STATA
TECHNICAL
BULLETIN

March 1992
STB-6

A publication to promote communication among Stata users

Subscriptions are available from Stata Corporation, email stata@stata.com, telephone 979-696-4600 or 800-STATAPC, fax 979-696-4601. Current subscription prices are posted at www.stata.com/bookstore/stb.html.

Previous Issues are available individually from StataCorp. See www.stata.com/bookstore/stbj.html for details.

Submissions to the STB, including submissions to the supporting files (programs, datasets, and help files), are on a nonexclusive, free-use basis. In particular, the author grants to StataCorp the nonexclusive right to copyright and distribute the material in accordance with the Copyright Statement below. The author also grants to StataCorp the right to freely use the ideas, including communication of the ideas to other parties, even if the material is never published in the STB. Submissions should be addressed to the Editor. Submission guidelines can be obtained from either the editor or StataCorp.

Contents of this issue

an1.1 STB categories and insert codes

Inserts in the STB are presently categorized as follows:

General Categories:			
an	announcements	*ip*	instruction on programming
cc	communications & letters	*os*	operating system, hardware, & interprogram communication
dm	data management		
dt	data sets	*qs*	questions and suggestions
gr	graphics	*tt*	teaching
in	instruction	*zz*	not elsewhere classified
Statistical Categories:			
sbe	biostatistics & epidemiology	*srd*	robust methods & statistical diagnostics
sed	exploratory data analysis	*ssa*	survival analysis
sg	general statistics	*ssi*	simulation & random numbers
smv	multivariate analysis	*sss*	social science & psychometrics
snp	nonparametric methods	*sts*	time-series, econometrics
sqc	quality control	*sxd*	experimental design
sqv	analysis of qualitative variables	*szz*	not elsewhere classified

In addition, we have granted one other prefix, *crc*, to the manufacturers of Stata for their exclusive use.

an16 Stata 3.0 released

Ted Anderson, CRC, 800-782-8272, 800-248-8272 (Canada), FAX 310-393-7551

I am pleased to announce that Stata 3.0 has been released and is shipping now. You should have already received a copy of the *Stata News* detailing the updates and providing upgrade information, but if you have not, please call. We will tell you about it over the phone and get the materials to you as quickly as possible.

an16.1 Implications for the STB

Joseph Hilbe, Editor, STB, FAX 602-860-1446

The release of Stata 3.0 causes a problem for the STB. Stata 3.0 has, among other things, an enhanced programming language that is incompatible with previous versions. It can, however, execute old Stata programs, but only after a change is made to them.

The STB will accept future submissions written in either Stata 2.1 or 3.0, but I am going to require that all programs run properly under Stata 3.0. Stata 3.0 includes a new `version` command that tells it the release under which the program was written. All programs written in 3.0 should include the line '`version 3.0`' immediately following the `program define`. If the program is written under 2.1, the line should read '`version 2.1`'.

Stata 2.1 does not provide a `version` command, however, and the inclusion of this line will prevent the program from running under 2.1 even if the program was written for 2.1. This problem is easily addressed. The ado-file `version.ado` is included in the `an16.1` directory of the STB-6 disk. If you continue to use Stata 2.1, please install the file; it reads

```
program define version
if "%_1"!="2.1" {
                di in red "ado-file not version 2.1"
                exit 198
        }
end
```

I suggest installing this in your `c:\ado` or `\stata\ado` directories. This ado-file will add a `version` command to your Stata and check that the version number of the program is 2.1. Ado-files written for 3.0 will not run with your Stata, but any ado-files that can run, will run. They will also run correctly under version 3.0 because Stata's internal `version` command will reset the interpretation of the program back to the old level.

If you use old ado-files previously published in the STB, and if you do upgrade to Stata 3.0, you will need to do something to make the old ado-files work. Before doing anything, however, look in the new manual to see if you still need the old STB program. Many of the previously published programs have been incorporated into the new Stata. Many, however, have not.

There are two ways you can make the old ado-files work which are outlined in the new manual—see [0] new. The best choice is to go back and edit each of the old ado-files and add a '`version 2.1`' immediately following the `program define`.

crc12 Oops!

It had to happen; there is an error in Stata 3.0. In [5s] ci (volume 2, p. 180), we state that the standard error of the mean s_μ is defined $\sqrt{s^2/(n-1)}$. We, of course, should have said that the standard error of the mean is $\sqrt{s^2/n}$.

The good news is that the manual accurately reflects the software. The bad news is that the manual really does accurately reflect the software, at least for those of you who received early shipments. Thus, the error also appears in the ci and cii commands themselves. Installing the crc directory from STB-6 disk will fix the problem. Install it even if you are uncertain whether your version has this problem. Be careful not to fall into old habits: the procedure for installing the crc directory has changed—see [7] stb_dos or [7] stb_unix.

If you do not have access to the diskettes, the fix is easy. Line 35 of cii.ado (found in \stata\ado or /usr/local/stata/ado) reads mac def S_4=`s'/sqrt(`n'-1). Change it to read mac def S_4=`s'/sqrt(`n'). That will fix both cii and ci.

dm2.2 Stat/Transfer 2.0 review update

Joseph Hilbe, Editor, STB, FAX 602-860-1446

In dm2, I reviewed DBMS/COPY and Stat/Transfer. If you will recall, my basic conclusion was that DBMS/COPY was a more professional effort, but that Stat/Transfer was adequate for certain applications and substantially less expensive.

Circle Systems has upgraded Stat/Transfer to produce Stat/Transfer Version 2 and has now produced a thoroughly professional data conversion utility. Among my previous criticisms was that it included only a menu mode, and not a very good one at that. It can now be run in both menu and command modes and the menu system has been completely redone. One nice feature is that it remembers how menus were previously filled in, so you can run it using menus and then later redo the transfer by typing only a single command. I have run it from both the DOS and Window's environments.

Circle Systems has expanded Stat/Transfer's coverage, for instance, it is the only transfer utility that supports Lotus 1-2-3 for Windows and Excel 3.0 for Windows. Like DBMS/COPY, it now supports SAS transport files (but DBMS/COPY is still unique in that it supports regular DOS and Unix SAS files, too). Stat/Transfer 2.0 supports:

Database	Spreadsheet	Statistical
Alpha Four	Excel	Gauss
Clipper	Lotus 1-2-3 (all)	SAS Transport
dBASE II–IV	Quattro Pro	SPSS export
Foxbase	Symphony	Stata
Paradox		SYSTAT

As before, Stat/Transfer users may select variables to be transferred; that is, it is easy to choose a subset of variables for conversion between packages. A new feature, it is now possible to specify variable types.

Stat/Transfer comes with a new 62-page manual. In most cases, however, a user will never need to consult the manual; the on-screen menu selections together with appropriate default settings (which one can modify) allow near intuitive use.

I put Stat/Transfer 2.0 through the same series of tests as previously. In two of the tests, the old Stat/Transfer did not produce desirable results. One test involved a source file with duplicates names. The old version simply transferred the duplicate names! The new version properly adds a numeric value, starting with "1", to make duplicate names unique, a solution similar to that adopted by DBMS/COPY. The second test checked what happens when there is an illegal variable name in the source file. Previously, Stat/Transfer produced a data set containing the illegal names, making the result unusable. Version 2.0, like DBMS/COPY, converts illegal variable names into legal ones.

Stat/Transfer 2.0 impressed me as an excellent transfer or conversion utility. If you use any of the major database or spreadsheet programs, or use Gauss, SAS, SPSS, or SYSTAT together with Stata, the time saved having Stat/Transfer will be well worth the expense.

Version 2.0 can be purchased for $129 from Circle Systems or from Computing Resource Center (1-800-782-8272). Upgrades from version 1.4 are available for $60. Academic rates have been set at $60 for new purchases and an upgrade price of only $30. Site licenses can also be arranged.

sed6 Quartiles, outliers, and normality: Some Monte Carlo results

Lawrence C. Hamilton, Department of Sociology, Univ. of New Hampshire, l_hamilton@unhh.unh.edu

In an *American Statistician* article, Frigge, Hoaglin, and Iglewicz (1989) note that different statistical packages employ a variety of definitions in constructing their boxplots. These can lead to substantially different-looking plots, based on the same data. Two main areas of disagreement are (1) approximations for sample quartiles and (2) identification of "severe" outliers. Their article gives examples, and lists eight alternative definitions for sample quartiles.

Boxplot quartiles determine not only the range spanned by the central box (drawn from Q1 to Q3), but also which observations get identified as "outliers" and, in some implementations, "severe outliers" (for an introduction to boxplots, see Hamilton 1990). Stata's `summarize, detail` and `graph, box` estimate quartiles with a version of Frigge et al.'s definition 5 (though Stata adds a provision for case weights). The `iqr` program described in STB-3 (Hamilton 1991) builds upon `summarize` and so also uses this definition. On the other hand, the `lv` (letter values) command introduced with Stata 3.0 reports "hinges" (approximate quartiles) based on definition 6.[1] The results differ for certain sample sizes: $n = 7$, 11, 15, 19, …. When they do differ, the hinges (definition 6) are closer to the center, thus producing a narrower "midspread" (approximate interquartile range) and labeling more cases as outliers.

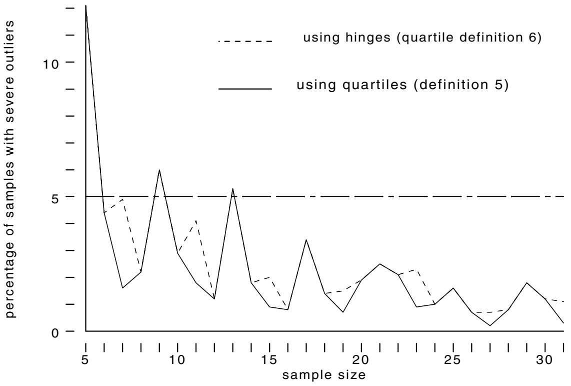
Hoaglin, Iglewicz, and Tukey (1986) performed a Monte Carlo investigation of outliers in samples from Gaussian populations. They found that the percentage of samples containing severe outliers (values below $\mathrm{Q1} - 3\mathrm{IQR}$ or above $\mathrm{Q3} + 3\mathrm{IQR}$) declines with increasing sample size, from $n = 10$ to 300. With still larger samples this pattern theoretically must reverse, however. We know that 100% of infinite-size Gaussian samples will contain outliers.

Figures 1–5 use new Monte Carlo data to explore this terrain in greater detail.[2] Figure 1 shows the observed percentage of samples from Gaussian populations that contain severe outliers, based on 1,000 artificial samples per sample size, $n = 5$ to 31. In small samples, both definition 5 and definition 6 behave unsmoothly. They "peak" (detect more outliers) at $n = 5$, 9, 13, 17, …. As expected, definition 6 detects more outliers than definition 5 at $n = 7$, 11, 15, 19, ... These cycles dampen, and differences between the two estimates fade, as sample size increases.

After about $n = 15$, the proportion of samples containing severe outliers falls below 5% (Figure 2). It eventually levels off and then rises slowly, exceeding 5% again only as sample sizes surpass about 20,000 cases (Figure 3). Circles in Figure 3 depict the experimental results, while the up-to-right line (actually a curve) shows theoretical expectations based on Gaussian population quartiles. Figure 4 carries the theoretical curve further, showing that the percentage of samples with outliers exceeds 90% as n surpasses about one million cases.

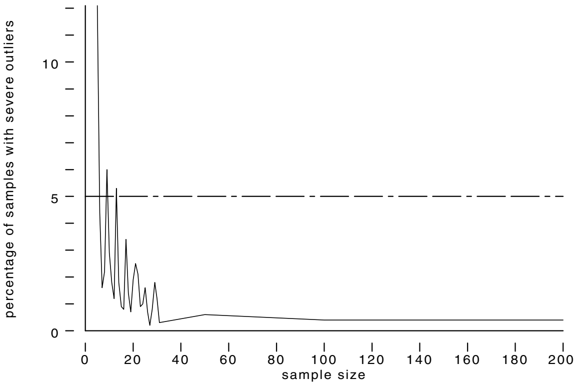
For $n > 2500$, experimental results generally follow Gaussian-population theory—because sampling variation in quartile estimates no longer plays a large role. Experimental standard deviations of the quartile estimates dropped from about .55 at $n = 5$ to .03 at $n = 2500$ and .008 at $n = 30,000$ (Figure 5).

