Some aspects of practical data analysis of "glm-type" data using R.

Søren Højsgaard
Biometry Research Unit
Danish Institute of Agricultural Sciences
sorenh@agrsci.dk

October 27, 2004

Contents

1 Introduction

This note illustrates some aspects of practical data analysis of "glm-type" data using R. Thus we do not claim that this is THE appropriate analysis of the specific data. In the note, R code is put into boxes with thick lines while outout is put into boxes with thin lines, e.g.

```
>x <- 1:10
>x
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

1.1 R Packages used

- In this note we use the package `geepack` which are on CRAN, `www.R-project.org`.
- In addition we use the package `doBy` which is NOT on CRAN. `doBy` can be downloaded from `http://genetics.agrsci.dk/~sorenh/misc`.
- Finally we use the `pda` package (which is also not on CRAN, but available from Brian Yandells homepage, `http://www.stat.wisc.edu/~yandell/`.) Note that `pda` is planned to go on CRAN (hopefully later this year). The future of `doBy` is more uncertain. Perhaps someone will incorporate in one of these general utilities packages on CRAN.

The `doBy` package contains various facilities for making group wise plots (surely other – and better – facilities are available for this in R. We have just found that the facilities in `doBy` are simple to use for novices.) In addition the package contains facilities for working with generalized estimating equations (GEE)s through the function `geeglm()`. This function is not much more than a "glm–like" front end to the `geese()` function in the `geepack` package. Finally the package contains the `esticon()` function by which one can specify quite general contrasts. The `pda` package contains lots of material. Here we only focus on using the `lsmean()` function.

1.2 Pig growth – `dietox`

The data used here are described by Lauridsen, Højsgaard, and Sørensen (1999) and contains growth data for a pig feeding experiment. Data is available as the `dietox` data set in the `doBy` package for `R`.

One of the questions asked in connection with the experiment was whether copper added to pig feed increase/decrease growth. Copper (hereafter abbreviated Cu) was used in three levels Cu=1: No copper, Cu=2: 35 mg/kg feed and Cu=3: 175 mg/kg feed. Here we shall analyze data as if they were layed out as a factorial experiment (even though the design was a (almost) balanced incomplete block design – because there is an issue of a litter effect). The weight of slaughter pigs were measured weekly over a 12 week period.

2 Loading data

Data can be loaded as:

```
>library(doBy)
>data(dietox)
>dietox[1:5, ]
```

```
    Weight      Feed Time  Pig Evit Cu Litter
1 26.50000        NA    1 4601    1  1      1
2 27.59999  5.200005    2 4601    1  1      1
3 36.50000 17.600000    3 4601    1  1      1
4 40.29999 28.500000    4 4601    1  1      1
5 49.09998 45.200001    5 4601    1  1      1
```

Cu is coded with levels 1,2 and 3 meaning that `R` will regard Cu as a numeric variable, which it is not. To turn Cu into a factor we do:

```
>dietox$Cu <- as.factor(dietox$Cu)
```

Note: If instead data was saved as a comma–searated file (a .csv file) it could be loaded as

```
>dietox <- read.csv("dietox.csv")
```

3 Looking at data

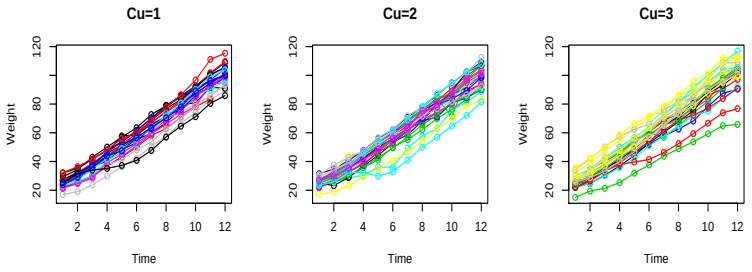

Figure 1: FILENAME: dietox01

The weight as function of time is shown in Figure 1. which suggests

- Approximately linear growth curves
- Some tendency for variance to increase with mean

This plot is produced using the `plotBy()` function in the `doBy` package as follows: First, make space for 1 row and 3 columns of plots:

```
>par(mfrow = c(1, 3))
```

Then call the `plotBy()` function:

