R 入門 (1)

公衆衛生学教室セミナー　2004 年 5 月 13 日　中澤　港

（注）この文書において，\は，半角の¥記号を意味する。

1　R のインストール方法

Windows　会津大学の CRAN ミラーから R-1.8.1 のインストール用ファイル[*1]をダウンロードし，ダブルクリックして実行するだけでインストールは完了する。なお，既に 1.9.0/1.9.1/2.0 開発版が出ているが，barplot() 関数にバグがあるのでお薦めしない。

Macintosh　MacOS9 以前のユーザは R-1.7.1 までしか使えない。MacOS X では最新版が使える。同じく会津大学の CRAN ミラーから OS9 以前のユーザは rm171.sit[*2]をダウンロードしてダブルクリックしてインストールし，OS X ユーザは R.dmg[*3]をダウンロードしてダブルクリックしてできるフォルダ内の R.pkg をダブルクリックしてインストールすればよい。詳細は青木先生のサイトの解説[*4]を参照。

Linux/FreeBSD　詳しくは説明しないが，同じく会津大学の CRAN ミラーからバイナリパッケージまたはソース tar ball をダウンロードしてインストールすることができる。

2　使い方の基本

以下の解説は Windows 版による。

Rgui の起動は，デスクトップの R のアイコンをダブルクリックするだけでいい（なお，このアイコンのプロパティで作業ディレクトリを設定すれば，そこが既定の作業ディレクトリになる)。ウィンドウが開き，>という記号が表示されて入力待ちになる。この記号をプロンプトと呼ぶ。R への対話的なコマンド入力は，基本的にプロンプトに対して行う。そうやって入力した手続きは，File メニューの Save History ... で保存でき，後で File の Source で呼び出せば再現できる。プロンプトに対して source(”プログラムファイル名”) としても同じことになる（但し，Windows ではファイルパス中，ディレクトリ（フォルダ）の区切りは/または\\で表すことに注意)。また，上向き矢印キーで既に入力したコマンドを呼び戻すことができる。

2.1　最も基本の操作

終了	`q()`
付値（計算結果を任意の変数に保存する）	`<-`
関数定義（例：平均と標準偏差をリストとして返す関数）	`meansd <- function(X) { list(mean(X),sd(X)) }`
ライブラリインストール（例：CRAN から vcd をダウンロード）	`install.packages("vcd")`

関数定義は何行にも渡って行うことができ，最終行の値が戻り値となる。関数内の変数は局所化されているので，関数内で変数に付値しても，関数外には影響しない。関数内で変数の値を本当に変えてしまいたいときは，通常の付値でなくて，<<-を用いる。つまり例えば次の画面のようになる。この画面において，>は R のコンソールに入力を促す記号であり，+は継続行（関数やコマンドの途中の行）において入力を促す記号である。そのどちらでもない行が R の出力である。

[*1] ftp://ftp.u-aizu.ac.jp/pub/lang/R/CRAN/bin/windows/base/rw1081.exe
[*2] ftp://ftp.u-aizu.ac.jp/pub/lang/R/CRAN/bin/macos/rm171.sit
[*3] ftp://ftp.u-aizu.ac.jp/pub/lang/R/CRAN/bin/macosx/R.dmg
[*4] http://aoki2.si.gunma-u.ac.jp/R/begin.html

```
> x <- 2
> y <- function(a) {
+       x <- x+a
+       x }
> y(5)
[1] 7
> x
[1] 2
> z <- function(a) {
+       x <<- x+a
+       x }
> z(5)
[1] 7
> x
[1] 7
```

2.2 データファイル読み込み

1行目が変数名のタブ区切りテキストファイルのデータフレームへの読み込み	`dat <- read.delim("ファイル名")`
1行目が変数名のコンマ区切りテキストファイルのデータフレームへの読み込み	`dat <- read.csv("ファイル名")`

2.3 変数型宣言

カテゴリ変数宣言（データが文字列の場合は自動的に factor になるので不要）	`dat$C <- as.factor(dat$C)`
順序変数宣言	`dat$I <- as.ordered(dat$I)`
量的変数宣言	`dat$X <- as.single(dat$X)`

