`exactLoglinTest`: A Program for Monte Carlo Conditional Analysis of Log-linear Models

Brian S. Caffo

February 5, 2013

Nuisance parameters are parameters that are not of direct interest to the inferential question in hand. In a frequentist or likelihood paradigm, a common tool for eliminating nuisance parameters is to condition on their sufficient statistics. The same technique is useful (though rarely used) in a Bayesian settings, as it eliminates the need to put priors on nuisance parameters.

For log-linear models, conditional analysis suffers from two main drawbacks.

1. The set of lattice points contained in the conditional distribution is difficult to manage, computationally or analytically.

2. The sufficient statistics for the nuisance parameters are not ancillary to the parameters of interest.

In this manuscript we address only the first drawback using `exactLoglinTest`.

1 The Problem

The observed data, $y = (y_1, \ldots, y_n)$, are modeled as Poisson counts with a means, $\mu = (\mu_1, \ldots, \mu_n)$, satisfying

$$\log \mu = x\beta$$

under the null hypothesis. Here x is a full rank $n \times p$ design matrix. It is easily shown that the sufficient statistics for β under the null hypothesis are $x^t y$, where a superscript t denotes a transpose. Let h be a test statistic of interest where larger values of h support the alternative hypothesis. Two examples are the Pearson Chi-Squared statistic and the deviance. An exact test relative to h can be performed via the conditional P-value

$$\text{Prob}\{h(y) \geq h(y_{obs}) | x^t y = x^t y_{obs}\} = \sum_{\{y \in \Gamma\}} \frac{I\{h(y) \geq h(y_{obs})\}}{C \prod y_i!}$$

where y_{obs} is the observed table, C is a normalizing constant and $\Gamma = \{y | x^t y = x^t y_{obs}\}$ (often referred to as the reference set).

The term "exact" is used to refer to tests that guarantee the nominal type I error rate unconditionally. Thus a test that never rejects the null hypothesis is technically exact in any situation. Therefore, exactness is not in itself a sufficient condition for a test to be acceptable. Moreover, this example (never rejecting) is particularly relevant in our setting because Γ may contain one or few elements. Hence the conditional P-value will be exactly or near one regardless of the evidence in the data vis-a-vis the two hypotheses. However, it is also the case that the conservative conditional tests can produce P-values that are smaller than those calculated via Chi-squared approximations (see Subsection 3.2 for an example).

1.1 Binomial Calculations

Conditional inference for Poisson log-linear models contains conditional inference for binomial-logit models as a special case. Consider a binomial logit models of the form, $b_i \sim \text{Bin}(n_i, p_i)$ for $i = 1, \ldots, k$ and

$$\text{logit}(p_i) = z_i \gamma + x_i' \beta, \tag{1}$$

where γ is a scalar and β is a p dimensional vector. Frequently, x_i' contains only a strata indicator and an intercept term. In this case conditioning on the sufficient statistic for β results in standard conditional logistic regression. For this purpose, we suggest the `coxph` function as described in [7]. Instead we consider the more general case where β is arbitrary vector of nuisance parameters. However, the reader should again be warned that the loss of information from conditioning can sometimes be quite severe in these problems and hence produce useless results.

Consider testing $H_0\gamma = 0$ versus some alternative. The following model model is equivalent to the null modell for (1):

$$y_{ij} \sim \text{Poisson}(\mu_{ij}) \qquad \log(\mu_{i1}) = \alpha_i + x_i'\beta \qquad \log(\mu_{i2}) = \alpha_i, \tag{2}$$

for $j = 1, 2$ and $i = 1, \ldots, k$. The sufficient statistics for the α_i are $y_{i1} + y_{i2} = y_{i+}$. Then it is easy to show that the conditional distribution of $y_{i1}|y_{i+}$ is precisely the model given by (1) where

$$\begin{aligned} p_i &= \mu_{i1}/\mu_{i+} \\ b_i &= y_{i1} \\ n_i &= y_{i+}. \end{aligned}$$

Therefore, conditioning out the nuisance parameters $\{\alpha_i\}$ and β for the Poisson log-linear model yields exactly the same (null) conditional distribution as conditioning out β in model (1). Furthermore, this exercise indicates exactly how to perform the calculations, which is useful since `exactLoglinTest` only accepts models in the form of Poisson log-linear models.

Currently `exactLoglinTest` is useful for tests of $\gamma = 0$. With modifications, the central ideas could be used to calculate a Monte Carlo estimate of the conditional likelihood for γ. (It is possible to use `mcexact` as is for this purpose. However, we have had mixed success in this endeavor and it is best avoided due to numerical instability.)

2 `exactLoglinTest`

The software `exactLoglinTest` is an implementation of the algorithms presented in [2] and [3]. At the heart of both algorithms is a sequentially generated rounded normal approximation to the conditional distribution. We refer the reader to those papers for a more complete description.

You can obtain a copy of `exactLoglinTest` at as well as a no-web [6] version of this document at

`http://www.biostat.jhsph.edu/~bcaffo/downloads.htm`

You can install `exactLoglinTest` with `R CMD INSTALL`, on Unix and Linux, while the binaries are available for Windows. Assuming it is installed, one can load `mcexact` with

```
> library(exactLoglinTest)
> set.seed(1)
```

Here, the optional argument `lib.loc` is necessary if the package has been installed into one of the paths that R automatically checks. We also set the random number seed to a specific value which is a good practice for Monte Carlo procedures.

3 Examples

3.1 Residency Data

Assuming `exactLoglinTest` has been properly installed, the residency data can be obtained by the command

```
> data(residence.dat)
```

This data is a 4×4 table of persons' residence in 1985 by their residence in 1980. See Table 1 for the complete data. The data frame, `residence.dat`, contains the counts stacked by the rows. The extra term `sym.pair` is used to fit a quasi-symmetry model. For details on the quasi-symmetry model see [1]. To obtain a Monte Carlo goodness of fit test of quasi-symmetry versus a saturated model involves the following command

```
> resid.mcx <- mcexact(y ~ res.1985 + res.1980 + factor(sym.pair),
+                      data = residence.dat,
+                      nosim = 10 ^ 2,
+                      maxiter = 10 ^ 4)
> resid.mcx

                deviance    Pearson
observed.stat 2.98596233 2.98198696
pvalue        0.43615976 0.43615976
mcse          0.03240488 0.03240488
```

The default method is the importance sampling of [2]. Using this method, the number of desired simulations `nosim` may not be met in `maxiter` iterations and no warning is issued if this occurs. The returned value is a list storing the results of the Monte Carlo simulation and all of the relevant information necessary to restart the simulation. More information can be obtained with `summary`

```
> summary(resid.mcx)

$conde1
      15       14       10 
294.8800 166.8939 238.3175 

$condv1

  56.78443 -39.34468  36.79749
 -39.34468  51.14382 -39.59345
  36.79749 -39.59345  59.74533

$dens
function (y) 
sum(-lgamma(y + 1))
<environment: namespace:exactLoglinTest>

$dobs
[1] 2.985962 2.981987

$mu.hat
         15          14          10          16          13          12 
  294.87999   166.89392   238.31746 10192.00000    63.22609   261.12001 
         11           9           8           7           6           5 
17819.00000   167.56253   311.10608   501.68254 13677.00000    91.21138 
          4           3           2           1 
  123.77391   370.43747    95.78862 11607.00000 

$n
[1] 16

$n1
[1] 3

$nosim
[1] 100

$s
                   [,1]
(Intercept)       55981
res.1985NE        11929
res.1985S         18986
res.1985W         10888
```

```
res.1980NE        12197
res.1980S         18486
res.1980W         10717
factor(sym.pair)2   187
factor(sym.pair)3   538
factor(sym.pair)4   187
factor(sym.pair)5 13677
factor(sym.pair)6   740
factor(sym.pair)8 17819

$stat
function (y = NULL, mu = NULL, rowlabels = FALSE)
{
    if (rowlabels)
        c("deviance", "Pearson")
    else {
        temp <- y != 0
        c(2 * sum(y[temp] * log(y[temp]/mu[temp])), sum((y -
            mu)^2/mu))
    }
}
<environment: namespace:exactLoglinTest>

$tdf
[1] 3

$x
   (Intercept) res.1985NE res.1985S res.1985W res.1980NE res.1980S res.1980W
15           1          0         1         0          0         0         1
14           1          0         0         0          0         0         1
10           1          0         0         0          0         1         0
16           1          0         0         1          0         0         1
13           1          1         0         0          0         0         1
12           1          0         0         1          0         1         0
11           1          0         1         0          0         1         0
9            1          1         0         0          0         1         0
8            1          0         0         1          0         0         0
7            1          0         1         0          0         0         0
6            1          0         0         0          0         0         0
5            1          1         0         0          0         0         0
4            1          0         0         1          1         0         0
3            1          0         1         0          1         0         0
2            1          0         0         0          1         0         0
1            1          1         0         0          1         0         0
   factor(sym.pair)2 factor(sym.pair)3 factor(sym.pair)4 factor(sym.pair)5
15                 0                 0                 0                 0
14                 0                 0                 0                 0
10                 0                 0                 0                 0
16                 0                 0                 0                 0
13                 0                 0                 1                 0
12                 0                 0                 0                 0
11                 0                 0                 0                 0
9                  0                 1                 0                 0
8                  0                 0                 0                 0
7                  0                 0                 0                 0
6                  0                 0                 0                 1
5                  1                 0                 0                 0
4                  0                 0                 1                 0
```

```
3                  0                  1                  0                  0
2                  1                  0                  0                  0
1                  0                  0                  0                  0
   factor(sym.pair)6 factor(sym.pair)8
15                 0                  0
14                 0                  0
10                 1                  0
16                 0                  0
13                 0                  0
12                 0                  0
11                 0                  1
9                  0                  0
8                  0                  0
7                  1                  0
6                  0                  0
5                  0                  0
4                  0                  0
3                  0                  0
2                  0                  0
1                  0                  0

$x1
   (Intercept) res.1985NE res.1985S res.1985W res.1980NE res.1980S res.1980W
15           1          0         1         0          0         0         1
14           1          0         0         0          0         0         1
10           1          0         0         0          0         1         0
   factor(sym.pair)2 factor(sym.pair)3 factor(sym.pair)4 factor(sym.pair)5
15                 0                 0                 0                 0
14                 0                 0                 0                 0
10                 0                 0                 0                 0
   factor(sym.pair)6 factor(sym.pair)8
15                 0                 0
14                 0                 0
10                 1                 0

$x2invt
   (Intercept) res.1985NE res.1985S res.1985W res.1980NE res.1980S res.1980W
16          -1        0.5         0         1        0.5         0         1
13           1       -0.5         0        -1       -0.5         0         0
12           0        0.0         1         0        0.0         1         0
11           0        0.0         0         0        0.0         0         0
9            0        0.0        -1         0        0.0         0         0
8            2       -1.0        -1        -1       -1.0        -1        -1
7            0        0.0         0         0        0.0         0         0
6            0        0.0         0         0        0.0         0         0
5           -1        1.0         1         1        0.0         0         0
4           -1        0.5         0         1        0.5         0         0
3            0        0.0         1         0        0.0         0         0
2            1       -1.0        -1        -1        0.0         0         0
1            0        0.5         0         0        0.5         0         0
   factor(sym.pair)2 factor(sym.pair)3 factor(sym.pair)4 factor(sym.pair)5
16               0.5               0.5              -0.5                 1
13              -0.5              -0.5               0.5                -1
12               0.0              -1.0               0.0                 0
11               0.0               0.0               0.0                 0
9                0.0               1.0               0.0                 0
8               -1.0               0.0               0.0                -2
7                0.0               0.0               0.0                 0
```

```
6               0.0               0.0               0.0                 1
5               1.0               0.0               0.0                 1
4               0.5               0.5               0.5                 1
3               0.0               0.0               0.0                 0
2               0.0               0.0               0.0                -1
1              -0.5              -0.5              -0.5                 0
   factor(sym.pair)6 factor(sym.pair)8
16                 1                 1
13                -1                -1
12                -1                -2
11                 0                 1
9                  1                 1
8                 -1                 0
7                  1                 0
6                  0                 0
5                  0                 0
4                  1                 1
3                 -1                -1
2                  0                 0
1                  0                 0

$y
   15    14    10    16    13    12    11     9     8     7     6     5     4
  286   176   225 10192    63   270 17819   172   302   515 13677    87   124
    3     2     1
  366   100 11607

$ord
 [1] 15 14 10 16 13 12 11  9  8  7  6  5  4  3  2  1

$glm.fit

Call:  glm(formula = formula, family = poisson, data = data, x = TRUE,
    y = TRUE)

Coefficients:
       (Intercept)           res.1985NE            res.1985S            res.1985W
            1.6281               3.8411               0.5692               4.1120
        res.1980NE            res.1980S            res.1980W   factor(sym.pair)2
            3.8901              -0.1752               3.4892             -0.9561
 factor(sym.pair)3    factor(sym.pair)4    factor(sym.pair)5   factor(sym.pair)6
           -0.1728              -4.8118               7.8953              4.0206
 factor(sym.pair)7    factor(sym.pair)8    factor(sym.pair)9  factor(sym.pair)10
                NA               7.7658                   NA                  NA

Degrees of Freedom: 15 Total (i.e. Null);  3 Residual
Null Deviance:            131000
Residual Deviance: 2.986         AIC: 159.2

$maxiter
[1] 10000

$startiter
[1] 101

$sumdw
[1] 18.64563 18.64563
```

```
$sumdwsq
[1] 19.12212 19.12212

$sumw
[1] 42.74954

$sumwsq
[1] 42.36458

$impconst
[1] -466390.4

$phat
[1] 0.4361598 0.4361598

$mcse
[1] 0.03240488 0.03240488

$perpos
[1] 1

attr(,"class")
[1] "babSummary"
```

The t degrees of freedom refers to degrees of freedom used as a tuning parameter within the algorithm while the `df` refers to the model degrees of freedom. In this case, the Monte Carlo standard error, `mcse`, seems too large. As mentioned previously, `mcexact` , stores the relevant information for restarting the simulation

```
> resid.mcx <- update(resid.mcx, nosim = 10 ^ 4, maxiter = 10 ^ 6)
> resid.mcx

                 deviance     Pearson
observed.stat 2.985962330 2.981986964
pvalue        0.400636040 0.400930887
mcse          0.003196472 0.003196835
```

It is important to note that `update` only resumes the simulation with changes to simulation-specific parameters. It will not allow users to change the model formulation; one must rerun `mcexact` independently to do that.

This example illustrates the point that the underlying algorithms are very efficient when the cell counts are large. Of course, when this is the case, the large sample approximations are nearly identical to the conditional results

```
> pchisq(c(2.986, 2.982), 3, lower.tail = FALSE)

[1] 0.3937887 0.3944088
```

3.2 Pathologists' Tumor Ratings

The following example is interesting in that the large sample results differ drastically from the conditional results. Moreover, the conditional results are less conservative. The data, given in Table 2 can be obtained via

```
> data(pathologist.dat)
```

