

CONTENTS

【コラム】新しい学びの機会を広めたい—小学生とプログラミング—… 田島 篤

【解説】情報教育と国語教育の連携を考える… 阿部 圭一

【解説】情報教育と統計教育 No.1 Rによる1行プログラミング… 奥村 晴彦

COLUMN

新しい学びの機会を広めたい —小学生とプログラミング—

基
般

プログラミングという新しい学びの機会を、より多くの子供たちに届けたい——。こうした考えに基づき筆者の所属する日経 BP 社では小学生向けのプログラミング本を相次いで発行した。具体的には『小学生からはじめるわくわくプログラミング』（以下わくプロ、図-1）を2013年7月に、『Raspberry Pi ではじめるどきどきプログラミング』を2014年2月に発行^{☆1}。わくプロは、米 MIT メディアラボが開発したビジュアルプログラミング環境「Scratch」を用いた小学生向けの自習書で、順調に版を重ねている。

わくプロの企画段階では、同社にとって大きな挑戦となる小学生向け書籍の発行を危惧する意見も多かった。それでも発行に踏み切れたのは、著者の阿部和広先生が10年以上にわたって小学生を対象にしたプログラミングのワークショップを実施しており（図-2）、その実践の場から得られた知見を書籍に盛り込んでもらえるかと確信したからだ。

一般に入門書では、限定した内容を平易に書くことに注力しがちである。対してわくプロの場合は、子供に本当に分かってもらえる丁寧な書き方に加え、ワークショップで実際にウケるポイント、理解するのが難しいポイントをしっかり押さえた内容にしていた。また、子供が紙面を通じて学びの場を疑似体験できるように、本書の主人公であるキャラクターが試行錯誤しながらプログラムを作り上げていく過程を盛り込んだ構成にしていた。

このように著者が実践の場を通じて長年培ってきた知見を紙面に反映できたことが、さらには著者の強い要望に応じてフルカラーの書籍ながら本体価格を1,900円に抑えたことが、広くご利用いただけている理由だと考える。そしてもう一つ、著者が何より重視していたのが「わくわく感」。ページをめくることが楽しくなる紙面こそが、書籍で

学び続けてもらうために不可欠な要素ではないだろうか。

読者からは「もっと続きをやりたい」という声をいただいている。これに応えるためには、継続的に学びの機会を提供していくことが大きな使命になる。阿部先生に倣って、継続的な取り組みとなるよう、書籍の枠を越えた活動をしていきたいと考えている。

田島 篤（日経 BP 社）



図-1 『小学生からはじめるわくわくプログラミング』



図-2 ワークショップの様子

☆1 ほかに『小学生から楽しむ Ruby プログラミング』（2014年7月発行）、『5才からはじめるすくすくプログラミング』（2014年10月発行）、『スターディノではじめるうきうきロボットプログラミング』（2014年11月発行）がある。

情報教育と国語教育の連携を考える

阿部圭一

(元静岡大学/愛知工業大学)

情報教育と国語教育との目標の重なり

情報教育と国語教育は、図-1に示すように一部重なった目標を持つ。目標が重なるということは、教育内容も重なる可能性や、同じ目標に対して異なる側面からアプローチする可能性がある。本稿では、両者がどのように重なっているかを検討し、その重なり部分に対して情報教育と国語教育の間でどのような連携が考えられるかを考察する。ただし、

- (1) ここで言う「情報教育」は「情報処理教育」よりも広い概念である。両者の関係については後に述べる。
- (2) 国語教育には、小中高校だけでなく、大学において「日本語表現」などの科目名で開講されている日本人に対する日本語教育を含む。

