Robust Covariate Control in Cluster-Randomized Trials with MPLUS and WinBUGS

Jiaquan Fan[1] and David Judkins[1]
[1]Westat, 1600 Research Blvd., Rockville, MD 20850

Abstract

Two more statistical software packages, MPLUS and WinBUGS, were tested as a continuation of a previous study of analysis method/software robustness in the analysis of clustered data. We want to evaluate robustness in the context of a randomized complete block design, where each "plot" is a small group of children at the same nursery school and a series of measurements of each child are made. We constructed a series of super populations in which the standard assumptions of hierarchical (mixed effects) linear models were violated. The results were compared with HLM, SUDAAN, PROC MIXED and a semi-parametrical analysis of variance procedure we tested before.

Key Words: Cluster-randomized trials, multi-level modeling, Bayesian inference using Gibbs Sampling

1. Introduction and Background

This work is a continuation of our 2006 study (Fan and Judkins, 2006) in which we undertook a simulation to study the robustness of some standard software options for covariate control in the context of cluster-randomized trials. We also developed and tested a new semi-parametric method which we called semi-parametric ANOVA.

In this study, we continue the line of research by including two more software systems, MPLUS and WINBUGS, in the simulation study. We also found that there is a problem of potential serial correlation in the program we used before to generate simulated data because the new seeds were generated by calling SAS random number generation functions. To fix this problem, we changed to a new and improved SAS function for random number generation and a method of drawing new seeds from a pre-determined sequence of numbers. Another change we made in the current study is to increase the variance for treatment effects when researching power.

The application of interest was a randomized experiment with alternate preschool instructional paradigms, loosely referred to as curricula. There were four alternate curricula and one control curriculum. All five arms were assigned to a recruited sample of 120 Even Start schools. The schools were deeply stratified into 24 blocks, each containing five schools. Within a block, the five schools were then randomly assigned to the five arms. The curricula involved instructional materials, instructional strategies, teacher training, teacher observation, and teacher consultation. Within the schools, parents of age-eligible children were recruited into the study. Measurements were conducted in the spring of 2004, prior to the introduction of the new curricula, and repeated at one-year intervals in 2005 and 2006. Measurements involved formal assessments of pre-literacy, social competency (teacher observation), parent interviews, and video-taping and behaviour-coding of staged parent-child interactive reading and toy-play sessions to gauge parenting skills. There was considerable turnover in the student-body each year, but there is some overlap of sample across years, and of course, there is considerable organizational and staffing stability. So one set of important covariates

involved school-level past performance and child-teacher ratios. Another important set of covariates involved parent socio-economic status, native language, and child demographics (age, race, sex, and disability status). Native language, in particular, has a huge effect on English pre-literacy.

For analysis, we wanted something that was robust to unequal student sample sizes per school, school-level nonresponse, deep stratification, heteroscedasticity, non-Gaussian errors and interactions. We therefore developed superpopulations that had the features of interest, generated samples from them, and tested several alternative analysis procedures on them, using type I error rates and statistical power as evaluation criteria.

In section 2, we discuss the superpopulations that we simulated. In section 3, we provide more detail on the analysis methods studied. In section 4, we present results. In section 5, we give some ideas for further research.

2. Simulated Superpopulations

Given the application, we built a series of superpopulations with an increasing number of violations of standard models. All shared a common form of having two child-level covariates, one school-level covariate, a random effect at the school level, and student level random error. The project-level covariate was built with a structure similar to the outcome of interest because the way it will be generated in the application is to take the average of students at the school the prior year. All of the superpopulations share a common model structure:

$$y_{ijk} = \mu + \beta_i + \alpha_i + X_{ijk}\theta + Z_{ij}\gamma + u_{ij} + e_{ijk},$$
$$Z_{ij} = \beta_i + u_{ij} + \bar{X}_{ij.}\theta + v_{ij}$$

where:

The indices stand respectively for block (i), treatment (j), child (k);

y_{ijk} is the outcome variable;

μ is the overall mean;

β_i is the (fixed) block effect;