```
>plotBy(Weight ~ Time, subject = Pig, group = Cu, title = "Cu=",
     data = dietox, col = 1:100, lines = T)
```

3.1 Looking at the growth curve

Next we calculate the mean and variance for each combination of Cu and Time using the `summaryBy()` function, which is also in the `doBy` package:

```
>m.dietox <- summaryBy(Weight ~ Cu + Time, data = dietox,
     FUN = c(mean, var))
>m.dietox[1:5, ]
```

```
  Cu Time mean.Weight var.Weight
1  1    1    25.34782  12.264414
2  2    1    25.50000   9.656658
3  3    1    26.14999  18.799118
4  1    2    29.48695  15.213879
5  2    2    29.33599  16.070716
```

Figure 2 gives an idea of the growth curves. Figure 2 suggests that

- Growth is not quite linear. The curves are curved (slightly S–shaped)!
- If there is a treatment effect, then it is small!

The plot is produced by:

```
>par(mfrow = c(1, 1))
>plotBy(mean.Weight ~ Time, subject = Cu, data = m.dietox,
     lines = T, col = c("black", "red", "green"), silent = F)
```

```
  symbol colour group subject line
1      1  black  .by.       1    1
2      1    red  .by.       2    1
3      1  green  .by.       3    1
```

3.2 Modelling the mean structure

Based on Figure 2 we fit polynomial models to the means as

```
>lm1 <- lm(mean.Weight ~ Cu * Time, data = m.dietox)
>lm2 <- lm(mean.Weight ~ Cu * (Time + I(Time^2)), data = m.dietox)
>lm3 <- lm(mean.Weight ~ Cu * (Time + I(Time^2) + I(Time^3)),
     data = m.dietox)
```

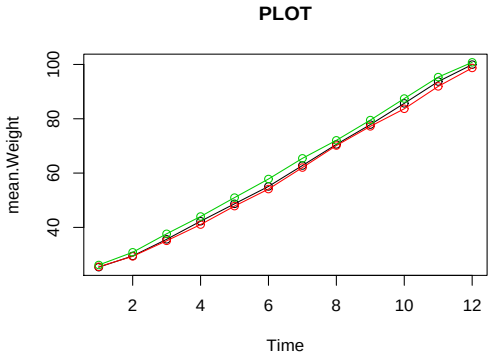

Figure 2: FILENAME: dietox01mean

- and plot the residuals (Figure 3):

```
>par(mfrow = c(1, 3))
>plotBy(resid(lm1) ~ Time, subject = Cu, data = m.dietox,
     lines = T, col = 1:3)
>plotBy(resid(lm2) ~ Time, subject = Cu, data = m.dietox,
     lines = T, col = 1:3)
>plotBy(resid(lm3) ~ Time, subject = Cu, data = m.dietox,
     lines = T, col = 1:3)
```

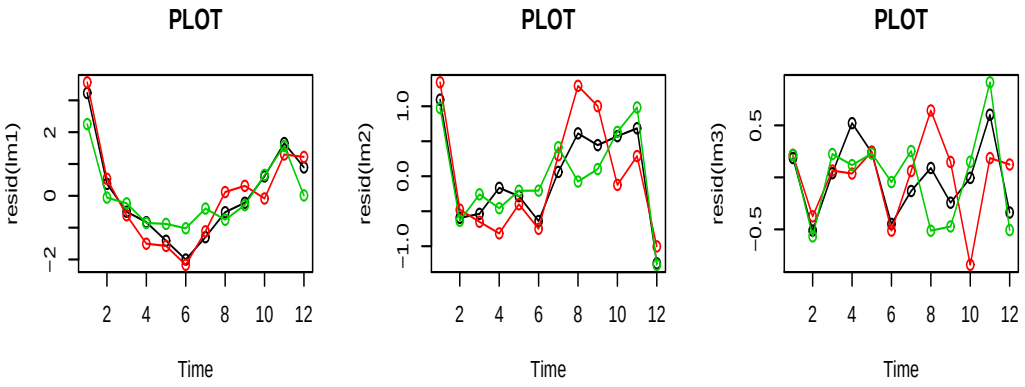

Figure 3: FILENAME: dietox-residplots01

The residual plots comfirm the S–shaped curve: A 3rd degree polynomial is needed to remove systematic patterns in the residuals. Inspired by this we fit the same model to the original data:

```
>mf <- formula(Weight ~ Cu * (Time + I(Time^2) + I(Time^3)))
>lm4 <- lm(mf, data = dietox)
>plotBy(resid(lm4) ~ Time, subject = Pig, data = dietox,
     lines = T, col = 1:3)
```

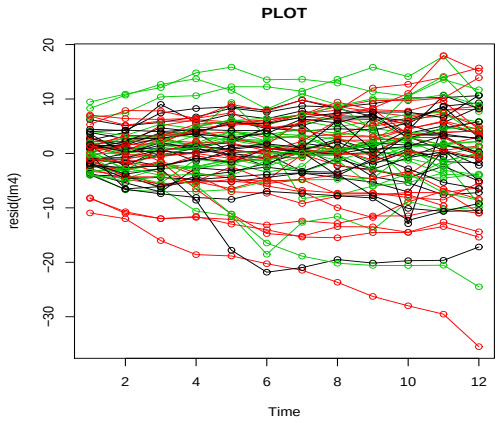

Figure 4: FILENAME: dietox-residplots02

Figure 4 shows that a 3rd degree polynomial seems to remove all systematic effects, but also that the variance increases with time (and hence with the mean). Finally the plot shows that measurements on the same animal tend to be positively correlated.

3.3 Investigating the relationship between the mean and variance

Next we can plot the log variance against the log mean and fit a straight line to these data (see Figure 5):

```
>plot(log(var.Weight) ~ log(mean.Weight), data = m.dietox)
>l <- lm(log(var.Weight) ~ log(mean.Weight), data = m.dietox)
>abline(l, lwd = 2, col = "red")
>l
```

```
Call:
lm(formula = log(var.Weight) ~ log(mean.Weight), data = m.dietox)

Coefficients:
     (Intercept)  log(mean.Weight)
          -1.134             1.212
```

The plot suggests that

$$\log Var(y) \approx a + b \log E(y) \tag{1}$$

and hence

$$Var(y) \approx e^a \cdot E(y)^b$$

The slope b is about one suggesting that the variance is approximately proportional to the mean.

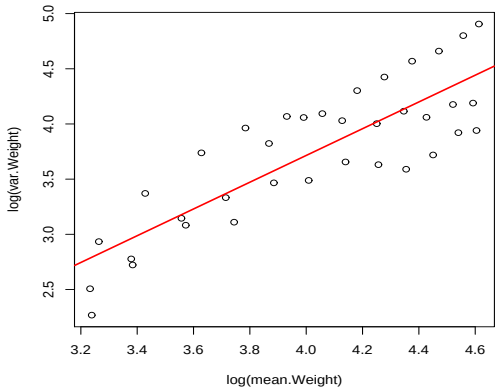

Figure 5: FILENAME: dietox-010

Note that we can calculate the mean and variance for each combination of time and Cu in this example since there are many observations for each combination of `Time x Cu`. In situations where this is not the case an alternative idea can be used: We found that a 3rd degree polynomial seems to remove practically all systematic variation from data. Let $\hat{\mu}$ and e denote the fitted values and the residuals under the 3rd degree model `lm4`. Then we can plot $\log e^2$ against $\log \hat{\mu}$:

```
>plot(log(fitted(lm4)), log(resid(lm4)^2))
>l <- lm(log(resid(lm4)^2) ~ log(fitted(lm4)))
>abline(l, col = "red")
>l
```

```
Call:
lm(formula = log(resid(lm4)^2) ~ log(fitted(lm4)))

Coefficients:
     (Intercept)  log(fitted(lm4))
          -2.611             1.188
```

The idea behind doing so is as follows: The residuals $e_i = y_i - \hat{\mu}_i$ have mean $E(e_i) = 0$ and the variance is $Var(e_i) \approx Var(y_i)$. Now since $E(e_i) = 0$ we have $Var(e_i) = E(e_i^2)$. A simple estimate of $E(e_i^2)$ is e_i^2. So discovering an (approximately) linear relationship between $\log e_i^2$ and $\log \hat{\mu}_i$ suggests the variance function in (1). Above we find that the slope is aboout 1 in accordance with the previous findings. The plot is in Figure 6.

4 Fitting a gee model

Based on the previous findings interest is in fitting a model which:

1. Describes the mean structure (and we have already seen that a 3rd degree polynomial could be a good starting point).

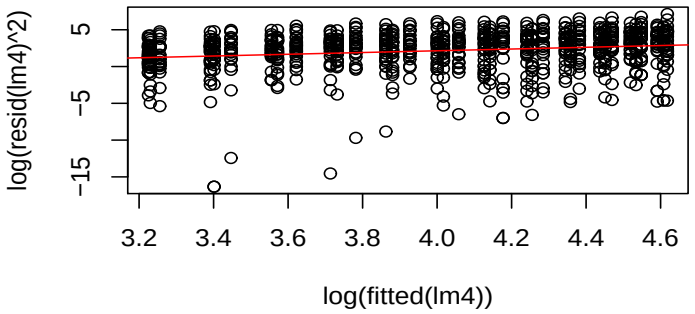

Figure 6: FILENAME: dietox-varmeanplots01

2. Accounts for that the variance is approximately proportional to the mean.
3. Accounts for that there are repeated measurements on the same animal

One way to meet these three requirements is to fit a gee model:

```
>gee1 <- geeglm(mf, data = dietox, id = Pig, family = poisson("identity"),
     corstr = "ar1")
```

For comparison we fit a quasi–poisson model which only accounts for 1. and 2. above as

```
>qpo1 <- glm(mf, data = dietox, family = quasipoisson("identity"))
```

It is informative to compare the regression coefficients and the models and the standard errors of the 1) gee model, 2) the quasi–poisson model and 3) the linear model:

```
>sgee1 <- summary(gee1)
>sqpo1 <- summary(qpo1)
>slm4 <- summary(lm4)
>Egee <- sgee1$coef[, 1]
>Eqpo <- sqpo1$coef[, 1]
>Elm <- slm4$coef[, 1]
>SEgee <- sgee1$coef[, 2]
>SEqpo <- sqpo1$coef[, 2]
>SElm <- slm4$coef[, 2]
>Rgee <- sgee1$coef[, 2]/slm4$coef[, 2]
>Rqpo <- sqpo1$coef[, 2]/slm4$coef[, 2]
>round(cbind(Egee, Eqpo, Elm, SEgee, SEqpo, SElm, Rgee,
     Rqpo), 3)
```

```
                Egee   Eqpo    Elm SEgee SEqpo  SElm  Rgee  Rqpo
(Intercept)   21.858 21.140 21.152 0.693 1.726 2.395 0.289 0.721
Cu2            0.527  0.879  0.699 0.941 2.389 3.316 0.284 0.720
Cu3            0.043 -0.004 -0.042 1.036 2.433 3.350 0.309 0.726
Time           2.885  3.560  3.553 0.360 1.249 1.535 0.235 0.814
I(Time^2)      0.614  0.471  0.472 0.070 0.237 0.270 0.260 0.880
I(Time^3)     -0.026 -0.018 -0.018 0.004 0.013 0.014 0.279 0.931
Cu2:Time      -0.405 -0.804 -0.662 0.637 1.724 2.123 0.300 0.812
Cu3:Time       0.857  0.888  0.918 0.609 1.761 2.146 0.284 0.820
Cu2:I(Time^2)  0.018  0.119  0.093 0.115 0.327 0.372 0.310 0.878
Cu3:I(Time^2) -0.096 -0.099 -0.105 0.109 0.334 0.377 0.291 0.887
Cu2:I(Time^3)  0.001 -0.006 -0.005 0.006 0.018 0.019 0.317 0.928
Cu3:I(Time^3)  0.003  0.003  0.003 0.006 0.018 0.019 0.294 0.938
```

The output shows that 1) the parameter estimates are almost identical whereas 2) the standard errors differ quite a bit. `Rgee` gives the ratio between the standard errors for the gee model and the linear model while `Rqpo` gives the ratio between the standard errors for the quasi poisson model and the linear model. The gee model reduces the standard error to about 0.3 of the standard errors for the linear model whereas the quasi poisson model reduces the standard error to about 0.8 of the standard errors for the linear model.

So what can be concluded from this: We have three different models (or perhaps more appropriately: three different estimation methods) which produce practically the same parameter estimates but with markedly different standard errors of the estimates?

We claim that the standard errors for the gee model are the most reliable, i.e. the other two models overestimate the standard error. The reason being that the gee model adjusts for the fact that measurements at two different time points on the same pig tend to be more alike than on two different pigs. We shall return to this in Section A.

5 Model selection

We proceed by adding terms sequentially to the models

