22S:138
Lab session 1
Using R functions for simple Bayesian analysis

Aug. 29, 2003

1 Getting started

Log onto the lab computers using you HawkID and password. If you do not know these, you can use the temporary password for the day.

2 The course home page

Bring up Internet Explorer or Netscape. Type in the URL (address) of the course homepage:

`www.stat.uiowa.edu/~kcowles/s138_2000`

We will download two files from the "Handouts" section. Right-click on the file name `bayes`; then choose "Save link as" or "Save target as." Save the file in the "Temp" folder if you want to use it only during this lab. If you wish to access the file later without having to download it again, save it to your own floppy disk in the A: drive.

Right-click and save `plot.beta` as well.

3 Using my R functions

I have written some functions to make some simple Bayesian calculations and graphics easy. Those of you who know R or Splus are welcome to write and use better functions of your own, but these of mine are adequate for the assigned work at the beginning of the semester. They are based on the Minitab and Matlab macros by Jim Albert. Later in the semester we will use the software package WinBUGS to fit more complicated models.

Bring up R from the All Programs menu.

You will need to use the "source" command to install each of my functions in your R work area before using it.

To install my function to do simple sequential Bayesian analysis with Bayes theorem, and to give it the name "s138.bayes," enter the following command:

```
> s138.bayes <- source("c:\\temp\\bayes.txt")$value
```

To run an R function, enter its name, followed by parentheses.

```
> s138.bayes()
```


Try using this function to work the problem we did in lecture with Tasha's disease status and the screening test. The prompts, and your appropriate responses are as follows:

```
Input number of models, followed by return key
(Leave blank and hit return to exit):
(Leave blank and hit return to exit):
1: 2
2:
Input names of models, one on each line:
1: disease
2: no disease
3:
Input prior probabilities of models, one on each line:
1: .001
2: .999
3:
Input number of possible outcomes, followed by return key:
1: 2
2:
Input the name of each possible outcome, one on each line:
1: +
2: -
3:
[1] "2"          "disease"    "no disease" "0.001"      "0.999"
Input the likelihood of each possible outcome under each model:
Model 1 :
1: .95
2: .05
3:
Model 2 :
1: .05
2: .95
3:
[1] 2 2
Table of priors and likelihoods
1       disease 0.001    0.95    0.05
2       no disease      0.999    0.05    0.95
Input number of observations, followed by return key:
1: 3
2:
Input the names of the observations, one on each line:
1: +
2: +
3:-
4:
```

The function produces the following output:

```
 Observation: +
Update Based on Bayes' Theorem
          Models   Prior   Like      Prod      Post
[1,] "disease"    "0.001" "0.95" "0.00095" "0.01866"
[2,] "no disease" "0.999" "0.05" "0.04995" "0.98134"


 Observation: +
Update Based on Bayes' Theorem
          Models     Prior   Like      Prod     Post
[1,] "disease"    "0.01866" "0.95" "0.01773" "0.2654"
[2,] "no disease" "0.98134" "0.05" "0.04907" "0.7346"


 Observation: -
Update Based on Bayes' Theorem
          Models    Prior   Like      Prod      Post
[1,] "disease"    "0.2654" "0.05" "0.01327" "0.01866"
[2,] "no disease" "0.7346" "0.95" "0.69787" "0.98134"
```

Notice that at each step, the "posterior" from the previous step becomes the new "prior."

