R: 統計分析のためのソフトウェア

2018 年 3 月 29 日

R 入門として、ネット上や参考書が多く出版されています。統計分析にとどまらず、最近は機械学習、AI といった分野にも拡大されています。ここではもっとも基本的な項目を説明してみましょう。

• R ノート：データ解析とグラフィックスのためのプログラミング環境

`https://cran.r-project.org/doc/contrib/manuals-jp/R-intro-170.jp.pdf`

• 本格的な詳しいマニュアル、約４００ページにもなる。R が提供するヘルプドキュメントの和訳が提供される。さらに参考画像もあって、大いに役立つ。たとえば、関数の引数のオプションを適切に処理することで、違和感のない素晴らしいものができる。

R 基本統計関数マニュアル（Mase-Rstatman.pdf 間瀬茂氏から引用）：

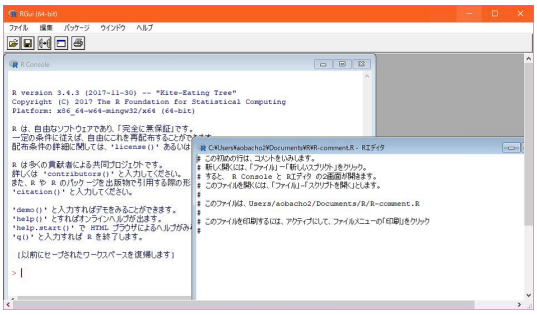
(i) `https://cran.r-project.org/doc/contrib/manuals-jp/Mase-Rstatman.pdf`

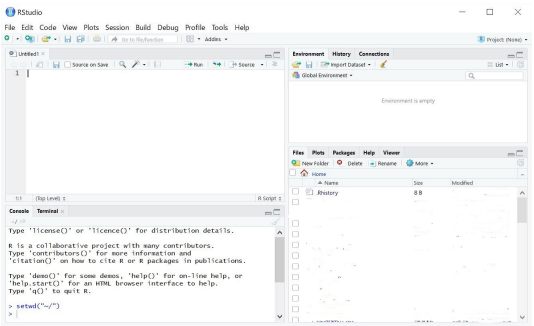
(ii) `http://www.okadajp.org/RWiki/?RWiki`

1 R の基本事項

1.1 データ演算、データの取り込みと出力

プログラム RGui(64bit) の起動アイコンをクリックすると、「起動画面（R Console）」と「エディター画面（R エディタ）」が表れます。いずれかの画面をクリックしたとき、ファイルの選択タグが異なることに注意。もし一つしか出ない（起動画面）のみであるときには、ファイル・タグで新しいスクリプトを開きます。既にエディターをつかって読み込む場合であれば、メインの「ファイル」-「保存」したのちに、エディターを起動して「スクリプト」を書くように、このようなウィザード（窓）の中にエディター・スクリプトウィザードが表れます。左側の図です。この状態でも十分使えますが、これに加えて、もう少し使い易い補助のプログラムがこの右側にある「RStudio」です。これはちょっと細かな設定やプログラムの修正を施していくときには、RGui(64bit) の単体より便利ですから、インストールした方がよいでしょう。コンソールの実行、関数の読み込み、ファイルリストの表示など、統合的に処理できます。左のコンソールだけでは、単一行の処理ができるものと考えてください。また作業ディレクトリ (working deirectry) の変更 `setwd("ディレクトリ")` やその確認命令 `getwd()` も確かめましょう。





コマンドとして使用できる文字は半角英数字で、大文字と小文字は区別されます。組み込みコマンドの多くは小文字が使われるから、個人の入力では、逆に大文字や初めの文字に大文字とかにする工夫もあります。変数名にはピリオド「.」、アンダースコアー「_」が使え、後半に意味を付けられる。

オブジェクトは,「ベクトル」と「リスト」をつかいます。コメントは「#」、Enter を押すならば、入力行のおしまいを意味し、Shift + Enter を押した場合は、続きの行入力する場合です。ベクトルとリストの違いは、ベクトルが同じ処理が可能な数の連続等のデータ列で数の演算ができるもの。リストは異なった属性のデータの並びで、行が一つのデータベースのためのデータ値となっているもの。先頭の c は column の意味で、入力の頭につけて、

```
xdat < - c(1,2,3,4,5,6)
```

などと入力でき、 > sum(xdat) で答えの値 > [1] 21 という結果になる。 [1] は結果の 1 行目ということです。 <- の意味は変数に代入するということですが、通常の等号のうちで、代入式

```
> xdat2 = c(10:20); sum(xdat2)
> [1] 165
```

