Notes for laboratory session 7

Logistic Regression II: Model checking

Consider the contraceptive use data set:

```
. list

          age     educat       more       cuse          N
 1.       <25        Low         No          0         10
 2.       <25        Low         No          1          4
 3.       <25        Low        Yes          0         53
 4.       <25        Low        Yes          1          6
 5.       <25       High         No          0         50
 6.       <25       High         No          1         10
 7.       <25       High        Yes          0        212
 8.       <25       High        Yes          1         52
 9.     25-29        Low         No          0         19
10.     25-29        Low         No          1         10
11.     25-29        Low        Yes          0         60
12.     25-29        Low        Yes          1         14
13.     25-29       High         No          0         65
14.     25-29       High         No          1         27
15.     25-29       High        Yes          0        155
16.     25-29       High        Yes          1         54
17.     30-39        Low         No          0         77
18.     30-39        Low         No          1         80
19.     30-39        Low        Yes          0        112
20.     30-39        Low        Yes          1         33
21.     30-39       High         No          0         68
22.     30-39       High         No          1         78
23.     30-39       High        Yes          0        118
24.     30-39       High        Yes          1         46
25.     40-49        Low         No          0         46
26.     40-49        Low         No          1         48
27.     40-49        Low        Yes          0         35
28.     40-49        Low        Yes          1          6
29.     40-49       High         No          0         12
30.     40-49       High         No          1         31
31.     40-49       High        Yes          0          8
32.     40-49       High        Yes          1          8
```

Measures of goodness of fit

Goodness of fit tests are, by definition, those that compare the observed to the fitted values. In logistic regression there are two such statistics: The **Pearson chi-square** and the **deviance**.

In the contraceptive data example, if age is not used as a continuous variable, there are 8 covariate categories (=2×4) in each category of contraceptive use. Some data manipulation (see also **Appendix**) is in order:

```
. reshape wide N, i(age more educat) j(cuse)
(note:  j = 0 1)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                       32   ->      16
Number of variables                   6   ->       6
j variable (2 values)              cuse   ->   (dropped)
xij variables:
                                       N   ->   N0 N1
-----------------------------------------------------------------------------
```

```
. sort age more educat
. by age more: gen n1=sum(N1)
. by age more: gen n0=sum(N0)
. by age more: drop if _n<_N
. drop educat N0 N1
. rename n1 N1
. rename n0 N0
. generate tot=N0+N1
. label var tot "Total observations (n_i)"

. list
          age       more    contage         N1         N0        tot
  1.      <25         No         20         14         60         74
  2.      <25        Yes         20         58        265        323
  3.    25-29         No       27.5         37         84        121
  4.    25-29        Yes       27.5         68        215        283
  5.    30-39         No         35        158        145        303
  6.    30-39        Yes         35         79        230        309
  7.    40-49         No         45         79         58        137
  8.    40-49        Yes         45         14         43         57
```

a. What distribution does variable `N1` have and what is the variable's meaning (i.e., what does it measure)?

Consider the alternative analysis of contraceptive use by age and desire for more children:

```
. xi: blogit N1 tot i.age i.more
i.age                 Iage_1-4     (naturally coded; Iage_1 omitted)
i.more                Imore_0-1    (naturally coded; Imore_0 omitted)

Logit estimates                                   Number of obs   =       1607
                                                  LR chi2(4)      =     128.88
                                                  Prob > chi2     =     0.0000
Log likelihood = -937.40449                       Pseudo R2       =     0.0643
------------------------------------------------------------------------------
_outcome |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  Iage_2 |   .3678306   .1753673      2.097   0.036        .024117    .7115443
  Iage_3 |   .8077888   .1597533      5.056   0.000        .494678      1.1209
  Iage_4 |   1.022618   .2039337      5.014   0.000       .6229158    1.422321
 Imore_1 |   -.824092   .1171128     -7.037   0.000      -1.053629   -.5945552
   _cons |  -.8698414   .1571298     -5.536   0.000       -1.17781   -.5618727
------------------------------------------------------------------------------
```

`N1` is the number of women using contraceptives in each of the eight `age×more` categories and `tot` is the total number of women. `blogit` performs the logistic regression on this binomial sample (i.e., the sample of `N1` out of `tot` women using contraception). Compare these estimates with the output in the previous lecture.

