Section 2.1: Displays of Categorical Data

Thomas Scofield

2/5/2015

Some of the plots/displays of data that you see in Chapter 2 are reproduced in this document. It is possible, generally speaking, to do this only if I have access to their data. When I did not know of a way to access their data, I produce a similar display using the Math 143 Calvin student survey data from 2004, which I load next, and give the name **ss**:

```
ss=read.csv("http://www.calvin.edu/~scofield/data/csv/ssurv.csv")
head(ss)
```

```
##   gender class gpa height pulse childrank numchildren haircut randomnum
## 1      F    So 3.6     NA    NA         2           3    2.00         6
## 2      F    So 3.4     NA    NA         4           4   10.00         7
## 3      M    Fr 3.0     71    68         2           4    0.00        17
## 4      M    So 2.6     72   100         2           1   15.00         3
## 5      M    So 2.2     68   101         4           3   11.00        13
## 6      M    So 2.4     72    74         2           2    9.99        11
##   speedtickets cds smoker hourssleep selfhandedness momhandedness
## 1            0  15    Non        8.0              R             L
## 2            0  10    Non        8.0              R             R
## 3            2  53    Non        5.5              R             R
## 4            3 170  Smoke        7.0              L             R
## 5            0  55  Smoke        8.0              R             R
## 6            0 101    Non        6.0              R             R
##   dadhandedness   region oncampus cupscoffee birthday overtwenty
## 1             R Suburban        Y          1       Th          Y
## 2             R    Rural        Y          0       Fr          N
## 3             R Suburban        Y          0       Th          N
## 4             R Suburban        Y          2       We          Y
## 5             R Suburban        Y          0       Mo          N
## 6             R Suburban        Y          0       Th          N
```

One Categorical Variable (p. 47)

I will use the categorical variable **selfhandedness** found in the **ss** data frame. We can produce a frequency table of the values in this variable, much like Table 2.1:

```
xtabs(~selfhandedness, data=ss)
```

```
## selfhandedness
##       L   R
##   1  31 248
```

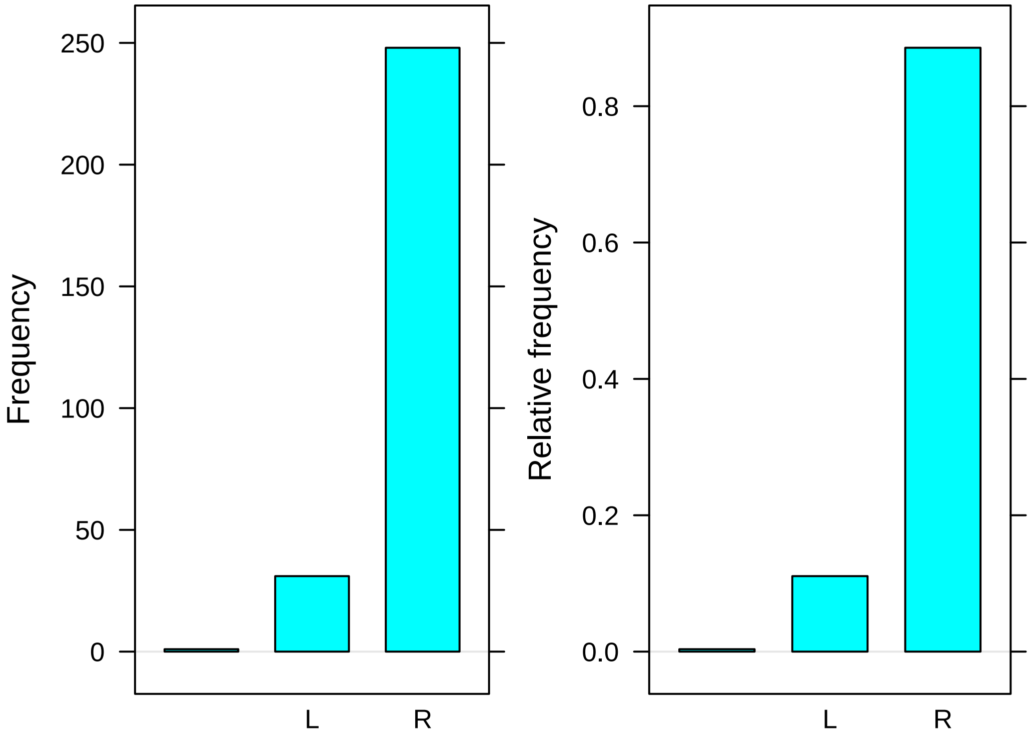
Evidently, there were 31 left-handed respondents, 248 right-handed ones, and 1 who did not respond. If, instead, we want a *relative frequency* table, like the one in Table 2.2, we do what we did above, then ask the results to be converted to proportions:

```
prop.table(xtabs(~selfhandedness, data=ss))
```

```
## selfhandedness
##                       L           R 
## 0.003571429 0.110714286 0.885714286
```

