kadai

kamiyama

2019/2/1

R初心者が作る初心者マニュアル

自身のファイルを読み込み3群比較（1要因分散分析・対応なし）を行うまでを解説

初心者にとってRは敷居が高いと感じる。 その理由として、融通が利かず一定の知識がないと何もできないためと思われる。 エクセルではなんとなく操作しても大抵の関数を使用することができるが、Rでは準備が必要となる。 しかし、その準備は決して高度な知識を要求されるわけではない。イメージとしては

必要なものを揃える

決まった型にはめ込む

だけであるといえる。 今回手持ちのデータを3群比較するまでに必要な最低限の知識を説明する。なお、統計についての詳しい説明は割愛させていただく。

①　R・Rstudioのインストール

Rのインストールは特に難しいことはないため省略。 RstudioはRの入力処理、結果表示やパッケージのインストールを容易にしてくれるため、同時にインストトールすべき。

②　Rstudioについて

Rstudioを起動すると左に[Console]、右上に[Environment]、右下に[Files]が表示される。 入力はconsoleで行うと即時に結果を返してくれるが、入力した内容の訂正や保存を考えると、Scriptで行う方がベターと思われる。 [Script]は左上のFileボタンのすぐ下▼をクリックすると最上段にRscriptがあるのでこれをクリック。 するとconsoleの上段に[script]が表示され、4画面となる。 scriptでは終了時にsaveを選択すれば内容がそのまま保存されるため便利である。

③　ディレクトリー

初心者にとっての関門の一つがディレクトリーであるだろう。言い換えれば作業しているフォルダのことである。 Rでは作業ディレクトリーにあるファイルしか読み込まないため、ディレクトリーを意識する必要がある。 後で変更することも可能だが、初心者的には初めに指定し、そこへ必要なファイルを移動させる方がベターと思われる。 左上のFileをクリックし、New projectをクリックし、R専用のフォルダを作っておけばわかりやすいだろう。 なお、現在のディレクトリーを確認する方法は

```
getwd()
```

```
## [1] "C:/Users/kamiy/Desktop/R folder/project/project1/report"
```

私の場合はデスクトップにRフォルダがあり、その中のprojectフォルダの中のproject1フォルダの中のreportフォルダがこのディレクトリーである。 ＊もちろん人によって違う

④パッケージのインストールとその使用

初心者にとって2つ目の関門がパッケージである。今回は必要ないため時間がなければ飛ばして構わない。 パッケージは言わばアプリのようなものである。 慣れればなんでもないが、初めは混乱しうる。 Rをインストールすれば全て揃ったと勘違いしてしまうが、そうではないため

必要なpackageを適宜インストール

してくる必要がある。さらに、

インストールするだけではまだ使えない。

これも混乱を招く。使用すると宣言する必要がある。 いずれもRstudioであればコンソールで入力しなくても、[Files]ビューのpackagesをクリックすると左上にinstallがあるので, これをクリックしてパッケージ名を入力すればインストールしてくれる。さらに、インストールするとlibraryに追加されるので、 クリックして□にチェックが入れば使用できるようになる。 一応、描画編集用のパッケージであるggplot2を使用宣言する場合は

```
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

⑤関数について

すでに先ほどlibrary関数を使用している。libraryは関数名(”package名”)で作動する。関数によって()内に入る要素（引数と呼ぶ）が違ってくるが、逆に言えば、()内を揃えるだけで作動させることができる。今回の統計で使用する関数はoneway.test関数とTukeyHSD関数である。それぞれ

oneway.test(変数ベクトル~因子群分類)

TukeyHSD(aov(変数ベクトル~因子群分類))

である。つまり、データを読み込んだのち変数ベクトルを作成し、群を因子に分類するだけで実行可能であるが、そのためにはデータの種類の理解が必要である。

⑥データの種類

初心者にとってとっつきにくいと感じるのがここである。しかし、これはこういう決まりとして覚えるしかない。 まずはベクトル。ベクトルは同じ型（文字なら文字、数値なら数値）を複数持っているものである。 例えば