This is the same as when we carried out the `logit` command on the total sample. We can now derive the deviance manually by following the formula given above. To derive $\hat{\mu}_i$ the expected number of women using contraception in each of the sixteen `age×more` categories we proceed as follows (note that `blogit` produces estimates of *counts* not probabilities):

```
. predict yhat
(option n assumed; predicted no. of cases)
```

Note that we are predicting **counts** with the `predict` command after the `blogit`. Then the deviance is generated as follows:

```
. gen di = 2*(N1*log(N1/yhat) + (tot-N1)*log((tot-N1)/(tot-yhat)) )

. gen D=sum(di)

. display "Deviance = " D[_N]
Deviance = 16.788813
. display " p = " chi2tail(3, D[_N])
p = .00078105
```

So the p value is p=0.0008, which means that the additive two-factor model does not fit the data adequately. This result is consistent to the analyses shown in the previous lecture.
Note that the square root of `di` is the *deviance* residual.

b. Why does the deviance statistic above as well as the Pearson statistic have a chi-square distribution with 3 degrees of freedom?

Pearson chi-square

The Pearson chi-square statistic is derived similarly:

```
. gen r=(N1-yhat)/sqrt(yhat*(1-yhat/tot))

. gen X2=sum(r^2)

. display "Pearson X2=" X2[_N]
Pearson X2=16.283419

. display "p = " chi2tail(3, X2[_N])
p =  .00099191
```

The Pearson chi-square statistic is close to the deviance statistic and is associated with a highly significant p value, which is further evidence for the inadequacy of the two-factor additive model. Notice that `r` is called the *Pearson* residual.

The Hosmer and Lemeshow statistic

When individual data are involved, there is a definite need for a goodness of fit statistic. The Hosmer-Lemeshow (HL) statistic fills this need. **Note that the asymptotic distribution of the deviance and Pearson statistic is *not* chi-square if we have individual-subject data (or when the number of categories *k* increases as *n* increases)!**

We return to the original data set.

We need to do this, because STATA implements the HR statistic as part of the `lfit` command that follows the `logistic` command and the latter can only handle individual-level data.

```
. quietly xi: logit cuse i.more contage [freq=N]

. lfit, group(6) table

Logistic model for cuse, goodness-of-fit test
(Table collapsed on quantiles of estimated probabilities)

_Group      _Prob     _Obs_1     _Exp_1     _Obs_0     _Exp_0     _Total
     1    0.1632         58       52.7        265      270.3        323
     2    0.2135         68       60.4        215      222.6        283
     3    0.2743         79       84.8        230      224.2        309
     4    0.3828         65       90.2        187      161.8        252
     5    0.4633        158      140.4        145      162.6        303
     6    0.5730         79       78.5         58       58.5        137

       number of observations =      1607
             number of groups =         6
      Hosmer-Lemeshow chi2(4) =     17.48
                  Prob > chi2 =    0.0016
```

The p value of the Hosmer-Lemeshow chi-square is 17.48, which compared to a chi-square with 4 degrees of freedom results in a p value of 0.0016. This is evidence that the two-factor covariance model with no interaction does not fit the data adequately. Note that we chose g=6 as the total number of groups was 8.

The HR statistic is computed as follows:

Step 1. Carry out the logistic regression and generate the predicted probabilities

Step 2. Sort the predicted probabilities

Step 3. Group observations based on the predicted probabilities. Resolve (STATA) ties by assigning all observations with the same predicted value in the same group.