Severe outliers challenge analysts in several ways. They may signify measurement errors, mixed populations, omitted variables, or other trouble. Whatever their cause, they tend to disproportionately influence analytical results. Figures 1–3 emphasize another reason for paying attention to outliers: they should be rare, if samples truly come from a Gaussian population. Less than 5% of samples drawn experimentally from a Gaussian population included any severe outliers, for sample sizes ranging from about $n = 15$ to 20,000. The presence of any severe outliers in samples of this size could therefore be viewed as sufficient evidence to reject normality at the 5% significance level.



Experimental Results, n=5 to 31

Figure 1



Experimental Results, n=5 to 200

Figure 2

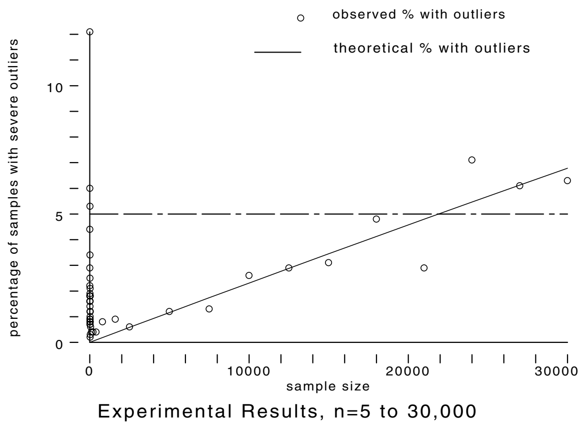


Figure 3

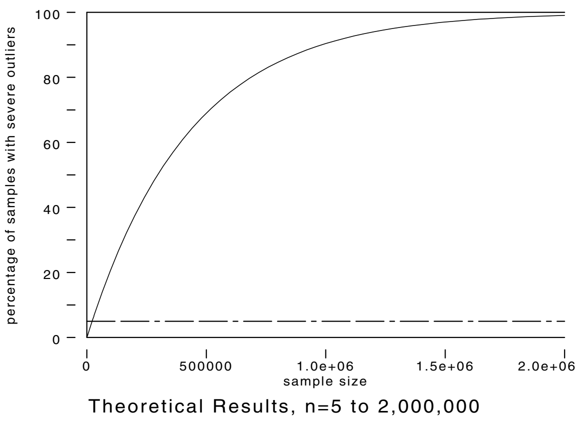


Figure 4

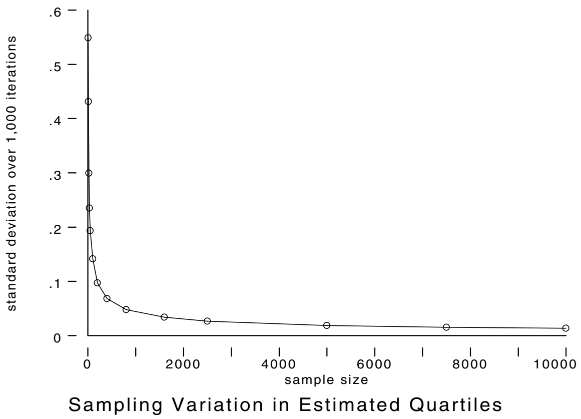


Figure 5

Notes

1. According to Frigge et al., SPSS and Statgraphics use definition 5; Minitab and Systat use definition 6. SAS defaults to definition 4, but offers 5 and certain others as options.

2. The accompanying data set, `outliers.dta`, contains the results graphed in Figures 1–4. I will be happy to share the complete Monte Carlo data (44,000 observations in two datasets) with anyone who sends two 1.44mb disks.

References

Frigge, M., D. C. Hoaglin, and B. Iglewicz. 1989. Some implementations of the boxplot. *The American Statistician* 43: 50–54.

Hamilton, L. C. 1990. *Modern Data Analysis: A First Course in Applied Statistics.* Pacific Grove, CA: Brooks/Cole.

——. 1991. Resistant normality check and outlier identification. *Stata Technical Bulletin* 3: 15–18.

Hoaglin, D. C., B. Iglewicz, and J. W. Tukey. 1986. Performance of some resistant rules for outlier labeling. *Journal of the American Statistical Association* 81: 991–999.

smv4 Oneway multivariate analysis of variance (MANOVA)

Joseph Hilbe, Editor, STB, FAX 602-860-1446

The syntax for `manova` is

`manova` *grp_var dep_vars*`, matrix`

`manova` is written under Stata 3.0. `manova` allows the user to perform a oneway multivariate analysis of variance on two dependent variables and one independent variable. The latter may be over two or more levels or groups. A `matrix` option is available that provides output of the W (within-groups sum of squares and cross-products) and T (total SS and CP) matrices together with their respective determinants. `Wilk's Lambda` λ is provided with an F statistic and corresponding significance.

In the case of non-factorial designs, λ is the ratio of the determinant of W to the determinant of T. It is a multivariate test evaluating whether the groups significantly differ in their measures on the set of dependent variables.

The displayed ANOVA table provides univariate F tests for each dependent variable. The F tests show whether the levels or groups differ for each dependent variable.

The following example consists of a data set comprised of one independent variable with four levels (coded as 1–4), and two dependent variables. As a comparison, I have provided partial output of an SPSS/PC+ MANOVA run on the same data set. I have placed `manova4.dta` on the STB-6 diskette for your use.

`manova.ado` is the most basic MANOVA procedure. Although a simple factorial design can be constructed as an ado-file, one cannot easily write an ado-file for a model allowing three dependent variables. This would necessitate finding the determinant of a 3×3 matrix—a more tedious task. Matrices of greater dimension would be required for models with a greater number of dependent variables.

Example

```
. describe

Contains data from manova4.dta
  Obs:    33 (max=126264)
 Vars:     3 (max=    99)
Width:    12 (max=   200)
  1. iv          float  %9.0g
  2. dv1         float  %9.0g
  3. dv2         float  %9.0g
Sorted by:
Note:  Data has changed since last save

. manova iv dv1 dv2, m
              Oneway Multivariate Analysis of Variance
      Number of obs in model  = 33        Number of groups = 4
      Model degrees freedom   = 3,29
 Variable |  HypSS      ErrSS      HypMS     ErrMS      F    Prob>F
--------------------------------------------------------------------
 dv1      |  7449.2531 32410.9893 2483.0844 1117.6203   2.22  0.1069
 dv2      | 23481.0713 43160.9893 7827.0238 1488.3100   5.26  0.0051
--------------------------------------------------------------------
 Wilk´s lambda (df:6,56) =      0.3327                 6.85  0.0000

    W matrix
                    32410.9893     32756.2393
                    32756.2393     43160.9893
    T matrix
                    39860.2424     40946.8788
                    40946.8788     66642.0606

 W determinant  = 325919149.1571     T determinant  = 979721808.9394
```

Below is output from SPSS/PC+ for the same problem:

```
           SPSS/PC+ The Statistical Package for IBM PC
MANOVA dv1 dv2 by iv(1,4).
* * ANALYSIS  OF  VARIANCE -- DESIGN   1 * *
 EFFECT .. IV
 Multivariate Tests of Significance (S = 2, M = 0, N = 13 )
 Test Name      Value  Approx. F Hypoth. DF   Error DF  Sig. of F

 Pillais       .77740    6.14659       6.00      58.00      .000
 Hotellings   1.67518    7.53830       6.00      54.00      .000
 Wilks         .33266    6.84870       6.00      56.00      .000
 Roys          .59124
 - - - - - - - - - -
```

```
   Univariate F-tests with (3,29) D. F.
   Variable   Hypoth. SS   Error SS Hypoth. MS   Error MS    F Sig. of F
   DV1        7449.25314 32410.9893 2483.08438 1117.62032  2.22176  .107
   DV2        23481.0713 43160.9893 7827.02377 1488.30998  5.25900  .005
  ----------------------------------------------------------------------
```

References

Bruning, J. L. and B. L. Kintz. 1987. *Computational Handbook of Statistics*, 3d ed. Glenview, IL: Scott, Foresman & Co.

Johnson, R. A. and D. W. Wichern. 1988. *Applied Multivariate Statistical Analysis.* Englewood Cliffs, NJ: Prentice–Hall.

Tabachnick, B. G. and L. S. Fidell. 1989. *Using Multivariate Analysis.* New York: Harper & Row.

smv5 Performing loglinear analysis of cross-classifications

D. H. Judson, Dept. of Sociology, Washington State Univ., Pullman, WA 99164-4020

Cross-classified count data is a common occurrence in the social, behavioral, and health sciences. Often only the table itself is available for analysis. `loglin` analyzes up to four-way tables of counts:

`loglin` *count varlist* [`=`*exp*] [`in` *range*] [`if` *exp*]`, fit(`*margins to be fitted*`)`
[`ltol(`*#*`) iter(`*#*`) offset(`*variable*`) anova keep resid collapse`]

[loglin will work with either Stata 2.1 or 3.0, but 2.1 users must install version.ado; see an16.1—ed.] `loglin` estimates a Poisson maximum-likelihood loglinear model. There are two cases: (1) You have only a summary table, and *count* indicates the number of cases that fall in each level of *varlist*, or (2) you have full information on all cases, so that each case should count once. If you fall into case 2, you would be better served to use the `poisson` command on your full data set.

For `loglin`, the *count* variable should contain positive integers reflecting the number of cases that fall in the cross-classification of *varlist*. The counts must be positive for each observed combination of the independent variables. If a count is zero, you may assume that it is a structural zero and replace it with a missing value or a zero cell weight; you may add a small positive constant, .5 for example, to zero cells; or, best of all, you may get more data.

Cell weights

If you specify `=`*exp*, `loglin` will assume that the expression represents cell weights. The default option for cell weights is no rescaling. If you wish to specify a particular cell as a structural zero, you may specify a cell weight of zero for that cell.

Functional form

This model falls in the class of generalized linear models with a categorical design matrix, a log link, and a Poisson distributed disturbance. Thus, the program generates a design matrix similar to the `anova` command, which is then passed to the `poilog`, a modification of the `poisson` regression program. The functional form of the model is log-linear:

$$\ln\big(\mathrm{E}(\mathrm{count})\big) = \mathrm{pred} + \mathrm{offset}$$

where the predicted value is a linear combination of the design matrix for the categorical independent variables in *varlist*. Unlike `poisson`, the `predict` command may not be used after `loglin`; specify `resid` instead, which will add the predicted cell frequencies to the copy of the data in memory. If `offset()` is specified, it is added to the predicted value for the purposes of estimation, so that the prediction is actually a predicted rate.

anova option and constraints

Like `anova`, the design matrix for `loglin` is not identified, hence constraints must be imposed on estimated parameters in order to generate an unique solution. There are two types used in this command: ANOVA-like and regression-like. In regression-like constraints, redundant levels of independent variables are summarily dropped (the first level is dropped, then any interaction with it). In ANOVA-like constraints, the first level is dropped, but the missing level is set equal to -1 times the sum of all the other levels. Interpret regression-like parameter estimates as deviations from the baseline level, and interpret ANOVA-like parameter estimates as deviations from the grand mean. To activate ANOVA-like constraints, specify the `anova` option. Otherwise, regression-like constraints will be used.

keep option

loglin normally drops the variables it generates for estimation. If you specify keep, these variables will be retained in the data for future use. Only the 1st-order variables (i.e., A1, A2, $\cdots$ Ax, B1, B2,$\cdots$ By, C1, C2, $\cdots$ Cz, etc.) will be labeled. Keeping the variables allows the user to create a new design matrix from the already existing variables, or to generate specialized models, but it does add substantially to the size of the data set. keep does not work when collapse is specified.

resid option

If you specify resid, estimated expected cell frequencies, residuals and standardized residuals will be calculated, displayed, and stored in the variables _cellhat, _resid, and _stdres. Residuals are calculated as actual cell count minus the estimated expected cell count. Standardized residuals are calculated as residual divided by the square root of the estimated expected cell count. If collapse is also specified, the above will be displayed but not retained in the data.

collapse option

Specify the collapse option only if your data contains more variables than you wish to work with in the specific model fit and you wish to analyze the subset specified in *varlist* as if it were the complete table.

collapse calculates cell counts for the variables in *varlist*, adding together the counts from all other variables not in *varlist* and placing them in appropriate cells (i.e., it collapses the table). It then generates a temporary data set on which it performs analysis. After calculations are completed, it restores the original data set. Note that if you specify both the resid and collapse options, your estimated expected cell frequencies, residuals, and standardized residuals will be displayed, but not saved with your original data set.

fit(margins to be fit)

The fit option is the engine that drives the loglin command. To specify a loglinear model, the fit option *must* be specified. This program generates hierarchical models, so that only the highest interaction is specified; all lower-level interactions will automatically be included. Separate the margins by commas, and specify interactions with a blank. The fit notation follows that developed by Feinberg (1981). For example, suppose we have summary data with three independent variables—iv1, iv2, and iv3—with counts coded in a variable called dv. If we wish to fit an independence model (i.e., [1] [2] [3] in Feinberg's notation), we type

```
loglin dv iv1 iv2 iv3, fit(iv1,iv2,iv3)
```

If we wish to fit a saturated model ([123] in Feinberg's notation), we type

```
loglin dv iv1 iv2 iv3, fit(iv1 iv2 iv3)
```

Estimation

loglin generates the appropriate design matrix for the configurations fit and passes that matrix to the poilog command for estimation. poilog, which is a minor modification of poisson, uses iteratively reweighted least squares, the estimates of which are equivalent to maximum-likelihood (See McCullagh and Nelder 1983, 31–34, for a discussion of the algorithm).