では、まず情報教育と国語教育の重なりを巨視的に見てみよう。現行の学習指導要領小学校国語の目標は次のように規定されている。

国語を適切に表現し正確に理解する能力を育成し、伝え合う力を高めるとともに、思考力や想像力及び言語感覚を養い、国語に対する関心を深め国語を尊重する態度を育てる。

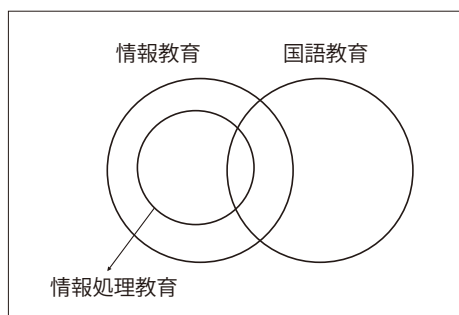


図-1 情報教育と国語教育の目標の重なり

中学校国語の目標もほぼ同じ記述である。

かつては国語の目標は文学鑑賞と日本語の基本運用能力に重点が置かれていた。左下前述の学習指導要領の目標を見て分かる通り、現在は表現・理解・伝達というコミュニケーションの基本能力の育成にもかなり重きを置くようになっている。これは、現行の1つ前(2002年度から実施)の学習指導要領から顕著である。実際、国語において高いシェアを占める光村図書の教科書では、特に小学校でコミュニケーションにかかわる活動に多くの授業時間を当てている^{☆1}。話し合う、調べて報告する、考えをまとめて書いたり発表したりする、など。

もちろん、国語教育における表現・理解・伝達とは、主に日本語を用いてのという限定がある。一方、情報教育において扱われるコミュニケーション能力は、あらゆる情報の表現・理解・伝達に開かれたものである。しかし、我々のコミュニケーションの大きな割合は言語すなわち日本語を介して行われる。したがって、国語教育と情報教育とはコミュニケーション能力の育成という分野で目標をほとんど共有していることになる。両者の連携が必須であるゆえである。

1つの事例

情報教育と国語教育の重なりを示す良い例が、中学校の国語の中で、優れた単元学習を永年続けたこ

^{☆1} http://www.mitsumura-tosho.co.jp/kyokasho/s_kokugo/tangen/index.html/

	授業での作業	情報教育としての見方
(1)	1学年の国語教科書丸1冊の中から「ことば」という単語の用例をすべて1枚ずつ別のカードに抜き書きする	—
(2)	「ことば」という単語の使われ方や意味が同じものを集めて、分類したカードの山を作る	KJ法の一段階 差異とか、それに基づく分類とかの概念をおぼろげながら得る
(3)	カードの山のそれぞれに、それらのカードに共通する意味をニックネームとして付ける	情報の超要約作業 適切な見出し・ファイル名・フォルダ名・メールの件名を付ける作業と共通する

表-1 大村はまの授業とその情報教育としての見方

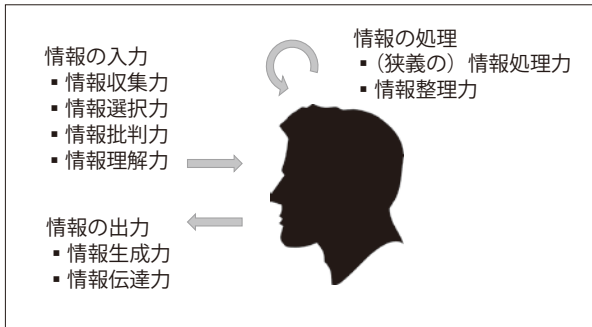


図-2 情報を「扱う」ために必要な能力

とで有名な大村はまの授業の1つにある^{☆2}。この授業で大村は表-1の左欄に示す作業を生徒に行わせた。この作業によって、生徒一人ひとりが私家版の辞書の1項目を作ったことになる。

この授業実践は情報教育と見ても通用すると、筆者は考える。表-1の右欄には、作業(2)、(3)を情報教育として見た内容を示す。

この授業は典型的な一例に過ぎないが、情報教育とも国語教育とも捉えることができる授業実践は多くあるだろう。以下では、この重なりへのアプローチを両方の側から検討した上で、両者の連携の在り方を考える。

情報教育の側からの考察

ICT（情報通信技術）を駆使した情報処理能力の育成という意味での情報処理教育は、情報教育の一部でしかないというのが筆者の考えである。人が情報を「扱う」ためには、情報の入力・処理・出力の

☆2 大村はま・荻谷剛彦・荻谷夏子：教えることの復権，ちくま新書，p.038，(2003)

各面で、図-2に示す多様な能力を総合的に伸ばす必要がある。これらの能力はICTによって支援される場合もあれば、人手によったり、ICT以前の情報を「扱う」能力が重要となったりする場合もある。

