

STAT 457, Fall, 2010

Lab #4 - More “Beginnings” of R

topics:

- 2 way table using R features
- vectors and matrices
- R scripts and commands
- R graphics

=====

The table below shows the number of doctoral degrees awarded in the United States between 2001 and 2006.

Year	Engineering	Science	Education	Health	Humanities	Other
2001	5323	20643	6436	1591	5213	2159
2002	5511	20017	6349	1541	5178	2141
2003	5079	19529	6503	1654	5051	2209
2004	5280	20001	6643	1633	5020	2180
2005	5777	20498	6635	1720	5013	2480
2006	6425	21564	6226	1785	4999	2436

We want to investigate the question “Has higher education candidates changed their areas of interest since the turn of the century?”.

A good place to statistically start would be to find marginal and conditional distributions of the disciplines over the years of the study, then generate graphs and tests to determine if there is indication of relationship between the variables. If we were to do this study “by hand”, we would possibly proceed as follows.

Step 1: Make a 2 way table of the data

Step 2: Install marginal columns on our table

Step 3: Create the marginal distribution of DISCIPLINE and graph this distribution

Step 4: Compute the conditional distribution of DISCIPLINE by YEAR and graph the results with side-by-side segmented box plots

Step 5: Determine levels of independence between the variables DISCIPLINE and YEAR

I put together a series of scripts to do these steps in R. A script is a set of commands in R which are written in NOTEPAD text editor (or any basic text editor—but not WORD, because all the extra “format tags”, etc., produced by WORD confuse R). You write scripts in NOTEPAD and save them as `.txt` format, but you include a `.R` in the title block, so R distinguishes your work as a script instead of just any old `.txt` format. By putting `.R` in the title block, even though you save it as a `.txt` file extension, you get a unique little icon for the file, different from the `.txt` NOTEPAD icon you usually get. Note that you can save the file without the `.R` in the title, and still have R read it as a script, but it causes extra file searching work for you, to tell R to look for a `.txt` instead

of a .R file.

We will demonstrate the R “structure” and some of the basic R commands and routines through doing this 2 categorical investigation. I don't proclaim that this is the only way to design this problem or approach it in R, only one way which, I hope, illustrates R.

We typed up our data in `phd.txt` (written in NOTEPAD, “tab delimited format”) and is shown below.

2001	5323	20643	6436	1591	5213	2159
2002	5511	20017	6349	1541	5178	2141
2003	5079	19529	6503	1654	5051	2209
2004	5280	20001	6643	1633	5020	2180
2005	5777	20498	6635	1720	5013	2480
2006	6425	21564	6226	1785	4949	2436

Download `phd.txt` to your desktop. Also download the first script, `step1.R` to the desktop. Open up WORD, putting your name, section number, and lab number as heading, then minimize.

Open up R, minimize R Commander, and **CLEAR WINDOW** of R Console. We now want to tell R where to look for our data file and our script, so that when we tell R to “go fetch” these files, it doesn't say it can't find them! So click on **FILE CHG DIR** (for change directory) and make sure you have the Desktop clicked for the directory chosen. Now click on **FILE OPEN SCRIPT** and upload `step1.R` into R. A new window will open in the R Gui screen with the following script. All lines beginning with # and ending with a carriage return (ENTER) are called “comment lines” of the program, and are non-executable statements inserted in R programs for tutorial and explanation purposes. Once you get good at writing scripts you don't need so many notes, but it is still a good programming idea to occasionally notate your routines with explanations. Also note that for long commands (like the `colnames` one in `step1`, I put a carriage return and indent the remaining part of the command to let me know that this is still part of the original `colnames` command. I could have typed it on one long line, or put carriage returns anywhere I wanted and not indented at all, but it is a “style thing” with me. This is how I keep the width of my scripts of short enough width and still be able to tell when I have more of the same command to do. Also, I usually skip a line in my script when I do another task, to have a better visual break in my script routine. Note that all of these carriage returns, indents (i.e., hitting the space bar 5 or 6 times), and line skips do not interfere with the R execution. They are “ignored” by R. I do them just to give my scripts some form of uniform structure. Also note that the last line prints out what R has stored for the variable `phdtable`. The script is reproduced below.

```
# put file phd.txt onto the DESKTOP
# change directories by FILE CHANGE DIRECTORY
# make a table of the data from the tab delimited file
# and store the result as the variable phdtable
phdtable ← read.table("phd.txt")
```

```
# now print out phdtable on the screen
```

```

phdtable

# add column names to the table phdtable
colnames(phdtable) ← c("year", "Engineering", "Science",
    "Education", "Health", "Humanities", "Other")

# print out phdtable on the screen again
phdtable

