Teratology - Alternative Binary Models

December 4, 2025

First the teratology data are loaded. The data set exists in two different versions, "teratology" shows the aggregated count data whereas "teratology2" includes the original data.

```
library(catdata)

## Lade nötiges Paket:  MASS

data(teratology)
data(teratology2)
```

For the first two models, the simple independence model and the quasi-likelihood model, the aggregated data are needed.

```
attach(teratology)
```

The simple and naive independence model is fitted by the following command.

```
mLogit <- glm(cbind(D,L) ~ as.factor(Grp), family=binomial())
summary(mLogit)

##
## Call:
## glm(formula = cbind(D, L) ~ as.factor(Grp), family = binomial())
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.1440     0.1292   8.855  < 2e-16 ***
## as.factor(Grp)2  -3.3225     0.3308 -10.043  < 2e-16 ***
## as.factor(Grp)3  -4.4762     0.7311  -6.122 9.22e-10 ***
## as.factor(Grp)4  -4.1297     0.4762  -8.672  < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 509.43  on 57  degrees of freedom
```

```
## Residual deviance: 173.45  on 54  degrees of freedom
## AIC: 252.92
##
## Number of Fisher Scoring iterations: 5
```

Now the quasi–likelihood model is fitted. The coefficients are the same as in the independence model before, only the standard errors have to be multiplied by $\sqrt{\hat{\phi}}$.

```
mQuasi <- glm(cbind(D,L) ~ as.factor(Grp), family=quasibinomial(link="logit"))
summary(mQuasi)

##
## Call:
## glm(formula = cbind(D, L) ~ as.factor(Grp), family = quasibinomial(link = "logit"))
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.1440     0.2187   5.231 2.81e-06 ***
## as.factor(Grp)2    -3.3225     0.5600  -5.933 2.18e-07 ***
## as.factor(Grp)3    -4.4762     1.2375  -3.617 0.000656 ***
## as.factor(Grp)4    -4.1297     0.8061  -5.123 4.14e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 2.864945)
##
##     Null deviance: 509.43  on 57  degrees of freedom
## Residual deviance: 173.45  on 54  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

The next model to be fitted is a GEE with independence correlation structure. For that purpose the library ”gee” is loaded.

```
library(gee)

## Warning:  Paket ’gee’ wurde unter R Version 4.5.2 erstellt
```

Now we use the original data set ”teratology2”.

```
detach(teratology)
attach(teratology2)
```

The GEE is fitted by the following command. The coefficients are again equal to those from the independence model, the standard errors for the independence models can be found in the column ”Naive S.E.”. The new standard errors from the GEE are those in the column ”Robust S.E.”.

```
mGee <- gee(y ~ as.factor(Grp), id=Rat, family=binomial)

## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate

##      (Intercept) as.factor(Grp)G2 as.factor(Grp)G3
##         1.143981        -3.322513        -4.476184
## as.factor(Grp)G4
##        -4.129663

summary(mGee)

##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Logit
##  Variance to Mean Relation: Binomial
##  Correlation Structure:     Independent
##
## Call:
## gee(formula = y ~ as.factor(Grp), id = Rat, family = binomial)
##
## Summary of Residuals:
##         Min          1Q      Median          3Q         Max
## -0.75840979 -0.10169492 -0.03448276  0.24159021  0.96551724
##
##
## Coefficients:
##                   Estimate Naive S.E.    Naive z
## (Intercept)       1.143981  0.1296195   8.825688
## as.factor(Grp)G2 -3.322513  0.3319396 -10.009393
## as.factor(Grp)G3 -4.476185  0.7335486  -6.102098
## as.factor(Grp)G4 -4.129663  0.4778032  -8.643020
##                  Robust S.E.   Robust z
## (Intercept)        0.2758667   4.146861
## as.factor(Grp)G2   0.4400582 -7.550168
## as.factor(Grp)G3   0.6104577 -7.332507
## as.factor(Grp)G4   0.5763810 -7.164814
##
## Estimated Scale Parameter:  1.006633
## Number of Iterations:  1
##
## Working Correlation
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    0    0    0    0    0    0    0    0     0
##  [2,]    0    1    0    0    0    0    0    0    0     0
##  [3,]    0    0    1    0    0    0    0    0    0     0
##  [4,]    0    0    0    1    0    0    0    0    0     0
```