Step 4. Calculate a Pearson chi-square statistic based on the $2\times g$ contingency table that results from step 3 and the response variable.

Let's compute the statistic manually (note that the size of the groups would be close to 1607/6=268 subjects):

Hand calculation of the HR statistic

```
. quietly xi: logit cuse i.more contage [freq=N]

. predict phat
(option p assumed; Pr(cuse))

. sort phat
. list age more phat N
```

	age	more	phat	N	
1.	**<25**	**Yes**	**.1632108**	**212**	
2.	**<25**	**Yes**	**.1632108**	**52**	
3.	**<25**	**Yes**	**.1632108**	**6**	= 323 subjects group 1
4.	**<25**	**Yes**	**.1632108**	**53**	
5.	25-29	Yes	.2135374	155	
6.	25-29	Yes	.2135374	14	
7.	25-29	Yes	.2135374	54	= 283 subjects group 2
8.	25-29	Yes	.2135374	60	
9.	**30-39**	**Yes**	**.2742955**	**112**	
10.	**30-39**	**Yes**	**.2742955**	**118**	
11.	**30-39**	**Yes**	**.2742955**	**33**	= 309 subjects group 3
12.	**30-39**	**Yes**	**.2742955**	**46**	
13.	<25	No	.3081821	50	
14.	<25	No	.3081821	10	
15.	<25	No	.3081821	10	
16.	<25	No	.3081821	4	
17.	**40-49**	**Yes**	**.3700797**	**8**	
18.	**40-49**	**Yes**	**.3700797**	**35**	
19.	**40-49**	**Yes**	**.3700797**	**8**	= 252 subjects group 4
20.	**40-49**	**Yes**	**.3700797**	**6**	
21.	25-29	No	.3827633	27	
22.	25-29	No	.3827633	19	
23.	25-29	No	.3827633	65	
24.	25-29	No	.3827633	10	
25.	**30-39**	**No**	**.4633063**	**77**	
26.	**30-39**	**No**	**.4633063**	**78**	
27.	**30-39**	**No**	**.4633063**	**68**	= 303 subjects group 5
28.	**30-39**	**No**	**.4633063**	**80**	
29.	40-49	No	.5729807	46	
30.	40-49	No	.5729807	31	
31.	40-49	No	.5729807	48	= 145 subjects group 6
32.	40-49	No	.5729807	12	

The Hosmer-Lemeshow statistic is calculated as a Pearson chi-square statistic based on the 2×6 table. Its value is 17.48. The associated p value based on a chi-square distribution with four degrees of freedom is 0.0016.

```
. di "p = " chi2tail(4, 17.48)
p = .00155892
```

Model checking

Recall the best model as identified in the previous lecture:

```
. gen contage2=contage*contage

.  xi: logit cuse contage contage2 i.more i.more*contage [freq=N], nolog
i.more                Imore_0-1    (naturally coded; Imore_0 omitted)
i.more*contage        ImXcon_#     (coded as above)
Note: Imore_1 dropped due to collinearity.
Note: contage dropped due to collinearity.

Logit estimates                                   Number of obs   =       1607
                                                  LR chi2(4)      =     143.33
                                                  Prob > chi2     =     0.0000
Log likelihood = -930.18024                       Pseudo R2       =     0.0715

------------------------------------------------------------------------------
    cuse |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 contage |   .2331551   .0651087      3.581   0.000       .1055445    .3607658
contage2 |  -.0024113   .0009398     -2.566   0.010      -.0042532   -.0005693
 Imore_1 |   1.292637   .5810191      2.225   0.026       .1538601    2.431413
ImXcon_1 |  -.0659373   .0176673     -3.732   0.000      -.1005645   -.0313101
   _cons |  -5.216035   1.123734     -4.642   0.000      -7.418513   -3.013557
```