A uniform association model accounts for the ordinal nature of the ratings by associating ordinal scores with the pathologist's ratings [see 1]. Specifically, we can test a uniform association model against the saturated model with

```
> path.mcx <- mcexact(y ~ factor(A) + factor(B) + I(A * B),
+                     data = pathologist.dat,
+                     nosim = 10 ^ 4,
+                     maxiter = 10 ^ 4)
> summary(path.mcx)
```

```
$conde1
          20           19           18           17           15           14
6.344616e-01 3.692094e+00 1.749242e+01 1.770945e-01 2.998565e-02 1.123799e+00
          13           12           10            9            8           25
3.429048e+01 2.235819e+00 3.025095e-04 7.301670e-02 1.434878e+01 2.335250e+00
          24           23           22
2.110050e+00 1.552251e+00 2.440110e-03

$condv1

  3.992947e-01 -1.512630e-01 -0.2302951722 -1.683236e-02 -1.354645e-02
 -1.512630e-01  1.285627e+00 -1.1337928058 -9.767044e-04 -2.392629e-03
 -2.302952e-01 -1.133793e+00  1.5130688567 -1.460797e-01  1.802070e-02
 -1.683236e-02 -9.767044e-04 -0.1460797229  1.643946e-01 -1.958202e-03
 -1.354645e-02 -2.392629e-03  0.0180207042 -1.958202e-03  2.886289e-02
 -8.111453e-02 -5.988614e-01  0.6966776416 -1.548698e-02 -1.038945e-02
  1.894272e-01  4.925053e-01 -0.8031364378  1.162690e-01 -2.453056e-03
 -5.915681e-02  7.589214e-02  0.0632142608 -7.761053e-02 -1.042780e-02
 -1.685691e-04 -3.183465e-06  0.0002077854 -3.388633e-05 -1.587280e-05
 -9.327963e-03 -3.675231e-02  0.0488399809 -2.572783e-03 -1.225006e-03
 -2.138232e-03  6.014399e-02 -0.1636048047  1.032278e-01 -1.557045e-03
 -3.855792e-01  1.536587e-01  0.2120661394  1.882458e-02 -1.530052e-02
  2.419696e-01 -6.495954e-01  0.3874878075  1.913509e-02  1.404431e-02
  1.437071e-01  4.949123e-01 -0.5989177210 -3.768255e-02  1.302339e-03
 -9.565519e-05  1.019615e-03 -0.0006350826 -2.754771e-04 -4.573679e-05

 -0.0811145281  0.1894272278 -0.0591568137 -1.685691e-04 -9.327963e-03
 -0.5988613532  0.4925052899  0.0758921360 -3.183465e-06 -3.675231e-02
  0.6966776416 -0.8031364378  0.0632142608  2.077854e-04  4.883998e-02
 -0.0154869817  0.1162689632 -0.0776105282 -3.388633e-05 -2.572783e-03
 -0.0103894478 -0.0024530562 -0.0104277955 -1.587280e-05 -1.225006e-03
  0.7869316765 -0.5710434774 -0.1405318761 -1.493944e-04 -2.672783e-02
 -0.5710434774  1.7623815394 -1.1342630814  3.904828e-04  4.405264e-02
 -0.1405318761 -1.1342630814  1.4174614975 -1.347161e-04 -8.302405e-03
 -0.0001493944  0.0003904828 -0.0001347161  3.022609e-04 -2.146740e-05
 -0.0267278342  0.0440526416 -0.0083024047 -2.146740e-05  6.998117e-02
 -0.0268356441 -1.4105516405  1.2509438439 -2.596956e-04 -6.002243e-02
  0.0916538934 -0.1873660854  0.0697198468 -1.178181e-04  1.057451e-02
 -0.1607465057  0.0332474968  0.0732946813  1.746923e-04 -6.426580e-03
  0.0694956953  0.1515129370 -0.1414406076 -5.596201e-05 -4.068555e-03
 -0.0003991607  0.0025894889 -0.0015662853 -9.049401e-07 -7.872439e-05

 -0.0021382316 -0.3855791564  0.2419696249  1.437071e-01 -9.565519e-05
  0.0601439882  0.1536587142 -0.6495953877  4.949123e-01  1.019615e-03
 -0.1636048047  0.2120661394  0.3874878075 -5.989177e-01 -6.350826e-04
  0.1032278276  0.0188245778  0.0191350909 -3.768255e-02 -2.754771e-04
 -0.0015570454 -0.0153005192  0.0140443088  1.302339e-03 -4.573679e-05
 -0.0268356441  0.0916538934 -0.1607465057  6.949570e-02 -3.991607e-04
 -1.4105516405 -0.1873660854  0.0332474968  1.515129e-01  2.589489e-03
  1.2509438439  0.0697198468  0.0732946813 -1.414406e-01 -1.566285e-03
 -0.0002596956 -0.0001178181  0.0001746923 -5.596201e-05 -9.049401e-07
 -0.0600224324  0.0105745120 -0.0064265799 -4.068555e-03 -7.872439e-05
  2.4262895610  0.0039554188  0.0273610511 -3.269460e-02  1.373295e-03
```

```
  0.0039554188  0.4009976184 -0.2561891403 -1.449531e-01  1.423006e-04
  0.0273610511 -0.2561891403  0.8167173280 -5.599893e-01 -5.387153e-04
 -0.0326945978 -0.1449530621 -0.5599893053  7.069878e-01 -2.034924e-03
  0.0013732947  0.0001423006 -0.0005387153 -2.034924e-03  2.431391e-03

$dens
function (y)
sum(-lgamma(y + 1))
<environment: namespace:exactLoglinTest>

$dobs
[1] 16.21453 14.72928

$mu.hat
          20           19           18           17           15           14
6.344616e-01 3.692094e+00 1.749242e+01 1.770945e-01 2.998565e-02 1.123799e+00
          13           12           10            9            8           25
3.429048e+01 2.235819e+00 3.025095e-04 7.301670e-02 1.434878e+01 2.335250e+00
          24           23           22           21           16           11
2.110050e+00 1.552251e+00 2.440110e-03 8.417609e-06 3.934524e-03 3.199129e-01
           7            6            5            4            3            2
6.025400e+00 5.552502e+00 6.689408e-07 1.039869e-03 1.316071e+00 3.559247e+00
           1
2.112364e+01

$n
[1] 25

$n1
[1] 15

$nosim
[1] 10000

$s
            [,1]
(Intercept)  118
factor(A)2    12
factor(A)3    69
factor(A)4     7
factor(A)5     3
factor(B)2    26
factor(B)3    38
factor(B)4    22
factor(B)5     6
I(A * B)     898

$stat
function (y = NULL, mu = NULL, rowlabels = FALSE)
{
    if (rowlabels)
        c("deviance", "Pearson")
    else {
        temp <- y != 0
        c(2 * sum(y[temp] * log(y[temp]/mu[temp])), sum((y -
            mu)^2/mu))
    }
}
```

```
<environment: namespace:exactLoglinTest>

$tdf
[1] 3

$x
   (Intercept) factor(A)2 factor(A)3 factor(A)4 factor(A)5 factor(B)2
20           1          0          0          0          1          0
19           1          0          0          1          0          0
18           1          0          1          0          0          0
17           1          1          0          0          0          0
15           1          0          0          0          1          0
14           1          0          0          1          0          0
13           1          0          1          0          0          0
12           1          1          0          0          0          0
10           1          0          0          0          1          1
9            1          0          0          1          0          1
8            1          0          1          0          0          1
25           1          0          0          0          1          0
24           1          0          0          1          0          0
23           1          0          1          0          0          0
22           1          1          0          0          0          0
21           1          0          0          0          0          0
16           1          0          0          0          0          0
11           1          0          0          0          0          0
7            1          1          0          0          0          1
6            1          0          0          0          0          1
5            1          0          0          0          1          0
4            1          0          0          1          0          0
3            1          0          1          0          0          0
2            1          1          0          0          0          0
1            1          0          0          0          0          0
   factor(B)3 factor(B)4 factor(B)5 I(A * B)
20          0          1          0       20
19          0          1          0       16
18          0          1          0       12
17          0          1          0        8
15          1          0          0       15
14          1          0          0       12
13          1          0          0        9
12          1          0          0        6
10          0          0          0       10
9           0          0          0        8
8           0          0          0        6
25          0          0          1       25
24          0          0          1       20
23          0          0          1       15
22          0          0          1       10
21          0          0          1        5
16          0          1          0        4
11          1          0          0        3
7           0          0          0        4
6           0          0          0        2
5           0          0          0        5
4           0          0          0        4
3           0          0          0        3
2           0          0          0        2
1           0          0          0        1
```

```
$x1
   (Intercept) factor(A)2 factor(A)3 factor(A)4 factor(A)5 factor(B)2
20           1          0          0          0          1          0
19           1          0          0          1          0          0
18           1          0          1          0          0          0
17           1          1          0          0          0          0
15           1          0          0          0          1          0
14           1          0          0          1          0          0
13           1          0          1          0          0          0
12           1          1          0          0          0          0
10           1          0          0          0          1          1
9            1          0          0          1          0          1
8            1          0          1          0          0          1
25           1          0          0          0          1          0
24           1          0          0          1          0          0
23           1          0          1          0          0          0
22           1          1          0          0          0          0
   factor(B)3 factor(B)4 factor(B)5 I(A * B)
20          0          1          0       20
19          0          1          0       16
18          0          1          0       12
17          0          1          0        8
15          1          0          0       15
14          1          0          0       12
13          1          0          0        9
12          1          0          0        6
10          0          0          0       10
9           0          0          0        8
8           0          0          0        6
25          0          0          1       25
24          0          0          1       20
23          0          0          1       15
22          0          0          1       10

$x2invt
   (Intercept) factor(A)2 factor(A)3 factor(A)4 factor(A)5 factor(B)2
21           0          0          0          0          0          0
16           0          0          0          0          0          0
11           0          0          0          0          0          0
7           -1         -1         -2         -3         -4         -1
6            1          1          2          3          4          2
5            0          0          0          0          1          0
4            0          0          0          1          0          0
3            0          0          1          0          0          0
2            1          2          2          3          4          1
1            0         -2         -3         -4         -5         -2
   factor(B)3 factor(B)4 factor(B)5 I(A * B)
21          0          0          1        0
16          0          1          0        0
11          1          0          0        0
7          -2         -3         -4        1
6           2          3          4       -1
5           0          0          0        0
4           0          0          0        0
3           0          0          0        0
2           2          3          4       -1
1          -3         -4         -5        1
```

```
$y
20 19 18 17 15 14 13 12 10  9  8 25 24 23 22 21 16 11  7  6  5  4  3  2  1
 0  7 14  1  0  0 36  2  0  0 14  3  0  3  0  0  0  0  7  5  0  0  2  2 22

$ord
 [1] 20 19 18 17 15 14 13 12 10  9  8 25 24 23 22 21 16 11  7  6  5  4  3  2  1

$glm.fit

Call:  glm(formula = formula, family = poisson, data = data, x = TRUE,
    y = TRUE)

Coefficients:
(Intercept)   factor(A)2   factor(A)3   factor(A)4   factor(A)5   factor(B)2
      1.188       -3.643       -6.501      -15.507      -24.718       -3.199
 factor(B)3   factor(B)4   factor(B)5     I(A * B)
     -7.915      -14.176      -22.186        1.863

Degrees of Freedom: 24 Total (i.e. Null);  15 Residual
Null Deviance:           267.7
Residual Deviance: 16.21         AIC: 82.48

$maxiter
[1] 10000

$startiter
[1] 1

$sumdw
[1] 0

$sumdwsq
[1] 0

$sumw
[1] 0

$sumwsq
[1] 0

$phat
[1] NaN

$mcse
[1] NaN

$perpos
[1] 0

attr(,"class")
[1] "babSummary"
```

The previous code chunk takes about 1 minute on my laptop. It is worth comparing these results to the asymptotic Chi-squared results

```
> pchisq(c(16.214, 14.729), 15, lower.tail = FALSE)
```

```
[1] 0.3679734 0.4711083
```

3.3 Alligator Food Choice Data Using MCMC

In this example we illustrate the algorithm from[3] using the data and Poisson log-linear model from Table 3. The alligator data is a good choice for MCMC as the percent of valid tables generated using `method = "bab"` is very small, less than 1% of the tables simulated. It is often the case that the MCMC algorithm will be preferable when the table is large and/or sparse. Of course, using MCMC introduces further complications in reliably running and using the output of the algorithm.

The algorithm from [3] uses local moves to reduce the number of tables with negative entries that the chain produces. You can specify this method by using `method = "cab"`. The parameter `p` represents the average proportion of table entries left fixed. So a chain with `p=.9` will leave most of the table entries fixed from one iteration to the next. A high value of `p` will result in a high proportion of valid (non-negative) simulated tables. Too large of a value of `p` causes the chain to mix slowly because the tables will be very similar from one iteration to the next. However, it is sometimes the case that a small value of `p` will produce too many tables with negative entries. Hence the Metropolis/Hastings/Green algorithm will stay at the current table for long periods and again result in a slowly mixing chain. It is also worth mentioning that for large values of `p` the algorithm is theoretically irreducible, but may not be practically irreducible. Therefore, it is advisable to both tinker with the chain some and make final runs using multiple values of `p`.

The program allows for the option to save the chain goodness of fit statistics, so that some initial tinkering can be performed. This is specified with the `savechain = TRUE` option. If using impartance sampling, `method = "bab"`, then `savechain` saves both the statistic values and the importance weights on the log scale.

```
> data(alligator.dat)
> alligator.mcx <- mcexact(y ~ (lake + gender + size) * food + lake * gender * size,
+                          data = alligator.dat,
+                          nosim = 10 ^ 3,
+                          method = "cab",
+                          savechain = TRUE,
+                          batchsize = 100,
+                          p = .4)
> summary(alligator.mcx)

$conde1
        75         74         73         72         70         69         68 
1.31600087 0.48346172 0.18073692 6.51660315 1.13800295 0.81453108 0.25072319 
        67         60         59         58         57         55         54 
1.60473559 0.18105409 0.13394981 0.26096632 0.21080481 1.97444067 0.52630855 
        53         52         50         49         48         47         40 
1.21928164 6.54106633 5.63440188 2.92618991 5.58172191 5.31553870 0.13185304 
        39         38         37         35         34         33         32 
0.07554443 0.51708045 0.55718074 0.79011836 0.16310491 1.32752883 9.50014868 
        30         29         28         27         20         19         18 
1.78879234 0.71943767 4.82138891 6.12481782 1.78671996 1.50458067 0.82331031 
        17         80         79         78         77 
0.24350239 1.03386967 1.05416634 0.33141537 1.79927933 

$condv1

  0.8158428899 -4.024685e-02 -1.553110e-02 -0.349190606 -0.1432926821
 -0.0402468520  3.425437e-01 -6.423501e-03 -0.132000575  0.0112669026
 -0.0155310983 -6.423501e-03  1.418543e-01 -0.056732328  0.0054324960
 -0.3491906063 -1.320006e-01 -5.673233e-02  1.944240192  0.0541580685
 -0.1432926821  1.126690e-02  5.432496e-03  0.054158069  0.7401576216
  0.0126479801 -6.698040e-02  2.194671e-03  0.021894398 -0.0553905322
  0.0051999933  1.668882e-03 -4.001101e-02  0.018815577 -0.0196291578
  0.0427174386  1.701258e-02  1.118021e-02 -0.206501050 -0.1424549232
 -0.0107478602  9.515460e-04  3.247559e-04  0.007024218 -0.0004989307
  0.0009304963 -5.024699e-03  2.000605e-04  0.004639496 -0.0013730257
```

0.0020305479	1.354304e-03	-1.518614e-03	-0.002545615	-0.0007124931
0.0045797255	1.497876e-03	5.267360e-04	-0.023797868	0.0028035524
-0.1980993866	6.075710e-03	5.407843e-03	0.184127587	0.1496365772
0.0078073066	-6.873110e-02	1.977967e-03	0.056534616	-0.0136143474
0.0102063142	2.962669e-03	-3.255049e-02	0.042264127	-0.0142522175
0.1366844470	4.568478e-02	1.659121e-02	-0.517973431	-0.0627665744
0.1993899950	-1.367119e-02	-9.238751e-03	-0.110748159	-0.2337196537
-0.0140866155	9.559156e-02	-3.555370e-03	-0.054177923	0.0183523013
-0.0188685469	-7.591748e-03	4.410161e-02	-0.012509049	0.0190353775
-0.0920784386	-4.329516e-02	-1.459001e-02	0.361106082	0.0891353606
-0.0080752925	1.324869e-04	2.071072e-04	0.004468277	-0.0004131863
-0.0003415250	-3.020569e-03	9.053576e-05	0.002609869	-0.0006105333
-0.0016771963	2.878589e-05	-2.867681e-03	-0.003494169	-0.0009174062
0.0074260713	2.258904e-03	1.114944e-03	-0.050775447	0.0030639985
-0.0849167457	-2.050967e-03	1.872812e-03	0.068924106	0.0719471826
-0.0003087532	-2.193521e-02	5.410853e-04	0.017323964	-0.0013128616
-0.0105051563	-5.477681e-03	-3.527090e-02	0.048514814	0.0065273111
0.0687785132	2.385286e-02	1.798516e-02	-0.556154021	-0.0165875622
0.0895235521	1.612037e-03	-2.541660e-03	-0.056763707	-0.0857398927
0.0001866741	2.717585e-02	-7.686077e-04	-0.018541918	0.0017031657
0.0102985658	5.228012e-03	4.154175e-02	-0.030921208	-0.0082281417
-0.0562173667	-2.215968e-02	-1.715349e-02	0.505456857	0.0177359687
-0.0052008245	-7.118337e-03	-2.684526e-04	0.004642330	-0.0863596513
-0.0037068574	3.700360e-02	-1.540702e-03	-0.015858387	0.0046324516
0.0008862818	-2.389580e-03	6.526212e-03	-0.009244601	0.0021926126
0.0022527803	-1.239194e-03	2.709523e-04	-0.010948376	0.0042210349
-0.2410652748	1.935201e-02	6.659537e-03	0.082691382	-0.2126596733
0.0155880999	-1.594698e-01	3.834239e-03	0.056894076	0.0217753875
0.0064294250	4.752970e-03	-5.802036e-02	0.013744369	0.0068771464
0.0956599413	4.855803e-02	1.879302e-02	-0.603778526	0.0232833502

0.0126479801	0.0051999933	0.0427174386	-0.0107478602	9.304963e-04
-0.0669804029	0.0016688823	0.0170125805	0.0009515460	-5.024699e-03
0.0021946710	-0.0400110088	0.0111802099	0.0003247559	2.000605e-04
0.0218943983	0.0188155772	-0.2065010499	0.0070242177	4.639496e-03
-0.0553905322	-0.0196291578	-0.1424549232	-0.0004989307	-1.373026e-03
0.4699468498	-0.0123104240	-0.0821830095	-0.0022970012	1.151175e-02
-0.0123104240	0.1777975938	-0.0324663406	-0.0010022218	-1.121173e-03
-0.0821830095	-0.0324663406	1.0664840652	0.0033920589	-2.549591e-04
-0.0022970012	-0.0010022218	0.0033920589	0.1407906696	-2.286272e-02
0.0115117504	-0.0011211725	-0.0002549591	-0.0228627205	1.069917e-01
-0.0034299062	0.0045169851	0.0030080740	-0.0420030278	-3.002572e-02
-0.0020880710	-0.0008521845	-0.0088988247	-0.0374788962	-2.663943e-02
-0.0133854403	-0.0085222713	-0.0618389683	-0.0462481557	5.159438e-03
0.0922495421	-0.0037306178	-0.0296581971	0.0056789335	-2.767602e-02
-0.0064150666	0.0406810936	-0.0145578408	0.0147496924	9.484085e-03
-0.0410168313	-0.0138090712	0.2110348280	0.0191931125	1.080015e-02
0.0189441710	0.0105416530	0.0975417670	-0.0913671295	1.576411e-02
-0.1193893942	0.0050729944	0.0404832621	0.0160377018	-7.596599e-02
0.0107283400	-0.0487382052	0.0166439096	0.0322719142	2.399055e-02
0.0403061795	0.0114673746	-0.3465284760	0.0183331781	1.530134e-02
-0.0003138193	-0.0006671666	0.0027339907	-0.0032659214	-1.864083e-05
0.0067659440	-0.0005726100	0.0000333626	0.0001957909	-2.969134e-03
-0.0009585807	0.0081423905	0.0068915377	0.0059071292	4.404260e-03
-0.0011963837	-0.0020195148	-0.0188973131	-0.0003265022	8.651739e-05
0.0010770017	-0.0028448799	-0.0308240999	-0.0048706854	3.978516e-04
0.0288261683	-0.0009880553	-0.0103895660	0.0006047980	-3.078922e-03
0.0051290539	0.0425381838	-0.0263261141	0.0054666140	4.126056e-03

-0.0064310939	-0.0138033770	0.2932539021	-0.0020637710	-6.395668e-04
-0.0010315134	0.0035525044	0.0344151709	0.0029796051	-3.912278e-04
-0.0369965501	0.0015564646	0.0115881677	-0.0007040545	4.921347e-03
-0.0049868857	-0.0513676031	0.0227106812	-0.0075922700	-6.167792e-03
0.0067072418	0.0135653298	-0.3489096838	0.0056361985	1.550741e-03
0.0007924853	-0.0005548973	-0.0040734208	-0.0302563131	7.942514e-03
-0.0072037068	-0.0015637203	-0.0052136601	0.0075861027	-3.075352e-02
-0.0007833930	0.0003018796	-0.0123086040	0.0104799047	8.203715e-03
0.0008049234	-0.0001603018	-0.0268160003	0.0014269821	1.287728e-03
0.0244478758	0.0079935766	0.0250158238	-0.0233704312	4.201409e-03
-0.2191987175	0.0075580068	0.0330149286	0.0052909104	-2.827038e-02
0.0080217961	-0.0770320848	0.0089504876	0.0043979191	3.485831e-03
0.0353895251	0.0107155495	-0.2959519775	0.0087738144	8.881541e-03

0.0020305479	0.0045797255	-1.980994e-01	0.0078073066	0.0102063142
0.0013543036	0.0014978761	6.075710e-03	-0.0687311011	0.0029626691
-0.0015186139	0.0005267360	5.407843e-03	0.0019779671	-0.0325504915
-0.0025456149	-0.0237978676	1.841276e-01	0.0565346157	0.0422641267
-0.0007124931	0.0028035524	1.496366e-01	-0.0136143474	-0.0142522175
-0.0034299062	-0.0020880710	-1.338544e-02	0.0922495421	-0.0064150666
0.0045169851	-0.0008521845	-8.522271e-03	-0.0037306178	0.0406810936
0.0030080740	-0.0088988247	-6.183897e-02	-0.0296581971	-0.0145578408
-0.0420030278	-0.0374788962	-4.624816e-02	0.0056789335	0.0147496924
-0.0300257241	-0.0266394265	5.159438e-03	-0.0276760239	0.0094840853
0.1718084526	-0.0494526993	1.537225e-02	0.0077831025	-0.0607035817
-0.0494526993	0.1585909826	1.904074e-02	0.0088418845	0.0215914319
0.0153722508	0.0190407350	9.145374e-01	-0.0668014199	-0.1176860809
0.0077831025	0.0088418845	-6.680142e-02	0.3215796804	-0.0370742254
-0.0607035817	0.0215914319	-1.176861e-01	-0.0370742254	0.6085968528
0.0278206701	-0.0663303568	-4.807257e-01	-0.1381671907	-0.3107271639
0.0312802546	0.0191538978	-4.635803e-01	0.0495094373	0.0711436767
0.0225639360	0.0172948393	4.854106e-02	-0.2137723814	0.0235998902
-0.1307969846	0.0344243907	6.743091e-02	0.0239238152	-0.3683467327
0.0343035956	-0.0917555929	1.713166e-01	0.0650401894	0.1565394565
0.0026361368	0.0005520898	-1.233563e-02	0.0005286044	0.0050599043
0.0018306898	0.0005905866	8.828207e-05	-0.0068328922	0.0029809887
-0.0269601665	0.0085506529	8.740781e-03	0.0027425457	-0.0521044107
0.0100231951	-0.0099735877	7.719214e-03	0.0028308240	0.0176508644
0.0029767060	0.0003022203	-1.638771e-01	0.0012566405	0.0285369530
0.0011702958	0.0004960990	5.296996e-04	-0.0342591332	0.0060393209
-0.0216164891	0.0043688967	3.849739e-02	0.0104258340	-0.3055965349
0.0078809809	-0.0076458349	1.668007e-01	0.0455489802	0.1953557053
-0.0041195445	0.0022932975	1.764678e-01	-0.0027174578	-0.0373721063
-0.0026895340	-0.0005066030	-1.233410e-03	0.0432829004	-0.0098612167
0.0300545227	-0.0053338500	-5.242853e-02	-0.0153484186	0.3636482428
-0.0049405014	-0.0047913569	-1.543108e-01	-0.0464030121	-0.1943050371
0.0110257996	0.0031221958	2.323909e-02	-0.0095204502	-0.0065092455
0.0104701647	0.0049723134	-7.438347e-03	0.0245082784	-0.0046016531
-0.0395963274	0.0089755435	-4.592797e-03	-0.0046054720	0.0056462534
0.0015435686	-0.0054937043	-1.777478e-03	-0.0020498769	0.0005614384
0.0040343291	0.0075599150	-4.487586e-02	0.0031110794	0.0013684638
0.0063176806	0.0083299532	3.964997e-03	-0.0294235374	-0.0001360425
-0.0192634126	0.0053647356	3.678396e-03	0.0010115479	-0.0196713847
0.0052224958	-0.0372803569	3.207155e-02	0.0065124972	0.0160335874

