

Statistics 120 Mosaic Plots

Music Listening

Finally, we can compute the proportion of people who listen to classical music in each of the age/education groups.

<i>Old</i>		<i>Young</i>	
<i>High Ed.</i>	<i>Low Ed.</i>	<i>High Ed.</i>	<i>Low Ed.</i>
52.5%	18.9%	32.3%	27.5%

The music-listening habits of younger people seem to be fairly independent of education level. This is not true for older people.

Who Listens To Classical Music?

The following table of values shows a sample of 2300 music listeners classified by age, education and whether they listen to classical music.

	Education			
	<i>High</i>		<i>Low</i>	
	Classical Music			
Age	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
<i>Old</i>	210	190	170	730
<i>Young</i>	194	406	110	290

This is a $2 \times 2 \times 2$ contingency table.

Summary

The result of our “analysis” is a series of tables. From these tables we can see:

1. There are slightly more old people than young people in the sampled group.
2. The younger people are more highly educated than the older ones.
3. The likelihood of listening to classical music depends on both age and education level.

Old Versus Young

The effect of age and education on musical taste can be investigated by breaking the observations down into more homogenous groups. The most obvious split is by age. There are 1300 older people and 1000 younger people.

<i>Old</i>	<i>Young</i>
56.5%	43.5%

This is almost certainly a result of the way in which the sample was taken.

Mosaic Plots

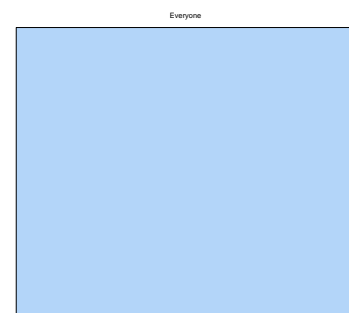
- Mosaic plots give a graphical representation of these successive decompositions.
- Counts are represented by rectangles.
- At each stage of plot creation, the rectangles are split parallel to one of the two axes.

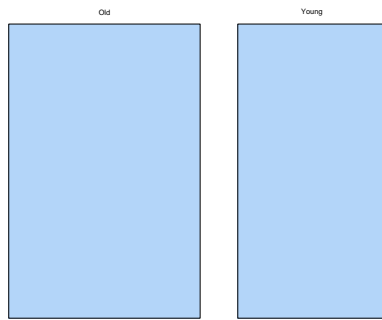
Education Level

Within the old and young groups we can now find the proportions falling into each of the high and low education categories.

<i>Old</i>		<i>Young</i>	
<i>High Ed.</i>	<i>Low Ed.</i>	<i>High Ed.</i>	<i>Low Ed.</i>
30.8%	69.2%	60.0%	40.0%

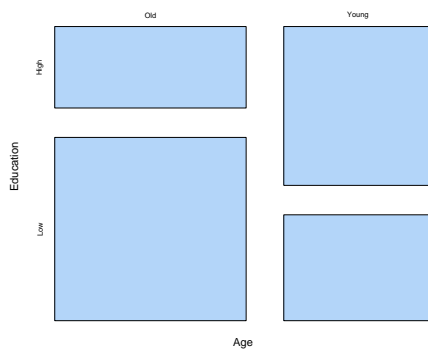
The *young* group is clearly more highly educated than the *old* group.





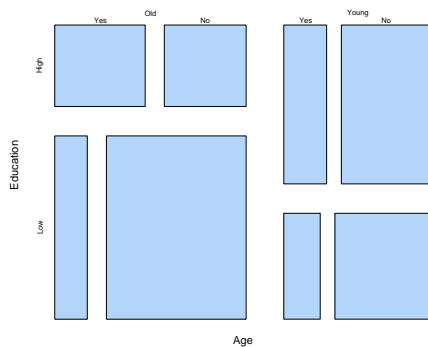
Creating Mosaic Plots

- In order to produce a mosaic plot it is necessary to have:
 - A contingency table containing the data.
 - A preferred ordering of the variables, with the “response” variable last.



