

## Rにおける自分で作成した関数の使い方<sup>†</sup>

Rでは既存の関数のみならず、自分で作成した関数を使用する事もできます。実際、より高度な分析をする際には、既存の関数には用意されていないような計算をしなければいけない場合が多々あり、その場合は自分で関数を作成しなければなりません。これは関数を定義するとも言われます。以下「関数を作成する」と「関数を定義する」は同じことを意味しているとします。この「自分で関数を定義する」という作業は、複雑な計算になるとけっこう大変です。そのような関数をRを起動する度に、いちいち作成し直さないといけないとするならば、不便な事この上ないでしょう。自分で1度作成したら、後々も繰り返し使用したいというのが人情です。Rでは自分で一度作成した関数を保存し、再びRを起動したときに使用する事ができるようにできます。以下その手順を解説します。

### 関数の作成、保存、再利用

関数の作成方法については、後程もう少し詳しく述べるとして、まず作成し、保存し、再使用する手順について述べます。

例えば R-Console 上で

```
> testfunc = function(x, y, z){x+y+z}
```

と入力すると、関数 `testfunc()` が定義され、以後この関数を使用する事ができます(なお `=` の代わりに `<-` を用いることもできます)。例えば

以下のように入力してみると

```
> testfunc(1,2,3)
```

```
[1] 6
```

のように出力されます。`testfunc(x, y, z)` が `x+y+z` を計算し出力している事がわかります。このように関数を自作する場合は、関数名について既存のRの関数と同じ名前を付けないように注意する必要があります。間違っても同じ名前を付けてしまうと既存の方の関数が使えなくなります。この場合後述の `rm()` 関数によって間違っても作った関数を削除する必要があります。

この関数は作業スペースを保存しておけば、再起動の際にそのまま使用する事ができます。しかしながら、(作業スペースを保存せずに)Rを起動しなおすと、この関数はもう一度作成しなおさないと使用できなくなります。せっかく作った関数ですので、後々繰り返し使用したいと思うのは当然でしょう。そのような場合は `source()` 関数を用いる事によって、自作の関数を読み込み、読み込んだ関数のみ再度 R 上で使用できる様にすることができます。以下その手順を述べましょう。

---

<sup>†</sup>この資料は私のゼミおよび講義でRの使用法を説明するために作成した資料です。ホームページ上で公開しており、自由に参照して頂いて構いません。ただし、内容について、一応検証してありますが、間違いがあるかもしれません。間違いがあった場合でもそれによって生じるいかなる損害、不利益について責任は負いかねますのでご了承下さい。

まず今度は R Console 上ではなく新しく R のスクリプトファイルというのを作成します。「ファイル」→「新しいスクリプト」とクリックしましょう。すると「無題 - R エディタ」と表示されたウィンドウが表示されます。ここに先ほどの `testfunc` の定義を書き、「ファイル」→「別名で保存」とクリックし、ファイルの種類のところを「R files (\*.R)」にして適当なフォルダに保存しましょう。例えば `testfunc` という名前で保存します(`testfunc.R` というファイルが保存されます)。

さて、保存したら、このファイルを R の作業ディレクトリへ移しましょう(もしくは最初から作業ディレクトリに保存してもよいでしょう)。あとは R Console 上で

```
> source("testfunc.R")
```

とすれば、`testfunc` を先ほどと同様に、使用する事が出来るようになります。またこれは R 上で「ファイル」→「R コードのソースを読み込む」によって `testfunc.R` を読み込んで同じです。

また実は作業ディレクトリにファイルを移動させなくても、パス(ファイルが保存してあるフォルダの事)を指定してあげれば `source()` 関数で読み込めます。例えば先ほどの `testfunc.R` を C ドライブ上の `Rfiles` というフォルダに保存したとしましょう。この場合

```
> source("C:/Rfiles/testfunc.R")
```

によって読み込むことができます。

ちなみに自分で作成した関数は `rm()` 関数によって削除する事ができます。例えば

```
> rm(testfunc)
```

とすると以後は `testfunc(1,2,3)` などと入力しても

```
> testfunc(1,2,3)
```

エラー: 関数 "testfunc" を見つけることができませんでした

のように表示されます。このまま作業スペースを保存すると、この関数は再定義しない限り使用不可能になります。これは例えば、すでに R に組み込まれている関数を同じ関数名を間違えて使用して新しく自作関数を定義してしまった場合など、自作関数を消去したい場合などに必要になります。

### 複数の自作関数を使用したい場合

複数の自作関数を使用したい時は上記の作業を繰り返してもよいですし、もしくは 1 つのファイルに 2 つの関数の定義を書いておけばそのファイルを読み込んだ時に、その 2 つの関数は一度に読み込まれます。例えば「ファイル」→「スクリプトを開く」で先ほどの `functest.R` を開いて 2 行目に

```
testfunc2 = function(x,y,z){x+y-z}
```

