

# R データの読み込みと簡単な処理・作図

2008/05/23 玉木 一郎 (森林生態生理)

## 1 データ解析編

### 1.1 拡張子の表示

まず解析を始める前に拡張子が表示されるようにしておきましょう。

Windows XP： フォルダ > ツール > フォルダオプション > 表示 > 詳細設定の「登録されている拡張子は表示しない」のチェックを外す。

Windows Vista： フォルダ > 整理 > フォルダと検索のオプション > 表示 > 詳細設定の「登録されている拡張子は表示しない」のチェックを外す。

Max OS X： Finder > 環境設定... > 詳細の「すべてのファイル拡張子を表示」をチェック。

### 1.2 入力データの作成

毎木データ maiboku.xls をファイル > 別名で保存 > csv 形式 (maiboku.csv) を選択して保存。欠損値がある場合は NA を入れておいて下さい。<sup>\*1</sup>

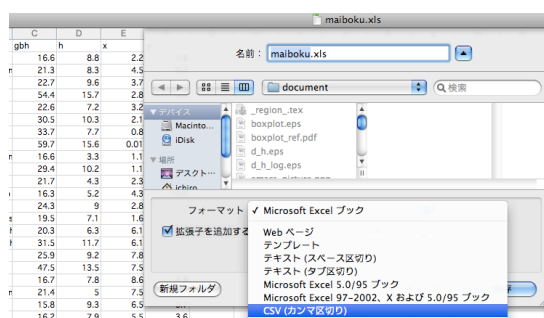


図 1 csv 形式で保存

### 1.3 データファイルの読み込み

保存した csv ファイルのあるフォルダ (例えば C:/data/R) を指定して、データを読み込み変数 d に代入します。

```
> setwd("C:/data/R") # 作業フォルダを指定
> d <- read.table("maiboku.csv", header = T, sep = ",")
> # csv ファイルを読み込んで変数 d に代入
>
```

<-は代入を意味します。#以降の文字列は無視されるので注意書きを書き込むことができ

<sup>\*1</sup> データ中で日本語などの全角文字は扱うことは可能ですが全て半角英数文字にしておいたほうが無難です。また、半角英数文字でもスペースや +, -, \*, / は入れないで下さい。代わりに\_が良いでしょう。

ます。また、コマンド内のスペースも無視されるので、コマンドが見やすくなるように適当に入れてあります。エラーが表示されていなければ読み込んでいるはずですが、変数名 `d` を入力して確認してみましょう。その後 `str(d)` と入力して下さい。

```
> d # ちゃんと読み込めたか確認
  id species gbh h x y
1  1 sakaki 16.6 8.8 2.2 0.6
2  2 takanotsume 21.3 8.3 4.5 0.3
3  3 ryoubu 22.7 9.6 3.7 1.2
4  4 konara 54.4 15.7 2.8 3.2
5  5 soyogo 22.6 7.2 3.2 4.4
以下略...
>
> str(d) # 要約された情報で確認する場合
'data.frame': 2329 obs. of 8 variables:
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ species: Factor w/ 30 levels "abemaki","akamatsu",...: 21 29 20 12 26
2 2 12 29 2 ...
 $ gbh     : num  16.6 21.3 22.7 54.4 22.6 30.5 33.7 59.7 16.6 29.4 ...
 $ h       : num   8.8  8.3  9.6 15.7  7.2 10.3  7.7 15.6  3.3 10.2 ...
 $ x       : num   2.2  4.5  3.7  2.8  3.2  2.1  0.8  0.01  1.1  1.1 ...
 $ y       : num   0.6  0.3  1.2  3.2  4.4  3.9  5.3  4.9  8.2  8.2 ...
 $ dbh     : num    5.28  6.78  7.23 17.32  7.19 ...
 $ ba      : num   21.9  36.1  41.0 235.5  40.6 ...
>
```

