Analysis of multivariate competing risks data

Klaus Holst & Thomas Scheike

August 5, 2019

Overview

- marginal modelling with standard errors cif,

- cause specific hazards

- cumulative incidence modelling

  - random effects simple cif

  - Luise model

When looking at multivariate survival data with the aim of learning about the dependence that is present, possibly after correcting for some covariates different approaches are available in the mets package

- Binary models and adjust for censoring with inverse probabilty of censoring weighting

- Bivariate surival models of Clayton-Oakes type

  - With regression structure on dependence parameter

  - With additive gamma distributed random effects

  - Special functionality for polygenic random effects modelling such as ACE, ADE ,AE and so forth.

- Plackett OR model model

  - With regression structure on OR dependence parameter

- Cluster stratified Cox

Typically it can be hard or impossible to specify random effects models with special structure among the parameters of the random effects. This is possible for our specification of the random effects models.

To be concrete about the model structure assume that we have paired binomial data $T_1, \delta_1, T_2, \delta_2, X_1, X_2$ where the censored survival responses are $T_1, \delta_1, T_2, \delta_2$ and we have covariates X_1, X_2.

The focus of this vignette is describe how to work on bivariate survival data using the addtive gamma-random effects models. We present two different ways of specifying different dependence structures.

The basic models assumes that each subject has a marginal on Cox-form

$$\lambda_0(t) \exp(X_{ki}^T \beta)$$

then two types of models can be considered.

- Univariate models with a single random effect for each cluster and with a regression design on the varince.

- Multivariate models with multiple random effects for each cluster.

The univariate models are then given a given cluster random effects Z_k with parameter θ the joint survival function is given by the Clayton copula and on the form

$$\psi(\theta, \psi^{-1}(\theta, S_1(t, X_{k1})) + \psi^{-1}(\theta, S_1(t, X_{k1}))$$

where ψ is the Laplace transform of a gamma distributed random variable with mean 1 and variance θ.

We then model the variance within clusters by a cluster specific regression design such that

$$\theta = z_j^T \alpha$$

where z is the regression design (specified by theta.des in the software).

This model can be fitted using a pairwise likelihood or the pseudo-likelihood using either

- twostage

- twostageMLE

For the Multivariate models we are given a multivarite random effect each subject $(Z_1, ..., Z_d)$ with d random effects. The total random effect for each subject is then specified using a regression design on these random effects, with a regression vector v_j such that the total random effect is $\{v_1^T (Z_{1, ..., Z_d})\}$. Each random effect has an associated parameter $(\lambda_1, ..., \lambda_d)$ and Z_j is Gamma distributed with

- mean $lambda_j / v_1^T \lambda$

- variance $\setminus(\lambda_j / (v_1^T \lambda)^2\}$.

The key assumption to make the two-stage fitting possible is that

$$lamtot = v_j^T \lambda$$

with clusters.

The DEFAULT parametrization (var.par=1) uses the variances of the random effecs

$$\theta_j = \lambda_j / (v_1^T \lambda)^2$$

For alternative parametrizations one can specify how the parameters relate to λ_j with the argument var.par=0.

For both types of models the basic model assumptions are that given the random effects of the clusters the survival distributions within a cluster are independent and ' on the form

$$P(T > t | x, z) = exp(-Z \cdot Laplace^{-1}(lamtot^{-1}, S(t|x)))$$

with the inverse laplace of the gamma distribution with mean 1 and variance 1/lamtot.

Finally the parameters $(\lambda_1, \ldots, \lambda_d)$ are related to the parameters of the model by a regression construction M (d x k), that links the d λ parameters with the k underlying α parameters

$$\lambda = M\alpha$$

here using theta.des to specify these low-dimension association. Default is a diagonal matrix. This can be used to make structural assumptions about the variances of the random-effects as is needed for the ACE model for example. In software M is called theta.des

We consider K independent clusters, with n_k subject within each cluster. For each cluster we are given a set of independent random effects $V = (V_1, \ldots, V_m)^T$. We let $(V_1, \ldots, V_m)^T$ be independent Gamma distributed with $V_l \sim \Gamma(\eta_l, \nu_l), l = 1, \ldots, p$ independent gamma distributed random variables such that $E(V_l) = \eta_l/\nu$ and $Var(V_l) = \eta_l/\nu^2$. %%Let $\nu = (\nu_1, \ldots, \nu_p)$. The $\eta = (\eta_1, \ldots, \eta_m)$ parameters are given such that $\eta = D\theta$. Letting the rows in the matrix be denoted as $Q_i, \ldots, Q_m$. %%%As is commonly done [1]