2.4 確認

データ構造表示	`str(dat)`
データフレーム内変数名一覧	`names(dat)`

3 データ入力の方法

3.1 データが少ないとき

データ数がごく少ない場合は，R のプログラム内で直接付値すればいい。例えば，身長 155 cm，160 cm，170 cm の 3 人の平均 (mean) と標準偏差 (sd) を出すためには，

```
dat <- c(155,160,170)
```

とすれば，R のプログラム内で扱える変数 `dat` ができる。そこで，`mean(dat)` とすれば平均値が得られるし，`sd(dat)` とすれば不偏標準偏差が得られるが，これらをまとめてやるには，

```
cat("mean=",mean(dat),"sd=",sd(dat),"\n")
```

とすればいい。cat() はコンマで区切られた要素を結合して表示する関数で，""で括られたものは文字列，そうでないものは数値となる。"\n"は改行記号を表す。また，クロス集計表を入力したい場合は行列として入力することは簡単である。例えば，次の表のようなデータがある場合を考えよう。

曝露の有無	疾病あり	疾病なし
曝露あり	20	10
曝露なし	12	18

これを dat という変数に付値するには，

```
dat <-  matrix(c(20,12,10,18),nc=2)
rownames(dat) <- c('曝露あり','曝露なし')
colnames(dat) <- c('疾病あり','疾病なし')
print(dat)
```

とすればよい（計算するだけなら下 3 行は不要)。matrix() は行列を定義する関数で，第 1 要素のベクトルを第 2 要素に従って並べてくれる。nc=2 は列数が 2 であることを意味し，この場合，第 1 要素のベクトルは，左上，左下，右上，右下の順に読まれる。その後，曝露と疾病が独立であるかどうかをカイ二乗検定するには，chisq.test(dat) とするだけでよい。

3.2 ある程度大きなデータを単発で入れる場合

ある程度大きなデータを入力するときは，プログラムに直接書くのは見通しが悪くなるので，データとプログラムは分離するのが普通である。同じ調査を繰り返しするとか，きわめて大きなデータであるとかでなければ，表計算ソフトで入力するのが手軽であろう。きわめて単純な例として，10 人の対象者についての身長と体重のデータが次の表のように得られているとする。

対象者 ID	身長 (cm)	体重 (kg)
1	170	70
2	172	80
3	166	72
4	170	75
5	174	55
6	199	92
7	168	80
8	183	78
9	177	87
10	185	100

この表は，表計算ソフト（Microsoft Excel や OpenOffice.org の calc など）で入力するとよい。一番上の行には変数名を入れる。日本語対応版なら漢字やカタカナ，ひらがなも使えるが，半角英数字（半角ピリオドも使える）にしておくのが無難である。ここでは，PID, HT, WT としよう（大文字と小文字は区別されるので注意)。入力が終わったら，一旦，そのソフトの標準の形式で保存しておく（ハングアップしても困らないように)。

次に，この表をタブ区切りテキスト形式で保存する。Microsoft Excel の場合，メニューバーの「ファイル (F)」から「名前を付けて保存」を選び，現れるウィンドウの一番下の「ファイルの種類 (T)」のプルダウンメニューから「テキスト（タブ区切り）(*.txt)」を選ぶと，自動的にその上の行のファイル名の拡張子も xls から txt に変わるので，「保存 (S)」ボタンを押せば OK である。複数のシートを含むブックの保存をサポートした形式でないとかいった警告がでてくるが無視して「はい」を選んでよい。その直後に Excel を終了しようとすると，何も変更していないのに「保存しますか」と聞く警告ウィン