## 1.4 データフレームの編集

ここでは `gbh` (胸高周囲長) から `dbh` (胸高直径)、`ba` (胸高断面積) を計算し、データフレーム `d` の列に追加します。

```
> d$dbh <- d$gbh/pi # d に新たな列 dbh を追加し、gbh/pi の計算結果を代入
> d$ba <- (d$dbh/2)^2 * pi # 同様に ba を計算
> head(d, 5) # 上から 5 行を取り出して確認
  id species gbh h x y      dbh      ba
1  1 sakaki 16.6 8.8 2.2 0.6  5.283944  21.92837
2  2 takanotsume 21.3 8.3 4.5 0.3  6.780001  36.10350
3  3 ryoubu 22.7 9.6 3.7 1.2  7.225634  41.00548
4  4 konara 54.4 15.7 2.8 3.2 17.316058 235.49839
5  5 soyogo 22.6 7.2 3.2 4.4  7.193803  40.64499
>
```

## 1.5 データの集計

どんな樹種があつて、樹種数と各樹種の個体数が知りたいときは？

→ `levels()`, `nlevels()`, `table()`

```
> levels(d$species) # d$species で species 列を取り出す
[1] "abemaki"      "akamatsu"    "aohada"      "arakashi"
[5] "asebi"        "azukinashi"  "harienju"    "hisakaki"
[9] "honoki"       "inutsuge"    "isonoki"     "konara"
[13] "kuri"         "mansaku"     "marubaaodamo" "nejiki"
[17] "nezumisashi" "noriutsugi"  "okamenoki"   "ryoubu"
[21] "sakaki"       "sakura_sp"   "sawafutagi"  "shashanbo"
[25] "shidekobushi" "soyogo"      "sudajii"     "sugi"
[29] "takanotsume" "urajironoki"

> nlevels(d$species) # 水準数を返す
[1] 30

> table(d$species) # 集計

      abemaki      akamatsu      aohada      arakashi      asebi
           26           507            22            6           20
 azukinashi  harienju    hisakaki      honoki    inutsuge
           3             1             1             1           32
  isonoki     konara      kuri      mansaku  marubaaodamo
           11           260            25            35            1
  nejiki  nezumisashi  noriutsugi  okamenoki  ryoubu
           23           166             4             1           449
 sakaki  sakura_sp   sawafutagi  shashanbo  shidekobushi
           14             2             3            13            20
 soyogo  sudajii     sugi  takanotsume  urajironoki
           584             1            12            85            1

>
```

abemaki の dbh の平均・標準偏差が知りたいときは？→ `mean()`, `sd()`

```
> mean(d[d$species == "abemaki", "dbh"])
[1] 11.41998
> sd(d[d$species == "abemaki", "dbh"])
[1] 2.203317
>
```

`d[d$species == "abemaki", "dbh"]` で species が abemaki である列の、行 dbh を取り出します。これ以外にも様々なデータの取り出し方があるのでいろいろ試してみてください（例えば `d$dbh[d$species == "abemaki"]` でも良いです）。このとき、abemaki を他の樹種に置き換えれば、その樹種の dbh の平均値・標準偏差が計算できます。dbh を ba に置き換えれば同様の値が計算できます。しかし、一種ずつ計算するのは大変です。以下に `tapply()` で全樹種についていっぺんに計算し、知りたい樹種の結果だけを取り出

す方法を示します。

```
> tapply(d$dbh, d$species, mean)
  abemaki    akamatsu    aohada    arakashi    asebi
  11.419979  10.969386   8.611729   5.899343   6.059029
 azukinashi  harienju    hisakaki    honoki    inutsuge
   8.849015   5.570423   5.347606   9.040001   6.143381
  isonoki     konara     kuri     mansaku  marubaaodamo
   5.816390  10.456969   8.334626   6.249787   7.257465
  nejiki  nezumisashi  noriutsugi  okamenoki  ryoubu
   5.989762   6.472045   6.525353   6.843663   7.349130
  sakaki  sakura_sp  sawafutagi  shashanbo  shidekobushi
   5.545413   5.713662   8.392771   6.544941   6.202268
  soyogo  sudajii    sugi  takanotsume  urajironoki
   7.879096   5.506761  11.382231   7.758523   9.390142
>
```

ここから特定の樹種、例えば `honoki` の結果を取り出すこともできます。

```
> res_mean <- tapply(d$dbh, d$species, mean)
> # いったん結果を変数 res_mean に代入
> res_mean["honoki"]
honoki
 9.04
```

