QV Calculator: Quasi-variances in Xlisp-Stat and on the Web

David Firth
Nuffield College, Oxford OX1 1NF, UK

http://www.stats.ox.ac.uk/~firth/qvcalc/

Revised 3 May 2000

Abstract

The most common summary of a fitted statistical model, a list of parameter estimates and standard errors, does not give the precision of estimated combinations of the parameters, such as differences or ratios. For this, covariances also are needed; but space constraints typically mean that the full covariance matrix cannot routinely be reported. In the important case of parameters associated with the discrete levels of an experimental factor or with a categorical classifying variable, the identifiable parameter combinations are linear contrasts. The *QV Calculator* computes 'quasi-variances' which may be used as an alternative summary of the precision of the estimated parameters. The summary based on quasi-variances is simple and permits good approximation of the standard error of any desired contrast. The idea of such a summary has been suggested by Ridout (1989) and, under the name 'floating absolute risk', by Easton, Peto & Babiker (1991). It applies to a wide variety of statistical models, including linear and nonlinear regressions, generalized-linear and GEE models, Cox proportional-hazard models for survival data, generalized additive models, *etc.*

The *QV Calculator* is written in Xlisp-Stat (Tierney, 1990) and can be used either directly by users who have access to Xlisp-Stat or through a web interface by those who do not. The user either supplies the covariance matrix for the effect parameters of interest, or, if using Xlisp-Stat directly, can generate that matrix by interaction with a model object.

Key words: floating absolute risk, generalized linear model, survival analysis.

1 The basic idea: quasi-variances

A standard presentational device, both in computer output and in published work, is a list of model parameter estimates and associated standard errors. Such a summary is not sufficient to provide standard errors for estimated *combinations* of the parameters, such as differences or ratios, which may often be of interest: for this, estimated covariances are also needed. Space constraints, however, typically mean that the full covariance matrix cannot routinely be reported.

In the important case of parameters $\beta_1, \ldots, \beta_p$ associated with the p levels of an experimental factor or with a p-category classifying variable, the identifiable linear combinations are *contrasts* of the form $\sum_{j=1}^{p} c_j \beta_j$, with $\sum_{j=1}^{p} c_j = 0$, including the simple between-level or between-group comparisons $\beta_i - \beta_j$. From the usual list of estimates and standard errors, with for example the constraint $\beta_1 = 0$ imposed, only the $p - 1$ comparisons $\{\beta_j = \beta_j - \beta_1, j = 2, \ldots, p\}$ can be made; standard errors for

other contrasts are not available in the absence of the relevant estimated covariances $\{\widehat{\mathrm{cov}}(\hat{\beta}_i, \hat{\beta}_j) : i, j = 2, \ldots, p\}$. The solution implemented in the *QV Calculator* is as follows: from the full covariance matrix, determine p 'quasi-variances', represented by $q_1, \ldots, q_p$, such that

$$\widehat{\mathrm{var}}\left(\sum_{i=1}^{p} c_i \hat{\beta}_i\right) \approx \sum_{i=1}^{p} c_i^2 q_i,$$

for all contrast vectors $\mathbf{c} = (c_1, \ldots, c_p)$. Then if $q_1, \ldots, q_p$ are reported along with the p estimates $\hat{\beta}_1, \ldots, \hat{\beta}_p$, an approximate standard error is available for any contrast $\sum c_i \hat{\beta}_i$ in the intuitively appealing form $(\sum c_i^2 q_i)^{1/2}$; the variance of a simple contrast $\hat{\beta}_i - \hat{\beta}_j$ is approximated simply by $q_i + q_j$. This basic idea appears to have been suggested first by Ridout (1989) and then, independently and under the name 'floating absolute risk', by Easton, Peto & Babiker (1991).

