Converting Table Data to Data for Modeling

Joseph M. Hilbe
Arizona State University; Jet Propulsion Laboratory
hilbe@asu.edu : 27 June, 2014

Analysts many times have access to data presented in tables. Table data is also displayed in journal articles as well as in more informal settings. In any case, analysts can usually convert such data into a form that is able to be analyzed and even modeled using statistical methods. This short monograph explains how this may be accomplished.

Consider a simple table of data, presented as,

		X		
		0	1	
	0	5	7	12
Y				
	1	8	15	23
Total		13	22	35

I have used variable names Y and X to keep the discussion simple. Y may take on a meaning such as "1= passed the statistics final exam; 0=failed" and X as "1=Studied"; 0= "Partied instead" The values in each cell are the number students common to each {x,y} pair.

Y==0 and X==0: 5
Y==0 and X==1: 7
Y==1 and X==0: 8
Y==1 and X==1: 15

If we use the statistics exam example, we may interpret the table information as:

Of the 13 students who partied instead of studied, 5 failed the exam and 9 still passed.
Of the 22 students who studied instead of partied, 15 passed the exam and 7 failed.

The summary values for each level of X at the bottom of the table are called *marginals.* The same is the case for the right hand side summary levels of Y.

Each of the variables, Y and X, are binary. If Y is considered as the response; i.e., the variable to be modeled, and X as the explanatory predictor used to explain Y, the data may be modeled using a *logistic regression*. In order to do this though, the data must be formatted in a manner suitable for entering into regression software.

When converting table data to data to be modeled, it is mandatory that each cell within the marginals have a separate line. For the above table, there are 4 cells, therefore we must create 4 lines providing coverage for each alternative.

It is standard to place the response variable as the left-most column. Divide the 0s and 1s by the number of levels in the X variable. Here the X variable has two levels, 0 and 1. The following schemata provides an example of how the data can be structured, covering all alternatives

Y	X	cnt
0	0	5
0	1	7
1	0	8
1	1	15

The values of the *cnt* variable are entered into the model as a frequency weight into a logistic regression model. Using R with the **glm** function we have,

```
> y <- c(0,0,1,1)
> x <- c(0,1,0,1)
> cnt <- c(5,7,8,15)
> summary(mod1 <- glm(y~ x, weights=cnt, family=binomial))

              .     .      .
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.4700     0.5701   0.824    0.410
x             0.2921     0.7311   0.400    0.689

              .     .       .

# ODDS RATIOS
> exp(coef(mod1))
(Intercept)           x
   1.600000    1.339286
```

Note that odds ratios must be requested after the model has been estimated. I named the model *mod1*. The model named must be used when requesting subsequent model statistics.

The odds ratios may be calculated directly from the values in the table. The schemata are:

Odds of X==1: y(1)/y(0)
Odds of X==0: y(1)/y(0)

The odds ratio, OR, is the ratio of the two odds: (X==1)/(X==0). Here we have

Odds of X==1: y(1)/y(0) = 15/7 = 2.142857
Odds of X==0: y(1)/y(0) = 8/5 = 1.6 intercept
Odds ratio: 2.142857/1.6 = 1.339286 x

Using Stata the data may be typed into the editor. Then, once the default variable names, *var1*, *var2*, and *var3* have been changed to *y*, *x* and *cnt*, the analyst may model the data using the **glm**, **logit,** or **logistic** commands. Here I use the glm command with the *nolog* and *nohead* options to that the iteration log and header statistics are not displayed on the screen.

```
. l

     +-------------+
     | y   x   cnt |
     |-------------|
  1. | 0   0     5 |
  2. | 0   1     7 |
  3. | 1   0     8 |
  4. | 1   1    15 |
     +-------------+

. glm y x [fw=cnt], fam(bin) nolog nohead

------------------------------------------------------------------------------
             |                 OIM
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   .2921364   .7311114     0.40   0.689    -1.140815    1.725088
       _cons |   .4700036   .5700877     0.82   0.410    -.6473478    1.587355
------------------------------------------------------------------------------
```

ODDS RATIO

```
. glm y x [fw=cnt], fam(bin) nolog nohead eform

------------------------------------------------------------------------------
             |                 OIM
           y | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.339286    .979167     0.40   0.689     .3195583    5.613017
       _cons |        1.6   .9121403     0.82   0.410     .5234322    4.890796
------------------------------------------------------------------------------
```

The data can also be formated so that there is no need for a *cnt* variable. Convert the data to what is termed observation data:

```
Y       X
0       0
0       0
0       0
0       0
0       0
0       1
0       1
  . . .
1       1
1       1
1       1
1       1
```