Convergence

The parameters ltol() and iter() may be used to control the maximization process. ltol() specifies the maximum change in the log likelihood that will be accepted as indicating convergence (default 1e-7), and iter() specifies the maximum number of iterations (default 100).

Theory of loglinear models

Assume that we have observations on N cases on k discrete variables $A_1, \cdots, A_k$.

Let

A_1 take on $0, 1, 2, \ldots, n_1$ discrete values;

A_2 take on $0, 1, 2, \ldots, n_2$ values;

...

A_k take on $0, 1, 2, \ldots, n_k$ values.

We arrange these observations in a contingency table or cross-tabulation or cross-classification table. For example, let A_1 take on 2 values, A_2 take on 2 values, and A_3 take on 2 values. Let A_1 represent race of murderer (0=white, 1=nonwhite), A_2 represent race of victim (0=white, 1=nonwhite), and A_3 represent death penalty (0=life, 1=death). This data is presented in Agresti (1984, 32). If we place the appropriate counts in the cells, then the cross-tabulation is

	Race of murderer			
	White		Nonwhite	
Death penalty:	Yes	No	Yes	No
White victim	19	132	11	52
Nonwhite victim	0	9	6	97

We have data on 3 variables, but there is a 4th variable, and the most important variable. Cell counts are the variable of real interest; they range from 0 to N for any cell. So we are particularly interested in modeling the distribution of cell counts.

Let $Y_{n_1,n_2,\ldots,n_k}$ be independently distributed POI($\mu_{n_1,n_2,\ldots,n_k}$). That is, each cell count is independently Poisson distributed with its own parameter $\mu_{n1,n2,\ldots,nk}$. Under the assumption that the cell counts are independently distributed, we can write the likelihood function as

$$L\left(\mu|Y\right) = \prod_{j=1}^{n_1 n_2 \cdots n_k} \frac{e^{-\mu_j}\,\mu_j^{Y_j}}{Y_j!}$$

and the log-likelihood function as:

$$\ln L\left(\mu|Y\right) = \sum_{j=1}^{n_1 n_2 \cdots n_k} -\ln(Y_j!) - \mu_j + Y_j \ln(\mu_j)$$

We have specified the stochastic component of the table; if we estimated these parameters using the cell frequency data, we would have one unique parameter for each cell. This would perfectly reproduce the table, but would not be particularly informative. To take the death penalty example, such a model would be presented verbally as “there are X whites who kill whites who get the death penalty; Y whites who kill nonwhites who get the death penalty; Z whites who kill nonwhites who do not get the death penalty; etc.” Obviously, this information can be obtained simply by looking at the cell counts.

We wish to find a more parsimonious model that summarizes, with “reasonable” goodness of fit, the data in the table. Thus, we specify a structural component to answer the question: What is the effect of race of victim, race of murderer, etc.? A convenient specification is as follows:

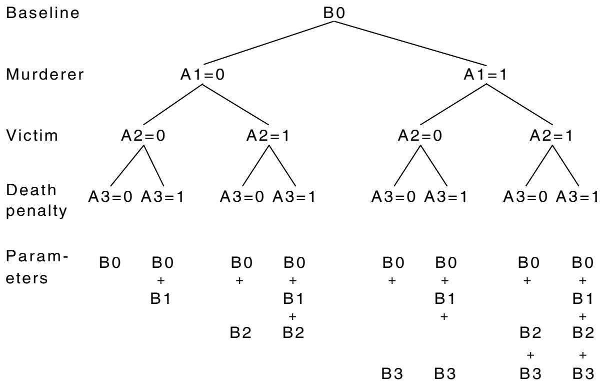
Let $\ln\mu = X\beta$, where X is a $(n_1 \times \cdots \times n_k \times v)$ design matrix of nonstochastic variables, β is a $(v \times 1)$ vector of design parameters, and μ is a $(n_1 \times \cdots \times n_k \times 1)$ vector of log cell parameters. Note that $\ln\mu_j = X_j\beta = X_{j,1}\beta_1 + X_{j,2}\beta_2 + \cdots + X_{j,v}\beta_v$. As can be seen, each row of the matrix X times the parameter vector β, determines the log of the cell parameter μ. This is substituted into the likelihood function, partial derivatives taken, and the root(s) of this equation in β can be solved iteratively using, for example, the Newton–Raphson algorithm or iteratively reweighted least squares.

The final question to be answered is, what is X and how is it specified? Since the only other variables of interest are $A_1,\ldots,A_k$, we use X as a design matrix of indicator variables specifying unique parameters for specific cells. Consider the example of a 2 × 2 × 2 death penalty table. If we were to specify the model that all the cell frequencies are the same (meaning that all the Poisson counts are governed by the same parameter, or that, e.g., P[death penalty|nonwhite murderer]=P[death penalty|white murderer], we specify X as a $n_1 \times \cdots \times n_k \times 1$ matrix and β as a 1×1 vector. Thus,

$$\ln(\mu) = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \cdot \left(\,\beta\,\right) = \begin{pmatrix} \beta \\ \beta \\ \vdots \\ \beta \end{pmatrix}$$

If this model fits the data well, we have simplified the structure of the table, from a maximum of 8 parameters, 1 per cell, to this minimum of 1.

Now consider a more sophisticated model: We wish to specify a baseline level of occupants in each cell, plus an additional amount in all death penalty cells (so that the margins of our estimated cell frequencies will be the same as the margins in the obtained table), an additional amount in all nonwhite victim cells (to fit that margin), and an additional amount in all nonwhite murderer cells (to fit that margin). This results in



(Drawn using Stage)

What does this model say?

White murderers ($A_1 = 0$) who kill white victims ($A_2 = 0$) receive the nondeath penalty ($A_3 = 0$) a baseline number of times (β_0).

White murderers ($A_1 = 0$) who kill white victims ($A_2 = 0$) receive the nondeath penalty ($A_3 = 0$) a baseline number of times (β_0) plus an additional amount (β_1) that adjusts for the relative frequency of death penalties.

White murderers ($A_1 = 0$) who kill nonwhite victims ($A_2 = 1$) receive the nondeath penalty ($A_3 = 0$) a baseline number of times (β_0) plus a race of victim effect (β_2).

White murderers ($A_1 = 0$) who kill nonwhite victims ($A_2 = 0$) receive the nondeath penalty ($A_3 = 0$) a baseline number of times (β_0) plus an additional amount (β_1) that adjusts for the relative frequency of death penalties plus a race of victim effect (β_2).

Additional effects can be determined for other combinations in the table. Please note that this interpretation depends on the use of regression-like constraints (Long 1984, 405) in the estimation process. If ANOVA-like constraints or some other constraints were used, the interpretation of the meaning of the parameters would be altered. However, by diagramming the table and collecting the parameters affecting each cell, the interpretation of the particular constraints used becomes clear. In general, individual parameters should be interpreted with great care, since their estimated value usually depends upon the constraints chosen. See Elswick, Gennings, Chinchilli, and Dawson (1991) for an excellent discussion of estimability and a simple method for determining estimable parameters.

