

R-入門

東葛飾高等学校 大橋 真也

あらすじ

前回の「 $\alpha-\omega$ 」で、 $\text{T}_\text{E}\text{X}$ について書きましたが、実はその原稿を書いていたときから気になっていたものがありました。それが今回紹介する「R」です。「R」は、統計処理のソフトウェアですが、使い方によっては様々な分野での活用が可能です。今回はこの「R」の入門編を書きます。したがって、 $\text{T}_\text{E}\text{X}$ の記事の連載化は早くも中断させていただきます。

1 Rとは何か

みなさんは、Rというソフトウェアを知っているでしょうか。Rは統計処理や(データを用いた)グラフィックスに使われるシステムです。統計ならば、ExcelやSPSSで十分だという人もいるかもしれません。しかしRはフリーウェア(正確に言えばGNU)であり、いくつかのコマンドを覚えるだけで、プログラミングを容易に行うことができます。また頻繁にバージョンアップを行っており、乱数発生などをはじめとして様々な部分で最新のアルゴリズムが使われていることが素晴らしいところです。

Rは現在、大学や研究機関などで統計分析などが必要な分野ではかなり多く使われています。つまりRは統計処理ソフトウェアとしては「本物」なのです。高等学校では統計処理に関して、学校で教える機会が減ってきています。しかしながら実際に大学で文科系の学生も含めて最も使われている数学は統計なのではないでしょうか。

ここではRを紹介するとともに、高等学校でのRの活用の可能性について触れていきます。

いと考えています。

2 Rのインストール

$\text{T}_\text{E}\text{X}$ のときのように詳細に書くつもりはありません。RはWindows, MacOSX, Linuxなどの様々なプラットフォームに対応したプログラムが配布されています。Windowsでは、Windows95以降であれば動作可能です。

<http://cran.md.tsukuba.ac.jp/>などのCRAN¹からダウンロードしてください。インストールは簡単です。なお2005年10月11日現在の最新のバージョンは2.2.0です。

インストール時に注意する点は2点。インストーラのコンポーネントの選択では、「Version for East Asian languages」も必ず選んでおかないと、メニューが化けます。もう一点は、最初に起動した際に「編集」→「GUIプリファレンス」でFontをMS Gothicなどに変更して、この結果を`\Program Files\R\etc\`にRconsoleとして保存しておくことを忘れないでください。

¹Comprehensive R Archive Network のことです。「包括的 R アーカイブ」?

これさえしておけば、次回以降は文字化けもなく普通に使えるはずです。

3 簡単な使い方

3.1 電卓

Rは電卓としても使うことができます。もちろん関数電卓程度の数学関数は揃っています。>というプロンプトに続けて式を書いて、Enterキーを押すだけです。答えの最初についている[1]は先頭が一番目の要素であることを表しています。

```
> 1+2*3^4
[1] 163
> sin(pi/6)
[1] 0.5
> sqrt(10)
[1] 3.162278
```

3.2 ベクトル

データなどの数列はベクトルと呼ばれています。下の例で、1:10は1から10までの数、c()は、combineの略で数列をまとめるときに使います。つまりa<-c(1:10)は、1から10までの数列をaに代入するという意味になります。

```
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> a<-c(1:10)
> a
[1] 1 2 3 4 5 6 7 8 9 10
> a^3
[1] 1 8 27 64 125 216 343
[8] 512 729 1000
> sin(a)
[1] 0.8414710 0.9092974 0.1411200
[4] -0.7568025 -0.9589243 -0.2794155
[7] 0.6569866 0.9893582 0.4121185
[10] -0.5440211
> a+1
[1] 2 3 4 5 6 7 8 9 10 11
```

3.3 行列の計算

下の例では、

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, B = \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}, C = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$$

について様々な計算を行っています。

行列の掛け算は%%を用いることに注意が必要です。*では、対応する成分同士の計算になってしまいます。

solve()は、逆行列を求めるときや連立方程式を解くときに使います。

eigen()は、固有値や固有ベクトルを求める関数です。

```
> a<-matrix(1:4,nrow=2)
> a
      [,1] [,2]
[1,] 1 3
[2,] 2 4
> det(a)
[1] -2
> solve(a)
      [,1] [,2]
[1,] -2 1.5
[2,] 1 -0.5
> b<-matrix(c(3,1,2,4),nrow=2,byrow=T)
> b
      [,1] [,2]
[1,] 3 1
[2,] 2 4
> a%*%b
      [,1] [,2]
[1,] 9 13
[2,] 14 18
> a*b
      [,1] [,2]
[1,] 3 3
[2,] 4 16
> c<-matrix(c(5,0))
> solve(b,c)
      [,1]
[1,] 2
[2,] -1
> eigen(b)
$values
[1] 5 2

$vectors
      [,1] [,2]
[1,] -0.4472136 -0.7071068
[2,] -0.8944272 0.7071068
```