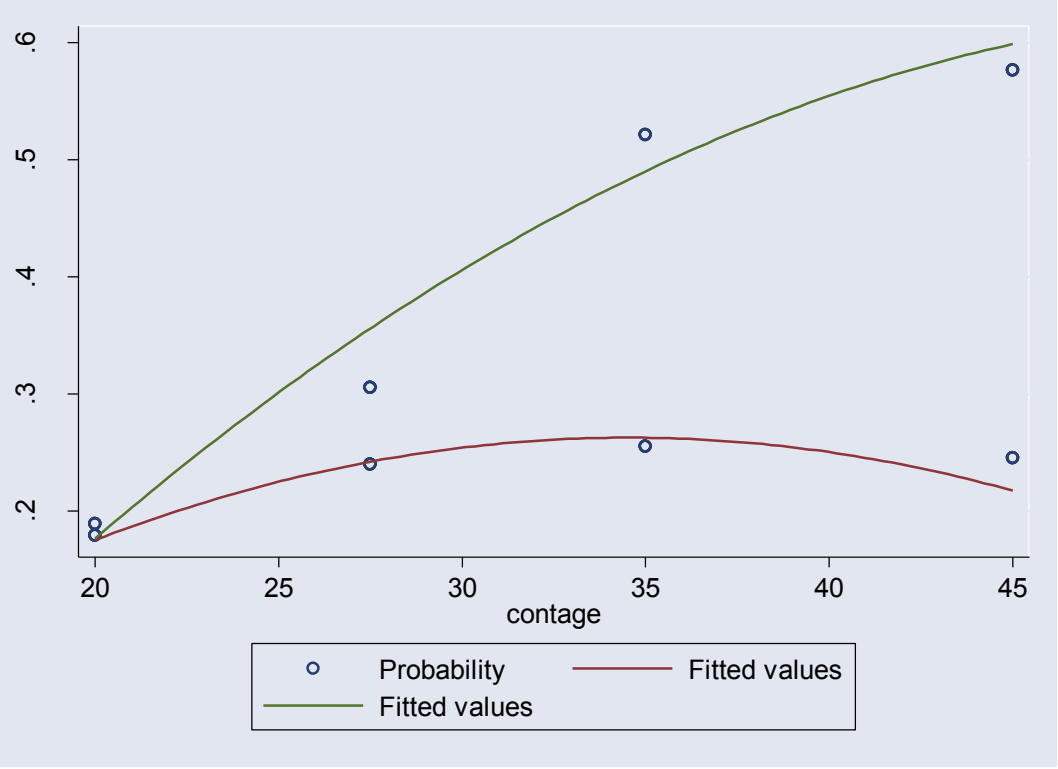
Model checking, is based on residuals and influence measures as was the case in linear regression.

Residuals and influence measures

In the example, we produce the fitted values for the probability of contraceptive use as follows:

```
. sort more

. quietly xi: logit cuse i.more i.age i.more*i.age [freq=N]

. predict prob
(option p assumed; Pr(cuse))

. label var prob "Probability"

. quietly xi: logit cuse i.more contage contage2 i.more*contage[freq=N]

. predict phat

. gen phat1=phat if more==1
(16 missing values generated)

. gen phat0=phat if more==0
(16 missing values generated)

sc prob contage,c(.) || qfit phat1 contage|| qfit phat0 contage xlab() ylab()
```

c. The predicted probabilities `prob` from the model that includes `more` and `age` as well as `more×age` interaction are equal to the observed probabilities of the data. Why?



