HOKKAIDO UNIVERSITY

Title	Rで改善する生態学のデータ解析
Author(s)	久保, 拓弥; 竹中, 明夫; 粕谷, 英一
Citation	日本生態学会誌, 58(3), 219-224
Issue Date	2008-11-30
Doc URL	https://hdl.handle.net/2115/35063
Type	journal article
File Information	rprog2008.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

[学術情報] R で改善する生態学のデータ解析[1]

久保拓弥 * ・竹中明夫 **・粕谷英一 ***

* 北海道大学大学院地球環境科学研究院

** 国立環境研究所

*** 九州大学理学研究院

R improves the data analyses in ecology

とは何か?

R はここ数年のあいだに生態学だけでなくデータ解析が必要とされる多くの学術分野で急速に普及してきた統計ソフトウェア (データ解析環境とも呼ばれる) である (R Development Core Team 2008)。この記事では、筆者たちが 2008 年 3 月の生態学会福岡大会で企画開催した自由集会「データ解析で出会う統計的問題 — R プログラミングの基礎」にそって、とくに (1) R はみかけほど難しくない、(2) R は生態学のデータ解析をより良いものにする、といったところに力点をおきつつ R を紹介していきたい。

なお、前記の「R プログラミング自由集会」に関するさまざまな資料、R のインストール方法、そしてこの記事で紹介する有用な文献・ソフトウェアなどへの最新のリンクは `http://hosho.ees.hokudai.ac.jp/~kubo/ce/` 以下にまとめているので、そちらも参照していただきたい。

多くの分野の研究者たちのあいだで R の利用が急速に拡大してきた理由は R が「完全に無料」というだけでなく「すごく便利なソフトウェア」であるからだ。しかしここではその便利さについて紹介する前に、まずは一般的な R の特徴から説明したい。R は特定のソフトウェア会社ではなく有志の統計学専門家・ソフトウェア技術者たちが開発している “free software” であり、誰でも無料で入手できる。無料で入手できるだけなら単なる freeware にすぎないのだが、R は free software であるところが重要であり、つまり R は GPL2 という free softoware ライセンスのもとで「自由にプログラムのソースコードを読める」「好きなように改造してよい」「自由に再配布してよい」といったことが許されている。生態学にかぎらず学術研究で使うソフトウェアにおいてとくに重要なのは「ソースコードを読んでよい」という部分だ。データ解析では統計学などにもとづく複雑な計算が必要とされることがよくある。しかし多くの統計ソフトウェアでは「実際どのように計算されているか」は隠蔽されていて、ユーザーはそれを知ることが難しい。開発もとのソフトウェア会社に質問しても正しい回答が得られる保証はない。このようなオープン性だけを見ても R が他の統計ソフトウェアに比べて格段にすぐれている。R が研究者間の「使いものになる」データ解析共通プラットホームとなっている理由のひとつである。

ソースコードが公開されているからといって、自分で何もかも読解する必要はない。誰か読める人をさがせばよいし、一番簡単にはその部分を書いた人に質問すればよい。R-help という R のメイリングリストではこのような質問・回答が毎日あたりまえのようにやりとりされている。

R と生態学研究

それではここからはとくに生態学研究におけるデータ解析で R が役にたちそうな側面を紹介していくことにしよう。

R はこれひとつで調査・実験データの整理・加工、統計モデルをつかったパラメーター推定計算、データや結果の作図をこなせるソフトウェアである。作図に関しては投稿論文の清書版図版水準も実現できる。また R を使うことでデータ解析者が「解析できること」を拡大する。たとえば、近年の生態学会大会の発表で一般化線形モデル (GLM; 久保・粕谷 2006

[1] この PDF ファイルは以下の記事の原稿段階のものです: 久保拓弥, 竹中明夫, 粕谷英一. 2008. R で改善する生態学のデータ解析. 日本生態学会誌 58: 219-224 (2008 年 11 月号)