3.4 求解・微分

単なる統計ソフトとあなどることなかれ。ニュートン法による求解も簡単にできます。本来ベクトルを表す c は、ここでは $c(0,2)$ は 0 から 2 までの範囲という意味です。6 回反復して、1.414213 という解が求まっています。

n 次方程式の解も求めることができます。ここでは $c(1,3,2)$ は $x^2 + 3x + 2 = 0$ の係数を意味しています。同様に $c(1,-4,1,-4)$ は、 $x^3 - 4x^2 + x - 4 = 0$ を意味しています。

微分は導関数を単に求める場合とあとで数値を代入する場合とでは、使う関数が異なっています。高等学校の教科書程度の微分ならば容易に計算できるでしょう。 $g(2)$ の結果表示されているものは、最初のものが x^3 に $x = 2$ を代入したもの。あとのものが導関数 $g(x)$ に $x = 2$ を代入した結果です。

```
> f<-function(x) x^2-2
> uniroot(f,c(0,2))
$root
[1] 1.414213

$f.root
[1] -6.855473e-07

$iter
[1] 6

$estim.prec
[1] 6.103516e-05

> polyroot(c(1,3,2))
[1] -0.5+0i -1.0-0i
> polyroot(c(1,-4,1,-4))
[1] 0.25+0i 0.00+1i 0.00-1i
> f<-expression(x^x)
> D(f,"x")
x^(x - 1) * x + x^x * log(x)
> g<-deriv(~ x^3,"x",func=T)
> g(2)
[1] 8
attr(,"gradient")
      x
[1,] 12
```

ここでは誌面の関係で紹介しませんが、数値積分も簡単にできます。

4 簡単な統計解析

もちろん統計解析に関する関数は充実しています。また多くのパッケージがあり、それらを使うとかなり専門的なこともできます。次の例は高等学校程度(?)の統計です。行頭の+は 1 行目を途中で終了すると、複数行を入力できることを促すプロンプトです。

```
> data<-c(3470,2550,2920,2530,3280,2840,2520,3350,
+ 3610,3430)
> mean(data)
[1] 3050
> sd(data)
[1] 426.8229
> quantile(data)
      0%      25%      50%      75%     100%
2520.0 2622.5 3100.0 3410.0 3610.0
> t.test(data,mu=3240)

One Sample t-test

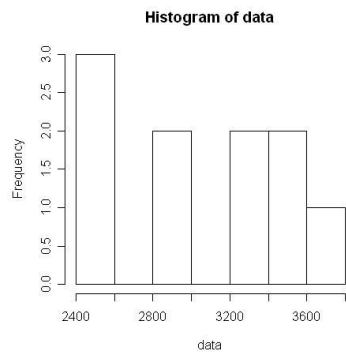
data: data
t = -1.4077, df = 9, p-value = 0.1928
alternative hypothesis: true mean is not equal to
3240
95 percent confidence interval:
 2744.669 3355.331
sample estimates:
mean of x
 3050
```

5 グラフィックス

グラフィックス機能が充実していることも R の特長です。統計に関する分布関数や棒グラフ円グラフなどかなり複雑なものを描けるだけでなく、普通に関数のグラフも描くことができますので、統計以外の分野でも活用することが可能です。

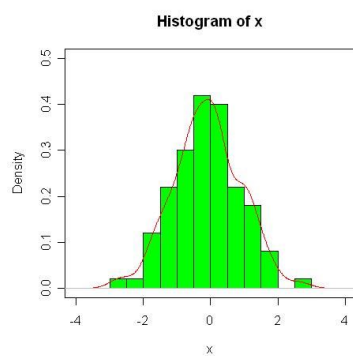
次の図は先ほどのデータを元にヒストグラムを描いたものです。

```
> hist(data)
```



