Displaying Data with Tables and Barplots

Alan T. Arnholt
Appalachian State University
arnholt@math.appstate.edu
Creating Tables from Data

The R function `table()` can be used to create a contingency table of the counts at each combination of factor levels. The function `table()` works on objects which can be interpreted as factors (including character strings), or a list (or data frame) of factors. The user should refer to the online help files by typing `?table` at the R prompt for specific questions.

A **factor** in R is a vector of data, usually taking a small number of distinct values. Many authors call this type of data **categorical**.

A **list** is a collection of S objects called components. Lists are used to unite related data that have different structures.

A **data frame** is a list of variables of the same length with unique row names. (All of the data sets in BSDA are stored as data frames)
The data frame **EPIDURAL** is stored on my web page. This example uses the `read.csv()` function to read a comma delimited data set from the internet. For R help on reading external data type `?read.table` or `?read.csv` at the R prompt. Information is also available with examples on the slides **Reading Data Into R**

```r
> site <- "http://www1.appstate.edu/~arnholta/PASWS/DATA/EPIDURAL.csv"
> EPIDURAL <- read.csv(file=url(site))
> attach(EPIDURAL)
> EPIDURAL[1:5,]  # Read the first five rows of data

<table>
<thead>
<tr>
<th>Doctor</th>
<th>kg</th>
<th>cm</th>
<th>Ease</th>
<th>Treatment</th>
<th>OC</th>
<th>Complications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. B</td>
<td>116</td>
<td>172</td>
<td>Difficult</td>
<td>Traditional Sitting</td>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>Dr. C</td>
<td>86</td>
<td>176</td>
<td>Easy</td>
<td>Hamstring Stretch</td>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>Dr. B</td>
<td>72</td>
<td>157</td>
<td>Difficult</td>
<td>Traditional Sitting</td>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>Dr. B</td>
<td>63</td>
<td>169</td>
<td>Easy</td>
<td>Hamstring Stretch</td>
<td>2</td>
<td>None</td>
</tr>
<tr>
<td>Dr. B</td>
<td>114</td>
<td>163</td>
<td>Impossible</td>
<td>Traditional Sitting</td>
<td>0</td>
<td>None</td>
</tr>
</tbody>
</table>
```
Comments about **EPIDURAL**

Note that the variables *Doctor, Ease, Treatment, and Complications* are factors (categorical). We will also consider *OC* as a factor although it is not technically stored in R as a factor.

```r
> is.factor(Treatment)
[1] TRUE
> is.factor(OC)
[1] FALSE
> is.numeric(OC)
[1] TRUE
```
Tables

Use the R function `table()` to create frequency tables of Treatment versus Ease and Doctor versus Ease similar to those in Tables 1 and 2.

### Table 1: Table of Treatment versus Ease

<table>
<thead>
<tr>
<th>treatment</th>
<th>Difficult</th>
<th>Easy</th>
<th>Impossible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamstring Stretch</td>
<td>8</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td>Traditional Sitting</td>
<td>12</td>
<td>32</td>
<td>6</td>
</tr>
</tbody>
</table>

### Table 2: Table of Doctor versus Ease

<table>
<thead>
<tr>
<th>Doctor</th>
<th>Difficult</th>
<th>Easy</th>
<th>Impossible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. A</td>
<td>3</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>Dr. B</td>
<td>10</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Dr. C</td>
<td>3</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Dr. D</td>
<td>4</td>
<td>13</td>
<td>3</td>
</tr>
</tbody>
</table>
Using `table()`

```r
table(Treatment, Ease)

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Difficult</th>
<th>Easy</th>
<th>Impossible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamstring Stretch</td>
<td>8</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td>Traditional Sitting</td>
<td>12</td>
<td>32</td>
<td>6</td>
</tr>
</tbody>
</table>

table(Doctor, Ease)

<table>
<thead>
<tr>
<th>Doctor</th>
<th>Difficult</th>
<th>Easy</th>
<th>Impossible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. A</td>
<td>3</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>Dr. B</td>
<td>10</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Dr. C</td>
<td>3</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Dr. D</td>
<td>4</td>
<td>13</td>
<td>3</td>
</tr>
</tbody>
</table>
```
More with `table()`