Entering the Data

- To enter the data, we must settle on the order in which the values.
- The order of values in an R array is with the first subscript varying most quickly, the second subscript varying next most quickly, etc.
- In the case of the music data we can take the first subscript to correspond to *Age*, the second to *Education* and the third to *Listening*.
- The steps are then (i) entering the data, (ii) shaping it as an array and (iii) labelling the extents.



Order for Data Entry

	Education			
	<i>High</i>		<i>Low</i>	
	Classical Music			
Age	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>
<i>Old</i>	210	190	170	730
<i>Young</i>	194	406	110	290

	Data Order			
1	5	3	7	
2	6	4	8	

The Perceptual Basis for Mosaic Plots

- It is tempting to dismiss mosaic plots because they represent counts as rectangular areas, and so provide a distorted encoding.
- In fact, the important encoding is length.
- At each stage the comparison of interest is of the lengths of the sides of pieces of the most recently split rectangle.

Data Entry

```
> music = c(210, 194, 170, 110,
            190, 406, 730, 290)

> dim(music) = c(2, 2, 2)

> dimnames(music) =
  list(Age = c("Old", "Young"),
        Education = c("High", "Low"),
        Listen = c("Yes", "No"))
```

Data Inspection

```
> music
, , Listen = Yes

      Education
Age   High Low
Old   210 170
Young 194 110

, , Listen = No

      Education
Age   High Low
Old   190 730
Young 406 290
```

The Data

- There is very good documentation on who survived and who did not survive the sinking of the Titanic.
- R has a data set called “Titanic” which gives data on the passengers on the Titanic, cross-classified by:
 - Class: 1st, 2nd, 3rd, Crew.
 - Sex: Male, Female.
 - Age: Child, Adult.
 - Survived: No, Yes.

Producing A Mosaic Plot

The R function which produces mosaic plots is called `mosaicplot`. The simplest way to produce a mosaic plot is:

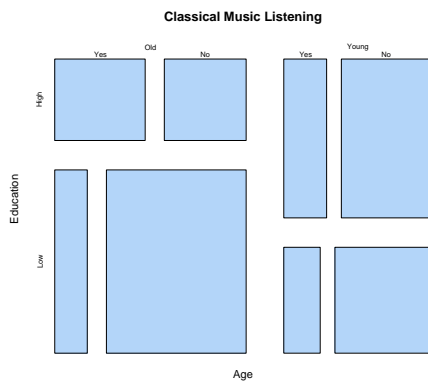
```
> mosaicplot(music)
```

It is also easy to colour the plot and to add a title.

```
> mosaicplot(music, col = hcl(240),
             main = "Classical Music Listening")
```

Adults	Survivors		Non-Survivors	
	Male	Female	Male	Female
1st Class	57	140	118	4
2nd Class	14	80	154	13
3rd Class	75	76	387	89
Crew	192	20	670	3

Children	Survivors		Non-Survivors	
	Male	Female	Male	Female
1st Class	5	1	0	0
2nd Class	11	13	0	0
3rd Class	13	14	35	17
Crew	0	0	0	0



Producing a Mosaic Plot

First load the “Titanic” data from the R data library.

```
> data(Titanic)
```

Next produce the mosaic.

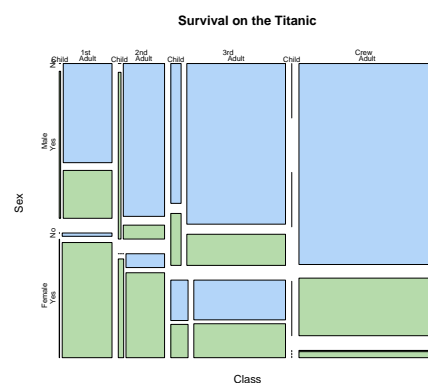
```
> mosaicplot(Titanic,
             main = "Survival on the Titanic",
             col = hcl(c(240, 120)),
             off = c(5, 5, 5, 5))
```

Note the use of `col=` to produce alternating coloured rectangles — green for survivors and blue for non-survivors. Also note that the `off=` argument is used to squeeze out a little of the space between the blocks.