[1] ; and

To facilitate our two-stage construction we also assume that $\nu = Q_i^T \eta$ for all $i = 1, \ldots, n_k$ such that $Q_i^T V$ is also Gamma distributed with $\Gamma(1, \nu)$, that is has variance ν^{-1} and mean 1. We get back to specific models where this is the case, but this assumption is often reasonable and needed [2]

[2] ; and

Let $\Psi(\eta_l, \nu, \cdot)$ denote the Laplace transform of the Gamma distribution $\Gamma(\eta_l, \nu)$, and let its inverse be $\Psi^{-1}(\eta_l, \nu, \cdot)$. For simplicity we also assume that η is the same across clusters.

Assume that the marginal survival distribution for subject i within cluster k is given by $S_{X_{k,i}}(t)$ given covariates $X_{k,i}$.

Now given the random effects of the cluster V_k and the covariates $X_{k,i}$ $i = 1, \ldots, n_k$ we assume that subjects within the cluster are independent with survival distributions

$$\exp(-(Q_{k,i} V_k) \Psi^{-1}(\nu, \nu, S_{X_{k,i}}(t))).$$

A consequence of this is that the hazards given the covariates $X_{k,i}$ and the random effects V_k are given by

$$\lambda_{k,i}(t; X_{k,i}, V_{k,i}) = (Q_{k,i} V_k) D_3 \Psi^{-1}(\nu, \nu, S_{X_{k,i}}(t)) D_t S_{X_{k,i}}(t) \quad (1)$$

where D_t and D_3 denotes the partial derivatives with respect to t and the third argument, respectively.

Further, we can express the multivariate survival distribution as

$$S(t_1, \ldots, t_m) = \exp(-\sum_{i=1}^{m} (Q_i V) \Psi^{-1}(\eta_l, \nu_l, S_{X_{k,i}}(t_i)))$$

$$= \prod_{l=1}^{p} \Psi(\eta_l, \eta, \sum_{i=1}^{m} Q_{k,i} \Psi^{-1}(\eta, \eta, S_{X_{k,i}}(t_i))). \quad (2)$$

In the case of considering just pairs, we write this function as $C(S_{k,i}(t), S_{k,j}(t))$.

In addition to survival times from this model, we assume that we independent right censoring present $U_{k,i}$ such that the given V_k and the covariates $X_{k,i}$ $i = 1, \ldots, n_k$ $(U_{k,1}, \ldots, U_{k,n_k})$ of $(T_{k,1}, \ldots, T_{k,n_k})$, and the conditional censoring distribution do not depend on V_k. We can also express this via counting processes $N_{k,i}(t) = I(T_{k,i} < t, T_{k,i} < U_{k,i})$ and with at risk indicators $Y_{k,i}(t) = I(T_{k,i} > t, U_{k,i} > t)$, and the censoring indicators $\delta_{k,i} = I(T_{k,i} < U_{k,i})$.

%%%Due to the marginal specification we can estimate apply the two-stage approach %%%as in [3]. We return to this in the next section.

One consequence of the model strucure is that the Kendall's can be computed for two-subjects (i, j) across two clusters "1" and "2" as

$$E\left(\frac{(Q_{1i}V_1 - Q_{1j}V_2)(Q_{2i}V_1 - Q_{2j}V_2)}{(Q_{1i}V_1 + Q_{2i}V_2)(Q_{1j}V_1 + Q_{2j}V_2)}\right) \tag{3}$$

under the assumption that that we compare pairs with equivalent marginals ($S_{X_{1,i}}(t) = S_{X_{2,i}}(t)$ and $S_{X_{1,j}}(t) = S_{X_{2,j}}(t)$) and that $S_{X_{1,i}}(\infty) = S_{X_{1,j}}(\infty) = 0$. %%%We return to another characetrization %%%of the dependence via the cross hazards ratio. Here we also use that η is the same across clusters. The Kendall's tau would be the same for (**??**) due to the same additive structure for the frailty terms, and the random effects thus have the same interpretation in terms of Kendall's tau.