Example

```
. use agr72

. describe

Contains data from c:\data\work\agr72.dta
  Obs:    24 (max= 10380)
 Vars:     4 (max=   100)
  1. count        float  %9.0g            # OF INDIVIDUALS
  2. dept         float  %9.0g            BERKELEY DEPARTMENT
  3. male         float  %9.0g            1=MALE APPLICANT
  4. admitted     float  %9.0g            1=ADMITTED TO BERKELEY
Sorted by:  dept  male  admitted

. * The following listing exhibits the "standard" data structure assumed by loglin.
. list, nodisp
         count       dept       male   admitted
  1.        19          1          0          0
  2.        89          1          0          1
  3.       313          1          1          0
  4.       512          1          1          1
  5.         8          2          0          0
  6.        17          2          0          1
  7.       207          2          1          0
  8.       353          2          1          1
  9.       391          3          0          0
 10.       202          3          0          1
 11.       205          3          1          0
 12.       120          3          1          1
 13.       244         4         0         0
 14.       131         4         0         1
 15.       279         4         1         0
 16.       138         4         1         1
 17.       299         5         0         0
 18.        94         5         0         1
 19.       138         5         1         0
 20.        53         5         1         1
 21.       317         6         0         0
 22.        24         6         0         1
 23.       351         6         1         0
 24.        22         6         1         1

. * First, we fit the independence model.  This means that the dept margin,
. * the male margin, and the admitted margin will be completely reproduced,
. * but the individual cells may not be completely reproduced.
. * Assume regression-like constraints unless otherwise indicated.

. loglin count dept male admitted, fit(dept,male,admitted) keep resid
Variable dept = A
Variable male = B
Variable admitted = C
Margins fit: dept,male,admitted
Note: Regression-like constraints are assumed.  The first level of each
variable (and all interactions with it) will be dropped from estimation.

Iteration 0: Log Likelihood =  -1404.5778 Change =  1
Iteration 1: Log Likelihood =  -1137.0895 Change =  267.4883
Iteration 2: Log Likelihood =  -1128.3747 Change =  8.7148
Iteration 3: Log Likelihood =  -1128.3649 Change =  .0098
Iteration 4: Log Likelihood =  -1128.3669 Change =  -.002
Note: Convergence achieved. Last change =   0.000000000

Poisson Regression (Log link function)        Number of cells = 24
Goodness of fit chi2(16) =  2097.672          Model chi2(7) = 552.422
Prob > chi2            =    0.0000            Prob > chi2   = 0.0000
Log Likelihood         = -1128.367

Variable |  Coefficient    Std. Error      t    Prob > |t|        Mean
---------+-------------------------------------------------------------
  _count |                                                   188.58333
---------+-------------------------------------------------------------
      A2 |     -.466793       .052737    -8.851     0.000    .16666667
      A3 |     -.016208       .046488    -0.349     0.730    .16666667
      A4 |     -.163844       .048316    -3.391     0.002    .16666667
      A5 |     -.468504       .052765    -8.879     0.000    .16666667
      A6 |     -.267522       .049723    -5.380     0.000    .16666667
      B2 |      .382868       .030275    12.646     0.000    .5
      C2 |     -.456739       .030507   -14.972     0.000    .5
   _cons |      5.44498       .039189   138.942     0.000    1
---------+------------------------------------------------------------

 count dept male admitted  _cellhat _resid  _stdres
   19    1    0    0       231.593  -212.593  -13.970
   89    1    0    1       146.678  -57.678   -4.762
  313    1    1    0       339.627  -26.627   -1.445
  512    1    1    1       215.101  296.898   20.244
    8    2    0    0       145.211  -137.211  -11.386
   17    2    0    1        91.969  -74.969   -7.817
  207    2    1    0       212.950   -5.950   -0.408
  353    2    1    1       134.871  218.129   18.783
  391    3    0    0       227.869  163.131   10.807
  202    3    0    1       144.320   57.680    4.801
  205    3    1    0       334.167  -129.167   -7.066
  120    3    1    1       211.643  -91.643   -6.299
  244    4    0    0       196.593   47.407    3.381
  131    4    0    1       124.511    6.489    0.581
  279    4    1    0       288.301   -9.301   -0.548
  138    4    1    1       182.594  -44.594   -3.300
  299    5    0    0       144.963  154.037   12.794
   94    5    0    1        91.811    2.189    0.228
  138    5    1    0       212.586  -74.586   -5.116
   53    5    1    1       134.640  -81.640   -7.036
  317    6    0    0       177.232  139.768   10.499
   24    6    0    1       112.249  -88.249   -8.329
  351    6    1    0       259.908   91.092    5.650
   22    6    1    1       164.611  -142.611  -11.115

. * Describe the data to see changes wrought by the resid and keep options.
. describe
Contains data from c:\data\work\agr72.dta
  Obs:    24 (max= 10380)
 Vars:    15 (max=   100)
  1. count       float  %9.0g             # OF INDIVIDUALS
  2. dept        float  %9.0g             BERKELEY DEPARTMENT
  3. male        float  %9.0g             1=MALE APPLICANT
  4. admitted    float  %9.0g             1=ADMITTED TO BERKELEY
  5. _count      float  %6.0f             CELL FREQUENCY
  6. A2          byte   %8.0g             dept==     2.0000
  7. A3          byte   %8.0g             dept==     3.0000
  8. A4          byte   %8.0g             dept==     4.0000
  9. A5          byte   %8.0g             dept==     5.0000
 10. A6          byte   %8.0g             dept==     6.0000
 11. B2          byte   %8.0g             male==     1.0000
 12. C2          byte   %8.0g             admitted==     1.0000
 13. _cellhat    float  %8.3f             ESTIMATED EXPECTED CELL FREQ.
 14. _resid      float  %7.3f             RESIDUAL
 15. _stdres     float  %7.3f             STANDARDIZED RESIDUAL
Sorted by:  dept  male  admitted
Note:  Data has changed since last save
. format A2-C2 %2.0f
. * Listing the variables A2-C2 displays the "design matrix" for the
. * loglinear model.  In addition, this design matrix can be modified to
. * fit more esoteric models such as those described in Agresti, 1984.
. list A2-C2, nodisplay
      A2  A3  A4  A5  A6  B2  C2
  1.  0   0   0   0   0   0   0
  2.  0   0   0   0   0   0   1
  3.  0   0   0   0   0   1   0
  4.  0   0   0   0   0   1   1
  5.  1   0   0   0   0   0   0
  6.  1   0   0   0   0   0   1
  7.  1   0   0   0   0   1   0
  8.  1   0   0   0   0   1   1
  9.  0   1   0   0   0   0   0
 10.  0   1   0   0   0   0   1
 11.  0   1   0   0   0   1   0
 12.  0   1   0   0   0   1   1
 13.  0   0   1   0   0   0   0
 14.  0   0   1   0   0   0   1
 15.  0   0   1   0   0   1   0
 16.  0   0   1   0   0   1   1
 17.  0   0   0   1   0   0   0
 18.  0   0   0   1   0   0   1
 19.  0   0   0   1   0   1   0
 20.  0   0   0   1   0   1   1
 21.  0   0   0   0   1   0   0
 22.  0   0   0   0   1   0   1
 23.  0   0   0   0   1   1   0
 24.  0   0   0   0   1   1   1
. * Now we fit the less parsimonious model [dept male],[dept admitted]:
. * This model assigns the following parameters to cells:
. *  dept male admitted
. *    1    0    0       constant
. *    1    0    1       constant +           C2
. *    1    1    0       constant +      B2
. *    1    1    1       constant +      B2 + C2
. *    2    0    0       constant + A2
. *    2    0    1       constant + A2      + C2        + CA22
. *    2    1    0       constant + A2 + B2      + AB22
. *    2    1    1       constant + A2 + B2 + C2 + AB22
. *    3    0    0       constant + A3
. *    3    0    1       constant + A3      + C2        + CA23
. *    3    1    0       constant + A3 + B2      + AB32
. *    3    1    1       constant + A3 + B2 + C2 + AB32
. *    4    0    0       constant + A4
. *    4    0    1       constant + A4      + C2        + CA24
. *    4    1    0       constant + A4 + B2      + AB42
. *    4    1    1       constant + A4 + B2 + C2 + AB42
. *    5    0    0       constant + A5
. *    5    0    1       constant + A5      + C2        + CA25
. *    5    1    0       constant + A5 + B2      + AB52
. *    5    1    1       constant + A5 + B2 + C2 + AB52
. *    6    0    0       constant + A6
. *    6    0    1       constant + A6      + C2        + CA26
. *    6    1    0       constant + A6 + B2      + AB62
. *    6    1    1       constant + A6 + B2 + C2 + AB62
. *
. * Note that all of the above functions are estimable.
. *
. * Note in the following that interactions specified in the fit option
. * do not need to be in the same order as in the variable specification.
. loglin count dept male admitted, fit(admitted dept, dept male) keep resid
Variable dept = A
Variable male = B
Variable admitted = C
Margins fit: admitted dept, dept male
Note: Regression-like constraints are assumed.  The first level of each
variable (and all interactions with it) will be dropped from estimation.
Iteration 0: Log Likelihood =  -1404.5778 Change =  1
Iteration 1: Log Likelihood =  -404.8278 Change =  999.75
Iteration 2: Log Likelihood =  -120.38834 Change =  284.43946
Iteration 3: Log Likelihood =  -92.429359 Change =  27.958981
Iteration 4: Log Likelihood =  -90.437172 Change =  1.992187
Iteration 5: Log Likelihood =  -90.398109 Change =  .039063
Note: Convergence achieved. Last change =   0.000000000
Poisson Regression (Log link function)         Number of cells= 24
Goodness of fit chi2(6) =    21.734            Model chi2(17) = 2628.359
Prob > chi2             =    0.0014            Prob > chi2    = 0.0000
Log Likelihood          =   -90.398
Variable |  Coefficient    Std. Error       t    Prob > |t|       Mean
---------+-------------------------------------------------------------
  _count |                                                   188.58333
---------+-------------------------------------------------------------
      A2 |     -1.43096       .232683    -6.150     0.000   .16666667
      A3 |      2.30438       .116079    19.852     0.000   .16666667
      A4 |      1.86308       .120481    15.464     0.000   .16666667
      A5 |      2.03498       .119666    17.005     0.000   .16666667
      A6 |      2.11643       .119283    17.743     0.000   .16666667
    AB22 |      1.07581       .228598     4.706     0.000   .08333333
    AB32 |     -2.63462       .123429   -21.345     0.000   .08333333
    AB42 |     -1.92709       .124644   -15.461     0.000   .08333333
    AB52 |     -2.75479       .135098   -20.391     0.000   .08333333
    AB62 |     -1.94356       .126826   -15.325     0.000   .08333333
      B2 |      2.03325        .10233    19.870     0.000   .5
      C2 |       .59346       .068381     8.679     0.000   .5
    CA22 |     -.050595        .10968    -0.461     0.649   .08333333
    CA23 |     -1.20915       .097259   -12.432     0.000   .08333333
    CA24 |     -1.25833       .101516   -12.395     0.000   .08333333
    CA25 |     -1.68296       .117333   -14.343     0.000   .08333333
    CA26 |     -3.26911       .167069   -19.567     0.000   .08333333
   _cons |      3.64886       .105828    34.479     0.000   1
---------+-----------------------------------------------------------

count  dept male admitted  _cellhat _resid _stdres
  19     1    0    0       38.431  -19.431   -3.134
  89     1    0    1       69.569   19.431    2.330
 313     1    1    0      293.569   19.431    1.134
 512     1    1    1      531.431  -19.431   -0.843
   8     2    0    0        9.188   -1.188   -0.392
  17     2    0    1       15.812    1.188    0.299
 207     2    1    0      205.812    1.188    0.083
 353     2    1    1      354.188   -1.188   -0.063
 391     3    0    0      384.998    6.002    0.306
 202     3    0    1      208.002   -6.002   -0.416
 205     3    1    0      211.002   -6.002   -0.413
 120     3    1    1      113.998    6.002    0.562
 244     4    0    0      247.633   -3.633   -0.231
 131     4    0    1      127.367    3.633    0.322
 279     4    1    0      275.367    3.633    0.219
 138     4    1    1      141.633   -3.633   -0.305
 299     5    0    0      294.077    4.923    0.287
  94     5    0    1       98.923   -4.923   -0.495
 138     5    1    0      142.923   -4.923   -0.412
  53     5    1    1       48.077    4.923    0.710
 317     6    0    0      319.031   -2.031   -0.114
  24     6    0    1       21.969    2.031    0.433
 351     6    1    0      348.969    2.031    0.109
  22     6    1    1       24.031   -2.031   -0.414
```

```
. describe
Contains data from c:\stata\work\agr72.dta
  Obs:    24 (max= 10380)
 Vars:    25 (max=   100)
  1. count        float  %9.0g           # OF INDIVIDUALS
  2. dept         float  %9.0g            BERKELEY DEPARTMENT
  3. male         float  %9.0g            1=MALE APPLICANT
  4. admitted     float  %9.0g            1=ADMITTED TO BERKELEY
  5. _count       float  %6.0f            CELL FREQUENCY
  6. A2           byte   %8.0g            dept==     2.0000
  7. A3           byte   %8.0g            dept==     3.0000
  8. A4           byte   %8.0g            dept==     4.0000
  9. A5           byte   %8.0g            dept==     5.0000
 10. A6           byte   %8.0g            dept==     6.0000
 11. B2           byte   %8.0g            male==     1.0000
 12. C2           byte   %8.0g            admitted== 1.0000
 13. CA22         byte   %8.0g
 14. CA23         byte   %8.0g
 15. CA24         byte   %8.0g
 16. CA25         byte   %8.0g
 17. CA26         byte   %8.0g
 18. AB22         byte   %8.0g
 19. AB32         byte   %8.0g
 20. AB42         byte   %8.0g
 21. AB52         byte   %8.0g
 22. AB62         byte   %8.0g
 23. _cellhat     float  %8.3f            ESTIMATED EXPECTED CELL FREQ.
 24. _resid       float  %7.3f            RESIDUAL
 25. _stdres      float  %7.3f            STANDARDIZED RESIDUAL
Sorted by:  dept  male  admitted
Note:  Data has changed since last save
```

```
. * Again we list A2-AB62 to display the "design matrix."
```

```
. format A2-AB62 %2.0f
```

```
. list A2-AB62,nodisplay
     A2  A3  A4  A5  A6  B2  C2  CA22  CA23  CA24  CA25  CA26  AB22 AB32
AB42 AB52  AB62
  1.  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  0   0
  2.  0   0   0   0   0   0   1   0   0   0   0   0   0   0   0  0   0
  3.  0   0   0   0   0   1   0   0   0   0   0   0   0   0   0  0   0
  4.  0   0   0   0   0   1   1   0   0   0   0   0   0   0   0  0   0
  5.  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0  0   0
  6.  1   0   0   0   0   0   1   1   0   0   0   0   0   0   0  0   0
  7.  1   0   0   0   0   1   0   0   0   0   0   0   1   0   0  0   0
  8.  1   0   0   0   0   1   1   1   0   0   0   0   1   0   0  0   0
  9.  0   1   0   0   0   0   0   0   0   0   0   0   0   0   0  0   0
 10.  0   1   0   0   0   0   1   0   1   0   0   0   0   0   0  0   0
 11.  0   1   0   0   0   1   0   0   0   0   0   0   0   1   0  0   0
 12.  0   1   0   0   0   1   1   0   1   0   0   0   0   1   0  0   0
 13.  0   0   1   0   0   0   0   0   0   0   0   0   0   0   0  0   0
 14.  0   0   1   0   0   0   1   0   0   1   0   0   0   0   0  0   0
 15.  0   0   1   0   0   1   0   0   0   0   0   0   0   0   1  0   0
 16.  0   0   1   0   0   1   1   0   0   1   0   0   0   0   1  0   0
 17.  0   0   0   1   0   0   0   0   0   0   0   0   0   0   0  0   0
 18.  0   0   0   1   0   0   1   0   0   0   1   0   0   0   0  0   0
 19.  0   0   0   1   0   1   0   0   0   0   0   0   0   0   0  1   0
 20.  0   0   0   1   0   1   1   0   0   0   1   0   0   0   0  1   0
 21.  0   0   0   0   1   0   0   0   0   0   0   0   0   0   0  0   0
 22.  0   0   0   0   1   0   1   0   0   0   0   1   0   0   0  0   0
 23.  0   0   0   0   1   1   0   0   0   0   0   0   0   0   0  0   1
 24.  0   0   0   0   1   1   1   0   0   0   0   1   0   0   0  0   1
```

```
. * For comparison purposes, we re-estimate the model using anova-like
. * constraints.  Note that expected cell frequencies, residuals,
. * standardized residuals, and goodness-of-fit chi-square statistics
. * all remain the same.  The parameter estimates change, and they are
. * interpreted differently than with regression-like constraints.
. loglin count dept male admitted, fit(dept male,dept admitted) resid anova
Variable dept =  A
Variable male =  B
Variable admitted =  C
Margins fit: dept male,dept admitted
Note: Anova-like constraints are assumed. The first level of each
variable (and all interactions with it) will be dropped from estimation.
The variable codings are constrained to sum to zero, so the first
level will equal -1 times the sum of the other levels.
Iteration 0: Log Likelihood = -1404.5778 Change = 1
Iteration 1: Log Likelihood = -404.8278 Change = 999.75
Iteration 2: Log Likelihood = -120.38834 Change = 284.43946
Iteration 3: Log Likelihood = -92.429359 Change = 27.958981
Iteration 4: Log Likelihood = -90.437172 Change = 1.992187
Iteration 5: Log Likelihood = -90.398109 Change = .039063
Note: Convergence achieved. Last change = 0.000000000
Poisson Regression (Log link function)         Number of cells=       24
Goodness of fit chi2(6) =  21.734              Model chi2(17) = 2628.359
Prob > chi2             =   0.0014             Prob > chi2    =   0.0000
Log Likelihood          = -90.398
Variable  |  Coefficient   Std. Error       t    Prob > |t|       Mean
-----------------------------------------------------------------------
   count  |                                                   188.58333
-----------------------------------------------------------------------
      A2  |     -.761805      .087896     -8.667     0.000            0
      A3  |      .539039      .039247     13.734     0.000            0
      A4  |      .426922        .0402     10.620     0.000            0
      A5  |     -.027346      .048433     -0.565     0.577            0
      A6  |     -.333357      .067454     -4.942     0.000            0
    AB22  |      1.21993      .086536     14.097     0.000            0
    AB32  |      -.63529      .036305    -17.499     0.000            0
    AB42  |     -.281525       .03699     -7.611     0.000            0
    AB52  |     -.695373      .042671    -16.296     0.000            0
    AB62  |     -.289757      .038207     -7.548     0.000            0
    AC22  |      .597214      .040075     14.902     0.000            0
    AC32  |      .017937      .034316      0.523     0.606            0
    AC42  |     -.006653      .036313     -0.183     0.000            0
    AC52  |     -.218969      .043538     -5.029     0.000            0
    AC62  |     -1.01204      .065215    -15.518     0.000            0
      B2  |      .334605      .019502     14.615     0.000            0
      C2  |     -.325782      .019502    -16.705     0.000            0
   _cons  |      4.80567      .025989    184.909     0.000            1
-----------------------------------------------------------------------
count   dept  male  admitted    _cellhat    _resid   _stres
 19     1     0     0             38.431     -19.431   -3.134
 89     1     0     1             69.569      19.431    2.330
313     1     1     0            293.569      19.431    1.134
512     1     1     1            531.431     -19.431   -0.843
  8     2     0     0              9.188      -1.188   -0.392
 17     2     0     1             15.812       1.188    0.299
207     2     1     0            205.812       1.188    0.083
353     2     1     1            354.188      -1.188   -0.063
391     3     0     0            384.998       6.002    0.306
202     3     0     1            208.002      -6.002   -0.416
205     3     1     0            211.002      -6.002   -0.413
120     3     1     1            113.998       6.002    0.562
244     4     0     0            247.633      -3.633   -0.231
131     4     0     1            127.367       3.633    0.322
279     4     1     0            275.367       3.633    0.219
138     4     1     1            141.633      -3.633   -0.305
299     5     0     0            294.077       4.923    0.287
 94     5     0     1             98.923      -4.923   -0.495
138     5     1     0            142.923      -4.923   -0.412
 53     5     1     1             48.077       4.923    0.710
317     6     0     0            319.031      -2.031   -0.114
 24     6     0     1             21.969       2.031    0.433
351     6     1     0            348.969       2.031    0.109
 22     6     1     1             24.031      -2.031   -0.414
```

```
. *  Finally, we fit the so-called "saturated model."  The saturated model
. *  specifies a number of parameters equal to the number of cells; thus,
. *  all cell frequencies are fit perfectly.

. loglin dept male admitted =count, fit(male admitted dept) keep resid
Variable dept = A
Variable male = B
Variable admitted = C
Margins fit: male admitted dept
Note: Regression-like constraints are assumed.  The first level of each
variable (and all interactions with it) will be dropped from estimation.

Iteration 0: Log Likelihood =  -1404.5778 Change =  1
Iteration 1: Log Likelihood =  -490.0153 Change =  914.5625
Iteration 2: Log Likelihood =  -121.71842 Change =  368.29688
Iteration 3: Log Likelihood =  -82.286781 Change =  39.431639
Iteration 4: Log Likelihood =  -79.618813 Change =  2.667968
Iteration 5: Log Likelihood =  -79.530922 Change =  .087891
Note: Convergence achieved. Last change =   0.000000000

Poisson Regression (Log link function)          Number of cells = 24
Goodness of fit chi2(0) =     0.000             Model chi2(23)  = 2650.094
Prob > chi2             =         .             Prob > chi2     = 0.0000
Log Likelihood          =   -79.531

Variable |  Coefficient    Std. Error       t    Prob > |t|       Mean
---------+-----------------------------------------------------------
  _count |                                                   188.58333
---------+-----------------------------------------------------------
      A2 |    -.864997       .421566    -2.052     0.051   .16666667
      A3 |     3.02427       .234985    12.870     0.000   .16666667
      A4 |     2.55273       .238242    10.715     0.000   .16666667
      A5 |       2.756       .236654    11.646     0.000   .16666667
      A6 |     2.81446       .236252    11.913     0.000   .16666667
      B2 |     2.80176       .236338    11.855     0.000   .5
    BC22 |    -1.05208       .262776    -4.004     0.000   .25
    BA22 |     .451513       .430985     1.048     0.305   .08333333
    BA23 |    -3.44746       .251585   -13.703     0.000   .08333333
    BA24 |    -2.66772       .252075   -10.583     0.000   .08333333
    BA25 |    -3.57495       .257782   -13.868     0.000   .08333333
    BA26 |    -2.69988       .248721   -10.855     0.000   .08333333
  BCA222 |     .832054       .510522     1.630     0.116   .04166667
  BCA223 |       1.177       .299636     3.928     0.001   .04166667
  BCA224 |     .970089       .302697     3.205     0.004   .04166667
  BCA225 |     1.25226       .330408     3.790     0.001   .04166667
  BCA226 |      .86318       .402771     2.143     0.042   .04166667
      C2 |      1.5442       .252786     6.109     0.000   .5
    CA22 |    -.790426       .497809    -1.588     0.125   .08333333
    CA23 |    -2.20464       .267231    -8.250     0.000   .08333333
    CA24 |    -2.16617       .275025    -7.876     0.000   .08333333
    CA25 |    -2.70135       .279089    -9.679     0.000   .08333333
    CA26 |    -4.12505       .329765   -12.509     0.000   .08333333
   _cons |     2.94444       .229475    12.831     0.000   1
---------+---------------------------------------------------------

count   dept male admitted _cellhat _resid  _stdres
  19      1    0    0      19.000    0.000    0.000
  89      1    0    1      89.000   -0.000   -0.000
 313      1    1    0     313.000   -0.000   -0.000
 512      1    1    1     512.000    0.000    0.000
   8      2    0    0       8.000    0.000    0.000
  17      2    0    1      17.000    0.000    0.000
 207      2    1    0     207.000   -0.000   -0.000
 353      2    1    1     353.000    0.000    0.000
 391      3    0    0     391.000    0.000    0.000
 202      3    0    1     202.000    0.000    0.000
 205      3    1    0     205.000    0.000    0.000
 120      3    1    1     120.000   -0.000   -0.000
 244      4    0    0     244.000    0.000    0.000
 131      4    0    1     131.000   -0.000   -0.000
 279      4    1    0     279.000    0.000    0.000
 138      4    1    1     138.000    0.000    0.000
 299      5    0    0     299.000   -0.000   -0.000
  94      5    0    1      94.000   -0.000   -0.000
 138      5    1    0     138.000    0.000    0.000
  53      5    1    1      53.000    0.000    0.000
 317      6    0    0     317.000    0.000    0.000
  24      6    0    1      24.000    0.000    0.000
```

```
    351      6    1    0     351.000   -0.000   -0.000
     22      6    1    1      22.000   -0.000   -0.000
```