```

Note that the string `c(...)` in the `colnames` command is called a vector (1 row by 7 column array in this case), and that for us, this vector is a string (or text) vector instead of a number type of vector.

With the script window highlighted (blue header on window) click on **EDIT **RUN ALL** to get the output called for in the script.**

This is the initial printout of `pdhhtable`. Notice how the generic titles are given the columns and rows by R.

```

> phdtable
  V1  V2  V3  V4  V5  V6  V7
1 2001 5323 20643 6436 1591 5213 2159
2 2002 5511 20017 6349 1541 5178 2141
3 2003 5079 19529 6503 1654 5051 2209
4 2004 5280 20001 6643 1633 5020 2180
5 2005 5777 20498 6635 1720 5013 2480
6 2006 6425 21564 6226 1785 4949 2436

```

This is the printout of `phdtable` with column titles.

```

> phdtable
  year Engineering Science Education Health Humanities Other
1 2001      5323   20643      6436   1591      5213   2159
2 2002      5511   20017      6349   1541      5178   2141
3 2003      5079   19529      6503   1654      5051   2209
4 2004      5280   20001      6643   1633      5020   2180
5 2005      5777   20498      6635   1720      5013   2480
6 2006      6425   21564      6226   1785      4949   2436

```

Below is the script `step2.R`.

```

# make a matrix out of the table phdtable
phdmatrix ← as.matrix(phdtable[, 2:7])
phdmatrix

# name the matrix rows
rownames(phdmatrix) ← c("2001", "2002", "2003",
    "2004", "2005", "2006")

phdmatrix

```

```
# add marginal totals to the matrix
twoway ← addmargins(phdmatrix)
twoway
dim(twoway)
```

In order to do stacked bar graphs, as well as do specific computation on our rows and columns, it is appropriate to make a matrix (which is a rectangular array of numbers) out of our table. The matrix `phdmatrix` is shown below. Notice how the rows are numbered 1 through 6 and columns have word titles. Row 3, column 4 entry (called `phdmatrix[4,3]`) has value 6643, standing for the number of education PhD's in the 4th year, 2004.

```
> phdmatrix
      Engineering Science Education Health Humanities Other
[1,]          5323    20643         6436    1591         5213    2159
[2,]          5511    20017         6349    1541         5178    2141
[3,]          5079    19529         6503    1654         5051    2209
[4,]          5280    20001         6643    1633         5020    2180
[5,]          5777    20498         6635    1720         5013    2480
[6,]          6425    21564         6226    1785         4949    2436
```

Next, we see `phdmatrix` which has row names given as the specific year, instead of the bracketed names. Again we used a text vector to list the years for row names.

```
> rownames(phdmatrix) ← c("2001", "2002", "2003", "2004", "2005", "2006")
> phdmatrix
      Engineering Science Education Health Humanities Other
2001          5323    20643         6436    1591         5213    2159
2002          5511    20017         6349    1541         5178    2141
2003          5079    19529         6503    1654         5051    2209
2004          5280    20001         6643    1633         5020    2180
2005          5777    20498         6635    1720         5013    2480
2006          6425    21564         6226    1785         4949    2436
```