```
>anovalm4 <- anova(lm4)
>anovaqpo1 <- anova(qpo1, test = "F")
>anovagee1 <- anova(gee1)
```

```
>anovalm4
```

```
Analysis of Variance Table

Response: Weight
             Df Sum Sq Mean Sq    F value    Pr(>F)
Cu            2    980     490     9.9751 5.224e-05 ***
Time          1 492513  492513 10028.0637 < 2.2e-16 ***
I(Time^2)     1   1014    1014    20.6385 6.350e-06 ***
I(Time^3)     1    294     294     5.9771   0.01470 *
Cu:Time       2     35      17     0.3547   0.70152
Cu:I(Time^2)  2     52      26     0.5266   0.59080
Cu:I(Time^3)  2      8       4     0.0828   0.92057
Residuals   849  41697      49
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

```
>anovaqpo1
```

```
Analysis of Deviance Table

Model: quasipoisson, link: identity

Response: Weight

Terms added sequentially (first to last)


             Df Deviance Resid. Df Resid. Dev          F    Pr(>F)
NULL                          860     9105.9
Cu            2     16.1       858     9089.8    10.2481 4.001e-05
Time          1   8362.8       857      727.0 10656.8815 < 2.2e-16
I(Time^2)     1     27.8       856      699.2    35.4091 3.906e-09
I(Time^3)     1      5.8       855      693.4     7.3664   0.00678
Cu:Time       2      1.2       853      692.2     0.7735   0.46171
Cu:I(Time^2)  2      1.2       851      691.0     0.7794   0.45903
Cu:I(Time^3)  2      0.2       849      690.8     0.1281   0.87979

NULL
Cu           ***
Time         ***
I(Time^2)    ***
I(Time^3)    **
Cu:Time
Cu:I(Time^2)
Cu:I(Time^3)
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

```
>anovagee1
```

```
Analysis table for GEE models

                 X2.stat DF Pr(>|X^2|)
Cu             1.5221015  2    0.46718
Time        6905.6249513  1    0.00000
I(Time^2)     89.0758061  1    0.00000
I(Time^3)     61.6253666  1    0.00000
Cu:Time        0.9430137  2    0.62406
Cu:I(Time^2)   5.8956159  2    0.05245
Cu:I(Time^3)   2.1170065  2    0.34697
```

We see that only under the gee model any treatment effect comes near to being significant. This is closely related to that the standard errors of the parameter estimates are smallest under the gee model. Lauridsen et al. (1999) find, by a different analysis (random regression) that there is a statistically significant effect of Cu – but the effect is clearly very small.

Dropping the highest order term from the model gives the model gee2 below, and this model will be our focus in the following:

```
>gee2 <- update(gee1, . ~ . - Cu:I(Time^3))
>summary(gee2)
```

```
Call:
geeglm(formula = Weight ~ Cu + Time + I(Time^2) + I(Time^3) +
    Cu:Time + Cu:I(Time^2), family = poisson("identity"), data = dietox,
    id = Pig, corstr = "ar1")

 Coefficients:
                 Estimate    RobustSE          X^2    P(>X^2)
(Intercept)   21.78946075 0.709163599 944.06107981 0.00000000
Cu2            0.56048295 0.926072249   0.36629844 0.54502888
Cu3            0.21255942 1.033674186   0.04228569 0.83707617
Time           2.96637338 0.304312314  95.01946433 0.00000000
I(Time^2)      0.59484762 0.047701492 155.50613493 0.00000000
I(Time^3)     -0.02522423 0.002456949 105.40078602 0.00000000
Cu2:Time      -0.44639624 0.419359426   1.13310002 0.28711506
Cu3:Time       0.65091898 0.379920208   2.93541210 0.08665655
Cu2:I(Time^2)  0.02803206 0.030587932   0.83986545 0.35943526
Cu3:I(Time^2) -0.04752989 0.027632755   2.95859257 0.08542227

 Estimated Scale Parameters:
             estimate    san.se     wald            p
(Intercept) 0.7752835 0.1434884 29.19360 6.549508e-08

Correlation Structure:     ar1

Estimated Correlation Parameters:
       estimate      san.se     wald p
alpha 0.9572027 0.009545219 10056.25 0
Number of clusters:   72   Maximum cluster size: 12
```

6 Estimating contrasts

Suppose first we want to estimate the predicted value for Cu=2 and Cu=3 at Time=7. The traditional way of doing this in R is by using the `predict()` function:

```
>dnew <- data.frame(Time = c(7, 7), Cu = as.factor(c(2,
     3)))
>predict(gee2, dnew)
```

```
       1        2
61.85898 65.48972
```

The difficulty arises when wanting to estimate the difference in those predicted values. A solution to this problem is provided by the `esticon()` function in the package XXXX as follows. Define

```
>L.Cu2 <- c(1, 1, 0, 7, 7^2, 7^3, 7, 0, 7^2, 0)
>L.Cu3 <- c(1, 0, 1, 7, 7^2, 7^3, 0, 7, 0, 7^2)
>L.Cu2
```

```
 [1]   1   1   0   7  49 343   7   0  49   0
```

```
>L.Cu3
```

```
 [1]   1   0   1   7  49 343   0   7   0  49
```

Then the predicted values and the difference between them is

```
>sum(gee2$coef * L.Cu2)
```

```
[1] 61.85898
```

```
>sum(gee2$coef * L.Cu3)
```

```
[1] 65.48972
```

```
>sum(gee2$coef * (L.Cu3 - L.Cu2))
```

```
[1] 3.630748
```

The problem in practice is that when estimating such functions (contrasts) we always want an estimate of the standard error and often we want to test the hypothesis that the contrast is equal to a specifed value. To obtain this we use the `esticon()` function

```
>L <- rbind(L.Cu2, L.Cu3, diff = L.Cu3 - L.Cu2)
>L
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
L.Cu2    1    1    0    7   49  343    7    0   49     0
L.Cu3    1    0    1    7   49  343    0    7    0    49
diff     0   -1    1    0    0    0   -7    7  -49    49
```

```
>esticon(gee2, L)
```

```
      beta0  Estimate Std.Error    X2.value DF Pr(>|X^2|)
L.Cu2     0 61.858975  1.376609 2019.223943  1   0.000000
L.Cu3     0 65.489723  1.809338 1310.107187  1   0.000000
diff      0  3.630748  2.284166    2.526602  1   0.111941
```

So the estimated difference is not significantly different from `beta0=0`.

7 LSmeans

We conclude by showing how to calculate LSmeans. This is done using the `lsmean()` function in the `pda` package. One has to be a little bit careful when using `lsmean()` as illustrated in the following:

Consider

```
>lsmean(gee2)
```

```
  Cu     Time     pred       se
1  1 6.480836 41.01404 1.983820
2  2 6.480836 39.85888 1.947625
3  3 6.480836 43.44879 2.292466
```

The values in the `pred` column are "wrong", (and it is the hope that this error in the `pda` package will be fixed). To get the right answer one needs to define the quadratic and cubic terms directly and rewrite the model in terms of these as:

```
>dietox2 <- dietox
>dietox2$Time2 <- dietox$Time^2
>dietox2$Time3 <- dietox$Time^3
>gee3 <- geeglm(Weight ~ Cu * Time + Cu * Time2 + Time3,
     data = dietox2, id = Pig, family = poisson("identity"),
     corstr = "ar1")
>lsmean(gee3)
```

```
  Cu     Time    Time2    Time3     pred       se
1  1 6.480836 53.85366 502.7456 60.36739 1.025467
2  2 6.480836 53.85366 502.7456 59.54448 1.176289
3  3 6.480836 53.85366 502.7456 62.23879 1.649893
```

For completeness we write out the details of what the specific LSmeans are in this case:

```
>mT <- mean(dietox$Time)
>mT2 <- mean(dietox$Time^2)
>mT3 <- mean(dietox$Time^3)
>L1 <- c(1, 0, 0, mT, mT2, mT3, 0, 0, 0, 0)
>L2 <- c(1, 1, 0, mT, mT2, mT3, mT, 0, mT2, 0)
>L3 <- c(1, 0, 1, mT, mT2, mT3, 0, mT, 0, mT2)
>esticon(gee2, rbind(L1, L2, L3))
```

