Rによる数値計算と統計解析

Numerical Computing and Statistical Analysis by R

作花 一志，胡 明（京都情報大学院大学）

Kazuyuki Sakka and Ming Hu（The Kyoto College of Graduate Studies for Informatics）

Abstract

As an easy, free, and powerful programming language for statistical computing, R has attracted growing attention recently. In this paper, we introduce some basic applications in the field of (1) numerical computing, such as differential and integral calculus and differential equation; (2) graphics processing, such as 3D graph; (3) statistical analysis, such as scatter plot and regression analysis. We also show some benefit of R compared with Excel through numerical examples.

1. はじめに

Rは統計解析用に開発された有名なフリーソフトで，統計学を基本から学び多変量解析までを修得するのに役立ち，ビジネス界で実用されている。またグラフィックス機能が充実していて3D描画も容易にできるので，シミュレーションに適している。そのため近年データサイエンスのツールとして注目されている。

Rは非常にたくさんの関数，ライブラリを持ち，しかも毎月のように増補され，適用範囲は多方面に広がっている。

大学初年度の数学である線形代数や微積分学の教科書のかなりの部分は行列，行列式，導関数，不定積分の計算で占められ，特異な技巧を必要とするものも少なくないが，Rを使えば簡単に求まる。また行列・一次変換を修得してから学ぶ固有値問題や，微分積分を修得してから学ぶ微分方程式の解なども短いステップで解くことができる。

この小文では数学教育の手段として有用なプログラムを紹介するもので，第2節は作花が第3節は胡が執筆した。主に参考したサイトは[1]，[2]である。なお結果のカラー図や詳しいプログラムコードはウェブサイト[3]を参照されたい。

2. 数値計算とグラフィックス

2.1 方程式の解法

非線形方程式　$x^2-2^x=0$ を解いてみよう。
x=2が解になることはすぐにわかり，またx=4も明らかに解である。解はこの二つだけだろうか。これ以上は直観でも解析でもわからない。そこでグラフを描いてx軸との交点を調べる。するともう一つ負の解があるが，これは数値的にしか求まらない。幸いRではunirootという便利な関数が装備されている。
下記プログラムを実行するとコンソールに

```
$root
[1] -0.7666825
```

と，またグラフィック画面には赤字でこの値が表示される（図1)。整方程式の場合はxの係数を昇べき順にベクトルで与えてpolyroot関数により虚根も含め簡単に求まる。

```
fn <- function(x) x^2-2^x
curve (fn, -3, 5)                # この範囲の f(x)プロット
abline(h = 0,  col = 4)     # 青で x 軸を描く
abline(v = 0,  col = 4)     # 青で y 軸を描く
Sol <- uniroot(fn,  c (-1,  0))  # c の範囲で方程式を解く
Sol
text (-1, -2, Sol$root, col = "red")
title(main="Equation      x^2=2^x")

# 整方程式 x^2-2x+3=0の解
# polyroot (c(3, -2, 1))
```

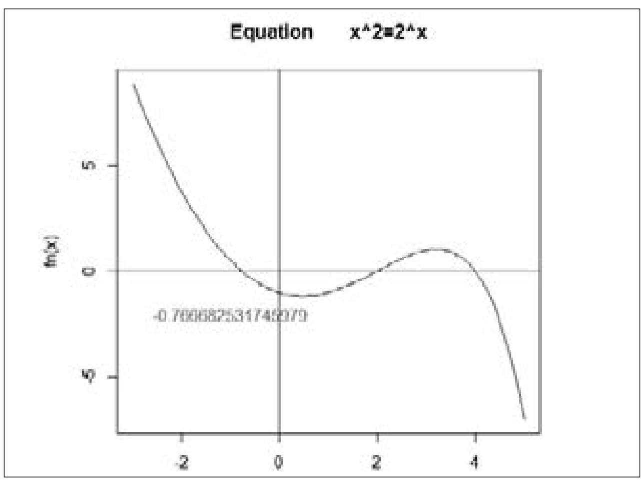

図1 $x^2-2^x=0$ の解

2.2 微分

関数をexpressionで定義しDを使うとその導関数が得られる。ただし関数形だけで関数値は求まらないしグラフも描けない。関数f1とその導関数f2を求めるには

```
f1 <- deriv(~*****, "x", func=T)
f2 <-function(x) attr(f1(x), "gradient")
```

とすればよい。f1(0)よりx=0における関数値を，f2(0)より微分係数を求めることができる。

```
curve (f1(x), x1, x2, ylim=c(y1, y2))
curve (f2(x), x1, x2, lty=3, ylim=c(y1, y2), add=T)
```

とすれば $x_1<x<x_2$，$y_1<y<y_2$ の範囲で二つのグラフを同一座標に描くことができる。lty=3はy=f2(x)を破線で描くことを意味する。導関数f2(x)=0となるxにおいてf1(x)は極値をとる。f1(x)＝sin(x)＋2 / (x＋3)の結果であり，そのようなxは負で2個，正で2個存在する(図2)。f1(x)のグラフを描き，極値を求め，方程式を解くことは煩雑な計算をしないで容易に求められる。

```
# 関数形だけで値の計算プロットは不可
f0<-expression (x^3+a*x+b*cos(x))
Dif<-D(f0, "x")

f1 <-deriv(~sin(x)+2/(x+3), "x", func=T)
f2 <-function(x) attr(f1(x), "gradient")
curve(f1(x), -6, 6, ylim=c(-5, 10))
curve(f2(x), -6, 6, lty=3, ylim=c(-5, 10), add=T)
abline(v = 0,  col = 5) #  x 軸を描く
abline(h = 0,  col = 5) #  y 軸を描く
```

