

R の基本操作 (メモ)

第 10 回 農環技研 統計 GIS コース

三輪哲久 (農業環境技術研究所 生態系計測研究領域長)

October, 2011

目次

1	はじめに – 注意事項と参考文献	2
2	R を使ってみよう	2
2.1	ホームディレクトリと作業ディレクトリ	2
2.2	R の開始と終了	2
3	R 環境設定のまとめ (メモ)	3
3.1	ホームディレクトリと作業ディレクトリ	3
3.2	環境変数	3
3.3	R の操作	4
4	An Introduction to R	5
4.1	基本操作	5
4.2	基本演算	5
4.3	実行例	7

1 はじめに – 注意事項と参考文献

- 下記の説明で、「ディレクトリ」と「フォルダ」は同義。
- 下記の説明で、「\」=「¥」（ディレクトリの区切り記号）
- 役に立つ文書（インストールした R のフォルダに有る）
 - c:¥Program Files¥R¥R-2.13.1¥doc¥manual¥R-intro.pdf
R 入門。
 - c:¥Program Files¥R¥R-2.13.1¥doc¥rw-FAQ（テキストファイル）
Windows 版 R のインストールに関する参考資料。
- 役に立つ Web サイト。
 - <http://www.r-project.org/>
R プロジェクトの本家のサイト。
‘CRAN’ をクリックするとミラーサイトを選べる。
 - <http://www.okada.jp.org/RWiki/>
R の情報交換のサイト。
- 下記の説明で、ファイルシステムなど、OS (UNIX, Mac, Windows) に依存する部分については、おもに Windows に関して記述されている（筆者の利用環境）

2 R を使ってみよう

2.1 ホームディレクトリと作業ディレクトリ

- ホームディレクトリ（ホームフォルダ，home directory）
 - 各ユーザーに固有のフォルダで、各種の文書ファイルを保存する。
 - 特に指定しなければ、「マイドキュメント」フォルダが使われる。
 - c:¥Users¥username¥Documents¥（Win 7, Vista）
 - c:¥Documents and Settings¥username¥My Documents¥（XP）
- 作業ディレクトリ
（作業フォルダ，working directory）
 - R で行なう個別の仕事ごとに作成する。
 - R のショートカット・アイコンも各仕事ごとに作成（コピー）し、「プロパティ」（アイコンを右クリック）の「作業フォルダ」で指定する。

2.2 R の開始と終了

- R のアイコンをダブルクリックすると R が開始する。
- R を終了するには、q() と入力する。
このとき、“作業スペースを保存しますか?” と聞かれる。“はい” をクリックすると、現在までの作業内容が作業ディレクトリの中の ‘.RData’ と ‘.Rhistory’ の 2 つのファイルに保存される。次回、その作業ディレクトリから R を実行したときに、それまでの作業内容が継承される。



図 1. 作業フォルダの指定

3 R環境設定のまとめ(メモ)

「環境変数」という言葉を知らない人は、この第3節を読み飛ばしても構わない。

3.1 ホームディレクトリと作業ディレクトリ

内容	rw-FAQ	R-intro
ホームディレクトリ (home directory)	§2.14	
<ul style="list-style-type: none">各ユーザーに1つ。Rに限らず、各種ソフトウェアの設定用ファイルが置かれる。また、このディレクトリ(フォルダ)以下に文書ファイル(ワープロ作成の文書ファイルなど)が保存されることもある。環境変数 HOME で指定する。 その指定がなければ、「マイドキュメント」: c:\Documents and Settings\username\My Documents\ (XP) c:\Users\username\Documents\ (Win 7, Vista)		
R_USER ディレクトリ	§2.14	
<ul style="list-style-type: none">Rに固有の設定(Rconsole, Rdevgaなど)を保存する。環境変数 R_USER で指定する。 その指定が無ければ、ホームディレクトリが使われる。		
作業ディレクトリ(作業フォルダ, working directory)	§2.14	§1.5
<ul style="list-style-type: none">まとまった仕事ごとに別々のディレクトリを指定すると便利。 §2.2, 2.5「ショートカット」アイコンの「プロパティ」(右クリック)の「作業フォルダ」で指定する。getwd() コマンドで現在の作業ディレクトリが分かる。setwd("ディレクトリへのパス") コマンドで変更できる。 あるいは、「File」メニュー → 「Change dir...」で変更できる。作業スペース.RData(後述)は、このディレクトリに保存される。		