## 1.6 編集したデータフレームの保存

```
> write.table(d, file = "maiboku_edited.csv", quote = F,
+ row.names = F, sep = ",")
>
```

コマンドの途中で改行すると + マークが付いて入力続けることができます。コマンドが長くなる場合はこのように適当に改行したほうが見やすいでしょう。保存したファイルは最初に `setwd()` で指定したフォルダに保存されます。ちゃんと保存されたか確認してみてください。うまく保存できていれば Excel などで開くことができるはずです。

## 2 作図編

### 2.1 ヒストグラムの作成

データには先ほど `dbh` などを計算したデータフレーム `d` を用います。 `konara` の `dbh` の分布を見るためにヒストグラムを描いてみます。

```
> DBH <- d$dbh[d$species == "konara"]
> hist(DBH)
>
```

表示された図はメニューのファイル > 別名で保存 > ...、で任意のファイル形式で保存することができます (Windows)。Windows の人で後で Word や PowerPoint に貼付ける場合には、メタファイル形式が便利でしょう。

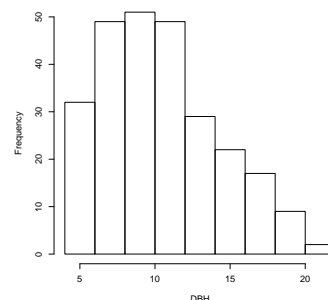


図2 コナラの頻度分布図

## 2.2 散布図の作成

直径が大きい樹木ほど樹高が高いことが考えられます。2変量の間関係を見たい場合には散布図が有効です。横軸に dbh、縦軸に h (樹高) をとった散布図を描いてみましょう。

```
> plot(d$dbh, d$h, xlab = "dbh", ylab = "height")
>
```

plot() は括弧の中に x, y の順にデータを入れるだけで、散布図を描くことができます。xlab, ylab でそれぞれ横軸、縦軸のラベルを指定できます。対数軸にしたいときは下のようになります。

```
> plot(d$dbh, d$h, xlab = "dbh", ylab = "height", log = "xy")
>
```

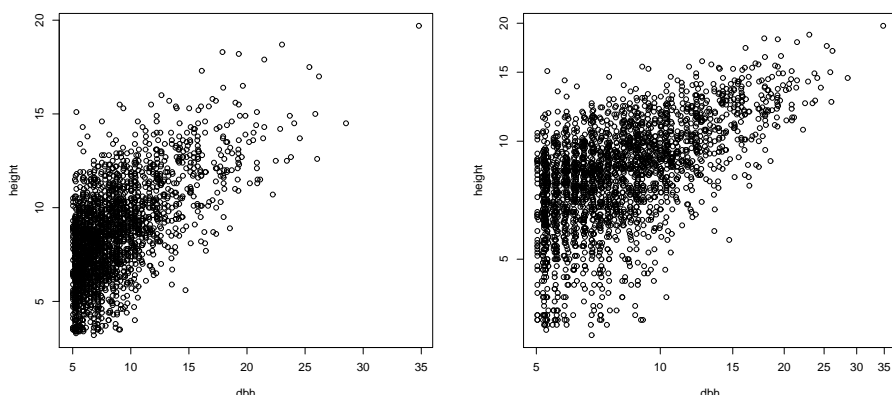


図3 D-H 関係図 (左: 通常目盛、右: 対数目盛)

## 2.3 箱ひげ図の作成

箱ひげ図は複数のデータ分布の違いを比較するのに便利な図です。聞いたことがないという人がいるかもしれませんが、この機会に覚えると良いでしょう。

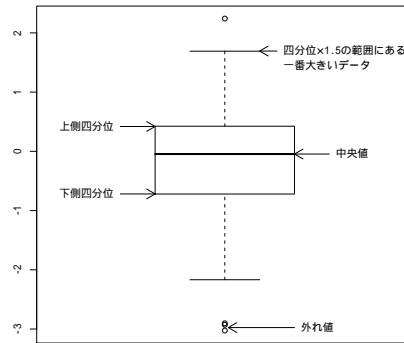


図4 箱ひげ図の説明

樹種による dbh の違いを箱ひげ図で比較してみましょう。

```
> boxplot(d$dbh ~ d$species)
>
```

boxplot() の括弧の中は (データ グループ変数) です (グループ分けがない場合はデータだけで OK です)。通常はこれだけで十分なのですが、今回のデータでは、このままだと軸ラベルが見にくいので、以下のコマンドで修正します。

```
> par(oma = c(3, 0, 0, 0)) # 余白の調整
> boxplot(d$dbh ~ d$species, las = 3) # las で軸に対するラベルの向きを変更
>
```