Comparison of loglinear results: GAUSS and the loglin command

I have compared the results from `loglin` with results from several other packages, most importantly, the GAUSS loglinear analysis module, which uses the Newton–Raphson algorithm to maximize the log-likelihood directly (Long 1990). I have used three data sets included on the STB-6 disk: `agr72.dta`, a three-way cross-classification of admissions to graduate school at UC-Berkeley; (Sex by Department by Whether admitted, Agresti 1984, 71–73); `agr67.dta`, a three-way cross-classification of dumping severity in operations for duodenal ulcer patients (Operation type by Hospital by Dumping severity, Agresti 1984, 67; originally presented in Grizzle, Starmer, and Koch 1969); and `4way.dta`, a hypothetical four-way data set (`iv1` by `iv2` by `iv3` by `iv4`). To summarize, I found no differences between GAUSS loglinear and `loglin` to the fourth decimal place in any parameter estimate, estimated standard error, t-statistic, or probability value. I found an occasional small difference at the sixth decimal place and a very occasional small difference at the fifth decimal place. I found no differences to the second decimal place in any likelihood ratio chi-square, estimated cell frequency, or standardized residual, and likewise found small differences at the third and fourth decimal places. In no case was any substantive conclusion at risk.

References

Agresti, A. 1984. *Analysis of Ordinal Categorical Data.* New York: John Wiley & Sons.

Elswick, R. K., Jr., C. Gennings, V. M. Chinchilli, and K. S. Dawson. 1991. A simple approach for finding estimable functions in linear models. *The American Statistician* 45: 51–53.

Feinberg, S. 1981. *The Analysis of Cross-Classified Categorical Data.* Cambridge, MA: The MIT Press.

Grizzle, J. E., C. F. Starmer, and G. G. Koch. 1969. Analysis of categorical data by linear models. *Biometrics* 25: 489–504.

Long, J. S. 1984. Estimable functions in log-linear models. *Sociological Methods and Research* 12: 399–432.

——. 1990. Loglinear analysis. In *Gauss Applications.* Aptech Systems, Inc., 26250 196th Place SE, Kent, WA 98042, 206-631-6679.

McCullagh, P., and J. A. Nelder. 1983. *Generalized Linear Models.* London: Chapman and Hall.

sqv2 | A graphical method for assessing the goodness of fit of logit models

D. H. Judson, Dept. of Sociology, Washington State Univ., Pullman, WA 99164-4020

Assessing the goodness of fit of logit models is troublesome. Stata's `logit` command incorporates one method in its `tabulate` option. The option collapses the predicted values of the logit model into zero or one predictions: If the logit model prediction is greater than or equal to .5, the predicted value is one. If the logit model prediction is less than .5, the predicted value is zero. The `tabulate` option is less desirable, however, because it collapses into a binary space a prediction which is fundamentally continuous in nature; the predicted logit function.

An alternative suggestion is a pseudo-R^2 measure, with analogous properties to the regression R^2. One of these is already used in `logiodds.ado` *[and the 3.0* `logistic` *command—ed.].* Two others that have been proposed are described in Aldrich and Nelson (1984, 57–58). Aldrich and Nelson's Pseudo-R^2 = $c/(N+c)$, where c is the likelihood-ratio chi-square statistic and N is the sample size. They also suggest a measure derived from McKelvey and Zavoina (1975), which is written as pseudo-R^2 = ExSS$/($ExSS $+ 3.29 * N)$, where ExSS = ("Explained Sum of Squares") $= \sum(Y^\star - Y')$. In this formula, $Y^\star$ equals the predicted value of the dependent variable Y, and Y' equals the mean value of the predicted dependent variable $Y^\star$. These pseudo-R^2 measures, while potentially useful, are not ideal because they are only "analogous" to the regression R^2 and they are not widely accepted, reported, or used.