0.136684447	0.199389995	-0.014086615	-0.018868547	-0.092078439	-8.075292e-03
0.045684776	-0.013671191	0.095591557	-0.007591748	-0.043295161	1.324869e-04
0.016591212	-0.009238751	-0.003555370	0.044101605	-0.014590006	2.071072e-04

-0.517973431	-0.110748159	-0.054177923	-0.012509049	0.361106082	4.468277e-03
-0.062766574	-0.233719654	0.018352301	0.019035378	0.089135361	-4.131863e-04
-0.041016831	0.018944171	-0.119389394	0.010728340	0.040306179	-3.138193e-04
-0.013809071	0.010541653	0.005072994	-0.048738205	0.011467375	-6.671666e-04
0.211034828	0.097541767	0.040483262	0.016643910	-0.346528476	2.733991e-03
0.019193113	-0.091367130	0.016037702	0.032271914	0.018333178	-3.265921e-03
0.010800152	0.015764113	-0.075965988	0.023990547	0.015301337	-1.864083e-05
0.027820670	0.031280255	0.022563936	-0.130796985	0.034303596	2.636137e-03
-0.066330357	0.019153898	0.017294839	0.034424391	-0.091755593	5.520898e-04
-0.480725741	-0.463580280	0.048541056	0.067430908	0.171316639	-1.233563e-02
-0.138167191	0.049509437	-0.213772381	0.023923815	0.065040189	5.286044e-04
-0.310727164	0.071143677	0.023599890	-0.368346733	0.156539457	5.059904e-03
1.503918926	0.164600551	0.071204797	0.167381313	-0.677100460	4.695245e-03
0.164600551	1.252463499	-0.118296805	-0.192684682	-0.441662591	-5.307306e-03
0.071204797	-0.118296805	0.518281932	-0.073402417	-0.148750150	-3.581620e-04
0.167381313	-0.192684682	-0.073402417	0.893541760	-0.327086254	-6.944460e-04
-0.677100460	-0.441662591	-0.148750150	-0.327086254	1.656756915	6.742172e-03
0.004695245	-0.005307306	-0.000358162	-0.000694446	0.006742172	1.153530e-01
0.002819630	-0.001191993	0.004659606	-0.001781557	0.000383091	-4.778090e-03
0.026668139	-0.004665924	-0.005117530	-0.000641135	0.016926474	-2.800300e-02
-0.037818899	0.013909448	0.002186016	0.009837101	-0.043887524	-3.571017e-02
0.113147871	0.200781837	-0.003401055	-0.041407888	-0.114822249	-3.614297e-02
0.024669704	-0.002426615	0.046490898	-0.009831266	-0.027328022	8.970305e-04
0.195411776	-0.061092053	-0.019378781	0.391373453	-0.202173393	9.953006e-03
-0.629165812	-0.154036446	-0.045847810	-0.219482742	0.674651768	1.532328e-02
-0.107238921	-0.240634075	0.004451623	0.049488468	0.133139452	-7.117190e-02
-0.027149146	0.003530685	-0.057050324	0.012666865	0.030896774	3.613956e-03
-0.211227649	0.074212969	0.025406638	-0.456000489	0.219987481	1.867529e-02
0.616341264	0.182605788	0.053147674	0.233946502	-0.789375048	1.723452e-02
0.008999189	-0.260972246	0.033519295	0.043431156	0.063640040	-2.159670e-02
-0.006044357	0.039851777	-0.102017679	0.013767105	0.011765222	2.062231e-03
0.009181862	0.041829423	0.010906384	-0.121554621	0.019396283	6.943866e-03
0.005061192	0.022461373	0.007449918	0.006761405	-0.060423423	1.216517e-03
0.031157969	-0.120722284	0.002607279	0.006424957	0.067273836	-1.716722e-02
0.015218451	-0.003257817	-0.014753543	-0.005901417	0.022821510	4.771677e-04
0.009955762	0.003966056	-0.001104606	-0.026059722	0.014212452	2.790378e-03
-0.085044651	0.090452799	0.021723403	0.022042148	-0.280304823	7.033318e-03

-3.415250e-04	-1.677196e-03	7.426071e-03	-0.0849167457	-0.0003087532
-3.020569e-03	2.878589e-05	2.258904e-03	-0.0020509665	-0.0219352061
9.053576e-05	-2.867681e-03	1.114944e-03	0.0018728124	0.0005410853
2.609869e-03	-3.494169e-03	-5.077545e-02	0.0689241062	0.0173239643
-6.105333e-04	-9.174062e-04	3.063999e-03	0.0719471826	-0.0013128616
6.765944e-03	-9.585807e-04	-1.196384e-03	0.0010770017	0.0288261683
-5.726100e-04	8.142390e-03	-2.019515e-03	-0.0028448799	-0.0009880553
3.336260e-05	6.891538e-03	-1.889731e-02	-0.0308240999	-0.0103895660
1.957909e-04	5.907129e-03	-3.265022e-04	-0.0048706854	0.0006047980
-2.969134e-03	4.404260e-03	8.651739e-05	0.0003978516	-0.0030789220
1.830690e-03	-2.696017e-02	1.002320e-02	0.0029767060	0.0011702958
5.905866e-04	8.550653e-03	-9.973588e-03	0.0003022203	0.0004960990
8.828207e-05	8.740781e-03	7.719214e-03	-0.1638771363	0.0005296996
-6.832892e-03	2.742546e-03	2.830824e-03	0.0012566405	-0.0342591332
2.980989e-03	-5.210441e-02	1.765086e-02	0.0285369530	0.0060393209
2.819630e-03	2.666814e-02	-3.781890e-02	0.1131478711	0.0246697038
-1.191993e-03	-4.665924e-03	1.390945e-02	0.2007818365	-0.0024266148
4.659606e-03	-5.117530e-03	2.186016e-03	-0.0034010546	0.0464908983
-1.781557e-03	-6.411350e-04	9.837101e-03	-0.0414078879	-0.0098312657
3.830910e-04	1.692647e-02	-4.388752e-02	-0.1148222493	-0.0273280220

-4.778090e-03	-2.800300e-02	-3.571017e-02	-0.0361429717	0.0008970305
6.557929e-02	-1.547777e-02	-1.943793e-02	0.0005091210	-0.0138327064
-1.547777e-02	3.094219e-01	-1.160108e-01	0.0086907879	0.0031034165
-1.943793e-02	-1.160108e-01	3.637149e-01	0.0187296685	0.0059130901
5.091210e-04	8.690788e-03	1.872967e-02	0.4714766856	-0.0109139362
-1.383271e-02	3.103417e-03	5.913090e-03	-0.0109139362	0.1235097858
4.534662e-03	-1.187391e-01	5.381296e-02	-0.0587627701	-0.0147005706
6.292025e-03	6.539765e-02	-1.602430e-01	-0.2241175628	-0.0523788765
3.786937e-03	1.842151e-02	1.547241e-02	-0.3572279036	0.0102151810
-4.928482e-02	1.195804e-02	1.295517e-02	0.0104191045	-0.1018399168
1.147858e-02	-2.038041e-01	6.850122e-02	0.0475284349	0.0117377042
1.244720e-02	6.458735e-02	-2.018523e-01	0.1375360944	0.0355025428
2.937507e-03	1.182648e-02	-5.204105e-03	0.0299827004	-0.0025510328
-1.714947e-02	8.106909e-03	-1.073269e-03	-0.0028256827	0.0123298802
4.247656e-03	-6.979366e-02	2.168306e-02	-0.0045674867	-0.0019578378
8.056672e-04	3.830464e-03	-1.099984e-02	-0.0033860727	-0.0012352890
9.786888e-04	-9.057320e-04	6.515808e-03	-0.0042817885	0.0009671716
-1.608835e-02	-7.631369e-05	6.653857e-03	0.0006616462	-0.0054453700
1.736801e-03	-3.454768e-02	1.208764e-02	0.0008786607	0.0002064774
5.290730e-03	1.466011e-02	-7.727823e-02	-0.0017998888	-0.0011907025

-1.050516e-02	0.0687785132	0.0895235521	0.0001866741	0.0102985658
-5.477681e-03	0.0238528552	0.0016120372	0.0271758486	0.0052280118
-3.527090e-02	0.0179851580	-0.0025416599	-0.0007686077	0.0415417484
4.851481e-02	-0.5561540208	-0.0567637067	-0.0185419176	-0.0309212077
6.527311e-03	-0.0165875622	-0.0857398927	0.0017031657	-0.0082281417
5.129054e-03	-0.0064310939	-0.0010315134	-0.0369965501	-0.0049868857
4.253818e-02	-0.0138033770	0.0035525044	0.0015564646	-0.0513676031
-2.632611e-02	0.2932539021	0.0344151709	0.0115881677	0.0227106812
5.466614e-03	-0.0020637710	0.0029796051	-0.0007040545	-0.0075922700
4.126056e-03	-0.0006395668	-0.0003912278	0.0049213471	-0.0061677919
-2.161649e-02	0.0078809809	-0.0041195445	-0.0026895340	0.0300545227
4.368897e-03	-0.0076458349	0.0022932975	-0.0005066030	-0.0053338500
3.849739e-02	0.1668007443	0.1764677727	-0.0012334100	-0.0524285293
1.042583e-02	0.0455489802	-0.0027174578	0.0432829004	-0.0153484186
-3.055965e-01	0.1953557053	-0.0373721063	-0.0098612167	0.3636482428
1.954118e-01	-0.6291658123	-0.1072389208	-0.0271491461	-0.2112276487
-6.109205e-02	-0.1540364458	-0.2406340750	0.0035306845	0.0742129688
-1.937878e-02	-0.0458478104	0.0044516231	-0.0570503236	0.0254066378
3.913735e-01	-0.2194827417	0.0494884681	0.0126668650	-0.4560004887
-2.021734e-01	0.6746517683	0.1331394519	0.0308967739	0.2199874812
9.953006e-03	0.0153232783	-0.0711719005	0.0036139558	0.0186752932
4.534662e-03	0.0062920249	0.0037869371	-0.0492848219	0.0114785772
-1.187391e-01	0.0653976471	0.0184215136	0.0119580358	-0.2038041110
5.381296e-02	-0.1602430177	0.0154724070	0.0129551724	0.0685012160
-5.876277e-02	-0.2241175628	-0.3572279036	0.0104191045	0.0475284349
-1.470057e-02	-0.0523788765	0.0102151810	-0.1018399168	0.0117377042
6.151768e-01	-0.3176066871	0.0471329342	0.0104320376	-0.4087806088
-3.176067e-01	1.4890769536	0.1447732782	0.0381692362	0.1813445124
4.713293e-02	0.1447732782	0.5789378210	-0.0164284063	-0.0825796375
1.043204e-02	0.0381692362	-0.0164284063	0.1800463720	-0.0260685127
-4.087806e-01	0.1813445124	-0.0825796375	-0.0260685127	0.8007708560
1.820352e-01	-0.8804246893	-0.2172358612	-0.0590512078	-0.3263102299
-2.466519e-03	0.0007957924	-0.0557748530	0.0020604199	0.0010699294
-2.344774e-03	0.0001208034	0.0046388545	-0.0075583664	-0.0009470604
3.073025e-02	-0.0046902190	0.0043755897	-0.0007547275	-0.0230596792
-3.685949e-03	0.0224426879	0.0063491494	0.0013060650	0.0044479716
3.107155e-03	-0.0092669100	-0.0130457711	-0.0016307621	-0.0038912371

-4.624942e-05	-0.0077262545	-0.0022657895	0.0129212507	-0.0024021863
-8.966067e-03	0.0013718780	-0.0022946661	-0.0016326178	0.0167548823
-2.541220e-03	0.0165986880	0.0194980582	0.0008039441	0.0108603110

-0.056217367	-0.0052008245	-0.0037068574	0.0008862818	0.0022527803
-0.022159678	-0.0071183370	0.0370036022	-0.0023895801	-0.0012391935
-0.017153495	-0.0002684526	-0.0015407023	0.0065262124	0.0002709523
0.505456857	0.0046423303	-0.0158583875	-0.0092446014	-0.0109483758
0.017735969	-0.0863596513	0.0046324516	0.0021926126	0.0042210349
0.006707242	0.0007924853	-0.0072037068	-0.0007833930	0.0008049234
0.013565330	-0.0005548973	-0.0015637203	0.0003018796	-0.0001603018
-0.348909684	-0.0040734208	-0.0052136601	-0.0123086040	-0.0268160003
0.005636198	-0.0302563131	0.0075861027	0.0104799047	0.0014269821
0.001550741	0.0079425142	-0.0307535162	0.0082037154	0.0012877282
-0.004940501	0.0110257996	0.0104701647	-0.0395963274	0.0015435686
-0.004791357	0.0031221958	0.0049723134	0.0089755435	-0.0054937043
-0.154310830	0.0232390947	-0.0074383469	-0.0045927974	-0.0017774784
-0.046403012	-0.0095204502	0.0245082784	-0.0046054720	-0.0020498769
-0.194305037	-0.0065092455	-0.0046016531	0.0056462534	0.0005614384
0.616341264	0.0089991893	-0.0060443567	0.0091818615	0.0050611920
0.182605788	-0.2609722460	0.0398517765	0.0418294234	0.0224613731
0.053147674	0.0335192947	-0.1020176794	0.0109063844	0.0074499184
0.233946502	0.0434311561	0.0137671048	-0.1215546210	0.0067614051
-0.789375048	0.0636400397	0.0117652223	0.0193962830	-0.0604234231
0.017234520	-0.0215967030	0.0020622308	0.0069438657	0.0012165166
0.012447199	0.0029375071	-0.0171494739	0.0042476556	0.0008056672
0.064587352	0.0118264767	0.0081069094	-0.0697936618	0.0038304642
-0.201852329	-0.0052041047	-0.0010732692	0.0216830609	-0.0109998409
0.137536094	0.0299827004	-0.0028256827	-0.0045674867	-0.0033860727
0.035502543	-0.0025510328	0.0123298802	-0.0019578378	-0.0012352890
0.182035183	-0.0024665186	-0.0023447736	0.0307302483	-0.0036859485
-0.880424689	0.0007957924	0.0001208034	-0.0046902190	0.0224426879
-0.217235861	-0.0557748530	0.0046388545	0.0043755897	0.0063491494
-0.059051208	0.0020604199	-0.0075583664	-0.0007547275	0.0013060650
-0.326310230	0.0010699294	-0.0009470604	-0.0230596792	0.0044479716
1.583382825	-0.0022002735	-0.0093130769	0.0013987267	-0.0418356071
-0.002200273	0.8988764880	-0.1368368333	-0.0948722569	-0.0455162475
-0.009313077	-0.1368368333	0.5915667707	-0.0550559425	-0.0256378945
0.001398727	-0.0948722569	-0.0550559425	0.4136545311	-0.0190373073
-0.041835607	-0.0455162475	-0.0256378945	-0.0190373073	0.2127275191
0.022911911	-0.2098486594	0.0340223545	0.0102161672	0.0084707499
0.010018870	0.0448232473	-0.2209108774	0.0148128614	0.0091130637
0.000369723	0.0086045805	0.0084271753	-0.0567556715	0.0015421246
-0.150771143	-0.0107823773	0.0218349479	-0.0202341089	-0.0554444513

