Summary statistics and graphics: Categorical Data

Patrick Breheny

September 1, 2016

The next two labs will be summary statistics and statistical graphics. Today's lab will focus on categorical data; next week's will address continuous data. Our data set for today will be the `titanic` data set, an interesting data set with 4 variables, all of which are categorical:

```
> titanic <- read.delim("http://myweb.uiowa.edu/pbreheny/data/titanic.txt")
> head(titanic)

  Class  Sex   Age Survived
1   3rd Male Child     Died
2   3rd Male Child     Died
3   3rd Male Child     Died
4   3rd Male Child     Died
5   3rd Male Child     Died
6   3rd Male Child     Died

> nrow(titanic)

[1] 2201

> ncol(titanic)

[1] 4

> dim(titanic)

[1] 2201    4
```

1 Tables

1.1 One-way tables

As we discussed in lecture, the basic summary statistic for categorical data is the *count*. For example, we might want to know the number of 1st/2nd/3rd class passengers and crew aboard the ship. This is most easily accomplished using the `table` function:

```
> table(titanic$Class)


 1st  2nd  3rd Crew
 325  285  706  885
```

An alternative way of expressing this information is the fraction of total passengers who fell into each of these categories. This can either be done directly by dividing by the number of passengers, or automated using the `prop.table` function:

```
> tab <- table(titanic$Class)
> tab/nrow(titanic)


      1st       2nd       3rd      Crew
0.1476602 0.1294866 0.3207633 0.4020900

> prop.table(tab)


      1st       2nd       3rd      Crew
0.1476602 0.1294866 0.3207633 0.4020900
```

Yet another way is as a percentage, or rate per 100 passengers:

```
> round(100*prop.table(tab), 1)


 1st  2nd  3rd Crew
14.8 12.9 32.1 40.2
```

1.2 Two- and three-way tables

The above approaches allow us to look at one variable at a time. If we want to look at multiple variables at the same time, we need to construct multi-way tables. This is easily accomplished by adding more variables to the `table` function:

```
> with(titanic, table(Class, Survived))

      Survived
Class  Died Survived
  1st   122      203
  2nd   167      118
  3rd   528      178
  Crew  673      212
```

Fractions are still interesting – perhaps even more so – in a multi-way table, but we have several ways of going about calculating them: (a) a fraction out of all passengers, (b) a fraction of the passengers in that row (Class, in the above example), and (c) a fraction of the passengers in that column (Survival, in the above example). All of these can be calculated with `prop.table`:

```
> tab <- with(titanic, table(Class, Survived))
> prop.table(tab)    ## Overall proportion

      Survived
Class        Died   Survived
  1st  0.05542935 0.09223080
  2nd  0.07587460 0.05361199
```

```
  3rd  0.23989096 0.08087233
  Crew 0.30577010 0.09631985

> prop.table(tab, 1) ## Row-wise proportion

      Survived
Class       Died  Survived
  1st  0.3753846 0.6246154
  2nd  0.5859649 0.4140351
  3rd  0.7478754 0.2521246
  Crew 0.7604520 0.2395480

> prop.table(tab, 2) ## Column-wise proportion

      Survived
Class        Died   Survived
  1st  0.08187919 0.28551336
  2nd  0.11208054 0.16596343
  3rd  0.35436242 0.25035162
  Crew 0.45167785 0.29817159
```

In other words, 24% of people on board were 3rd class passengers who died, 75% of the 3rd class passengers died, and 45% of the people who died were members of the crew.

The same logic can be extended to higher-way tables as well:

```
> tab <- with(titanic, table(Class, Survived, Sex))
> tab

, , Sex = Female

      Survived
Class  Died Survived
  1st     4      141
  2nd    13       93
  3rd   106       90
  Crew    3       20

, , Sex = Male

      Survived
Class  Died Survived
  1st   118       62
  2nd   154       25
  3rd   422       88
  Crew  670      192

> prop.table(tab, c(1,3))

, , Sex = Female

      Survived
Class        Died   Survived
  1st  0.02758621 0.97241379
```

```
  2nd  0.12264151 0.87735849
  3rd  0.54081633 0.45918367
  Crew 0.13043478 0.86956522

, , Sex = Male

      Survived
Class        Died   Survived
  1st  0.65555556 0.34444444
  2nd  0.86033520 0.13966480
  3rd  0.82745098 0.17254902
  Crew 0.77726218 0.22273782
```

The `c(1,3)` syntax tells `R` that we want to calculate proportions over all levels of `Class` and `Sex`. The output tells us, for instance, that 97% of female 1st class passengers survived (141/145).

1.3 Questions

- How many 2nd class male passengers died?
- How many children were in second class?
- What fraction of 3rd class children survived?
- What fraction of adults who survived were crew members?