Clusters stratified Cox models

Show how efficient the stratified Cox is with GOF and all

```
library(mets)
 data(diabetes)
 margph <- phreg(Surv(time,status)~treat+strata(id),data=
     diabetes)
```

```
library(mets)
gg <- gof (margph)

par(mfrow=c(2,2))
plot(gg)
```

Univariate plackett model twostage models

```
library(mets)
 data(diabetes)

 # Marginal Cox model with treat as covariate
 margph <- phreg(Surv(time,status)~treat+cluster(id),data=
     diabetes)
 # Clayton-Oakes, MLE
 fitco1<-twostageMLE(margph,data=diabetes,theta=1.0)
 summary(fitco1)

```

```
10   # Plackett model
11   mph <- phreg(Surv(time,status)~treat+cluster(id),data=
         diabetes)
12   fitp <- survival.twostage(mph,data=diabetes,theta=3.0,Nit
         =40,
13       clusters=diabetes$id,var.link=1,model="plackett")
14   summary(fitp)
15
16   # Clayton-Oakes
17   fitco2 <- survival.twostage(mph,data=diabetes,theta=0.0,
         detail=0,
18       clusters=diabetes$id,var.link=1,model="clayton.oakes
             ")
19   summary(fitco2)
20   fitco3 <- survival.twostage(margph,data=diabetes,theta=1.0,
         detail=0,
21       clusters=diabetes$id,var.link=0,model="clayton.oakes
             ")
22   summary(fitco3)
23
24   # without covariates but with stratafied
25   marg <- phreg(Surv(time,status)~+strata(treat)+cluster(id),
         data=diabetes)
26   fitpa <- survival.twostage(marg,data=diabetes,theta=1.0,
27       clusters=diabetes$id,score.method="optimize")
28   summary(fitpa)
29
30   fitcoa <- survival.twostage(marg,data=diabetes,theta=1.0,
         clusters=diabetes$id,
31       model="clayton.oakes")
32   summary(fitcoa)
33
34
35   # Piecewise constant cross hazards ratio modelling
36   d <- subset(simClaytonOakes(2000,2,0.5,0,stoptime=2,left=0),
         !truncated)
37   udp <- piecewise.twostage(c(0,0.5,2),data=d,score.method="
         optimize",
38               id="cluster",timevar="time",
39               status="status",model="clayton.oakes",silent=0)
40   summary(udp)
```

Univariate gamma (clayton-oakes) model twostage models

Looking at the data

```
1   library(mets)
2   data(diabetes)
3
4   # Marginal Cox model with treat as covariate
5   margph <- phreg(Surv(time,status)~treat+cluster(id),data=
         diabetes)
6   # Clayton-Oakes, MLE
7   fitco1<-twostageMLE(margph,data=diabetes,theta=1.0)
8   summary(fitco1)
9
10  # Plackett model
```

```
11  mph <- phreg(Surv(time,status)~treat+cluster(id),data=
        diabetes)
12  fitp <- survival.twostage(mph,data=diabetes,theta=3.0,Nit
        =40,
13      clusters=diabetes$id,var.link=1,model="plackett")
14  summary(fitp)
15
16  # Clayton-Oakes
17  fitco2 <- survival.twostage(mph,data=diabetes,theta=0.0,
        detail=0,
18      clusters=diabetes$id,var.link=1,model="clayton.oakes
            ")
19  summary(fitco2)
20  fitco3 <- survival.twostage(margph,data=diabetes,theta=1.0,
        detail=0,
21      clusters=diabetes$id,var.link=0,model="clayton.oakes
            ")
22  summary(fitco3)
23
24  # without covariates but with stratafied
25  marg <- phreg(Surv(time,status)~+strata(treat)+cluster(id),
        data=diabetes)
26  fitpa <- survival.twostage(marg,data=diabetes,theta=1.0,
27      clusters=diabetes$id,score.method="optimize")
28  summary(fitpa)
29
30  fitcoa <- survival.twostage(marg,data=diabetes,theta=1.0,
        clusters=diabetes$id,
31      model="clayton.oakes")
32  summary(fitcoa)
33
34
35  # Piecewise constant cross hazards ratio modelling
36  d <- subset(simClaytonOakes(2000,2,0.5,0,stoptime=2,left=0),
        !truncated)
37  udp <- piecewise.twostage(c(0,0.5,2),data=d,score.method="
        optimize",
38      id="cluster",timevar="time",
39      status="status",model="clayton.oakes",silent=0)
40  summary(udp)
```