α_i is the treatment effect;

X_{ijk} is a vector of two child level covariates ($X1$ = FamilyIncome, $X2$ = MothersEducation);

Z_{ij} is the baseline school-level average of the outcome variable measured on a different set of students prior to the intervention;

u_{ij} is the school level-random effect;

e_{ijk} is a child level random error;

$\bar{X}_{ij.}$ is a vector of school-level averages of child level covariates;

v_{ij} is a normally distributed random error term reflecting the error caused by basing the project-level fixed covariate on a small sample from the prior year rather than a long-run average;

u_{ij}, v_{ij} and e_{ijk} are mutually independent.

Because the theory is better developed for balanced designs, we introduced imbalance both at the school and the child level. Note that standard multi-level software assumes

that all the random errors are normal and homoscedastic. So we developed superpopulations that violated those assumptions. Finally, we allowed interactions. We simulated a series of superpopulations that violated various combinations of these standard assumptions to various degrees while generally keeping the violations within the range that we thought might reasonably occur in our application.

Seven different superpopulations with no treatment effect ($\alpha_i = 0$) were generated to test robustness of type 1 error rates. Superpopulation 1 satisfies most of the standard assumptions. The numbering of superpopulations 2 through 7 generally reflects increasing severe violations of standard assumptions:

Superpopulation 1: There are 24 blocks with five schools per block and each school contains exactly 12 children. There is no school-level nonresponse and the school- and child-level random errors are normally distributed. Residual variances are constant with $var(u_{ij}) = 12.81$ and $var(e_{ijk}) = 55.26$. The block effect is very large with. $\beta_i = 2i$. v_{ij} is normal in all superpopulations with $var(v_{ij}) = 6$.

Superpopulation 2: Same as superpopulation 1 except that the number of children per school is allowed to vary. The number of children per school follows a Poisson distribution with mean 12.

Superpopulation 3: Same as Superpopulation 2 except that there are two schools missing at random (for a total of 118 schools). The missing schools are from different blocks.

Superpopulation 4: Same as Superpopulation 3 except that the school- and child-level random errors have different variances in different blocks:

Block 1 – 6 has u_{ij} and e_{ijk} with variances 3 and 56,
Block 7 – 12 has u_{ij} and e_{ijk} with variances 6 and 42,
Block 13 – 18 has u_{ij} and e_{ijk} with variances 9 and 28,
Block 19 – 24 has u_{ij} and e_{ijk} with variances 12 and 14.

Superpopulation 5: Same as Superpopulation 3 except that the school- and child-level random errors have different variances in different treatment groups:

Treatment 1 has u_{ij} and e_{ijk} with variances 3 and 70,
Treatment 2 has u_{ij} and e_{ijk} with variances 6 and 56,
Treatment 3 has u_{ij} and e_{ijk} with variances 9 and 42,
Treatment 4 has u_{ij} and e_{ijk} with variances 12 and 28.
Control has u_{ij} and e_{ijk} with variances 15 and 14.

Superpopulation 6: Same as Superpopulation 3 except that school- and child-level random errors have Gamma distributions. u_{ij} has shape parameter $\alpha = 2$ and scale parameter $\beta = 0.395$ and e_{ijk} has $\alpha = 3$ and $\beta = 0.233$. Note that in this population, the school-level errors are more seriously non-normal than the student-level errors. Both skew and kurtosis are stronger for the school-level errors.

Superpopulation 7: Same as Superpopulation 4 except that there are treatment group effects for individual blocks but no effect on average. That is, within each single block

there are significant differences between the treatment groups, but when schools are aggregated to the treatment level, these differences average out.

Another three superpopulations with treatment effect were generated to compare type II error rates. For each of these superpopulations, all four experimental arms are assumed to be equally effective with $\alpha_i = 2.5$. This number was picked to give power in a range where we thought we might see the largest differences in power among the techniques.

Superpopulation 8: Model is the same as Superpopulations 4 except that treatment effect is added.

Superpopulation 9: Same as Superpopulations 5 except that treatment effect is added.