としても演算できる version がありますから、

```
10+11+... + 20 = 165
```

確かめてください。しかし等号には比較の意味もありますから、注意します。

ファイルに記録された c s v データ (カンマ区切りのテキストデータ) を読むためには

```
> xfdat = read.csv("c:\\Rfiles\\rdat123.txt",
+ header = FALSE,sep=",")
```

とすると、ドライブ c: にあるファイルから、カンマ区切りで xfdat にデータがはいる。入力を促すプロンプト > につづけて、 > xfdat と入れると、その内容が表示される。結果などのデータを書きだすには、

```
> xout = c(10,11,12,13,14,15,16,17,18,19,20,21,22)
> write(xout, "c:\\Rfiles\\rdat124.txt", sep=" ", ncolumns= 5)
> write(c("mean(xout)=", mean(xout)), "c:\\Rfiles\\rdat124.txt",
+ append=TRUE)
```

とすれば、mean(xout)=を付け加えられた (append)、平均の結果を書くことができます。

表計算ソフト Excel とデータのやり取りは、read.tabel と write.table をもちいます。読み込みは c s v の形式でできますから、データの形式を data.frame で整えて行い、書き出しをします。たとえば

```
> outdat = data.frame(
+ 番号 = c( 101, 102, 103, 104, 105),
+ 身長 = c(173, 172, 168, 170, 174))
> write.table(outdat,"\\Rfiles\\smp.csv",sep=",",row.names=FALSE)
```

これを read.table で読むと、header = TRUE にして、列名の「番号」「身長」が書かれていることに確認します。これによって右側のテーブルが得られます。左側番号列はエクセルによるもので内容は項目の番号 $101, 101, \cdots$ が outdat[,1] で、 outdat[,2] が項目の身長 $173, 172, \cdots$ に対応します。

	番号	身長
1	101	173
2	102	172
3	103	168
4	104	170
5	105	174

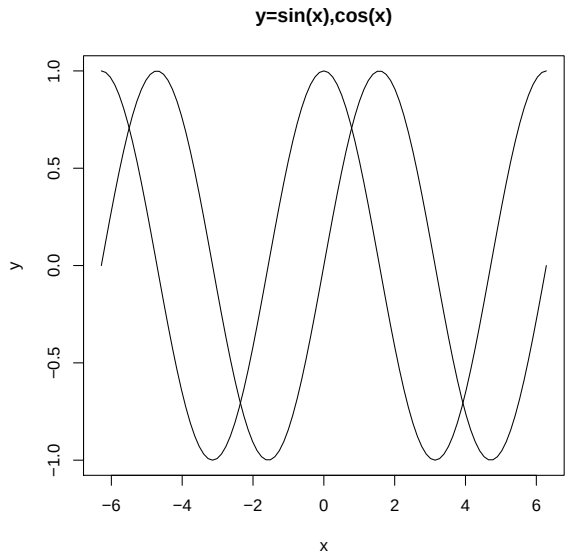
1.2 グラフの作成

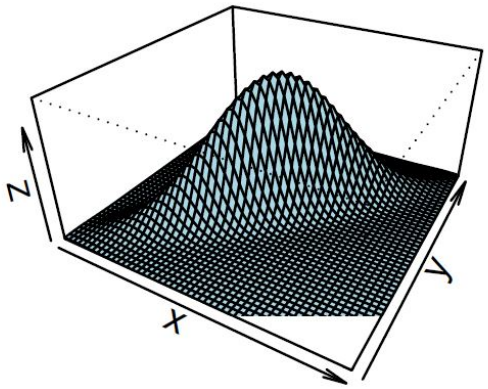
- plot(xdat,ydat, オプション) ; 関数のプロット、オプションは「type = l (エル)」: 結ばれた曲線, 「type = p 」: （デフォルト）座標の値を結ぶ点列, 「type = h 」: ヒストグラムの形に描く. 「xlabel = x 軸のラベル名」, 「ylabel = y 軸のラベル名」, 「main = グラフの表題名」,
- curve(expr,from,to,option), オプション add = TRUE で複数のグラフを描ける。

もし描いたグラフをファイル保存するためには、描いたグラフ画面「R Graphics Device」をアクティブにしておいてから、上側に並ぶタグ・メニューで「ファイル」-「別名で保存」-各ファイル形式の選択とファイルの名づけをおこないます。たとえば、

```
> curve(sin(x),-2*pi,2*pi,xlabel='x', ylabel='y',main='y=sin(x),cos(x)')
>  # sin(x) の描画, x 範囲、表題
> curve(cos(x),-2*pi,2*pi,add=TRUE) # 追加の cos(x)
```