To get a bar chart as in Figure 2.1(a), we can do something like these, one which gives frequencies, and the other (more complicated) one giving relative frequencies:

```
bargraph(~selfhandedness,data=ss)
barchart(prop.table(xtabs(~selfhandedness,data=ss)),horizontal=FALSE,ylab="Relative frequency")
```



Though the book does it, I do not produce here an example of a pie chart. Rest assured, RStudio can do them. However, I side with those who think pie charts are a bad idea. Look at this graphic for some insight into why I think bar graphs are quite superior: http://www.calvin.edu/~stob/courses/m241/F10/pie.jpg

Two-Way Tables (p. 49)

The Lock book refers to a data set containing different survey data (i.e., not from Calvin students, different set of questions). It is found in the data frame called **StudentSurvey**. We produce a two-way table for the two categorical variables "gender" and "preferred award", as in Table 2.5. First, however, I look at the variable names to see what RStudio calls them. As this document is being produced using R Markdown, which seems unaware of the packages available at the RStudio Console, I must also load the **Lock5withR** package:

```
require(Lock5withR)
```

```
## Loading required package: Lock5withR
##
## Attaching package: 'Lock5withR'
##
## The following object is masked from 'package:datasets':
##
##     CO2
```

```r
names(StudentSurvey)
```

```
##  [1] "Year"       "Gender"     "Smoke"      "Award"      "HigherSAT"
##  [6] "Exercise"   "TV"         "Height"     "Weight"     "Siblings"
## [11] "BirthOrder" "VerbalSAT"  "MathSAT"    "SAT"        "GPA"
## [16] "Pulse"      "Piercings"  "Sex"
```

The variables of interest to us are called **Sex** and **Award**. Next, we make a 2-way table:

```r
xtabs(~Sex + Award, data=StudentSurvey)
```

```
##         Award
## Sex       Academy Nobel Olympic
##   Female       20    76      73
##   Male         11    73     109
```

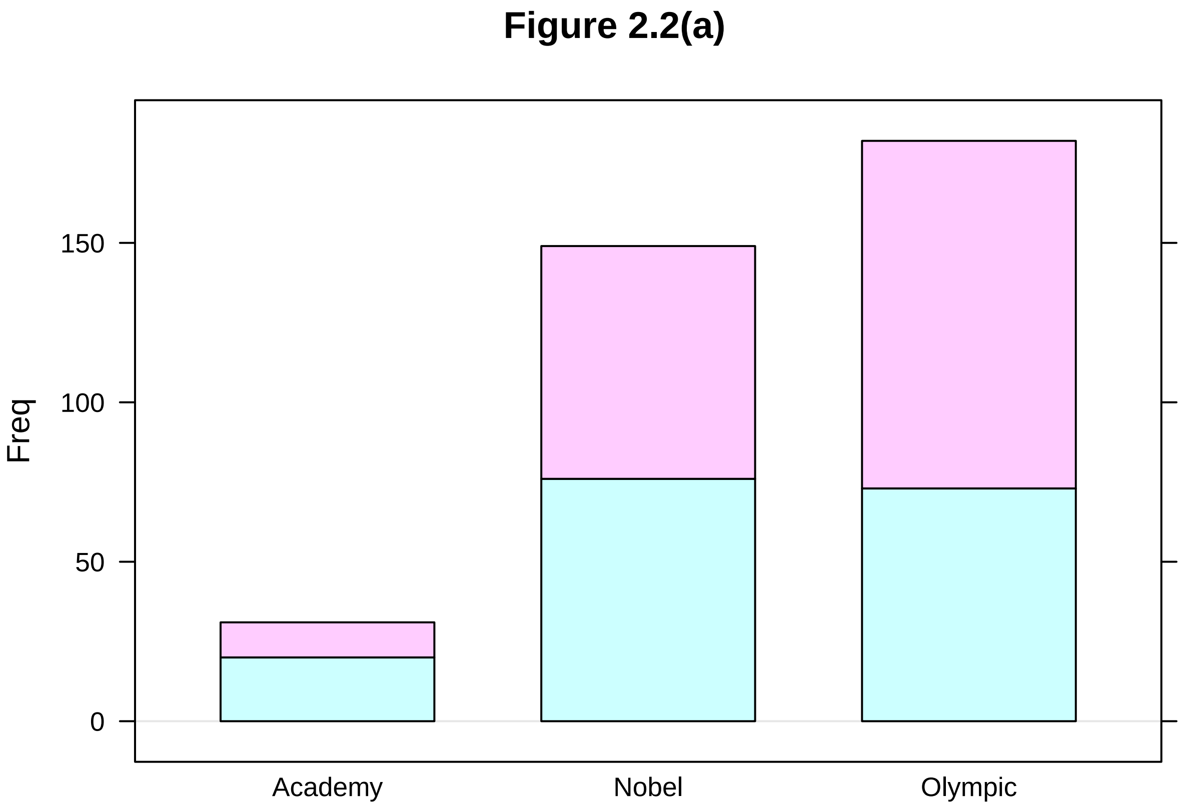
Actually, the result differs from Table 2.5 in that it has totals (also known as marginal totals) for each row and column. We can get this as well, processing the table with an extra addmargins() command:

```r
addmargins(xtabs(~Sex + Award, data=StudentSurvey))
```

```
##         Award
## Sex       Academy Nobel Olympic Sum
##   Female       20    76      73 169
##   Male         11    73     109 193
##   Sum          31   149     182 362
```