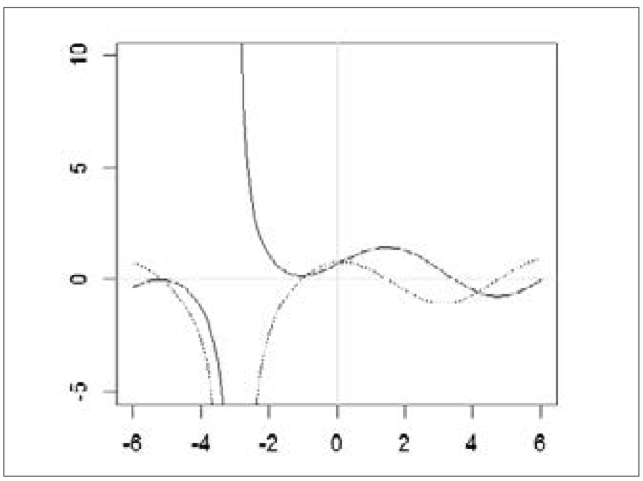

図2 y=sin(x)+2/(x+3)とその導関数（破線）

2.3 積分

ある関数f(x)をaからbまでの定積分した結果は

```
f<-function(x)  ******
integrate(f, a, b)
```

である。関数としては初等関数で表されるものなら何でもよく，また積分の上限下限として∞(Inf)も使用できる。

f a, b	sin(x) [0, 1]	1/x [0.001, 1]	exp(-x) [1, ∞)	$2/(1+x^2)$ [-1, 1]	√x/cos(x) [0, 1]
解析値	1-cos(1) 0.459698	-ln(0.001) 6.907755	1/e 0.367879	π	?
R	0.45969	6.907755	0.36787	3.14159	0.86417

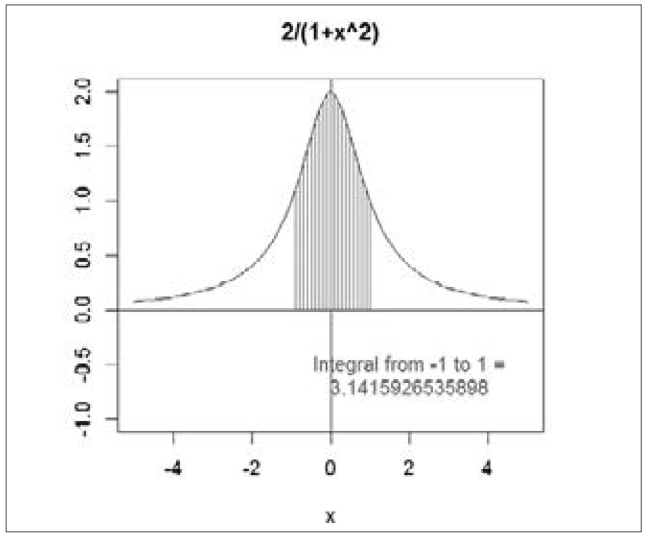

図3 定積分

0からtまでの積分した結果を新たな関数と定義すると，この値が求まり，プロットできるのはtがある1点のときだけである。ある関数とその原始関数のグラフを同一座標に描くことはできないものか？

そこで0からtまでの積分しさらにその値をプロットする関数を作り，tをある値からある値まで動かしてみよう。下記はf(x)=cos(x)を0からtまで積分した値をyとし，(t, y)をプロットする関数をf1として，tを0.05刻みで-2から10までプロットするプログラムである。結果は図4で太線で描かれている。

```
f <-function(x) cos(x)
f1 <-function(t, x1, x2) {
    y=integrate(f, 0, t)$value
    par(cex=0.7)
    plot(t, y, xlim=c(x1, x2), ylim=c(-2, 10), pch=20)
  }
x2=10;h=0.05;n=x2/h
for (i in 0:n) {
 t=h*i;f1(t, -2, x2);par(new=T)
}
abline(h=0, col=4);abline(v=0, col=4)
par(cex=1.0)
curve(f(x), -1, 10, col=2, add=T)
title("原始関数")
mtext("原関数", 3, 0, col=2)
```

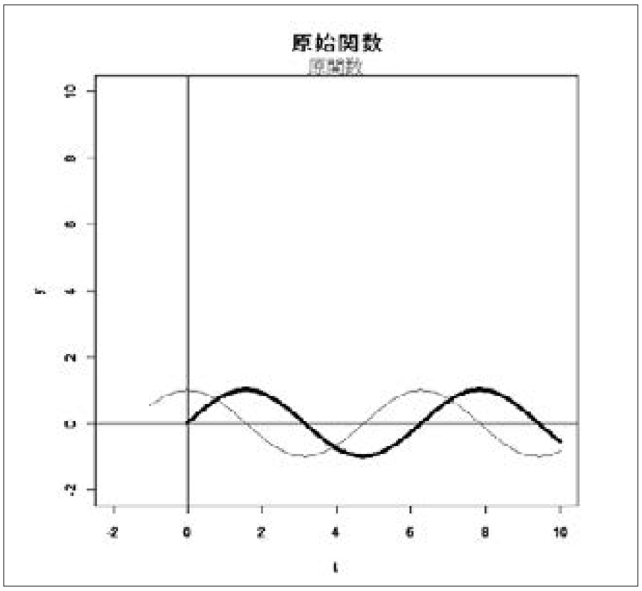

図4 cos(x)