とすればつぎの図が得られるはずです。





つぎの関数 $f(x)$ は標準正規分布を表しています。この曲線はベル型（釣鐘形）で、統計ではよくお馴染みのもので、中心極限定理や多くの分野で基本的な働きをします。

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2},\ -\infty < x < \infty$$

R のグラフでは、上の関数を描くことができますが、2 変量関数 $f(x, y)$ の 2 次元関数を描くこともできます。

２変量正規分布は、相関係数 $0 < \rho < 1$ の値によって、見かけはずいぶん変わりますが、

$$f(x,y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-(x^2 - 2\rho xy + y^2)/(2(1-\rho^2))\right\}, \quad -\infty < x, y < \infty$$

という形で表せ、$\rho = 0$ であれば、x, y 軸のどちらからもみても同じで、独立な場合となります。

```
> x <- seq(-3,3,length=50)      # x 方向の分点
> y <- x                        # y 方向の分点
> rho <- 0.9                    # 2 次元正規分布の定数
> gauss3d <- function(x,y) {    # 2 次元正規分布の関数
+  1/(2*pi*sqrt(1-rho^2))*exp(-(x^2-2*rho*x*y+y^2) / (2*(1-rho^2)))}
> z <- outer(x,y,gauss3d)       # 外積をとって z 方向の大きさを求める
> z[is.na(z)] <- 1              # 欠損値を 1 で補う
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
```

ここで persp(x 軸のデータ，y 軸のデータ，z 軸のデータ，col = 色， theta = 横回転の角度，phi = 縦回転の角度，expand = 拡大率，border=NA) をもちいています。

1.3 組み込み関数

- 平均値： mean(データ名, na.rm=TRUE); # オプション na.rm=TRUE は欠測値 (NA) を無視 (remove) する
- 中央値： medain(データ名); データを小さいものから大きさ順にならば、全体の５０％に位置する値を返します
- 範囲： range(データ名); 最大値（max) - 最小値 (min) の値

データを画面から、データ名 xdat に入力するときにはまず > xdat=scan() として Enter キーを入れる。その度ごとに、行番号が出るので順次繰り返す。データの最終には何もせず、そのまま 空白行 Enter キーとすればよい。

2 統計データの操作

2.1 一様乱数、乱数の生成

ヒストグラム (histogram) の作成：

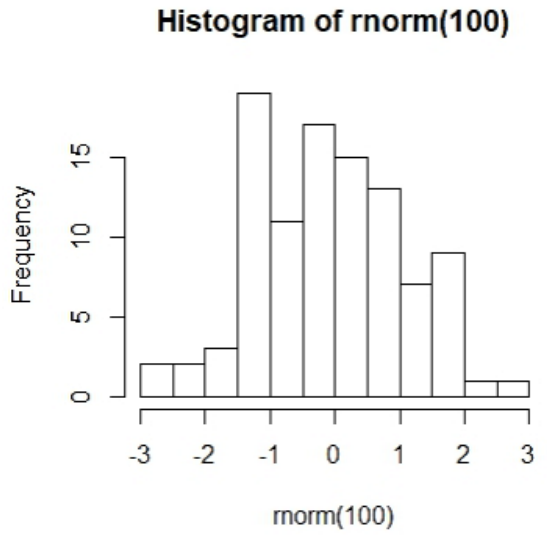
```
> xdat =rnom(100) # 100 個の正規乱数を生成し、xdat とおく
> hist(xdat, 6, xlab='rnorm(100)', ylab='Frequency', main = 'Histogram of rnorm(100) ')
> # xdat を 6 分割（６個の階級）にしたヒストグラム
```

