医学統計学演習：資料 2

芳賀昭弘 *

1　復習：R を使った計算とデータの読み書き

R を起動し、適切な場所（前回作成した本演習用のディレクトリ）に「ディレクトリの変更（移動)」後、次のようにデータを読み込んでみよう。

```
> data <- read.csv("EOM2.csv")
> data
```

EOM2.csv ファイルがデータという名前のデータフレームで読み込まれて、その内容が記述される。
（ヒント）：ダブルクォーテーションは pdf からコピーすると R 上うまく反映しませんのでご注意を。

2　スクリプト

emacs や R のテキストエディタを使って以下が書かれている内容を記載し、test_script.R という名前で保存しよう。

```
# test_script.R の内容
data <- read.csv("EOM2.csv")
print(data)
# ここまで
```

print(data) は cat(data) でもよい。次に R のコンソール画面で次のように書くと · · ·

```
> source("test_script.R")
```

1 の復習でやったのと同じ結果が出力されましたね。つまり、これまで 2 行で与えていた結果が、> source("test_script.R") と 1 行打つだけで得られるようになった。source("***.R") は***.R に書かれた内容を順に実行していくということ。このような方法を**スクリプト**と呼び、ファイルをスクリプトファイルと呼ぶ。これを応用し、次のような内容に test_script.R を書き換えて実行してみよう。

```
# test_script.R の内容
data <- read.csv("EOM2.csv")
subtra.0 <- data$Just.after - data$Before
```

* Electronic address: haga@tokushima-u.ac.jp

```
subtra.1 <- data$One.hr.after - data$Before
EOM3 <- data.frame(data,subtra.0,subtra.1)
print(EOM3)
# ここまで
```

処理の内容が理解できているでしょうか？１行目でデータを読み込み、投与後と投与前の差分（２行目)、投与後１時間と投与前の差分（３行目）を計算し、それを元のデータにくっつけています（４行目)。どうでしょう？これがどれほど便利なものか、まだわからないと思います。しかし、同じような処理を繰り返す必要がある場合、スクリプト処理は非常に便利になります。実際、スクリプトを使えばどのような処理でも、R のコンソール画面では１行（source(***.R)）書けば済むことになります。本講義では簡単な統計解析しか行わないため、スクリプト処理を行う必要がないものばかりですが、スクリプトという考え方に慣れる目的で、処理をなるべくスクリプトファイルに書いて実行することにします。

3　関数を作る

関数は、入力値 (x) に応じた出力値 ($f(x)$) を与えるものと考えることができる。スクリプトでも同じことができるが、function() という関数が R で提供されているのでそちらを使うほうが便利な場合がある。それでは、次のような内容が書かれた varp.R ファイルを作成してください。

```
# varp.R の内容
varp <- function(x){
sampvar <- var(x)*(length(x)-1)/length(x)
sampvar
}
# ここまで
```

これは標本分散を計算する関数です（var は R で与えられている不変分散を計算する関数だが、ここで作った varp は標本分散を計算する)。保存したら、

```
> source("varp.R")
```

と打っておく。これで、varp 関数を R で使うことができるようになる。試しに、EOM2.csv の投与前（Before）の標本分散を求めてみよう。

```
> varp(data$Before)
```

標本分散が正しく出力されているでしょうか？[*1]

今回の例では値を引き渡す変数（引数という）x はベクトルでした。スカラーを引数にしたり高次元行列を引数にしたりすることも勿論可能。複数の変数を引数にすることも可能（→ 練習問題)。

練習問題： x と y を入力すると、次式の値を出力する関数を作成せよ。

$x^2 + y^2 - 2xy$

また、$x = 5$、$y = -2$ の結果を出力せよ。

ヒント：プログラム上の関数名を fxy とすると、x, y の二変数なので、関数は fxy <- function(x,y) とする。

[*1] var(data$Before) と比べてみよう。どちらが大きい？その理由は？

4　グラフの作成

データを得たら、先ずグラフ化して眺めてみることが肝要です。グラフに描くことで、データの傾向が何となく掴めるということだけでなく、データに異常がないか、データ収集やデータの前処理に間違いがないかを確認するという意味でも、決して欠かしてはいけない非常に大事なステップであると言えます。R には様々なグラフ描画機能がついています。R のグラフ機能をマスターすれば、R を使っていつでもどこでもお好みのグラフを作ることが可能となります。ここでは、ヒストグラムと２次元プロットの描画方法について説明します。