2 Approximation method

In the *QV Calculator* the quasi-variances $\{q_i\}$ are chosen to minimize the mean square of errors on the log scale, $\log(q_i + q_j) - \log v_{ij}$, over the set of variances v_{ij} for simple contrasts $\hat{\beta}_i - \hat{\beta}_j$. Thus *relative* error is controlled: small contrast variances are deliberately approximated more accurately than large ones, but the relative errors $\{(q_i + q_j)/v_{ij} - 1\}$ are of similar magnitude.

Ridout (1989) also suggests controlling relative error, by minimizing the mean of $\{v_{ij}/(q_i+q_j)\}^{1/2} - 1 + \frac{1}{2}\log\{(q_i+q_j)/v_{ij}\}$. In practice Ridout's method gives very similar results to those obtained from the *QV Calculator*, and indeed it can be shown (Menezes, 1999) that they are equivalent to first order. Easton, Peto and Babiker (1991), on the other hand, suggest choosing the quasi-variances to minimize the mean square of $(q_i + q_j) - v_{ij}$. This controls absolute rather than relative error, and can lead to poor approximation of small contrast variances (in terms of relative error) in order to achieve good approximations to larger ones, which is undesirable.

3 An example

McCullagh & Nelder (1989, pp204–8) analyse the rates of occurrence of damage incidents to cargo ships, using a Poisson-type loglinear model with allowance for overdispersion. The main-effects model reported by McCullagh & Nelder is summarized as follows by GLIM (Francis *et al.*, 1993):

```
? $display e$  !  parameter estimates, with "conventional" standard errors
          estimate        s.e.     parameter
     1      -6.406      0.2827      1
     2     -0.5433      0.2309      TYPE(2)
     3     -0.6874      0.4271      TYPE(3)
     4    -0.07596      0.3779      TYPE(4)
     5      0.3256      0.3067      TYPE(5)
     6      0.6971      0.1946      BUILT(2)
     7      0.8184      0.2208      BUILT(3)
     8      0.4534      0.3032      BUILT(4)
     9      0.3845      0.1538      PERIOD(2)
scale parameter 1.691
```

```
? $display v$ ! prints the variance-covariance matrix of estimates
  0.0799
 -0.0530  0.0533
 -0.0458  0.0428  0.1824
 -0.0396  0.0390  0.0384  0.1428
 -0.0408  0.0404  0.0412  0.0387  0.0941
 -0.0266  0.0038  0.0030  0.0020 -0.0002  0.0379
 -0.0343  0.0138  0.0043  0.0024 -0.0025  0.0272  0.0487
 -0.0344  0.0160  0.0126 -0.0111  0.0049  0.0281  0.0367  0.0919
 -0.0094  0.0009 -0.0002 -0.0003  0.0013 -0.0036 -0.0089 -0.0147  0.0237
```

Suppose that we are interested in the 5-level factor TYPE (the type of ship). The *QV Calculator* computes the quasi-variances as follows:

Level	Quasi-variance	Quasi-s.e.
1	4.038E-2	0.201
2	1.269E-2	0.113
3	0.140	0.374
4	0.105	0.324
5	5.393E-2	0.232

The '`quasi-s.e.`' column simply presents square roots of the quasi-variances $\{q_i\}$, which are then quasi standard errors on the same scale as the estimated effects.

The *QV Calculator* also gives an indication of the accuracy achieved in the approximation of contrast variances. Typically this is rather good; for example, relative errors for the standard errors of the ten simple contrasts in this case are all less than 1%. The quasi-variance approximation of contrast variances is exact in certain special situations, notably when $p = 3$: in that case the three quasi-variances q_1, q_2, q_3 summarize perfectly the two variances $\mathrm{var}(\hat{\beta}_2)$, $\mathrm{var}(\hat{\beta}_3)$ and single covariance $\mathrm{cov}(\hat{\beta}_2, \hat{\beta}_3)$. When $p > 3$ the quasi-variance approximation is exact only in special models, such as normal-linear or Poisson-loglinear models with a balanced design matrix; Menezes (1999) explores such 'exact' situations, and a fairly wide range of others where the approximation is nearly exact, in detail.