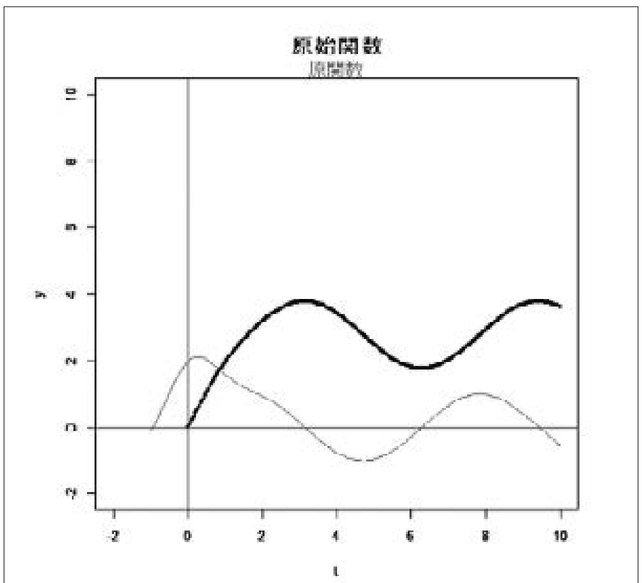

図5 2exp($-x^2$)+sin(x)

図4の場合は原始関数がsin(x)と簡単にわかるが，一般に原始関数は存在しても初等関数では表されないことが多い。しかしこの方法だとほとんどの関数の原始関数のグラフを描くことができる。図5は原関数が2exp($-x^2$)+sin(x)の場合である。

2.4 微分方程式の解

微分方程式を解くとは x，y，y'，y'' などを含む方程式からyをxの関数として表すことで，微分積分学修得の後で学ぶのが通例である。その起源はニュートンの運動方程式に始まり，これまでさまざまな解法が研究されているが，解析的方法は一般に非常に難解・技巧的であり，数値的にしか解けない場合も多い。

ところがRにはodeと言う便利な関数が装備されていて，y'と初期値を与えれば，非線形でも連立でも高階微分の場合でも容易に解くことができる。プログラムは次頁に載せたが，まずdfnで微分方程式y'を定義する。timesでは0から5まで0.1刻みで y，y'の値を計算しoutというmatrixに収める。outの第1列，第2列はtとyで，headでその最初の3行だけをコンソールに出力し，またplotによりグラフを描く。図6は

$y' = x+2y \quad y(0)=2$ の解を図示しているが解析解

$y = 9\exp(-2x)/2+x/2-1/4$ が存在している。

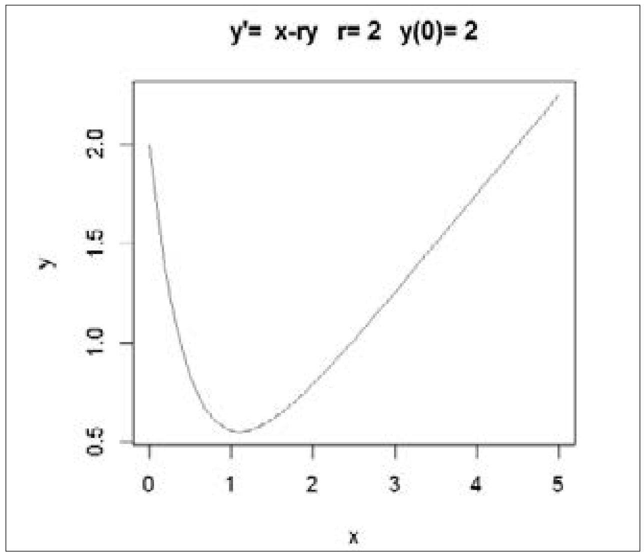

図6 1階微分方程式の解

2階微分方程式の場合は y→y[1]，y'→y[2] と置換し2元連立方程式に変換する。

y''=y[2]' だから

outの各列は t，y，y' であり，前述のようにその最初の3行だけをコンソールに出力し，またplotにより値をプロットした。図7は y''+y=0 y(0)=0 y'(0)=1の解を図示したもので実線はy，破線はy'の値を表している。なお解析解は y=sin(x) である。