Superpopulation 10: Same as Superpopulation 6 except that treatment effect is added.

The components of variance in the model for the superpopulations are shown in Table 1. Naturally, there positive variance between treatment arms only for superpopulations 8, 9 and 10. All other variance components are constant across superpopulations. Also note that the between-block variance is very large. This was done with the aim of making it large enough to matter.

Table 1: Components of Variances

Component	*Magnitude*
Between block (fixed)	192
Between arm (fixed)	0 or 1
Child-level covariates (fixed)	3.4
School-level covariates (fixed)	18
School-level random effect (random)	13
Child level error (random)	55

3. Analysis Methods

The analysis methods/software we studied included HLM (Raudenbush, et al, 2004), SAS PROC MIXED (SAS Institute Inc., 2006), SUDAAN (Research Triangle Institute, 2001), semi-parametric ANOVA (Fan and Judkins, 2006), WinBUGS and MPLUS as listed in Table 2.

For PROC MIXED, we used school as subject, block as fixed effects and the restricted maximum likelihood option. An example of the code used is shown below:

```
proc mixed data=population      method=REML;
class block schoolid treatment;
model y=treatment block FamilyIncome MothersEducation Z/ solution ;
random int/ type = un subject = schoolid;
run;
```

Table 2: Analysis Methods Tested

Method	*Analysis Type*	*Software Package*
SUDAAN setup1	GEE	SUDAAN
SUDAAN setup 2	GEE	SUDAAN
SUDAAN setup 3	GEE	SUDAAN
HLM2	Frequentist likelihood	HLM
HLM2 with robust standard error	Frequentist likelihood	HLM
MPLUS	Frequentist likelihood	MPLUS
PROC MIXED	Frequentist likelihood	SAS
Semi-parametric ANOVA	Semi-parametric	SAS
WinBUGS	Bayesian MCMC	WinBUGS

For HLM, we used a 2 level HLM model with student as the first level and school as the second level. Indicator variables for four of the five treatments and 23 of the 24 blocks were entered as fixed effects in addition to FamilyIncome, MothersEducation, and *Z*. The estimation method was also restricted maximum likelihood. In this study we included results for HLM with regular standard error calculation as well as with the robust standard error option. More details on the HLM code are given in the appendix in Fan, Judkins, 2006.

For SUDAAN, the three options are
Setup 1: single strata, school as PSU, dummy variables for blocks entered as model variables;
Setup 2: single strata, block as PSU;
Setup 3: block as strata, school as PSU.

The semi-parametric ANOVA was inspired by Rosenbaum (2002) but has much in common with a line of papers mostly by Gary Koch and coauthors (Koch, et al, 1982 and 1998; Stokes, Davis, and Koch, 2000; Lavange, Durham, and Koch, 2005) that was launched by Quade (1967). For details, see section 2 in Fan, Judkins, 2006.

For MPLUS, we use version 5.21 with multi-level add-on. An example of the code used is shown below:

```
TITLE: 2-level fixed effects model to test treatment effect

DATA: FILE IS C:\Project\2010JSM\mplus\LKpop3.dat;
VARIABLE:
NAMES ARE y   schoolid   treat1-treat4 block2- block24
              faminc mothedu Z;
USEVARIABLES ARE y school treat1-treat4 block2-block24
                        faminc mothedu Z;
WITHIN     = faminc mothedu block2-block24;
BETWEEN = treat1-treat4 Z;
CLUSTER   = schoolid;

ANALYSIS:    TYPE = TWOLEVEL;
MODEL:       %WITHIN%        y ON x1 - x3 b2 - b24;
             %BETWEEN%     y ON r1 (p1) r2 (p2) r3 (p3) r4 (p4) ;
```

Mplus limit variables name to 8 characters, so family income and mother education were coded as faminc and mothedu in the program. The within and between parts of the model

correspond to level 1 and level 2 of a conventional multilevel regression model with a random intercept, where level1 is the student level and level 2 is the school level. We use the default estimator which is maximum likelihood with robust standard errors (p. 229, Muthén and Muthén, 1998-2007). The testing is done with a MODEL TEST statement using Wald test.