-0.2410652748	1.558810e-02	0.0064294250	0.0956599413
0.0193520057	-1.594698e-01	0.0047529695	0.0485580295
0.0066595372	3.834239e-03	-0.0580203619	0.0187930166
0.0826913815	5.689408e-02	0.0137443694	-0.6037785259
-0.2126596733	2.177539e-02	0.0068771464	0.0232833502
0.0244478758	-2.191987e-01	0.0080217961	0.0353895251
0.0079935766	7.558007e-03	-0.0770320848	0.0107155495
0.0250158238	3.301493e-02	0.0089504876	-0.2959519775
-0.0233704312	5.290910e-03	0.0043979191	0.0087738144
0.0042014090	-2.827038e-02	0.0034858310	0.0088815411
0.0040343291	6.317681e-03	-0.0192634126	0.0052224958
0.0075599150	8.329953e-03	0.0053647356	-0.0372803569
-0.0448758614	3.964997e-03	0.0036783964	0.0320715476

```
  0.0031110794 -2.942354e-02  0.0010115479  0.0065124972
  0.0013684638 -1.360425e-04 -0.0196713847  0.0160335874
  0.0311579694  1.521845e-02  0.0099557622 -0.0850446512
 -0.1207222840 -3.257817e-03  0.0039660564  0.0904527985
  0.0026072789 -1.475354e-02 -0.0011046065  0.0217234029
  0.0064249566 -5.901417e-03 -0.0260597225  0.0220421477
  0.0672738362  2.282151e-02  0.0142124521 -0.2803048232
 -0.0171672177  4.771677e-04  0.0027903781  0.0070333176
  0.0009786888 -1.608835e-02  0.0017368011  0.0052907299
 -0.0009057320 -7.631369e-05 -0.0345476832  0.0146601060
  0.0065158075  6.653857e-03  0.0120876423 -0.0772782272
 -0.0042817885  6.616462e-04  0.0008786607 -0.0017998888
  0.0009671716 -5.445370e-03  0.0002064774 -0.0011907025
  0.0031071553 -4.624942e-05 -0.0089660668 -0.0025412200
 -0.0092669100 -7.726254e-03  0.0013718780  0.0165986880
 -0.0130457711 -2.265790e-03 -0.0022946661  0.0194980582
 -0.0016307621  1.292125e-02 -0.0016326178  0.0008039441
 -0.0038912371 -2.402186e-03  0.0167548823  0.0108603110
  0.0229119108  1.001887e-02  0.0003697230 -0.1507711428
 -0.2098486594  4.482325e-02  0.0086045805 -0.0107823773
  0.0340223545 -2.209109e-01  0.0084271753  0.0218349479
  0.0102161672  1.481286e-02 -0.0567556715 -0.0202341089
  0.0084707499  9.113064e-03  0.0015421246 -0.0554444513
  0.6091206760 -5.884372e-02 -0.0232385607 -0.1394702478
 -0.0588437212  4.417250e-01 -0.0167168738 -0.0900659188
 -0.0232385607 -1.671687e-02  0.1894059794 -0.0427410709
 -0.1394702478 -9.006592e-02 -0.0427410709  0.9062865924

$dens
function (y)
sum(-lgamma(y + 1))
<environment: namespace:exactLoglinTest>

$dobs
[1] 50.26369 52.56769

$mu.hat
        75          74          73          72          70          69
 1.31600087  0.48346172  0.18073692  6.51660315  1.13800295  0.81453108
        68          67          60          59          58          57
 0.25072319  1.60473559  0.18105409  0.13394981  0.26096632  0.21080481
        55          54          53          52          50          49
 1.97444067  0.52630855  1.21928164  6.54106633  5.63440188  2.92618991
        48          47          40          39          38          37
 5.58172191  5.31553870  0.13185304  0.07554443  0.51708045  0.55718074
        35          34          33          32          30          29
 0.79011836  0.16310491  1.32752883  9.50014868  1.78879234  0.71943767
        28          27          20          19          18          17
 4.82138891  6.12481782  1.78671996  1.50458067  0.82331031  0.24350239
        80          79          78          77          76          71
 1.03386967  1.05416634  0.33141537  1.79927933  5.78126929  5.50319734
        66          65          64          63          62          61
 8.19200719  2.51212650  0.64784087  0.23712452 10.07938194 13.52352617
        56          51          46          45          44          43
 0.21322497  1.73890282  8.54214760  2.21010335  0.41355173  0.93803014
        42          41          36          31          26          25
 5.93259017  2.50572461  0.71834133  3.21909922 12.54556326  0.28923627
        24          23          22          21          16          15
```

```
 0.04191298  0.33400181  2.81785276  1.51699618  4.64188666  6.78594333
         14          13          12          11          10           9
 2.05888356  1.33968017  2.63141457 13.18407836  1.30530606  0.77160008
          8           7           6           5           4           3
 0.41339354  0.14414063  4.36555969  3.12203064  0.66493568  0.42361598
          2           1
 0.98094241  7.80847529

$n
[1] 80

$n1
[1] 40

$nosim
[1] 1000

$s
                   [,1]
(Intercept)         219
lake2                48
lake3                53
lake4                63
gender2              89
size2                95
food2                61
food3                19
food4                13
food5                32
lake2:food2          19
lake3:food2          18
lake4:food2          20
lake2:food3           7
lake3:food3           8
lake4:food3           1
lake2:food4           1
lake3:food4           4
lake4:food4           3
lake2:food5           3
lake3:food5          10
lake4:food5           6
gender2:food2        28
gender2:food3         6
gender2:food4         6
gender2:food5        14
size2:food2          16
size2:food3          13
size2:food4           8
size2:food5          13
lake2:gender2        17
lake3:gender2        13
lake4:gender2        24
lake2:size2          28
lake3:size2          29
lake4:size2          22
gender2:size2        22
lake2:gender2:size2   2
lake3:gender2:size2   1
```

```
lake4:gender2:size2   10

$stat
function (y = NULL, mu = NULL, rowlabels = FALSE)
{
    if (rowlabels)
        c("deviance", "Pearson")
    else {
        temp <- y != 0
        c(2 * sum(y[temp] * log(y[temp]/mu[temp])), sum((y -
            mu)^2/mu))
    }
}
<environment: namespace:exactLoglinTest>

$tdf
[1] 3

$x
   (Intercept) lake2 lake3 lake4 gender2 size2 food2 food3 food4 food5
75           1     0     0     1       1     0     0     0     0     1
74           1     0     0     1       1     0     0     0     1     0
73           1     0     0     1       1     0     0     1     0     0
72           1     0     0     1       1     0     1     0     0     0
70           1     0     0     1       0     1     0     0     0     1
69           1     0     0     1       0     1     0     0     1     0
68           1     0     0     1       0     1     0     1     0     0
67           1     0     0     1       0     1     1     0     0     0
60           1     0     1     0       1     1     0     0     0     1
59           1     0     1     0       1     1     0     0     1     0
58           1     0     1     0       1     1     0     1     0     0
57           1     0     1     0       1     1     1     0     0     0
55           1     0     1     0       1     0     0     0     0     1
54           1     0     1     0       1     0     0     0     1     0
53           1     0     1     0       1     0     0     1     0     0
52           1     0     1     0       1     0     1     0     0     0
50           1     0     1     0       0     1     0     0     0     1
49           1     0     1     0       0     1     0     0     1     0
48           1     0     1     0       0     1     0     1     0     0
47           1     0     1     0       0     1     1     0     0     0
40           1     1     0     0       1     1     0     0     0     1
39           1     1     0     0       1     1     0     0     1     0
38           1     1     0     0       1     1     0     1     0     0
37           1     1     0     0       1     1     1     0     0     0
35           1     1     0     0       1     0     0     0     0     1
34           1     1     0     0       1     0     0     0     1     0
33           1     1     0     0       1     0     0     1     0     0
32           1     1     0     0       1     0     1     0     0     0
30           1     1     0     0       0     1     0     0     0     1
29           1     1     0     0       0     1     0     0     1     0
28           1     1     0     0       0     1     0     1     0     0
27           1     1     0     0       0     1     1     0     0     0
20           1     0     0     0       1     1     0     0     0     1
19           1     0     0     0       1     1     0     0     1     0
18           1     0     0     0       1     1     0     1     0     0
17           1     0     0     0       1     1     1     0     0     0
80           1     0     0     1       1     1     0     0     0     1
79           1     0     0     1       1     1     0     0     1     0
```

78	1	0	0	1	1	1	0	1	0	0
77	1	0	0	1	1	1	1	0	0	0
76	1	0	0	1	1	1	0	0	0	0
71	1	0	0	1	1	0	0	0	0	0
66	1	0	0	1	0	1	0	0	0	0
65	1	0	0	1	0	0	0	0	0	1
64	1	0	0	1	0	0	0	0	1	0
63	1	0	0	1	0	0	0	1	0	0
62	1	0	0	1	0	0	1	0	0	0
61	1	0	0	1	0	0	0	0	0	0
56	1	0	1	0	1	1	0	0	0	0
51	1	0	1	0	1	0	0	0	0	0
46	1	0	1	0	0	1	0	0	0	0
45	1	0	1	0	0	0	0	0	0	1
44	1	0	1	0	0	0	0	0	1	0
43	1	0	1	0	0	0	0	1	0	0
42	1	0	1	0	0	0	1	0	0	0
41	1	0	1	0	0	0	0	0	0	0
36	1	1	0	0	1	1	0	0	0	0
31	1	1	0	0	1	0	0	0	0	0
26	1	1	0	0	0	1	0	0	0	0
25	1	1	0	0	0	0	0	0	0	1
24	1	1	0	0	0	0	0	0	1	0
23	1	1	0	0	0	0	0	1	0	0
22	1	1	0	0	0	0	1	0	0	0
21	1	1	0	0	0	0	0	0	0	0
16	1	0	0	0	1	1	0	0	0	0
15	1	0	0	0	1	0	0	0	0	1
14	1	0	0	0	1	0	0	0	1	0
13	1	0	0	0	1	0	0	1	0	0
12	1	0	0	0	1	0	1	0	0	0
11	1	0	0	0	1	0	0	0	0	0
10	1	0	0	0	0	1	0	0	0	1
9	1	0	0	0	0	1	0	0	1	0
8	1	0	0	0	0	1	0	1	0	0
7	1	0	0	0	0	1	1	0	0	0
6	1	0	0	0	0	1	0	0	0	0
5	1	0	0	0	0	0	0	0	0	1
4	1	0	0	0	0	0	0	0	1	0
3	1	0	0	0	0	0	0	1	0	0
2	1	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	0	0	0	0

	lake2:food2	lake3:food2	lake4:food2	lake2:food3	lake3:food3	lake4:food3
75	0	0	0	0	0	0
74	0	0	0	0	0	0
73	0	0	0	0	0	1
72	0	0	1	0	0	0
70	0	0	0	0	0	0
69	0	0	0	0	0	0
68	0	0	0	0	0	1
67	0	0	1	0	0	0
60	0	0	0	0	0	0
59	0	0	0	0	0	0
58	0	0	0	0	1	0
57	0	1	0	0	0	0
55	0	0	0	0	0	0
54	0	0	0	0	0	0
53	0	0	0	0	1	0

52	0	1	0	0	0	0
50	0	0	0	0	0	0
49	0	0	0	0	0	0
48	0	0	0	0	1	0
47	0	1	0	0	0	0
40	0	0	0	0	0	0
39	0	0	0	0	0	0
38	0	0	0	1	0	0
37	1	0	0	0	0	0
35	0	0	0	0	0	0
34	0	0	0	0	0	0
33	0	0	0	1	0	0
32	1	0	0	0	0	0
30	0	0	0	0	0	0
29	0	0	0	0	0	0
28	0	0	0	1	0	0
27	1	0	0	0	0	0
20	0	0	0	0	0	0
19	0	0	0	0	0	0
18	0	0	0	0	0	0
17	0	0	0	0	0	0
80	0	0	0	0	0	0
79	0	0	0	0	0	0
78	0	0	0	0	0	1
77	0	0	1	0	0	0
76	0	0	0	0	0	0
71	0	0	0	0	0	0
66	0	0	0	0	0	0
65	0	0	0	0	0	0
64	0	0	0	0	0	0
63	0	0	0	0	0	1
62	0	0	1	0	0	0
61	0	0	0	0	0	0
56	0	0	0	0	0	0
51	0	0	0	0	0	0
46	0	0	0	0	0	0
45	0	0	0	0	0	0
44	0	0	0	0	0	0
43	0	0	0	0	1	0
42	0	1	0	0	0	0
41	0	0	0	0	0	0
36	0	0	0	0	0	0
31	0	0	0	0	0	0
26	0	0	0	0	0	0
25	0	0	0	0	0	0
24	0	0	0	0	0	0
23	0	0	0	1	0	0
22	1	0	0	0	0	0
21	0	0	0	0	0	0
16	0	0	0	0	0	0
15	0	0	0	0	0	0
14	0	0	0	0	0	0
13	0	0	0	0	0	0
12	0	0	0	0	0	0
11	0	0	0	0	0	0
10	0	0	0	0	0	0
9	0	0	0	0	0	0
8	0	0	0	0	0	0

7	0	0	0	0	0	0
6	0	0	0	0	0	0
5	0	0	0	0	0	0
4	0	0	0	0	0	0
3	0	0	0	0	0	0
2	0	0	0	0	0	0
1	0	0	0	0	0	0

	lake2:food4	lake3:food4	lake4:food4	lake2:food5	lake3:food5	lake4:food5
75	0	0	0	0	0	1
74	0	0	1	0	0	0
73	0	0	0	0	0	0
72	0	0	0	0	0	0
70	0	0	0	0	0	1
69	0	0	1	0	0	0
68	0	0	0	0	0	0
67	0	0	0	0	0	0
60	0	0	0	0	1	0
59	0	1	0	0	0	0
58	0	0	0	0	0	0
57	0	0	0	0	0	0
55	0	0	0	0	1	0
54	0	1	0	0	0	0
53	0	0	0	0	0	0
52	0	0	0	0	0	0
50	0	0	0	0	1	0
49	0	1	0	0	0	0
48	0	0	0	0	0	0
47	0	0	0	0	0	0
40	0	0	0	1	0	0
39	1	0	0	0	0	0
38	0	0	0	0	0	0
37	0	0	0	0	0	0
35	0	0	0	1	0	0
34	1	0	0	0	0	0
33	0	0	0	0	0	0
32	0	0	0	0	0	0
30	0	0	0	1	0	0
29	1	0	0	0	0	0
28	0	0	0	0	0	0
27	0	0	0	0	0	0
20	0	0	0	0	0	0
19	0	0	0	0	0	0
18	0	0	0	0	0	0
17	0	0	0	0	0	0
80	0	0	0	0	0	1
79	0	0	1	0	0	0
78	0	0	0	0	0	0
77	0	0	0	0	0	0
76	0	0	0	0	0	0
71	0	0	0	0	0	0
66	0	0	0	0	0	0
65	0	0	0	0	0	1
64	0	0	1	0	0	0
63	0	0	0	0	0	0
62	0	0	0	0	0	0
61	0	0	0	0	0	0
56	0	0	0	0	0	0
51	0	0	0	0	0	0

46	0	0	0	0	0	0
45	0	0	0	0	1	0
44	0	1	0	0	0	0
43	0	0	0	0	0	0
42	0	0	0	0	0	0
41	0	0	0	0	0	0
36	0	0	0	0	0	0
31	0	0	0	0	0	0
26	0	0	0	0	0	0
25	0	0	0	1	0	0
24	1	0	0	0	0	0
23	0	0	0	0	0	0
22	0	0	0	0	0	0
21	0	0	0	0	0	0
16	0	0	0	0	0	0
15	0	0	0	0	0	0
14	0	0	0	0	0	0
13	0	0	0	0	0	0
12	0	0	0	0	0	0
11	0	0	0	0	0	0
10	0	0	0	0	0	0
9	0	0	0	0	0	0
8	0	0	0	0	0	0
7	0	0	0	0	0	0
6	0	0	0	0	0	0
5	0	0	0	0	0	0
4	0	0	0	0	0	0
3	0	0	0	0	0	0
2	0	0	0	0	0	0
1	0	0	0	0	0	0

	gender2:food2	gender2:food3	gender2:food4	gender2:food5	size2:food2
75	0	0	0	1	0
74	0	0	1	0	0
73	0	1	0	0	0
72	1	0	0	0	0
70	0	0	0	0	0
69	0	0	0	0	0
68	0	0	0	0	0
67	0	0	0	0	1
60	0	0	0	1	0
59	0	0	1	0	0
58	0	1	0	0	0
57	1	0	0	0	1
55	0	0	0	1	0
54	0	0	1	0	0
53	0	1	0	0	0
52	1	0	0	0	0
50	0	0	0	0	0
49	0	0	0	0	0
48	0	0	0	0	0
47	0	0	0	0	1
40	0	0	0	1	0
39	0	0	1	0	0
38	0	1	0	0	0
37	1	0	0	0	1
35	0	0	0	1	0
34	0	0	1	0	0
33	0	1	0	0	0

32	1	0	0	0	0
30	0	0	0	0	0
29	0	0	0	0	0
28	0	0	0	0	0
27	0	0	0	0	1
20	0	0	0	1	0
19	0	0	1	0	0
18	0	1	0	0	0
17	1	0	0	0	1
80	0	0	0	1	0
79	0	0	1	0	0
78	0	1	0	0	0
77	1	0	0	0	1
76	0	0	0	0	0
71	0	0	0	0	0
66	0	0	0	0	0
65	0	0	0	0	0
64	0	0	0	0	0
63	0	0	0	0	0
62	0	0	0	0	0
61	0	0	0	0	0
56	0	0	0	0	0
51	0	0	0	0	0
46	0	0	0	0	0
45	0	0	0	0	0
44	0	0	0	0	0
43	0	0	0	0	0
42	0	0	0	0	0
41	0	0	0	0	0
36	0	0	0	0	0
31	0	0	0	0	0
26	0	0	0	0	0
25	0	0	0	0	0
24	0	0	0	0	0
23	0	0	0	0	0
22	0	0	0	0	0
21	0	0	0	0	0
16	0	0	0	0	0
15	0	0	0	1	0
14	0	0	1	0	0
13	0	1	0	0	0
12	1	0	0	0	0
11	0	0	0	0	0
10	0	0	0	0	0
9	0	0	0	0	0
8	0	0	0	0	0
7	0	0	0	0	1
6	0	0	0	0	0
5	0	0	0	0	0
4	0	0	0	0	0
3	0	0	0	0	0
2	0	0	0	0	0
1	0	0	0	0	0

	size2:food3	size2:food4	size2:food5	lake2:gender2	lake3:gender2
75	0	0	0	0	0
74	0	0	0	0	0
73	0	0	0	0	0
72	0	0	0	0	0

70	0	0	1	0	0
69	0	1	0	0	0
68	1	0	0	0	0
67	0	0	0	0	0
60	0	0	1	0	1
59	0	1	0	0	1
58	1	0	0	0	1
57	0	0	0	0	1
55	0	0	0	0	1
54	0	0	0	0	1
53	0	0	0	0	1
52	0	0	0	0	1
50	0	0	1	0	0
49	0	1	0	0	0
48	1	0	0	0	0
47	0	0	0	0	0
40	0	0	1	1	0
39	0	1	0	1	0
38	1	0	0	1	0
37	0	0	0	1	0
35	0	0	0	1	0
34	0	0	0	1	0
33	0	0	0	1	0
32	0	0	0	1	0
30	0	0	1	0	0
29	0	1	0	0	0
28	1	0	0	0	0
27	0	0	0	0	0
20	0	0	1	0	0
19	0	1	0	0	0
18	1	0	0	0	0
17	0	0	0	0	0
80	0	0	1	0	0
79	0	1	0	0	0
78	1	0	0	0	0
77	0	0	0	0	0
76	0	0	0	0	0
71	0	0	0	0	0
66	0	0	0	0	0
65	0	0	0	0	0
64	0	0	0	0	0
63	0	0	0	0	0
62	0	0	0	0	0
61	0	0	0	0	0
56	0	0	0	0	1
51	0	0	0	0	1
46	0	0	0	0	0
45	0	0	0	0	0
44	0	0	0	0	0
43	0	0	0	0	0
42	0	0	0	0	0
41	0	0	0	0	0
36	0	0	0	1	0
31	0	0	0	1	0
26	0	0	0	0	0
25	0	0	0	0	0
24	0	0	0	0	0
23	0	0	0	0	0

22	0	0	0	0	0
21	0	0	0	0	0
16	0	0	0	0	0
15	0	0	0	0	0
14	0	0	0	0	0
13	0	0	0	0	0
12	0	0	0	0	0
11	0	0	0	0	0
10	0	0	1	0	0
9	0	1	0	0	0
8	1	0	0	0	0
7	0	0	0	0	0
6	0	0	0	0	0
5	0	0	0	0	0
4	0	0	0	0	0
3	0	0	0	0	0
2	0	0	0	0	0
1	0	0	0	0	0

	lake4:gender2	lake2:size2	lake3:size2	lake4:size2	gender2:size2
75	1	0	0	0	0
74	1	0	0	0	0
73	1	0	0	0	0
72	1	0	0	0	0
70	0	0	0	1	0
69	0	0	0	1	0
68	0	0	0	1	0
67	0	0	0	1	0
60	0	0	1	0	1
59	0	0	1	0	1
58	0	0	1	0	1
57	0	0	1	0	1
55	0	0	0	0	0
54	0	0	0	0	0
53	0	0	0	0	0
52	0	0	0	0	0
50	0	0	1	0	0
49	0	0	1	0	0
48	0	0	1	0	0
47	0	0	1	0	0
40	0	1	0	0	1
39	0	1	0	0	1
38	0	1	0	0	1
37	0	1	0	0	1
35	0	0	0	0	0
34	0	0	0	0	0
33	0	0	0	0	0
32	0	0	0	0	0
30	0	1	0	0	0
29	0	1	0	0	0
28	0	1	0	0	0
27	0	1	0	0	0
20	0	0	0	0	1
19	0	0	0	0	1
18	0	0	0	0	1
17	0	0	0	0	1
80	1	0	0	1	1
79	1	0	0	1	1
78	1	0	0	1	1

77	1	0	0	1	1
76	1	0	0	1	1
71	1	0	0	0	0
66	0	0	0	1	0
65	0	0	0	0	0
64	0	0	0	0	0
63	0	0	0	0	0
62	0	0	0	0	0
61	0	0	0	0	0
56	0	0	1	0	1
51	0	0	0	0	0
46	0	0	1	0	0
45	0	0	0	0	0
44	0	0	0	0	0
43	0	0	0	0	0
42	0	0	0	0	0
41	0	0	0	0	0
36	0	1	0	0	1
31	0	0	0	0	0
26	0	1	0	0	0
25	0	0	0	0	0
24	0	0	0	0	0
23	0	0	0	0	0
22	0	0	0	0	0
21	0	0	0	0	0
16	0	0	0	0	1
15	0	0	0	0	0
14	0	0	0	0	0
13	0	0	0	0	0
12	0	0	0	0	0
11	0	0	0	0	0
10	0	0	0	0	0
9	0	0	0	0	0
8	0	0	0	0	0
7	0	0	0	0	0
6	0	0	0	0	0
5	0	0	0	0	0
4	0	0	0	0	0
3	0	0	0	0	0
2	0	0	0	0	0
1	0	0	0	0	0

	lake2:gender2:size2	lake3:gender2:size2	lake4:gender2:size2
75	0	0	0
74	0	0	0
73	0	0	0
72	0	0	0
70	0	0	0
69	0	0	0
68	0	0	0
67	0	0	0
60	0	1	0
59	0	1	0
58	0	1	0
57	0	1	0
55	0	0	0
54	0	0	0
53	0	0	0
52	0	0	0