For a two-level factor, *i.e.*, $p = 2$, the notion of quasi-variances is redundant: there is only one contrast, and no covariance to take into account. Thus, for example, it would be pointless to compute quasi-variances for the two-level factor PERIOD in the above example: an infinite number of exact quasi-variance pairs (q_1, q_2) is available, and the choice among them arbitrary. If asked to compute quasi-variances for the effect of such a binary variable, the *QV Calculator* reports an error.

4 Suggested use of quasi-variances in presenting results

The quasi-variances may be used either as a supplement to, or in place of, 'conventional' standard errors which relate to contrasts with a reference category. Their interpretation is the same as that of the variances of independent estimated means $\hat{\mu}_1, \ldots, \hat{\mu}_p$ in a one-way layout; hence the terminology, 'quasi-variances'.

The presentation of p quasi-variances (or their square roots, 'quasi standard errors') has two main advantages over a conventional set of $p - 1$ standard errors of contrasts relative to a reference category. First, by construction the quasi-variances allow fairly accurate approximation of the variance of any desired contrast by any subsequent reader, without needing the full set of covariances. Second, the presentation of quasi-variances

facilitates the combination of results from different studies, where conventional standard errors might be problematic if different parametrizations (different reference categories, for example) were used in the reports of the different studies.

5 Using the *QV Calculator* with Xlisp-Stat

The *QV Calculator* is called in one of two ways: the function `qvcalc` works with a user-supplied variance-covariance matrix, while function `get-qvs` operates on a Lisp-Stat model object. (For general information on Lisp-Stat and object-oriented statistical modelling, see Tierney, 1990.) In either case the result is a quasi-variance object, an instance of the object prototype `qv-proto`. Methods are available for displaying such an object, for computing the exact and approximated variance of any contrast, and for calculating the worst error of approximation in the set of all possible contrasts.

5.1 Function `qvcalc`

Suppose that we are interested in the 5-level factor TYPE (the type of ship) in the example of Section 3. The relevant part of the covariance matrix is

```
0
0  0.0533
0  0.0428  0.1824
0  0.0390  0.0384  0.1428
0  0.0404  0.0412  0.0387  0.0941
```

Notice the column of zeros here, which reflects the fact that GLIM has set the parameter for TYPE(1) to zero and accordingly has omitted the corresponding column from the displayed variance-covariance matrix. Other model-fitting packages have different conventions, and it does not matter which one is used as long as the covariance matrix corresponding to all levels of the factor (i.e, all 5 levels here rather than just 4) is provided. Since the *QV Calculator* operates only on contrast variances, its results are independent of any constraints used in fitting the model.

The quasi-variances are computed, and stored with name `ship-type-qvs`, by

```
> (def ship-type-qvs
      (qvcalc 5 'VCL (list 0
                           0 0.0533
                           0 0.0428 0.1824
                           0 0.0390 0.0384 0.1428
                           0 0.0404 0.0412 0.0387 0.0941)))
```

The resultant object can now be displayed, or operated upon in various ways. Information can be obtained by sending `ship-type-qvs` the `:help` message:

```
> (send ship-type-qvs :help)
QV-PROTO
Quasi-variances for the coefficients associated with a factor
 in a regression model
Help is available on the following:

ADD-METHOD ADD-SLOT APPROXIMATED-SIMPLE-CONTRAST-VARIANCES
CONTRAST-VARIANCE COVARIANCE-MATRIX DELETE-DOCUMENTATION DELETE-METHOD
DELETE-SLOT DOC-TOPICS DOCUMENTATION FACTOR-NAME GET-METHOD HAS-METHOD
HAS-SLOT HELP INTERNAL-DOC ISNEW LEVEL-NAMES LEVELS METHOD-SELECTORS
```

```
MODEL-NAME NEW OWN-METHODS OWN-SLOTS PARENTS PRECEDENCE-LIST PRINT PROTO
QUASI-VARIANCES REPARENT RESPONSE-NAME RETYPE SHOW SIMPLE-CONTRAST-LABELS
SIMPLE-CONTRAST-VARIANCES SLOT-NAMES SLOT-VALUE SUMMARY WORST-ERRORS
```

