

Rにおける特殊命令

飯島勇人*

2007年5月7日

目次

1	for (繰り返し命令)	1
1.1	基本的な使い方	1
1.2	実用的な使い方	2
2	ifelse (条件分岐)	3
2.1	基本的な使い方	3
2.2	実用的な使い方	4

1 for (繰り返し命令)

for() を使うと、繰り返し命令を出すことができます。内容としては、少々高度な部類に入りますが、少し複雑なデータ操作には欠かすことができない関数なので、ぜひ覚えてください。

1.1 基本的な使い方

for は、規定回数まである一連の操作を繰り返すのに用います。たとえば、

```
> temp <- numeric(0) #データを作っておきます
> for (i in 1:10) {
  temp <- c(temp, rep(i, 10))
}
```

と書いてみましょう。すると、1 が 10 個、2 が 10 個...10 が 10 個、と並んだ数列を作成することができます。ちなみに

```
> rep(1:10, 10)
```

* 北海道大学大学院農学院専門研究員

とすると、1、2...10、1、2...10... が 10 回繰り返される数列ができます。

より一般的な形で書くと、以下のようになります。

```
> for (i in 1:n) {  
ここに繰り返させたい動作内容  
}
```

最初のカッコ内の意味は、i という文字に、1 から n まで数字を入れる、という意味である。そして、i に最初の数字 (この場合は 1) が投入されると、繰り返させたい動作内容を行い、終了後、今度は i に 2 が投入され、繰り返させたい内容が行われ... というようにして、最終的に n が投入されるまで動作が続きます。

つまり、単なる同じ動作の繰り返しなのですが、例えば動作を行わせる対象が、1 列目、2 列目、というように、次々と進んでいく場合に、有効な方法です。

1.2 実用的な使い方

for 文は、複数のデータフレーム (あるいは split() で分割したデータフレーム) に連続的に操作をさせるときに力を発揮します。例えば、以下のようなデータフレームがあったとします (実際に実在する何かと名称がとても似ているかもしれませんが、それは気分的な問題です。また、データは乱数で発生させているので、関係はまったくありません ^^ ;)

```
> Team <- rep(c("Figthers", "Hawks", "Lions", "Marines", "Buffaloes", "Eagles"), 10)  
> Rank <- rep(1:6, 10)  
> Practice <- runif(60, 5, 8)  
> Benchpress <- rnorm(60, 80, 20)  
> Hitrate <- (Practice)/3 + rnorm(60, 0.5, 0.2)  
> data <- data.frame(Team, Rank, Practice, Benchpress, Hitrate)
```

例えば、チームごとに Practice と Hitrate の散布図を見たいとします。通常であれば、チームごとにデータを分けたあと、チームごとに散布図を描かなければなりません。for を使うと、楽ができます。

```
> seperate <- split(data, data$Team)  
> par(ask=T) #図を描くときに、確認を求めさせる設定  
> for (i in 1:nlevels(data$Team)) {  
    plot(seperate[[i]]$Practice, seperate[[i]]$Hitrate)  
}  
> par(ask=F)
```

ちなみに、

```

> seperate <- split(data, data$Team)
> par(mfrow=c(3, 2))
> for (i in 1:nlevels(data$Team)) {
  plot(seperate[[i]]$Practice, seperate[[i]]$Hitrate)
}

```

とすれば、一つの画面に全て表示できます。

また、カテゴリーごとに分けたデータに関して、複雑な処理を行うときにも for は有効です。例えば、チームごとに、全列の平均値を出し、全チームのデータをひとまとめにする作業を行います。

```

> seperate <- split(data, data$Team)
> temp <- NULL
> for (i in 1:nlevels(data$Team)) {
  temp <- rbind(temp, mean(seperate[[i]][,-1]))
}
> temp <- as.data.frame(temp)
> result <- cbind(levels(data$Team), temp)

```

チームごとに、チーム内での最大打率との差を計算する場合はこんな感じです。

```

> seperate <- split(data, data$Team)
> temp <- NULL
> for (i in 1:nlevels(data$Team)) {
  seperate[[i]]$Dif <- max(seperate[[i]][, 5]) - seperate[[i]][, 5]
  temp <- rbind(temp, seperate[[i]])
}
> result <- as.data.frame(temp)
#上位のチーム順にするなら
> result <- result[order(result$Rank), ]

```

2 ifelse (条件分岐)

ifelse() は、ある条件をつけて、その条件に当てはまる場合はこうする、当てはまらない場合はこうする、という動作をさせる関数です。

2.1 基本的な使い方

```

> ifelse(条件式, 当てはまる場合, 当てはまらない場合)

```

のようにして使います。

2.2 実用的な使い方

NA を何か別な値に変換するとき

```
> Team <- rep(c("Figthers", "Hawks", "Lions", "Marines", "Buffaloes", "Eagles"), 10)
> Benchpress <- c(rnorm(20, 80, 20), NA, rnorm(20, 80, 20), NA, rnorm(18, 80, 20))
> data <- data.frame(Team, Benchpress)
> mean(data$Benchpress)
[1] NA #NA が返されてしまう
> data$Benchpress <- ifelse(is.na(data$Benchpress), 0, data$Benchpress) #NA なら 0 を、そうでなければ下のデータを挿入
> mean(data$Benchpress) #今度は平均値が表示される
```