なおこのプログラムにはパッケージdeSolveをインストールしておく必要がある。

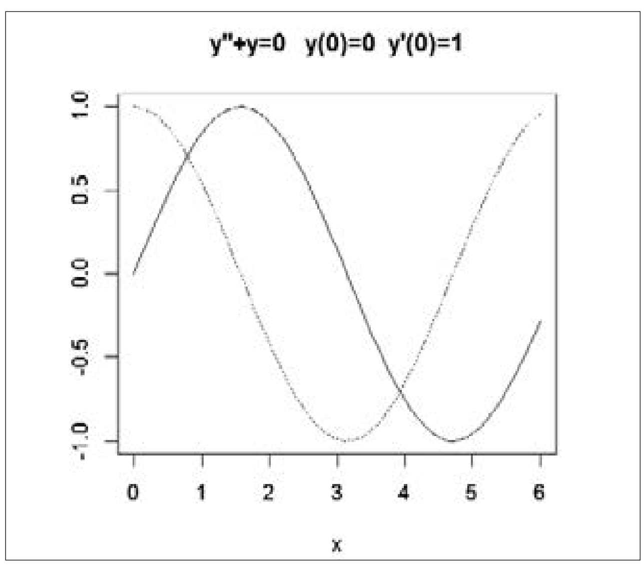

図7 2階微分方程式の解

```
library(deSolve)
# 1階微分方程式の数値解法
#  y'=x-ry     y(0)=y0     r, y0 はパラメータ
r=2;y0=2;fm="x-ry"
   dfn <-function(x, y, parms){list(x-y*r)}
times <-seq(from = 0, to = 5, by = 0.1)
out <-ode(y = y0, times = times, func = dfn, parms=null)
head (out, n = 3)
plot(out[, 1], out[, 2],
col=2 , type="l", xlab="x", ylab="y")
title(paste("y'= ", fm, "  r=", r, "  y(0)=", y0))
# 2階微分方程式の数値解法
#   y''+y=0   y(0)=0  y'(0)=1
#    y→y[1] y'→y[2] とおくと  y''=y[2]' だから
#   y[1]'=y[2]
#  y[2]'=-y[1]
y0 <- c( 0, 1)          # 初期条件  t=0 にて y=0 v=1
df <- function(t, y, parms) {
  dy1 <- y[2]
  dy2 <-  -y[1]
  list(c(dy1, dy2))}
times <- seq(from = 0, to = 6, by = 0.1)
out <- ode (times = times, y = y0, func = df, parms = NULL,
   method = rkMethod("rk45ck"))
head (out, n = 3)
plot(out[, 1], out[, 2], lty =1, type="l", xlab="", ylab="")
par(new=T);plot(out[,1],out[,3],lty=3,type="l",xlab="x",
ylab="")
mtext("y", 2, -1, col=2);mtext("y'", 4, -1, col=3)
title("y''+y=0   y(0)=0  y'(0)=1")
```

2.5 3D図形その他

この節の図はすべて[3]でご覧いただきたい。

z= f(x, y)をプロットするにはライブラリfieldとrglをインストールしておくとよい。zを定義し

f <- function(x, y) sin(x^2+y^2)

x, yの範囲を指定し

x <- seq(-3, 3, length = 60)

y <- seq(-3, 3, length = 60)

z <- outer(x, y, f)

persp (x, y, z, オプション)で図8を描くことができる。

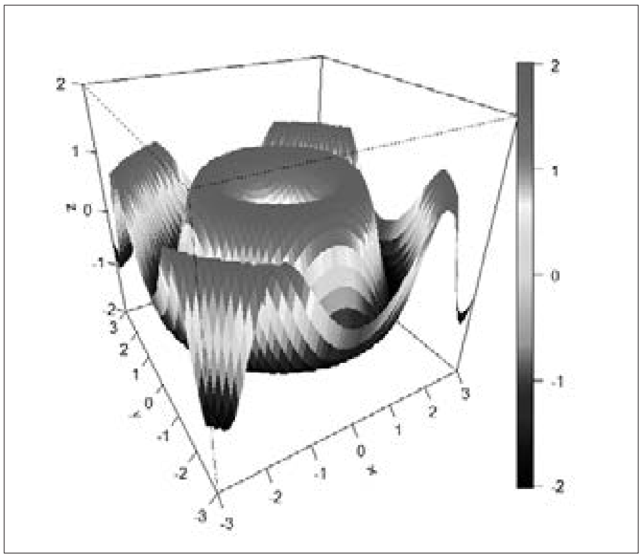

図8 z=sin(x^2+y^2)