50	0	0	0
49	0	0	0
48	0	0	0
47	0	0	0
40	1	0	0
39	1	0	0
38	1	0	0
37	1	0	0
35	0	0	0
34	0	0	0
33	0	0	0
32	0	0	0
30	0	0	0
29	0	0	0
28	0	0	0
27	0	0	0
20	0	0	0
19	0	0	0
18	0	0	0
17	0	0	0
80	0	0	1
79	0	0	1
78	0	0	1
77	0	0	1
76	0	0	1
71	0	0	0
66	0	0	0
65	0	0	0
64	0	0	0
63	0	0	0
62	0	0	0
61	0	0	0
56	0	1	0
51	0	0	0
46	0	0	0
45	0	0	0
44	0	0	0
43	0	0	0
42	0	0	0
41	0	0	0
36	1	0	0
31	0	0	0
26	0	0	0
25	0	0	0
24	0	0	0
23	0	0	0
22	0	0	0
21	0	0	0
16	0	0	0
15	0	0	0
14	0	0	0
13	0	0	0
12	0	0	0
11	0	0	0
10	0	0	0
9	0	0	0
8	0	0	0
7	0	0	0

```
6                     0                   0                   0
5                     0                   0                   0
4                     0                   0                   0
3                     0                   0                   0
2                     0                   0                   0
1                     0                   0                   0

$x1
   (Intercept) lake2 lake3 lake4 gender2 size2 food2 food3 food4 food5
75           1     0     0     1       1     0     0     0     0     1
74           1     0     0     1       1     0     0     0     1     0
73           1     0     0     1       1     0     0     1     0     0
72           1     0     0     1       1     0     1     0     0     0
70           1     0     0     1       0     1     0     0     0     1
69           1     0     0     1       0     1     0     0     1     0
68           1     0     0     1       0     1     0     1     0     0
67           1     0     0     1       0     1     1     0     0     0
60           1     0     1     0       1     1     0     0     0     1
59           1     0     1     0       1     1     0     0     1     0
58           1     0     1     0       1     1     0     1     0     0
57           1     0     1     0       1     1     1     0     0     0
55           1     0     1     0       1     0     0     0     0     1
54           1     0     1     0       1     0     0     0     1     0
53           1     0     1     0       1     0     0     1     0     0
52           1     0     1     0       1     0     1     0     0     0
50           1     0     1     0       0     1     0     0     0     1
49           1     0     1     0       0     1     0     0     1     0
48           1     0     1     0       0     1     0     1     0     0
47           1     0     1     0       0     1     1     0     0     0
40           1     1     0     0       1     1     0     0     0     1
39           1     1     0     0       1     1     0     0     1     0
38           1     1     0     0       1     1     0     1     0     0
37           1     1     0     0       1     1     1     0     0     0
35           1     1     0     0       1     0     0     0     0     1
34           1     1     0     0       1     0     0     0     1     0
33           1     1     0     0       1     0     0     1     0     0
32           1     1     0     0       1     0     1     0     0     0
30           1     1     0     0       0     1     0     0     0     1
29           1     1     0     0       0     1     0     0     1     0
28           1     1     0     0       0     1     0     1     0     0
27           1     1     0     0       0     1     1     0     0     0
20           1     0     0     0       1     1     0     0     0     1
19           1     0     0     0       1     1     0     0     1     0
18           1     0     0     0       1     1     0     1     0     0
17           1     0     0     0       1     1     1     0     0     0
80           1     0     0     1       1     1     0     0     0     1
79           1     0     0     1       1     1     0     0     1     0
78           1     0     0     1       1     1     0     1     0     0
77           1     0     0     1       1     1     1     0     0     0
   lake2:food2 lake3:food2 lake4:food2 lake2:food3 lake3:food3 lake4:food3
75           0           0           0           0           0           0
74           0           0           0           0           0           0
73           0           0           0           0           0           1
72           0           0           1           0           0           0
70           0           0           0           0           0           0
69           0           0           0           0           0           0
68           0           0           0           0           0           1
67           0           0           1           0           0           0
```

60	0	0	0	0	0	0
59	0	0	0	0	0	0
58	0	0	0	0	1	0
57	0	1	0	0	0	0
55	0	0	0	0	0	0
54	0	0	0	0	0	0
53	0	0	0	0	1	0
52	0	1	0	0	0	0
50	0	0	0	0	0	0
49	0	0	0	0	0	0
48	0	0	0	0	1	0
47	0	1	0	0	0	0
40	0	0	0	0	0	0
39	0	0	0	0	0	0
38	0	0	0	1	0	0
37	1	0	0	0	0	0
35	0	0	0	0	0	0
34	0	0	0	0	0	0
33	0	0	0	1	0	0
32	1	0	0	0	0	0
30	0	0	0	0	0	0
29	0	0	0	0	0	0
28	0	0	0	1	0	0
27	1	0	0	0	0	0
20	0	0	0	0	0	0
19	0	0	0	0	0	0
18	0	0	0	0	0	0
17	0	0	0	0	0	0
80	0	0	0	0	0	0
79	0	0	0	0	0	0
78	0	0	0	0	0	1
77	0	0	1	0	0	0

	lake2:food4	lake3:food4	lake4:food4	lake2:food5	lake3:food5	lake4:food5
75	0	0	0	0	0	1
74	0	0	1	0	0	0
73	0	0	0	0	0	0
72	0	0	0	0	0	0
70	0	0	0	0	0	1
69	0	0	1	0	0	0
68	0	0	0	0	0	0
67	0	0	0	0	0	0
60	0	0	0	0	1	0
59	0	1	0	0	0	0
58	0	0	0	0	0	0
57	0	0	0	0	0	0
55	0	0	0	0	1	0
54	0	1	0	0	0	0
53	0	0	0	0	0	0
52	0	0	0	0	0	0
50	0	0	0	0	1	0
49	0	1	0	0	0	0
48	0	0	0	0	0	0
47	0	0	0	0	0	0
40	0	0	0	1	0	0
39	1	0	0	0	0	0
38	0	0	0	0	0	0
37	0	0	0	0	0	0
35	0	0	0	1	0	0

```
34          1           0           0           0           0           0
33          0           0           0           0           0           0
32          0           0           0           0           0           0
30          0           0           0           1           0           0
29          1           0           0           0           0           0
28          0           0           0           0           0           0
27          0           0           0           0           0           0
20          0           0           0           0           0           0
19          0           0           0           0           0           0
18          0           0           0           0           0           0
17          0           0           0           0           0           0
80          0           0           0           0           0           1
79          0           0           1           0           0           0
78          0           0           0           0           0           0
77          0           0           0           0           0           0
   gender2:food2 gender2:food3 gender2:food4 gender2:food5 size2:food2
75             0             0             0             1           0
74             0             0             1             0           0
73             0             1             0             0           0
72             1             0             0             0           0
70             0             0             0             0           0
69             0             0             0             0           0
68             0             0             0             0           0
67             0             0             0             0           1
60             0             0             0             1           0
59             0             0             1             0           0
58             0             1             0             0           0
57             1             0             0             0           1
55             0             0             0             1           0
54             0             0             1             0           0
53             0             1             0             0           0
52             1             0             0             0           0
50             0             0             0             0           0
49             0             0             0             0           0
48             0             0             0             0           0
47             0             0             0             0           1
40             0             0             0             1           0
39             0             0             1             0           0
38             0             1             0             0           0
37             1             0             0             0           1
35             0             0             0             1           0
34             0             0             1             0           0
33             0             1             0             0           0
32             1             0             0             0           0
30             0             0             0             0           0
29             0             0             0             0           0
28             0             0             0             0           0
27             0             0             0             0           1
20             0             0             0             1           0
19             0             0             1             0           0
18             0             1             0             0           0
17             1             0             0             0           1
80             0             0             0             1           0
79             0             0             1             0           0
78             0             1             0             0           0
77             1             0             0             0           1
   size2:food3 size2:food4 size2:food5 lake2:gender2 lake3:gender2
75           0           0           0             0             0
```

74	0	0	0	0	0
73	0	0	0	0	0
72	0	0	0	0	0
70	0	0	1	0	0
69	0	1	0	0	0
68	1	0	0	0	0
67	0	0	0	0	0
60	0	0	1	0	1
59	0	1	0	0	1
58	1	0	0	0	1
57	0	0	0	0	1
55	0	0	0	0	1
54	0	0	0	0	1
53	0	0	0	0	1
52	0	0	0	0	1
50	0	0	1	0	0
49	0	1	0	0	0
48	1	0	0	0	0
47	0	0	0	0	0
40	0	0	1	1	0
39	0	1	0	1	0
38	1	0	0	1	0
37	0	0	0	1	0
35	0	0	0	1	0
34	0	0	0	1	0
33	0	0	0	1	0
32	0	0	0	1	0
30	0	0	1	0	0
29	0	1	0	0	0
28	1	0	0	0	0
27	0	0	0	0	0
20	0	0	1	0	0
19	0	1	0	0	0
18	1	0	0	0	0
17	0	0	0	0	0
80	0	0	1	0	0
79	0	1	0	0	0
78	1	0	0	0	0
77	0	0	0	0	0

	lake4:gender2	lake2:size2	lake3:size2	lake4:size2	gender2:size2
75	1	0	0	0	0
74	1	0	0	0	0
73	1	0	0	0	0
72	1	0	0	0	0
70	0	0	0	1	0
69	0	0	0	1	0
68	0	0	0	1	0
67	0	0	0	1	0
60	0	0	1	0	1
59	0	0	1	0	1
58	0	0	1	0	1
57	0	0	1	0	1
55	0	0	0	0	0
54	0	0	0	0	0
53	0	0	0	0	0
52	0	0	0	0	0
50	0	0	1	0	0
49	0	0	1	0	0

```
48                0           0           1           0            0
47                0           0           1           0            0
40                0           1           0           0            1
39                0           1           0           0            1
38                0           1           0           0            1
37                0           1           0           0            1
35                0           0           0           0            0
34                0           0           0           0            0
33                0           0           0           0            0
32                0           0           0           0            0
30                0           1           0           0            0
29                0           1           0           0            0
28                0           1           0           0            0
27                0           1           0           0            0
20                0           0           0           0            1
19                0           0           0           0            1
18                0           0           0           0            1
17                0           0           0           0            1
80                1           0           0           1            1
79                1           0           0           1            1
78                1           0           0           1            1
77                1           0           0           1            1
   lake2:gender2:size2 lake3:gender2:size2 lake4:gender2:size2
75                   0                   0                   0
74                   0                   0                   0
73                   0                   0                   0
72                   0                   0                   0
70                   0                   0                   0
69                   0                   0                   0
68                   0                   0                   0
67                   0                   0                   0
60                   0                   1                   0
59                   0                   1                   0
58                   0                   1                   0
57                   0                   1                   0
55                   0                   0                   0
54                   0                   0                   0
53                   0                   0                   0
52                   0                   0                   0
50                   0                   0                   0
49                   0                   0                   0
48                   0                   0                   0
47                   0                   0                   0
40                   1                   0                   0
39                   1                   0                   0
38                   1                   0                   0
37                   1                   0                   0
35                   0                   0                   0
34                   0                   0                   0
33                   0                   0                   0
32                   0                   0                   0
30                   0                   0                   0
29                   0                   0                   0
28                   0                   0                   0
27                   0                   0                   0
20                   0                   0                   0
19                   0                   0                   0
18                   0                   0                   0
```

```
17                  0                   0                   0
80                  0                   0                   1
79                  0                   0                   1
78                  0                   0                   1
77                  0                   0                   1

$x2invt
   (Intercept) lake2 lake3 lake4 gender2 size2 food2 food3 food4 food5
76           0     0     0     0       0     0     0     0     0     0
71           0     0     0     0       0     0     0     0     0     0
66           0     0     0     0       0     0     0     0     0     0
65           0     0     0     0       0     0     0     0     0     0
64           0     0     0     0       0     0     0     0     0     0
63           0     0     0     0       0     0     0     0     0     0
62           0     0     0     0       0     0     0     0     0     0
61           0     0     0     1       0     0     0     0     0     0
56           0     0     0     0       0     0     0     0     0     0
51           0     0     0     0       0     0     0     0     0     0
46           0     0     0     0       0     0     0     0     0     0
45           0     0     0     0       0     0     0     0     0     0
44           0     0     0     0       0     0     0     0     0     0
43           0     0     0     0       0     0     0     0     0     0
42           0     0     0     0       0     0     0     0     0     0
41           0     0     1     0       0     0     0     0     0     0
36           0     0     0     0       0     0     0     0     0     0
31           0     0     0     0       0     0     0     0     0     0
26           0     0     0     0       0     0     0     0     0     0
25           0     0     0     0       0     0     0     0     0     0
24           0     0     0     0       0     0     0     0     0     0
23           0     0     0     0       0     0     0     0     0     0
22           0     0     0     0       0     0     0     0     0     0
21           0     1     0     0       0     0     0     0     0     0
16           0     0     0     0       0     0     0     0     0     0
15           0     0     0     0       0     0     0     0     0     0
14           0     0     0     0       0     0     0     0     0     0
13           0     0     0     0       0     0     0     0     0     0
12           0     0     0     0       0     0     0     0     0     0
11           0     0     0     0       1     0     0     0     0     0
10           0     0     0     0       0     0     0     0     0     0
9            0     0     0     0       0     0     0     0     0     0
8            0     0     0     0       0     0     0     0     0     0
7            0     0     0     0       0     0     0     0     0     0
6            0     0     0     0       0     1     0     0     0     0
5            0     0     0     0       0     0     0     0     0     1
4            0     0     0     0       0     0     0     0     1     0
3            0     0     0     0       0     0     0     1     0     0
2            0     0     0     0       0     0     1     0     0     0
1            1    -1    -1    -1      -1    -1    -1    -1    -1    -1
   lake2:food2 lake3:food2 lake4:food2 lake2:food3 lake3:food3 lake4:food3
76           0           0           0           0           0           0
71           0           0           0           0           0           0
66           0           0           0           0           0           0
65           0           0           0           0           0           0
64           0           0           0           0           0           0
63           0           0           0           0           0           1
62           0           0           1           0           0           0
61           0           0          -1           0           0          -1
56           0           0           0           0           0           0
```

51	0	0	0	0	0	0
46	0	0	0	0	0	0
45	0	0	0	0	0	0
44	0	0	0	0	0	0
43	0	0	0	0	1	0
42	0	1	0	0	0	0
41	0	-1	0	0	-1	0
36	0	0	0	0	0	0
31	0	0	0	0	0	0
26	0	0	0	0	0	0
25	0	0	0	0	0	0
24	0	0	0	0	0	0
23	0	0	0	1	0	0
22	1	0	0	0	0	0
21	-1	0	0	-1	0	0
16	0	0	0	0	0	0
15	0	0	0	0	0	0
14	0	0	0	0	0	0
13	0	0	0	0	0	0
12	0	0	0	0	0	0
11	0	0	0	0	0	0
10	0	0	0	0	0	0
9	0	0	0	0	0	0
8	0	0	0	0	0	0
7	0	0	0	0	0	0
6	0	0	0	0	0	0
5	0	0	0	0	0	0
4	0	0	0	0	0	0
3	0	0	0	-1	-1	-1
2	-1	-1	-1	0	0	0
1	1	1	1	1	1	1

	lake2:food4	lake3:food4	lake4:food4	lake2:food5	lake3:food5	lake4:food5
76	0	0	0	0	0	0
71	0	0	0	0	0	0
66	0	0	0	0	0	0
65	0	0	0	0	0	1
64	0	0	1	0	0	0
63	0	0	0	0	0	0
62	0	0	0	0	0	0
61	0	0	-1	0	0	-1
56	0	0	0	0	0	0
51	0	0	0	0	0	0
46	0	0	0	0	0	0
45	0	0	0	0	1	0
44	0	1	0	0	0	0
43	0	0	0	0	0	0
42	0	0	0	0	0	0
41	0	-1	0	0	-1	0
36	0	0	0	0	0	0
31	0	0	0	0	0	0
26	0	0	0	0	0	0
25	0	0	0	1	0	0
24	1	0	0	0	0	0
23	0	0	0	0	0	0
22	0	0	0	0	0	0
21	-1	0	0	-1	0	0
16	0	0	0	0	0	0
15	0	0	0	0	0	0

```
14            0          0          0          0          0          0
13            0          0          0          0          0          0
12            0          0          0          0          0          0
11            0          0          0          0          0          0
10            0          0          0          0          0          0
9             0          0          0          0          0          0
8             0          0          0          0          0          0
7             0          0          0          0          0          0
6             0          0          0          0          0          0
5             0          0          0         -1         -1         -1
4            -1         -1         -1          0          0          0
3             0          0          0          0          0          0
2             0          0          0          0          0          0
1             1          1          1          1          1          1
   gender2:food2 gender2:food3 gender2:food4 gender2:food5 size2:food2
76             0             0             0             0           0
71             0             0             0             0           0
66             0             0             0             0           0
65             0             0             0             0           0
64             0             0             0             0           0
63             0             0             0             0           0
62             0             0             0             0           0
61             0             0             0             0           0
56             0             0             0             0           0
51             0             0             0             0           0
46             0             0             0             0           0
45             0             0             0             0           0
44             0             0             0             0           0
43             0             0             0             0           0
42             0             0             0             0           0
41             0             0             0             0           0
36             0             0             0             0           0
31             0             0             0             0           0
26             0             0             0             0           0
25             0             0             0             0           0
24             0             0             0             0           0
23             0             0             0             0           0
22             0             0             0             0           0
21             0             0             0             0           0
16             0             0             0             0           0
15             0             0             0             1           0
14             0             0             1             0           0
13             0             1             0             0           0
12             1             0             0             0           0
11            -1            -1            -1            -1           0
10             0             0             0             0           0
9              0             0             0             0           0
8              0             0             0             0           0
7              0             0             0             0           1
6              0             0             0             0          -1
5              0             0             0            -1           0
4              0             0            -1             0           0
3              0            -1             0             0           0
2             -1             0             0             0          -1
1              1             1             1             1           1
   size2:food3 size2:food4 size2:food5 lake2:gender2 lake3:gender2
76           0           0           0             0             0
71           0           0           0             0             0
```