```
##  [5,]    0    0    0    0    1    0    0    0    0     0
##  [6,]    0    0    0    0    0    1    0    0    0     0
##  [7,]    0    0    0    0    0    0    1    0    0     0
##  [8,]    0    0    0    0    0    0    0    1    0     0
##  [9,]    0    0    0    0    0    0    0    0    1     0
## [10,]    0    0    0    0    0    0    0    0    0     1
## [11,]    0    0    0    0    0    0    0    0    0     0
## [12,]    0    0    0    0    0    0    0    0    0     0
## [13,]    0    0    0    0    0    0    0    0    0     0
## [14,]    0    0    0    0    0    0    0    0    0     0
## [15,]    0    0    0    0    0    0    0    0    0     0
## [16,]    0    0    0    0    0    0    0    0    0     0
## [17,]    0    0    0    0    0    0    0    0    0     0
##       [,11] [,12] [,13] [,14] [,15] [,16] [,17]
##  [1,]     0     0     0     0     0     0     0
##  [2,]     0     0     0     0     0     0     0
##  [3,]     0     0     0     0     0     0     0
##  [4,]     0     0     0     0     0     0     0
##  [5,]     0     0     0     0     0     0     0
##  [6,]     0     0     0     0     0     0     0
##  [7,]     0     0     0     0     0     0     0
##  [8,]     0     0     0     0     0     0     0
##  [9,]     0     0     0     0     0     0     0
## [10,]     0     0     0     0     0     0     0
## [11,]     1     0     0     0     0     0     0
## [12,]     0     1     0     0     0     0     0
## [13,]     0     0     1     0     0     0     0
## [14,]     0     0     0     1     0     0     0
## [15,]     0     0     0     0     1     0     0
## [16,]     0     0     0     0     0     1     0
## [17,]     0     0     0     0     0     0     1
```

For the following beta–binomial model the library ”VGAM” with its function ”vglm” and the data set ”teratology” is needed.

```
library(VGAM)

## Lade nötiges Paket:  stats4
## Lade nötiges Paket:  splines
```

```
detach(teratology2)
attach(teratology)
```

Furthermore we construct the Variable N as sum of all fetuses in one litter. We will use N to make a subset with $N > 1$ for the beta–binomial model.

```
N <- D + L
```

Now the beta–binomial model is fitted.

```
mBetaBin <- vglm(cbind(D,L) ~ as.factor(Grp), family=betabinomial, subset=N>1)
summary(mBetaBin)

##
## Call:
## vglm(formula = cbind(D, L) ~ as.factor(Grp), family = betabinomial,
##     subset = N > 1)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept):1       1.3451     0.2444   5.505 3.70e-08 ***
## (Intercept):2      -1.1402     0.3234  -3.525 0.000423 ***
## as.factor(Grp)2  -3.0868     0.5209  -5.926 3.10e-09 ***
## as.factor(Grp)3  -3.8647     0.8634  -4.476 7.61e-06 ***
## as.factor(Grp)4  -3.9197     0.6838  -5.732 9.92e-09 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(mu), logitlink(rho)
##
## Log-likelihood: -93.2973 on 109 degrees of freedom
##
## Number of Fisher scoring iterations: 8
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## 'as.factor(Grp)3', 'as.factor(Grp)4'
```

For the following two mixed models again the original data are required.

```
detach(teratology)
attach(teratology2)
```

With the function ”glmmPQL” from the ”MASS”–library a mixed model is fitted by penalized quasi–likelihood, the mixed model contains random intercepts but no random slopes.