There are 5+7+8+15 = 35 observations or lines. This is the grand total value from the table information above. Using Stata the expand command "expands" the data to observation format based on the values of *cnt.* When the data is modeled without the *cnt* frequency weight, it appears as,

```
. expand cnt
(31 observations created)

. glm y x, fam(bin) nolog nohead

------------------------------------------------------------------------------
             |                 OIM
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   .2921364   .7311114     0.40   0.689    -1.140815    1.725088
       _cons |   .4700036   .5700877     0.82   0.410    -.6473478    1.587355
------------------------------------------------------------------------------

. glm, eform nohead

------------------------------------------------------------------------------
             |                 OIM
           y | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.339286    .979167     0.40   0.689     .3195583    5.613017
       _cons |        1.6   .9121403     0.82   0.410     .5234322    4.890796
------------------------------------------------------------------------------
```

=======

I shall complicate the table a bit more, making X into a three level variable.

```
-------------------------------------------------------------
1=Died;    |                 Type of admission
0=Alive    |  Elective Admit     Urgent Admit  Emergency Admit
-----------+-------------------------------------------------
         0 |              770              161               51|     982
         1 |              364              104               45|     513
-----------------------------------------------------------------|
                         1134              265               96|    1495
```

Using the lowest, or *Elective*, level of *type* as the reference, the ratios of each of the higher levels of type may be used as the numerator in calculating the odds ratios. There are 6 cells total.

```
> (104/161)/(364/770)    #  Level 2
[1] 1.36646

> (45/51)/(364/770)      #  Level 3
[1] 1.866516

> (364/770)              # Intercept
[1] 0.4727273
```

Using R the data may be set up as:

```
> died <- c(0,0,0,1,1,1)
> type <- c(1,2,3,1,2,3)
> cnt  <- c(770, 161, 51, 364, 104, 45)

> summary(mod2 <- glm(died ~ factor(type), family=binomial, weights=cnt))

                .     .     .
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.74924    0.06361 -11.779  < 2e-16 ***
factor(type)2  0.31222    0.14097   2.215  0.02677 *
factor(type)3  0.62407    0.21419   2.914  0.00357 **
---

                .     .     .

> exp(coef(mod2))
  (Intercept) factor(type)2 factor(type)3
    0.4727273     1.3664596     1.8665158
```

It is clear that the odds ratios above match the by-hand method used with the actual table data.

The table data may be converted to Stata as well. The data as listed from the table is entered into the Stata editor as:

```
. l

     +-------------------+
     | died   type   cnt |
     |-------------------|
  1. |    0      1   770 |
  2. |    0      2   161 |
  3. |    0      3    51 |
  4. |    1      1   364 |
  5. |    1      2   104 |
     |-------------------|
  6. |    1      3    45 |
     +-------------------+
```

Died serves as the reference, and *type* is entered as above with alternative 1-2-3's. The data appears as

```
. glm died i.type [fw=cnt], fam(bin) nolog nohead eform

------------------------------------------------------------------------------
             |                 OIM
        died | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        type |
          2  |    1.36646   .1926297     2.21   0.027      1.03658    1.801319
          3  |   1.866516   .3997832     2.91   0.004     1.226635    2.840193
             |
       _cons |   .4727273   .0300691   -11.78   0.000     .4173185    .5354929
------------------------------------------------------------------------------
```

This is the same model as before.

Again, it is possible to expand the data so that *cnt* does not have to be used. The odds ratios are identical.

```
. expand cnt
(1489 observations created)

. glm died i.type, fam(bin) nolog nohead eform

------------------------------------------------------------------------------
             |                 OIM
        died | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        type |
          2  |    1.36646   .1926297     2.21   0.027      1.03658    1.801319
          3  |   1.866516   .3997832     2.91   0.004     1.226635    2.840193
             |
       _cons |   .4727273   .0300691   -11.78   0.000     .4173185    .5354929
------------------------------------------------------------------------------
```

This data is a replica of the **medpar** data. I can load it and determine if the models above are consistent with it,

```
.  use medpar, clear

. glm died i.type, fam(bin) nolog nohead eform

------------------------------------------------------------------------------
             |                 OIM
        died | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        type |
Urgent Ad..  |    1.36646   .1926297     2.21   0.027      1.03658    1.801319
Emergency..  |   1.866516   .3997832     2.91   0.004     1.226635    2.840193
             |
       _cons |   .4727273   .0300691   -11.78   0.000     .4173185    .5354929
------------------------------------------------------------------------------
```

Again, the results identical.

Higher level data sets may be created from more complex tables. All an analyst must remember is to set aside levels for each combination of cells. This method can be used in evaluation of model goodness it fit, predicted probablity, and other feature of count and binomial modeling.