66	0	0	0	0	0
65	0	0	0	0	0
64	0	0	0	0	0
63	0	0	0	0	0
62	0	0	0	0	0
61	0	0	0	0	0
56	0	0	0	0	0
51	0	0	0	0	1
46	0	0	0	0	0
45	0	0	0	0	0
44	0	0	0	0	0
43	0	0	0	0	0
42	0	0	0	0	0
41	0	0	0	0	-1
36	0	0	0	0	0
31	0	0	0	1	0
26	0	0	0	0	0
25	0	0	0	0	0
24	0	0	0	0	0
23	0	0	0	0	0
22	0	0	0	0	0
21	0	0	0	-1	0
16	0	0	0	0	0
15	0	0	0	0	0
14	0	0	0	0	0
13	0	0	0	0	0
12	0	0	0	0	0
11	0	0	0	-1	-1
10	0	0	1	0	0
9	0	1	0	0	0
8	1	0	0	0	0
7	0	0	0	0	0
6	-1	-1	-1	0	0
5	0	0	-1	0	0
4	0	-1	0	0	0
3	-1	0	0	0	0
2	0	0	0	0	0
1	1	1	1	1	1

	lake4:gender2	lake2:size2	lake3:size2	lake4:size2	gender2:size2
76	0	0	0	0	0
71	1	0	0	0	0
66	0	0	0	1	0
65	0	0	0	0	0
64	0	0	0	0	0
63	0	0	0	0	0
62	0	0	0	0	0
61	-1	0	0	-1	0
56	0	0	0	0	0
51	0	0	0	0	0
46	0	0	1	0	0
45	0	0	0	0	0
44	0	0	0	0	0
43	0	0	0	0	0
42	0	0	0	0	0
41	0	0	-1	0	0
36	0	0	0	0	0
31	0	0	0	0	0
26	0	1	0	0	0

```
25           0          0          0          0          0
24           0          0          0          0          0
23           0          0          0          0          0
22           0          0          0          0          0
21           0         -1          0          0          0
16           0          0          0          0          1
15           0          0          0          0          0
14           0          0          0          0          0
13           0          0          0          0          0
12           0          0          0          0          0
11          -1          0          0          0         -1
10           0          0          0          0          0
9            0          0          0          0          0
8            0          0          0          0          0
7            0          0          0          0          0
6            0         -1         -1         -1         -1
5            0          0          0          0          0
4            0          0          0          0          0
3            0          0          0          0          0
2            0          0          0          0          0
1            1          1          1          1          1
   lake2:gender2:size2 lake3:gender2:size2 lake4:gender2:size2
76                   0                   0                   1
71                   0                   0                  -1
66                   0                   0                  -1
65                   0                   0                   0
64                   0                   0                   0
63                   0                   0                   0
62                   0                   0                   0
61                   0                   0                   1
56                   0                   1                   0
51                   0                  -1                   0
46                   0                  -1                   0
45                   0                   0                   0
44                   0                   0                   0
43                   0                   0                   0
42                   0                   0                   0
41                   0                   1                   0
36                   1                   0                   0
31                  -1                   0                   0
26                  -1                   0                   0
25                   0                   0                   0
24                   0                   0                   0
23                   0                   0                   0
22                   0                   0                   0
21                   1                   0                   0
16                  -1                  -1                  -1
15                   0                   0                   0
14                   0                   0                   0
13                   0                   0                   0
12                   0                   0                   0
11                   1                   1                   1
10                   0                   0                   0
9                    0                   0                   0
8                    0                   0                   0
7                    0                   0                   0
6                    1                   1                   1
5                    0                   0                   0
```

```
4                    0                   0                   0
3                    0                   0                   0
2                    0                   0                   0
1                   -1                  -1                  -1

$y
75 74 73 72 70 69 68 67 60 59 58 57 55 54 53 52 50 49 48 47 40 39 38 37 35 34
 1  0  1  9  2  1  0  0  0  0  0  1  4  1  1  4  5  3  6  6  0  1  0  1  2  0
33 32 30 29 28 27 20 19 18 17 80 79 78 77 76 71 66 65 64 63 62 61 56 51 46 45
 1  9  0  0  6  7  3  2  1  0  1  0  0  1  8  3  9  2  2  0 10 13  0  2  8  1
44 43 42 41 36 31 26 25 24 23 22 21 16 15 14 13 12 11 10  9  8  7  6  5  4  3
 0  1  7  3  0  3 13  1  0  0  2  2  3  3  2  2  3 16  2  1  0  0  4  5  0  0
 2  1
 1  7

$ord
 [1] 75 74 73 72 70 69 68 67 60 59 58 57 55 54 53 52 50 49 48 47 40 39 38 37 35
[26] 34 33 32 30 29 28 27 20 19 18 17 80 79 78 77 76 71 66 65 64 63 62 61 56 51
[51] 46 45 44 43 42 41 36 31 26 25 24 23 22 21 16 15 14 13 12 11 10  9  8  7  6
[76]  5  4  3  2  1

$glm.fit

Call:  glm(formula = formula, family = poisson, data = data, x = TRUE,
    y = TRUE)

Coefficients:
        (Intercept)                lake2                lake3
            2.05521             -1.63848             -1.13663
              lake4              gender2                size2
            0.54922              0.52380             -0.58146
              food2                food3                food4
           -2.07445             -2.91414             -2.46327
              food5          lake2:food2          lake3:food2
           -0.91673              2.69369              2.93633
        lake4:food2          lake2:food3          lake3:food3
            1.78051              1.40080              1.93159
        lake4:food3          lake2:food4          lake3:food4
           -1.12946             -1.12562              0.66172
        lake4:food4          lake2:food5          lake3:food5
           -0.57527             -0.74052              0.79119
        lake4:food5        gender2:food2        gender2:food3
           -0.76658              0.46296              0.62756
      gender2:food4        gender2:food5          size2:food2
            0.60643              0.25257             -1.33626
        size2:food3          size2:food4          size2:food5
            0.55704              0.73024             -0.29058
      lake2:gender2        lake3:gender2        lake4:gender2
            0.22857             -0.88912             -1.42290
        lake2:size2          lake3:size2          lake4:size2
            2.69410              1.80790              0.08019
      gender2:size2  lake2:gender2:size2  lake3:gender2:size2
           -0.46243             -3.15012             -2.86267
lake4:gender2:size2
            1.01299

Degrees of Freedom: 79 Total (i.e. Null);  40 Residual
Null Deviance:            307.2
```

```
Residual Deviance: 50.26        AIC: 293.7

$p
[1] 0.4

$batchsize
[1] 100

$startiter
[1] 1001

$mhap
[1] 5

$chain
        deviance  Pearson
   [1,] 50.26369 52.56769
   [2,] 50.26369 52.56769
   [3,] 50.26369 52.56769
   [4,] 50.26369 52.56769
   [5,] 50.26369 52.56769
   [6,] 50.26369 52.56769
   [7,] 50.26369 52.56769
   [8,] 50.26369 52.56769
   [9,] 50.26369 52.56769
  [10,] 50.26369 52.56769
  [11,] 50.26369 52.56769
  [12,] 50.26369 52.56769
  [13,] 50.26369 52.56769
  [14,] 50.26369 52.56769
  [15,] 50.26369 52.56769
  [16,] 50.26369 52.56769
  [17,] 50.26369 52.56769
  [18,] 50.26369 52.56769
  [19,] 50.26369 52.56769
  [20,] 50.26369 52.56769
  [21,] 50.26369 52.56769
  [22,] 50.26369 52.56769
  [23,] 50.26369 52.56769
  [24,] 50.26369 52.56769
  [25,] 50.26369 52.56769
  [26,] 50.26369 52.56769
  [27,] 50.26369 52.56769
  [28,] 50.26369 52.56769
  [29,] 50.26369 52.56769
  [30,] 50.26369 52.56769
  [31,] 50.26369 52.56769
  [32,] 50.26369 52.56769
  [33,] 50.26369 52.56769
  [34,] 50.26369 52.56769
  [35,] 50.26369 52.56769
  [36,] 50.26369 52.56769
  [37,] 50.26369 52.56769
  [38,] 50.26369 52.56769
  [39,] 50.26369 52.56769
  [40,] 50.26369 52.56769
  [41,] 50.26369 52.56769
  [42,] 50.26369 52.56769
```

```
 [43,] 50.26369 52.56769
 [44,] 50.26369 52.56769
 [45,] 50.26369 52.56769
 [46,] 50.26369 52.56769
 [47,] 50.26369 52.56769
 [48,] 50.26369 52.56769
 [49,] 50.26369 52.56769
 [50,] 50.26369 52.56769
 [51,] 50.26369 52.56769
 [52,] 50.26369 52.56769
 [53,] 50.26369 52.56769
 [54,] 50.26369 52.56769
 [55,] 50.26369 52.56769
 [56,] 50.26369 52.56769
 [57,] 50.26369 52.56769
 [58,] 50.26369 52.56769
 [59,] 50.26369 52.56769
 [60,] 50.26369 52.56769
 [61,] 50.26369 52.56769
 [62,] 50.26369 52.56769
 [63,] 50.26369 52.56769
 [64,] 50.26369 52.56769
 [65,] 50.26369 52.56769
 [66,] 50.26369 52.56769
 [67,] 50.26369 52.56769
 [68,] 50.26369 52.56769
 [69,] 50.26369 52.56769
 [70,] 50.26369 52.56769
 [71,] 50.26369 52.56769
 [72,] 50.26369 52.56769
 [73,] 50.26369 52.56769
 [74,] 50.26369 52.56769
 [75,] 50.26369 52.56769
 [76,] 50.26369 52.56769
 [77,] 50.26369 52.56769
 [78,] 50.26369 52.56769
 [79,] 50.26369 52.56769
 [80,] 50.26369 52.56769
 [81,] 50.26369 52.56769
 [82,] 50.26369 52.56769
 [83,] 47.04594 43.66446
 [84,] 47.04594 43.66446
 [85,] 47.04594 43.66446
 [86,] 47.04594 43.66446
 [87,] 47.04594 43.66446
 [88,] 47.04594 43.66446
 [89,] 47.04594 43.66446
 [90,] 47.04594 43.66446
 [91,] 47.04594 43.66446
 [92,] 47.04594 43.66446
 [93,] 47.04594 43.66446
 [94,] 47.04594 43.66446
 [95,] 47.04594 43.66446
 [96,] 47.04594 43.66446
 [97,] 47.04594 43.66446
 [98,] 47.04594 43.66446
 [99,] 47.04594 43.66446
[100,] 47.04594 43.66446
```

```
[101,] 47.04594 43.66446
[102,] 47.04594 43.66446
[103,] 47.04594 43.66446
[104,] 47.04594 43.66446
[105,] 47.04594 43.66446
[106,] 47.04594 43.66446
[107,] 47.04594 43.66446
[108,] 47.04594 43.66446
[109,] 47.04594 43.66446
[110,] 47.04594 43.66446
[111,] 47.04594 43.66446
[112,] 47.04594 43.66446
[113,] 47.04594 43.66446
[114,] 47.04594 43.66446
[115,] 47.04594 43.66446
[116,] 47.04594 43.66446
[117,] 47.04594 43.66446
[118,] 47.04594 43.66446
[119,] 47.04594 43.66446
[120,] 47.04594 43.66446
[121,] 47.04594 43.66446
[122,] 47.04594 43.66446
[123,] 47.04594 43.66446
[124,] 47.04594 43.66446
[125,] 47.04594 43.66446
[126,] 47.04594 43.66446
[127,] 47.04594 43.66446
[128,] 47.04594 43.66446
[129,] 47.04594 43.66446
[130,] 47.04594 43.66446
[131,] 47.04594 43.66446
[132,] 47.04594 43.66446
[133,] 47.04594 43.66446
[134,] 47.04594 43.66446
[135,] 44.78849 43.88544
[136,] 44.78849 43.88544
[137,] 44.78849 43.88544
[138,] 44.78849 43.88544
[139,] 44.78849 43.88544
[140,] 44.78849 43.88544
[141,] 44.78849 43.88544
[142,] 44.78849 43.88544
[143,] 44.78849 43.88544
[144,] 44.78849 43.88544
[145,] 44.78849 43.88544
[146,] 44.78849 43.88544
[147,] 44.78849 43.88544
[148,] 44.78849 43.88544
[149,] 44.78849 43.88544
[150,] 44.78849 43.88544
[151,] 44.78849 43.88544
[152,] 44.78849 43.88544
[153,] 44.78849 43.88544
[154,] 44.78849 43.88544
[155,] 44.78849 43.88544
[156,] 44.78849 43.88544
[157,] 44.78849 43.88544
[158,] 44.78849 43.88544
```

```
[159,] 44.78849 43.88544
[160,] 44.78849 43.88544
[161,] 44.78849 43.88544
[162,] 44.78849 43.88544
[163,] 44.78849 43.88544
[164,] 44.78849 43.88544
[165,] 44.78849 43.88544
[166,] 44.78849 43.88544
[167,] 44.78849 43.88544
[168,] 44.78849 43.88544
[169,] 44.78849 43.88544
[170,] 44.78849 43.88544
[171,] 44.78849 43.88544
[172,] 44.78849 43.88544
[173,] 44.78849 43.88544
[174,] 44.78849 43.88544
[175,] 44.78849 43.88544
[176,] 44.78849 43.88544
[177,] 44.78849 43.88544
[178,] 44.78849 43.88544
[179,] 44.78849 43.88544
[180,] 44.78849 43.88544
[181,] 44.78849 43.88544
[182,] 44.78849 43.88544
[183,] 44.78849 43.88544
[184,] 44.78849 43.88544
[185,] 44.78849 43.88544
[186,] 44.78849 43.88544
[187,] 44.78849 43.88544
[188,] 44.78849 43.88544
[189,] 44.78849 43.88544
[190,] 44.78849 43.88544
[191,] 44.78849 43.88544
[192,] 44.78849 43.88544
[193,] 44.78849 43.88544
[194,] 44.78849 43.88544
[195,] 44.78849 43.88544
[196,] 44.78849 43.88544
[197,] 44.78849 43.88544
[198,] 44.78849 43.88544
[199,] 44.78849 43.88544
[200,] 44.78849 43.88544
[201,] 44.78849 43.88544
[202,] 44.78849 43.88544
[203,] 44.78849 43.88544
[204,] 44.78849 43.88544
[205,] 44.78849 43.88544
[206,] 44.78849 43.88544
[207,] 44.78849 43.88544
[208,] 44.78849 43.88544
[209,] 44.78849 43.88544
[210,] 44.78849 43.88544
[211,] 44.78849 43.88544
[212,] 44.78849 43.88544
[213,] 44.78849 43.88544
[214,] 44.78849 43.88544
[215,] 44.78849 43.88544
[216,] 44.78849 43.88544
```

```
[217,] 44.78849 43.88544
[218,] 44.78849 43.88544
[219,] 44.78849 43.88544
[220,] 44.78849 43.88544
[221,] 44.78849 43.88544
[222,] 44.78849 43.88544
[223,] 44.78849 43.88544
[224,] 44.78849 43.88544
[225,] 44.78849 43.88544
[226,] 44.78849 43.88544
[227,] 44.78849 43.88544
[228,] 44.78849 43.88544
[229,] 44.78849 43.88544
[230,] 44.78849 43.88544
[231,] 44.78849 43.88544
[232,] 44.78849 43.88544
[233,] 44.78849 43.88544
[234,] 44.78849 43.88544
[235,] 44.78849 43.88544
[236,] 44.78849 43.88544
[237,] 44.78849 43.88544
[238,] 44.78849 43.88544
[239,] 44.78849 43.88544
[240,] 44.78849 43.88544
[241,] 44.78849 43.88544
[242,] 44.78849 43.88544
[243,] 44.78849 43.88544
[244,] 44.78849 43.88544
[245,] 44.78849 43.88544
[246,] 44.78849 43.88544
[247,] 44.78849 43.88544
[248,] 44.78849 43.88544
[249,] 44.78849 43.88544
[250,] 44.78849 43.88544
[251,] 44.78849 43.88544
[252,] 44.78849 43.88544
[253,] 44.78849 43.88544
[254,] 44.78849 43.88544
[255,] 44.78849 43.88544
[256,] 44.78849 43.88544
[257,] 44.78849 43.88544
[258,] 44.78849 43.88544
[259,] 44.78849 43.88544
[260,] 44.78849 43.88544
[261,] 44.78849 43.88544
[262,] 44.78849 43.88544
[263,] 44.78849 43.88544
[264,] 44.78849 43.88544
[265,] 44.78849 43.88544
[266,] 44.78849 43.88544
[267,] 44.78849 43.88544
[268,] 44.78849 43.88544
[269,] 44.78849 43.88544
[270,] 44.78849 43.88544
[271,] 44.78849 43.88544
[272,] 44.78849 43.88544
[273,] 44.78849 43.88544
[274,] 44.78849 43.88544
```

```
[275,] 44.78849 43.88544
[276,] 44.78849 43.88544
[277,] 44.78849 43.88544
[278,] 44.78849 43.88544
[279,] 44.78849 43.88544
[280,] 44.78849 43.88544
[281,] 44.78849 43.88544
[282,] 44.78849 43.88544
[283,] 44.78849 43.88544
[284,] 44.78849 43.88544
[285,] 44.78849 43.88544
[286,] 44.78849 43.88544
[287,] 44.78849 43.88544
[288,] 44.78849 43.88544
[289,] 44.78849 43.88544
[290,] 44.78849 43.88544
[291,] 44.78849 43.88544
[292,] 44.78849 43.88544
[293,] 44.78849 43.88544
[294,] 44.78849 43.88544
[295,] 44.78849 43.88544
[296,] 44.78849 43.88544
[297,] 44.78849 43.88544
[298,] 44.78849 43.88544
[299,] 44.78849 43.88544
[300,] 44.78849 43.88544
[301,] 44.78849 43.88544
[302,] 44.78849 43.88544
[303,] 44.78849 43.88544
[304,] 44.78849 43.88544
[305,] 44.78849 43.88544
[306,] 44.78849 43.88544
[307,] 45.77456 44.93293
[308,] 45.77456 44.93293
[309,] 45.77456 44.93293
[310,] 45.77456 44.93293
[311,] 45.77456 44.93293
[312,] 45.77456 44.93293
[313,] 45.77456 44.93293
[314,] 45.77456 44.93293
[315,] 45.77456 44.93293
[316,] 45.77456 44.93293
[317,] 45.77456 44.93293
[318,] 45.77456 44.93293
[319,] 45.77456 44.93293
[320,] 45.77456 44.93293
[321,] 45.77456 44.93293
[322,] 45.77456 44.93293
[323,] 45.77456 44.93293
[324,] 45.77456 44.93293
[325,] 45.77456 44.93293
[326,] 45.77456 44.93293
[327,] 45.77456 44.93293
[328,] 45.77456 44.93293
[329,] 45.77456 44.93293
[330,] 45.77456 44.93293
[331,] 45.77456 44.93293
[332,] 45.77456 44.93293
```

```
[333,] 45.77456 44.93293
[334,] 45.77456 44.93293
[335,] 45.77456 44.93293
[336,] 45.77456 44.93293
[337,] 45.77456 44.93293
[338,] 45.77456 44.93293
[339,] 45.77456 44.93293
[340,] 45.77456 44.93293
[341,] 45.77456 44.93293
[342,] 45.77456 44.93293
[343,] 45.77456 44.93293
[344,] 45.77456 44.93293
[345,] 45.77456 44.93293
[346,] 45.77456 44.93293
[347,] 45.77456 44.93293
[348,] 45.77456 44.93293
[349,] 45.77456 44.93293
[350,] 45.77456 44.93293
[351,] 45.77456 44.93293
[352,] 45.77456 44.93293
[353,] 45.77456 44.93293
[354,] 45.77456 44.93293
[355,] 45.77456 44.93293
[356,] 45.77456 44.93293
[357,] 45.77456 44.93293
[358,] 45.77456 44.93293
[359,] 45.77456 44.93293
[360,] 45.77456 44.93293
[361,] 45.77456 44.93293
[362,] 45.77456 44.93293
[363,] 45.77456 44.93293
[364,] 45.77456 44.93293
[365,] 45.77456 44.93293
[366,] 45.77456 44.93293
[367,] 45.77456 44.93293
[368,] 45.77456 44.93293
[369,] 45.77456 44.93293
[370,] 45.77456 44.93293
[371,] 45.77456 44.93293
[372,] 45.77456 44.93293
[373,] 45.77456 44.93293
[374,] 45.77456 44.93293
[375,] 45.77456 44.93293
[376,] 45.77456 44.93293
[377,] 45.77456 44.93293
[378,] 45.77456 44.93293
[379,] 45.77456 44.93293
[380,] 45.77456 44.93293
[381,] 45.77456 44.93293
[382,] 45.77456 44.93293
[383,] 45.77456 44.93293
[384,] 45.77456 44.93293
[385,] 45.77456 44.93293
[386,] 45.77456 44.93293
[387,] 45.77456 44.93293
[388,] 45.77456 44.93293
[389,] 45.77456 44.93293
[390,] 45.77456 44.93293
```

```
[391,] 45.77456 44.93293
[392,] 45.77456 44.93293
[393,] 45.77456 44.93293
[394,] 45.77456 44.93293
[395,] 45.77456 44.93293
[396,] 45.77456 44.93293
[397,] 45.77456 44.93293
[398,] 45.77456 44.93293
[399,] 45.77456 44.93293
[400,] 45.77456 44.93293
[401,] 45.77456 44.93293
[402,] 45.77456 44.93293
[403,] 45.77456 44.93293
[404,] 45.77456 44.93293
[405,] 45.77456 44.93293
[406,] 45.77456 44.93293
[407,] 45.77456 44.93293
[408,] 45.77456 44.93293
[409,] 45.77456 44.93293
[410,] 45.77456 44.93293
[411,] 45.77456 44.93293
[412,] 45.77456 44.93293
[413,] 45.77456 44.93293
[414,] 45.77456 44.93293
[415,] 45.77456 44.93293
[416,] 45.77456 44.93293
[417,] 45.77456 44.93293
[418,] 45.77456 44.93293
[419,] 45.77456 44.93293
[420,] 45.77456 44.93293
[421,] 45.77456 44.93293
[422,] 45.77456 44.93293
[423,] 45.77456 44.93293
[424,] 45.77456 44.93293
[425,] 45.77456 44.93293
[426,] 45.77456 44.93293
[427,] 45.77456 44.93293
[428,] 45.77456 44.93293
[429,] 45.77456 44.93293
[430,] 45.77456 44.93293
[431,] 45.77456 44.93293
[432,] 45.77456 44.93293
[433,] 45.77456 44.93293
[434,] 45.77456 44.93293
[435,] 45.77456 44.93293
[436,] 45.77456 44.93293
[437,] 45.77456 44.93293
[438,] 45.77456 44.93293
[439,] 45.77456 44.93293
[440,] 45.77456 44.93293
[441,] 45.77456 44.93293
[442,] 45.77456 44.93293
[443,] 45.77456 44.93293
[444,] 45.77456 44.93293
[445,] 45.77456 44.93293
[446,] 45.77456 44.93293
[447,] 45.77456 44.93293
[448,] 45.77456 44.93293
```

```
[449,] 45.77456 44.93293
[450,] 45.77456 44.93293
[451,] 45.77456 44.93293
[452,] 45.77456 44.93293
[453,] 45.77456 44.93293
[454,] 45.77456 44.93293
[455,] 45.77456 44.93293
[456,] 45.77456 44.93293
[457,] 45.77456 44.93293
[458,] 45.77456 44.93293
[459,] 45.77456 44.93293
[460,] 45.77456 44.93293
[461,] 45.77456 44.93293
[462,] 45.77456 44.93293
[463,] 45.77456 44.93293
[464,] 45.77456 44.93293
[465,] 45.77456 44.93293
[466,] 45.77456 44.93293
[467,] 45.77456 44.93293
[468,] 45.77456 44.93293
[469,] 45.77456 44.93293
[470,] 45.77456 44.93293
[471,] 45.77456 44.93293
[472,] 45.77456 44.93293
[473,] 45.77456 44.93293
[474,] 45.77456 44.93293
[475,] 45.77456 44.93293
[476,] 45.77456 44.93293
[477,] 45.77456 44.93293
[478,] 45.77456 44.93293
[479,] 45.77456 44.93293
[480,] 45.77456 44.93293
[481,] 45.77456 44.93293
[482,] 45.77456 44.93293
[483,] 45.77456 44.93293
[484,] 45.77456 44.93293
[485,] 45.77456 44.93293
[486,] 45.77456 44.93293
[487,] 45.77456 44.93293
[488,] 45.77456 44.93293
[489,] 45.77456 44.93293
[490,] 45.77456 44.93293
[491,] 45.77456 44.93293
[492,] 45.77456 44.93293
[493,] 45.77456 44.93293
[494,] 45.77456 44.93293
[495,] 45.77456 44.93293
[496,] 45.77456 44.93293
[497,] 45.77456 44.93293
[498,] 45.77456 44.93293
[499,] 45.77456 44.93293
[500,] 45.77456 44.93293
[501,] 45.77456 44.93293
[502,] 45.77456 44.93293
[503,] 45.77456 44.93293
[504,] 45.77456 44.93293
[505,] 45.77456 44.93293
[506,] 45.77456 44.93293
```

```
[507,] 45.77456 44.93293
[508,] 45.77456 44.93293
[509,] 45.77456 44.93293
[510,] 45.77456 44.93293
[511,] 45.77456 44.93293
[512,] 45.77456 44.93293
[513,] 45.77456 44.93293
[514,] 45.77456 44.93293
[515,] 45.77456 44.93293
[516,] 45.77456 44.93293
[517,] 45.77456 44.93293
[518,] 45.77456 44.93293
[519,] 45.77456 44.93293
[520,] 45.77456 44.93293
[521,] 45.77456 44.93293
[522,] 45.77456 44.93293
[523,] 50.36559 53.50135
[524,] 50.36559 53.50135
[525,] 50.36559 53.50135
[526,] 50.36559 53.50135
[527,] 50.36559 53.50135
[528,] 50.36559 53.50135
[529,] 50.36559 53.50135
[530,] 50.36559 53.50135
[531,] 50.36559 53.50135
[532,] 50.36559 53.50135
[533,] 50.36559 53.50135
[534,] 50.36559 53.50135
[535,] 50.36559 53.50135
[536,] 50.36559 53.50135
[537,] 50.36559 53.50135
[538,] 50.36559 53.50135
[539,] 50.36559 53.50135
[540,] 50.36559 53.50135
[541,] 50.36559 53.50135
[542,] 50.36559 53.50135
[543,] 50.36559 53.50135
[544,] 50.36559 53.50135
[545,] 50.36559 53.50135
[546,] 50.36559 53.50135
[547,] 50.36559 53.50135
[548,] 50.36559 53.50135
[549,] 50.36559 53.50135
[550,] 50.36559 53.50135
[551,] 50.36559 53.50135
[552,] 50.36559 53.50135
[553,] 50.36559 53.50135
[554,] 50.36559 53.50135
[555,] 50.36559 53.50135
[556,] 50.36559 53.50135
[557,] 50.36559 53.50135
[558,] 50.36559 53.50135
[559,] 50.36559 53.50135
[560,] 50.36559 53.50135
[561,] 50.36559 53.50135
[562,] 50.36559 53.50135
[563,] 50.36559 53.50135
[564,] 50.36559 53.50135
```

```
[565,] 50.36559 53.50135
[566,] 50.36559 53.50135
[567,] 50.36559 53.50135
[568,] 50.36559 53.50135
[569,] 50.36559 53.50135
[570,] 50.36559 53.50135
[571,] 50.36559 53.50135
[572,] 50.36559 53.50135
[573,] 50.36559 53.50135
[574,] 50.36559 53.50135
[575,] 50.36559 53.50135
[576,] 50.36559 53.50135
[577,] 50.36559 53.50135
[578,] 50.36559 53.50135
[579,] 50.36559 53.50135
[580,] 50.36559 53.50135
[581,] 50.36559 53.50135
[582,] 50.36559 53.50135
[583,] 50.36559 53.50135
[584,] 50.36559 53.50135
[585,] 50.36559 53.50135
[586,] 50.36559 53.50135
[587,] 50.36559 53.50135
[588,] 50.36559 53.50135
[589,] 50.36559 53.50135
[590,] 50.36559 53.50135
[591,] 50.36559 53.50135
[592,] 50.36559 53.50135
[593,] 50.36559 53.50135
[594,] 50.36559 53.50135
[595,] 50.36559 53.50135
[596,] 50.36559 53.50135
[597,] 50.36559 53.50135
[598,] 50.36559 53.50135
[599,] 50.36559 53.50135
[600,] 50.36559 53.50135
[601,] 50.36559 53.50135
[602,] 50.36559 53.50135
[603,] 50.36559 53.50135
[604,] 50.36559 53.50135
[605,] 50.36559 53.50135
[606,] 50.36559 53.50135
[607,] 50.36559 53.50135
[608,] 50.36559 53.50135
[609,] 50.36559 53.50135
[610,] 50.36559 53.50135
[611,] 50.36559 53.50135
[612,] 50.36559 53.50135
[613,] 50.36559 53.50135
[614,] 50.36559 53.50135
[615,] 50.36559 53.50135
[616,] 50.36559 53.50135
[617,] 50.36559 53.50135
[618,] 50.36559 53.50135
[619,] 50.36559 53.50135
[620,] 50.36559 53.50135
[621,] 50.36559 53.50135
[622,] 50.36559 53.50135
```

```
[623,] 50.36559 53.50135
[624,] 50.36559 53.50135
[625,] 50.36559 53.50135
[626,] 50.36559 53.50135
[627,] 50.36559 53.50135
[628,] 50.36559 53.50135
[629,] 50.36559 53.50135
[630,] 50.36559 53.50135
[631,] 50.36559 53.50135
[632,] 50.36559 53.50135
[633,] 50.36559 53.50135
[634,] 50.36559 53.50135
[635,] 50.36559 53.50135
[636,] 50.36559 53.50135
[637,] 50.36559 53.50135
[638,] 50.36559 53.50135
[639,] 50.36559 53.50135
[640,] 50.36559 53.50135
[641,] 50.36559 53.50135
[642,] 50.36559 53.50135
[643,] 50.36559 53.50135
[644,] 50.36559 53.50135
[645,] 50.36559 53.50135
[646,] 50.36559 53.50135
[647,] 50.36559 53.50135
[648,] 50.36559 53.50135
[649,] 50.36559 53.50135
[650,] 50.36559 53.50135
[651,] 50.36559 53.50135
[652,] 50.36559 53.50135
[653,] 50.36559 53.50135
[654,] 50.36559 53.50135
[655,] 50.36559 53.50135
[656,] 50.36559 53.50135
[657,] 50.36559 53.50135
[658,] 50.36559 53.50135
[659,] 50.36559 53.50135
[660,] 50.36559 53.50135
[661,] 50.36559 53.50135
[662,] 50.36559 53.50135
[663,] 50.36559 53.50135
[664,] 50.36559 53.50135
[665,] 50.36559 53.50135
[666,] 50.36559 53.50135
[667,] 50.36559 53.50135
[668,] 50.36559 53.50135
[669,] 50.36559 53.50135
[670,] 50.36559 53.50135
[671,] 50.36559 53.50135
[672,] 50.36559 53.50135
[673,] 50.36559 53.50135
[674,] 50.36559 53.50135
[675,] 50.36559 53.50135
[676,] 50.36559 53.50135
[677,] 50.36559 53.50135
[678,] 50.36559 53.50135
[679,] 50.36559 53.50135
[680,] 50.36559 53.50135
```

```
[681,] 50.36559 53.50135
[682,] 50.36559 53.50135
[683,] 50.36559 53.50135
[684,] 50.36559 53.50135
[685,] 50.36559 53.50135
[686,] 50.36559 53.50135
[687,] 50.36559 53.50135
[688,] 50.36559 53.50135
[689,] 50.36559 53.50135
[690,] 50.36559 53.50135
[691,] 50.36559 53.50135
[692,] 38.35763 37.72696
[693,] 38.35763 37.72696
[694,] 38.35763 37.72696
[695,] 38.35763 37.72696
[696,] 38.35763 37.72696
[697,] 38.35763 37.72696
[698,] 38.35763 37.72696
[699,] 38.35763 37.72696
[700,] 38.35763 37.72696
[701,] 38.35763 37.72696
[702,] 38.35763 37.72696
[703,] 38.35763 37.72696
[704,] 38.35763 37.72696
[705,] 38.35763 37.72696
[706,] 38.35763 37.72696
[707,] 38.35763 37.72696
[708,] 38.35763 37.72696
[709,] 38.35763 37.72696
[710,] 38.35763 37.72696
[711,] 38.35763 37.72696
[712,] 38.35763 37.72696
[713,] 38.35763 37.72696
[714,] 38.35763 37.72696
[715,] 38.35763 37.72696
[716,] 38.35763 37.72696
[717,] 38.35763 37.72696
[718,] 38.35763 37.72696
[719,] 38.35763 37.72696
[720,] 38.35763 37.72696
[721,] 38.35763 37.72696
[722,] 38.35763 37.72696
[723,] 38.35763 37.72696
[724,] 38.35763 37.72696
[725,] 38.35763 37.72696
[726,] 38.35763 37.72696
[727,] 38.35763 37.72696
[728,] 38.35763 37.72696
[729,] 38.35763 37.72696
[730,] 38.35763 37.72696
[731,] 38.35763 37.72696
[732,] 38.35763 37.72696
[733,] 38.35763 37.72696
[734,] 38.35763 37.72696
[735,] 38.35763 37.72696
[736,] 38.35763 37.72696
[737,] 38.35763 37.72696
[738,] 38.35763 37.72696
```

```
[739,] 38.35763 37.72696
[740,] 38.35763 37.72696
[741,] 38.35763 37.72696
[742,] 38.35763 37.72696
[743,] 38.35763 37.72696
[744,] 38.35763 37.72696
[745,] 38.35763 37.72696
[746,] 38.35763 37.72696
[747,] 38.35763 37.72696
[748,] 38.35763 37.72696
[749,] 38.35763 37.72696
[750,] 38.35763 37.72696
[751,] 38.35763 37.72696
[752,] 38.35763 37.72696
[753,] 38.35763 37.72696
[754,] 38.35763 37.72696
[755,] 38.35763 37.72696
[756,] 38.35763 37.72696
[757,] 38.35763 37.72696
[758,] 38.35763 37.72696
[759,] 38.35763 37.72696
[760,] 38.35763 37.72696
[761,] 38.35763 37.72696
[762,] 38.35763 37.72696
[763,] 38.35763 37.72696
[764,] 38.35763 37.72696
[765,] 38.35763 37.72696
[766,] 38.35763 37.72696
[767,] 38.35763 37.72696
[768,] 38.35763 37.72696
[769,] 38.35763 37.72696
[770,] 38.35763 37.72696
[771,] 38.35763 37.72696
[772,] 38.35763 37.72696
[773,] 38.35763 37.72696
[774,] 38.35763 37.72696
[775,] 38.35763 37.72696
[776,] 38.35763 37.72696
[777,] 38.35763 37.72696
[778,] 38.35763 37.72696
[779,] 38.35763 37.72696
[780,] 38.35763 37.72696
[781,] 38.35763 37.72696
[782,] 38.35763 37.72696
[783,] 38.35763 37.72696
[784,] 38.35763 37.72696
[785,] 38.35763 37.72696
[786,] 38.35763 37.72696
[787,] 38.35763 37.72696
[788,] 38.35763 37.72696
[789,] 38.35763 37.72696
[790,] 38.35763 37.72696
[791,] 38.35763 37.72696
[792,] 38.35763 37.72696
[793,] 38.35763 37.72696
[794,] 38.35763 37.72696
[795,] 38.35763 37.72696
[796,] 38.35763 37.72696
```

```
[797,] 38.35763 37.72696
[798,] 38.35763 37.72696
[799,] 38.35763 37.72696
[800,] 38.35763 37.72696
[801,] 38.35763 37.72696
[802,] 38.35763 37.72696
[803,] 38.35763 37.72696
[804,] 38.35763 37.72696
[805,] 38.35763 37.72696
[806,] 38.35763 37.72696
[807,] 38.35763 37.72696
[808,] 38.35763 37.72696
[809,] 38.35763 37.72696
[810,] 38.35763 37.72696
[811,] 38.35763 37.72696
[812,] 38.35763 37.72696
[813,] 38.35763 37.72696
[814,] 38.35763 37.72696
[815,] 38.35763 37.72696
[816,] 38.35763 37.72696
[817,] 38.35763 37.72696
[818,] 38.35763 37.72696
[819,] 38.35763 37.72696
[820,] 38.35763 37.72696
[821,] 38.35763 37.72696
[822,] 38.35763 37.72696
[823,] 38.35763 37.72696
[824,] 38.35763 37.72696
[825,] 38.35763 37.72696
[826,] 38.35763 37.72696
[827,] 38.35763 37.72696
[828,] 38.35763 37.72696
[829,] 38.35763 37.72696
[830,] 38.35763 37.72696
[831,] 38.35763 37.72696
[832,] 38.35763 37.72696
[833,] 38.35763 37.72696
[834,] 38.35763 37.72696
[835,] 38.35763 37.72696
[836,] 38.35763 37.72696
[837,] 38.35763 37.72696
[838,] 38.35763 37.72696
[839,] 38.35763 37.72696
[840,] 38.35763 37.72696
[841,] 38.35763 37.72696
[842,] 38.35763 37.72696
[843,] 38.35763 37.72696
[844,] 38.35763 37.72696
[845,] 38.35763 37.72696
[846,] 38.35763 37.72696
[847,] 38.35763 37.72696
[848,] 38.35763 37.72696
[849,] 38.35763 37.72696
[850,] 38.35763 37.72696
[851,] 38.35763 37.72696
[852,] 38.35763 37.72696
[853,] 38.35763 37.72696
[854,] 38.35763 37.72696
```

```
[855,] 38.35763 37.72696
[856,] 38.35763 37.72696
[857,] 38.35763 37.72696
[858,] 38.35763 37.72696
[859,] 38.35763 37.72696
[860,] 38.35763 37.72696
[861,] 38.35763 37.72696
[862,] 38.35763 37.72696
[863,] 38.35763 37.72696
[864,] 38.35763 37.72696
[865,] 38.35763 37.72696
[866,] 38.35763 37.72696
[867,] 38.35763 37.72696
[868,] 38.35763 37.72696
[869,] 38.35763 37.72696
[870,] 38.35763 37.72696
[871,] 38.35763 37.72696
[872,] 38.35763 37.72696
[873,] 38.35763 37.72696
[874,] 38.35763 37.72696
[875,] 38.35763 37.72696
[876,] 38.35763 37.72696
[877,] 38.35763 37.72696
[878,] 38.35763 37.72696
[879,] 38.35763 37.72696
[880,] 38.35763 37.72696
[881,] 38.35763 37.72696
[882,] 38.35763 37.72696
[883,] 38.35763 37.72696
[884,] 38.35763 37.72696
[885,] 38.35763 37.72696
[886,] 38.35763 37.72696
[887,] 38.35763 37.72696
[888,] 38.35763 37.72696
[889,] 38.35763 37.72696
[890,] 38.35763 37.72696
[891,] 38.35763 37.72696
[892,] 38.35763 37.72696
[893,] 38.35763 37.72696
[894,] 38.35763 37.72696
[895,] 38.35763 37.72696
[896,] 38.35763 37.72696
[897,] 38.35763 37.72696
[898,] 38.35763 37.72696
[899,] 38.35763 37.72696
[900,] 38.35763 37.72696
[901,] 38.35763 37.72696
[902,] 38.35763 37.72696
[903,] 38.35763 37.72696
[904,] 38.35763 37.72696
[905,] 38.35763 37.72696
[906,] 38.35763 37.72696
[907,] 38.35763 37.72696
[908,] 38.35763 37.72696
[909,] 38.35763 37.72696
[910,] 38.35763 37.72696
[911,] 38.35763 37.72696
[912,] 38.35763 37.72696
```