The alternative which I propose makes use of Stata's strength in graphical methods. Instead of collapsing or summarizing information from data, as all of the above methods do, let us take advantage of the subtlety of the human eye. This example makes use of a data set called `example.dta` on disk. In this data set (results from which are reported more fully in Judson and Duran-Aydintug 1990), we recorded experimental subjects' choices (1 or 0) in four different choice situations (`story1-story4`) where there were estimated rewards (`rewards1` and `rewards0`) and costs (`costs1` and `costs0`) associated with the choices. The first example assumes one intercept exists for all four choice situations, and generates a single predicted logit function. The file `logit1.do` generates the graph.

```
. use example.dta
. logit choice lvratio lcratio,tabulate

Iteration 0:  Log Likelihood =-82.121327
Iteration 1:  Log Likelihood = -66.62887
Iteration 2:  Log Likelihood =-64.207386
Iteration 3:  Log Likelihood = -64.01465
Iteration 4:  Log Likelihood =-64.012898
Iteration 5:  Log Likelihood =-64.012898
Logit Estimates                                  Number of obs =    126
                                                 chi2(2)       =  36.22
Log Likelihood =-64.012898                       Prob > chi2   = 0.0000
Variable |  Coefficient    Std. Error       t    Prob > |t|        Mean
---------+-------------------------------------------------------------
  choice |                                                     .6428571
---------+-------------------------------------------------------------
 lvratio |     .3401437      .0804834     4.226     0.000      1.649815
 lcratio |     .1112064      .0638161     1.743     0.084      .3032699
   _cons |     .2662457      .2202884     1.209     0.229             1
---------+-------------------------------------------------------------
    Comparison of Outcomes and Probabilities
 Outcome |    Pr < .5    Pr >= .5 |      Total
---------+------------------------+-----------
 Failure |        22          23 |         45
 Success |         9          72 |         81
---------+------------------------+-----------
   Total |        31          95 |        126
. predict predict
. gen logodds = log(predict/(1-predict))
. lab var logodds "PREDICTED LOGIT FUNCTION"
. sort logodds
. graph choice predict logodds,  symbol(oi) connect(.L) jitter(2) xline(0)
yline(.5) l1("CHOICE") t1("GRAPH OF LOGIT FUNCTION VS. ACTUAL CHOICE")
. * The above command generates figure 1
```

As can be seen, the immediate question that the eye assesses is how closely does the line come to the “bulk” of the points? By using the jitter option, the points have simulated mass. The vertical and horizontal lines mark off the regions corresponding to those in the tabulation in the `logit` command. Those points falling in the upper-right and lower-left quadrants are “hits,” while those points fall in the other two quadrants are “misses.” In addition, outliers become apparent, as in the one case far in the lower-right quadrant: The predicted logit function for this case is quite close to one, but instead the subject chose option zero.

The second example assumes that each separate story has its own intercept, and thus estimates a parameter for each story (`story1-story4`). While this is a more realistic model in our experimental context, it makes graphical analysis somewhat more troublesome. The file `logit2.do` generates four graphs on one screen, one for each story.

```
. use example.dta
. logit choice story1 story2 story3 story4 lvratio lcratio,tabulate nocons
Iteration 0:  Log Likelihood =-87.336545
Iteration 1:  Log Likelihood =-62.575621
Iteration 2:  Log Likelihood =-59.498732
Iteration 3:  Log Likelihood =-59.156149
Iteration 4:  Log Likelihood =-59.149952
Iteration 5:  Log Likelihood = -59.14995
Logit Estimates                                  Number of obs =    126
Log Likelihood = -59.14995
Variable |  Coefficient    Std. Error       t    Prob > |t|        Mean
---------+-------------------------------------------------------------
  choice |                                                     .6428571
---------+-------------------------------------------------------------
  story1 |      .369128      .3946969     0.935     0.352      .3174603
  story2 |     .6762513      .4486641     1.507     0.134      .2936508
  story3 |      .965393       .553165     1.745     0.084      .2301587
  story4 |    -1.045009      .5369355    -1.946     0.054      .1587302
 lvratio |     .3156666       .083842     3.765     0.000      1.649815
 lcratio |     .1114089      .0716004     1.556     0.122      .3032699
---------+-------------------------------------------------------------

    Comparison of Outcomes and Probabilities
 Outcome |    Pr < .5    Pr >= .5 |      Total
---------+------------------------+-----------
 Failure |        27          18 |         45
 Success |         9          72 |         81
---------+------------------------+-----------
   Total |        36          90 |        126
```

```
. predict predict
. gen logodds = log(predict/(1-predict))
. lab var logodds "PREDICTED LOGIT FUNCTION"
. sort story logodds
. graph choice predict logodds, by(story) symbol(oi) connect(.L) jitter(2)
xline(0) yline(.5) l1("CHOICE") t1("GRAPH OF LOGIT FUNCTION VS. ACTUAL
CHOICE")
. * The above command generates figure 2
```

Although this graph is somewhat less aesthetically pleasing, it does illustrate not only goodness of fit but also certain analytically interesting aspects of the data. In the case of story 2, there exists enough variation in predictions to make the complete logit function appear, and the marginal distribution of choices corroborates this. In the case of stories 1, 3, and 4, however, there is a preponderance of cases in one category or the other. For example, in story 3 virtually all predictions are in the upper-right quadrant, and most cases did indeed choose that option. In story 4, however, the predictions were almost uniformly in the lower-left quadrant, and most cases chose that option.

In summary, this technique adds one more graphical method to the data analyst's arsenal of methods. It is quite easy to apply and generates a result that is readily interpretable.

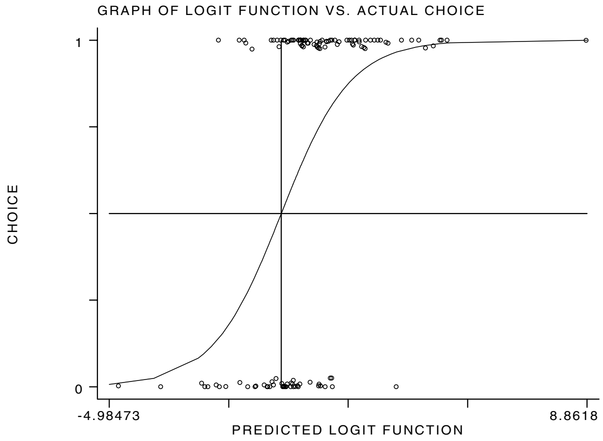


Figure 1

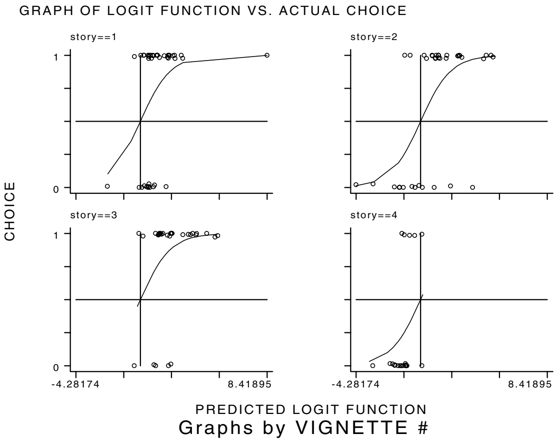


Figure 2

References

Aldrich, J. H. and F. D. Nelson. 1984. *Linear Probability, Logit, and Probit Models.* Beverly Hills, CA: Sage Publications.

Judson, D. H. and C. Duran-Aydintug 1990. Test of the satisfaction-balance decision model using direct numeric estimation. *Social Forces* 70: 475–494.

McKelvey, R. D. and W. Zavoina. 1975. A statistical model for the analysis of ordinal level dependent variables. *Journal of Mathematical Sociology* 4: 103–120.

srd11 | Generating ordered ("cascading") dummy variables

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvarda.bitnet

The syntax for the `cascade` command is

`cascade` *varname* [`if` *exp*] [`in` *range*], `generate(`*newvar*`)`

[cascade will work with either Stata 2.1 or 3.0, but 2.1 users must install version.ado; see an16.1—ed.] `cascade` adds a set of new dummy variables to your data set that are "cascading"; that is, rather than dummy variables that are coded 0/1 for each particular value in the variable (as occurs using '`tab` *varname*, `gen(`*newvar*`)`'—see [5s] tabulate in the Stata 3.0 reference manual), this creates variables coded 0/1 where a 1 is given if the case has the value for that variable or any *lower* value for that variable.

The `generate()` "option" is not optional. *newvar* should have no more than 6 characters.

The values of *varname* do not have to start with 1 and do not have to be consecutive, but they do have to be stored as `byte`s. If the variable is not a `byte`, you can easily convert it by using either `compress` or `recast` assuming it is an integer between −127 and 126.

This ado-file is useful when your categorical variable is *ordered*. In this case, each dummy variable shows the amount of change between categories rather than the amount of change between the category of interest and the reference category. For more, including a discussion of the meaning of these dummy's in Cox proportional hazards and in logistic regression, see Walter, Feinstein, and Wells (1987).

The ado-file automatically uses the lowest category (i.e., the category with the lowest value) as the reference category. If instead you want to use the highest category as the reference category, form a new variable that is equal to $a + b - oldvar$, where a and b are the lowest and highest values of the categories. Thus, if you have a variable coded 1–4 and want to use 4 as the reference category, type ‘gen *newvar*=5-*oldvar*’. Note that this trick only works if the lowest category is non-negative. In any case, after forming the new variable, proceed as above.

Example 1

```
. use auto
(1978 Automobile Data)
. compress
mpg was int now byte
rep78 was int now byte
trunk was int now byte
turn was int now byte
foreign was int now byte
make was str18 now str17
. cascade rep78 if rep78!=., gen(rep78)
```

The following output can be obtained at any time by removing the asterisks (comment markers) from the second and third last lines of the ado-file. This output shows the resulting coding (an easier-to-read example has been made up below this output by editing).

```
-> rep78=        1
        rep78    rep782    rep783    rep784    rep785
  2.        1         0         0         0         0

-> rep78=        2
        rep78    rep782    rep783    rep784    rep785
 10.        2         1         0         0         0

-> rep78=        3
        rep78    rep782    rep783    rep784    rep785
 40.        3         1         1         0         0

-> rep78=        4
        rep78    rep782    rep783    rep784    rep785
 58.        4         1         1         1         0

-> rep78=        5
        rep78    rep782    rep783    rep784    rep785
 69.        5         1         1         1         1

-> rep78=        .
        rep78    rep782    rep783    rep784    rep785
 74.        .         .         .         .         .
```

Here is the output, edited for clarity—you can see where the name “cascading” comes from.

```
        rep78    rep782    rep783    rep784    rep785
  2.        1         0         0         0         0
 10.        2         1         0         0         0
 40.        3         1         1         0         0
 58.        4         1         1         1         0
 69.        5         1         1         1         1
 74.        .         .         .         .         .
```

Contrast the above cascading dummies with “normal” dummy variables which would have one “1” in each of the second through fifth rows (as shown just below).

```
        rep78    rep782    rep783    rep784    rep785
  2.        1         0         0         0         0
 10.        2         1         0         0         0
 40.        3         0         1         0         0
 58.        4         0         0         1         0
 69.        5         0         0         0         1
 74.        .         .         .         .         .
```

The following `summary` and `tabulate` commands show that the correct number of cases are in each group; 97% of the cases have a 1 for `rep782` because 97% of the cases have greater than a 1 on `rep78` (of the non-missing cases); similarly, 16% of the non-missing cases have a 1 for `rep785` since 16% of the non-missing cases have a 5 for `rep78`.

```
. summ rep7*
Variable |   Obs        Mean   Std. Dev.       Min        Max
---------+---------------------------------------------------
   rep78 |    69    3.405797   .9899323          1          5
  rep782 |    69    .9710145   .1689948          0          1
  rep783 |    69    .8550725   .3546068          0          1
  rep784 |    69    .4202899   .4972216          0          1
  rep785 |    69    .1594203   .3687494          0          1
. tab rep78
      Repair|
 Record 1978|      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          2        2.90        2.90
          2 |          8       11.59       14.49
          3 |         30       43.48       57.97
          4 |         18       26.09       84.06
          5 |         11       15.94      100.00
------------+-----------------------------------
      Total |         69      100.00
```