Help can be obtained on a specific topic:

```
> (send ship-type-qvs :help :worst-errors)
WORST-ERRORS
Message args ()
Computes the worst error incurred by the quasi-variance approximation,
in the space of all possible contrasts.  The result is a 2-element list
containing the smallest and largest ratios of approximate to exact
standard error.
> (send ship-type-qvs :worst-errors)
(0.978997638650173 1.01584336156903)
```

The largest error incurred in this example is thus in the region of just 2%. It is computed from a formula derived by Menezes (1999), based on an eigenvalue calculation.

Information about the model and the effect of interest may be stored, as a reminder to the user, in the slots `:model-name` and `:factor-name`. For example,

```
>  (send ship-type-qvs :model-name "main effects model")
"main effects model"
>  (send ship-type-qvs :factor-name "ship type")
"ship type"
```

A nicely-formatted summary of the quasi-variances and associated information about the quality of approximation is obtained by

```
> (send ship-type-qvs :summary)
QUASI-VARIANCE SUMMARY
Model name: main effects model
Factor name: ship type
Covariance matrix:

     0.00         0         0         0         0
        0       5.33E-2   4.28E-2   3.90E-2   4.04E-2
        0       4.28E-2   0.18      3.84E-2   4.11E-2
        0       3.90E-2   3.84E-2   0.14      3.87E-2
        0       4.04E-2   4.11E-2   3.87E-2   9.40E-2

Quasi-variances (and corresponding quasi-se's) computed from
the above matrix by the QV Calculator:

       Level     Quasi-variance     Quasi-s.e.
       -----     --------------     ----------
         1           4.037E-2        0.201
         2           1.268E-2        0.113
         3          0.141            0.375
         4          0.105            0.324
         5           5.385E-2        0.232

For the standard error of a simple contrast, i.e., of a difference
between two levels, the error incurred by the quasi-variance
approximation is between -.7% and 0.9%.

In the set of *all* possible contrasts the approximation error
is between -2.1% and 1.6%.
```

The three mandatory arguments to `qvcalc` are: (i) the number of levels, (ii) a quoted symbol indicating the style in which variance-covariance information will be entered, and (iii) the variance-covariance information itself. An optional fourth argument can be set to `t` if the result required is the formatted summary rather than the quasi-variance object itself; this is used in communicating with the web interface (see Section 6).

```
> (help 'qvcalc)
QVCALC                                                    [function-doc]
Args: (levels style data &optional (web nil))
LEVELS   - the number of levels of the factor of interest (including
           any reference level)
STYLE    - the way in which the variance-covariance information will
           be supplied.  This must be one of
           'VC    -- the whole variance-covariance matrix, either as a
                     list or as a Lisp-Stat matrix representation
           'VCL   -- the lower triangle of the variance-covariance matrix,
                     as a list
           'CORR  -- coefficient standard errors (including a zero for any
                     reference level) and lower triangle of the
                     correlation matrix (excluding the diagonal), as a
                     single list.
           (Only styles 'VCL and 'CORR are available from the web input
           form.)
DATA     - a list of the variance-covariance information in the style
           chosen
WEB      - indicates whether this function is being called from the web
           page.  If so, the function returns a string summarizing the
           quasi-variance approximation, rather than a quasi-variance
           object.
Example of use:
(qvcalc 5 'VCL (list
             0
             0 0.0533
             0 0.0428 0.1830
             0 0.0390 0.0384 0.1427
             0 0.0404 0.0411 0.0387 0.0940))
for the ship-type example given on the QV Calculator web page
(http://www.stats.ox.ac.uk/~firth/qvcalc/); see also file 'example.lsp'.
The result is a quasi-variance object, unless WEB is non-nil in which
case it is a nicely-formatted summary string.  The summary string can be
obtained by sending the :SUMMARY message to the quasi-variance object.
```