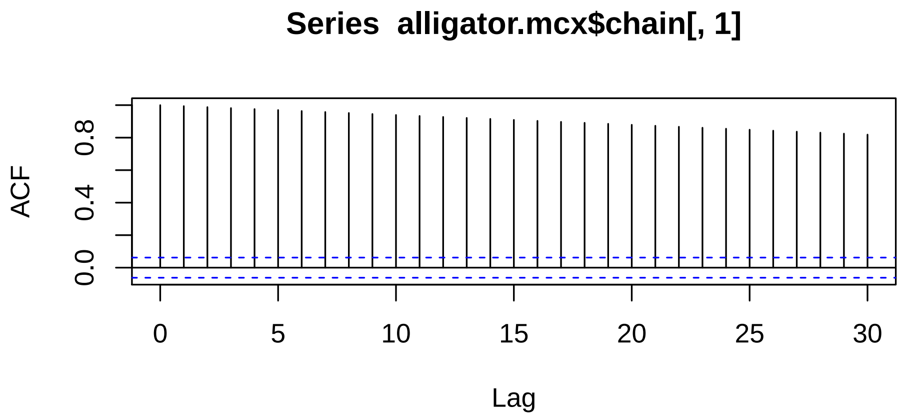
```
[913,] 38.35763 37.72696
[914,] 38.35763 37.72696
[915,] 38.35763 37.72696
[916,] 38.35763 37.72696
[917,] 38.35763 37.72696
[918,] 38.35763 37.72696
[919,] 38.35763 37.72696
[920,] 38.35763 37.72696
[921,] 38.35763 37.72696
[922,] 38.35763 37.72696
[923,] 38.35763 37.72696
[924,] 38.35763 37.72696
[925,] 38.35763 37.72696
[926,] 38.35763 37.72696
[927,] 38.35763 37.72696
[928,] 38.35763 37.72696
[929,] 38.35763 37.72696
[930,] 38.35763 37.72696
[931,] 38.35763 37.72696
[932,] 38.35763 37.72696
[933,] 38.35763 37.72696
[934,] 38.35763 37.72696
[935,] 38.35763 37.72696
[936,] 38.35763 37.72696
[937,] 38.35763 37.72696
[938,] 38.35763 37.72696
[939,] 38.35763 37.72696
[940,] 38.35763 37.72696
[941,] 38.35763 37.72696
[942,] 38.35763 37.72696
[943,] 38.35763 37.72696
[944,] 38.35763 37.72696
[945,] 38.35763 37.72696
[946,] 38.35763 37.72696
[947,] 38.35763 37.72696
[948,] 38.35763 37.72696
[949,] 38.35763 37.72696
[950,] 38.35763 37.72696
[951,] 38.35763 37.72696
[952,] 38.35763 37.72696
[953,] 38.35763 37.72696
[954,] 38.35763 37.72696
[955,] 38.35763 37.72696
[956,] 38.35763 37.72696
[957,] 38.35763 37.72696
[958,] 38.35763 37.72696
[959,] 38.35763 37.72696
[960,] 38.35763 37.72696
[961,] 38.35763 37.72696
[962,] 38.35763 37.72696
[963,] 38.35763 37.72696
[964,] 38.35763 37.72696
[965,] 38.35763 37.72696
[966,] 38.35763 37.72696
[967,] 38.35763 37.72696
[968,] 38.35763 37.72696
[969,] 38.35763 37.72696
[970,] 38.35763 37.72696
```