ドウがでるが，既に保存してあるので「いいえ」と答えていい（「はい」を選んでも同じ内容が上書きされるだけだが)。

あとは R で読み込めばいい。この例のように，複数の変数を含む変数名付きのデータを読み込むときは，データフレームという構造に付値するのが普通である。保存済みのデータが D:ドライブのルートディレクトリの desample.txt だとすれば，R のプロンプトに対して，dat <- read.delim("d:/desample.txt") と打てば，データが dat というデータフレームに付値される。確認のためにデータを表示させたければ，ただ dat と打てばいいし，データ構造を見たければ，str(dat) とすればよい。読み込まれた変数に対して分析したいとき，例えばこの例の身長の平均と標準偏差を出したければ，

```
cat("mean=",mean(dat$HT),"sd=",sd(dat$HT),"\n")
```

とすればよい。一々 dat$と打つのが面倒ならば，attach(dat) とすれば，それ以降のセッション中，detach(dat) するまで，dat$を入力しなくても良くなる。例えば，このデータで身長と体重の相関係数を出して検定したいときは次のようにすればよい。

```
attach(dat)
cor.test(HT,WT)
detach(dat)
```

3.3 大量のデータあるいは継続的に何度も繰り返してとるデータの場合

Microsoft Access や Oracle，あるいは postgreSQL などのデータベースソフトを使い，入力用に設計したフォームから入力するのが一般的である（データベースソフトはバックエンドにして，PHP4 や apache httpd などを組み合わせ，ウェブアプリとするのが流行であるが，tcl/tk とデータベースアクセス用のライブラリを組み合わせ，R で直接入力システムを作ることも可能である)。それぞれ一長一短あるが，ここで詳しく説明することは大変なので，専門家に相談した方がよい。

4 R における層別の扱い方

4.1 データの例

10 人の対象者についての身長と体重のデータが次の表のように得られているとする。

対象者 ID	身長 (cm)	体重 (kg)	性別	年齢
1	170	70	M	54
2	162	50	F	34
3	166	72	M	62
4	170	75	M	41
5	164	55	F	37
6	159	62	F	55
7	168	80	F	67
8	183	78	M	47
9	157	47	F	49
10	185	100	M	45

これを 1 行目の変数名を PID, HT, WT, SEX, AGE とし，タブ区切りテキスト形式で，D:ドライブのルートディレクトリに stsample.txt というファイル名で保存したとする。次に，

```
dat <- read.delim("d:/stsample.txt")
```

と打って，データを dat というデータフレームに付値することで，分析の準備が完了する。

4.2 データフレームの一部だけの解析をする

R では，変数名の後に [ ] で条件設定をすることで，変数の一部だけを分析することが可能である。例えば，男性だけの身長の平均と標準偏差（言うまでもないが念のために書いておくと，もちろん不偏標準偏差である）を出したければ，

```
cat("mean=",mean(dat$HT[dat$SEX=='M']),"sd=",sd(dat$HT[dat$SEX=='M']),"\n")
```

とすればよい。しかし，同じ条件でたくさんの変数の一部だけの解析をしたいときに，いちいち [dat$SEX=='M'] とつけるのは面倒だろう。そういう場合は，省力化のために関数定義をしてしまおう。

```
cmeansd <- function(X,C) { cat("mean=",mean(X[C]),"sd=",sd(X[C]),"\n") }
```

としておけば，次からは，

```
cmeansd(dat$HT,dat$SEX=='M')
cmeansd(dat$HT,dat$SEX=='F')
```

などととするだけで，男女別に身長の平均と標準偏差を表示することができる。表示するのでなく，次のように平均と標準偏差の値を返すような関数定義にすれば，

```
cmeansd.noprint <- function(X,C) { list(mean=mean(X[C]),sd=sd(X[C])) }
```

得られた結果を別の関数で使うこともできる。

条件設定は一致することを意味する==だけでなく，不等号も使えるし，is.na() などの関数も使えるので，40 歳以上だけについて身長の平均値と標準偏差を計算したければ，さっき定義した cmeansd() 関数を使えばいいのだが，どうせだから N も表示するように cmeansd() 関数を再定義することにすると，

```
cmeansd <- function(X,C) {
    cat("N=",length(X[C]),"\t mean=",mean(X[C]),"\t sd=",sd(X[C]),"\n")
}
cmeansd(dat$HT,dat$AGE>=40)
```