Two further, alternative ways to make the object `ship-type-qvs` are thus

```
> (def ship-type-qvs
     (qvcalc 5 'CORR (list
          0  0.2309  0.4271  0.3779  0.3067   ; five standard errors
          0
          0  0.4340                           ; plus the lower triangle
          0  0.4468  0.2380                   ; of the correlation matrix
          0  0.5707  0.3142  0.3338)))
```

and

```
> (def ship-type-qvs
    (qvcalc 5 'VC
      #2A((0  0      0      0      0     )
          (0  0.0533 0.0428 0.0390 0.0404)    ; the whole 5 by 5
          (0  0.0428 0.1830 0.0384 0.0411)    ; covariance matrix
          (0  0.0390 0.0384 0.1427 0.0387)    ; for TYPE
          (0  0.0404 0.0411 0.0387 0.0940))))
```

5.2 Function get-qvs

5.2.1 Application of get-qvs to a model object

If the model of interest has itself been fitted in Lisp-Stat, the relevant variance-covariance information can be extracted directly from the model object by get-qvs, whose result is then the required quasi-variance object.

The file example.lsp when loaded into Xlisp-Stat makes an object, ship-model, corresponding to the main-effects model for the ship damage data. The contents of the file are reproduced in Appendix A.

The estimated parameters and standard errors as reported by poissonreg-model are

```
Constant                -6.40590      (0.282761)
TYPE(B)                -0.543344      (0.230935)
TYPE(C)                -0.687402      (0.427886)
TYPE(D)                 -7.596142E-2  (0.377864)
TYPE(E)                 0.325579      (0.306734)
BUILT(65-69)            0.697140      (0.194591)
BUILT(70-74)            0.818427      (0.220771)
BUILT(75-79)            0.453427      (0.303211)
OPERATED(75-79)         0.384467      (0.153799)
```

To make the quasi-variance object for TYPE as before, we can use either

```
> (def ship-type-qvs (get-qvs ship-model (list 1 2 3 4)
                       :model-name "main effects model"
                       :factor-name "ship type"))
```

or

```
> (def ship-type-qvs (get-qvs ship-model "type"
                       :model-name "main effects model"
                       :factor-name "ship type"))
```

The first argument of get-qvs is the model object of interest. The second argument identifies which are the particular model coefficients of interest, either by indicating their positions (list 1 2 3 4) in the list of parameter estimates (starting at 0), or using a string (here "type") which unambiguously matches the beginning of the variable name as printed out with the estimates and standard errors. The full description of get-qvs is

```
> (help 'get-qvs)
GET-QVS                                                    [function-doc]
Args: (model-object subset &key (reference-level 1) (factor-name nil)
           (model-name nil))
MODEL-OBJECT     - a model object (which inherits from
                   REGRESSION-MODEL-PROTO, or is an instance of
                   GEE-PROTO or COX-REGRESSION-PROTO, for example)
```

```
SUBSET           - either a predictor-name prefix string or a list of
                   indices corresponding to the coefficients of the
                   factor of interest
REFERENCE-LEVEL  - the level (if any) whose coefficient has been set to
                   zero to identify the parameters.  This is level 1 by
                   default.  Set this to NIL for a parametrization where
                   no coefficient has been constrained to zero and the
                   variance-covariance matrix of all p coefficients is
                   available, or to another value if a different
                   coefficient has been set to zero.
FACTOR-NAME      - an optional string as a reminder of which factor the
                   quasi-variance structure refers to.
MODEL-NAME       - an optional string as a reminder of which model the
                   quasi-variance structure refers to.
Computes quasi-variances from the covariance matrix of a subset of
estimated coefficients in a model.  The coefficients concerned should
correspond to the levels of a categorical variable (factor).
Result is a quasi-variance object, from prototype QV-PROTO.
```