Example: Survival on the Titanic



On Sunday, April 14th, 1912 at 11:40pm, the RMS Titanic struck an iceberg in the North Atlantic. Within two hours the ship had sunk. At best reckoning 705 survived the sinking, 1,523 did not.



Example: Sexual Discrimination at Berkeley

- In the 1980s, a court case brought against the University of California at Berkeley by women seeking admission to graduate programs there.
- The women claimed that the proportion of women admitted to Berkeley was much lower than that for men, and that this was the result of discrimination.

Gender	Admitted	Rejected	%Admitted
Male	1198	1493	44.5
Female	557	1278	30.4

- It is clear that a higher proportion of males is being admitted.

Producing The Plot

To produce the mosaic plot of the UCB data we must reverse order of the extents of the data array. We can either do this using `aperm`, or by using the `sort=` argument to `mosaicplot`.

```
> data(UCBAdmissions)
> mosaicplot(UCBAdmissions, sort = 3:1,
             col = hcl(c(120, 10)),
             main = "Student admissions at UC Berkeley")
```

(The order of the variables in the R data set is: Admit, Gender, Dept – the reverse of what `mosaicplot` expects.)

The University Case

The Dean of Letters and Science at Berkeley was a famous statistician (called Peter Bickel) and he was able to argue that the difference in admissions rates was not caused by sexual discrimination in the Berkeley admissions policy, but was caused by the fact that males and females generally sought admission to different departments.

The Dean broke the admissions data down by department and showed that within each program there was no admission discrimination against women. Indeed, there seemed to be some admissions bias in favour of women.

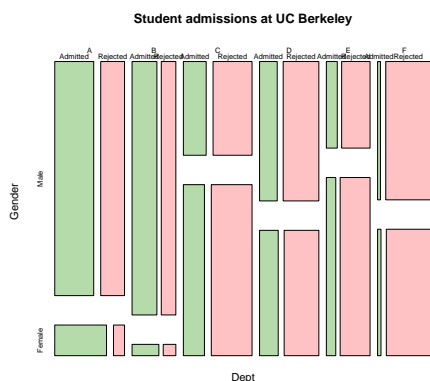
```
> UCBAdmissions
, , Dept = A
      Gender
Admit  Male Female
Admitted 512  89
Rejected 313  19
, , Dept = B
      Gender
Admit  Male Female
Admitted 353  17
Rejected 207   8
, , Dept = C
      Gender
Admit  Male Female
Admitted 120 202
Rejected 205 391
, , Dept = D
      Gender
Admit  Male Female
Admitted 138 131
Rejected 279 244
```

etc.

		Male	Female
Department A	Admitted	512	89
	Rejected	313	19
Department B	Admitted	353	17
	Rejected	207	8
Department C	Admitted	120	202
	Rejected	205	391
Department D	Admitted	138	131
	Rejected	279	244
Department E	Admitted	53	94
	Rejected	138	299
Department F	Admitted	22	24
	Rejected	351	317

Collapsing Contingency Tables

- The Berkeley Admissions contingency table has three dimensions.
 - Admit: Admitted, Rejected.
 - Gender: Male, Female.
 - Dept: A, B, C, D, E, F.
- To obtain the overall effect, ignoring Dept, we have to compute the table which would result by summing over departments.
- This is called *collapsing* the original table.



Collapsing Tables in R

The process of collapsing contingency tables is easy in R. The `apply` function can be used to collapse over the third subscript of the `UCBAdmissions` table.

```
> apply(UCBAdmissions, 1:2, sum)
      Gender
Admit  Male Female
Admitted 1198  557
Rejected 1493 1278
```

The effect of the `apply` statement is to sum the table over all the subscripts not given in the second argument.

Collapsing Tables in R

The order of the subscripts specified in an apply statement determines the order of the subscripts in the result. Reversing the order of subscripts in the previous apply statement produces the transpose of the original result.

```
> apply(UCBAdmissions, 2:1, sum)
      Admit
Gender  Admitted Rejected
Male    1198     1493
Female   557     1278
```