Model checking through residuals and influence measures

```
. quietly  xi: logit cuse  contage2 i.more*contage [freq=N],nolog
. predict p, resid
. predict s, rstand
. predict d, deviance
. predict h,hat
. predict D, dbeta
. predict DX2, dx2
. predict Dd, dd
. predict n, n
```

Notice that `n` is the number of the covariate pattern. These are

Covariate pattern (`n`)	Age (`age`)	Desire for more children (`more`)
1	<25	Yes
2	<25	No
3	25-29	Yes
4	25-29	No
5	30-39	Yes
6	30-39	No
7	40-49	Yes
8	40-49	No

```
. table contage, contents(mean prob  mean phat) by(more)

----------+----------------------
Desires   |
more      |
children? |
and       |
contage   | mean(prob)  mean(phat)
----------+----------------------
No        |
       20 |   .1891892    .1798393
     27.5 |   .3057851     .348013
       35 |   .5214521    .4976501
       45 |   .5766423     .597039
----------+----------------------
Yes       |
       20 |   .1795666    .1760204
     27.5 |   .2402827     .240777
       35 |   .2556634    .2641383
       45 |    .245614     .217312
----------+----------------------
```

Residuals

```
. sum p s d

Variable |     Obs        Mean    Std. Dev.       Min        Max
---------+--------------------------------------------------------
       p |      32   -.0119643    .5481008  -.9751577   .8286497
       s |      32   -.0045499    .9110746  -1.243111   1.458263
       d |      32   -.0143877    .5493285   -.985256   .8287445
```

In situations where the number of subjects per category is fairly large (as is the case here), the central-limit theorem provides a criterion for deciding how large a residual has to be before is considered problematic.

Note!!! Disregard the `Mean` and `Std. Dev.` column in the above output. We used the `sum` command in order to simply get the minimum and maximum values of the residuals.

d. A residual larger than 2.0 should be inspected more carefully. Why?

We see that no residuals are too large as no residual reaches that threshold. However, the 6th and 8th category (`more==0` and `contage==35,45`) are associated with a large Cook's distances. Here a criterion similar to the linear-regression situation of a Cook's distance larger than 1.0 being considered large is adopted.

```
. preserve
. sort n

. qui by n:keep if _n==1

. li  n age more  D  DX2 Dd h

             n        age       more          D        DX2         Dd          h
  1.         1        <25        Yes   .8055608   .1648559   .1639945   .8301184
  2.         2        <25         No   .1768137   .1126808    .111197    .610767
  3.         3      25-29        Yes   .0004628   .0006483   .0006486   .4164959
  4.         4      25-29         No   .9659234   1.545324   1.577495   .3846388
  5.         5      30-39        Yes    .563625   .3171211    .319338   .6399403
  6.         6      30-39         No   4.459163   2.126531   2.127018   .6770984
  7.         7      40-49        Yes   1.001881   .6698966   .6502974   .5992909
  8.         8      40-49         No    7.95146   1.496048   1.488333   .8416463
. restore
```

Distance and influence measures

The leverage can be considered in a similar manner as in the linear-regression case. The sum of the diagonal elements of the hat matrix is (p+1) so any leverage twice the average value (i.e., a leverage larger than $2(p+1)/k$) should be considered further (Pregibon, 1981). The average value here is 5/8=0.625, so there are no overly influential categories.

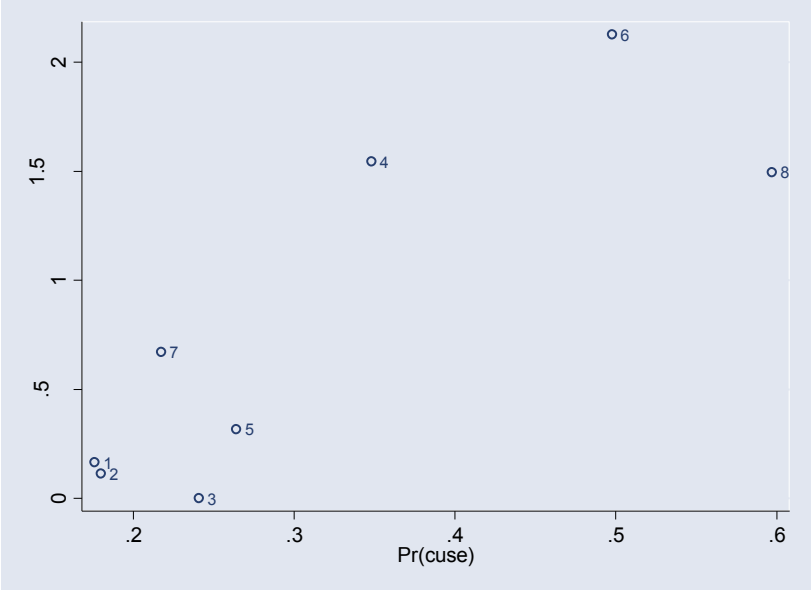
Hosmer and Lemeshow also recommend inspecting graphically the model fit by plotting ΔX^2 and ΔD as well as D against the estimated probility $\hat{\pi}_j = P(Y=1 | X=j)$ for covariate pattern j. Poorly fit points will be located at the top left and top right corner of the graph, and in general do not