We mimick the bar charts on p. 53 below.

```r
barchart(xtabs(~Award + Gender, data=StudentSurvey), horizontal=FALSE, main="Figure 2.2(a)")
```



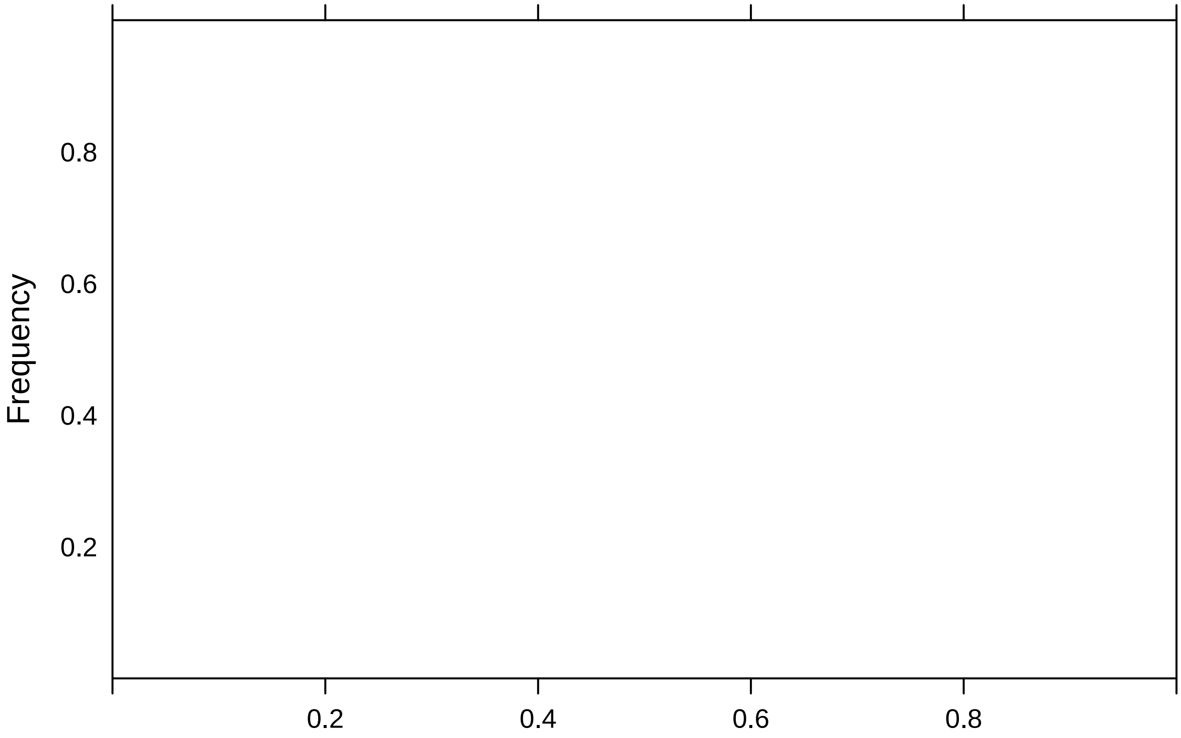
You might try out modifications to the above on your own, to see how changes affect the plots. Here are two possible modifications:

```
barchart(xtabs(~Sex + Award, data=StudentSurvey), horizontal=FALSE)
barchart(xtabs(~Award + Gender, data=StudentSurvey))
```

Next, using what seems to me a much simpler command, I produce something like Figure 2.1(b):

```
bargraph(~Award+Sex, data=StudentSurvey, groups=Sex)
```

```
## Warning in Ops.factor(Award, Sex): '+' not meaningful for factors
```


Frequency
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8