と書き込み `testfunc2` を定義しましょう。このファイルを「ファイル」→「保存」によって保存すると、

`source("testfunc.R")`を読み込むと一度に `testfunc` と `testfunc2` という2つの関数を読み込んでくれます。一つ一つの関数の定義が長い場合は別々のスクリプトファイルに、短い場合はこのようにまとめておいた方がよいでしょう。

また、少し手間がかかりますが、使いようによっては便利なやり方として、特定のフォルダに置いてある R のスクリプトファイルを R の起動時に自動的に読み込むようにすることができます。例えば、メモ帳を開いて、`source("C:/Rfiles/testfunc.R")` と書き込み、そのファイルを「ファイル」→「名前を付けて保存」で「ファイルの種類」を「すべてのファイル(\*.\*)」にし、「.Rprofile」(最初にドットがついていることに注意)と名前を付けて、R の起動ディレクトリに保存します(R の起動ディレクトリは起動直後に `getwd()` 関数によって調べられます)。すると以後は R を起動すると自動的に `.Rprofile` を読み込み、そこに書かれてある `source("C:/Rfiles/testfunc.R")` を実行してくれるので上記のコマンドをわざわざ R Console 上で実行しなくても `testfunc()` 関数が実行できるようになります。この `.Rprofile` ファイルを編集するには R を起動し「ファイル」→「スクリプトを開く」にしファイル名の横の部分を「All files (\*.\*)」にして `.Rprofile` を読み込んで行います。

### 自作関数の中で自作関数を使用する場合

自作の関数の定義の中で自作の関数を使用する事もできます。ただし使用するにはすでにそれらの自作関数が R に読み込まれていなくてははいけません。例えば新しく

```
> testfunc3 = function(x, y, z) {testfunc(x, y, z)+testfunc2(x, y, z)}
```

と定義すると、`testfunc()` 関数と `testfunc2()` 関数が読み込まれていれば、

```
> testfunc3(1, 2, 3)
```

```
[1] 6
```

となり使用する事ができます。もちろんこの場合もこの `testfunc3()` 関数を R のスクリプトファイルに書き込んで保存できます(例えば先ほどの `testfunc.R` ファイルの 3 行目に書き込んで保存すればこのファイルを読み込めばこれ 3 つの関数が一度に使用できるようになります。

さらに自作の関数の定義の中に既存の関数を使用する事もできます。ただしパッケージなどの関数を使用するにはすでにそれらの関数が R に読み込まれていなくてははいけません。

### 少し特殊な R による関数

R では関数を実行した結果として、計算結果だけではなくて、新たな別の関数を返すようにする事もできます。これはやや複雑ですが、引数が複数あるような関数に関して、ある引数が与えられたもとで、他の引数についてその関数の最適化を行いたい時などに非常に便利であるのでここで簡単に説明しておきましょう。

例えば

```
> tfunc = function(x, y) {x+y}
```

という関数を定義すると、これは引数として  $x$  と  $y$  を持ち

```
> tfunc(2,3)
```

とすると、上記関数の定義の  $\{\}$  内の計算結果として 5 を返してきます。それではこの  $\{\}$  内に新たに関数を定義したらどのようなになるでしょうか？具体的には

```
> tfunc2 = function(x,y){function(a,b){a*x+b*y}}
```

などとしたらどのようなになるでしょうか？この時、関数 `tfunc2` に 引数として  $x=2$  と  $y=3$  を与える、すなわち

```
> tfunc2(2,3)
```

とすると、これは何を返してくるのでしょうか？実はこれは、`tfunc2` の定義において  $x=2$ ,  $y=3$  とした時の  $\{\}$  の中身、すなわち  $a, b$  を引数とした新たな関数  $2a+3b$  を返してきます。つまり、`tfunc2(2,3)` は  $2a+3b$  を計算する関数という事です。よって、この関数に例えば引数として  $a=1, b=1$  を与えてやると

```
> tfunc2(2,3)(1,1)
```

```
[1] 5
```

のように  $2a+3b$  を計算した値が返ってきます。これは `tfunc2(2,3)` というのがあたりに  $2a+3b$  を計算する関数になっていると見なせるという事ですし、もしくは `tfunc2` を  $f(a, b, x, y) = ax + by$  のような関数とみることができ、`tfunc(x,y)(a,b)` で  $a*x+b*y$  を計算していると思なす事もできます。

このようにして自作の R 関数を活用すればより高度な分析が可能になり分析の幅が飛躍的に広がるでしょう。ぜひ関数を自作し、活用して下さい。