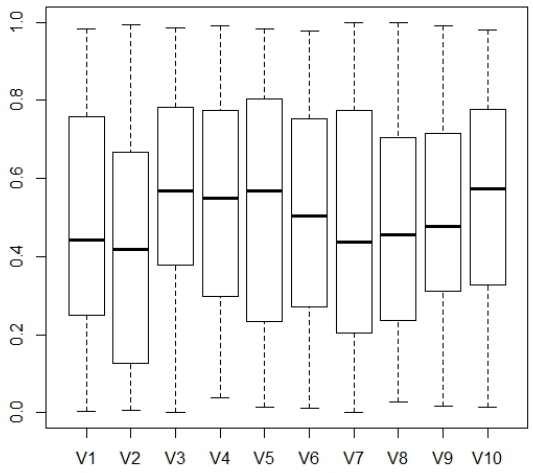
ヒストグラムを描くには、基本的には > hist(データファイル名) といれればよい。図ですが「グラフ」とは呼びません。またヒストグラムは連続型データに対して、データの階級に分類し、それを区間に分割してつなげたものですから、階級値の間隔はゼロでつながっています。いわゆる、棒グラフとは違います。離散型データに対しては棒グラフは意味を付けられます。つぎの図は正規乱数 (平均 0, 分散 1) を１００個生成した

ものをまとめています。一方、正規乱数をエクセルで =NORM.INV(RAND(),平均、標準偏差) で個数分をコピペすると、その個数だけ正規乱数をつくれます。この生成方法は「逆変換法」というもので一様乱数の値をもちいて、分布関数の逆関数により、正規乱数が求められます。これを集計整理しています。保存をｃｓｖ（カンマ区切りのデータ）にすると、これを R に読み込むには、たとえば、変数として Adata = scan("ファイル名.csv") などとすればよいです。

箱ひげ図 (boxplot)：1000 個の一様乱数を生成して、１００個ずつの１０組に分けた箱ひげ図を描く例を以下で述べます。

```
> A1000 = matrix(runif(1e3), nrow=100, ncol=10)
>  # unif （一様）を 1e3 = 1*10^3 = 1000 個つくり、１００行、１０列の行列とする
> dfA = as.data.frame(A1000)
>  # この結果 data.frame を ファイル名 dfA に引き渡す
> boxplot(dfA) # デフォルト形式での箱ひげ図を描く
```





セミコロン (;) を同じ行につづけると、命令を追加することができる。

```
> adat=rnorm(20)  # 20 個の正規乱数；
+ bdat=rnorm(20)  # 20 個の正規乱数；
+ cdat=rnorm(100)  # 100 個の正規乱数；
+ ddat=rnorm(100)  # 100 個の正規乱数；
+ edat=rnorm(1000)  # 1000 個の正規乱数；
+ boxplot(adat, bdat, cdat, ddat, edat) # 外側にある丸い点は「外れ値」を表す
```

ここで xdat: 横軸データ, ydat:縦軸データ, name.arg = 横軸変数名, ylim = 縦軸範囲 を表します。

棒グラフ (bar chart)：barplot(ydat,name.arg=xdat, ylim=c(0,200))

円グラフ (pie chart) ： pie(datname, radious=size, col=, main='title name ') # 項目名は names (xdata) =c('name1','name2','name3','name4','name5') それぞれに名づけ

幹葉図 (stem and leaf):

```
> xdat = c(118 125 132 145 152 137 122 133 134 152 120 146 150 147 152)
> stem(xdat, scale=1)
```

ならば、頭は2桁の数字（幹)、末尾の数字（葉）で並べられる。たとえば、1 5 2 が3個もあることが分かる。

```
The decimal point is 1 digits(s) to the right of the |
11 | 8
12 | 025
13 | 2347
14 | 567
15 | 0222
```

補足 ヒストグラムの命令 `hist()` は入力したデータにさらに細かな設定 (plot.histogram) を与えることができます。既定値は `right=TRUE` ですが、データを区間の階級に分割するとき、連続データですから、左半開区間 $a_1 < x \leq a_2, a_2 < x \leq a_3, a_3 < x \leq a_4, \cdots$ としていきます。ですから、最小値は a_1 よりも大きい値、逆にデータの最小値よりも小さい階級の初期値を a_1 とします。つぎの分割の個数は、3種類 (i) スタージェス法 (Sturges) 、(ii) スコット法 (Scott)、(iii) FD 法 (Freedman-Diaconis) からの選択ができます。既定は `breaks = ''Sturges''` で (i) の選択ですが、これらを少し説明します。一般に一つ山の形では、2項分布は正規分布で近似されます。これから、2項係数 ${}_nC_k = \dfrac{n!}{k!(n-k)!}$ の総和 $k = 0, 1, 2, \cdots, n$ が 2^n となることとこの山形を等分することを考えます。もし n 個のデータを h 階級に分けるならば、$n = \sum_{i=0}^{h-1} {}_{h-1}C_i = 2^{h-1}$ より、$h - 1 = \log_2 n$ となりますから、$1 + \log_2 n$ を整数化した方法がスタージェス法と言われています。(ii) ではデータ数 n の3乗根とその標準偏差 $\hat{\sigma}(x)$ をつかい、調整定数をかけた $h = \dfrac{7}{2}\dfrac{\hat{\sigma}(x)}{\sqrt[3]{n}}$ を階級の項数として計算したものです。(iii) は四分位数 Q_1, Q_3 の差、Interquantile を $\mathrm{IQR}(x) = Q_3 - Q_1$ として $h = 2\dfrac{\mathrm{IQR}(x)}{\sqrt[3]{n}}$ としたもので、正規分布のような場合には $\dfrac{7}{2}\hat{\sigma}(x) \fallingdotseq 2\mathrm{IQR}(x)$ となります。