```
mMixPql<- glmmPQL(y ~ as.factor(Grp), random=~1 | Rat, family=binomial)

## Iteration 1
## Iteration 2
## Iteration 3
## Iteration 4
## Iteration 5
## Iteration 6

summary(mMixPql)

## Linear mixed-effects model fit by maximum likelihood
##   Data: NULL
```

```
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | Rat
##         (Intercept)  Residual
## StdDev:    1.455986 0.8006729
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects:  y ~ as.factor(Grp)
##                      Value Std.Error  DF   t-value p-value
## (Intercept)       1.687058 0.3057620 549  5.517553       0
## as.factor(Grp)G2 -4.130280 0.6141687  54 -6.724993       0
## as.factor(Grp)G3 -5.274063 0.9808235  54 -5.377178       0
## as.factor(Grp)G4 -5.109542 0.7474739  54 -6.835747       0
##  Correlation:
##                  (Intr) a.(G)G2 a.(G)G3
## as.factor(Grp)G2 -0.498
## as.factor(Grp)G3 -0.312  0.155
## as.factor(Grp)G4 -0.409  0.204   0.128
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -3.5217300 -0.3475892 -0.1548062  0.2821502  5.1647080
##
## Number of Observations: 607
## Number of Groups: 58
```

In order to fit a mixed model by maximum likelihood we load the library "glmmML".

```
library(glmmML)

## Warning:  Paket 'glmmML' wurde unter R Version 4.5.2 erstellt
```

For a mixed model to be fitted by Gauss–Hermite quadrature we need the function "glmmML" with the option "method='ghq'", the favoured number of quadrature points is determined by the option "n.points".

```
mGaussH <- glmmML(y ~ as.factor(Grp), cluster=Rat, method = "ghq", n.points = 14,
                  boot = 0)
summary(mGaussH)

##
## Call:  glmmML(formula = y ~ as.factor(Grp), cluster = Rat, method = "ghq",      n.point
##
##
##                    coef se(coef)      z Pr(>|z|)
```

```
## (Intercept)       1.802   0.3623  4.976 6.50e-07
## as.factor(Grp)G2 -4.515   0.7360 -6.134 8.58e-10
## as.factor(Grp)G3 -5.855   1.1899 -4.921 8.63e-07
## as.factor(Grp)G4 -5.594   0.9191 -6.087 1.15e-09
##
## Scale parameter in mixing distribution:  1.533 gaussian
## Std. Error:                              0.2878
##
##         LR p-value for H_0: sigma = 0:  2.7e-15
##
## Residual deviance: 435.6 on 602 degrees of freedom  AIC: 445.6
```

Again we change the data set, for the discrete mixture model ”teratology” is required.

```
detach(teratology2)
attach(teratology)
```

For discrete mixture models the library ”flexmix” with its functions ”flexmix” and ”stepFlexmix” can be used.

```
library(flexmix)

## Warning:  Paket 'flexmix' wurde unter R Version 4.5.2 erstellt
## Lade nötiges Paket:  lattice
```

In ”stepFlexmix” the procedure is run several times, the maximum likelihood solution is returned. The favoured number of iterations can be specified by the option ”nrep”. The number of components is determined by the option ”k”.

Due to random processes the results of different runs of ”stepFlexmix” will differ slightly.

```
detach(package:VGAM)
library(stats4)
```

```
mDiscmix <-stepFlexmix(cbind(D,L) ~ 1, k = 2, nrep=5,
                 model = FLXMRglmfix(family = "binomial",fixed =~as.factor(Grp)))

## 2 : * * * * *

summary(mDiscmix)

##
## Call:
## stepFlexmix(cbind(D, L) ~ 1, model = FLXMRglmfix(family = "binomial",
##     fixed = ~as.factor(Grp)), k = 2, nrep = 5)
##
##        prior size post>0 ratio
## Comp.1   0.7   43     53 0.811
```

```
## Comp.2   0.3   15     53 0.283
##
## 'log Lik.' -92.89085 (df=6)
## AIC: 197.7817   BIC: 210.1444
```

```
parameters(mDiscmix)
```

```
##                          Comp.1     Comp.2
## coef.as.factor(Grp)2 -4.310938 -4.3109375
## coef.as.factor(Grp)3 -5.510352 -5.5103523
## coef.as.factor(Grp)4 -5.083383 -5.0833831
## coef.(Intercept)      2.461390 -0.2096271
```