また図9は小球150個を描いたものでマウスでドラッグすると画像が回転する。

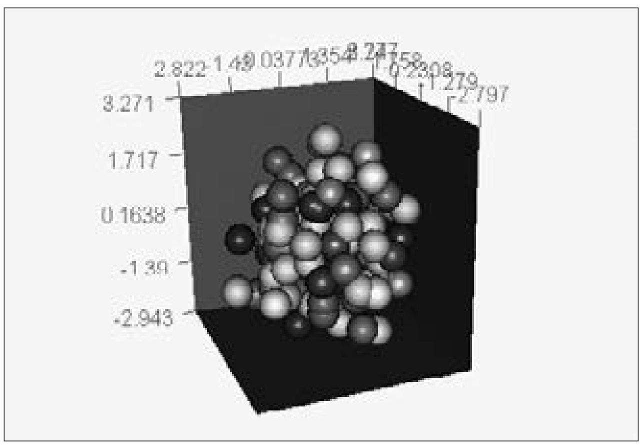

図9

図10は次の連立漸化式

$x_2 = a*x_1+y_1+b+c/(1+x_1^2)$; $y_2=b*x_1$

を6万回繰り返し計算して3000回ごとに色を変えてプロットしたものである。

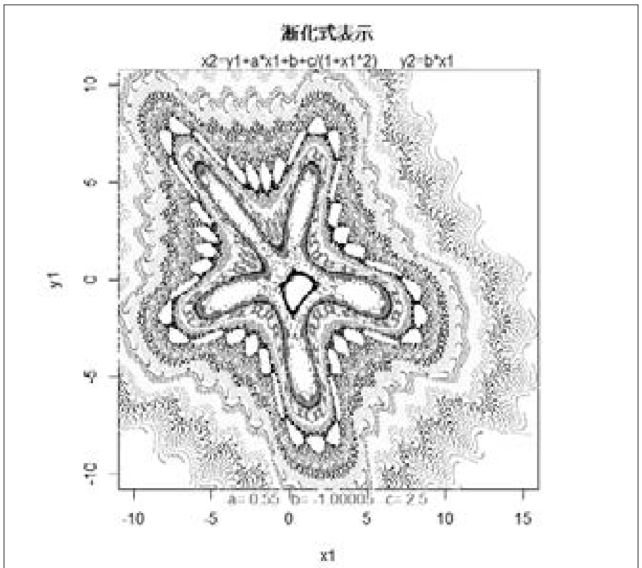

図10 カオス図形 a=0.55;b=−1.00005;c=2.5

a, b, cの値がわずかに変わっても全く違った図になる。

また図11，12は有名なフラクタル図形マンデルブローの複素数漸化式

$z_{n+1}=z_n^{\alpha}+C \qquad z_0=0$

を図示したものである。

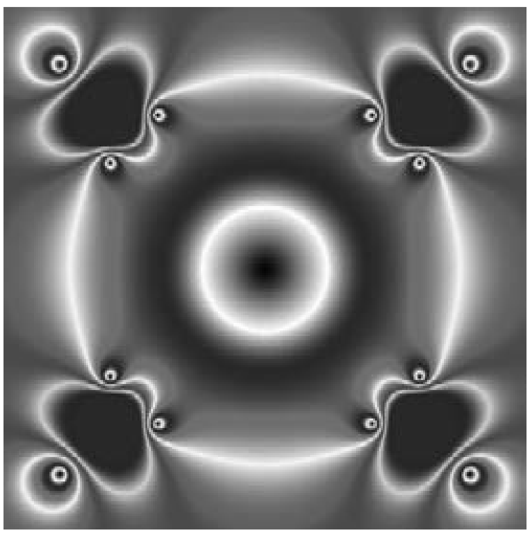

図11 マンデルブロー　α=−3

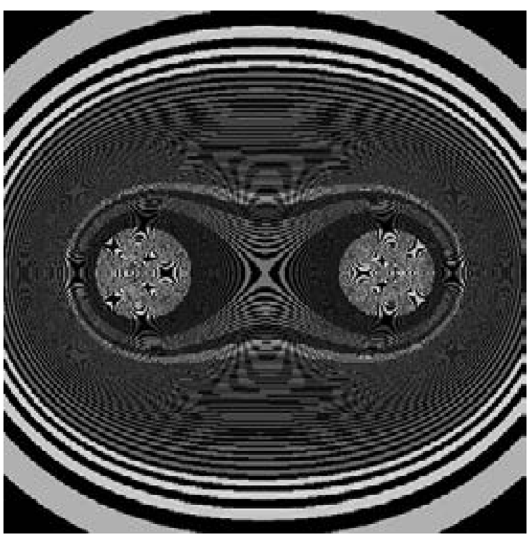

図12 マンデルブロー　α=2.01

Cも複素数でその実部も虚部も-1.2 ～1.2 間を500回分割して計算してある。α=-3の場合を計算したがαは小数でもよい。15枚の画像を重ねgifファイルとして保存しブラウザから開いて閲覧する。

Imageという関数で行列を可視化することができる。xは下記のような4行2列のmatrixであるが

1	5
2	6
3	7
4	8

各カラーコードにそってImage関数で描くと図13ができる。ただし下段左から右へ，上段左から右への順になる。転置行列t(x)を可視化すると図14が描ける。またこれを画像imgx.pngとして保存することができる。

```
# 行列の可視化
x<-matrix(c(1, 2, 3, 4, 5, 6, 7, 8), 4, 2)
image (x,  col=c(1, 2, 3, 4, 5, 6, 7, 8))
dev. new()
png(”imgx.png”)
imgx<-image(x, col=c(1, 2, 3, 4, 5, 6, 7, 8))
dev.off()
```

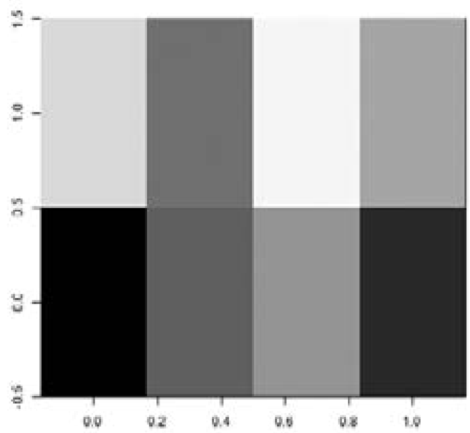

図13 行列xを可視化

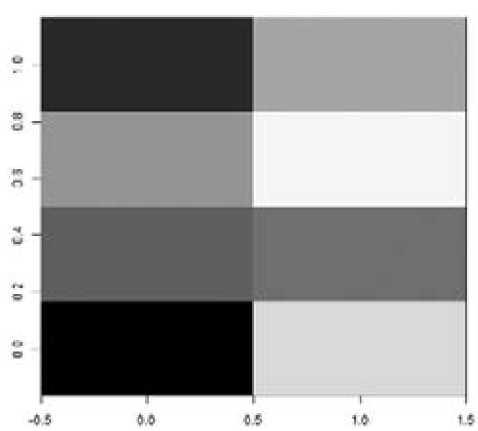

図14 転置行列t(x)を可視化

3. 統計解析

3.1 ヒストグラム

ヒストグラムとは，縦軸に度数，横軸に階級をとった統計グラフの一種で，データの分布状況を視覚的に認識するために主に統計学や数学，画像処理等で用いられる。柱図表，度数分布図，柱状グラフともいう。

Excelではまず度数分布表を作ってからヒストグラムを描くという手順が必要であるが，Rでは横幅の調整や縦軸を確率に指定することや密度表示することなどもできる（図15）。

```
# 密度表示
x <- c(rnorm(1000, 3, 2)) # 平均は3, 標準偏差は2での
乱数を発生する。
hist(x, freq=FALSE)
lines(density(x), col="orange", lwd=2)
```