言語、日本人にとって日本語は、情報の中で大きな割合と役割を占める。したがって、日本語で表された情報に対する上に挙げたさまざまな能力は、日本語を「話す」「聞く」「読む」「書く」能力と大きな共通部分を持つ。さらにはその背後にある「言語を用いて考える」能力とも。これら5つの能力の育成は国語教育の目的にほかならない。

豊福晋平は、ICTを単なる一斉授業のための補助的道具でなく、学習者主導の知的生産能力の育成に活用する方向を提示している¹⁾。情報教育を、このような知的生産能力の育成のレベルで考えるならば、国語教育との重なりはさらに大きくなる。

国語教育の側からの考察

筆者は、やや総花的な表現である学習指導要領国語の目標をより具体的に解釈し、国語教育が大きく次の3分野に分けられると考えている。

- (1) 文学作品の鑑賞および創作
- (2) 日本語という言語の理解およびその基本的運用能力
- (3) コミュニケーションの手段および思考力の基礎としての日本語能力

人と人との間のコミュニケーションには、情緒的・感情的側面も含まれるが、それは強引に(1)に含めるとして、(3)でのコミュニケーションは、「事



1. 基礎的な言語能力	
2. 「型」に合わせる能力	
3. 情報をうまく伝える能力	3a. 伝える内容を選ぶ能力 3b. 伝える表現を選ぶ能力
4. 情報機器を使う能力	

表-2 野田尚史による日本語能力の分類

ソーシャル・メディアでのテキスト	まとまった文章
コミュニケーション自体が目的（自己完結的コミュニケーション）のことが多い	何らかの用件でのコミュニケーション（道具的コミュニケーション）
短く、内容も単純	目的を達成するだけの長さ、内容
感覚的・感情的	説明的・論理的
相手からの頻繁なフィードバック	フィードバックがあるとは限らない
文脈共有・文脈依存性が高い	文脈共有・文脈依存性が低い

表-3 ソーシャル・メディアでのテキストとまとまった文章との傾向の違い

実と意見」に代表される、論理的な情報伝達に限るものとする。言わば思考力に裏打ちされたコミュニケーション能力である。

情報教育との重なりが生じるのは、(3) においてである。つまり、(3) におけるコミュニケーションとは「情報の伝達」であり、その前提として発信者の「情報表現」の能力・技術と、受信者の「情報理解」の能力・技術も同時に問われるからである。

情報表現としての日本語技術に関して、野田尚史は話し方と書き方のリテラシーを表-2に示すように分類した²⁾。その上で、日本における日本語リテラシー教育は、1と2に偏っていて、3と4が弱いと論じている。筆者はこの意見に賛成である。

情報教育と国語教育の連携の在り方

では、情報教育と国語教育との間でどのような連携が考えられるだろうか。

1つには、国語教育の中で、(3) のコミュニケーションの手段としての日本語能力の育成にかかわる単元の目的や内容を、情報教育の立場から考えたり援助したりすることであろう。逆に、情報教育の中で日本語を扱う題材を国語教育の立場から検討して助言をもらうのもよい。具体例は後に述べる。

また、集団で1つの成果物を作るという授業体験も望ましい。どの教科の中で行うかという問題はあ

るが、ブレインストーミング、ディベート、共同作業による1つの文書の作成など。こういう活動の中で、いかにICTを活用するかという点について、情報教育の側から助言ができるとうい。

他面では、情報教育と国語教育との間の役割分担や補い合いも考えなければならない。インターネットとスマートフォンの普及によって、小中高生の言語環境が激変していることにも留意する必要がある。

橋元良明は、各種メディアの利用時間の経年変化データから、若者の文字離れは起こっておらず、テキスト系情報行動の時間はむしろ長くなっていると言う³⁾。しかし、その内容が短文化された単純な文章のほうへ移行していることは、客観的データはないものの、事実であろう。ソーシャル・メディアにおける短文中心のテキストと、社会において必要とされるある程度まとまった文章との傾向の違いを表-3にまとめた。このような違いの理解に立って、まとまった論理的な文章の読み書きが教えられる必要がある。

この状況を背景に、日本語文章を読む技術については、速読と熟読の双方を訓練しなければならない。速読、すなわち日本語で書かれた膨大な記述の中から自分にとって必要な情報を素早く見出しつかみとるには、情報スキミング・情報スキミングの技術が役立つ。他方、熟読の機会が減っていることについて、Patricia M. Greenfieldは文献4)の結論で次