3.2 環境変数

内容	rw-FAQ	R-intro
環境変数 (environmental variable)	§2.15	
<ul style="list-style-type: none">各種プログラムで必要となる情報を指定する。 Rでは、例として次のようなものがある。 LANGUAGE=en (英語モードで使う) R_USER=. (作業ディレクトリ「.’」をR_USERとする) R_LIBS="c:\Program Files\R\myRlib" (パッケージの指定) 環境変数をR_USER=.としておけば、作業ディレクトリにRconsoleファイルが保存される。作業ディレクトリごとに異なる言語・フォントを使うことができる。 <ul style="list-style-type: none">「ショートカットアイコン」の「プロパティ」(右クリック)の「リンク先」で指定する。あるいは、ファイル「.Renviron」を作成し、そのファイルの中に記述する。ファイル「.Renviron」はホームディレクトリに置く。		

3.3 R の操作

内容	rw-FAQ	R-intro
R の開始	§2.5	
R のショートカットアイコンをダブルクリックする。		
R の終了と作業スペース (workspace)	§6	
<ul style="list-style-type: none"> ● R を終了するには、<code>q()</code> と入力する。 ● R の終了時に、“作業スペースを保存しますか?” と聞かれる。 “はい” をクリックすると、現在までの作業内容が作業ディレクトリの中の <code>‘.RData’</code> と <code>‘.Rhistory’</code> の 2 つのファイルに保存される。次回、その作業ディレクトリから R を実行したときに、それまでの作業内容が継承される。 ● したがって、異なる仕事には、異なる作業ディレクトリを用意すると混乱がなくなる。 ● 必要がなくなれば、ファイル <code>‘.RData’</code> を消去すればよい。 		
2 つのファイル <code>Rconsole</code> と <code>Rdevga</code>	§5.2	
<ul style="list-style-type: none"> ● <code>Rconsole</code> → R のコンソール画面の設定 ● <code>Rdevga</code> → R のグラフィックスのフォントの設定 ● <code>Rconsole</code> では、使用する言語、使用するフォントや色、画面の大きさなどを指定する。R を実行した後に、“Edit” → “GUI preferences...” のメニューから変更できる (<code>Rconsole</code> ファイルを直接編集してもよい)。 ● これら 2 つのファイルは、ホームディレクトリ (<code>R_USER</code> ディレクトリ) に置く。 		
R のメニューやメッセージに使われる言語	§3.3	
<ul style="list-style-type: none"> ● メニューから “Edit” → “GUI preferences...” と進み、“Language for menus” の欄で、“en” または “ja” 指定してファイル <code>Rconsole</code> に保存する。 ● 環境変数 <code>LANGUAGE</code> で指定してもよい。 <ul style="list-style-type: none"> LANGUAGE=en (英語モードで使う) LANGUAGE=ja (日本語モードで使う) アイコンごとに別の言語で使用できる。 ● 英語の練習を兼ねて、英語モードで使うことが望ましい。 	§3.5	
R コンソールでの入力	§5.1	
<ul style="list-style-type: none"> ● Help メニュー → Console キーボードの使い方の説明。 ● 入力中に Tab キーを押すと、コマンドが補完される。 		
注意: R の中では、ディレクトリの区切り記号は、 <code>‘/’</code> または、 <code>‘\’</code> のどちらでもよい。たとえば	§5.1	
<pre>> .libPaths("c:/Program Files/R/myRlib") > .libPaths("c:\\Program Files\\R\\myRlib")</pre>		
のどちらでもよい (<code>‘.libPaths()’</code> はライブラリのあるディレクトリを指定するコマンド)。		