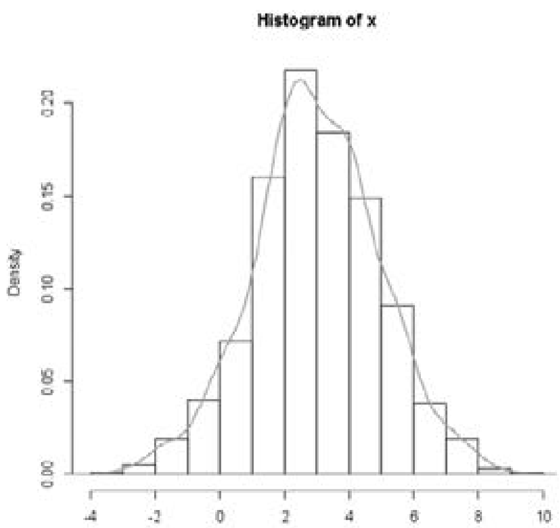

図15

図16のようにRで複数のヒストグラムを同時に描くこともできる[4]。

```
# 複数のヒストグラムを描く方法
x <- rnorm(1000, 10, 5)
y <- rnorm(1000, 15, 5)
hist(x,col = "#ff00ff40", border="#ff00ff", breaks = 20)
#左端のピンク色のヒストグラム
hist(y,col = "#0000ff40",border="#0000ff",breaks = 20,
add = TRUE) #右端の青色のヒストグラム
```

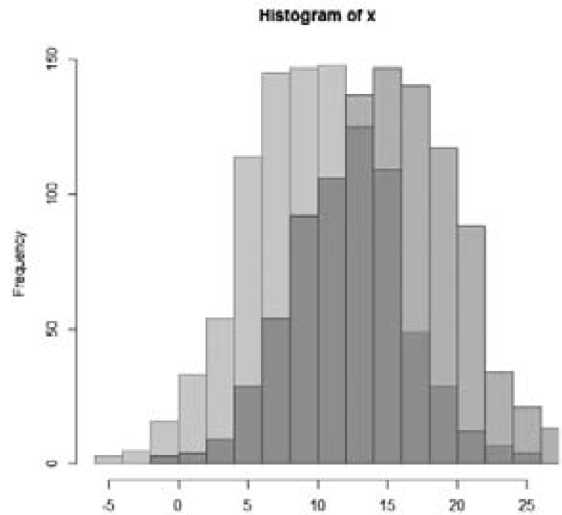

図16

二つランダムに生成した正規分布に従うデータ列のヒストグラムを同時に表示した。そこに，被っている部分はディスプレイ上では紫色になっている。

3.2 散布図・相関係数

散布図とは，縦軸，横軸に2項目の量や大きさ等を対応させ，データを点でプロットしたものである。各データは2項目の量や大きさ等を持ったものである。2項目データの分布，相関関係を把握できることは散布図の特長である。

Excelではデータの選択などすべてマウス操作で行うがRではplotという便利な関数を利用して，簡単に描くことができる。

以下，最高気温（x）とかき氷の販売数（y）を変数として，Rで散布図を描くことを紹介する[5]。縦横の直線はxとyの平均である。最高気温とかき氷の販売数とはきわめて強い相関があることがわかる。

```
気温
<-c(21, 22, 23, 24, 24, 25, 25, 26, 26, 27, 27, 28, 29,
32, 28, 24, 31, 30, 32, 33, 32, 34, 35, 35, 36)
販売数
<-c(298, 312, 321, 333, 322, 331, 315, 324, 312, 340,
365, 358, 364, 410, 378, 315, 368, 395, 393, 410,
415, 456, 468, 442, 486)
#気温とかき氷の販売数データを読み込み
plot(気温, 販売数, main="気温とかき氷の販売数")
abline(v=mean(気温), h=mean(販売数), col=3)
r<-cor(販売数, 気温)      #相関係数
text(24, 450, paste("相関関係=", round(r, 3)), col=2)
#グラフに相関関係を表示する。
```

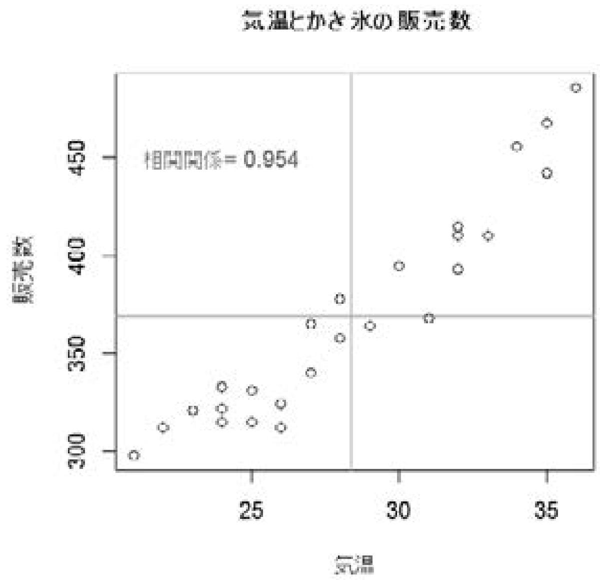

図17

次の不動産データ表を読み込ませて，相関係数を求めてみよう[5]。

番号	専有面積（㎡）	徒歩時間（分）	築年数（年）	賃料（万円）
物件1	28	12	32	6.4
物件2	30	15	31	7.2
物件3	51	6	12	10
物件4	44	20	11	9.8
物件5	38	7	10	10
物件6	47	12	11	10.5
物件7	40	12	13	11
物件8	52	3	14	12
物件9	55	4	16	12.4
物件10	63	3	14	13.2
物件11	54	10	10	13.5

プログラムの第1行

d4<-read.table("malcuster.txt", header=T)