のように述べている。

テレビ、ビデオ・ゲーム、インターネットが持つ視覚的特性によって視覚的な知性がかかり発達するかもしれない。しかし、その代償として、深いレベルでの処理、すなわち、注意深い知識獲得、帰納的な分析、批判的思考、想像力、熟考が失われるように思われる(筆者訳)。

Maryanne Wolf と Mirit Barzillai は、オンラインで熟読を行う学習機会を多くすべきだと提案している⁵⁾。そのための教材の開発は、情報教育者と国語教育者が共同で取り組むべき課題である。

書くことについても、日ごろ LINE, Twitter など書いているその場限りの短文ではない、ある程度まとまった論理的な文章を書く機会をたくさん設けるべきである。

その種の文章の書き方を教えるのが難しい理由の1つは、文章を書くためには一度に並行していくつもの作業をしなければならないからである。向後千春は、その対策として、こうした複数の作業を分割して取り組んでいくという戦略をとることを勧め、具体案を示している⁶⁾。このような作業の中で、ワープロソフトのアウトライン機能や Mind Map, 文章チェック・ソフトのようなツールの活用が考えられる。また、それらの背後にある情報科学的な考え方(たとえば階層化、構造化)も参考になるだろう。

さらに、次のような試みも検討されてよい。

- メール本文を与えて、適切な件名を付けさせる(超要約作業の実習)。

- 会議やイベントの案内文、簡単な報告書、提案書、依頼文などの不十分な例を示し、抜け落ちている情報(表-2の3a)やあいまいな点(表-2の3b)を指摘させる。

- プレゼンテーションを行わせた後、それを(説明用のメモではなく)1つの独立した文書として文章化させる。それによって次の2点を考えさせる。

- (1) 視聴覚を用いた提示と文字だけ(図表の挿入は可)による提示との違い
- (2) 受け手がその場に居る場合(同期的)と、受け手がその場に居ない場合(非同期的)とのコミュニケーションの違い

情報教育と国語教育の重なるの核心には、「情報の整理の仕方・まとめ方」があると筆者は思う。これが ICT の進歩や社会の進展に伴って変化することにも注意すべきである。

参考文献

- 1) 豊福晋平：日本の学校教育情報化はなぜ停滞するのか、情報処理, Vol.56, No.4, pp.316-321 (Apr. 2015).
- 2) 野田尚史：話し方と書き方のリテラシー, 日本語学, Vol.28-2, pp.24-33 (2009).
- 3) 橋元良明：今の日本は、簡単に世論統制ができてしまいかねない危険な状態がある, 長澤秀行 編著, メディアの苦悩, 光文社新書, p.250 (2014).
- 4) Greenfield, P. M. : Technology and Information Education : What is Taught, What is Learned, Science, Vol.323, pp.69-71 (2 Jan. 2009).
- 5) Wolf, M. and Barzillai, M. : The Importance of Deep Reading, Educational Leadership, Vol.66, No.6, pp.32-37 (2009)
- 6) 向後千春：200字の法則 伝わる文章を書く技術, 永岡書店 (2014).

(2015年3月11日受付)

阿部圭一 (正会員) kei1.abe@nifty.com

1968年名古屋大学大学院工学研究科博士課程満了。工学博士。静岡大学工学部、情報学部を経て、2006年4月から2013年3月まで愛知工業大学教授。



Rによる1行プログラミング

奥村晴彦

三重大学教育学部

情報教育と統計教育の接点を扱う3回シリーズの第1回として、話題のR言語を使った情報+統計教育を紹介する。

Rによる1行プログラミング

問 正規分布の乱数を百万個作り、配列xに入れよ。

答 `x = rnorm(1000000)`

問 xの度数分布図を描け。

答 `hist(x)`

(ちょっと寂しいので色を付けよう)

`hist(x, col="gray")`

これは、R言語を使った私の授業でウォーミングアップ的に行う小課題の一例である。次ページ図-1にRStudio (Rのフロントエンドの1つ)で上のことを実行した画面を示す。

注目すべき点は、ほとんどの作業が1行プログラムでできてしまうこと、そして、1行1行の効果が即時に得られることである。エラーが出ても↑キーで履歴を遡って修正できる。