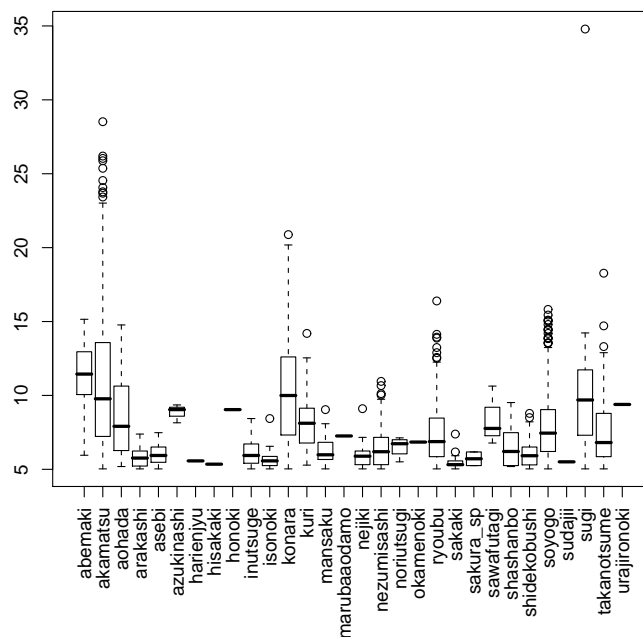


図5 各樹種の dbh 分布

この図をぱっと見て分かることには、上にひずんだ分布の樹種が多いとか、最大 dbh の個体を持つ樹種は sugi だとか、akamatsu や konara は小さいサイズから大きいサイズまで幅広い dbh 分布を持つのだなということがあげられます。サンプル数が図からは分

からないのが短所ですが、大まかなデータの分布を把握するのに役立ちます。

## 3 おまけ

最後に R を使って一連の作業を行う際に知っておくと便利なことを記述しておきます。

### 3.1 ヘルプの利用

知りたい関数の前に?をつけるか `help()` でヘルプを見ることができます。また、断片的な語句で検索したいときは `help.search()` で検索することができます。

```
> ?write.table # もしくは help(write.table)
> help.search("write")
>
```

### 3.2 作業の流れ

最初のうちは一行一行、R のコンソールにコマンドを打ち込むのも良いですが、長くなると大変だし、あとで何を行ったか確認するためにもファイルにその過程を記しておくことが望ましいでしょう。適当なエディタを利用して、一連の手順を記述したテキストファイルを作成します（拡張子は `.txt` のままでも良いですが、ここでは `.r` としておきます）。このファイルからコピー&ペーストで実行すれのも良いですが、`source()` でファイルを読み込むと便利です。例えば `test.r` に以下の文字列を記述して保存し、

```
cat("Hello, world and R!\n") # 画面に Hello, world and R!を表示
```

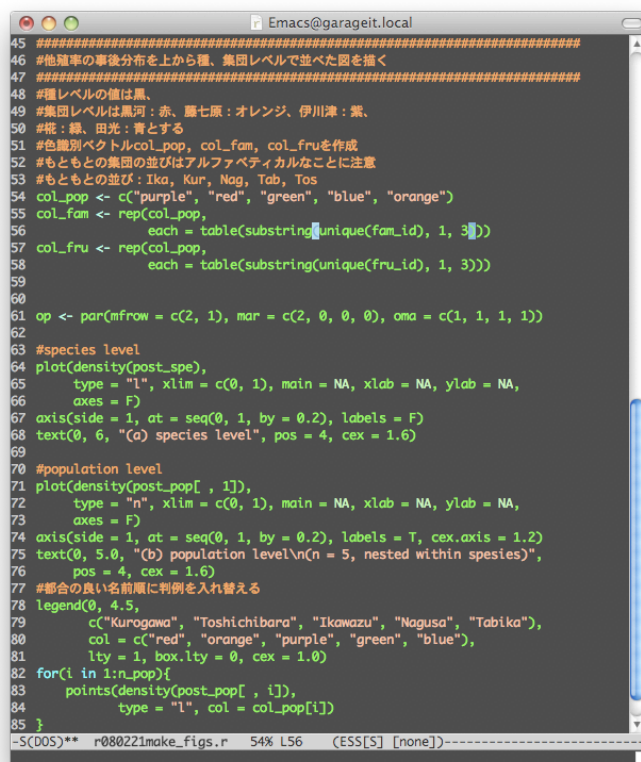
保存したフォルダを指定して、R コンソールから以下のように実行します。

```
> source("test.r")
Hello, world and R!
```

