Estimating Equations: Only Random Intercept, No Censoring

Kyle Grosser

Model 1

$$
\begin{aligned}
Y_{ij} &= \beta_0 + X_{ij}\beta + b_i + \epsilon_{ij} \\
\beta_0 &= 1.5 \\
\beta &= (2, -1) \\
b_i &\sim \mathrm{N}(0, 1) \\
\epsilon_{ij} &\sim \mathrm{N}(0, \sigma^2)
\end{aligned}
$$

The first covariate (X_1) is generated from $Bern(0.5)$ and is time-invariant while the second covariate X_2 is generated from $N(0, 1)$ and is time-variant.

Data Simulation

```
set.seed(95)
# number of clusters; obs. per cluster; total obs
n = 1000; m = 3; N = n*m
# number of fixed effects; fixed effects covariates
p = 3
# fixed effects
beta = c(1.5, 2, -1)
# random error sd; random errors
sigma = 1
epsilon = rnorm(N, 0, sigma)
# number of random effects
q = 1
Z = matrix(1, N)
if (q > 1) { Z = cbind(Z, X[, 1:(q - 1)]) }
# random effects
b = matrix(rnorm(n*q, 0, 1), n)
# b = rt(n = n, df = 10)
# response Y
sample.data = data.frame(id = rep(1:n, each = m), Y = beta[1] + rep(b, each = m) + epsilon)
if (p > 1) {
  sample.data = sample.data %>%
    mutate(X1 = rep(rbinom(n, 1, 0.5), each = m), Y = Y + beta[2] * X1)
}
if (p > 2) {
  sample.data = sample.data %>%
    mutate(X2 = rnorm(N, 0, 1), Y = Y + beta[3] * X2)
```

```
}
# matrix M used for transforming Y
M = rbind(matrix(1, 1, m), cbind(matrix(0, m - q, q), diag(m - q)))
# M^{-1}
M.inv = solve(M)
M.inv
```

```
##      [,1] [,2] [,3]
## [1,]    1   -1   -1
## [2,]    0    1    0
## [3,]    0    0    1
```

```
head(sample.data, 10) %>% knitr::kable()
```

id	Y	X1	X2
1	1.1519263	0	-0.2646837
1	0.1473649	0	0.1534724
1	1.0595039	0	0.8289796
2	5.2576885	1	-1.2676835
2	7.5093433	1	-1.3178394
2	5.0582387	1	-0.0502993
3	-0.5421857	1	1.0059039
3	0.1508604	1	0.1028496
3	1.6544365	1	0.6685712
4	3.3244345	1	0.7446222

If the covariate X1 is time invariant, then the following cell should return exactly 3

```
# check for time invariant
sample.data %>% group_by(id, X1) %>% summarise(n = n()) %>%
  .$n %>% mean()
```

```
## [1] 3
```

Parameter Estimation

Table 2: Parameter Estimates

Method	β_1	β_2	β_3	σ^2
Truth	1.500000	2.000000	-1.0000000	1.0000000
Estimating Equation	1.528028	1.950349	-0.9999919	0.9684793
lmer()	1.528166	1.947896	-1.0022265	0.9689231

Table 3: SE Estimates

Method	β_1	β_2	β_3
Estimating Equation	0.0515445	0.0740907	0.0207627
lmer()	0.0663446	0.0821077	0.0220431

Model 2

$$
\begin{aligned}
Y_{ij} &= \beta_0 + X_{ij}\beta + b_i + \epsilon_{ij} \\
\beta_0 &= 1.5 \\
\beta &= (2, -1) \\
b_i &\sim \mathrm{N}(0, 1) \\
\epsilon_{ij} &\sim \mathrm{N}(0, \sigma^2)
\end{aligned}
$$

The two covariates (X_1 and X_2) are generated from $N(0, 1)$ and are time-variant.