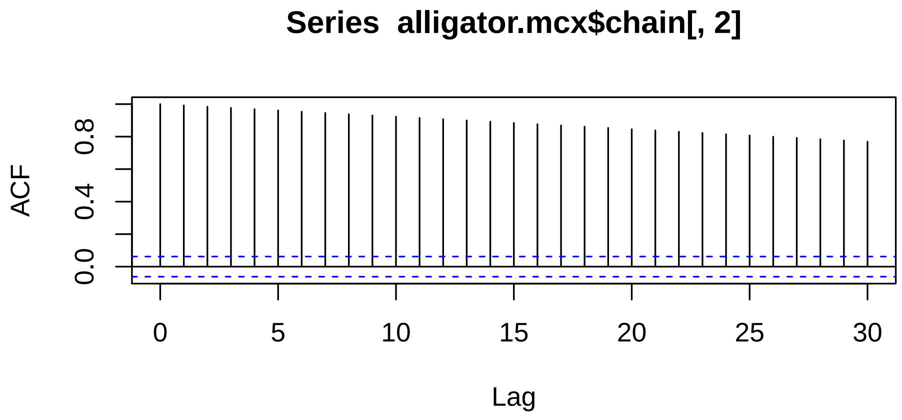
```
 [971,] 38.35763 37.72696
 [972,] 38.35763 37.72696
 [973,] 38.35763 37.72696
 [974,] 38.35763 37.72696
 [975,] 38.35763 37.72696
 [976,] 38.35763 37.72696
 [977,] 38.35763 37.72696
 [978,] 38.35763 37.72696
 [979,] 38.35763 37.72696
 [980,] 38.35763 37.72696
 [981,] 38.35763 37.72696
 [982,] 38.35763 37.72696
 [983,] 38.35763 37.72696
 [984,] 38.35763 37.72696
 [985,] 38.35763 37.72696
 [986,] 38.35763 37.72696
 [987,] 38.35763 37.72696
 [988,] 38.35763 37.72696
 [989,] 38.35763 37.72696
 [990,] 38.35763 37.72696
 [991,] 38.35763 37.72696
 [992,] 38.35763 37.72696
 [993,] 38.35763 37.72696
 [994,] 38.35763 37.72696
 [995,] 38.35763 37.72696
 [996,] 38.35763 37.72696
 [997,] 38.35763 37.72696
 [998,] 38.35763 37.72696
 [999,] 38.35763 37.72696
[1000,] 38.35763 37.72696

$current.batchmean
[1] 0 0

$bmsq
[1] 2.1117 2.1117

$nobatches
[1] 10

$phat
[1] 0.251 0.251

$mcse
[1] 0.1217247 0.1217247

$y1.start
 [1] 0 0 0 8 2 1 0 1 0 0 0 0 2 0 1 7 5 4 6 4 0 0 1 1 1 0 1 9 1 1 5 6 3 0 2 0 1 2
[39] 0 3

$perpos
[1] 0.02

attr(,"class")
[1] "cabSummary"
```

The chain of goodness of fit statistics are saved in `alligator.mcx$chain`. The saved chain is discarded if the simulations are resumed with `update`, even if `savechain = T` when the simulation is resumed.

We would want to look at the autocorrelation function of the goodness of fit statistics.

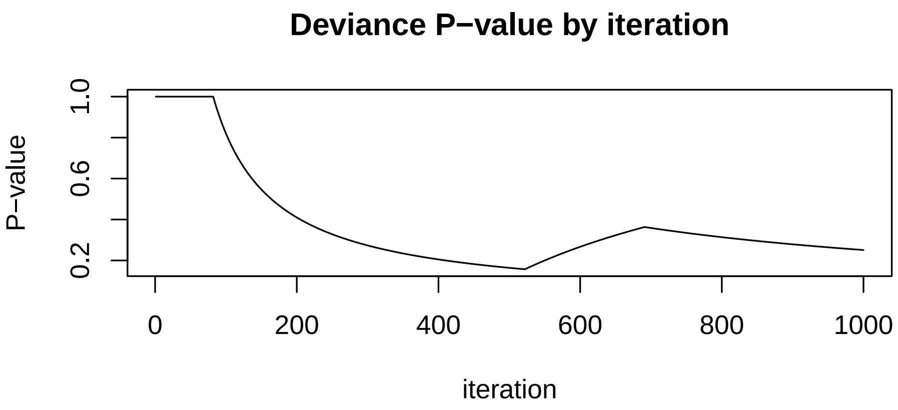
```
> par(mfrow = c(2, 1))
> acf(alligator.mcx$chain[,1])
> acf(alligator.mcx$chain[,2])
```

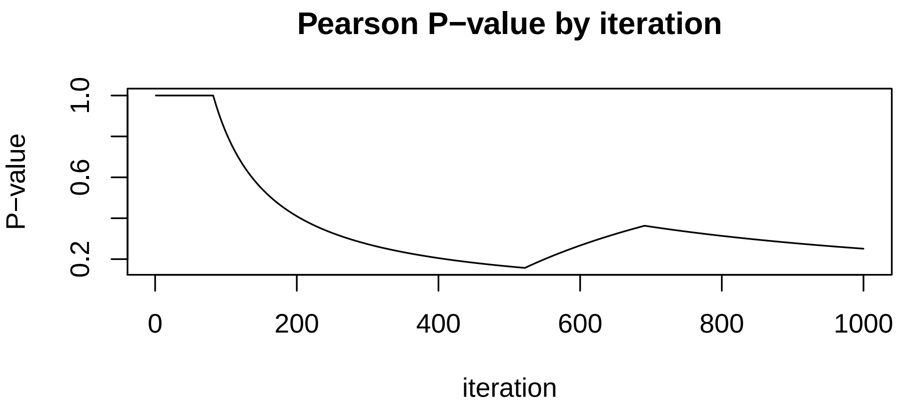




We would also want to look at the chain of P-values.

```
> dev.p <- cumsum(alligator.mcx$chain[,1] >= alligator.mcx$dobs[1]) / (1 : alligator.mcx$nosim)
> pearson.p <- cumsum(alligator.mcx$chain[,1] >= alligator.mcx$dobs[1]) / (1 : alligator.mcx$nosim)
> par(mfrow = c(2, 1))
> plot(dev.p, type = "l", ylab = "P-value", xlab = "iteration")
> title("Deviance P-value by iteration")
> plot(pearson.p, type = "l", ylab = "P-value", xlab = "iteration")
> title("Pearson P-value by iteration")
```





The P-values have apparently not stabilized. Also, there is an extremely slow decay in the autocorrelations of the chain of goodness of fit statistics. Therefore, we should execute a longer run using large batch sizes. While on the subject of batch sizes, note that `mcexact` does not require the total number of simulations to be a multiple of the batch size. If the algorithm terminates in the middle of completing a batch, it is not used in the P-value calculations. However, the simulations are not wasted if the algorithm is resumed with `update`.

One large final run of this data discarding all of the initial tinkering could be performed by setting `flush = TRUE` as an argument to `update`. Here, `flush = TRUE`, tells `update` to throw out all of the data used in the initial tinkering, except that it starts the new chain from the final table from the initial runs. This is a harmless way to burn the chain in while you are tinkering with it. Of course, the chain can be restarted at the default starting value, the observed data, by simply rerunning `mcexact` .

4 Application to Disclosure Limitation

Though there are certainly more rigorous procedures available [see 4], `exactLoglinTest` is a useful tool for exploring disclosure limitation in contingency tables. Consider the Czech Auto Worker's data given in Table 4. Suppose a researcher is concerned about the potential disclosure risk of releasing all two-way marginals from this table. The following code will load the Czech auto worker data into a data frame:

```
> data(czech.dat)
```

We will explore disclosure limitation by simulating tables from the hypergeometric distribution obtained by conditioning on all two way margins. However, we would like to save all of the simulated table entries, not just the deviance and Pearson statistics. This could be accomplished by changing the argument `stat` of `mcexact` to an appropriate statistic. However, the function `simulateConditional` performs this simulation for us. It returns the simulated tables in a matrix with each row being a complete simulated table.

Now we run the chain. Notice the `stat = cell.stat` option to load the newly defined statistic.

```
> chain <- simulateConditional(y ~ (A + B + C + D + E + F) ^ 2,
+                              data = czech.dat,
+                              method = "cab",
+                              nosim = 10 ^ 3,
+                              p = .4)
```

Now, chain is a matrix where each row is a simulated table. We were particularly concerned with cells 39, 48, and 55 which contained only one, two and two individuals respectively. Consider the proportion of tables which have greater than 0 but fewer than three individuals

```
> mean(chain[,39] > 0 & chain[,39] < 3)

[1] 0.389

> mean(chain[,48] > 0 & chain[,58] < 3)

[1] 0.11

> mean(chain[,55] > 0 & chain[,55] < 3)

[1] 0.896
```

We used the model in question because this model fixes all two-way margins. However, that model need not fit the data well (in fact, it doesn't). Therefore, in addition to simulating from the hypergeometric density, a user would likely also want to simulate from other densities, such as a uniform distribution on tables with these margins. Though the normal approximations for exactLoglinTest were tailored specifically to the hypergeometric density, it allows for other target distributions. Here the density must be specified on the log scale up to a constant. Since a uniform density is simply a constant we use a density that always returns 0.

```
> chain2 <- simulateConditional(y ~ (A + B + C + D + E + F) ^ 2,
+                               data = czech.dat,
+                               method = "cab",
+                               nosim = 10 ^ 3,
+                               p = .4,
+                               dens = function(y) 0)
> mean(chain2[,39] > 0 & chain2[,39] < 3)

[1] 0.363

> mean(chain2[,48] > 0 & chain2[,58] < 3)

[1] 0.59

> mean(chain2[,55] > 0 & chain2[,55] < 3)

[1] 1
```

Both simulations suggest that there are plenty of tables with higher counts than the observed counts for cells 39, 48 and 55. Hence the disclosure risk in releasing the two-way marginals seems minimal. However, it should be reiterated that this example is given only to illustrate how to obtain simulated tables from exactLoglinTest, further investigation of the chain and the data would be necessary for a thorough analysis of the disclosure risk.

4.1 Exact Score Test for Binomial Counts

The data given in A are obtained from the Cytel web site[1]. The data cross classify the survival of the Titanic passengers by class, gender and age. You can obtain the data with

```
> data(titanic.dat)
```

[1]http://www.cytel.com/

Following the analysis done at the Cytel web site, we view each person's survival as a binary outcome. We use a model where a person's age, sex and class are additive effects on the logit scale. In the light of the discussion from Subsection 1.1, this model is equivalent to the following:

```
> titanic.dat$alpha <- rep(1 : 16, 2)
> fit <- glm(y ~ (factor(class) + factor(age) + factor(sex)) : factor(surv) +
+            factor(surv) + factor(alpha),
+            family = poisson,
+            data = titanic.dat)
> summary(fit)

Call:
glm(formula = y ~ (factor(class) + factor(age) + factor(sex)):factor(surv) +
    factor(surv) + factor(alpha), family = poisson, data = titanic.dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7995  -1.7072  -0.0003   0.9135   3.5931

Coefficients: (5 not defined because of singularities)
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -18.7133  2170.2682  -0.009    0.993
factor(surv)1                   2.2477     0.2988   7.522 5.40e-14 ***
factor(alpha)2                 16.4218  2170.2684   0.008    0.994
factor(alpha)3                 18.9137  2170.2682   0.009    0.993
factor(alpha)4                 19.6645  2170.2682   0.009    0.993
factor(alpha)5                 19.3346  2170.2682   0.009    0.993
factor(alpha)6                 21.3136  2170.2682   0.010    0.992
factor(alpha)7                 20.6918  2170.2682   0.010    0.992
factor(alpha)8                 21.0027  2170.2682   0.010    0.992
factor(alpha)9                 -0.8226  3182.4092   0.000    1.000
factor(alpha)10                17.6670  2170.2683   0.008    0.994
factor(alpha)11                17.9902  2170.2682   0.008    0.993
factor(alpha)12                18.9552  2170.2682   0.009    0.993
factor(alpha)13                21.7355  2170.2682   0.010    0.992
factor(alpha)14                20.7316  2170.2682   0.010    0.992
factor(alpha)15                19.9737  2170.2682   0.009    0.993
factor(alpha)16                20.3374  2170.2682   0.009    0.993
factor(class)1:factor(surv)0   -0.8577     0.1573  -5.451 5.00e-08 ***
factor(class)2:factor(surv)0    0.1604     0.1738   0.923    0.356
factor(class)3:factor(surv)0    0.9201     0.1486   6.192 5.93e-10 ***
factor(class)1:factor(surv)1        NA         NA      NA       NA
factor(class)2:factor(surv)1        NA         NA      NA       NA
factor(class)3:factor(surv)1        NA         NA      NA       NA
factor(age)1:factor(surv)0      1.0615     0.2440   4.350 1.36e-05 ***
factor(age)1:factor(surv)1          NA         NA      NA       NA
factor(sex)1:factor(surv)0      2.4201     0.1404  17.236  < 2e-16 ***
factor(sex)1:factor(surv)1          NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4953.14  on 31  degrees of freedom
Residual deviance:  112.57  on 10  degrees of freedom
AIC: 283.97

Number of Fisher Scoring iterations: 15
```

The varianble `alpha` is added to correspond to the α_i terms from (2). Consider the gender effect in specific. Here, 2.42 suggests the odds of surviving for a male were roughly 9% that of a female. Furthermore, the estimate is highly significant. To calculate an exact P-value for this problem we use `simulateConditional` to simulate tables conditioning on all of the parameters except the one corresponding to the factor(surv) : factor(sex) interaction.

```
> chain <- simulateConditional(y ~  factor(surv) +
+                              (factor(class) + factor(age)) : factor(surv) +
+                              factor(alpha),
+                              dat = titanic.dat,
+                              nosim = 10 ^ 3,
+                              method = "cab",
+                              p = .1)
```

A P-value for a score test of $H_0 : \gamma = 0$ versus $H_a : \gamma < 0$ simply counts the proportion of tables with sufficient statistic for γ is smaller than the observed value. Using the notation from (2) the sufficient statistic for γ is $s_\gamma = \sum_i z_i y_i \equiv z'y$. We calculate the chain of sufficient statistics and the observed sufficient statistic below.

```
> z <- titanic.dat$sex * titanic.dat$surv
> sgamma <- chain %*% z
> sgamma.obs <- titanic.dat$y %*% z
> mean(sgamma <= sgamma.obs[1])

[1] 0.032
```

Apparently, none of the simulated tables have sufficient statistics for γ below that of the observed, which agrees closely with large sample results above.

5 Discussion and To Do

In this manual we investigated three straightforward examples of `exactLoglinTest` and considered two useful extensions of the program. The program was initially constructed calculate P-values for goodness of fit tests for contingency tables. However, the latter examples suggest a more user friendly interface for those problems would be useful.

Finally, it should be noted that only the inner-most calculations have been migrated to C. Possibly great gains in the speed of the algorithm could be attained by migrating more of the code (or more efficient R coding).

References

[1] Alan Agresti. *Categorical Data Analysis*. Wiley, New York, 1990.

[2] J.G. Booth and R.W. Butler. An importance sampling algorithm for exact conditional test in log-linear models. *Biometrika*, 86:321–332, 1999.

[3] Brian S. Caffo and James G. Booth. A markov chain monte carlo algorithm for approximating exact conditional probabilities. *the Journal of Compuatational and Graphical Statistics*, 10:730–745, 2001.

[4] Adrian Dobra, Claudia Tebaldi, and Mike West. Reconstruction of contingency tables with missing data. Technical report, Duke University, 2002.

[5] D. E. Edwards and T. Havranek. A fast procedure for model search in multidimesional contingency tables. *Biometrika*, 72:339–351, 1985.

[6] Friedrich Leisch. *Sweave User Manual*.

[7] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002.

A Tables

Residence	Residence in 1985			
in 1980	Northeast	Midwest	South	West
Northeast	11,607	100	366	124
Midwest	87	13,677	515	302
South	172	225	17,819	270
West	63	176	286	10,192

Source [1]

Table 1: Residency Data

	Pathologist B				
Pathologist A	1	2	3	4	5
1	22	2	2	0	0
2	5	7	14	0	0
3	0	2	36	0	0
4	0	1	14	7	0
5	0	0	3	0	3

Source [1]

Table 2: Pathologist Agreement Data

			Primary Food Choice				
Lake	Gender	Size	Fish	Invert	Reptile	Bird	Other
1	Male	Small	7	1	0	0	5
	Male	Large	4	0	0	1	2
	Female	Small	16	3	2	2	3
	Female	Large	3	0	1	2	3
2	Male	Small	2	2	0	0	1
	Male	Large	13	7	6	0	0
	Female	Small	3	9	1	0	2
	Female	Large	0	1	0	1	0
3	Male	Small	3	7	1	0	1
	Male	Large	8	6	6	3	5
	Female	Small	2	4	1	1	4
	Female	Large	0	1	0	0	0
4	Male	Small	13	10	0	2	2
	Male	Large	9	0	0	1	2
	Female	Small	3	9	1	0	1
	Female	Large	8	1	0	0	1

Source [1]
Model (FG, FL, FS, LGS) where F=food choice, L=lake, S=size, G=gender.

Table 3: Alligator Data

				B	no		yes	
F	E	D	C	A	no	yes	no	yes
neg	small	small	no		44	40	112	67
			yes		129	145	12	23
		large	no		35	12	80	33
			yes		109	67	7	9
	large	small	no		23	32	70	66
			yes		50	80	7	13
		large	no		24	25	73	57
			yes		51	63	7	16
pos	small	small	no		5	7	21	9
			yes		9	17	1	4
		large	no		4	3	11	8
			yes		14	17	5	2
	large	small	no		7	3	14	14
			yes		9	16	2	3
		large	no		4	0	13	11
			yes		5	14	4	4

Source [4] originally appeared in [5].

Table 4: Czech Auto Workers Data

			Class			
Surv	Sex	Age	Crew	First	Second	Third
no	F	Child	0	0	0	17
		Adult	3	4	13	89
	M	Child	0	0	0	35
		Adult	670	118	154	387
yes	F	Child	0	1	13	14
		Adult	20	140	80	76
	M	Child	0	5	11	13
		Adult	192	57	14	75