conform to the pattern defined by the majority of the points. In the following plots, we identify the points by the covariate pattern `n`.

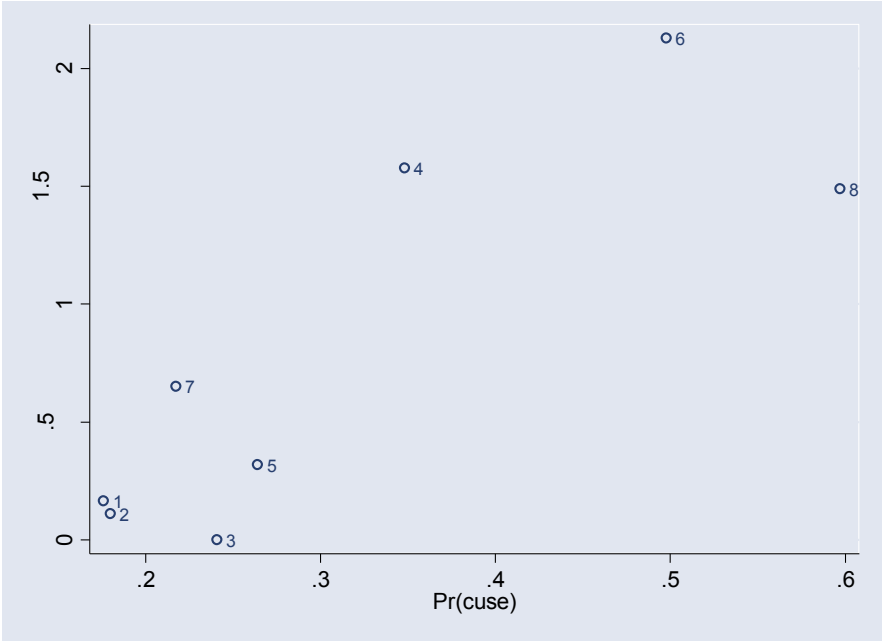
Distance and influence measures

The crude threshold for ΔX^2 and ΔD is 4.0, the approximation of the 95th percentile of the chi-square distribution with one degree of freedom (recall that $\chi^2_{1;0.95}=3.84$). By extension of the criterion of the Cook's distance, the threshold of D is 1.0.