The order of the variables in a table can be easily switched by

1. switching the order in the `table()`
2. using the R function `t()` (transpose)

```r
> table(Ease,Doctor)

Doctor
Ease      Dr. A Dr. B Dr. C Dr. D
Difficult  3  10  3  4
Easy       19  7  18 13
Impossible 1  4  0  3
```

```r
> t(table(Doctor,Ease))

Doctor
Ease    Dr. A Dr. B Dr. C Dr. D
Difficult  3  10  3  4
Easy       19  7  18 13
Impossible 1  4  0  3
```
Using the function `barplot()`

The function `barplot(x)` creates a barplot of the information in `x` where `x` is a either a vector or matrix of values describing the bars which make up the plot. If `x` is a vector, the plot consists of a sequence of rectangular bars with heights given by the values in the vector `x`. If `x` is a matrix and `beside` is `FALSE` then each bar of the plot corresponds to a column of `x`, with the values in the column giving the heights of stacked “sub-bars” making up the bar. If `x` is a matrix and `beside` is `TRUE`, then the values in each column are juxtaposed rather than stacked.

The object `x` is often created with the function `table()`.

There are numerous arguments to the function `barplot()` and the user should consult the help file by typing `?barplot` at the R prompt for more information.
Graphical Representation of Table 2

Chunk 5 code

```r
> par(mfrow=c(2,2))
> junk <- table(Doctor,Ease)
> junk

<table>
<thead>
<tr>
<th>Ease</th>
<th>Doctor</th>
<th>Difficult</th>
<th>Easy</th>
<th>Impossible</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dr. A</td>
<td>3</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Dr. B</td>
<td>10</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Dr. C</td>
<td>3</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Dr. D</td>
<td>4</td>
<td>13</td>
<td>3</td>
</tr>
</tbody>
</table>

> barplot(junk)
> barplot(junk,beside=TRUE)
> barplot(t(junk))
> barplot(t(junk),beside=TRUE)
> par(mfrow=c(1,1))
```
Chunk 5 Graphs

- Creating Tables from Data
- Reading Data from the Internet
- Comments about EPIDURAL
- Tables
- Using `table()`
- More with `table()`
- Using the function `barplot()`
- Graphical Representation of Table 2
- Chunk 5 Graphs
- Problems and Solutions
- Chunk 6 Graph
- Moving the Legend
- Pie Charts
- Pie Chart
- Last Slide
Problems and Solutions

The previous graphs have at least two problems:
1. Problem: the order of \textit{Ease} is alphabetical. Solution: arrange according to difficulty.
2. Problem: graphs have no legends. Solution: use argument \texttt{legend=TRUE}.

The following code addresses both problems (Chunk 6 Code)

```r
> ease <- factor(Ease, levels=c("Easy", "Difficult", "Impossible"))
> barplot(table(ease, Doctor), beside=TRUE, legend=TRUE, col=c("green", "blue", "red"),
         main="Graphical Representation of Table 2")
```
Chunk 6 Graph

Graphical Representation of Table 2

Dr. A
Dr. B
Dr. C
Dr. D

Easy
Difficult
Impossible
Moving the Legend

The default placement of the legend is not the best. In the code that follows, the range of the y-axis is extended with the use of `ylim`.

```r
> stuff <- table(ease, Doctor)
> barplot(stuff, beside=T, legend=T,
+       ylim=c(0, max(stuff)+10),
+       col=c("green", "blue", "red"), ylab="Frequency",
+       xlab="Doctor", main="Barplot with Legend")
```

Barplot with Legend

Dr. A Dr. B Dr. C Dr. D

Easy

Difficult

Impossible
Pie Charts

Quoted from the R help file on `pie()`. Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data. Cleveland (1985), page 264: “Data that can be shown by pie charts always can be shown by a dot chart. This means that judgements of position along a common scale can be made instead of the less accurate angle judgements.” If you still think you want a pie chart, here is a small example.

```r
> table(ease)
                   ease
               Easy  Difficult Impossible
               57       20         8
> pie(table(ease))
```
Pie Chart

- Alan’s Notes
- Creating Tables from Data
- Reading Data from the Internet
- Comments about EPIDURAL
- Tables
- Using `table()`
- More with `table()`
- Using the function `barplot()`
- Graphical Representation of Table 2
- Chunk 5 Graphs
- Problems and Solutions
- Chunk 6 Graph
- Moving the Legend
- Pie Charts
- Pie Chart
- Last Slide
1. Script for Tables and Barplots
2. Do problems 1.19 - 1.29 for homework.
3. See me if You Need Help!

Click here to pause the presentation.