## 3.3 役立つソフトウェア

### 3.3.1 テキストエディタ

R の一連のコマンドを記述したスクリプトファイルを作成する際には、Windows の場合は R 付属のエディタや OS 付属の `notepad` はシンプルすぎて使い勝手が良くありません。行番号を表示したり、対応する括弧や各コマンドを色分けして表示できるようなテキストエディタを使うと作業の効率を上げることができます。



```

45 #####
46 #他種率の事後分布を上から種、集団レベルで並べた図を描く
47 #####
48 #種レベルの値は黒、
49 #集団レベルは黒河：赤、轟七原：オレンジ、伊川津：紫、
50 #桜：緑、田光：青とする
51 #色識別ベクトルcol_pop, col_fam, col_fruを作成
52 #もともとの集団の並びはアルファベティカルなことに注意
53 #もともとの並び：Ika, Kur, Nag, Tab, Tos
54 col_pop <- c("purple", "red", "green", "blue", "orange")
55 col_fam <- rep(col_pop,
56               each = table(substring(unique(fam_id), 1, 3)))
57 col_fru <- rep(col_pop,
58               each = table(substring(unique(fru_id), 1, 3)))
59
60
61 op <- par(mfrow = c(2, 1), mar = c(2, 0, 0, 0), oma = c(1, 1, 1, 1))
62
63 #species level
64 plot(density(post_spe),
65       type = "l", xlim = c(0, 1), main = NA, xlab = NA, ylab = NA,
66       axes = F)
67 axis(side = 1, at = seq(0, 1, by = 0.2), labels = F)
68 text(0, 6, "(a) species level", pos = 4, cex = 1.6)
69
70 #population level
71 plot(density(post_pop[, 1]),
72       type = "n", xlim = c(0, 1), main = NA, xlab = NA, ylab = NA,
73       axes = F)
74 axis(side = 1, at = seq(0, 1, by = 0.2), labels = T, cex.axis = 1.2)
75 text(0, 5.0, "(b) population level\n(n = 5, nested within spesies)",
76       pos = 4, cex = 1.6)
77 #都合の良い名前順に判例を入れ替える
78 legend(0, 4.5,
79        c("Kurogawa", "Toshichibara", "Ikawazu", "Nagusa", "Tabika"),
80        col = c("red", "orange", "purple", "green", "blue"),
81        lty = 1, box.lty = 0, cex = 1.0)
82 for(i in 1:n_pop){
83   points(density(post_pop[, i]),
84          type = "l", col = col_pop[i])
85 }

```

図 6 Mac OS X 上の Carbon Emacs

例えばこんな具合になります。Windows の人は**秀丸エディタ**や Tinn-R がお勧めです。導入が面倒ではありますが Meadow (Emacs) も便利です。Mac OS X の人は mi や Emacs、Vim を使うと良いでしょう。その他の OS の人は言われなくとも、既にお気に入りのエディタがあることでしょう。

### 3.3.2 Ghostscript

R の画像を出力する際に、ps や eps 形式で出力したいことがあります（論文の投稿時など）。これらのファイルの出力結果と Windows 標準のグラフィックデバイスの見た目は若干異なります（軸や文字のサイズ等）。Windows ではこれらのファイルを直接扱うことはできません。Illustrator や Canvas を持っている人は、これらの画像を編集することも可能ですが、普通の人は持っていないと思います。編集はできませんが、Ghostscript (GSview) をインストールしておくことこれらのファイルを見ることができます。また、いったんファイルを開いておくと更新するたびに表示も更新してくれるので便利です。インストールは  $\text{\TeX}$  と一緒に行うのが簡単です。以下の URL を参考にして下さい。

<http://oku.edu.mie-u.ac.jp/okumura/texwiki/>