4 An Introduction to R

4.1 基本操作

内容	rw-FAQ	R-intro
Rの開始と終了		§1.5
<ul style="list-style-type: none">アイコンをダブルクリックして開始 (Windows)。q() コマンドで終了。 “作業スペースを保存しますか?” の問いに “はい” と答えると、途中経過が作業ディレクトリの “.RData” ファイルに保存される (次回に継続される)。		
ヘルプとコメント		§1.7
<ul style="list-style-type: none">help.start() (または、Help メニューから “Html help”)help(glm), または ?glm (特定の関数の情報) help("if") (場合によっては、""で囲む)help.search("glm"), または ??glm (広く情報を検索)help 画面の最後に与えられている examples を参考にして、自分の問題に当てはめればよい。‘#’ 以降は行の最後までコメント		§1.8
コマンドの履歴と編集	§5.1	§1.9
<ul style="list-style-type: none">Help メニュー → Console に、キーボード使い方の説明がある。上下矢印キーで過去のコマンドを呼び出せる。File メニューの “New Script” ウィンドウで作業すると編集が便利。コマンドを中断するには <ESC> キーを押す。		
オブジェクト		§1.11
<ul style="list-style-type: none">R で扱う対象はオブジェクト (object) とよばれ、R の実行中は保持される。object() (または ls()) コマンドで現在のオブジェクトを表示。rm(...) コマンドでオブジェクトを消去。		

4.2 基本演算

内容	rw-FAQ	R-intro
基本はベクトル		§2.1
<ul style="list-style-type: none">c() はベクトルを与える関数。スカラーは長さ 1 のベクトル。<pre>> x <- c(10.4, 5.6, 3.1, 6.4, 21.7) > y <- 15.0 > length(x); length(y) [1] 5 [1] 1</pre>代入 (‘<-’ 演算子) では値が表示されない。式だけを書くと値が表示される。直前の値は .Last.value にある。<pre>> 2*x # 要素ごとに計算。 [1] 20.8 11.2 6.2 12.8 43.4 > print(z <- .Last.value) # 直前の結果。 [1] 20.8 11.2 6.2 12.8 43.4</pre>		

内容	rw-FAQ	R-intro
<ul style="list-style-type: none"> ベクトルの各要素は繰返し計算に使われる。上記の‘2*x’参照。 ベクトル用の関数もある。 length(x), sum(x), mean(x), var(x), max(x), min(x), sort(x) など。 		§2.2
<p>規則的な配列。</p> <ul style="list-style-type: none"> c(1:10) (単に‘1:10’でもよい)。 seq(from=, to=, by=), または, seq(from=, to=, length=)。 rep(x, times=), および, rep(x, each=)。 		§2.3
<p>数値のほかにベクトルの要素となるもの。</p> <ul style="list-style-type: none"> 論理値。TRUE, FALSE。 <pre>> x >= 10.0 [1] TRUE FALSE FALSE FALSE TRUE</pre> x の要素ごとに計算されていることに注意。 論理演算子: <, <=, >, >=, ==, !=, &, , ! 文字列。 <pre>> print(z <- c("abc", "def", "ghi")) # 文字列ベクトル。 [1] "abc" "def" "ghi" > paste(z, c(1:3), sep="-") # 各要素ごとに結合。 [1] "abc-1" "def-2" "ghi-3" > paste(z, collapse="") # ベクトル内の要素を結合。 [1] "abcdefghi" 詳しくは, help(paste) を参照。</pre> 		§2.4 §2.6
<p>欠測値</p> <ul style="list-style-type: none"> 欠測値 NA (Not Available)。非数値 NaN (Not a Number)。 関数: is.na(x), is.nan(x) 		§2.5
<p>ベクトルの要素の指定。</p> <ul style="list-style-type: none"> <pre>> x <- c(10.4, 5.6, 3.1, 6.4, 21.7) > x[2] # 2 番目の要素。 [1] 5.6 > x[1:3] # 複数の要素。 [1] 10.4 5.6 3.1 > x[x >= 10.0] # 特定の条件を満たす要素。 [1] 10.4 21.7</pre> 		§2.7
<p>ベクトル以外のオブジェクト (objects)。いろいろある。</p> <ul style="list-style-type: none"> 行列 (matrix), 配列 (array), リスト (list), 関数 (function) など。 データ・フレーム (date frame) は統計解析で重要。 		§2.8

4.3 実行例

R version 2.13.1 (2011-07-08)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

Rは、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()'あるいは'licence()'と入力してください。

Rは多くの貢献者による共同プロジェクトです。
詳しくは'contributors()'と入力してください。
また、RやRのパッケージを出版物で引用する際の形式については
'citation()'と入力してください。