WINBUGS is a freely distributed Bayesian MCMC package developed by a team at the Medical research Council Biostatistics Research Unit in Cambridge. For simulation, we used WinBUG version 1.4.3. Normal distributions are assumed for both child level random error and the school level random effect. To specify the priors for these two random terms, we give Gamma (0.001, 0.001) to their precisions, which are the inverse of the variances of the normal distributions. For the fixed effects of treatment, block, and child and school level covariates, we use a 'flat' i.e. uniform prior across the whole real line, for each regression (fixed effect) coefficient. For the simulation, we choose a 400 burn in and 400 following iterations. These numbers were kept small because of the limited computing resource and the length of the time needed to run WinBUGS for 1000 simulations for each of the 10 superpopulations. More details on the code are given in the appendix in which there are two programs. The first program is the WinBUGS code for setting up the model, and the other program is a R program that perform these tasks: reading in the SAS generated simulation data; running the WinBUGS inside of R; reading the parameter estimates from WinBUGS back into R; and conducting tests of treatment effects. For the contrasts between treatment effects we use Z-tests and for the overall treatment effects we use a Wald Chi-square test.

4. Simulation Results

The simulation results are shown in Figures 1 through 4. In each of these, the horizontal axis reflects the various superpopulations. Type 1 error rates for the overall treatment effect are shown in Figure 1 (for the seven populations with no treatment effect). There is a separate curve for each of the nine analysis methods. Figure 2 shows the type-1 error simulation results for contrasts. Since the four contrasts do not differ much, the four tests were pooled together to be represented by a single line for each analysis method. Power for detecting overall treatment effect is shown in Figure 3 for the three populations with treatment effects. Similarly, power levels for the contrasts are shown in Figure 4, which shows only the average of the four contracts for each method.

For Sudaan, Mixed, HLM (with regular standard error), and Semi-ANOVA, the results are similar to the 2006 study. Adding MPLUS, WinBUGS, and HLM with robust standard error option, the lines in the figures show interesting groupings that are consistent in all four figures.

MPLUS, HLM with standard error option, and SUDAAN 1 are at the top with very large type 1 error for overall treatment effect and average treatment contrast effect. They are also high on the two plots of power simulations.

The lines for WinBUGS lie lower than the above group and are generally higher than the lines of remaining methods in both type I error plots and power simulation plots.

The remaining methods can be loosely grouped as three groups. The higher one is SUDAAN 2 and SUDAAN 3, the middle one has MIXED and HLM with regular standard error. As we observed before, MIXED and HLM have very similar results. And the lowest line is semi-ANOVA.

In conclusion, semi-parametric procedure is the most robust in preserving type I error rates for every superpopulation but it also has the lowest power among the 9 methods. MIXED and HLM with model-based standard errors perform fairly well in robustness, as well as having reasonable power performance.

The SUDAAN variants, WinBUGS, MPLUS, and HLM with robust standard error option all performed poorly for type I error rates. This makes their sometimes very higher power level irrelevant. Overall this result confirms the validity of our recommendation in 2006 to use either MIXED or HLM for robust covariate control.

One of the most interesting finding is that even when all the standard assumptions are met, most of the methods behave very poorly. Only the semi-parametric approach provided valid inference although the REML methods with model-based standard errors are not much inferior. We do not know the reasons why most of the methods perform so poorly but suspect that it has to do with the deep stratification and small sample sizes.