Multivariate gamma twostage models

```
1  library(mets)
2
3  # structured random effects model additive gamma ACE
4  # simulate structured two-stage additive gamma ACE model
5  data <- simClaytonOakes.twin.ace(2000,2,1,0,3)
6  out <- twin.polygen.design(data,id="cluster")
7  pardes <- out$pardes
8  pardes
9  des.rv <- out$des.rv
10  head(des.rv)
11  aa <- phreg(Surv(time,status)~x+cluster(cluster),data=data,
        robust=0)
```

```
12   ts <- survival.twostage(aa,data=data,clusters=data$cluster,
         detail=0,
13           theta=c(2,1),var.link=0,step=0.5,
14           random.design=des.rv,theta.des=pardes)
15   summary(ts)
```

```
1    library(mets)
2
3    set.seed(1000)
4    source("mets/R/sim.clayton.oakes.R")
5    data <- simClaytonOakes.family.ace(8000,2,1,0,3)
6    head(data)
7    data$number <- c(1,2,3,4)
8    data$child <- 1*(data$number==3)
9    out <- ace.family.design(data,member="type",id="cluster")
10   out$pardes
11   head(out$des.rv)
12
13   aa <- aalen(Surv(time,status)~+1,data=data,robust=0)
14   pa <- phreg(Surv(time,status)~+1+cluster(cluster),data=data)
15
16   # additive gamma models with and without pair call
17   # make ace random effects design
18
19   # simple random effects call
20   ts0 <- twostage(aa,data=data,clusters=data$cluster,
21       detail=1,var.par=1,var.link=0,
22       theta=c(2,1),
23       random.design=out$des.rv,theta.des=out$pardes)
24   summary(ts0)
25
26   ts00 <- twostage(pa,data=data,clusters=data$cluster,
27       detail=1,var.par=1,var.link=0,
28       theta=c(2,1),
29       random.design=out$des.rv,theta.des=out$pardes)
30   summary(ts00)
31
32
33   checkderiv=0
34   if (checkderiv==1) {
35   ts0 <- twostage(aa,data=data,clusters=data$cluster,
36       detail=1,numDeriv=1,Nit=0,var.par=1,
37       theta=log(c(2,1)/9),var.link=1,step=1.0,
38       random.design=out$des.rv,theta.des=out$pardes)
39   ts0$score
40   ts0$score1
41
42   ts0 <- twostage(aa,data=data,clusters=data$cluster,
43       detail=1,numDeriv=1,Nit=0,var.par=1,
44       theta=c(2,1)/9,var.link=0,step=1.0,
45       random.design=out$des.rv,theta.des=out$pardes)
46   ts0$score
47   ts0$score1
48
49
50   ts0 <- twostage(aa,data=data,clusters=data$cluster,
51       detail=1,numDeriv=1,Nit=0,var.par=0,
```

```
52      theta=log(c(2,1)),var.link=1,step=1.0,
53      random.design=out$des.rv,theta.des=out$pardes)
54    ts0$score
55    ts0$score1
56
57    ts0 <- twostage(aa,data=data,clusters=data$cluster,
58        detail=1,numDeriv=1,Nit=0,var.par=0,
59        theta=c(2,1),var.link=0,step=1.0,
60        random.design=out$des.rv,theta.des=out$pardes)
61    ts0$score
62    ts0$score1
63
64    }
65
66
67    # now specify fitting via specific pairs
68
69    # first all pairs
70    mm <- familycluster.index(data$cluster)
71    head(mm$familypairindex,n=10)
72    pairs <- matrix(mm$familypairindex,ncol=2,byrow=TRUE)
73    tail(pairs,n=12)
74    # make all pairs and pair specific design and pardes
75    # same as ts0 but pairs specified
76    ts <- twostage(aa,data=data,clusters=data$cluster,
77        theta=c(2,1),var.link=0,step=1.0,
78        random.design=out$des.rv,
79        theta.des=out$pardes,pairs=pairs)
80    summary(ts)
81
82    ts <- twostage(pa,data=data,clusters=data$cluster,
83        theta=c(2,1),var.link=0,step=1.0,
84        random.design=out$des.rv,
85        theta.des=out$pardes,pairs=pairs)
86    summary(ts)
87
88
89    # random sample of pairs
90    ssid <- sort(sample(1:48000,20000))
91
92    # take some of all