Data Simulation

```
set.seed(95)
# number of clusters; obs. per cluster; total obs
n = 1000; m = 3; N = n*m
# number of fixed effects; fixed effects covariates
p = 3
# fixed effects
beta = c(1.5, 2, -1)
# random error sd; random errors
sigma = 1
epsilon = rnorm(N, 0, sigma)
# number of random effects
q = 1
Z = matrix(1, N)
if (q > 1) { Z = cbind(Z, X[, 1:(q - 1)]) }
# random effects
b = matrix(rnorm(n*q, 0, 1), n)
# b = rt(n = n, df = 10)
# response Y
sample.data = data.frame(id = rep(1:n, each = m), Y = beta[1] + rep(b, each = m) + epsilon)
if (p > 1) {
  sample.data = sample.data %>%
    mutate(X1 = rnorm(N, 0, 1), Y = Y + beta[2] * X1)
}
if (p > 2) {
  sample.data = sample.data %>%
    mutate(X2 = rnorm(N, 0, 1), Y = Y + beta[3] * X2)
}
# matrix M used for transforming Y
M = rbind(matrix(1, 1, m), cbind(matrix(0, m - q, q), diag(m - q)))
# M^{-1}
```

```
M.inv = solve(M)
head(sample.data, 10) %>% knitr::kable()
```

id	Y	X1	X2
1	0.0477059	-1.2884559	-1.7373750
1	1.6164644	0.2657511	-0.7841251
1	-0.8481266	-0.3861912	1.9642277
2	0.1287388	-0.6115430	0.6381802
2	3.6236131	0.0620428	0.6919764
2	-0.9419655	-1.1598296	1.6302458
3	-2.1091750	-0.4787670	-0.3846410
3	-2.5669283	-0.0793958	0.6618466
3	-0.7864836	-0.2478424	0.6138064
4	6.8547182	3.4338902	2.0821190

Parameter Estimation

Table 5: Parameter Estimates

Method	β_1	β_2	β_3	σ^2
Truth	1.500000	2.000000	-1.000000	1.0000000
Estimating Equation	1.520555	2.023835	-1.017935	0.9676029
lmer()	1.503365	2.030332	-1.025199	0.9685622

Table 6: SE Estimates

Method	β_1	β_2	β_3
Estimating Equation	0.0369895	0.0210669	0.0212172
lmer()	0.0290890	0.0219074	0.0213883

Model 3

$$
\begin{aligned}
Y_{ij} &= \beta_0 + X_{ij}\beta + b_i + \epsilon_{ij} \\
\beta_0 &= 1.5 \\
\beta &= (2, -1, 0.5) \\
b_i &\sim \mathrm{N}(0, 1) \\
\epsilon_{ij} &\sim \mathrm{N}(0, \sigma^2)
\end{aligned}
$$

The first covariate (X_1) is generated from $Bern(0.5)$ and is time-invariant while the second and third covariates (X_2 and X_3) are generated from $N(0, 1)$ and are time-variant.

Data Simulation

```r
set.seed(95)
# number of clusters; obs. per cluster; total obs
n = 1000; m = 3; N = n*m
# number of fixed effects; fixed effects covariates
p = 4
# fixed effects
beta = c(1.5, 2, -1, 0.5)
# random error sd; random errors
sigma = 1
epsilon = rnorm(N, 0, sigma)
# number of random effects
q = 1
Z = matrix(1, N)
if (q > 1) { Z = cbind(Z, X[, 1:(q - 1)]) }
# random effects
b = matrix(rnorm(n*q, 0, 1), n)
# b = rt(n = n, df = 10)
# response Y
sample.data = data.frame(id = rep(1:n, each = m), Y = beta[1] + rep(b, each = m) + epsilon)
if (p > 1) {
  sample.data = sample.data %>%
    mutate(X1 = rep(rbinom(n, 1, 0.5), each = m), Y = Y + beta[2] * X1)
}
if (p > 2) {
  sample.data = sample.data %>%
    mutate(X2 = rnorm(N, 0, 1), Y = Y + beta[3] * X2)
}
if (p > 3) {
  sample.data = sample.data %>%
    mutate(X3 = rnorm(N, 0, 1), Y = Y + beta[4] * X3)
}
# matrix M used for transforming Y
M = rbind(matrix(1, 1, m), cbind(matrix(0, m - q, q), diag(m - q)))
# M^{-1}
M.inv = solve(M)
head(sample.data, 10) %>% knitr::kable()
```

id	Y	X1	X2	X3
1	0.8025445	0	-0.2646837	-0.6987635
1	-0.4524844	0	0.1534724	-1.1996985
1	1.4194115	0	0.8289796	0.7198152
2	5.5291780	1	-1.2676835	0.5429791
2	7.5329573	1	-1.3178394	0.0472279
2	4.9476774	1	-0.0502993	-0.2211226
3	-1.2346252	1	1.0059039	-1.3848790
3	-0.9678401	1	0.1028496	-2.2374010
3	1.4977678	1	0.6685712	-0.3133374
4	3.7790157	1	0.7446222	0.9091623