など参照) を使ったデータ解析がふえてきた理由のひとつは、R の glm() 関数が手軽に使えるようになったため、なのかもしれない。

じつは R はほとんどすべての有料ソフトウェアより、「おおくのことができる」ソフトウェアである。これは誰でも R の拡張機能 package を比較的容易に作成でき、さらに他の R ユーザーが作ったこのようなアドオン package をインターネット上で共有できる CRAN (The Comprehensive R Archive Network) というしくみが存在するためである。CRAN にはざまざまな package が登録されていて、これらは日々更新され、誰でもダウンロードできるようになっている。つまり共通プラットホーム R 上で動作するユーザーたちの共有財産となっている。また CRAN に登録しなくても、ネット上に公開されてる R プログラムを利用したり、ユーザーどうしで自作の R プログラムをあげたりもらったりすることはよくある。

R には拡張機能 package を追加しなくてもひととおりのデータ読みこみ・作図・解析の機能はそなわっている。しかし CRAN にはいろいろな学問分野で開発された専門性の高い package なども含まれている。生態学研究者にとって有用そうな pacakge をいくつか紹介してみよう: 一般化線形混合モデルの推定計算をする glmmML; 群集生態学や植生学であつかう多変量解析の計算関数を提供している vegan; Center for Tropical Forest Science (CTFS) の 50 ha 森林調査地解析用の関数とデータ (の一部) を公開している CTFS; 空間相関を考慮した一般化線形モデルをあつかう geoRglm ……などなど興味ぶかい package はつきない。

R の準備、R のインターフェイス

R をインストールするためには R のインストーラーが必要である。これはインターネット上から最新のものを入手するのがよい。「R インストール」といった語で検索すると、たくさんの解説ページがみつかるだろう。

R のユーザーインターフェイス (UI) は基本的にコマンドラインによる操作となる。たとえば、このようなかんじのもので R のユーザーインターフェイス (UI) は基本的にコマンドラインによる操作となる。たとえば、画面に入力を促すプロンプト (R では '>') が表示された状態で、print (1 + 1) とキーボードから入力してから enter キーを押す (以下、本稿では、最初に > が書かれている行は人間が入力するコマンドを表す)。

```
> print(1 + 1)
```

すると、以下のように計算結果が画面に表示される、という方式である。

```
[1] 2
```

なお、この例では単に 1 + 1 だけ入力してから enter キーを押しても結果は同じである。

```
> 1 + 1
[1] 2
```

なぜ R の UI はコマンドラインによる操作を採用しているのだろうか? それはデータ解析とはそれなりに複雑なものであり (とくに生態学では複雑なデータ構造をあつかうことが多い)、マウス主体のグラフィカル UI (GUI) にすると、じつはかえって不便なものになる、といったことが理由である。たとえば巨大な table 形式データに対する複雑な操作をする状況を考えてみよう。いわゆる表計算ソフトウェアの GUI だと手と目を駆使した細心の作業が要求される。しかしながら、R の UI ならばちょっとコマンドを入力するだけでいとも簡単に処理できてしまうことも少なくない。

コマンドラインの UI はとりつきがたいものなのだろうか? 「R が初めての統計ソフトウェア」となる人たちを観察していると、UI が「かべ」になって R がまったく動かせない、となることは実際のところほとんどないように思う。ただし限定された操作だけなら R commander という GUI も利用可能ではある。

R でプログラミング?