4.1　ヒストグラム（度数分布表）

EOM2.csv の投与前（Before）のデータは次のように見ることができる。

```
> data$Before
```

この変数を 0.1 間隔あたりの頻度で表したものが次の表で、これを度数分布表と呼ぶ。

投与前	人数
$0.1 \sim 0.2$	4
$0.2 \sim 0.3$	3
$0.3 \sim 0.4$	1
$0.4 \sim 0.5$	0
$0.5 \sim 0.6$	1

これを棒グラフで表したのが、ヒストグラムである。R では度数分布表をわざわざ作成しなくても、次のようにすればデータからヒストグラムを作ることができます。

```
> hist(data$Before)
```

ヒストグラムはデータの傾向を直感的に理解しやすくする一方、変数間隔の幅（ビン幅, Bin size）によって印象も変わってくる。ビンの数（ヒストグラムの棒の数）を κ とおくと、ビン幅として $\lceil \frac{\max(x)-\min(x)}{\kappa} \rceil$ を使う。データ数を n とした場合、R ではデフォルトで二項分布で適切な幅として得られているスタージェスの公式 $\kappa = 1 + \log_2 n$ で幅を決定しているので注意が必要である。他にも $\kappa = \sqrt{n}$ を使ったり、データに合わせて適切な値を設定してももちろん良い。ビン幅を変えるには、次のように breaks を使う；

```
> hist(data$Before, breaks=seq(0,1,0.05))
```

この例では、$0 \sim 1$ の範囲を 0.05 のビン幅でヒストグラムを生成している。

4.2　２次元プロット

２次元プロットでは、２つの変数を点や線で描画する。上のヒストグラムを２次元プロットで表示してみよう。

```
> h <- hist(data$Before)
> plot(h$mids, h$counts, xlim=c(0.1,0.6), ylim=c(0,5), type = "l")
```

１行目でヒストグラムデータを h という変数に格納し、２行目でビンの中間値（0.1 0.2 の場合は 0.15）を横軸（h$mids のこと）、対応する人数を縦軸（h$counts のこと）として横軸の範囲（xlim=c(0.1,0.6)）と縦軸

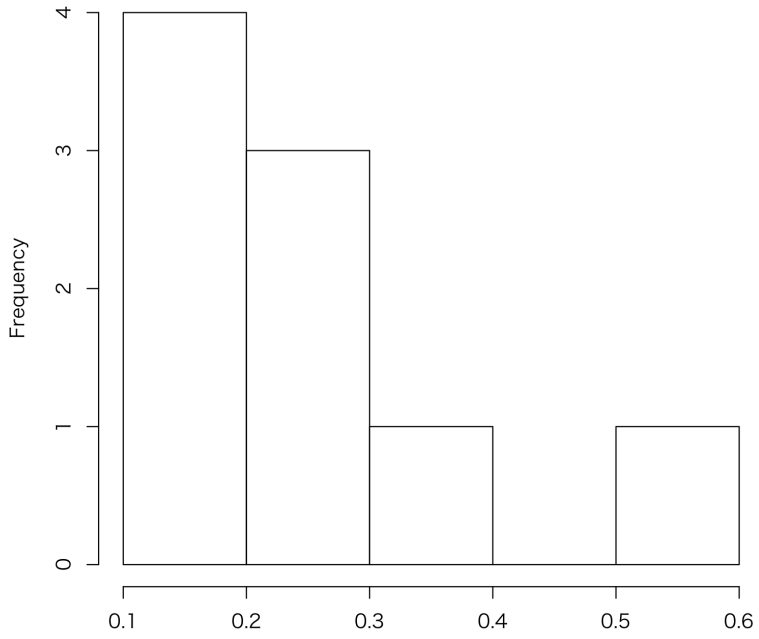

図 1　data$Before のヒストグラム

の範囲（ylim=c(0,5)）を指定して、線でプロット（type = “l”）した結果が図.2 になります。これを図.1 に重ねて表示してみましょう。

```
> h <- hist(data$Before, xlim=c(0.1,0.6), ylim=c(0,5), xlab="", ylab="")
> par(new=T)
> plot(h$mids, h$counts, xlim=c(0.1,0.6), ylim=c(0,5), type = "l")
```

２行目の par(new=T) を挿入することで、グラフを重ねて表示することができます（図.3)。１行目で挿入した xlab=“”, ylab=“”は、横軸と縦軸のラベルを空白にしたことを意味します。

１回の講義で網羅的に説明することは不可能なほど、R によるグラフ作成にはたくさんの機能があります。参考書を読んだり、使いたい機能をウェブで検索するなどして各自で学ぶようにしてください。