If the covariate X1 is time invariant, then the following cell should return exactly 3

```
# check for time invariant
sample.data %>% group_by(id, X1) %>% summarise(n = n()) %>%
  .$n %>% mean()
```

```
## [1] 3
```

Parameter Estimation

Table 8: Parameter Estimates

Method	β_1	β_2	β_3	β_4	σ^2
Truth	1.500000	2.000000	-1.0000000	0.5000000	1.0000000
Estimating Equation	1.536212	1.693153	-0.9999118	0.4841089	0.9682362
lmer()	1.528138	1.947580	-1.0022051	0.4934170	0.9691972

Table 9: SE Estimates

Method	β_1	β_2	β_3	β_4
Estimating Equation	0.0515533	0.0741103	0.0207658	0.0212228
lmer()	0.7064514	0.1164484	0.0220341	0.0221976

Model 3 did not converge when I forgot to add the random intercept to the outcome??

Model 4

$$
\begin{aligned}
Y_{ij} &= \beta_0 + X_{ij}\beta + b_i + \epsilon_{ij} \\
\beta_0 &= 1.5 \\
\beta &= (2, -1, 0.5, 1.2, -2) \\
b_i &\sim \mathrm{N}(0, 1) \\
\epsilon_{ij} &\sim \mathrm{N}(0, \sigma^2)
\end{aligned}
$$

The 5 covariates (X_1 through X_5) are all generated from $N(0, 1)$ and are time-invariant.

Data Simulation

```
set.seed(95)
# number of clusters; obs. per cluster; total obs
n = 1000; m = 3; N = n*m
# number of fixed effects; fixed effects covariates
p = 6
X = rnorm((p-1)*N, 0, 1) %>% matrix(N)
colnames(X) = paste0("X", 1:(p-1))
# colnames(X) = paste0("X", 1:(p-1))
```

```r
# fixed effects
beta = c(1.5, 2, -1, 0.5, 1.2, -2)
# random error sd; random errors
sigma = 1
epsilon = rnorm(N, 0, sigma)
# number of random effects
q = 1
Z = matrix(1, N)
if (q > 1) { Z = cbind(Z, X[, 1:(q - 1)]) }
# random effects
b = matrix(rnorm(n*q, 0, 1), n)
# b = rt(n = n, df = 10)
# response Y
sample.data = data.frame(id = rep(1:n, each = m), Y = beta[1] + rep(b, each = m) + epsilon, X) %>%
  mutate(Y = Y + X %*% beta[2:p])
# matrix M used for transforming Y
M = rbind(matrix(1, 1, m), cbind(matrix(0, m - q, q), diag(m - q)))
# M^{-1}
M.inv = solve(M)
head(sample.data, 10) %>% knitr::kable()
```

id	Y	X1	X2	X3	X4	X5
1	1.088980	-1.0291204	0.4163630	-0.9800188	1.0850511	-1.4897319
1	-4.519494	-1.6155258	0.8111325	0.9511474	-0.8121382	0.2113310
1	4.150703	-0.0278795	-2.1714046	0.4682862	0.3998702	-0.4764220
2	5.408852	-0.3211276	0.7991028	0.5947595	1.5186201	-0.9083294
2	3.123355	1.8803713	1.0556129	-0.3037672	-0.1426788	0.2921004
2	3.949502	0.6968069	0.1848564	1.7547048	0.2779999	1.7830416
3	10.914391	-0.8648773	0.2742729	0.1327550	1.5576200	-4.2051865
3	-4.004843	-1.0748854	-0.8592879	1.3162617	0.0248189	2.4477129
3	3.293762	0.9944123	-0.2061429	0.6653140	-1.2070870	0.2930672
4	4.102932	-0.2300460	0.8112998	0.4402497	0.1966966	-1.7038985

Parameter Estimation

Table 11: Parameter Estimates

Method	β_1	β_2	β_3	β_4	β_5	β_6	σ^2
Truth	1.500000	2.000000	-1.000000	0.5000000	1.200000	-2.000000	1.000000
Estimating Equation	1.466165	2.006247	-1.042904	0.4503960	1.180768	-1.962924	1.025343
lmer()	1.466784	1.997967	-1.037764	0.4525325	1.179149	-1.964902	1.027734

Table 12: SE Estimates

Method	β_1	β_2	β_3	β_4	β_5	β_6
Estimating Equation	0.0372384	0.0216952	0.0212127	0.0219566	0.0214870	0.0215201
lmer()	1.5189056	0.0225761	0.0227343	0.0244088	0.0226508	0.0234729