あとで例をあげるように、生態学では観測・実験データを得てから試行錯誤で作図や統計モデリング (観測データにみられるパターンの説明) にとりくむ場合が多い。これを手作業でこなすのはまったくたいへんなことであり、コンピューターによる自動処理つまりプログラミングが有効なわざとなる。

R を動かす命令は S 言語とよばれるプログラミング言語の一種である。つまり R のコマンドライン UI で 1 行なにかを命じて結果出力をみる、という作業は「1 行プログラムを書いてその実行結果を出力させていた」ことになる。R のプログラミングとは「1 行ごとに命令→実行」の単なる拡張であり、(1) 別のファイルなどにあらかじめ「これをやって、次はこれをやって、最後に図をだして」と命令をまとめて列挙しておく (2) R がそれをよみこむ (3) R が書かれている順番に処理していく、だけのことにすぎない。

冒頭でのべた「R プログラミング自由集会」では、R 利用の中でとくに重要なわざである「大量の観測データの自動作図」 (竹中) や「R で推定した結果をとりだす方法」 (粕谷) などについてとりあげた。それぞれの要点を簡単に紹介してみよう。

R で自動作図プログラミング

R は、統計解析ツールであると同時に、作図ツールでもある。さまざまな作図用の関数が用意されている。統計解析の場合と同様に、それらの関数を一行づつ打ち込んで作図することもできるし、関数を書き並べたプログラムを作成して一気に実行することもできる。

プログラムによる作図のメリットは、一度プログラムを書いてしまえば同様のグラフを何枚作成しようとほとんど手間が増えないことである。違うデータセットを使って同じ形式の図を描くこと、同じデータセットで図の様式を少し変更すること、いずれも容易である。前者なら処理対象のデータファイルを変えるだけだし、後者ならプログラムの一部を編集すればよい。たとえば同じ様式のグラフが何枚もあって、全部線の太さを変えたいという場合、一枚一枚、マウスで選択して太さを選んで…という作業は大変だ。プログラム方式なら線の太さを設定しているオプションのところだけ修正して実行すれば、新しい設定のグラフが一気にできあがる。何かの締め切り間際になって、元データの一部の間違いが発覚しても、少なくともグラフ作りに関しては慌てる必要がない。

どのようなデータの解析も、まずは図にして全体のパターンを眺めることから始まる。そのとき、生のデータが多すぎて、すべてをそのまま図にするのは面倒だからと、予断を持って情報量を減らしてしまうのはよいことではない。たとえば、プロットしてみればまったく線形な関係にはないことが明らかな 2 変量について、何も見ないで直線回帰してしまい、その傾きや切片だけ見て議論するのは明らかにまずい。プログラムによる作図なら、3 枚のグラフを描くのも 300 枚のグラフを描くのも手間はほとんど変わらないから、億劫がらずに作図できるだろう。

```
pdf("all_figures.pdf")  # グラフを書き込む PDF ファイルを準備
par(mfcol = c(3, 3))    # 1 ページに 3 x 3 のグラフを配置

sites <- c("A", "B", "C")    # 3 つのサイトがある.
fert.dose <- c(1, 10, 100)   # 3 つの施肥量を設定した.

# 二重の繰り返し構文により, 3 x 3 = 9 個の処理区に対応した
# ファイル名  A_F001.txt, A_F010.txt, ...,  C_F100.txt を順次生成し,
# そのファイルを読み込んで植物のサイズのヒストグラムを描画する.

for (site in sites) {  #  各サイトごとの繰り返し
  for (fert in fert.dose) {  # 各施肥量ごとの繰り返し
    file.name <-  sprintf("%s_F%03d.txt", site, fert) # ファイル名を作る
    data <- read.table (file.name)  #  ファイルを読み込む
    # 読み込んだデータをヒストグラムとして描画する
    hist(data[,1], xlim=c(0, 30), main = file.name)
  }
}
dev.off()  # PDF ファイルを閉じる.
```