2.2 確率分布の作成コマンド

- 二項分布 Binom(n,p) ; (random number of binomial distribution) `rbinom(12, size=5, p=0.8)` 引数は個数1 2、試行回数5、成功確率0．8を意味するもの。
- 一様乱数 (連続型) Uniform(0,1) ; (random number of uniform distribution) は `runif(120)` 個数が120個の単位区間 [0,1] の擬似乱数を生成する。
- 標準正規分布 N(0,1); (random number of standard normal distribution) `rnorm(120)` は平均 0, 標準偏差 1 (分散 1) の正規分布にしたがう 120 個の正規乱数を生成する。変数 x における密度関数の値 $\phi(x; \mu, \sigma^2) = \dfrac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$ を求めるには r の代わりに密度関数 (probability density) を意味する d として `dnorm(x, mean =` μ, `sd =` σ) とします。さらに q として `qnorm(x)` は、平均 mean $= \mu = 0$, 標準偏差 sd$= \sigma = 1$ の分布関数 (distribution function) $\Phi(x) = P(X \leq x) = \displaystyle\int_{-\infty}^{x} \phi(t; 0, 1)dt$ を求められます。
- 指数分布 (exponential); 乱数の生成は [r] をつけて、 `rexp(n=5, rate=`λ`)` : $P(X = t) = \lambda e^{\lambda t}, t > 0$
- 幾何分布 (geometric); 生成は `rgeom(n=5, prob=0.2)` : $P(X = k) = (1-p)^k p, k = 0, 1, 2 \cdots$, 確率

$p = 0.2$ の生起確率（k 回目に初めて成功が確率 p, 連続失敗の確率は $1-p$)、繰り返し個数は 5 回

• ポアソン分布 (Poisson); 生成は `rpois(n=5, lambda=3)` : $P(X=k) = \dfrac{\lambda^k}{k!}e^{-\lambda}, k = 0, 1, 2\cdots$, $\lambda = 3$ のとき、5 個生成。

以上の他に、統計数値表として、確率のパーセント点の計算には (i) t 分布 (Student のティー分布)、(ii) カイ二乗分布 (chi-square)、(iii) F 分布 (Snedecor のエフ分布)　に関する数表が必要です。この分布 (distribution) の意味の"d" を頭につけて、計算します。

(i) `>dnorm(x, mean =0, sd =1)` ; 平均を mean で、標準偏差を sd で指定する x での値

(ii) `>dt(x, df)` ; 自由度を df とする。密度は $f(x;n) = \dfrac{\Gamma((n+1)/2)}{\sqrt{2n}\Gamma(n/2)}\left(1+\dfrac{x^2}{n}\right)^{-(n+1)/2}, -\infty < x < \infty$

(iii) `>dchisq(x, df)` ; 自由度 df $= n$, 密度は $f(x;n) = \dfrac{1}{2^{n/2}\Gamma(n/2)}x^{n/2-1}e^{-x/2}, x > 0$

(iv) `>df(x, df1, df2)` ; 自由度 df1 $= n_1$, df2$= n_2$, 密度は

$$f(x;n_1,n_2) = \frac{\Gamma((n_1+n_2)/2)}{\Gamma(n_1/2)\Gamma(n_2/2)}\left(\frac{n_1}{n_2}\right)^{n_1/2}x^{n_1/2-1}\left(1+\frac{n_1}{n_2}x\right)^{-(n_1+n_2)/2}, x > 0$$