'demo()'と入力すればデモをみることができます。
'help()'とすればオンラインヘルプが出ます。
'help.start()'でHTMLブラウザによるヘルプがみられます。
'q()'と入力すればRを終了します。

```
> demo() # どんなデモがあるか表示。( '#' から行の最後までは、コメント。)  
> demo(graphics) # 'graphics' のデモ。  
      demo(graphics)  
      ---- ~~~~~  
Type <Return> to start :  
  
> help.start() # Html help のスタート。  
starting httpd help server ... 完了  
もし何も起きなければ、自分で 'http://127.0.0.1:23598/doc/html/index.html' を開いて  
ください  
> help(glm) # 'glm' のヘルプ  
> ?glm # この方式でもよい。  
  
> getwd() # 現在の作業ディレクトリ。  
[1] "C:/Users/user00/Documents/R/StatGIS10"  
> Sys.getenv("R_USER") # 環境変数の表示。  
[1] "."  
> q() # R の終了。  
  
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7) # 代入だけでは、結果は表示されない。  
> x # オブジェクトを入力すると、値が表示される。  
[1] 10.4 5.6 3.1 6.4 21.7  
> length(x)  
[1] 5  
> print(y <- 15.0) # 代入時に print() 関数を使うと結果を表示。  
[1] 15  
> length(y) # スカラーは長さ 1 のベクトル。  
[1] 1  
> 2*x # 要素ごとに計算。  
[1] 20.8 11.2 6.2 12.8 43.4  
> print(z <- .Last.value) # 直前の結果。  
[1] 20.8 11.2 6.2 12.8 43.4  
  
> c(1:10) # 規則的な数列。  
[1] 1 2 3 4 5 6 7 8 9 10  
> 1:10 # この形でもよい。  
[1] 1 2 3 4 5 6 7 8 9 10  
  
> print(z <- c("abc", "def", "ghi")) # 文字列ベクトル。  
[1] "abc" "def" "ghi"
```

```

> paste(z, c(1:3), sep="-") # 各要素ごとに結合。
[1] "abc-1" "def-2" "ghi-3"
> paste(z, collapse="") # ベクトル内の要素を結合。詳しくは, help(paste)。
[1] "abcdefghi"

> 0/0 # NaN (Not A Number)
[1] NaN

> x
[1] 10.4 5.6 3.1 6.4 21.7
> x >= 10.0 # 要素ごとに論理計算。
[1] TRUE FALSE FALSE FALSE TRUE
> x[2] # 2番目の要素。
[1] 5.6
> x[1:3] # 複数の要素。
[1] 10.4 5.6 3.1
> x[x >= 10.0] # 特定の条件を満たす要素。
[1] 10.4 21.7
> x[x >= 10.0] <- 10.0 # 打ち切り操作。
> x
[1] 10.0 5.6 3.1 6.4 10.0

> ls() # 現在使われているオブジェクト。
[1] "x" "y" "z"
> rm(list=ls()) # 全てのオブジェクトを消去。

> # データ (data frame) の読み込み。
> # read.table("ファイル名", header=TRUE)
> duncan <- read.table("duncan.tab", header=TRUE)
> duncan
  variety block yield
1      A1     1  39.2
2      A1     2  63.7

... 途中省略 ...

41      A7     5  58.1
42      A7     6  69.1
> # クリップボードからの読み込み。
> duncan1 <- read.table("clipboard", header=TRUE)
> duncan1
  variety block yield
1      A1     1  39.2
2      A1     2  63.7

... 途中省略 ...

41      A7     5  58.1
42      A7     6  69.1
> duncan$yield # "data frame 名$変数名" で参照する。
[1] 39.2 63.7 56.9 41.9 49.3 46.7 63.3 63.4 81.9 66.9 65.0 86.8 65.6 68.8
[15] 58.6 56.0 86.1 70.5 47.3 58.2 61.9 64.2 74.5 63.0 80.3 72.5 78.6 54.8
[29] 73.4 68.2 55.2 60.9 62.8 61.9 56.4 51.4 46.0 78.7 56.5 57.4 58.1 69.1
> yield
エラー: オブジェクト 'yield' がありません
> attach(duncan) # 変数名だけで参照できるようにする。
> yield
[1] 39.2 63.7 56.9 41.9 49.3 46.7 63.3 63.4 81.9 66.9 65.0 86.8 65.6 68.8
[15] 58.6 56.0 86.1 70.5 47.3 58.2 61.9 64.2 74.5 63.0 80.3 72.5 78.6 54.8
[29] 73.4 68.2 55.2 60.9 62.8 61.9 56.4 51.4 46.0 78.7 56.5 57.4 58.1 69.1