図 1. R の自動作図コード。記号 # から行末までは注釈であり R は自動的に読みとばしてくれる。

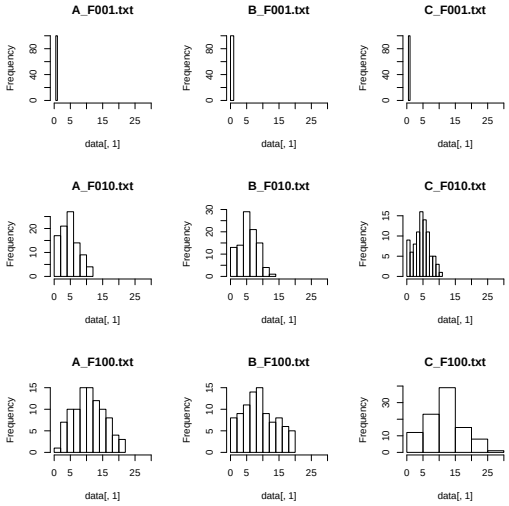

図 2. 図 1 のプログラムによって生成された図の一部

多数のデータから同じ形式の図をたくさん描く場合、いくつものファイルを順次読み込むなどの技を組み合わせることが必要になるが、こうした繰り返し作業はコンピューターのもっとも得意とするところである。プログラム言語としての R には、当然のように繰り返しのための構文が用意されている。R プログラムの例を示してみよう。この例 (図 1) では 3 つのサイト× 3 つの肥料処理区に対応する全部で 9 個のデータファイルを読んでそれぞれヒストグラムを描いている。一度に生成された 9 つのヒストグラムを図 2 に示している。

R による作図のとっかかりは、`plot()` という関数にデータを渡してみることだ。データの構造に合わせて適当なグラフが描かれるだろう。そのあとは、描画関数にさまざまなオプションを指定して欲しい図に近づけていけばよい。`plot` のほかにもデータセットを渡せば賢く作図してくれる関数 (高水準描画関数と呼ばれる) はいろいろある。上の例では、ヒストグラムを描く高水準描画関数 `hist()` を使っている。素朴に線を引いたり点を打ったりといった関数 (低水準描画関数と呼ばれる) もある。これらを組み合わせれば、お仕着せでないグラフを自由に作ることもできる。また、3D グラフを含め、いろいろなグラフ作成関数の package もある。

今回の自由集会の竹中の投影資料 (自由集会ページからダウンロード可能) やその詳細な web 版にあたる「R でプログラミング：データの一括処理とグラフ描き」(`http://takenaka-akio.cool.ne.jp/doc/r_auto/index.html`) も参考になるだろう。

R の `glm()` 推定結果のデータ構造

次に R の中でオブジェクト (データのかたまり) のとりあつかいについて簡単な例で説明してみよう. データ解析をしていると、『この方法はこんな状況のときに使っていいものだろうか』とか『こういうときにはこの方法が良さそうな気がするのだが … 』などと、統計的方法が適切かどうか迷うことがある。なかには、過去に検討されていてすでに結論が出ているものもあるだろうが、むしろ文献やインターネット上の情報では発見できないことも多い。また、答らしきものが見つかっても、よいとするものとだめとするものと両方があって判断に迷うこともある。そういうとき、自分 (あるいは自分のデータ) が直面している状況ではその手法が適切かどうか計算して調べれば、出てきた結論にも確信が持てるし精神的にもすっきりする。さらに、もしかすると方法についての論文として発表できるかもしれないといいことづくめである。だが、プログラム言語を習得してそのためのプログラムをゼロから書くとなると尻込みする方も多いだろう。

そんなときには R である。R に備えられている乱数はよくできているし、データ解析の各種の方法はすでに関数として存在することが多い。データ解析の手法の適切さを検討したいときにも計算の大部分は R の関数にやらせてしまうことができる。たとえば、自分の研究でよく出会う状況では一般化線形モデル (GLM) での検定が適切かという問題で悩んだとしてみよう。自分がよく出会いそうなデータセットを乱数で作って、GLM を適用し、その結果を見るということになるだろう。“GLM を適用し、その結果を見る” ところは、R では関数 `glm()` を使うのが普通である (使ったことのある方も少なくないだろう)。そこまでは R の関数 `glm()` にやらせればよい。R にすでにある関数に計算させた結果を再利用できるから大幅に省力化できる。