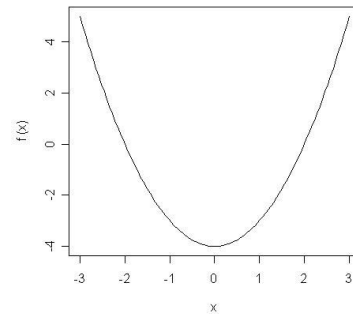
グラフの重ね合わせも可能です。またデータから連続な密度関数を描かせることもできます。色を指定していますが、誌面の関係上よく分からないかもしれません。

```
> x<-rnorm(100)
> hist(x,xlim=c(-4,4),ylim=c(0,0.5),prob=T,
+ col="green")
> par(new=T)
> plot(density(x),xlim=c(-4,4),ylim=c(0,0.5),
+ col="red",xlab="",ylab="",main="")
```



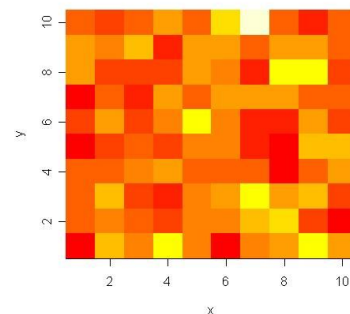
普通に関数のグラフも描くことができます。`function()`で、事前に関数を定義しておくことがコツです。`plot(sin,0,2*pi)`などすれば正弦曲線が描けます。もちろん様々なオプションも指定できますので、教材作成にも活用できるでしょう。また作ったグラフは様々な形式で出力することが可能です。

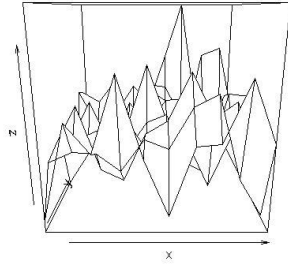
```
> f<-function(x)x^2-4
> plot(f,-3,3)
```



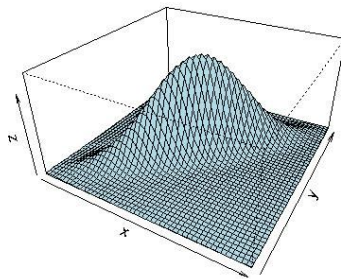
次の例は、正規乱数を 100 個とり、これを 10×10 の行列にして、その行列のイメージと 3D グラフを描きました。`image()` は行列のイメージを表示する関数です。面白い関数ですね。3次元のデータを作れば、`persp()`で3次元のグラフにすることが可能です。つまり数学的な関数データでなくとも地形データなどもグラフを作ることが可能になります。

```
> x<-y<-1:10
> z<-matrix(rnorm(100),ncol=10)
> image(x,y,z)
> persp(x,y,z)
```





もちろんそのようにデータを計算させることによって、普通の関数の3Dグラフも描くことができます。



6 おわりに

Rを使うメリットは、このようなプロも使うほどの高機能なソフトウェアであるだけでなく、様々な情報やパッケージが充実していることにあります。書籍なども2004年度より急速に発刊されています。また、群馬大学の青木先生のサイト (<http://aoki2.si.gunma-u.ac.jp/R/>) にいけば、統計解析に関する一通りのこ

とを学ぶこともできます。R についてのインストール情報や FAQ などは、Rjp-Wiki(<http://www.okada.jp.org/RWiki/>) がかなり充実しており、R に関する情報はすべてここから探し出せると言ってもいいほどのです。グラフィックスに関する情報も海外のサイトではかなり多く見つけることもできます。

無料でここまでできるソフトは、なかなかないのではないかと考えています。統計解析の好きな方以外でも、ぜひ一度 R をさわってみてください。そして高等学校ではあまり扱わなくなってしまった「統計」に少しでも興味を持ってくれるような話ができるように先生方も R で遊んで欲しいと思っています。

emath コラム

enumerate 環境で番号の書式を変える

これは emath ということではないが、**enumerate** 環境では、番号の深さによって括弧がついたり、ローマ数字になったりいろいろになるが、それを指定する。

【い】 千葉

【ろ】 埼玉

【は】 茨城

`\begin{enumerate}[【い】]` と `[]` の中に指定する。番号を `[イ]` と書けば、イ、ロ、ハ …, 番号を `[あ]` と書けば、あ、い、う …, 番号を `[A]` と書けば、A, B, C …, 番号を `[i]` と書けば、i, ii, iii, …