93    tsd <- twostage(aa,data=data,clusters=data$cluster,
94        theta=c(2,1)/10,var.link=0,step=1.0,
95        random.design=out$des.rv,iid=1,
96          theta.des=out$pardes,pairs=pairs[ssid,])
97    summary(tsd)
98
99    # same analyses but now gives only data that is used in the
        relevant pairs
100   ids <- sort(unique(c(pairs[ssid,])))
101
102   pairsids <- c(pairs[ssid,])
103   pair.new <- matrix(fast.approx(ids,c(pairs[ssid,])),ncol=2)
104   head(pair.new)
105
106   # this requires that pair.new refers to id's in dataid
        (survival, status and so forth)
107   # random.design and theta.des are constructed to be the
```

```
          array 3 dims via individual specfication from
          ace.family.design
108   dataid <- dsort(data[ids,],"cluster")
109   outid <- ace.family.design(dataid,member="type",id="cluster"
          )
110   outid$pardes
111   head(outid$des.rv)
112
113   tsdid <- twostage(aa,data=dataid,clusters=dataid$cluster,
114         theta=c(2,1)/10,var.link=0,step=1.0,
115         random.design=outid$des.rv,iid=1,
116         theta.des=outid$pardes,pairs=pair.new)
117   summary(tsdid)
118   coef(tsdid)
119   coef(tsd)
120   # same as tsd
121
122
123   # now direct specification of random.design and
          theta.design
124   # rather than taking the rows of the des.rv for the
          relevant pairs
125   # can make a pair specific specification of random effects
126
127   pair.types <- matrix(dataid[c(t(pair.new)),"type"],byrow=T,
          ncol=2)
128   head(pair.new)
129   head(pair.types)
130
131   # here makes pairwise design , simpler random.design og
          pardes, parameters
132   # stil varg, varc
133   # mother, child, share half rvm=c(1,1,0) rvc=c(1,0,1),
134   # thetadesmcf=rbind(c(0.5,0),c(0.5,0),c(0.5,0),c(0,1))
135   #
136   # father, child, share half rvf=c(1,1,0) rvc=c(1,0,1),
137   # thetadescf=rbind(c(0.5,0),c(0.5,0),c(0.5,0),c(0,1))
138   #
139   # child, child, share half rvc=c(1,1,0) rvc=c(1,0,1),
140   # thetadesmf=rbind(c(0.5,0),c(0.5,0),c(0.5,0),c(0,1))
141   #
142   # mother, father, share 0 rvm=c(1,0) rvf=c(0,1),
143   # thetadesmf=rbind(c(1,0),c(1,0),c(0,1))
144
145  theta.des <- array(0,c(4,2,nrow(pair.new)))
146  random.des <- array(0,c(2,4,nrow(pair.new)))
147   # random variables in each pair
148  rvs <- c()
149  for (i in 1:nrow(pair.new))
150  {
151     if (pair.types[i,1]=="mother" & pair.types[i,2]=="father
          ")
152     {
153     theta.des[,,i] <- rbind(c(1,0),c(1,0),c(0,1),c(0,0))
154     random.des[,,i] <- rbind(c(1,0,1,0),c(0,1,1,0))
155     rvs <- c(rvs,3)
156     } else {
157     theta.des[,,i] <- rbind(c(0.5,0),c(0.5,0),c(0.5,0),c
          (0,1))
```

```
158      random.des[,,i] <- rbind(c(1,1,0,1),c(1,0,1,1))
159      rvs <- c(rvs,4)
160      }
161    }
162    # 3 rvs here
163    random.des[,,7]
164    theta.des[,,7]
165    # 4 rvs here
166    random.des[,,1]
167    theta.des[,,1]
168    head(rvs)
169
170    tsdid2 <- twostage(aa,data=dataid,clusters=dataid$cluster,
171        theta=c(2,1)/10,var.link=0,step=1.0,
172        random.design=random.des,
173        theta.des=theta.des,pairs=pair.new,pairs.rvs=rvs)
174    summary(tsdid2)
175    tsd$theta
176    tsdid2$theta
177    tsdid$theta
178
179
180    # simpler specification via kinship coefficient for each
           pair
181
182    kinship <- c()
183    for (i in 1:nrow(pair.new))
184    {
185    if (pair.types[i,1]=="mother" & pair.types[i,2]=="father")
          pk1 <- 0 else pk1 <- 0.5
186    kinship <- c(kinship,pk1)