練習問題： グラフの線の太さを太くし且つ色を赤に変更せよ。

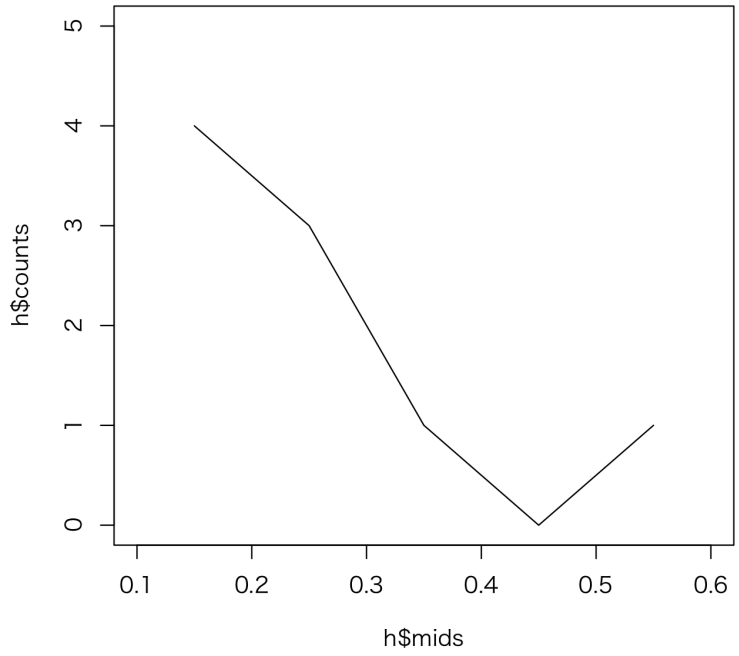

図 2　2 次元プロットの例

図 3　2 次元プロットとヒストグラムの重ね合わせ

5 演習

1. 正の整数である a と b（$a < b$）を入力すると、a から b までの整数の和を与える関数を作成せよ。
   また、$a = 5$、$b = 10$ の結果を出力せよ。
   ヒント：プログラム上の関数名を *linspace* とすると、a, b の二変数なので、関数は linspace <- function(a,b) とする。
   繰り返し演算は for 文を使うこと。for (i in a:b) とすると、i に a から b までの整数が入る。

```
for ( i in a:b) {
    func1 = func1 + i
}
```

2. 前節でおこなった 2 次元プロットとヒストグラムの重ね合わせのスクリプトファイルを作成せよ。（ヒント：EOM2.csv を読み込んで、図 3 が出力されるスクリプトを作る）
3. 上記のスクリプトで、“Before”を “Just.after”に変えて実行してみよう。結果は期待通りだったでしょうか？期待通りではなかった場合、何が悪かったのでしょうか？

manaba のレポートにスクリプトをコピー＆ペーストで貼り付けて解説すること。

引数	機能
type="p"	点プロット(デフォルト)
type="l"	線プロット(折れ線グラフ)
type="b"	点と線のプロット
type="c"	"b" において点を描かないプロット
type="o"	点プロットと線プロットの重ね書き
type="h"	各点から x 軸までの垂線プロット
type="s"	左側の値にもとづいて階段状に結ぶ
type="S"	右側の値にもとづいて階段状に結ぶ
type="n"	軸だけ描いてプロットしない(続けて低水準関数でプロットする場合)

引数	機能
log="x"	"x" (x 対数軸) , "y" (y 対数軸) , "xy" (両対数軸) の何れかを指定することが出来る (対数は常用対数のみ) .
xlim=c(0, 1), ylim=c(-1, 1)	長さ 2 のベクトルで x 座標, y 座標の最小値と最大値を与える. 他にも xlog, ylog で対数プロットが出来る. ベクトルを降順に並べる (例 : c(2, -2)) と, プロットの向きが逆になる.
axes=FALSE	軸の生成を抑制する. 軸の他に表題, 刻み, 目盛も描くかどうかを論理値で指定する (省略時は TRUE) . 他に xaxs, yaxs が指定出来る.

引数	機能
main="Title"	タイトルを与える文字列を指定する. この引数を省略するとは表題は描かれない.
sub="SubTitle"	サブタイトルを与える文字列を指定する. この引数を省略すると副題は描かれない.
xlab="X-Label", ylab="Y-Label"	それぞれ x 座標名, y 座標名を与える文字列を指定する. 省略すると, x 軸のデータとして与えられた引数の名前が座標名として描かれる.
ann = F	軸のラベルを描かないようにすることも出来る (xlab = "", ylab = "" を同時に指定した場合と同じ.) .
tmag=1.2	プロットの別の注釈するテキストに関する主なタイトルのテキストの拡大率を指定する.

図 4　R によるグラフ作成（参考 HP: http://cse.naro.affrc.go.jp/takezawa/r-tips/r/48.html）