とできるし，&（かつ）や|（または）を使ってこれらの条件を組み合わせることもできるので，40 歳以上の男性または 30 歳未満の女性についてとしたければ，やはり再定義後の cmeansd() 関数を使えば，

```
cmeansd(dat$HT,(((dat$AGE>=40)&(dat$SEX=='M'))|((dat$AGE<30)&(dat$SEX=='F'))))
```

とすればよい。条件式も変数に付値できるので，例えば 40 歳以上と未満をそれぞれ出したければ（注：“!” は論理式の否定を意味する），

```
overforty <- (dat$AGE>=40)
cmeansd(dat$HT,overforty)
cmeansd(dat$HT,!overforty)
```

とすればよい。なお，[ ] の中に数字を入れると，その順番のオブジェクトを参照することもできる。

4.3 層別の分析をする

R には，実は分類変数によって層別に任意の関数を適用する関数 tapply() が用意されている。例えば，平均値と標準偏差を返す関数 meansd() を定義してから，性別にそれを適用させるには，

```
meansd <- function(X) { list(mean=mean(X),sd=sd(X)) }
tapply(dat$HT,dat$SEX,meansd)
```

とすればよい。

5 よく使う関数一覧

5.1 一変数グラフ表示関数

度数分布図	barplot(table(C))
正規確率プロット	qqnorm(X)
ヒストグラム	hist(X)
箱ヒゲ図	boxplot(X)

5.2 一変数基本集計

度数分布表	table(C)
五数要約値［最小，Q1，中央値，Q3，最大］	fivenum(X)
サンプルサイズ	length(X)
合計	sum(X)
平均	mean(X) （sum(X)/length(X) と同値）
不偏分散	var(X) （sum((X-mean(X))^2)/(length(X)-1) と同値）
不偏標準偏差	sd(X) （sqrt(var(X)) と同値）

5.3 一変数の検定

分布の正規性（シャピロ＝ウィルクの検定）	shapiro.test(X)
母平均の検定	t.test(X,mu=**母平均**)
母比率の検定	binom.test(table(B)[2],length(B),p=**母比率**)

5.4 二変数グラフ表示

カテゴリ×カテゴリ＝モザイクプロット	mosaicplot(table(C1,C2))
量×カテゴリ＝層別箱ヒゲ図	boxplot(X~C)
量×カテゴリ＝ストリップチャート	stripchart(X~C) points(IX<-c(1.15,2.15),MX<-tapply(X,C,mean),pch=18) SX<-tapply(X,C,sd) arrows(IX,MX-SX,IX,MX+SX,code=3,angle=90,length=0.1)
量×量＝散布図	plot(Y~X) （plot(X,Y) と同値）
散布図に回帰直線を重ね書き	plot(Y~X); abline(lm(Y~X))

5.5 二変数基本集計

カテゴリ×カテゴリ＝クロス集計表	`table(C1,C2)`
量×量＝ピアソンの相関係数（量が正規分布するとき）	`cor(X,Y)`
量×量＝スピアマンの順位相関係数（量が正規分布しないとき）	`cor(X,Y,method="spearman")`

5.6 二変数の検定

次ページに，よく用いられる二変数の検定の一覧表をまとめる。なお，３つ以上の変数の関係をみたいときには無限に近いパタンがあるので，ここでは論じない。２つの変数の回帰関係に，カテゴリ変数による群間で差があるかどうかを見たい場合は共分散分析をする（例えば `summary(glm(Y~B+X+B:X))` で `B:X` の係数が有意にゼロと差があれば傾きが２群間で異なっていると判断でき，傾きに差がないとき，`summary(glm(Y~B+X))` で B をダミー変数化した変数の係数が有意にゼロと差があれば修正平均に差があると判断できる)。１つ以上の交絡要因の影響を調整して，２つのカテゴリ変数間の独立性をみたいときは，`mantelhaen.test(C1,C2,C3)` という関数（または，`TMP <- table(C1,C2,C3)` として３次元のクロス集計表 TMP を作ってから，`mantelhaen.test(TMP)` としてもよい）が使える（複数階層で２つのカテゴリ変数間の関連性，例えばオッズ比の均質性を検定するための Woolf の検定は，vcd ライブラリの `woolf.test(table(B1,B2,C))` で実行できる)。複数の独立変数で一つの変数のばらつきを説明したいときは，`glm(Y~X1+X2+...)` を使って重回帰分析やロジスティック回帰分析をしてもいい（独立変数群は線型和なら+で，交互作用だけなら:で，両方入れるときは*で結ぶ。`a*b` は `a+b+a:b` と同値)。そのとき AIC によって変数選択させたければ `step()` で囲めばいい（より細かく指定したいときは MASS ライブラリの `stepAIC()` を用いる)。