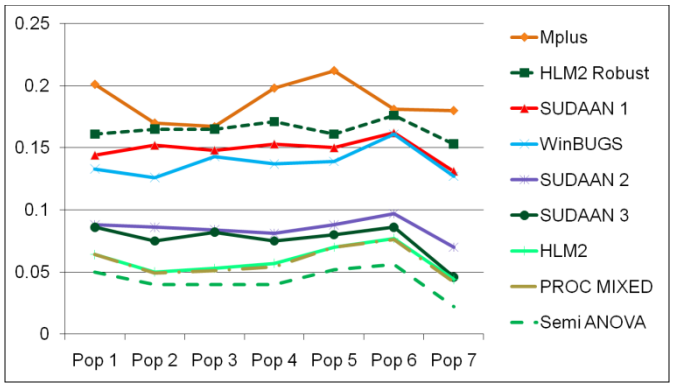

Figure 1: Type I error simulation: test for overall treatment effect

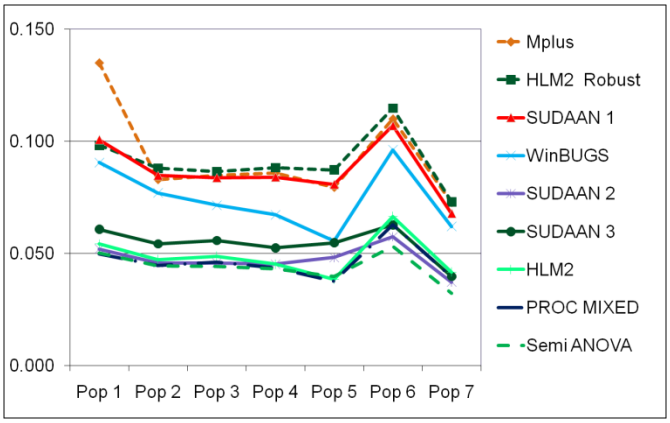

Figure 2: Type I error simulation: average of the tests for contrasts

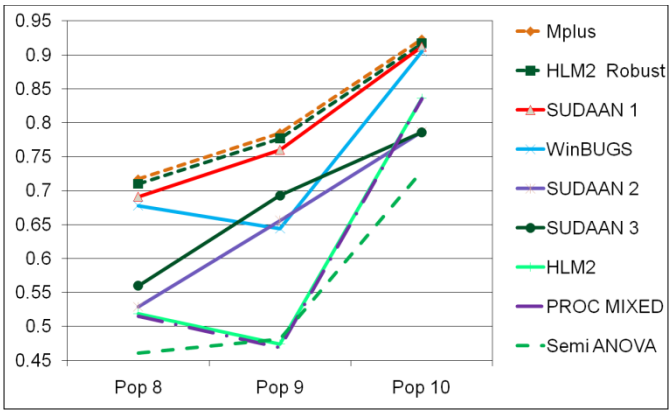

Figure 3: Power simulation: test for overall treatment effect

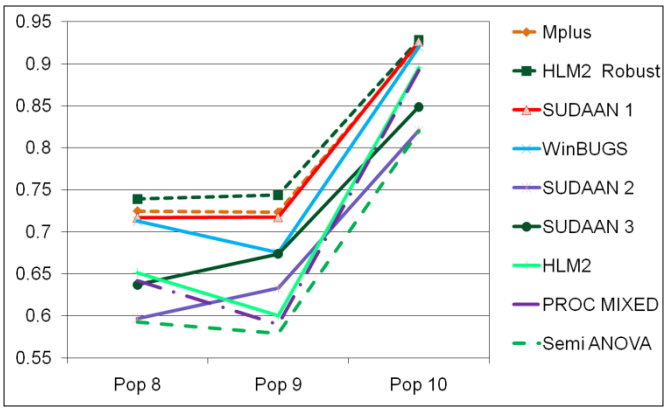

Figure 4: Power simulation: average of the tests for contrasts

5. Further Study

Due to the limitation of time and resources the simulations leaves much to be desired. For example, the WinBUGS simulations could have longer burn in and following iterations. For MPLUS, we used the default maximum likelihood estimator which proves to perform poorly. We actually find a similar problem with full maximum likelihood estimator in PROC MIXED in some trial runs. So it will be of interest to try other estimator options provided by MPLUS. For the semi-parametric procedure, there are possibilities to improve its power by using stratified randomization test on the school-averaged residuals which we can explore in future research.

References

Cheong, Y.F., Fotui, R.P., and Raudenbush, S.W. (2001). Efficiency and robustness of alternative estimators for two- and three-level models: The case of NAEP. Journal of Education and Behavioral Statistics, 26, 411-429.