```



```

> block
[1] 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 1 2 3 4 5 6 1
[38] 2 3 4 5 6
> block <- as.factor(block) # block 変数を分類変数として扱う。
> fm <- aov(yield ~ block + variety) # 乱塊法分散分析。
> summary(fm) # 結果の要約。
      Df Sum Sq Mean Sq F value    Pr(>F)
block    5   709.6   141.92    1.7833 0.146514
variety   6  2201.7   366.96    4.6111 0.001982 **
Residuals 30  2387.4    79.58
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> TukeyMC <- TukeyHSD(fm, "variety", conf.level=0.95) # Tukey の多重比較。
> TukeyMC

```

```

      Tukey multiple comparisons of means
      95% family-wise confidence level
Fit: aov(formula = yield ~ block + variety)

```

```

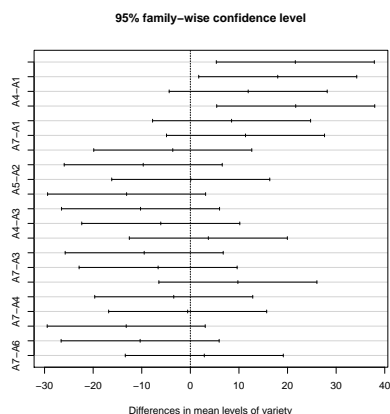
$variety
      diff      lwr      upr      p adj
A2-A1 21.60000000    5.341859 37.858141 0.0037694
A3-A1 17.98333333    1.725193 34.241474 0.0226659
A4-A1 11.90000000   -4.358141 28.158141 0.2720878
A5-A1 21.68333333    5.425193 37.941474 0.0036111
A6-A1  8.48333333   -7.774807 24.741474 0.6542984
A7-A1 11.35000000   -4.908141 27.608141 0.3233518
A3-A2 -3.61666667  -19.874807 12.641474 0.9914471
A4-A2 -9.70000000  -25.958141  6.558141 0.5060767
A5-A2  0.08333333  -16.174807 16.341474 1.0000000
A6-A2 -13.11666667  -29.374807  3.141474 0.1789977
A7-A2 -10.25000000  -26.508141  6.008141 0.4412442
A4-A3 -6.08333333  -22.341474 10.174807 0.8957640
A5-A3  3.70000000  -12.558141 19.958141 0.9903563
A6-A3 -9.50000000  -25.758141  6.758141 0.5302580
A7-A3 -6.63333333  -22.891474  9.624807 0.8520335
A5-A4  9.78333333   -6.474807 26.041474 0.4960790
A6-A4 -3.41666667  -19.674807 12.841474 0.9936813
A7-A4 -0.55000000  -16.808141 15.708141 0.9999998
A6-A5 -13.20000000  -29.458141  3.058141 0.1736352
A7-A5 -10.33333333  -26.591474  5.924807 0.4317003
A7-A6  2.86666667  -13.391474 19.124807 0.9975739

```

```

> plot(TukeyMC)
> detach(duncan) # "duncan"をサーチパスから除く。
> rm(list=ls()) # 全てのオブジェクトを消去。

```



```

> # data frame の作成 ( 回帰分析のシミュレーション )
> x <- c(1:10)
> y <- x + rnorm(10) # 正規乱数を加える。
> LinearData <- data.frame(var1=x, var2=y) # data frame の作成。
> plot(LinearData$var1, LinearData$var2)
> attach(LinearData) # 変数名のみで参照する ( "LinearData"をサーチパスに加える )
> fm1 <- lm(var2 ~ var1) # 1次回帰分析。
> summary(fm1)

```

Call:

```
lm(formula = var2 ~ var1)
```

Residuals:

```

      Min       1Q   Median       3Q      Max
-1.7111 -0.6835  0.2999  0.7476  1.3916

```

Coefficients:

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.4438     0.7454   0.595 0.568023
var1         0.8428     0.1201   7.015 0.000111 ***

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 1.091 on 8 degrees of freedom

Multiple R-squared: 0.8602, Adjusted R-squared: 0.8427

F-statistic: 49.21 on 1 and 8 DF, p-value: 0.0001109

```
> abline(fm1, col = "red") # 回帰直線を書き加える。
```

```
> q()
```