4 An example with 3 possible models and 5 possible outcomes

I have attached an excerpt from a book by Jim Albert, which describes a sequential Bayesian analysis using generalized Bayes' theorem. The R function may be used to carry out this example as follows:

```
> s138.bayes()
Input number of models, followed by return key
(Leave blank and hit return to exit):
1: 3
2:
Input names of models, one on each line:
1: good
2: mediocre
3: poor
4:
Input prior probabilities of models, one on each line:
1: .6
2: .3
3: .1
4:
Input number of possible outcomes, followed by return key:
1: 5
2:
```



```
Input the name of each possible outcome, one on each line:
1: A
2: B
3: C
4: D
5: F
6:
[1] "3"        "good"     "mediocre" "poor"     "0.6"      "0.3"      "0.1"
Input the likelihood of each possible outcome under each model:
Model 1 :
1: .4
2: .4
3: .2
4: 0
5: 0
6:
Model 2 :
1: .1
2: .2
3: .4
4: .2
5: .1
6:
Model 3 :
1: 0
2: 0
3: .3
4: .5
5: .2
6:
[1] 3 5
Table of priors and likelihoods
1       good            0.6      0.4     0.4     0.2     0       0
2       mediocre        0.3      0.1     0.2     0.4     0.2     0.1
3       poor            0.1      0       0       0.3     0.5     0.2
Input number of observations, followed by return key:
1: 4
2:
Input the names of the observations, one on each line:
1: B
2: B
3: C
4: C
5:


 Observation: B
Update Based on Bayes' Theorem
```

```
        Models Prior  Like   Prod  Post
[1,] "good"     "0.6" "0.4" "0.24" "0.8"
[2,] "mediocre" "0.3" "0.2" "0.06" "0.2"
[3,] "poor"     "0.1" "0"   "0"    "0"


 Observation: B
Update Based on Bayes' Theorem
        Models Prior  Like   Prod      Post
[1,] "good"     "0.8" "0.4" "0.32" "0.88889"
[2,] "mediocre" "0.2" "0.2" "0.04" "0.11111"
[3,] "poor"     "0"   "0"   "0"    "0"


 Observation: C
Update Based on Bayes' Theorem
        Models     Prior  Like      Prod  Post
[1,] "good"     "0.88889" "0.2" "0.17778" "0.8"
[2,] "mediocre" "0.11111" "0.4" "0.04444" "0.2"
[3,] "poor"     "0"       "0.3" "0"       "0"


 Observation: C
Update Based on Bayes' Theorem
        Models Prior  Like   Prod      Post
[1,] "good"     "0.8" "0.2" "0.16" "0.66667"
[2,] "mediocre" "0.2" "0.4" "0.08" "0.33333"
[3,] "poor"     "0"   "0.3" "0"    "0"
```

After the student receives his first grade (B), what is the principal's posterior probability that the student is a mediocre student?

5 Printing the output from an R function

Probably the easiest way to print R output is to copy it into a Notepad or Wordpad and edit and print it from there.

6 R graphics

You won't need this for the first homework, but you will for next week's homework.

Obtain my function for plotting beta densities:

```
> s138.plot.beta <- source("c:\\temp\\plot.beta.txt")$value
```

Then run it.

```
> s138.plot.beta()
```


We'll learn about the beta density next week. For right now, you just need to remember that each of the two parameters must be greater than 0.

```
Input alpha and beta parameters for the beta density, on separate lines, followed by re
1: 3.4
2: 29.1
3:
To remove the graphics window, enter
 dev.off()
 at the prompt.
```

A new window should appear, containing a plot. In the upper border of the plot, R will tell you the "device number" of the plot, and whether or not this is the "active" device. To print the plot in the active device, enter

```
dev.print(device=win.print)
```

If you ran this function many times, you would get a new graphics window every time. If you wish to print a plot *other* than the active one, you can make a different device active. For example, to make device 2 the active device and then print the plot in device 2, enter

```
dev.set(2)
dev.print(device=win.print)
```

To close an active graphics window, enter

```
dev.off()
```

It is also possible to copy an R graph into a Word document. Right click on the plot window; then select "Copy as bitmap." Then paste into and open Word document.

7 Ending a session

R maintains what it class the "workspace," in which it stores any function or other objects that have created during a given session. If you exit from R without saving the workspace image, then any new work you have done will be lost. If you are working on a lab computer and wish to save your work to a floppy disk so that you can reload it into R on another computer later, use the "File" menu to change the working directory to the A: drive now.

To exit from R enter

```
> q()
```

You will be asked whether you want to save the workspace image. If you respond "No," then any new work done during this session will be lost. If you are working on your own computer, and you want these functions to remain associated with R so you don't have to "source" them in again next time, choose "Yes." If you are working on a lab computer and have previously changed the working directory to A:, then the workspace image will be saved there.

Be sure to log off the computer (using the All Programs menu) before you leave.