A corresponding quasi-variance object for `year-built` can be made by

```
> (def ship-built-qvs (get-qvs ship-model "built"
    :model-name "main effects model"
    :factor-name "year ship built"))
```

and a printed summary obtained as before:

```
> (send ship-built-qvs :summary)
QUASI-VARIANCE SUMMARY
Model name: main effects model for ship damage rates
Response variable: DAMAGE-INCIDENT-COUNT
Factor name: year built
Covariance matrix:

        0         0         0         0
        0       3.79E-2   2.72E-2   2.81E-2
        0       2.72E-2   4.87E-2   3.67E-2
        0       2.81E-2   3.67E-2   9.19E-2

Quasi-variances (and corresponding quasi-se's) computed from
the above matrix by the QV Calculator:

       Level     Quasi-variance     Quasi-s.e.
       -----     --------------     ----------
         1           2.806E-2        0.168         reference category
         2           1.241E-2        0.111         BUILT(65-69)
         3           1.891E-2        0.138         BUILT(70-74)
         4           5.736E-2        0.239         BUILT(75-79)

For the standard error of a simple contrast, i.e., of a difference
between two levels, the error incurred by the quasi-variance
approximation is between  -3.6% and  6.5%.

In the set of *all* possible contrasts the approximation error
is between  -9.5% and  8.2%.
```

Some of the relative errors of approximation for BUILT are of the order of 8 or 9 percent in magnitude. This is quite a bit larger than the maximum error found above for TYPE,

but is still accurate enough for most purposes. It is worth bearing in mind that the covariance matrix itself was estimated, and is based on large-sample distribution theory.

5.2.2 To what models can `get-qvs` be applied?

A model object to be operated upon by `get-qvs` must have methods defined for the following four messages: `:response-variable`, `:predictor-names`, `:intercept` and `:coef-covariance-matrix`. Many Lisp-Stat model objects have methods for the first three of these messages, including all instances of `regression-model-proto` and its descendants (nonlinear regressions, generalized linear models) and instances of `gee-proto` (Lumley, 1996) and `cox-regression-proto` (Lumley, 1998). The fourth message, `:coef-covariance-matrix`, requires a new method to be defined. This is done automatically by the *QV Calculator* for instances and descendants of `regression-model-proto` (including nonlinear and generalized linear regressions) using the following definition:

```
(defmeth regression-model-proto :coef-covariance-matrix ()
"Method args: ()
Returns the estimated variance-covariance matrix of the
regression parameter estimates."
    (* (send self :xtxinv) (^ (send self :sigma-hat) 2)))
```

For other model objects, the method for `:coef-covariance-matrix` needs to be supplied by the user. For generalized estimating equation models and Cox proportional hazards models, for example, the required definitions are

```
(defmeth GEE-proto :coef-covariance-matrix ()
"Method args: ()
Returns the estimated variance-covariance matrix of the
regression parameter estimates."
    (send self :covariance-matrix))
```

and

```
(defmeth cox-regression-proto :coef-covariance-matrix ()
"Method args: ()
Returns the estimated variance-covariance matrix of the
regression parameter estimates."
    (send self :covariance-matrix))
```

In every case the method for `:coef-covariance-matrix` should return the matrix of estimated variances and covariances of all estimated regression coefficients, including any intercept term present in the model.

5.3 Further notes on quasi-variance objects

We have seen already the use of the methods `:summary`, `:worst-errors` and `:help` associated with a quasi-variance object. Other available methods are listed in response to the `:help` message. One other method worth mentioning specifically is `:contrast-variance`, which computes the variance of a specified contrast and its quasi-variance approximation. As an example, consider the contrast between ships built before 1970 with those built later:

```
> (send ship-built-qvs :contrast-variance (list 1 1 -1 -1))
Contrast coefficients:  (1 1 -1 -1)
Contrast variance:  .141
QV approximation:  .117
(0.141399933681981 0.116743532464883)
```

This particular contrast is in fact among those whose precision is least well approximated in this example, with a relative error of -9.1% on the scale of the standard deviation.

