資　　料

Rによる基礎統計分析

Note on "R" for Social Data Analysis

中野　康人*
Yasuto Nakano

I　"R"について

本稿の目的は、統計解析のためのフリーソフトである"R"（Ihaka and Gentleman, 1996）の基礎的な使用方法を紹介することにある。日本の社会学者が統計解析を行う場合、現在ではその多くは市販の統計パッケージ（SPSSやSASなど）に依存している。各統計パッケージは、ユーザや統計学者の意見を反映し、年々進歩して計量研究には欠かせないものとなっている。そのような現状で、あえて"R"を紹介するのには理由がある。

一言でいってしまえば、それは"R"がGNUスタイルのフリーソフトであるという点に集約できる。入手に金銭的なコストがほとんどかからないこと、ソースが公開されていて自由に改編できること、世界中の「知」が集約することによってアップグレードしていくこと。これらが"R"がフリーソフトであるが故に持つ性格である。特に金銭的な面は、若手研究者や予算に制約がある教育現場などでは大きな魅力となるだろう。オープンソースであるという点は、統計解析のアルゴリズムをきちんと把握していたいユーザーを安心させるだろう。

統計解析のためのフリーソフトはいくつかあるが(高田, 1997)、ここで"R"を紹介するのは、"R"が市販の言語"S"と高い親和性持つからである。"S"は、統計分析の総合的な環境として名高い。"S"の為に用意された様々な資源（プログラムや書籍、文献など）を、ほとんどそのまま"R"に援用できる。

"R"については、CRAN（The Comprehensive R Archive Network）のweb page（http://wwwr-project.org/）に詳しい。"R"を全く知らない人には、上記web pageに公開されている"What is R?"というドキュメントの導入部分を紹介するのが理解の助けになる。

Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNUproject which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different

implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs out of the box on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux) . It also compiles and runs on Windows 9x/NT/2000 and MacOS.

上記の通り、"R"は高度な統計解析手法を駆使できるものであるが、本稿は、主に社会統計学の基礎的な分析を"R"上で行う方法を紹介する。

"R"の入手とインストールについては、上記の web page を参照されたい。いくつかの OS 上で動作するが、本稿では Linux 版の"R"(Version 1.3.1) を使用する。

Ⅱ 起動・終了とヘルプ

"R"を起動するには、コマンドライン上で

```
R
```

と打ち込むだけでよい。そうすると図のような状態になり、> (プロンプト) がコマンド入力を待つ状態になる。基本的に、ここにコマンドを打ち込んで行くことにより、様々な分析を実行する。終了時は、

```
> q()
Save workspace image? [y/n/c]: y
```

である。"Save workspace image? [y/n/c]: " とあるのは、入力したデータなどを保存するか尋

ねるもので、"y"とすれば保存、"n"は保存せずに終了、"c"は終了をキャンセルして入力画面に戻る。保存の場合、ファイルは"R"を実行したディレクトリ上に保存される（データは".RData"に、コマンドの履歴は".Rhistory"にそれぞれ保存される。）。

出力結果は、基本的に標準出力（画面）に出力されるが、

```
> sink ("filename")
```

でファイル"*filename*"に出力が保存されるようになる。標準出力に戻すときは、引数なしの

```
> sink ( )
```

である。

また、

```
> help.start (gui ="irrelevant", browser = "netscape", remote = NULL)
```

で、オンラインヘルプが起動（ヘルプファイルを読み込んだnetscapeが起動）する。この他に、help ("*function*") で関数"*function*" のヘルプが、apropos ("*key*") でキーワード"*key*" を含む関数の一覧が得られる。

Ⅲ　データ

"R"が取り扱うデータは、その入力方法や性質にいくつかの種類がある。しかしここでは、ケース×変数の固定長データの取り扱いについてのみ触れる。

新規にデータを入力する場合、

```
> data.entry ( )
```

で、スプレッドシート型の入力ウィンドウが開く。テキスト形式でデータファイルが存在する場合は、

```
> read.table ("filename", header=TRUE, sep=" ", dec=".", na.strings ="NA")
```

で、ファイル"*filename*" からデータを読み込むことができる。ファイルの形式は、スペースで区切られた固定長データで、一行に一ケース、各列が変数を構成し、一行目は変数名となる。この形式は、read.table ( ) 関数のオプションで変更できる。

欠損値は、文字列NAで表される。データを作成する際に、欠損値をあらかじめNAで表記しておけば、"R"はこれを欠損値として処理する。またオプションna.stringsで異なる文字列を欠損値として処理できる。

foreginパッケージを導入していれば、SAS、SPSS、S-PLUSなどのバイナリデータも読み込める。

データにヘッダがついていれば、それが各変数の名前となる。以下の分析では、基本的に変数に付けられた名前をコマンドの引数にすることより、分析が行われる。また"R"では、データ全体や一部の変数群にも名前を付けることができる。この機能を利用すると、全ての変数の単純集計をとりたい場合や、一部の変数群にまとめて処理を施したいときなどに、コマンドの入力が簡単になる。

データ全体に名前を付けるには、データを読み込む際に、

```
> dataname <- read.table ("filename", header=TRUE)
```

とすれば、データ名"*dataname*"が定義される。一部の変数に名前を付けるには、

```
> vargroup <- data.frame (variable1,variable2, ..., variablem)
```

とすると、変数"*variable1 ... variablem*"の変数群"*vargroup*"ができる。データ名、変数群の中の特定の変数を指定するには、"*dataname $ variablename*"という風に"$"を間に入れて記述する。