同じことをCやJavaの類で行うのはとても面倒である。Excelでも(「百万」が「百」であっても)まともな度数分布図を描くには^{かみ}ネ申ワザが必要である。R言語では、上の程度のことなら、オマジナイ(`#include <stdio.h>`や`import matplotlib.pyplot as plt`の類)を唱えることなく、掛け値なしの1行プログラミングが可能である。

対話型の環境ということなら、Rに限らず、多数

存在する。古い人ならBASICのダイレクトモードを思い出すであろうし、Ruby (irbなど)、Python (特にIPython)もそうである。これらの中でRの特徴をより詳しく説明するために、次の問題を考えてみよう。

FizzBuzz問題：従来型解法

1から100までの整数を唱えよ。ただし、3の倍数ならFizz、5の倍数ならBuzz、両方の倍数ならFizzBuzzと唱えよ——これが有名なFizzBuzz問題である。これをRで解いてみよう。おそらくCやJavaを習った学生なら次のようなプログラムを考えるであろう(%%は剰余演算子、cat()は出力):

```
for (i in 1:100) {
  if (i %% 15 == 0)
    cat("FizzBuzz\n")
  else if (i %% 3 == 0)
    cat("Fizz\n")
  else if (i %% 5 == 0)
    cat("Buzz\n")
  else
    cat(i, "\n")
}
```

Rでこれを実行するには、テキストエディタで作っておいてRの入力画面にコピーするか、`fizzbuzz.R`といった名前でも保存してファイル名を指定して実行する。RStudioなら、ソースエディタの選択された部分を実行する機能がある。OSによっては、`fizzbuzz.R`の先頭に

```
#!/usr/bin/env Rscript
```