```
A<-c(1,2,3,4,5,6)
B<-c("a","b","c","d","e","f")
A
```

```
## [1] 1 2 3 4 5 6
```

```
B
```

```
## [1] "a" "b" "c" "d" "e" "f"
```

Aは実数(numeric)のベクトルであり、Bは文字列(character)のベクトルである。他にも理論値（logical)や複素数(complex)などがあるが、 今回は使わない。 <-は代入するという意味であり、c()は,で区切った複数の値を持たせることができる。 ＊大文字と小文字は別ものとして扱われるので注意。 ""で囲むと文字と認識されるため、数字であっても文字として扱うこともできる。例えば小文字の c に100（数値）大文字のCに"１００"(文字)を入力してみると

```
c<-100
C<-"100"
c+1
```

```
## [1] 101
```

大文字のC+1はエラーとなる。データの型が合っているかを調べるには

```
is.numeric(c)
```

```
## [1] TRUE
```

```
is.character(c)
```

```
## [1] FALSE
```

```
is.character(C)
```

```
## [1] TRUE
```

なお、大文字のCの文字を数値に変換すれば計算も可能である。

```
as.numeric(C)+1
```

```
## [1] 101
```

少し話が脱線したが、型を間違えるとエラーとなることを頭に入れておかないといけない。

続いて、クラスについて最低限説明する。 クラスは行列(matrix)や因子(factor)、データフレーム（data.frame)などを含む。 行列は同じ種類の値を持つ2次元のデータ構造であり、例えば先ほどのベクトルA,Bを使うと

```
matrix(A,2,3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
matrix(B,3,2)
```

```
##      [,1] [,2]
## [1,] "a"  "d"
## [2,] "b"  "e"
## [3,] "c"  "f"
```

ご覧の通り、matrix関数はmatrix(ベクトル,行数,列数)を指定すれば作動する。正確にはnrow=で行数,ncol=で列数を指定するのだが数値だけでも可能である。

続いて因子(factor)について。因子は深追いすると混乱するので簡単に説明するにとどめる。因子は要素を整数値（内部値）に置き換えて管理することである。factor関数はfactor(ベクトル、内部値、表示名）であるが、今回は単純に3群に分類するだけであるので、内部値、表示名はデフォルトを使用してベクトルだけ入力する。今回使用するデータは１群に20例ずつの60例のデータであるので、これをF1,F2,F3群に分類すると

```
fac<-factor(c(rep("F1",20),rep("F2",20),rep("F3",20)))
fac
```

```
##  [1] F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F1 F2 F2 F2
## [24] F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F2 F3 F3 F3 F3 F3 F3
## [47] F3 F3 F3 F3 F3 F3 F3 F3 F3 F3 F3 F3 F3 F3
## Levels: F1 F2 F3
```

rep関数はrep(A,B)でAをB回繰り返すである。 ちなみに内部値を見るには

```
as.integer(fac)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [36] 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

つまり、F1,F2,F3をコンピューターが勝手に1，2，3という内部値で管理していることがわかる。

最後にデータフレームであるが、名前を持った行列であり、普段エクセルに入力したデータなどはそれにあたる。気を付けることとしては、 行、列の長さがすべて同じである必要がある。ではいよいよデータの取り込みである。

⑦データの取り込み

初心者にとっての最大の関門と思われる。取り込んだデータを操作することも可能だが、初心者にはハードルが高いため、例のごとく型にはめ込む形で行いたい。まずはエクセルで必要な3群のデータが縦に並んだ表を作成する。一行目は群の名前を入力（なんでもよい）。続いて名前を付けて保存からテキスト(タブ区切り)を選び、名前をtest.txtとする。そしてそのファイルをRの作業ディレクトリーに移動させる。その後データフレームを取り込む関数であるread.tableを入力。

```
test<-read.table("test.txt",header=T)
test
```

```
##    F1 F2 F3
## 1  65 90 85
## 2  60 75 65
## 3  75 70 90
## 4  80 90 75
## 5  65 65 90
## 6  60 70 75
## 7  70 80 85
## 8  85 85 75
## 9  65 70 95
## 10 75 70 75
## 11 75 85 80
## 12 70 75 75
## 13 80 80 85
## 14 75 90 75
## 15 80 80 85
## 16 80 90 90
## 17 65 85 90
## 18 70 85 80
## 19 75 75 75
## 20 80 75 75
```