Donner, A. and Klar, N. (2000). Design and Analysis of Cluster Randomization Trials in Health Research. London: Arnold.

Fan, J., and Judkins, D. (2006). Robust covariate control in cluster-randomized trials. Proceedings of the Section on Survey Research Methods of the American Statistical Association, pp. 2988-2994.

Quade, D. (1967). Rank Analysis of Covariance. Journal of the American Statistical Association, 62, 1187-1200.

Jenkins, F., Lee, H., Cheah, B. and Leytush, O. (2006). Robustness of Hierarchical Linear Modeling of Complex Survey Data When Higher Levels of Aggregation are Left out of the Model. Proceedings of the Section on Survey Research Methods of the American Statistical Association.

Koch, G.G., Amara, I.A., Davis, G.W., and Gillings, D.B. (1982). A Review of Some Statistical Methods for Covariance Analysis of Categorical Data, Biometrics, 38, 563-595.

Koch, G.G., Tangen, C.M., Jung, J-W, and Amara, I.A. (1998). Issues for covariance analysis of dichotomous and ordered categorical data from andomized clinical trials and non-parametric strategies for addressing them, Statistics in Medicine, 17, 1863-92.

LaVange, L.M., Durham, T.A. and Koch, G.G. (2005). Randomization-based nonparametrick methods for the analysis of multicentre trials. Statistical Methods in Medical Research, 14, 281-301.

Muthén, L.K., Muthén, B.O. (1998-2007). Mplus User's Guide. Fifth Edition. Los Angeles, CA: Muthén& Muthén.

Stokes, M.E., Davis, C.E., and Koch, G.G. (2000), Categorical Data Analysis Using the SAS System, 2nd edition. Cary, NC: SAS Institute Inc.

Raudenbush, S.W., Bryk, A.S., Cheong, Y.K. and Congdon, R.T., Jr. (2004). HLM6: Hierarchical linear and nonlinear modeling. Lincolnwood, IL: Scientific Software International.

Research Triangle Institute (2001). SUDAAN User's Manual, Release 8.0. Research Triangle Park, NC: Research Triangle Institute.

Rosenbaum, P. R. (2002). Covariance adjustment in randomized experiments and observational studies, Statistical Science, 286-303.

SAS Institute Inc. 2006. SAS OnlineDoc® 9.1.3. Cary, NC: SAS Institute, Inc.

Appendix. Program Code for WinBUGS simulation

1. WinBugs Model Code

```
model {
#level 1 defintion
for (i in 1:N){
y[i]~dnorm(mu[i],tau)

mu[i]<-mua[i]+mub[i]
+beta[9]*b2[i]+beta[10]*b3[i]+beta[11]*b4[i]+
beta[12]*b5[i]+beta[13]*b6[i]+beta[14]*b7[i]+
beta[15]*b8[i]+beta[16]*b9[i]+beta[17]*b10[i]

mua[i]<-beta[1]*cons[i]+beta[2]*x1[i]+beta[3]*x2[i]+
        beta[4]*x3[i]+beta[5]*r1[i]+beta[6]*r2[i]+
        beta[7]*r3[i]+beta[8]*r4[i]+u2[clus[i]]*cons[i]
```

```
mub[i]<-beta[18]*b11[i]+beta[19]*b12[i]+
            beta[20]*b13[i]+beta[21]*b14[i]+beta[22]*b15[i]+
            beta[23]*b16[i]+beta[24]*b17[i]+beta[25]*b18[i]+
            beta[26]*b19[i]+beta[27]*b20[i]+beta[28]*b21[i]+
            beta[29]*b22[i]+beta[30]*b23[i]+beta[31]*b24[i]
}
#school level
for (j in 1:Nsch){
u2[j]~dnorm(0,tau.u2)}
#Priors for fixed effects


for (k in 1:31){beta[k]~dflat()}
#prior for random terms
tau~dgamma(0.001,0.001)
sigma2<-1/tau
tau.u2~dgamma(0.001,0.001)
sigma2.u2<-1/tau.u2
}
```