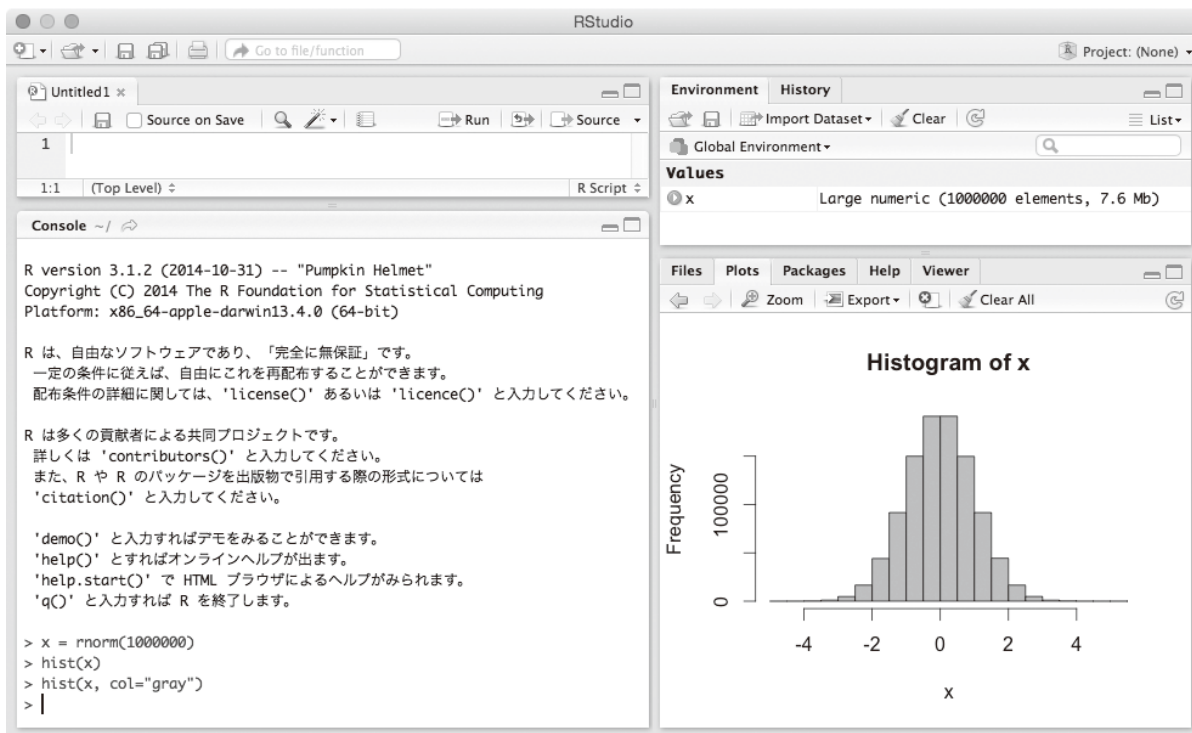


図-1 RStudio に本稿冒頭の例を打ち込んだところ

と書いて実行許可を与えれば、単独で実行できるスクリプトになる。

FizzBuzz 問題の 1 行プログラミング

より R 的な方法は、たとえば次のように、1 行プログラムを積み重ねて解くことである。ここでは対話型環境に 1 行ずつ打ち込むことを強調するために、R のプロンプト `>` を明示的に書いている：

```
> a = 1:100
> a[seq(3,100,3)] = "Fizz"
> a[seq(5,100,5)] = "Buzz"
> a[seq(15,100,15)] = "FizzBuzz"
> a
```

1 行プログラミングの特徴として、どの段階でも、上の例では `a` というベクトル (配列) に、途中段階の解が入っていることである。最後の行のように、R のコンソールに `a` とだけ打ち込めば、全要素が表示される。試しに 1 行目を打ち込んだ直後に `a` と打ち込めば、

```
> a = 1:100
> a
[1] 1 2 3 4 5 6 7 8
[9] 9 10 11 12 13 14 15 16
[17] 17 18 19 20 21 22 23 24
...
```

のように出力される (改行位置は端末の幅に依存する)。左端の `[]` 内は各行の先頭の要素の番号 (1 から始まる) である。これで `1:100` は 1 から 100 までの整数のベクトルであることが分かる。

`seq(from, to, by)` は、初項・上限・公差を与えて等差数列のベクトルを作る関数である：

```
> seq(3, 100, 3)
[1] 3 6 9 12 15 18 21...
```

したがって、

```
> a[seq(3,100,3)] = "Fizz"
```

は `a[3]="Fizz";a[6]="Fizz";...` と同じことで、配列 `a` の 3 の倍数番目の要素に "Fizz" を代入する。

最終的に上のプログラムの出力は次のようになる (整数型が文字列型に変換されたことに注意する)：



```
> a
[1] "1"      "2"      "Fizz"   "4"
[5] "Buzz"   "Fizz"   "7"      "8"
...
```

明示的に1行ごとに出力したいなら、次のようにする:

```
> cat(a, sep="\n")
1
2
Fizz
...
```

このように、Rの基本データ型はベクトル(配列)であり、スカラはベクトルの特別な場合(長さ1の場合)である。ベクトルデータの作り方はいろいろあり、上の `a = 1:100` もその1つであるが、次のように要素を列挙してもよい:

```
> x = c(3.14, 2.718, 0.577)
> 初夢 = c("富士", "鷹", "茄子")
```

変数(オブジェクト)名には日本語も使える。

ベクトルをうまく使うと、かなりの for ループや if 文を隠ぺいでき、場合によっては高速化が可能である。

文字列処理

行ごとに収めた単語を読み込み、「全角」英数字を「半角」に直し、英字を小文字に揃え、出現頻度順に10位まで出力したい。これも1行プログラムの並びで解くことができる。

2~3行目は、本誌紙面に合わせるため、途中で改行している。Rは文法的に完結しない行は継続すると見なす。継続行のプロンプトは+である。

```
> x = readLines("words.txt")
> x = chartr("0-9A-Za-z",
+           "0-9A-Za-z", x)
> x = tolower(x)
> s = sort(table(x), decreasing=TRUE)
> head(s, 10)
```

`readLines()` は、ファイル全体を読み込み、各行をベクトルとして返す。`chartr()` は、Unix

◆ R言語の代入は <- か = か

R言語(およびその前身のS言語)の代入は、もともと `x <- 3` または `3 -> x` のように書いたが、1998年のS Version 4から `x = 3` という書き方もできるようになった。Chambersの本 *Programming with Data: A Guide to the S Language* (1998) では = が主に使われている。R言語では2001年から = も使えるようになった。現在でも <- を使う人が多い(例: “Google’s R Style Guide”)が、本稿では = を使っている。