Below is the completed two way table of YEAR vs DISCIPLINE, with marginal row and column values. `twoway` is a matrix, with dimensions 7 rows by 7 columns—notice the `dim` command

```
> twoway ← addmargins(phdmatrix)
> twoway
      Engineering Science Education Health Humanities Other      Sum
2001          5323    20643         6436    1591         5213    2159    41365
2002          5511    20017         6349    1541         5178    2141    40737
2003          5079    19529         6503    1654         5051    2209    40025
2004          5280    20001         6643    1633         5020    2180    40757
2005          5777    20498         6635    1720         5013    2480    42123
2006          6425    21564         6226    1785         4949    2436    43385
Sum          33395    122252         38792    9924         30424    13605    248392
> dim(twoway)
[1] 7 7
```

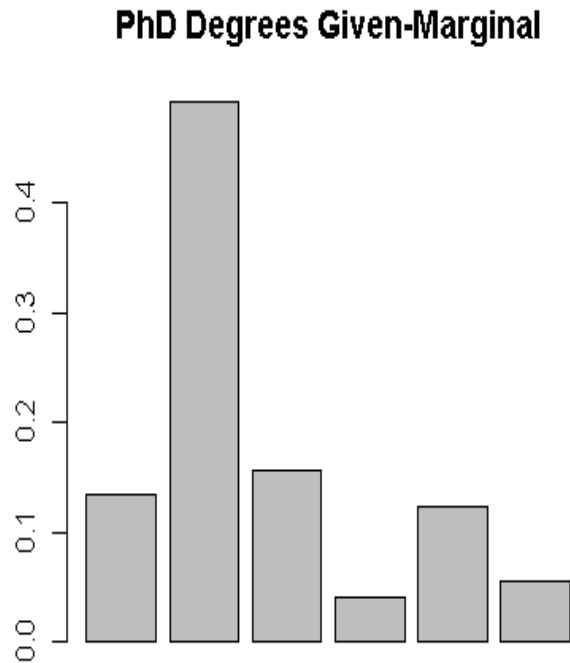
Next, we want to compute the marginal of DISCIPLINE, which is the row total of the `twoway` table, using script `step3.R`, which is shown below.

```
# create a marginal distribution and plot marginal

total.total ← twoway[7,7]
total.total
marginal.year ← c(twoway[7,1]/total.total, twoway[7,2]/total.total,
                  twoway[7,3]/total.total, twoway[7,4]/total.total,
                  twoway[7,5]/total.total,
                  twoway[7,6]/total.total)
marginal.year
```

```
barplot(marginal.year, main="PhD Degrees Given-Marginal")
```

Below is the side-by-side bar graph of the marginal given by step3.R .