2. R code for Reading SAS Generated Data and Run WinBUGS.

```
library("BRugs")
library("arm")
# ztest fuction
ztest<-function(x) {
x1<-pnorm(mean(x)/sqrt(var(x)))*2
if (x1 >1) {x2<-2-x1
return (x2)}
if (x1<1) return (x1)
}


#Create BUGSSIM function
BUGSIM<-function (iter){
write.table (iter,"C:/Project/2010JSM/bugs/iter.dat",
row.names=FALSE, col.names=FALSE)
#run sas program to create data
system ("'c:\\program files\\SAS92\\SASFoundation\\9.2\\sas.exe"
C:\\project\\2010JSM\\BUGS\\pop3_1')
pop3 <- read.table ("C:/Project/2010JSM/bugs/pop3.dat", header=TRUE)

#number of students
N <- nrow(pop3)
#number of schools
Nsch<-nlevels(factor(pop3$clus));

#constant intercep
cons<-rep(1,N)
x1<-pop3$x1
x2<-pop3$x2
x3<-pop3$x3
r1<-pop3$r1
```

```
r2<-pop3$r2
r3<-pop3$r3
r4<-pop3$r4
b1<-pop3$b1
b2<-pop3$b2
b3<-pop3$b3
b4<-pop3$b4
b5<-pop3$b5
b6<-pop3$b6
b7<-pop3$b7
b8<-pop3$b8
b9<-pop3$b9
b10<-pop3$b10
b11<-pop3$b11
b12<-pop3$b12
b13<-pop3$b13
b14<-pop3$b14
b15<-pop3$b15
b16<-pop3$b16
b17<-pop3$b17
b18<-pop3$b18
b19<-pop3$b19
b20<-pop3$b20
b21<-pop3$b21
b22<-pop3$b22
b23<-pop3$b23
b24<-pop3$b24

y<-pop3$y
clus<-pop3$clus
data <- list ("N", "Nsch","y",
"cons","x1","x2","x3","clus","r1","r2","r3","r4","b2","b3","b4","b5","b6","b7","b8","b9",
"b10"
,"b11","b12","b13","b14","b15","b16","b17","b18","b19","b20","b21","b22","b23","b24"
)
inits <- function() {list(beta=rep(0.1,31), tau=1, tau.u2=1, u2=rep(0.1, Nsch))}
parameters <- c("beta")
pop3.sim <- bugs (data, inits=inits, parameters, "C:/Project/2010JSM/bugs/pop3.bug",
n.chains=2, n.iter=800, DIC=FALSE)
attach.bugs (pop3.sim)
}

#run simulation N times, save in p and write out to pv.dat
N<-1000
p<-matrix(0,N,6)
for (iter in 1:N){
BUGSIM(iter)

#p-value for X1 using Z-test
T4vsC<-beta[,5]+beta[,6]+beta[,7]+beta[,8]
T2vs2<-beta[,5]+beta[,6]-beta[,7]-beta[,8]
T2vsC<-beta[,5]+beta[,6]
T1vsC<-beta[,5]
```

```
#Wald chisq test for overall trt effect df=4
#S=(L'beta)'(L'sigmaL)^(-1)(L'beta),  M%*%t(M)
par<-cbind(mean(beta[,5]),mean(beta[,6]),mean(beta[,7]),mean(beta[,8]))
V1<-cov(cbind(beta[,5],beta[,6],beta[,7],beta[,8]) )
chi<-par%*%solve(V1)%*%t(par)

p[iter,1]<-iter
p[iter,2]<-ztest(T4vsC)
p[iter,3]<-ztest(T2vs2)
p[iter,4]<-ztest(T2vsC)
p[iter,5]<-ztest(T1vsC)
p[iter,6]<-1-pchisq(chi,4)

write.table(p,"C:/Project/2010JSM/output/BUGpv_1.dat",row.names=FALSE,
col.names=FALSE)
}
```