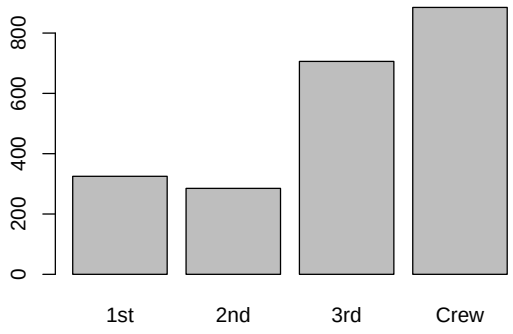
2 Graphs

As the above example(s) probably make clear, multi-way tables are informative but quickly get very cumbersome. Graphs are often superior to tables at quickly conveying information.

2.1 Basic bar plots

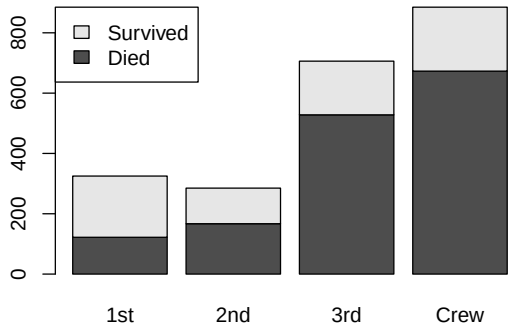
The basic plot for categorical data is the bar plot, which is pretty self-explanatory:

```
> tab <- table(titanic$Class)
> barplot(tab)
```



An extension of the bar chart that allows us to plot two variables at once is the *stacked bar plot*:

```
> tab <- with(titanic, table(Survived, Class))
> barplot(tab, legend=TRUE, args.legend=list(x="topleft"))
```



With this plot, we can see that crew and 3rd class passengers were much more plentiful on the ship than 1st and 2nd class passengers, and also that a higher percent of 1st class passengers survived than the others.

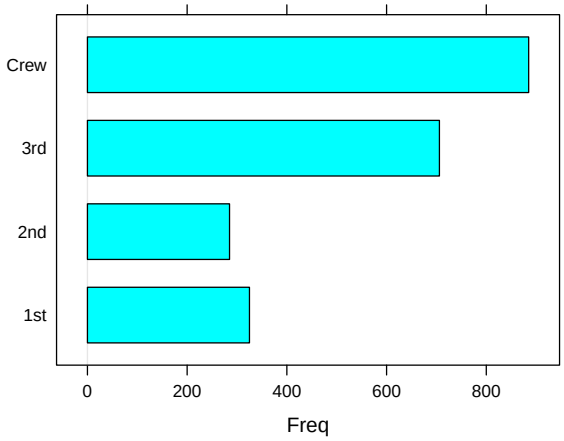
2.2 R Packages

It is essential to know the basic R plotting functions; however, in many situations it is difficult and/or tedious to make more complicated plots using standard R graphics. To facilitate the construction of these plots, several individuals have developed *packages* to assist in the making of these plots. One of the most common and widely used is the **lattice** package. This package is actually installed by default when you install R, but still needs to be loaded with:

```
> require(lattice) ## or library(lattice)
```

The lattice equivalent to **barplot** is **barchart**; simple plots are very similar for both functions:

```
> tab <- table(titanic$Class)
> barchart(tab)
```

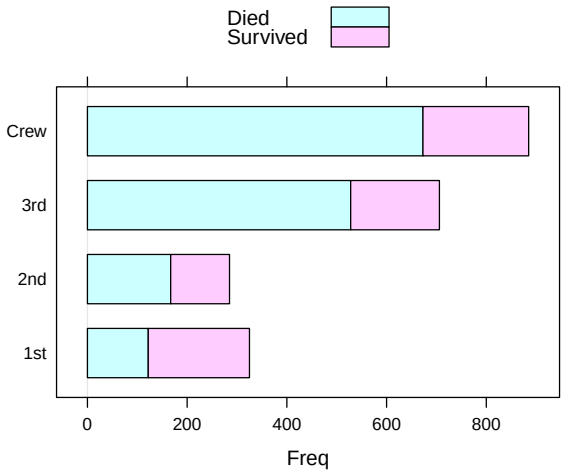


Other than differences in the defaults (which we can change with `horizontal=FALSE, col="gray"`), this is the same plot that we got from `barplot`.