説明変数が `x1` と `x2` の 2 つ、目的変数が `y` で、誤差が二項分布 (この場合はロジスティック回帰になる) の場合の GLM を例にしてみる。まず、データを GLM で分析するには、

```
> res <- glm(y ~ x1 + x2, family=binomial)
```

として、結果の詳細を見るため summary() を使い

```
> summary(res)
```

と入力するとその結果が以下のように表示される (一部省略).

```
Call:
glm(formula = y ~ x1 + x2, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0967  -0.2034   0.1427   0.2596   0.6844

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.5021     1.5775  -0.318   0.7503
x1           -0.3547     0.1805  -1.965   0.0494 *
x2            0.2963     0.1158   2.559   0.0105 *
```

ここまでは、glm() を使ったデータ解析で経験されている方も少なくないだろう。

検定の適切さを見るときには、有意確率や最大対数尤度などが必要になることが多い。また検定ではなくてモデル選択なら AIC や最大対数尤度が必要になるだろう。分析結果が入っている res や summary(res) からこれらの値を取り出すことが、検定の適切さなどをみるために必要である。たとえば有意確率を取り出してみる。

str() という関数を使うと、summary(res) や res などの中身がどうなっているのか構造を見ることができ、必要なものの取り出し方もわかってくる。str(summary(res)) として summary(res) の内部構造を見ていくと、coefficients というところに有意確率が入っていることがわかる。

```
> round(summary(res)$coefficients, 4)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.5021     1.5775  -0.318   0.7503
x1           -0.3547     0.1805  -1.965   0.0494
x2            0.2963     0.1158   2.559   0.0105
```

と行列のようになっているので、

```
> summary(res)$coefficients["x1","Pr(>|z|)"]
[1] 0.04939235
```

として、説明変数 x1 の効果の有意確率が取り出せる。同様に、説明変数 x2 についての有意確率なら

```
> summary(res)$coefficients["x2","Pr(>|z|)"]
[1] 0.01051021
```

となる。

AIC や最大対数尤度ならもっと簡単に取り出せて、たとえば、

```
> AIC(res)
[1] 36.54779
> logLik(res)
'log Lik.' -15.27389 (df=3)
```

である。

“R の計算結果から必要なものを取り出し再利用する” ことは、統計的手法の適切さを検討するときだけでなく、複雑な検定などに広く使えるパラメトリック・ブートストラップなどいろいろなときに役に立つ。おぼえておいて損のない使い方である。

R の参考文献

最後に R を使うときに参考になりそうな本を紹介しておきたい。R 利用の急速な拡大とともに、国内外で R 関連の書籍もどんどん出版されている。ここではそのすべてをとりあげることはとうていできないので、日本語の書籍を中心に何点かを紹介する。

入門的な R 本としては舟尾・高浪 (2006) が大学院生たちには人気があるようだ (ただし付録 CD に入っている R は古いのでネット上から最新版をインストールする必要がある)。Dalgaard (2002) やその翻訳本 (Dalgaard 2007) は最初に R による data.frame のあつかいが紹介されているので、R 初心者にはわかりやすい教科書である。またイギリスの植物生態学者 Michael J. Crawley の書いた R 使用を前提とした統計学教科書 (Crawley 2005; その翻訳本 Crawley 2008) は生態学だけでなく多くの分野で好評のようだ。掲載されている例題は生態学研究者にとってはなじみぶかいものが多い。これらはかなり「R より」で R 初心者を意識したものである。

より「ふつうの統計学教科書」だけど学習者が R を使うことを前提としている本も多数ある。いろいろな手法が列挙されているものとしては中澤 (2007) などがあげられる。

より発展的な内容の R 書籍としては、“MASS” 本こと Venables and Ripley (2002) がもっとも有名だろう (MASS とは書名の略)。これは R 開発の中核メンバーであり統計学専門家である著者たちが「R でできること」を丁寧に解説している一冊である。た