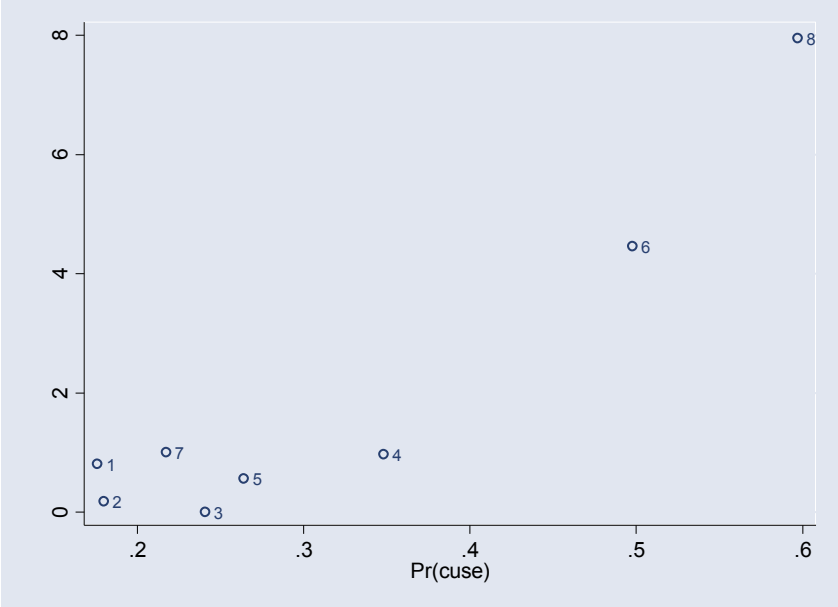
```
. sc DX2 phat, xlab() ylab() mlab(n)
```



```
. sc Dd phat, xlab() ylab() mlab(n)
```




```
. sc D phat, xlab() ylab() mlab(n)
```



We see that no point in the graphs above satisfies any criterion for an unusually poorly fit or influential point. The model fits the data well. At the most, we would like to explore category `n==6` and `n==8` (women ages 30-39 and 40-49 wanting no more children) a bit further.

Appendix
(Data manipulation - Instead of using the "`reshape`" command)

```
. use cuse.stata6.dta,clear

. sort  age  more cuse

. gen n1=N if cuse==1
(16 missing values generated)

. by  age more :li
```

```
-------------------------------------------------------------------------------
-> age = <25, more = No

          age     educat       more          N       cuse    contage         n1
  1.       <25        Low         No         10         No         20          .
  2.       <25       High         No         50         No         20          .
  3.       <25       High         No         10        Yes         20         10
  4.       <25        Low         No          4        Yes         20          4

-------------------------------------------------------------------------------
-> age = <25, more = Yes

          age     educat       more          N       cuse    contage         n1
  5.       <25        Low        Yes         53         No         20          .
  6.       <25       High        Yes        212         No         20          .
  7.       <25       High        Yes         52        Yes         20         52
  8.       <25        Low        Yes          6        Yes         20          6
```

```
. by  age more cuse:gen N1=sum(n1)

. by  age more :li
```

```
-------------------------------------------------------------------------------
-> age = <25, more = No

          age     educat       more          N       cuse    contage         n1         N1
  1.       <25        Low         No         10         No         20          .          0
  2.       <25       High         No         50         No         20          .          0
  3.       <25       High         No         10        Yes         20         10         10
  4.       <25        Low         No          4        Yes         20          4         14

-------------------------------------------------------------------------------
-> age = <25, more = Yes

          age     educat       more          N       cuse    contage         n1         N1
  5.       <25        Low        Yes         53         No         20          .          0
  6.       <25       High        Yes        212         No         20          .          0
  7.       <25       High        Yes         52        Yes         20         52         52
  8.       <25        Low        Yes          6        Yes         20          6         58
```

```
. by  age  more :gen tot=sum(N)

. by  age more :li

-------------------------------------------------------------------------------
-> age = <25, more = No

          age     more          N       cuse    contage         n1         N1        tot
  1.      <25       No         10         No         20          .          0         10
  2.      <25       No         50         No         20          .          0         60
  3.      <25       No         10        Yes         20         10         10         70
  4.      <25       No          4        Yes         20          4         14         74

-------------------------------------------------------------------------------
-> age = <25, more = Yes

          age     more          N       cuse    contage         n1         N1        tot
  5.      <25      Yes         53         No         20          .          0         53
  6.      <25      Yes        212         No         20          .          0        265
  7.      <25      Yes         52        Yes         20         52         52        317
  8.      <25      Yes          6        Yes         20          6         58        323


. by  age  more :keep if _n==_N
(24 observations deleted)

. drop n1  N
```