A list of all quasi-variance objects currently defined can be obtained by

```
> (qv-objects)
(SHIP-BUILT-QVS SHIP-TYPE-QVS)
```

An optional argument to `qv-objects` is a string with which the names of objects are compared: only those objects whose name begins with the given string are included in the list. Thus

```
> (qv-objects "ship-t")
(SHIP-TYPE-QVS)
```

5.4 Files in the *QV Calculator* distribution

There are just two files: the main file `qvcalc.lsp` contains all the code, and the other file `example.lsp` makes the main-effects model object `ship-model` for the ship damage data.

6 Using the web-based *QV calculator*

The web-based *QV Calculator* calls `qvcalc` to process the user-supplied variance-covariance information, and returns the formatted summary of the resulting quasi-variance object. Only the input styles `'VCL` and `'CORR`, in which the user supplies respectively the lower triangle of either the covariance matrix of the correlation matrix, are supported by the web input form. The URL is

http://www.stats.ox.ac.uk/~firth/qvcalc/

7 Some implementation details

7.1 Computing the quasi-variances

The criterion minimized by the quasi-variances $\{q_i\}$ is

$$\sum_{i,j} [\log v_{ij} - \log(q_i + q_j)]^2$$

where v_{ij} is the variance of simple contrast $\hat{\beta}_i - \hat{\beta}_j$. Formally, the optimization problem is equivalent to maximum likelihood estimation of the $\{q_i\}$ in the statistical model

$$\log v_{ij} \sim N(\mu_{ij}, \sigma^2)$$

with

$$\exp(\mu_{ij}) = q_i + q_j,$$

which is a generalized linear model with normal errors and exponential link. The *QV Calculator* exploits this fact to carry out the optimization using the `glim-proto` model-fitting routines of Xlisp-Stat; a new model prototype, `normal-exp-proto`, is defined in order to achieve this. This approach is similar to the one taken by Ridout (1989), whose approximation criterion is formally equivalent to maximum likelihood for a statistical model in which the contrast standard errors $\sqrt{v_{ij}}$ are gamma-distributed with means $\sqrt{(q_i + q_j)}$.

7.2 CGI for the web-based calculator

Communication between the web server and Xlisp-Stat is done on a Macintosh through a CGI script written in AppleScript. The CGI script used is based on a template by Jon Wiederspan, which may be found at http://www.comvista.com/. The part of the script specific to the *QV Calculator* is the passing of information supplied on the web form (stored as variables `NumberOfLevels`, `DataStyle` and `UserData` by the CGI script) to Xlisp-Stat, and the return by Xlisp-Stat of a string containing either the quasi-variance summary or an error message as appropriate (all error checking is carried out by Xlisp-Stat). The AppleScript code for this part is as follows:

```
set LispCode to ¬
    "(let* (    (levels " & NumberOfLevels & ") ¬
                (style '" & DataStyle & ") ¬
                (data (list " & UserData & ")) ¬
           ) ¬
           (qvcalc levels style data t))"
tell application "XLISP-STAT"
    «event miscdosc» LispCode
end tell
set OutputString to the result
```

8 Please cite use

The QV Calculator is provided free, but its author would appreciate acknowledgement in the form of citation in any published work that uses it. The present paper can be cited as:

Firth, D. (2000). Quasi-variances in Xlisp-Stat and on the web. *Journal of Statistical Software* **5.4**, 1–13. URL: http://www.stat.ucla.edu/journals/jss/.

9 References

Easton, D. F., Peto, J. & Babiker, A. G. A. G. (1991). Floating absolute risk: an alternative to relative risk in survival and case-control analysis avoiding an arbitrary reference group. *Statistics in Medicine* **10**, 1025–1035.

Francis, B., Green, M. & Payne, C., eds. (1983). *The GLIM system, Release 4 Manual.* Oxford: Clarendon Press.

Lumley, T. (1996). XLISP-Stat tools for building generalised estimating equation models. *Journal of Statistical Software* **1.3**, 1–20.

Lumley, T. (1998). Survival analysis in Xlisp-Stat: a semi-literate program. *Journal of Statistical Software* **3.2**, 1–90.