はこのテキストファイルmalcuster.txtを第1行はヘッダとして読み込み，データフレームd4を作ることを意味する。

さらに，Rでpairs.panels関数を利用して，Excelではできない各二つの要素の間に散布図，相関関係，ヒストグラムなど一つの図に表すこともできる。これにより図18が描かれ，上右側に表示する数値は各二つの要素の相関関係である。下左側に表示する図は各二つの要素の散布図を楕円と平滑近似曲線で表される。楕円の傾きは相関係数の符号を，また扁平度はその値を表す。中央の図は各要素のヒストグラムである。

```
#4種類の要素について散布図を描く
d4 <- read.table("malcuster.txt", header=T)
# テキストファイルを読み込む
d4
cor(d4[, 2:5])             # 相関行列
library(psych)
pairs.panels(d4[, 2:5], lm=TRUE )
# 散布図, 相関係数, ヒストグラム, 楕円は相関係数を表す
pairs.panels(d4[, 1:4])
```

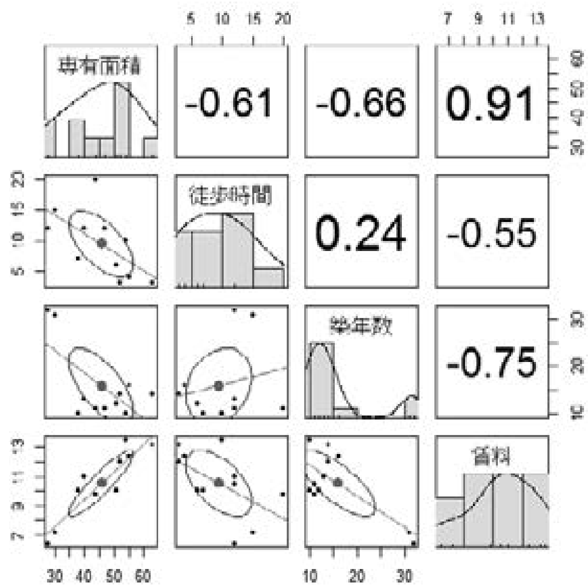

図18

次いでRでクラスター分析について紹介する。クラスター分析とは，異なる性質のものが混ざりあっている集団（対象）の中から互いに似たものを集めて集落（クラスター）を作り，対象を分類しようという方法を総称したものである[6]。このクラスター分析は統計解析や多変量解析の分野で基本的なデータ解析手法としてデータマイニングでも頻繁に利用されている。クラスター分析には，大きく分けると階層クラスター分析と非階層クラスター分析の二種類の方法がある。ここにRのhclust関数で階層クラスター分析を説明する。

```
dd4 <- dist(d4) # 距離
cl4 <- hclust(dd4, method="complete")
plot(cl4, hang=-1)  # tree
```

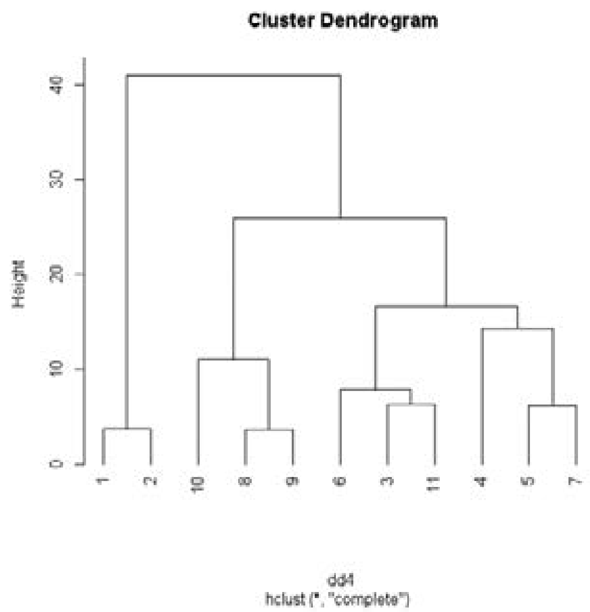

図19

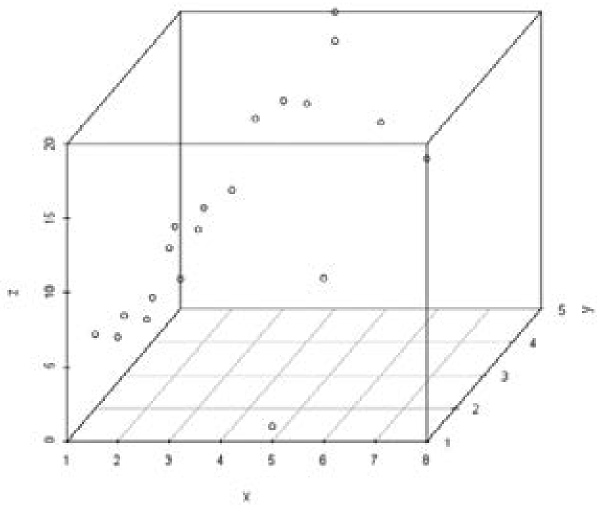

図20

このように，階層クラスター分析を行うとデンドログラム（樹形図）が表示される（図19)。この図によって各番号のデータがクラスターとして結合されていく過程を見ていくことができる。例えば，NO.8，NO.9とNO.10を例にとって見ると，NO.8とNO.9がまず結合される。これは，NO.8とNO.9がこれ以降一つのクラスターとして結合されたことを表す。さらには，これがNO.10と結合される。これは，NO.8とNO.9のクラスターにNO.10が組み込まれたことを表す。そして，デンドログラムでは，図の下の方で結合すればするほど近い関係にあるといえるので，NO.8とNO.9は非常に近い，NO.10はそれについで近いということがここから読み取れるのである。また，最も下で結合しているNO.1とNO.2及びNO.8とNO.9は，これらのデータ番号の中で最も近い二つと分かるのである。

