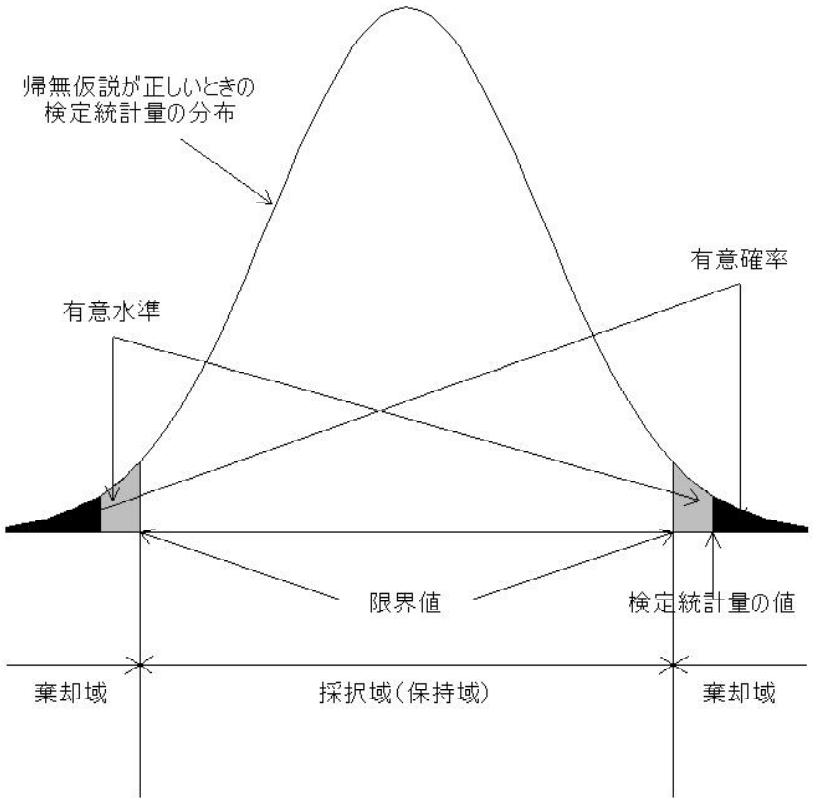
 # ｔ分布
dt(x, df)， pt(x, df)， **qt**(p, df)， **rt**(n, df)

 # 一様分布
dunif(x, min, max)， **punif**(x, min, max)， **qunif**(p, min, max)， **runif**(n, min, max)

 # ワイブル分布
dweibull(x, shape, scale)， **pweibull**(x, shape, scale)， **qweibull**(p, shape, scale)， **rweibull**(n, shape, scale)

非心分布は，非心パラメタncp=を指定すればよい.

統計的有意性検定の概念図を描く




```
par(mar=c(0,1,0,1))

ycep <- 0.2
xvals.a <- seq(-3,3, length=61)
dvals.a <- dnorm(xvals.a)+ycep
plot(xvals.a, dvals.a, type="l", ylim=c(0,0.65), axes=FALSE, ann=FALSE)
segments(-3,ycep, 3,ycep)

xvals.l <- seq(-3,-2, length=21)
dvals.l <- dnorm(xvals.l)+ycep
xvals.u <- seq(2,3, length=21)
dvals.u <- dnorm(xvals.u)+ycep
polygon(c(xvals.l, rev(xvals.l)), c(rep(ycep,21), rev(dvals.l)), col="gray")
polygon(c(xvals.u, rev(xvals.u)), c(rep(ycep,21), rev(dvals.u)), col="gray")

xvals.pl <- seq(-3,-2.3, length=8)
dvals.pl <- dnorm(xvals.pl)+ycep
xvals.pu <- seq(2.3,3, length=8)
dvals.pu <- dnorm(xvals.pu)+ycep
polygon(c(xvals.pl, rev(xvals.pl)), c(rep(ycep,8), rev(dvals.pl)), col="black")
polygon(c(xvals.pu, rev(xvals.pu)), c(rep(ycep,8), rev(dvals.pu)), col="black")

segments(-2,ycep, -2,0)
segments(2,ycep, 2,0)
arrows(-2,ycep, -0.5, ycep-0.05, code=1, length=0.1)
arrows( 2,ycep,  0.5, ycep-0.05, code=1, length=0.1)
text(0, ycep-0.05, "限界値")
arrows( 2.3,ycep,  2.3, ycep-0.05, code=1, length=0.1)
text(2.3, ycep-0.05, "検定統計量の値")

arrows(-2.2, ycep+0.025, -2.2, ycep+0.15, code=1, length=0.1)
arrows( 2.2, ycep+0.025, -2.2, ycep+0.15, code=1, length=0.1)
text(-2.2, ycep+0.17, "有意水準")
arrows(-2.5, ycep+0.01, 2.5, ycep+0.19, code=1, length=0.1)
arrows( 2.5, ycep+0.01, 2.5, ycep+0.19, code=1, length=0.1)
```

```
text(2.5, ycep+0.21, "有意確率")

arrows( -1, ycep+0.25, -1.7, ycep+0.3, code=1, length=0.1)
text(-2, ycep+0.34, "帰無仮説が正しいときの")
text(-2, ycep+0.32, "検定統計量の分布")

arrows(-2, ycep-0.1, 2, ycep-0.1, code=3, length=0.1)
arrows(-2, ycep-0.1, -3, ycep-0.1, code=1, length=0.1)
arrows( 2, ycep-0.1,  3, ycep-0.1, code=1, length=0.1)
text(0, ycep-0.12, "採択域（保持域）")
text(-2.5, ycep-0.12, "棄却域")
text(2.5, ycep-0.12, "棄却域")
```