McCullagh, P. & Nelder, J. A. (1989). *Generalized Linear Models.* London: Chapman & Hall.

Menezes, R. X. de (1999). More useful standard errors for group and factor effects in generalized linear models. D. Phil. Thesis, Department of Statistics, University of Oxford.

Ridout, M. S. (1989). Summarizing the results of fitting generalized linear models to data from designed experiments. In: *Statistical Modelling: Proceedings of GLIM89 and the 4th International Workshop on Statistical Modelling held in Trento, Italy, July 17-21, 1989* (A. Decarli *et al.*, eds.), pp 262–269. New York: Springer.

Tierney, L. (1990). *LISP-STAT: An Object-oriented Environment for Statistical Computing and Dynamic Graphics.* New York: Wiley.

A Contents of file `example.lsp`

```
;
; Rate model for the ship damage data of McCullagh & Nelder (1989)
; "Generalized Linear Models", page 205
;

(require "glim")
(require "qvcalc")

(def aggregate-months-service
  (list
   127 63 1095 1095 1512 3353 0 2244
   44882 17176 28609 20370 7064 13099 0 7117
   1179 552 781 676 783 1948 0 274
   251 105 288 192 349 1208 0 2051
   45 0 789 437 1157 2161 0 542))

(def count-is-informative (> aggregate-months-service 0))

(def damage-incident-count (select
 (list
   0 0 3 4 6 18 0 11
   39 29 58 53 12 44 0 18
   1 1 0 1 6 2 0 1
   0 0 0 0 2 11 0 4
   0 0 7 7 5 12 0 1)
 (which count-is-informative)))

(def ship-type (select (repeat '(A B C D E) (repeat 8 5))
                  (which count-is-informative)))
(def year-built (select
    (repeat (repeat '(60-64 65-69 70-74 75-79) (repeat 2 4)) 5)
    (which count-is-informative)))
(def operating-period (select (repeat '(60-74 75-79) 20)
                              (which count-is-informative)))
(def log-exposure (log (select aggregate-months-service
                               (which count-is-informative))))

(defmeth poissonreg-proto :fit-scale ()
   (/ (sum (^ (send self :chi-residuals) 2)) (send self :df)))
; use mean X^2 rather than mean deviance to estimate overdispersion
```

```
(send poissonreg-proto :estimate-scale t)
; for a "quasi-Poisson" model as reported by McCullagh & Nelder,
; with Poisson-based covariance matrix scaled up to allow for
; the apparent overdispersion

(send poissonreg-proto :epsilon 0.00000001)      ; use strict
(send poissonreg-proto :epsilon-dev 0.00000001)  ; convergence criteria

(def ship-model
(poissonreg-model
    (append (indicators ship-type)
            (indicators year-built)
            (indicators operating-period))
damage-incident-count
:offset log-exposure
:predictor-names
     (append (level-names ship-type :prefix 'type)
             (level-names year-built :prefix 'built)
             (level-names operating-period :prefix 'operated))
:response-name 'damage-incident-count))

; Now define quasi-variance objects for SHIP-TYPE and YEAR-BUILT

; Note that there is no point in quasi-variances for the third factor,
; OPERATING-PERIOD, since it only has two levels and so a single
; standard error suffices for the one and only contrast that exists.

(def ship-type-qvs (get-qvs ship-model "type"
      :model-name "main effects model for ship damage rates"
      :factor-name "ship type"))

; Here we could alternatively have used (list 1 2 3 4) in place of the
; prefix "type", since the four relevant coefficients appear in the 2nd
; thru 5th positions in the list for this model (and indexing in Lisp
; starts at 0).

(send ship-type-qvs :summary)

(def ship-built-qvs (get-qvs ship-model "built"
      :model-name "main effects model for ship damage rates"
      :factor-name "year built"))

; Here we could alternatively have used (list 5 6 7) in place of the
; prefix "built".

(send ship-built-qvs :summary)
```