6 参考文献

1. 中澤　港 (2003)『R による統計解析の基礎』（ピアソン・エデュケーション）1,800 円
2. 間瀬 茂・神保 雅一・鎌倉 稔成・金藤 浩司 (2004)『工学のための数学３　工学のための データサイエンス入門 – フリーな統計環境 R を用いたデータ解析 – 』（数理工学社）　2,300 円
3. 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)，印刷中
4. 渡辺利夫 (2004)『フレッシュマンから大学院生までのデータ解析・R 言語』（ナカニシヤ出版)，印刷中

カテゴリ変数間の独立性のカイ二乗検定	`chisq.test(table(C1,C2))`
カテゴリ変数間の独立性の Fisher の直接確率	`fisher.test(table(C1,C2))`
オッズ比とその信頼区間	`library(vcd); summary(oddsratio(table(B1,B2),log=F))`
カテゴリ変数間の関連性：ファイ係数とクラメールのV	`library(vcd); assoc.stats(table(C1,C2))`
２回の繰り返しの一致度：カッパ係数	`library(vcd); Kappa(table(C1,C2))`
順序変数×カテゴリ変数の出現頻度の傾向＝ Cochran-Armitage の検定	`prop.trend.test(table(B,I)[2,],table(I))`
２つのカテゴリ間で正規分布する量の分散に差があるか：等分散性の検定（いわゆる F 検定）	`var.test(X~B)`
２群間で正規分布する量に差があるか（等分散のとき）：平均値の差の検定（t 検定）	`t.test(X~B,var.equal=T)`
２群間で正規分布する量に差があるか（不等分散のとき）：平均値の差の Welch の方法	`t.test(X~B)`
２群間で正規分布しない量に差があるか：Wilcoxon の順位和検定	`wilcox.test(X~B)`
対応のある２つの正規分布する量の差の検定：paired-t 検定	`t.test(X,Y,paired=T)`
正規分布する量の分散がカテゴリ間で差がないか：バートレットの検定	`bartlett.test(X~C)`
正規分布する量がカテゴリ間で差がないか：一元配置分散分析＋多重比較	等分散なら `aov(X~C)` で一元配置分散分析し，C の主効果が有意なら `TukeyHSD(aov(X~C))` または `pairwise.t.test(X,C)`。前者は Tukey の方法，後者は Holm の方法で多重比較。`library(multcomp)` すれば，`simtest(X~C,type="Dunnett")` でダネットの多重比較（対照群との比較），`simtest(X~C,type="Williams")` でウィリアムズの多重比較もできる。不等分散でもやってしまう場合もあるが，クラスカル・ウォリスの検定を使うこともある。
正規分布しない量×カテゴリ変数：クラスカル・ウォリスの検定＋ホルムの方法で調整した Wilcoxon の順位和検定を多重実行	`kruskal.test(X~C)` と `pairwise.wilcox.test(X,C)`
量×量：無相関の検定（量が正規分布するとき）	`cor.test(X,Y)`
量×量：無相関の検定（量が正規分布しないとき）	`cor.test(X,Y,method="spearman")`
測定誤差がない量（または原因と考えられる量）X によって，結果と考えられる量 Y のばらつきを説明：回帰分析	`summary(lm(Y~X))`