```
   beta0 Estimate Std.Error X2.value DF Pr(>|X^2|)
L1     0 60.36739  1.025467 3465.462  1          0
L2     0 59.54448  1.176289 2562.445  1          0
L3     0 62.23879  1.649893 1423.018  1          0
```

A On why the gee model fits best

To justify the claim that the standard errors produced from the gee model are the most appropriate ones, reconsider the notion of variance of an estimator: Let $\hat{\theta}(y)$ be an estimator of a parameter θ based on a sample of data $y = (y_1, \ldots, y_n)$. The variance $Var(\hat{\theta})$ is a measure of how much $\hat{\theta}(y)$ will vary if the experiment is repeated under identical conditions a large number of times. To be specific let $y^1 = (y_1^1, \ldots, y_n^1), \ldots, y^R = (y_1^R, \ldots, y_n^R)$ denote the samples which are obtained after repeating the experiment R times. Let $\hat{\theta}(y^1), \ldots, \hat{\theta}(y^R)$ be the corresponding estimates calculated for each of the R experiments. Then the variance of $\hat{\theta}(y^1), \ldots, \hat{\theta}(y^R)$ is (when $R \to \infty$) equal to $Var(\hat{\theta})$.

In practice R is finite. Letting $\bar{\hat{\theta}}$ denote the average of $\hat{\theta}(y^1), \ldots, \hat{\theta}(y^R)$. Then

$$\mathbb{V}\mathrm{ar}(\hat{\theta}) \approx \frac{1}{R-1} \sum_r (\hat{\theta}(y_r) - \bar{\hat{\theta}})^2$$

In real life the experiment can not be repeated, but there is a statistical technique called "jacknife" by which one can mimic the replication of an experiment. We will not go into details about the method but refer to e.g. Efron (1982). Instead we will show that the jacknife technique is very simple to implement in practice:

```
>d <- unique(dietox$Pig)
>v1 <- v2 <- v3 <- NULL
>for (i in 1:length(d)) {
     print(c(i, d[i]))
     dsub <- subset(dietox, Pig != d[i])
     gee1 <- geeglm(mf, data = dsub, id = Pig, family = poisson("identity"),
         corstr = "ar1")
     qpo1 <- glm(mf, data = dsub, family = quasipoisson("identity"))
     lm4 <- lm(mf, data = dsub)
     v1 <- rbind(v1, summary(gee1)$coef[, 1])
     v2 <- rbind(v2, summary(qpo1)$coef[, 1])
     v3 <- rbind(v3, summary(lm4)$coef[, 1])
 }
>SEgeej <- sqrt(apply(v1, 2, var) * (nrow(v1) - 1))
>SEqpoj <- sqrt(apply(v2, 2, var) * (nrow(v2) - 1))
>SElmj <- sqrt(apply(v3, 2, var) * (nrow(v3) - 1))
```

```
>round(cbind(SEqpoj, SEgeej, SElmj, SEgee, SEqpo, SElm),
     3)
```

```
             SEqpoj SEgeej SElmj SEgee SEqpo  SElm
(Intercept)   0.724  0.722 0.801 0.693 1.726 2.395
Cu2           0.966  0.978 1.072 0.941 2.389 3.316
Cu3           1.058  1.062 1.165 1.036 2.433 3.350
Time          0.342  0.375 0.398 0.360 1.249 1.535
I(Time^2)     0.059  0.074 0.059 0.070 0.237 0.270
I(Time^3)     0.003  0.004 0.003 0.004 0.013 0.014
Cu2:Time      0.606  0.663 0.703 0.637 1.724 2.123
Cu3:Time      0.691  0.641 0.729 0.609 1.761 2.146
Cu2:I(Time^2) 0.098  0.121 0.105 0.115 0.327 0.372
Cu3:I(Time^2) 0.126  0.116 0.127 0.109 0.334 0.377
Cu2:I(Time^3) 0.005  0.006 0.005 0.006 0.018 0.019
Cu3:I(Time^3) 0.007  0.006 0.007 0.006 0.018 0.019
```

The first three columns contain the jacknife estimates of the standard errors under the three different estimation methods. They are practically identical and represent an approximation to the true standard error which one would find when repeating the experiment a large number of times. The next three columns contain the standard errors estimated using the different models, and we see that the gee model produces the standard errors which are closest to the "true" ones.

References

Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1982.

Charlotte Lauridsen, Søren Højsgaard, and Martin Tang Sørensen. Influence of dietary rapeseed oil, vitamin e and copper on the performance and the antioxidative status of pigs. *J. Anim. Sci.*, 77:906–916, 1999.