R Statements to Implement Examples

Caveat/Disclaimer: This shows how to implement examples that could be done with R at the time paper was written. Parts of examples not possible with R at the time are noted as needed below.

A. Example 1 - Randomized complete block design with binomial data

```r
library(car)
library(lme4)
library(effects)
library(lsmeans)
library(msm)
library(pbkrtest)
library(MASS)
library(gee)

M <- matrix (c(
1, 0, 98, 100,
1, 1, 94, 100,
2, 0, 95, 100,
2, 1, 36, 100,
3, 0, 93, 100,
3, 1, 85, 100,
4, 0, 94, 100,
4, 1, 88, 100,
5, 0, 99, 100,
5, 1, 91, 100,
6, 0, 61, 100,
6, 1, 82, 100,
7, 0, 84, 100,
7, 1, 43, 100,
8, 0, 92, 100,
8, 1, 71, 100), ncol=4, byrow=TRUE)

intro_binomial <- data.frame (M)
colnames (intro_binomial) <-  c("block", "Treatment", "Y", "N")
intro_binomial$block <- factor(intro_binomial$block)
intro_binomial$Treatment <- factor(intro_binomial$Treatment)

#/* pre-1990: ANOVA, normal approx to Binomial */
model1=lm(Y/N~Treatment+block,data=intro_binomial)
Anova(model1,type="III")
lsmeans(model1,pairwise~Treatment)

#/* pre-1990: variance stabilizing transformation  */
model2=lm(asin((Y/N)^.5)~Treatment+block,data=intro_binomial)
Anova(model2,type="III")
lsmeans(model2,pairwise~Treatment)

#/* Back-transform the Treatment means and std errors  */
#/* so they are expressed as estimated probabilities  */
Prob_Est=sin(lsmeans(model2,pairwise~Treatment)[[1]]$lsmean)^2
Std_Err_Prob_Est=deltamethod(list(~sin(x1)^2,~sin(x2)^2),mean=lsmeans(model2,pairwise~Treatment)[[1]]$SE^2),ses=TRUE)
LowerMu=sin(lsmeans(model2,pairwise~Treatment)[[1]]$lower.CL)^2
```
UpperMu = \text{sin}(\text{lsmeans(model2, pairwise~Treatment)}[[1]]^{(\text{upper.CL})^2})

data.frame(Treatment = c(0,1), Prob_Est, Std_Err_Prob_Est, LowerMu, UpperMu)

##/* naive GLMM - borrows linear predictor from ANOVA */
model9 = glmer(cbind(Y,N-Y)~Treatment+(1|block),family="binomial",data=intro_binomial)
Anova(model9,type="III",test.statistic=c("chisq"))

### Note can only get chisq test
lsmeans(model9, pairwise~Treatment)

##/* GEE-type model */
##/* you can run this with GENMOD, too */
##/* note estimated means (data scale) IDENTICAL to normal approx ANOVA */
model11 = gee(cbind(Y,N-Y)~Treatment,id=block,corstr="exchangeable",family="binomial",data=intro_binomial)
summary(model11)

##NOTE Anova and lsmeans does not work for gee.

##/* true GLMM */
##/* note - glmer uses laplace integrat approximation */
model12 = glmer(cbind(Y,N-Y)~Treatment+(1|block)+(1|block:Treatment),family="binomial",data=intro_binomial)
Anova(model12,type="III",test.statistic=c("chisq"))

### Note can only get chisq test
lsmeans(model12, pairwise~Treatment)

B. Example 2 – Split-plot experiment with count data

library(lme4)
library(lsmeans)
library(car)

M <- matrix(c(1,1,1,7,
1,1,2,3,
1,2,1,49,
1,2,2,22,
2,1,1,6,
2,1,2,17,
2,2,1,61,
2,2,2,8,
3,1,1,1,
3,1,2,17,
3,2,1,6,
3,2,2,4,
4,1,1,1,
4,1,2,1,
4,2,1,5,
4,2,2,2,
5,1,1,1,
5,1,2,12,
5,2,1,0,
5,2,2,97,
6,1,1,4,
6,1,2,3,
6,2,1,5,
6,2,2,17), ncol=4, byrow=TRUE)
count_splt_plt <- data.frame (M)
colnames (count_splt_plt) <-  c("block", "a", "b", "count")
count_splt_plt$block <- factor(count_splt_plt$block)
count_splt_plt$a <- factor(count_splt_plt$a)
count_splt_plt$b <- factor(count_splt_plt$b)