187    }
188    head(kinship,n=10)
189
190    out <- make.pairwise.design(pair.new,kinship,type="ace")
191    names(out)
192    # 4 rvs here , here independence since shared component has
           variance 0 !
193    out$random.des[,,9]
194    out$theta.des[,,9]
195
196
197    tsdid3 <- twostage(aa,data=dataid,clusters=dataid$cluster,
198        theta=c(2,1)/10,var.link=0,step=1.0,
199        random.design=out$random.design,
200        theta.des=out$theta.des,pairs=pair.new,pairs.rvs=out$
             ant.rvs)
201    summary(tsdid3)
202    coef(tsdid3)
203
204    # same as above tsdid2
205
206
207    # simple models, test for pairs structure
208
209    library(mets)
210
211    ts0 <- twostage(aa,data=data,clusters=data$cluster,
```

```
212     detail=0,numDeriv=1,Nit=10,
213     theta=c(0.17),var.link=0,step=1.0)
214 summary(ts0)
215 ts0$score; ts0$score1
216 ts0$Dscore; ts0$hess
217
218 mm <- familycluster.index(data$cluster)
219 head(mm$familypairindex,n=10)
220 pairs <- matrix(mm$familypairindex,ncol=2,byrow=TRUE)
221 head(pairs,n=12)
222 tail(pairs,n=12)
223 dim(pairs)
224
225 cc <- cluster.index(data$cluster)
226
227 ts0 <- twostage(aa,data=data,clusters=data$cluster,
228     detail=1,Nit=0,
229     theta=ts0$theta,var.link=0,pairs=pairs)
230 summary(ts0)
231
232
233
234 library(mets)
235
236 set.seed(100)
237 data <- simClaytonOakes.family.ace(8000,2,1,0,3)
238 head(data)
239 data$number <- c(1,2,3,4)
240 data$child <- 1*(data$number==3)
241
242 # make ace random effects design
243 out <- ace.family.design(data,member="type",id="cluster")
244 out$pardes
245 head(out$des.rv)
246
247 # makes marginal model (same for all)
248 aa <- aalen(Surv(time,status)~+1,data=data,robust=0)
249
250
251 mm <- familycluster.index(data$cluster)
252 head(mm$familypairindex,n=10)
253 pairs <- matrix(mm$familypairindex,ncol=2,byrow=TRUE)
254 head(pairs,n=12)
255 tail(pairs,n=12)
256 dim(pairs)
257 #
258
259 ts0 <- twostage(aa,data=data,clusters=data$cluster,
260     detail=1,Nit=10,
261     theta=c(0.2),var.link=0,step=1.0)
262 summary(ts0)
263
264 ts0 <- twostage(aa,data=data,clusters=data$cluster,
265     detail=1,Nit=10,numDeriv=1,
266     theta=c(0.2),var.link=0,step=1.0,pairs=pairs)
267 summary(ts0)
268 ts0$score
```

```
269    ts0$score1

270

271    ts0 <- twostage(aa,data=data,clusters=data$cluster,
272        detail=1,Nit=10,
273        theta=c(0.2),var.link=0,step=1.0,model="plackett")
274    summary(ts0)

275

276    ts0 <- twostage(aa,data=data,clusters=data$cluster,
277        detail=1,Nit=10,
278        theta=c(0.2),var.link=0,step=1.0,model="plackett",pairs=
              pairs)
279    summary(ts0)

280

281

282

283    theta.des <- model.matrix(~x1,data=data)

284

285    ts0 <- twostage(aa,data=data,clusters=data$cluster,
286        detail=1,Nit=10,theta.des=theta.des,
287        theta=c(0.2),var.link=0,step=1.0)
288    summary(ts0)

289

290    ts0 <- twostage(aa,data=data,clusters=data$cluster,
291        detail=1,Nit=10,theta.des=theta.des,
292        theta=c(0.2),var.link=0,step=1.0,pairs=pairs)
293    summary(ts0)

294

295    ts0 <- twostage(aa,data=data,clusters=data$cluster,
296        detail=1,Nit=10,theta.des=theta.des,
297        theta=c(0.2),var.link=0,step=1.0,model="plackett")
298    summary(ts0)

299

300    ts0 <- twostage(aa,data=data,clusters=data$cluster,
301        detail=1,Nit=10,theta.des=theta.des,
302        theta=c(0.2),var.link=0,step=1.0,model="plackett",pairs=
              pairs)
303    summary(ts0)
```