2.3 Grouping and conditioning

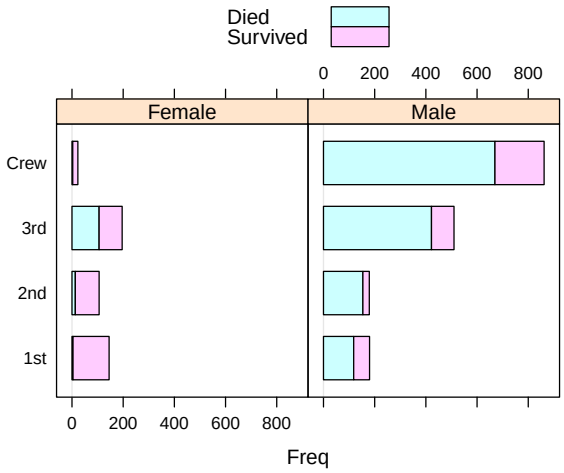
So why bother with `lattice`? The big advantage of lattice is that it allows you to easily create plots that take advantage of grouping and conditioning. *Grouping* is simply the use of an aesthetic property such as color or shape to represent a variable. We have already seen an example of this with the stacked barplot, but this is a little nicer in `lattice` since the legend doesn't get in the way of the plot:

```
> tab <- with(titanic, table(Class, Survived))
> barchart(tab, auto.key=TRUE)
```



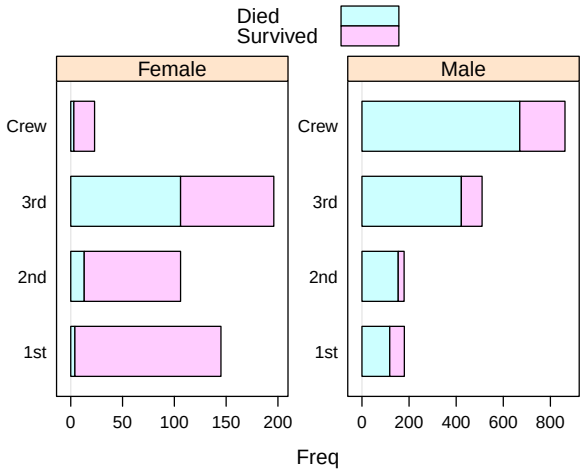
A more significant advantage is *conditioning*, which creates multiple small plots (panels) of different subsets of the data, with the subsets determined by the conditioning variables. For example, the following creates separate panels for males and females:

```
> tab <- with(titanic, table(Class, Sex, Survived))
> barchart(tab, auto.key=TRUE)
```



It's kind of hard to see what's going on in the female plot above because, by default, the scales in each panel are constrained to be equal, and there were a lot more men on board than women. We can allow the scales to differ in each panel with `scales="free"`:

```
> barchart(tab, auto.key=TRUE, scales="free")
```



2.4 Questions

Add `Age` to the above plot and then consider the following questions:

- The general policy in evacuation was "women and children first". How well did this policy hold up across the various classes?
- Overall, there was a striking class bias in terms of survival (62% of 1st class passengers survived compared with only 25% of 3rd class passengers). Does this trend hold up once you start making comparisons in smaller groups? If not, what explains the discrepancy?
- Overall, a (slightly) higher percentage of crew died than 3rd class passengers. Does this trend hold up once you start making comparisons in smaller groups? If not, what explains the discrepancy?

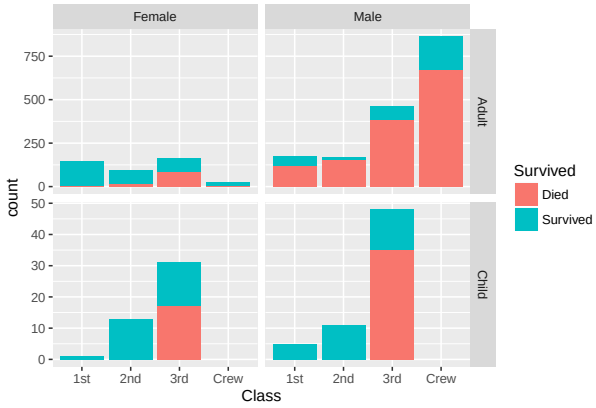
2.5 Installing new packages

Another popular graphics package is `ggplot2`. Unlike `lattice`, `ggplot2` must be installed:

```
> install.packages("ggplot2")
```

Once installed, it can be loaded using `require` or `library`. Note that you need to load packages like `lattice` and `ggplot2` every time you open R, but you only need to install a package once. Basic plots in `ggplot2` can be constructed using the `qplot` (for 'quick plot') function. To illustrate using the `titanic` data,

```
> require(ggplot2)
> qplot(Class, data=titanic, fill=Survived) + facet_grid(Age~Sex, scales="free")
```



Here, `fill=Survived` specifies that the color used to fill in the bars should depend on survival status. This sets up the basic plot, while `facet_grid` controls the conditioning. `Age Sex` lists the conditioning variables as well as whether they should be oriented in the vertical or horizontal direction. `scales="free"` means the same thing in `ggplot2` as it did in `lattice`, although note that the scales are not completely free, in that all panels in a row must share the same vertical scale. We can obtain another interesting plot by adding `position="fill"` to `qplot()`.

Whether you use `lattice`, `ggplot2`, or basic `R` graphics is up to you (personally, I use all three, depending on the task at hand). What is important is to appreciate how information-rich a plot like the above is and how much information it communicates. As the saying goes, the above picture is worth a thousand words in terms of all it communicates about the relationships between the four variables in this data set.