#### Naive Poisson Model

model3=glmer(count~a*b+(1|block)+(1|block:a),family="poisson",data=count_splt_plt,nAGQ = 1)  ####Note SAS used 1 quadrature point
Anova(model3,type="III",test.statistic=c("chisq"))  ####Note df does not match SAS
lsmeans(model3,pairwise~a*b)  ####Note df does not match SAS

####Negative binomial model

###NOTE glmmADMB interacts with the lsmeans package, that is, lsmeans will not work for lmer or glmer models after glmmADMB has been installed
#install.packages("glmmADMB", repos="http://r-forge.r-project.org")
#library(glmmADMB)
model6=glmmadmb(count~a*b+(1|block)+(1|block:a),family="nbinom2",data=count_splt_plt)
summary(model6)

C. Example 3 – Repeated Measure experiment with binomial data

library(lme4)
library(lsmeans)
library(car)
library(gee)

M <- matrix (c(0, 1, 1, 15, 50, 0, 1, 2, 12, 50, 0, 1, 3, 21, 50, 0, 1, 4, 9, 50, 0, 1, 5, 12, 50, 1, 1, 1, 20, 50, 1, 1, 2, 18, 50, 1, 1, 3, 22, 50, 1, 1, 4, 35, 50, 1, 1, 5, 27, 50, 0, 2, 1, 11, 50, 0, 2, 2, 9, 50, 0, 2, 3, 15, 50, 0, 2, 4, 12, 50, 0, 2, 5, 12, 50, 1, 2, 1, 9, 50, 1, 2, 2, 14, 50, 1, 2, 3, 13, 50, 1, 2, 4, 15, 50, 1, 2, 5, 17, 50, 0, 3, 1, 13, 50, 0, 3, 2, 14, 50, 0, 3, 3, 18, 50, 0, 3, 4, 27, 50, 0, 3, 5, 32, 50, 1, 3, 1, 12, 50, 1, 3, 2, 15, 50, 1, 3, 3, 14, 50, 1, 3, 4, 14, 50, 1, 3, 5, 14, 50,
```r
ex3_rptm <- data.frame(M)
colnames(ex3_rptm) <- c("trt", "plot", "time", "y", "n")
ex3_rptm$trt <- factor(ex3_rptm$trt)
ex3_rptm$plot <- factor(ex3_rptm$plot)
ex3_rptm$time <- factor(ex3_rptm$time)

#### split-plot-in-time binomial GLMM
model1=glmer(cbind(y,n-y)~trt*time+(1|plot)+(1|plot:trt),family="binomial",data=ex3_rptm)
Anova(model1,type="III")  ####Note df does not match SAS
lsmeans(model1,pairwise~trt*time) ####Note df does not match SAS

###G-side ar(1) GLMM can not be fit in R

****/*GEE model*/
****Requires moving plot effect to fixed*/
model2=gee(cbind(y,n-y)-plot+trt*time,id=plot:trt,corstr="AR-M",Mv=1,family="binomial",data=ex3_rptm)
summary(model2) #Does not match SAS output. Look at estimated scale parameter and working correlation matrix

****/*GEE model*//*GENMOD version*/
****/*requires moving plot effect to fixed*/
model3=gee(cbind(y,n-y)-plot+trt*time,id=plot:trt,corstr="exchangeable",family="binomial",data=ex3_rptm)
summary(model3)

****/*GEE model*//*GENMOD version*/
****/*requires moving plot effect to fixed*/
###NOTE THIS IS THE SAME AS MODEL 1 in R
model4=gee(cbind(y,n-y)-plot+trt*time,id=plot:trt,corstr="AR-M",Mv=1,family="binomial",data=ex3_rptm)
summary(model4)
```