二次元散布図だけではなく，Rでscatterplot3d関数（描いた立体図は回転できる）或いはplot3d関数（描画角度を指定して，静止画を描く）を利用して，三次元散布図を描くこともできる。ここにscatterplot3d関数で次の例を通して，三次元散布図を紹介する[4]。

```
# scatterplot3dを利用するには該当パッケージをインストールする必要がある。
library(scatterplot3d)
x <- c(5, 2, 6, 4, 1, 2, 3, 6, 1, 2, 3, 4, 1, 2,
3, 8, 1, 2, 3, 4)
y <- c(1, 2, 1, 4, 5, 1, 2, 3, 4, 5, 1, 5, 3, 4,
5, 1, 2, 3, 4, 5)
z <- c(1, 6, 11, 16, 2, 7, 12, 17, 3, 8, 13, 18, 4, 9,
14, 19, 5, 10, 15, 20)
scatterplot3d(x, y, z)
```

3.3 回帰分析

回帰分析は，目的変数と説明変数の間に数式で目的変数が説明変数によってどれくらい説明できるのかを定量的に分析することである。説明変数の個数により，単回帰分析（説明変数が一つの場合）と重回帰分析（説明変数は二つ以上の場合）が分けられる。用いられる方法によって，線形回帰分析と非線形回帰分析の二種類がある。

p66のプログラムに次のコードを付け加え，lmという関数を使って，気温によって，かき氷販売数を予測するように単回帰分析ができる。

```
result <- lm(販売数~気温) #回帰分析を行う
abline(result, col="blue") #回帰直線を描く,
result
b=round(result$coefficients[1], 2)
a=round(result$coefficients[2], 2)
text(24.5, 420, paste("y =", a, "x+", b), col=4)
```

result という命令の結果，コンソールに

```
Coefficients:
(Intercept)    気温
   36.21       11.74
```

と表示され，これが回帰直線の係数である。よってかき氷の販売数（y）と気温（x）の関係は以下の式で表される。

$$y = 11.74x + 36.21$$

これを図示したものが図21である。

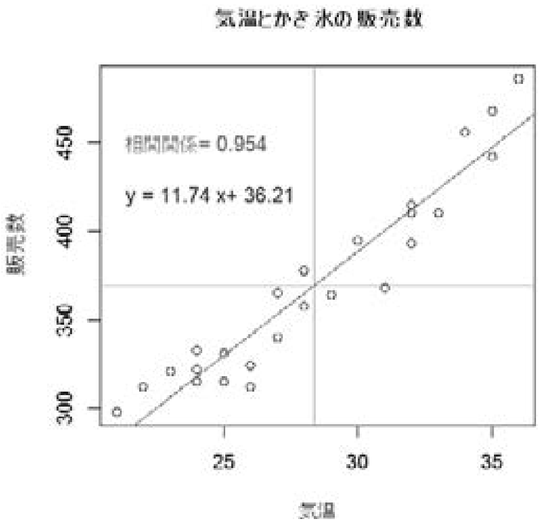

図21 単回帰分析

p67の不動産のデータを使って重回帰分析を試みよう。p67のプログラムに次のコードを付け加える。賃料（y）は専有面積(x_1)，徒歩時間(x_2)，築後年数（x_3）の関数とする。

```
x1=d4[,2]
x2=d4[,3]
x3=d4[,4]
y=d4[,5]
result2 <- lm(y~x1+x2+x3, data=d4)
result2
```

result2 という命令の結果，コンソールに

Coefficients:

(Intercept)　x1　x2　x3

5.67205　0.14084　-0.03076　-0.07985

と表示され，賃料（y）と専有面積（x_1），徒歩時間（x_2），築後年数（x_3）の関係は以下の重回帰式で表される。

$y=0.14084x_1-0.03076x_2-0.07985x_3+5.67205$

面積が広く，徒歩時間・築年数が小さいほど賃料は高くなる。

なお上記はいずれも目的変数は説明変数の1次式であるが，多項式あるいは超越関数の場合は非線形回帰分析となる。これについての例はp70の論文を参考されたい。

4. 考察

以上のようにRプログラミングは数学の学習に非常に有効である。これらは未完成であり，引き続き第2節関連では幾何学や画像処理，第3節関連では検定や多変量解析などの項目のプログラムを作って教材に供していく。

また地理データの可視化，天文シミュレーション，線形計画法などにも適用していく予定である。

【参考文献】

[1] http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html

[2] http://itbc-world.com/home/rfm/home/

[3] http://web1.kcg.edu/~sakka//num/R/web/index.htm

[4] Biostatistics; http://stat.biopapyrus.net/.

[5] 末吉正成，里洋平，酒巻隆治，小林雄一郎，大城信晃; Rではじめるビジネス統計分析；翔泳社； 2014/7/18.

[6] http://www.macromill.com/landing/words/b003.html

◆著者紹介

作花 一志　Kazuyuki Sakka

京都情報大学院大学教授。

京都大学大学院理学研究科博士課程修了（宇宙物理学専攻），京都大学理学博士。専門分野は古天文学，統計解析学。

元京都大学理学部・総合人間学部講師，元京都コンピュータ学院鴨川校校長，元天文教育普及研究会編集委員長。

胡 明　Ming Hu

京都情報大学院大学講師。

京都大学大学院情報学研究科博士課程修了（数理工学専攻），情報学博士。

研究分野はナッシュ均衡，マルチリーダ・フォロワゲームと均衡制約付き均衡問題。

日本オペレーションズ・リサーチ学会会員。

元日本学術振興会特別研究員。