Finally, step4.R is shown below, and computes the required conditional distributions.

```
# forming conditonal distribution of degree by year
# then making segmented bar graph

row.tot1 ← twoway[1,7]; row.tot2 ← twoway[2,7];
  row.tot3 ← twoway[3,7]; row.tot4 ← twoway[4,7];
  row.tot5 ← twoway[5,7]; row.tot6 ← twoway[6,7]
rowtot1
cond.2001 ← c(twoway[1,1]/row.tot1,twoway[1,2]/rowtot1,
  twoway[1,3]/rowtot1, twoway[1,4]/rowtot1,
  twoway[1,5]/rowtot1, twoway[1,6]/rowtot1)
cond.2001
cond.2002 ← c(twoway[2,1]/row.tot2,twoway[2,2]/rowtot2,
  twoway[2,3]/rowtot2, twoway[2,4]/rowtot2,
  twoway[2,5]/rowtot2, twoway[2,6]/rowtot2)
cond.2003 ← c(twoway[3,1]/row.tot3,twoway[3,2]/rowtot3,
  twoway[3,3]/rowtot3, twoway[3,4]/rowtot3,
  twoway[3,5]/rowtot3, twoway[3,6]/rowtot3)
cond.2004 ← c(twoway[4,1]/row.tot4,twoway[4,2]/rowtot4,
  twoway[4,3]/rowtot4, twoway[4,4]/rowtot4,
  twoway[4,5]/rowtot4, twoway[4,6]/rowtot4)
cond.2005 ← c(twoway[5,1]/row.tot5,twoway[5,2]/rowtot5,
  twoway[5,3]/rowtot5, twoway[5,4]/rowtot5,
  twoway[5,5]/rowtot5, twoway[5,6]/rowtot5)
cond.2006 ← c(twoway[6,1]/row.tot6,twoway[6,2]/rowtot6,
  twoway[6,3]/rowtot6, twoway[6,4]/rowtot6,
```

```

twoway[6,5]/rowtot6, twoway[6,6]/rowtot6)
cond.year ← c matrix(c(cond.2001, cond.2002, cond.2003, cond.2004,
                      cond.2005, cond.2006), nrow=6)
cond.year

# give column and row names to cond.year matrix

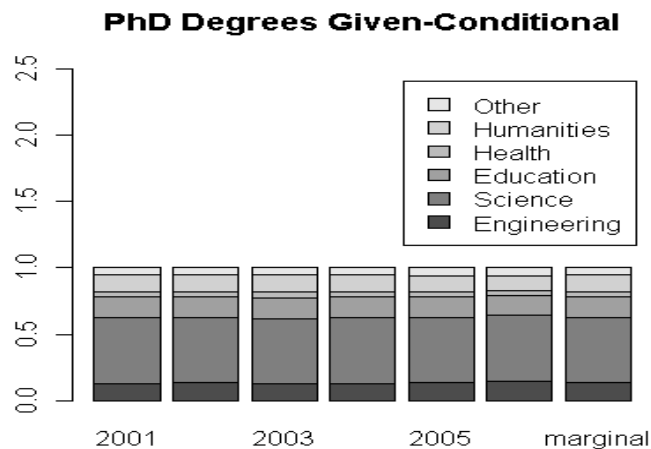
colnames(cond.year) ← c("2001", "2002", "2003", "2004", "2005",
                       "2006", marginal.year)
rownames(cond.year) ← c("Engineering", "Science", "Education",
                       "Health", "Humanities", "Other")
cond.year

# finally make a barplot of conditional distribution showing legend
# and another barplot with smaller y range to show detail
# of conditional distribution
# also show marginal segmented bar to compare to the conditionals

barplot(cond.year, main="PhD Degrees Given-Conditional",
        legend.text=TRUE, ylim=c(0, 2.5))
barplot(cond.year, main="PhD Degrees Given-Conditional",
        ylim=c(0, 1.0))

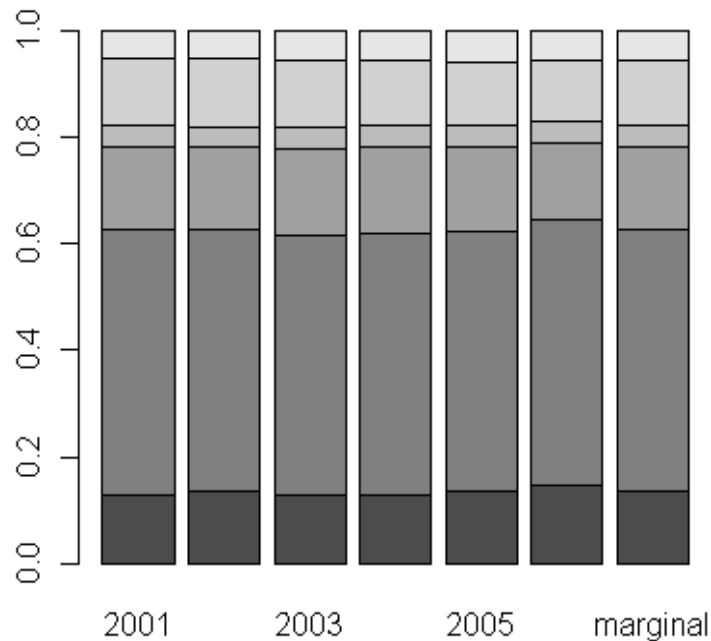
```

Below is the side-by-side segmented bar graph,



and another copy with a more magnified y axis scale (without the legend), so you can see the subtle distinctions. When looking at the bar graph with legend shown above, it seems like all 7 distributions are exactly alike, but with the more expanded y axis, you see that there is a slight bit of difference from year to year.