Ⅳ 単純集計

単純集計は、

```
> summary (variable)
```

で得られる。関数"*summary ()*"で得られる統計は、最小値、最大値、算術平均（mean)、中央値（median)、四分位数である。

度数分布が知りたい場合は、

```
> table (variable)
```

で、変数"*variable*"の度数分布が得られる。また、

```
> hist (variable)
```

で、変数"variable"のヒストグラムがプロットされる。高度な作図機能は、"R"の魅力の一つだが、コマンドをデフォルトで実行するだけでは実用にたえない。多くの場合、オプションなどで体裁を整えてやる必要がある。関数"*hist ()*"に関しても同様である。単純出力レベルでのグラフ出力としては、以下のようなものがあげられる。

グラフ	コマンド
散布図	plot ( )
対散布図	pairs ( )
棒グラフ	barplot ( )
円グラフ	piechart ( )
比率グラフ	mosaicplot ( )

V　クロス表とχ自乗検定

社会学者が行うデータ分析で、最も基礎的でかつ最も重要なものはクロス表分析である。"R"では、

```
> table (row-variable,col-variable)
```

で、クロス表を出力する。

さらに、

```
> summary (table (row-variable,col-variable) )
```

で、そのクロス表をもとにしたχ自乗検定が行われる。χ自乗検定に関しては、

```
> chisq.test (row-variable,col-variable, correct = TRUE,
                p = rep (1/length (x) , length (x) ) ,
                simulate.p.value = FALSE, B = 2000)
```

で、詳細なオプション設定ができる。"*row-variable*"がマトリクス（クロス表）で与えられると、"*col-variable*"は省略できる。"*correct*"を"*TRUE*"にすると、イェーツの連続性の補正が施される。

関数"*chisq.test ()*"のさらなるオプション（クラス）として、次のようなものがある。これらを、

オプション	出力
statistic	the value the chi-square test statistic.
parameter	the degrees of freedom of the approximate chi- square distribution of the test statistic, 'NA' f the p-value is computed by Monte Carlo simulation.
p.value	the p-value for the test.
method	a character string indicating the type of test performed, and whether Monte Carlo simulation or continuity correction was used.
data.name	a character string giving the name (s) of the data.
observed	the observed counts.
expected	the expected counts under the null hypothesis.

```
> chisq.test (x, y) $observed
```

のように関数の後、"$"に続けて打ち込むと、それぞれの数値が得られる。

デフォルトの関数"*table ()*"は、観測度数をもとにしたクロス表を出力するだけなので、あまり親切とはいえない。例えば、行パーセントや列パーセントなどを表したクロス表が必要とされることが多い。この点に関しては、坪田（1998）もしくはhttp://www.med.hiroshima-u.ac.jp/tech/saru/S/Welcome.html に、S言語用に書かれたクロス表関数が紹介されている。坪田（1998）に紹介されている関数をそのまま"R"に読み込むと、様々な体裁をもつクロス表を出力できる。

Ⅵ 相関

変数間の関連の測度に関しても、複数の関数が用意されている。最も基本的なピアソンの積率相関係数は、

```
> cor (variable1,variable2)
```

で、変数"*variable1*"と変数"*variable2*"の相関係数が計算される。変数名のかわりにdataframe（データ名、変数群名）を引数に指定すると、相関係数の行列が得られる。

相関係数の有意性検定を含めた、より詳細な情報を得るには、

```
> cor.test (variable1,variable2,
        alternative = c ("two.sided","less", "greater"),
```

method = c ("pearson" "kendall", "spearman"))

を用いる。"*alternative*"は帰無仮説検定にもとづいた検定の方法（片側か両側か）を指定する。"*method*"は、計算する相関係数の種類を指定する。順序変数同士の相関の場合は"*kendall*"を、間隔変数同士の相関の場合は"*spearman*"を指定する。

Ⅶ　おわりに

以上、社会統計学のごく基礎的な部分を"R"で行う手順を紹介した。市販の統計パッケージに慣れている人には、"R"による統計分析が繁雑なものに思われたかもしれない。GUIな操作体系が主流になっている現状では、コマンドを入力して行くCUIな操作体系を持つ"R"は、それだけで敬遠される理由を持つ。しかし、"R"の本領は、一般線形モデルや非線形回帰モデルなど、より高度な多変量解析およびその作図において発揮されるだろう。多変量解析に関する解説は本稿の範囲外であるが、デフォルトで用意されている関数だけでなく、世界中の利用者が持ち寄るパッケージや関数（先述したCRANのweb pageを参照）によって、市販の統計パッケージに劣らない分析が可能になる。分析手法の意味とプロセスをしっかりと把握しながら分析を行うには、うってつけの道具である。なお、"R"をweb serverと連動させることにより、ブラウザを利用したGUIな環境で統計分析を行うことができる（例えば、http://www.ism.ac.jp/~sato/index.htmlを参照）。こうしたシステムを構築するには、少なからぬ知識が必要であるが、CUIの問題を解決する手段が無いわけではない。

本稿をきっかけに、"R"の利用者が少しでも増えれば幸いである。

参考文献

[1] Ross Ihaka and Robert Gentleman. 1996."R: A Language for Data Analysis and Graphics,"*Journal of Computational and Graphical Statistics*,5（3）:299-314

[2] 坪田信孝.1998.『データ解析言語S』科学技術出版社.

[3] 高田洋.1997.「数理社会学のためのインターネット・リソース」,『理論と方法』.12（1）:103-111.

平成13年9月7日原稿受理　*社会学部