Example 2

The variable mpg, tabulated below, does not have 1 as its minimum value and is not limited to consecutive integers—dummy variables are only formed for those values that exist and that are larger than the minimum value.

```
. cascade mpg, gen(mpg)
. describe mpg*
  8. mpg          byte   %8.0g             Mileage (mpg)
 14. mpg14        byte   %8.0g             mpg==14
 15. mpg15        byte   %8.0g             mpg==15
(output for mpg16–mpg25 omitted)
 26. mpg26        byte   %8.0g             mpg==26
 27. mpg28        byte   %8.0g             mpg==28
(output for mpg29–mpg30 omitted)
 30. mpg31        byte   %8.0g             mpg==31
 31. mpg34        byte   %8.0g             mpg==34
 32. mpg35        byte   %8.0g             mpg==35
 33. mpg41        byte   %8.0g             mpg==41
```

Example 3

The following are example regressions, showing the difference between “normal” and “cascading” dummy variables:

```
. use auto
(1978 Automobile Data)
. compress rep78
rep78 was int now byte
. * First, let´s make traditional dummy´s:
. tab rep78, gen(rep)
      Repair|
 Record 1978|      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          2        2.90        2.90
          2 |          8       11.59       14.49
          3 |         30       43.48       57.97
          4 |         18       26.09       84.06
          5 |         11       15.94      100.00
------------+-----------------------------------
      Total |         69      100.00
. * and now, ordinal dummy´s:
. cascade rep78 if rep78!=., gen(rep78)
. * Here is a list of all the new dummy variables, and the original variable:
. desc rep*
 13. rep78        byte   %8.0g         Repair Record 1978
 14. rep1         byte   %8.0g         rep78==     1.0000
 15. rep2         byte   %8.0g         rep78==     2.0000
 16. rep3         byte   %8.0g         rep78==     3.0000
 17. rep4         byte   %8.0g         rep78==     4.0000
 18. rep5         byte   %8.0g         rep78==     5.0000
 19. rep782       byte   %8.0g         rep78==2
 20. rep783       byte   %8.0g         rep78==3
 21. rep784       byte   %8.0g         rep78==4
 22. rep785       byte   %8.0g         rep78==5
```

Regression using traditional dummies:

```
. * Regression using traditional dummy´s:
. reg mpg weight weightsq rep2-rep5
  Source |      SS         df       MS         Number of obs =      69
---------+------------------------------      F(  6,    62) =   23.07
   Model |  1616.29232     6  269.382053      Prob > F      =  0.0000
Residual |  723.910582    62  11.6759771      R-square      =  0.6907
---------+------------------------------      Adj R-square  =  0.6607
   Total |   2340.2029    68  34.4147485      Root MSE      =   3.417
Variable | Coefficient    Std. Error       t  Prob > |t|        Mean
---------+-------------------------------------------------------------
     mpg |                                                    21.28986
---------+-------------------------------------------------------------
  weight |    -.013743      .0042661    -3.221    0.002      3032.029
weightsq |    1.32e-06      6.78e-07     1.950    0.056       9812703
    rep2 |   -.6012862      2.706235    -0.222    0.825       .115942
    rep3 |   -1.057246      2.516164    -0.420    0.676      .4347826
    rep4 |   -1.525988      2.594396    -0.588    0.559      .2608696
    rep5 |    1.230881      2.701994     0.456    0.650      .1594203
   _cons |     50.7208      7.163491     7.080    0.000             1
---------+------------------------------------------------------------
```

I now show regression using ordinal dummy's. Note that you could obtain the coefficients shown below from those above using subtraction; similarly, you could obtain the p-values shown below using the `test` command on the above regression. However, at least in some cases, I find the use of ordinal dummy's easier.

```
. reg mpg weight weightsq rep782-rep785
  Source |      SS         df       MS         Number of obs =      69
---------+------------------------------      F(  6,    62) =   23.07
   Model |  1616.29232     6  269.382053      Prob > F      =  0.0000
Residual |  723.910582    62  11.6759771      R-square      =  0.6907
---------+------------------------------      Adj R-square  =  0.6607
   Total |   2340.2029    68  34.4147485      Root MSE      =   3.417
Variable | Coefficient    Std. Error       t   Prob > |t|        Mean
---------+-------------------------------------------------------------
     mpg |                                                    21.28986
---------+-------------------------------------------------------------
  weight |    -.013743      .0042661    -3.221    0.002      3032.029
weightsq |    1.32e-06      6.78e-07     1.950    0.056       9812703
  rep782 |   -.6012862      2.706235    -0.222    0.825      .9710145
  rep783 |   -.4559596      1.380617    -0.330    0.742      .8550725
  rep784 |   -.4687423      1.064018    -0.441    0.661      .4202899
  rep785 |    2.756869      1.347234     2.046    0.045      .1594203
   _cons |     50.7208      7.163491     7.080    0.000             1
---------+-------------------------------------------------------------
```

References

Walter, S. D., A. R. Feinstein, and C. K. Wells. 1987. Coding ordinal independent variables in multiple regression analyses. *American Journal of Epidemiology* 125: 319–323.

srd12 Some model selection statistics

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvard.bitnet

The syntax for `amemiya` is

`amemiya`

[amemiya will work with either Stata 2.1 or 3.0, but 2.1 users must install version.ado; see an16.1—ed.] Type the one word, **`amemiya`**, directly after typing a regression to obtain a set of model selection statistics. Note that neither `if` nor `in` are allowed—if you estimate a regression using either of these options, you should temporarily drop the effected cases!

How does one compare, in an overall way, non-nested models? Part of the answer comes from the literature on choosing subsets of right-hand-side variables (nested models). `amemiya.ado` produces several model selection criteria to help choose between models, regardless of whether they are nested, and to help decide whether a model is guilty of "overfit" (too many right-hand-side variables).

These statistics are not perfect answers to either goal. In particular, if you really are choosing only from among nested models, two statistics not produced here would be of help: (1) Mallow's Cp (Mallows 1973); and, (2) Mean Squared Error of Prediction (MSEP). A good discussion of both can be found in Miller (1990).

For this ado-file, I am more concerned with non-nested models where the response variable (dependent variable, left-hand-side variable) is the same in each model to be compared. Many now use R^2, adjusted R^2, RMSE, etc. These are produced already by Stata. Several other statistics, however, have been suggested and these are not produced by Stata but are in this ado-file. These statistics include:

1) The adjusted R^2 based on **Amemiya's Prediction Criterion**.

This statistic penalizes R^2 more heavily than adjusted R^2 for each additional degree of freedom used on the right-hand side of the equation. The formula is

$$\text{Amemiya's } R^2 = 1 - \left(\frac{n+p}{n-p}\right)(1-R^2)$$

whereas the formula for adjusted R^2 is $1 - [(n/(n-p))(1-R^2)]$. In both formulas, n is the number of observations in the model, while p is the number of right-hand side variables, not including the constant. As with other R^2-type measures, maximize this. The difference between this and the standard adjusted-R^2 (as reported by Stata) is that Amemiya's version has a higher penalty for adding variables.

2) **Hocking's Sp criterion** (closely related to both Cp and MSEP).

This is an adjustment of the residual Sum of Squares; the formula is

$$\frac{X(n-p-1)}{(n-p)(n-p-2)} \quad \text{where} \quad X = \sum\left(\frac{\text{residual}^2}{n-p-1}\right)$$

Minimize this criterion. This can be interpreted as measuring the "expected squared distance between the true and predicted values of the dependent variable y" (Thompson 1978, 130).

3) **Akaike's Information Criterion**, presented in both its logged and unlogged forms.

Certain texts, packages, etc., use logged while others use the unlogged form—it is unlogged simply by exponentiating). The formula is

$$\ln\left(\frac{\text{Residual SS}}{n}\right)\frac{2(p+1)}{n}$$

Minimize this criterion. This has a tradeoff between parsimony and precision: the first part of the formula shows the precision issue, while the second part shows the penalty for increasing parameters.

4) **Schwarz's Bayesian Criterion**, presented in both its logged and unlogged forms (same rationale as AIC, above). The formula is

$$\ln\left(\frac{\text{Residual SS}}{n}\right)\frac{p * \ln(n)}{n}$$

The unlogged version is obtained by exponentiating the logged result. Minimize this criterion. Bayesians interpret this as "choosing the *a posteriori* most probable model" (Judge et al. 1988).

5) Finally, the **prediction sum of squares** (PRESS) is presented; the formula is

$$\sum\left(\frac{\text{residual}}{1-hat}\right)^2$$

where `hat` is the diagonal of the hat matrix from Stata. Minimize this criterion, also. This is a cross-validation type measure, originally based on deleting one observation at a time and re-estimating the equation.

These statistics can, in part, be shown to imply the following F-ratios for retaining variables in a regression (this table is taken from Maddala 1988, 431).

Criterion	F-value implications
Adjusted R^2	$F < 1$
Amemiya's PC adjusted R^2	$F < \frac{2n}{n+p}$, where p is the number of variables retained
Hocking Sp	$F < 2 + \frac{k+1}{n-pp-1}$, where k is the number of $n-pp-1$ variables deleted and pp is the sum of variables kept and deleted
Akaike's AIC	$F < \frac{n-pp}{n-p}$