<- を使う理由として、= は関数の引数リストの中では名前付き引数を指定する = と衝突することが挙げられる。たとえば `mean(x = var)` では `x` は引数名と解釈される。もっとも、引数リストの中での代入は、遅延評価のため予想外の結果になることもあり、推奨されない。

の `tr` コマンドと同様に、対応する文字に変換する。`tolower()` は大文字を小文字に変換する。これらの関数はすべてベクトルに要素ごと作用する。`table()` は度数分布表を作り、`sort()` は整列する。`head()` は、Unixの同名コマンドと同様に、先頭だけを取り出す。

上の例では使っていないが、POSIX互換またはPerl互換の正規表現も使える。

また、文字列処理ではないが、上で求めた度数分布 `s` のエントロピーを求めるのも簡単である:

```
> p = s / sum(s)
> -sum(p * log2(p))
```

Rとは?

ここで改めてRを紹介する。

Rは、オープンソースの言語処理系である。対話形式で使われることが多い。特に統計・データ解析、統計グラフ作成に強い。追加パッケージによって機能拡張できる。公式パッケージだけで6,000種以上が公開されており、XML、JSON、Excelファイルの読み書きや、多倍長演算など、多くのことが簡単なコマンドで行える。

Rの元となったSは、1976年から、AT&T Bell研究所の統計学者たちによって開発された。Sは商用のS-PLUSに発展した。一方、オープンソースのRは、1990年代に、Auckland大学のRoss IhakaとRobert Gentleman（2人のR）によって実装された。現在は、彼らを含むR Core Teamによって開発が続けられている¹⁾。

Rは内部的にはSchemeに近い関数型言語である。ChambersはRの特徴をfunctional OOP（関数型オブジェクト指向プログラミング）と呼んでいる。

ソースコードのほか、Windows, Mac, Linux用のインストーラが配布されている。Windows用・Mac用のものには専用の実行環境が付属しているが、ターミナルに「R」と打ち込んでも使える。最近ではRStudio^{☆1}というフロントエンドがよく使われる。Emacsからも使える(ESS)。IPython Notebookからも使える(rpy2)。

学生用のPCで自由にRやRStudioがインストールできない場合は、別のLinuxサーバでRとRStudio Serverを動かしておけば、ブラウザから使える。

ヘルプは対話型環境内で参照できる。あるいはコマンドでたとえば

```
> ?rnorm
```

と打てばrnorm()のヘルプが表示される(現状では英語)。

.....
☆1 <http://www.rstudio.com>

情報と統計を融合した授業例

筆者は、かねてから情報処理と統計・データ解析とを融合した教育を模索していた²⁾。2009年度からはRを使った統計学の授業の中で試行錯誤を重ねている³⁾。適当な教科書がないので、教材はすべて自作し、Webで公開している⁴⁾。授業はPC教室(Windows)で行っている。最近ではR標準のユーザインタフェースとRStudioフロントエンドから好きな方を選ばせているが、ほとんどの学生がRStudioを使っている。

簡単な計算から、平均・標準偏差・相関係数などの統計量の計算、直線や曲線のあてはめ、*p*値や信頼区間の計算、グラフ描画、乱数を使ったシミュレーションなど、多様な実習を通じて、データ処理から統計学の基本的な考え方までを教えている。

Hello, world! を出力するのに何行も必要な実習と比べて、たった1行のプログラムでも意味のあるデータ処理ができるので、最近のデータサイエンスブームも手伝って、学生のモチベーションは高い。

参考文献

- 1) R Core Team: R: A Language and Environment for Statistical Computing, <http://www.R-project.org>
- 2) 奥村晴彦: 情報教育と統計, 情報処理学会研究報告コンピュータと教育, 2008-CE-97, pp.81-88 (2008).
- 3) 奥村晴彦: Rを使った情報教育, 情報処理学会 SSS2010 論文集, pp.77-80 (2010).
- 4) 奥村晴彦: 統計・データ解析, <http://oku.edu.mie-u.ac.jp/~okumura/stat/>

(2015年3月31日受付)

奥村晴彦 (正会員) okumura@okumuralab.org

三重大学教育学部教授(情報教育)。LHAデータ圧縮アルゴリズムの開発者、『LaTeX2e美文書作成入門』(現在第6版)の著者。