PhD Degrees Given-Conditional



Below is the matrix `cond.year`, with proper column and row names installed, from which the segmented bar graphs were generated from.

```
> colnames(cond.year) <- c("2001", "2002", "2003", "2004", "2005", "2006", "marginal")
> rownames(cond.year) <- c("Engineering", "Science", "Education", "Health",
+ "Humanities", "Other")
> cond.year
      2001      2002      2003      2004      2005      2006
Engineering 0.12868367 0.13528242 0.12689569 0.12954830 0.1371460 0.14809266
Science     0.49904509 0.49137148 0.48792005 0.49073779 0.4866225 0.49703815
Education   0.15559048 0.15585340 0.16247345 0.16299041 0.1575149 0.14350582
Health      0.03846247 0.03782802 0.04132417 0.04006674 0.0408328 0.04114325
Humanities  0.12602442 0.12710803 0.12619613 0.12316903 0.1190086 0.11407168
Other       0.05219388 0.05255664 0.05519051 0.05348774 0.0588752 0.05614844
marginal
Engineering 0.13444475
Science     0.49217366
Education   0.15617250
Health      0.03995298
Humanities  0.12248382
Other       0.05477230
```

Some comments on the code of `step4.R` script.

`row.tot1`, `row.tot2`, etc. are all assignments of the total values in each column. You can put multiple R commands on one line, instead of using a carriage return between each command, by placing a semicolon (;) as I did, if you want.

`cond.2001`, `cond.2002`, etc. are all the conditionals of DISCIPLINE by each YEAR.

We had to make a matrix out of `cond.year`, to make the barplots in R, as we did before for `marginal.year` -note that `cond.year` was designated with 6 rows in the -7-

`matrix` command, corresponding to each year.

Note the difference in `barplot` commands. The first one had to be scaled from 0 to 2.5 (indicated by the `c` vector) on the y axis in order to fit the legend properly, whereas I could rescale the y axis from 0 to 1.0 for the second `barplot`, without having a legend. The default value for a legend is `FALSE`, so if you don't specify `legend.text=TRUE` (or any reference to legend like I did) then you don't have a legend produced!

I "built" the matrix `cond.year` a column at a time, where first column was `cond.2001`, second column was `cond.2002`, etc., with last column being `marginal.year`. If I had not put the command `ncol=6`, I would have had a matrix one column by 42 rows, with rows 1 through 6 being `cond.2001`, rows 7 through 12 being `cond.2002`, etc.

=====

[1] Your WORD lab report this week will be to reproduce all input and output from all 4 scripts (`step1.R` through `step4.R`), plus all graphs generated from these scripts. This is the easy part of the lab. The hard part is the following assignment. I want you to spend most of your time studying each part of this lab, so you start to see what structures exist in R and try to understand what is going on here. Please ask questions of me for any clarifications. You may have to refer to the R tutorials posted on our web site or on the CRAN web site at times, and you will spend most of your quality time on this lab just thinking about what you see here. The more you can understand about what I did here, the easier future work in R will be as we progress in our understanding of R.

I do not profess that the way we did this two way table analysis is the most efficient way in R to do this task of determining independence of YEAR and DISCIPLINE. Hopefully, it is illustrative of much of the architecture of R which you must learn.

By the way, since the conditional distributions (as shown especially by the segmented bar graph) are all quite similar, YEAR seems independent of DISCIPLINE, and upper division academics seem to consistently pursue the same types of higher level degrees every year. This shows also in that the Marginal of DISCIPLINE looks like all the conditionals.

=====

EXTRA CREDIT

If you think you have mastered this, do problem #29 on page 42 of your STAT 451 text, in R; that is, use R to determine if working parents have changed attitudes about working and family obligations in the two polls taken. For our example above, we had 6 categories of YEAR and 6 categories of DISCIPLINE. For this problem #29, there are 2 categories of YEAR and 5 categories of RESPONSE. Remember, this is extra credit, so it is not mandatory for this lab. Copy/paste scripts developed, graphs produced, and output results on your WORD document.