

# RjpWiki アーカイブス

## 【NA,NaN,NULL,Inf (06.07.06)】

NULL (何も無い), Na (欠損値), NaN (非数), Inf (無限大) についてのネタを集めました.

### 1 NA, NaN, NULL, Inf

NULL (何も無い), Na (欠損値), NaN (非数), Inf (無限大) については, これらの値にどのような演算を施しても, 大抵はそのままの値 (NA や NaN) が返ってくる. すなわち, 原則として NaN にどのような演算を施しても結果は NaN になる. よって, 比較演算子 == すら使えないことになる.

```
x <- c(1.0, NA, 3.0, 4.0, 5.0)    # NA がある要素はどれかを調べようとしているの  
# ...  
x == NA                          # NA に対する演算は全て NA になってしまう  
[1] NA NA NA NA
```

これら 4 つの値の検査を行う関数がそれぞれ用意されている.

```
is.null()      : NULL  
is.na()       : NA  
is.nan()      : NaN  
is.finite()   : 有限か否か  
is.infinite() : 無限か否か
```

例えば, ある値が NA かどうかのテストは関数 `is.na()` で行なう (比較演算子 `==` では行なえないことに注意) が, これは NaN を代入しても TRUE が返ってしまう. そこで NaN か否か (NA か否かではない) を判定するために `is.nan()` という関数も用意されている.

```
is.na(NA)  
[1] TRUE  
is.nan(NA)  
[1] FALSE  
x <- c(1, NA, 3, 4)  
is.na(x)  
[1] FALSE TRUE FALSE FALSE
```

## 2 ベクトル要素にある NA の排除・置換

引数に NA が含まれるとエラーになる関数がある。このような関数に NA が含まれるデータを渡す場合の対処法としては以下のようなものがある。

```
( a <- c(1,2,NA,4,5,NA) )  
[1] 1 2 NA 4 5 NA  
a[!is.na(a)]                # NA を単に取り除く  
[1] 1 2 4 5  
a  
[1] 1 2 NA 4 5 NA  
a <- ifelse(is.na(a), 0, a)  # NA を 0 に置き換える  
a  
[1] 1 2 0 4 5 0
```

NA を置き換えるのではなく、NA を要素に持つベクトルから、NA を読み飛ばして要素を読み取することを考える。この場合は以下の様にすればよい。応用すれば『欠損値と負のデータを除去してデータを読み取る』ことも出来る。

```
x <- c(1, 2, NA, -4, 5, -6, 7, NA, 9, 10)  
y <- x[!is.na(x)]                # NA 以外の x の要素を y に格納  
y  
[1] 1 2 -4 5 -6 7 9 10  
x <- c(1, 2, NA, -4, 5, -6, 7, NA, 9, 10)  
y <- x[(!is.na(x)) & (x > 0)]    # NA と負の値以外の x の要素を y に格納  
y  
[1] 1 2 5 7 9 10
```

## 3 NULL の使い方の例

NULL は属性を取り出す関数が属性値がないときの値として返すなど『適当な結果がない』ことを示すために使われる。NULL と欠損値 NA を混同してしまいがちだが、NA はベクトルの要素となり得るが、NULL はベクトルなのでそれ自身でベクトルの要素にはならない点異なる。

```
mode(NULL)                    # モード "NULL" を持つ  
[1] "NULL"  
length(NULL)                  # 長さが 0 のベクトルである  
[1] 0  
( a <- 1:3 )  
[1] 1 2 3  
names(a)                      # a には names 属性が付いていないので NULL が返る  
NULL
```

上の手順を逆手にとって、属性値に NULL を代入して属性値を除去することができる。また、繰り返し文で処理を行うオブジェクトの初期値にも NULL が使われたりする。

```
names(a) <- c("a", "b", "c")
a
a b c
1 2 3
names(a) <- NULL
a
[1] 1 2 3
names(a)
NULL
x <- NULL # 長さ 0 のベクトル x を用意して
for (i in 1:10) x <- append(x, i*i) # x に 1^2, 2^2, 3^2 ... を順に格納する
x
[1] 1 4 9 16 25 36 49 64 81 100
```

NULL を長さが 0 の数値型ベクトルや文字型ベクトルと間違わないように注意。これらは型が異なるため、異なるオブジェクトとなっている。as.character, as.numeric などの型変換関数を使って NULL をある型のベクトルに変換すると、指定された型の長さが 0 のベクトルになる。

## 4 数値積分での Inf の使用例

Inf (無限大・ $\infty$ ) は、数値積分 integrate() を行う際に使うことができる。

```
myfunc <- function(x) x^2 # 関数を定義してから積分する
integrate(myfunc, 0, 1)
0.3333333 with absolute error < 3.7e-15
integrate(dnorm, -Inf, 1.96) # 無限を範囲に取ること (広義積分) も可
0.9750021 with absolute error < 1.3e-06
```