Rにおけるデータ入力と層別の方法

（注）この文書において，\は，半角の￥記号を意味する。

1　データ入力の方法

1.1　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。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')
```

とすればよい（計算するだけなら下 2 行は不要）。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。

1.2　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，表計算ソフトで入力する。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)
```

1.3 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 などのデータベースソフトを使ってフォームを作って入力するのが一般的である。または，html でフォームを書いて，cgi でデータ化するのでもよい。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，専門家に相談することも検討すべきであろう。

2 R における層別の扱い方

2.1 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。

2.2 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。

2.3 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。より詳しくは，http://phi.ypu.jp/swtips/R.html からリンクを辿られたい。マニュアルの有志による邦訳も公開されている（最新版ではないが）。