とえばデータのばらつきが負の二項分布であるときの一般化線形モデルの推定関数 `glm.nb()` とその周辺についてはこの本がもっともくわしく説明している。R の発展的な活用事例集としては岡田 (編 2004) は便利かもしれない。生残解析・空間統計学・時系列解析そのほか多彩な内容である。

R のもっとも重要な機能のひとつ、「データを一望できる」ようなグラフィックス機能については上記の本のすべてがとりあげている。R にはより情報集積度の高い図を作るしくみがいろいろと用意されている。その代表は `library(lattice)` や `grid` package である。Murrell (2005) や Sarkar (2008) にくわしく説明されていて、ぱらぱらとページをながめているだけで「R でこんな図が描けるのか」と驚くような内容である。

R のプログラミング、つまりくり返し処理などを含んだデータ解析の自動処理を解説する Ligges (2006) などの書籍もふえてきた。異色なのは「R プログラミングマニュアル」(間瀬 2007) で「プログラミング」と書かれているけれど「R でこういうふうにプログラミングしなさい」といった本というよりも、R でプログラミングにかぎらず何か作業するときに与える命令、つまり R の関数たちひとつひとつについていちいち具体例をつけて挙動を説明する、世界でも類例をみない R の reference manual になっている。

R をはじめよう!

良質な入門書やネット上の解説情報がふえたおかげで R を始めるのはずいぶんと簡単になった。しかし多くの人にとって R は「いままで使ったことのないタイプのソフトウェア」であることが多い。それゆえに「どうやってもうまくいかない」困難にぶつかることもよくある。そこで最後に R をはじめるコツみたいなものを提案してみたい。

R 入門書は読むだけでなく、自分の手で R を動かしながら勉強するのがよい。しかしどこかでうまくいかない状況に必ず直面するだろう。そのときはまわりにいる R つかいをさがし、質問してみるのがよいだろう。わからないことを聞く、というあたりまえのことにすぎない。しかしながら R 入門者の何割かはこれがうまくできなくて停滞する。質問の上手な R 入門者は上達が速い。そして R つかいたちはあなたの質問にきちんと対応してくれるだろう。なぜかといえば、彼ら彼女らもまたそうやって R を習得していったからである。そしてこれは「何もかもオープンに、皆と共有する」といった R を発展させてきた精神にも通じるものである。

参考文献

Crawley MJ (2005) Statistics: an introduction using R. John Wiley & Sons, West Sussex

Crawley MJ (2008) 統計学 : R を用いた入門書 (野間口謙太郎・菊池泰樹訳). 共立出版, 東京

Dalgaard P (2002) Introductory Statistics with R. Springer, New York

Dalgaard P (2007) R による医療統計学 (岡田昌史監訳). 丸善, 東京

舟尾暢男 (2005) The R Tips: データ解析環境 R の基本技・グラフィックス活用集. 九天社, 東京

舟尾暢男・高浪洋平 (2006) データ解析環境「R」. 工学社, 東京

久保拓弥・粕谷英一 (2006) 「個体差」の統計モデリング. 日本生態学会誌 56: 181-190

Ligges U (2006) R の基礎とプログラミング技法 (石田基広訳). シュプリンガー・ジャパン, 東京

間瀬茂 (2007) R プログラミングマニュアル. 数理工学社, 東京

Murrell P (2005) R Graphics. Chapman & Hall/CRC press, London

中澤港 (2007) R による保健医療データ解析演習. ピアソンエデュケーション, 東京

岡田昌史 (編) (2004) The R Book: データ解析環境 R の活用事例集. 九天社, 東京

R Development Core Team (2008) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. `http://www.R-project.org/`

Sarkar D (2008) Lattice: Multivariate data visualization with R. Springer, New York

Venables WN, Ripley BD (2002) Modern Applied Statistics with S (4th ed.). Springer, New York