The following example is taken from Madansky (1988, 187–188, discussion on 182–187) (and is on the STB-6 disk as **madansky.dta**). Note that the following do not exactly match what is in the text, but it is clear that at least some of the numbers in the text are wrong due to typo's. The value for AIC, however, always differs and I believe that the book is wrong; the values here match those given by SHAZAM.

```
. reg y x1
  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  1,    13) =   10.33
   Model |  206.577604     1  206.577604            Prob > F      =  0.0068
Residual |  260.075669    13  20.0058207            R-square      =  0.4427
---------+------------------------------            Adj R-square  =  0.3998
   Total |  466.653272    14  33.3323766            Root MSE      =  4.4728

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|    [95% Conf. Interval]
---------+-----------------------------------------------------------------
      x1 |   .9972144    .310331      3.213   0.007    .3267851    1.667644
   _cons |   12.34756   3.350041      3.686   0.003    5.110237    19.58489
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                             0.3631
Hocking´s Sp criterion is                                       1.5481
Akaike´s Information Criterion (AIC) is                         3.1196
     or, unlogged,                                             22.6371
Schwarz´s Bayesian Criterion (SBC) is                           3.2140
     or, unlogged,                                             24.8783
Prediction Sum of Squares (PRESS) is                          336.1367
```

```
. reg y x2
  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  1,    13) =   16.46
   Model |  260.749533     1  260.749533            Prob > F      =  0.0014
Residual |  205.903739    13  15.8387492            R-square      =  0.5588
---------+------------------------------            Adj R-square  =  0.5248
   Total |  466.653272    14  33.3323766            Root MSE      =  3.9798

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|    [95% Conf. Interval]
---------+-----------------------------------------------------------------
      x2 |   1.097699   .2705402      4.057   0.001    .5132321    1.682165
   _cons |   11.25614   2.944624      3.823   0.002    4.894666    17.61761
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                             0.4957
Hocking´s Sp criterion is                                       1.2256
Akaike´s Information Criterion (AIC) is                         2.8860
     or, unlogged,                                             17.9219
Schwarz´s Bayesian Criterion (SBC) is                           2.9804
     or, unlogged,                                             19.6963
Prediction Sum of Squares (PRESS) is                          260.1023
```

```
. reg y x3
  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  1,    13) =   68.53
   Model |  392.248783     1  392.248783            Prob > F      =  0.0000
Residual |   74.404489    13  5.72342223            R-square      =  0.8406
---------+------------------------------            Adj R-square  =  0.8283
   Total |  466.653272    14  33.3323766            Root MSE      =  2.3924

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|    [95% Conf. Interval]
---------+-----------------------------------------------------------------
      x3 |   -.912256   .1101955     -8.279   0.000   -1.150319   -.6741931
   _cons |   35.83242   1.730221     20.710   0.000    32.09451    39.57034
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                             0.8178
Hocking´s Sp criterion is                                       0.4429
Akaike´s Information Criterion (AIC) is                         1.8681
     or, unlogged,                                              6.4762
Schwarz´s Bayesian Criterion (SBC) is                           1.9625
     or, unlogged,                                              7.1174
Prediction Sum of Squares (PRESS) is                           98.3418
```

```
. reg y x1 x2

  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  2,    12) =  389.75
   Model |  459.578218     2  229.789109            Prob > F      =  0.0000
Residual |  7.07505472    12  .589587893            R-square      =  0.9848
---------+------------------------------            Adj R-square  =  0.9823
   Total |  466.653272    14  33.3323766            Root MSE      =  .76785

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|     [95% Conf. Interval]
---------+-----------------------------------------------------------------
      x1 |   .9784734   .0532824     18.364   0.000     .8623811    1.094566
      x2 |   1.081421   .0522045     20.715   0.000     .967677    1.195165
   _cons |   1.506976   .7775647      1.938   0.077    -.1871922    3.201144
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                             0.9802
Hocking´s Sp criterion is                                       0.0495
Akaike´s Information Criterion (AIC) is                        -0.3515
     or, unlogged,                                              0.7036
Schwarz´s Bayesian Criterion (SBC) is                          -0.2099
     or, unlogged,                                              0.8107
Prediction Sum of Squares (PRESS) is                           10.9673
```

```
. reg y x1 x3

  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  2,    12) =   39.32
   Model |  404.871672     2  202.435836            Prob > F      =  0.0000
Residual |  61.7816009    12  5.14846674            R-square      =  0.8676
---------+------------------------------            Adj R-square  =  0.8455
   Total |  466.653272    14  33.3323766            Root MSE      =   2.269

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|     [95% Conf. Interval]
---------+-----------------------------------------------------------------
      x1 |   .3024733   .1931731      1.566   0.143    -.1184147    .7233613
      x3 |  -.7958872   .1282436     -6.206   0.000    -1.075306   -.5164685
   _cons |   31.06062   3.461232      8.974   0.000     23.51924    38.60199
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                             0.8269
Hocking´s Sp criterion is                                       0.4320
Akaike´s Information Criterion (AIC) is                         1.8156
     or, unlogged,                                              6.1445
Schwarz´s Bayesian Criterion (SBC) is                           1.9572
     or, unlogged,                                              7.0792
Prediction Sum of Squares (PRESS) is                           96.8731
```

```
. reg y x2 x3

  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  2,    12) =   37.18
   Model |  401.811713     2  200.905856            Prob > F      =  0.0000
Residual |  64.8415597    12  5.40346331            R-square      =  0.8610
---------+------------------------------            Adj R-square  =  0.8379
   Total |  466.653272    14  33.3323766            Root MSE      =  2.3245

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|     [95% Conf. Interval]
---------+-----------------------------------------------------------------
      x2 |    .296209   .2226583      1.330   0.208    -.1889218    .7813398
      x3 |  -.7708549   .1508702     -5.109   0.000    -1.099573    -.442137
   _cons |   30.73721   4.182762      7.349   0.000     21.62375    39.85066
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                             0.8183
Hocking´s Sp criterion is                                       0.4534
Akaike´s Information Criterion (AIC) is                         1.8639
     or, unlogged,                                              6.4488
Schwarz´s Bayesian Criterion (SBC) is                           2.0055
     or, unlogged,                                              7.4299
Prediction Sum of Squares (PRESS) is                           90.7040
```

```
. reg y x1 x2 x3
  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  3,    11) =  277.78
   Model |  460.573689     3  153.524563            Prob > F      =  0.0000
Residual |  6.07958324    11  .552689385            R-square      =  0.9870
---------+------------------------------            Adj R-square  =  0.9834
   Total |  466.653272    14  33.3323766            Root MSE      =  .74343

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|    [95% Conf. Interval]
---------+-----------------------------------------------------------------
      x1 |   .8862159   .0859472     10.311   0.000    .6970473    1.075384
      x2 |   .9707826   .0967001     10.039   0.000     .757947    1.183618
      x3 |  -.1078856   .0803877     -1.342   0.207   -.2848178    .0690466
   _cons |   5.152686   2.818881      1.828   0.095    -1.05163      11.357
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                           0.9805
Hocking´s Sp criterion is                                     0.0507
Akaike´s Information Criterion (AIC) is                      -0.3698
     or, unlogged,                                            0.6909
Schwarz´s Bayesian Criterion (SBC) is                        -0.1810
     or, unlogged,                                            0.8345
Prediction Sum of Squares (PRESS) is                         10.4578
. reg y
  Source |       SS       df       MS               Number of obs =      15
---------+------------------------------            F(  0,    14) =       .
   Model |       0.00      0           .            Prob > F      =       .
Residual |  466.653272    14  33.3323766            R-square      =  0.0000
---------+------------------------------            Adj R-square  =  0.0000
   Total |  466.653272    14  33.3323766            Root MSE      =  5.7734

---------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|    [95% Conf. Interval]
---------+-----------------------------------------------------------------
   _cons |   22.45267   1.490691     15.062   0.000    19.25545    25.64988
---------------------------------------------------------------------------

. amemiya
Amemiya´s PC criterion R-squared is                           0.0000
Hocking´s Sp criterion is                                     2.3931
Akaike´s Information Criterion (AIC) is                       3.5709
     or, unlogged,                                           35.5475
Schwarz´s Bayesian Criterion (SBC) is                         3.6181
     or, unlogged,                                           37.2657
Prediction Sum of Squares (PRESS) is                        535.6989
```

References

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and Tsoung-Chao Lee. 1988. *Introduction to the Theory and Practice of Econometrics.* 2d ed. New York: John Wiley & Sons.

Madansky, A. 1988. *Prescriptions for Working Statisticians.* New York: Springer-Verlag.

Maddala, G. S. 1988. *Introduction to Econometrics.* New York: Macmillan Publishing Co.

Mallows, C. L. 1973. Some comments on Cp. *Technometrics* 15: 661–675.

Miller, A. J. 1990. *Subset Selection in Regression.* London: Chapman & Hall.

Thompson, M. L. 1978. Selection of variables in multiple regression: Part I. A review and evaluation *and* Part II. Chosen procedures, computations and examples. *International Statistical Review* 46: 1–19, 129–146.

tt3 Teaching biochemistry and chemistry with Stata: Understanding buffer solutions

Paul Geiger, Univ. of Southern California School of Medicine, EMAIL pgeiger@vm.usc.edu

Stata provides an ideal way for teaching students how to manipulate buffer calculations to further their understanding of this subject. For instance, in undergraduate chemistry or biochemistry the students usually learn the important property that the useful range of a buffer lies one pH unit on either side of the pK_a value of the specific compound chosen. Stata allows exploration of this “fact” using many pK_a values and concentration ranges and can show buffer behavior graphically. The student can ask “what if” questions easily and get results quickly for real and simulated compounds as well.

Buffer ions maintain solutions at constant pH, a term defined as the negative $\log_{10}$ of hydrogen ion concentration. A biochemical investigation can depend critically on selecting the correct buffer. Although buffer principles arise in beginning chemistry classes, even many advanced students don't fully understand the chemical basis of buffer action. Lack of thought often causes inappropriate selection of a buffer for a particular experimental design. Often, Tris buffer appears in the literature in experiments carried out at pH values near 7 to 7.4, a poor range for this compound (see below).

Buffering a solution depends on the use of weak acids or bases. These compounds don't completely dissociate in aqueous solution but exist in equilibrium mixtures as shown in the following generic dissociation reaction for a weak acid:

$$HB \rightleftharpoons H^+ + B^-$$

Adding strong acid (H^+) to a buffered system reduces the concentration of conjugate base (B^-) and the amount of conjugate acid (HB) correspondingly increases. Addition of strong base (OH^-) forces the reaction in the opposite direction. The following equation expresses the equilibrium constant, K_a, as the ratio of the product of the concentrations of hydrogen ion, $[H^+]$, and conjugate base, $[B^-]$, to the undissociated compound, $[HB]$:

$$K_a = \frac{[H^+][B^-]}{[HB]}$$

After rearranging, taking the log (base 10) and changing signs, we use the symbols pK_a and pH in the well-known Henderson-Hasselbalch equation (Atkinson et al. 1987; Segel 1976):

$$pH = pK_a + \log \frac{[B^-]}{[HB]}$$

This important equation allows the experimentalist to calculate the pH of a buffer solution if the molar ratio of buffer ions and the pK_a of the weak acid are known. Also, the molar ratio of $[B^-]/[HB]$ necessary to prepare a buffer solution at a specific pH value can be calculated if the pK_a is known.

The data file supplied in Stata format on the disk provides example values that might be used for instruction. The weak acid, HB, changes by 5 millimolar (mM) increments as does the conjugate base, B^-. The total concentration of acid and conjugate base add up to 100 mM, a reasonable concentration for certain biochemical work. Three buffers, N-2-acetamidoiminodiacetic acid (ADA), $pK_a = 6.6$; 3-(N-morpholino)-propanesulfonic acid (MOPS), $pK_a = 7.2$; and tris(hydroxymethyl)-aminomethane (Tris), $pK_a = 8.3$, illustrate realistic choices of `pK1`, `pK2` and `pK3` for demonstration (Boyer 1986, 39). Calculated pH values follow from use of the Stata `generate` command, the log (base10) of the concentration ratio, $[B^-]/[HB]$, shown in the table as variable `logB_HB`, and the Henderson-Hasselbalch equation, e.g., `generate pH1 = pK1 + logB_HB`. I chose twenty observations arbitrarily. More would require finer millimolar increments and perhaps show better definition of the curves at their extremes.

The first figure plots pH vs B^- and shows vividly how the buffers differ in their respective ranges of application. In fact the useful range really amounts to less than ±1 pH unit on close inspection. The second figure illustrates this point and gives some idea of the buffering capacity of a 100 mM solution of ADA. Using the Stata ado command '`dydx pH1 B, generate(dpH_dB)`' provides the variable, `dpH_dB`. This value plotted against `pH1` gives the illustrated U-shaped curve and a striking picture of the useful buffering range for ADA. A 1000 mM solution would use up that much more acid (base) before changing pH significantly away from the pK_a value. A plot of `dpH_dB` vs B^- also gives a U-shaped curve illustrating the change in pH for a change in amount of base present (not shown). Again, the least change occurs in the region closest to the 50/50 ratio of base to acid in the buffering solution.

In summary, I believe that Stata in the hands of teachers and students will lead to much better understanding and assimilation of chemical calculations as well as theory on both elementary and advanced levels. In addition, working with Stata is simple and straightforward. I invite comparison of the above exercise with the spreadsheet format for modeling developed by Atkinson et al., 1987.

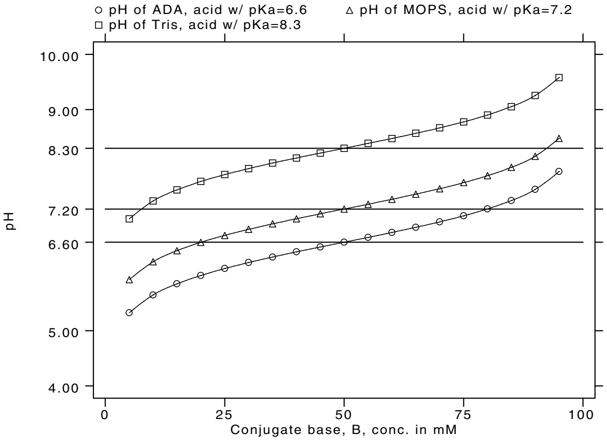


Figure 1

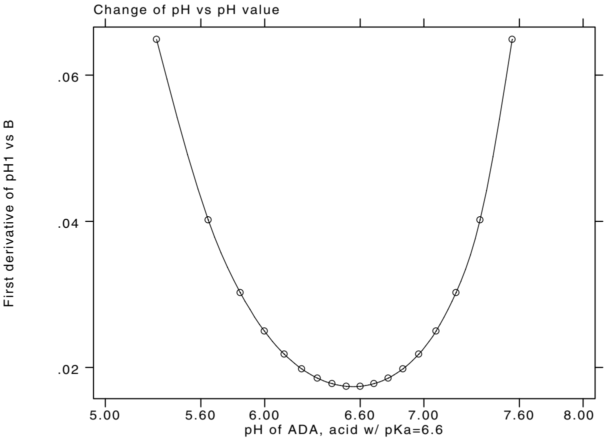


Figure 2

References

Atkinson, D. E., et al. 1987. *Dynamic Models in Biochemistry: A Workbook of Computer Simulations Using Electronic Spreadsheets*. Menlo Park, CA: Benjamin/Cummings.

Boyer, R. F. 1986. *Modern Experimental Biochemistry*. Menlo Park, CA: Addison–Wesley.

Segel, I. H. 1976. *Biochemical Calculations*. 2d ed. New York: John Wiley & Sons.