いくつかの図をまとめてPDFファイルに出力

単回帰分析において，説明変数x，基準変数y，誤差e，予測値pとしたときの，次の６個の散布図を３行２列にならべて，PDFファイルに出力する．

x-y，p-y，x-e，p-e，x-p y-e

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",")
> head(d1)
  id stress kyoufu support utsu work result
1  1     20    2.2      17   18    0      0
2  2     23    4.8      18   21    1      0
3  3     30    5.8      12   29    1      1
4  4     25    5.2      18   29    0      1
5  5     26    2.0       8   22    1      0
6  6     21    5.0      26   19    1      0
>
>
> # 記述統計量
> dtmp <- d1[,c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp),2)
> colnames(ktmp) <- c("N","Mean","SD",colnames(ctmp))
> ktmp
          N  Mean   SD stress kyoufu support  utsu  work result
stress  245 22.94 5.25   1.00   0.40   -0.34  0.62  0.03   0.44
kyoufu  245  4.05 1.17   0.40   1.00   -0.03  0.31  0.10   0.20
support 245 18.42 4.96  -0.34  -0.03    1.00 -0.51 -0.03  -0.39
utsu    245 20.29 6.49   0.62   0.31   -0.51  1.00  0.02   0.76
work    245  0.50 0.50   0.03   0.10   -0.03  0.02  1.00  -0.08
result  245  0.26 0.44   0.44   0.20   -0.39  0.76 -0.08   1.00
>
>

> # 単回帰分析
> reg.1 <- lm(utsu ~ stress, data=d1)
> summary(reg.1)

Call:
lm(formula = utsu ~ stress, data = d1)

Residuals:
     Min       1Q   Median       3Q      Max
-11.6277  -3.2181  -0.0374   3.0771  15.6674

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.73611    1.46777   1.864   0.0635 .
stress       0.76506    0.06238  12.264   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.114 on 243 degrees of freedom
Multiple R-squared:  0.3823,    Adjusted R-squared:  0.3798
F-statistic: 150.4 on 1 and 243 DF,  p-value: < 2.2e-16

>
>
> #偏回帰係数の信頼区間
> confint(reg.1)
                2.5 %    97.5 %
(Intercept) -0.1550688 5.6272812
stress       0.6421862 0.8879398
>
>
```

	id	stress	kyoufu	support	utsu	work	result
2	1	20	2.2	17	18	0	0
3	2	23	4.8	18	21	1	0
4	3	30	5.8	12	29	1	1
5	4	25	5.2	18	29	0	1
6	5	26	2	8	22	1	0
7	6	21	5	26	19	1	0
8	7	14	2.2	24	12	0	0
9	8	22	4.4	17	19	1	0
10	9	26	4.2	11	27	0	1
11	10	26	4.2	18	18	1	0
12	11	21	2	27	18	0	0
13	12	24	4.8	19	29	1	1
14	13	24	3.4	23	19	1	0
15	14	23	2.2	10	24	1	0
16	15	23	4.4	24	20	0	0
17	16	35	7.2	12	31	0	1
18	17	25	3.2	17	19	1	0
19	18	33	3.4	14	26	0	1
20	19	30	4.4	20	30	1	1
21	20	19	5.6	18	12	1	0

```
> #x,y,e の 散布図をPDFに出力

> pdf("残差プロット.pdf", paper="a4", width=7, height=15, family="Japan1")
> layout(matrix(c(1,2,3,4,5,6), 3, 2, byrow=T))      # ７×15インチの中に６個の図を3行2列に配置
>
> plot(d1$stress, d1$utsu, xlab="x", ylab="y",ylim=c(0,40),pch=20, las=1, main="x vs. y")
>
> plot(reg.1$fitted.values, d1$utsu, xlab="predicted y", ylab="y",
+  ylim=c(0,40),pch=20, las=1, main="predicted y vs. y")
>
> plot(d1$stress, reg.1$residuals, xlab="x", ylab="residual", pch=20, las=1,
+  main="x vs. e")
>
> plot( reg.1$fitted.values, reg.1$residuals, xlab="predicted y",
+  ylab="residual", pch=20, las=1, main="predicted y vs. e")
>
> plot(d1$stress, reg.1$fitted.values, xlab="x", ylab="predicted y",
+  ylim=c(0,40),pch=20, las=1, main="x vs. predicted y")
>
> plot(d1$utsu, reg.1$residuals, xlab="y", ylab="residual", pch=20,las=1,
+  main="y vs. e")
>
> layout(1)
> dev.off()
null device
          1
```

出力されたファイル