

# LME and Creatinine

## Joining Data

```
In [1]: library(data.table)
library(nlme)
library(parallel)
library(xtable)

creatinine.data <- data.table(read.csv('creat.csv'), key='group')
head(creatinine.data)
```

id	group	age	scr
1	1	35.765	0.182
1	1	37.990	0.088
3	1	51.083	0.156
3	1	52.386	0.116
3	1	52.805	0.087
3	1	52.997	0.067

```
In [2]: group.data <- data.table(
  group=c(1,2,3,4),
  kidney.disease=c(1,1,0,0),
  hypertension=c(1,0,1,0),
  key='group')
group.data
```

group	kidney.disease	hypertension
1	1	1
2	1	0
3	0	1
4	0	0

```
In [3]: creatinine.group.data <- creatinine.data[group.data]
setkey(creatinine.group.data, id)
head(creatinine.group.data)
```

id	group	age	scr	kidney.disease	hypertension
1	1	35.765	0.182	1	1
1	1	37.990	0.088	1	1
2	2	24.997	1.429	1	0
2	2	27.441	1.111	1	0
2	2	30.524	1.429	1	0
3	1	51.083	0.156	1	1

## Fitting LME Model

```
In [4]: fit.scr.model <- function(data, interaction.term=FALSE) {  
  formula = scr ~ age + kidney.disease  
  if (interaction.term) { formula <- update(formula, . ~ . + age:kidney.disease)  
  }  
  lme(formula,  
    random=reStruct(~age|id, pdClass='pdDiag'),  
    correlation=corExp(form=~age|id, nugget=TRUE),  
    method='ML', data=data,  
    control=lmeControl(maxIter=100, msMaxIter=100, niterEM=50))  
}
```

## Covariance for Subject $i$

```
In [5]: make.covariance <- function(model, i) {  
  # Error term, usually denoted epsilon  
  error.correlation <- corMatrix(model$modelStruct$corStruct)[[as.character(i)]]  
  if (is.null(error.correlation)) { error.correlation <- 1 }  
  # Random effects correlation,  $Z * G * \text{transpose}(Z)$ .  
  Z <- cbind(1, model$data[J(i), age])  
  random.correlation <- Z %%% as.matrix(model$modelStruct$reStruct$id) %%% t(Z)  
  # Convert correlation matrix into covariance matrix.  
  (random.correlation + error.correlation)*(model$sigma*model$sigma)  
}
```

## $\beta_j$ Covariance

### Maximum Likelihood Estimate

This assumes that the random effects model is correct. I can also be used as the *bread* part of the sandwich estimator.

```
In [6]: make.covariates <- function(model, data) model.matrix(model$terms, data)  
  
make.ml.parameter.covariance <- function(model) {  
  groups <- unique(model$groups)$id  
  chol2inv(chol(Reduce(`+`, lapply(groups, function(i) {  
    X <- make.covariates(model, model$data[J(i)])  
    t(X) %%% chol2inv(chol(make.covariance(model, i))) %%% X  
  }))))  
}
```

### Sandwich Estimate

```
In [7]: make.response <- function(data) data$scr

make.sandwich.parameter.covariance <- function(model) {
  bread <- make.ml.parameter.covariance(model)
  meat <- Reduce(`+`, lapply(unique(model$groups)$id, function(i) {
    X <- make.covariates(model, model$data[J(i)])
    y <- make.response(model$data[J(i)])
    weights <- chol2inv(chol(make.covariance(model, i)))
    residuals <- as.numeric(make.response(model$data[J(i)]) - X %*% model$coefficients$fixed)
    empirical.covariance <- outer(residuals, residuals)
    t(X) %*% weights %*% empirical.covariance %*% weights %*% X
  })))
  bread %*% meat %*% bread
}
```

## Bootstrap Estimate

To account for clustering of the data, we resample clusters.

```
In [8]: resample.clusters <- function(data) {
  resampled.data <- data[
    data.table(id=sample(unique(data$id), replace=TRUE),
      new.id=c(1:length(unique(data$id))),
      key='id')]
  resampled.data[,id:=NULL]
  setnames(resampled.data, 'new.id', 'id')
  setkey(resampled.data, id)
  resampled.data
}
```

## Models

### Without Interaction Term

```
In [9]: scr.model <- fit.scr.model(creatinine.group.data)
summary(scr.model, adjustSigma=FALSE)
```

Linear mixed-effects model fit by maximum likelihood

Data: data  
AIC BIC logLik  
-53.94986 -11.00314 34.97493

Random effects:

Formula: ~age | id  
Structure: Diagonal  
(Intercept) age Residual  
StdDev: 0.04643211 0.00522239 0.2633414

Correlation Structure: Exponential spatial correlation

Formula: ~age | id  
Parameter estimate(s):  
range nugget  
7.8894707 0.1759323

Fixed effects: list(formula)

	Value	Std.Error	DF	t-value	p-value
(Intercept)	1.5322224	0.03955663	965	38.73490	0
age	-0.0129154	0.00096383	965	-13.40006	0
kidney.disease	-0.2819162	0.02626024	617	-10.73548	0

Correlation:

	(Intr)	age
age	-0.849	
kidney.disease	-0.352	-0.082

Standardized Within-Group Residuals:

Min	Q1	Med	Q3	Max
-2.49081741	-0.56731338	-0.07887231	0.43113427	4.95897388

Number of Observations: 1585

Number of Groups: 619

```
In [10]: ml.parameter.covariance <- make.ml.parameter.covariance(scr.model)
sandwich.parameter.covariance <- make.sandwich.parameter.covariance(scr.model)

bootstrap.parameter.samples <- do.call(rbind, mclapply(
  replicate(1024, creatinine.group.data, simplify=FALSE), function(data) {
    resampled.data <- resample.clusters(data)
    fit.scr.model(resampled.data)$coefficients$fixed
  }, mc.cores=4))
bootstrap.parameter.covariance <- cov(bootstrap.parameter.samples)
```

```
In [11]: standard.errors <- data.frame(
  `Estimate`=scr.model$coefficients$fixed,
  `ML Standard Error`=sqrt(diag(ml.parameter.covariance)),
  `Sandwich Standard Error`=sqrt(diag(sandwich.parameter.covariance)),
  `Bootstrap Standard Error`=sqrt(diag(bootstrap.parameter.covariance)),
  check.names=FALSE)

print(xtable(standard.errors,
  caption='Standard error estimates for fixed effect parameters.',
  label='tab:standard_errors_no_interaction',
  digits=c(0, 6, 6, 6, 6)),
  booktabs=TRUE, file='standard_errors_no_interaction.tex',
  sanitize.colnames.function=identity,
  sanitize.rownames.function=identity,
  size='small')

standard.errors
```

	Estimate	ML Standard Error	Sandwich Standard Error	Bootstrap Standard Error
(Intercept)	1.53222240	0.0395566329	0.0393479582	0.041530790
age	-0.01291536	0.0009638288	0.0009409584	0.001030028
kidney.disease	-0.28191623	0.0262602392	0.0250273809	0.034516098

## With Interaction Term

```
In [12]: scr.model.interaction <- fit.scr.model(creatinine.group.data, interaction.term=TRUE)
summary(scr.model.interaction, adjustSigma=FALSE)
```

Linear mixed-effects model fit by maximum likelihood

Data: data  
 AIC BIC logLik  
 -127.9923 -79.67728 72.99617

Random effects:

Formula: ~age | id  
 Structure: Diagonal  
 (Intercept) age Residual  
 StdDev: 0.1317468 0.004823202 0.225834

Correlation Structure: Exponential spatial correlation

Formula: ~age | id  
 Parameter estimate(s):  
 range nugget

4.6700641 0.2299764

Fixed effects: list(formula)

	Value	Std.Error	DF	t-value	p-value
(Intercept)	1.1906763	0.05423200	964	21.955237	0.0000
age	-0.0031083	0.00143568	964	-2.165016	0.0306
kidney.disease	0.3135748	0.07137345	617	4.393438	0.0000
age:kidney.disease	-0.0166492	0.00185709	964	-8.965190	0.0000

Correlation:

	(Intr)	age	kdny.d
age	-0.928		
kidney.disease	-0.760	0.705	
age:kidney.disease	0.718	-0.773	-0.935

Standardized Within-Group Residuals:

Min	Q1	Med	Q3	Max
-2.30553348	-0.53490304	-0.07976542	0.36022352	5.46804808

Number of Observations: 1585

Number of Groups: 619

```
In [13]: ml.parameter.covariance.interaction <- make.ml.parameter.covariance(scr.model.interaction)
sandwich.parameter.covariance.interaction <- make.sandwich.parameter.covariance(scr.model.interaction)

bootstrap.parameter.samples.interaction <- do.call(rbind, mclapply(
  replicate(1024, creatinine.group.data, simplify=FALSE), function(data) {
    resampled.data <- resample.clusters(data)
    fit.scr.model(resampled.data, interaction.term=TRUE)$coefficients$fixed
  }, mc.cores=4))
bootstrap.parameter.covariance.interaction <- cov(bootstrap.parameter.samples.interaction)
```

```
In [14]: standard.errors.interaction <- data.frame(
  `Estimate`=scr.model.interaction$coefficients$fixed,
  `ML Standard Error`=sqrt(diag(ml.parameter.covariance.interaction)),
  `Sandwich Standard Error`=sqrt(diag(sandwich.parameter.covariance.interaction
)),
  `Bootstrap Standard Error`=sqrt(diag(bootstrap.parameter.covariance.interaction
)),
  check.names=FALSE)

print(xtable(standard.errors.interaction,
  caption='Standard error estimates for fixed effect parameters with in
teraction term.',
  label='tab:standard_errors_interaction',
  digits=c(0, 6, 6, 6, 6)),
  booktabs=TRUE, file='standard_errors_interaction.tex',
  sanitize.colnames.function=identity,
  sanitize.rownames.function=identity,
  size='small')

standard.errors.interaction
```

	Estimate	ML Standard Error	Sandwich Standard Error	Bootstrap Standard Error
(Intercept)	1.190676306	0.054231995	0.050269334	0.050769991
age	-0.003108265	0.001435677	0.001193725	0.001216667
kidney.disease	0.313574816	0.071373449	0.068324860	0.070047199
age:kidney.disease	-0.016649153	0.001857089	0.001705009	0.001706189