例ではF1,F2,F3群それぞれ20人ずつがテストを受けた結果のデータフレームをtestに取り込んだ。なお、一行目に列名をつけなければheader=Fにすればよい。また、一列目に行名があればcol.names=1と入力すればよい。＊日本語で入力した場合、文字化けを起こす可能性がある。Windowsであればそれを避けるにはfileEncording="CP932"を指定しておくとよい。むしろ日本語は使用しない方が無難かもしれない。 データの取り込みが無事終了すればもうできたも同然である。

⑧oneway.test関数で一要因分散分析を実施

3群の比較に対して t 検定を繰り返してはいけない。そこで今回、「抽出された標本が母平均の等しい3群から抽出される可能性が高いか」を検討するために行われる分散分析を使用する。oneway.test関数は平均の等価性に対するF検定が実施され、検定統計量Fの実現値に対応するp値も一緒に出力される。デフォルトでは分散の等質性が不要なvar.equal=FALSEとなっているが、等分散であればvar.equal=TRUEにする。 ⑤で述べた通り、oneway.testには変数ベクトルと群因子分類が必要である。 データフレームから変数ベクトルを作成するには抽出作業が必要である。以下のように

```
test2<-c(test$F1,test$F2,test$F3)
test2
```

```
##  [1] 65 60 75 80 65 60 70 85 65 75 75 70 80 75 80 80 65 70 75 80 90 75 70
## [24] 90 65 70 80 85 70 70 85 75 80 90 80 90 85 85 75 75 85 65 90 75 90 75
## [47] 85 75 95 75 80 75 85 75 85 90 90 80 75 75
```

test2には結果の点数がF1、F2,F3の順番で代入されている。これを用いて

```
oneway.test(test2~fac)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  test2 and fac
## F = 7.0415, num df = 2.000, denom df = 37.954, p-value = 0.002507
```

p値が0.002507であり、p<0.05である。よって帰無仮説「3群の母平均は等しい」という帰無仮説を棄却⇒3群の母平均は等しくないを採択できる。

⑨TukeyHSDで多重比較を行う。

3群の母平均は等しくないなら、どの群とどの群で得点平均に差があるかを知りたい。そこで可能な限りの組み合わせでp値を返してくれるTukeyHSD関数を使用する。先述の通り引数はoneway.testと同じであるので

```
TukeyHSD(aov(test2~fac))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = test2 ~ fac)
##
## $fac
##       diff        lwr       upr     p adj
## F2-F1 6.75  0.8998358 12.600164 0.0199934
## F3-F1 8.50  2.6498358 14.350164 0.0026126
## F3-F2 1.75 -4.1001642  7.600164 0.7528036
```

F1とF2,F1とF3でp<0.05であり、群間に有意差あり。一方F2とF3ではp=0.7528036であり有意差なしとなった。

⑩まとめ

以上、最低限必要でかつ初心者がつまづきやすいと思われるところを解説した。必要な関数を調べ、その決まった型にはめ込むことを意識すればRの強力な関数を使用することができる。今回説明を省略した部分も多く、可能なら入門書を一冊頑張ってやり遂げるともっと系統だった学習ができるだろう。Rを半年程度やってみた初心者としてRをこれから始める初心者へのアドバイスとしては、Rの膨大な知識があるにこしたことはないが、全て覚えようとすると挫折するので、最低限を理解しあとはその都度調べることを癖にするのがよいと思われる。