3　R のデータ分析

`https://to-kei.net/r-beginner/` 「全人類がわかる統計学」より

3.1　平均値、中央値、最頻値

データのサイズ : 行データ、列データの大きさを確認する `nrow(xdata)` xdat の行 (row) データ数, `ncol(xdata)` xdat の列 (column) データ数データの列名と行名 : 行データ、列データの名前を確認する `rownames(xdata)` xdat の行名 `colnames(xdata)` xdat の列名,

```
> matA + matB   # a_{i,j} + b_{i,j}
> matA - matB   # a_{i,j} - b_{i,j}
> matA / matB   # a_{i,j} / b_{i,j}
> matA %*% matB   # \sum_j \{a_{i,j} * b_{j,k}\} , 行列の積は 行と列のサイズに注意
> solve(matA) # Inverse(matA)

# xdata  の入力
> xdata <-c(12, 14, 25, 53, 6, 91, 25, 89, 77, 34)
# スクリプト（編集プログラム）に保存したデータの読み込み
> source("c:ファイル名.R")
```

命令	内容
mean(xdata)	xdata の平均値
var(xdata)	分散
sd(xdata)	標準偏差
median(xdata)	中央値
sum(xdata)	合計値
max(xdata)	最大値
min(xdata)	最小値
rev(xdata)	ベクトルを逆順にする。
order(xdata)	ベクトルの要素を小さい順に見たときに、何番目なのかを出力
sort(xdata)	ベクトルを小さい順（昇順）に並べる
rev(sort(xdata))	組み合わせてベクトルを降順に並べる

3.2 分散、標準偏差、四分位数

もし新しく処理を組み合わせて、 関数名 <- function(引数){関数内での処理} と定義できますから、たとえば、積和では

```
sekiwa <- function(xdata, ydata){sum(xdat * ydata)}   # sum(xdata[i] * ydata[i])
```

と定義できます。

```
> dat2
       a  b  c
 [1,]  1  1  2
 [2,]  2  5  4
 [3,]  3 23  6
 [4,]  4  6  8
 [5,]  5  4 10
 [6,]  6  2 13
 [7,]  7  9  8
 [8,]  8  8  2
 [9,]  9 10 19
[10,] 10  6 29
> dat2[5]
[1] 5
> dat2[,3]
 [1]  2  4  6  8 10 13  8  2 19 29
> cor(dat2[,2], dat2[,3])
[1] -0.0456211
> sum(dat2[,1])
[1] 55
>
```

左図では、入力データ３０個＝１０×３の一覧から、
dat2[5] は５番目のデータで、 dat2[,3] は ３列目（c 列）を取り出しています。
上側は縦の列ですが、横ベクトルに並んだデータを取り出しています。
また関数命令では、
\verb+ cor(dat2[,2], dat2[,3]) + は b 列と c 列の相関係数
(correlatin) を計算しています。さらに
\verb+ sum(dat2[,1]) + は a 列の和 (summation) を計算。
\includegraphics[width=8cm]{r3.eps}

つぎは行列の積は %*% という命令が対応して、ベクトルの内積です。結果（２乗の積）を確認してくだ

さい。

```
> matC
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> matC%*%matC
     [,1] [,2] [,3]
[1,]   30   36   42
[2,]   66   81   96
[3,]  102  126  150
> 1*1+2*4+3*7
[1] 30
```

これまでは簡約な部分のみを取り上げていますし、まだまだ多くの機能などがありますので、インターネット等で調べてみてください。専門性を高めた公開プログラムもありますし、また https://ja.wikipedia.org/wiki/R 言語 では、

ユーザープログラムを配信・利用できるCRANネットワーク機能 [編集]

- 世界中のRユーザが開発したRプログラム（ライブラリ）（これを「パッケージ」と呼ぶ）が**CRAN** (The Comprehensive R Archive Network) と呼ばれるネットワークで配信されており、それらをR環境単独でオンラインでダウンロード・インストール・アップグレードと一連の管理が可能である。**R-Forge**等の他のサーバーも設定できる。CRANはRにシームレス統合されているため利用可能な機能（基本機能・オプションプログラムの両方）は日々増加拡張している[5]。（「パッケージ」・「最近の展開」を参照のこと）

教育現場から実務・研究現場へ永続的に利用可能 [編集]

- マルチプラットフォーム・オープンソースで無償であるため誰もが同一作業環境を構築できる
- 「命令の文法が単純である」・「高水準な統計解析と視覚化機能・永続的な利用に耐える」などの理由で教育機関において統計学教育や統計処理を必要とする講義で利用し易いうえにプログラミングに手間取る事なく統計解析の教育・学習に専念できて解析のプロフェッショナルな道具であるので学習スキルは後々も実践で活かせる（「プログラムの入手」・「持続可能な統計環境」・「最近の展開」を参照のこと）

と書かれていますから、取り組んでみてください。