

Bootstrap Linear Mixed-Effects Models Using SAS® Procedures

Adams Kusi Appiah, University of Nebraska Medical Center, Omaha, NE

ABSTRACT

The bootstrap resampling technique is a general method for estimating the sampling distribution of a statistic of interest and is applied in many research applications. It can obtain more robust parameter estimates and confidence intervals in situations where no assumptions about the underlying distribution of the model are available. However, the main concern of the bootstrap method is how to generate a bootstrap distribution to resemble the true distribution of the observed data. In the context of linear mixed effects models, the distribution of samples should be generated to account for between-subject variability and residual variability in the data. This paper illustrates various ways to bootstrap data for linear mixed-effects models using the SURVEYSELECT and MIXED procedures available with SAS®/STAT software. These methods were applied to assess the uncertainty of parameter estimates in linear mixed effect models with data from the National Cooperative Gallstone Study (NCGS). The parametric, semiparametric, and non-parametric bootstrap methods for generating samples and estimating the parameters are discussed. We compare the results obtained by maximum likelihood (ML), restricted maximum likelihood (REML), and the bootstrap methods.

INTRODUCTION

Longitudinal data where each individual has repeated measurements of the response variable over time frequently arise in clinical trials and observational studies. For example, repeated measurement of blood lead levels of children exposed to lead at weekly in a month after randomization of two treatment groups (4-measurement), repeated measurement of CD4 counts of AIDS patients at each month in a year (12-measurement), etc. Thus, the response variables are correlated due to repeated measures on the same individual or from the clustering of individuals across study centers and require special statistical techniques for accurate analysis and inference.

In recent years, linear mixed-effects models (LMM) are utilized extensively to analyze longitudinal data. It is an extension of simple linear models to allow both fixed and random effects. However, one of the aspects requiring investigation is parameter uncertainty. Parameter uncertainty in LMM is usually obtained from the asymptotic covariance matrix at the maximum likelihood (ML) parameter estimates assuming a multivariate normal distribution. Using the covariance matrix, the standard error (SE) obtained are based on the inverse of the observed Fisher information matrix may be biased. Assessing the quality of estimates based upon finite data is a task of fundamental importance in data analysis. The bootstrap provides a simple and powerful means of evaluating the quality of estimators. Bootstrap methods could also represent an alternative approach for estimating the SE of parameters, as well as the confidence interval.

The primary concern when bootstrapping is how to generate a bootstrap samples close to the true distribution of the original sample. In the context of linear mixed effects models, the distribution of samples should be generated to account for between-subject variability and residual variability in the data.

Although the bootstrapping method is procedurally straightforward, its application to LMM using SAS procedures may warrant some special considerations. In the following section, we will describe various ways to bootstrap data for linear mixed-effects models using the SURVEYSELECT and MIXED procedures available with SAS®/STAT software with illustrations with data from the National Cooperative Gallstone Study (NCGS).

DATA SOURCE

The data are from the National Cooperative Gallstone Study (NCGS), where one of the major interests was to study the safety of the drug chenodiol for the treatment of cholesterol gallstones. In this study, patients were randomly assigned to high-dose (750 mg per day), low-dose (375 mg per day), or placebo. This dataset consists of a subset of data on patients who had floating gallstones and who were assigned to the high-dose and placebo groups. In the NCGS it was suggested that chenodiol would dissolve gallstones but in doing so might increase levels of serum cholesterol. As a result, serum cholesterol (mg/dL) was measured at baseline and 6, 12, 20, and 24 months of follow-up. Many cholesterol measurements are missing because of missed visits, laboratory specimens were lost or inadequate, or patient follow-up was terminated.

Variable List:

Subject ID, Treatment Group (1=High dose chenodiol, 2=Placebo), Response at Baseline, Response at Month 6, Response at Month 12, Response at Month 20, Response at Month 24.

Note: Response = Serum cholesterol (. denotes missing value). The following SAS program is used to read the gallstone data set into a SAS data set.

```
DATA mwsug2018;
INPUT ID Treatment Y1 Y2 Y3 Y4 Y5;
DATALINES;
  1 1 178 246 295 228 274
  2 1 254 260 278 245 340
  3 1 185 232 215 220 292
.
. data omitted
.
47 2 230 228 266 . .
48 2 373 309 332 . .
;
Run;
```

Model and Notation

The general linear mixed-effects model is

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon} \\ \boldsymbol{\gamma} &\sim N(\mathbf{0}, \mathbf{G}) \\ \boldsymbol{\epsilon} &\sim N(\mathbf{0}, \mathbf{R}) \end{aligned}$$

where \mathbf{y} is the response vector, $\boldsymbol{\beta}$ is the vector of fixed effects and $\boldsymbol{\gamma}$ the vector of random effects. The matrix \mathbf{X} and \mathbf{Z} are the design matrices and the matrices \mathbf{G} and \mathbf{R} are the variance-covariance matrix of the random effects and residuals respectively.

The estimates of the fixed effects and random effects are obtained by solving the following equations.

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{X}^t \hat{\mathbf{V}}^{-1} \mathbf{X})^{-1} \mathbf{X}^t \hat{\mathbf{V}}^{-1} \mathbf{y} \\ \hat{\boldsymbol{\gamma}} &= \hat{\mathbf{G}} \mathbf{Z}^t \hat{\mathbf{V}}^{-1} (\mathbf{y} - \mathbf{X} \hat{\boldsymbol{\beta}}) \end{aligned}$$

where $\text{var}(\mathbf{y}) = \hat{\mathbf{V}} = \mathbf{Z}^t \hat{\mathbf{G}} \mathbf{Z} + \hat{\mathbf{R}}$

The $\hat{\mathbf{V}}$ is estimated via maximum likelihood (ML) or restricted maximum likelihood (REML).

Subject Profiles

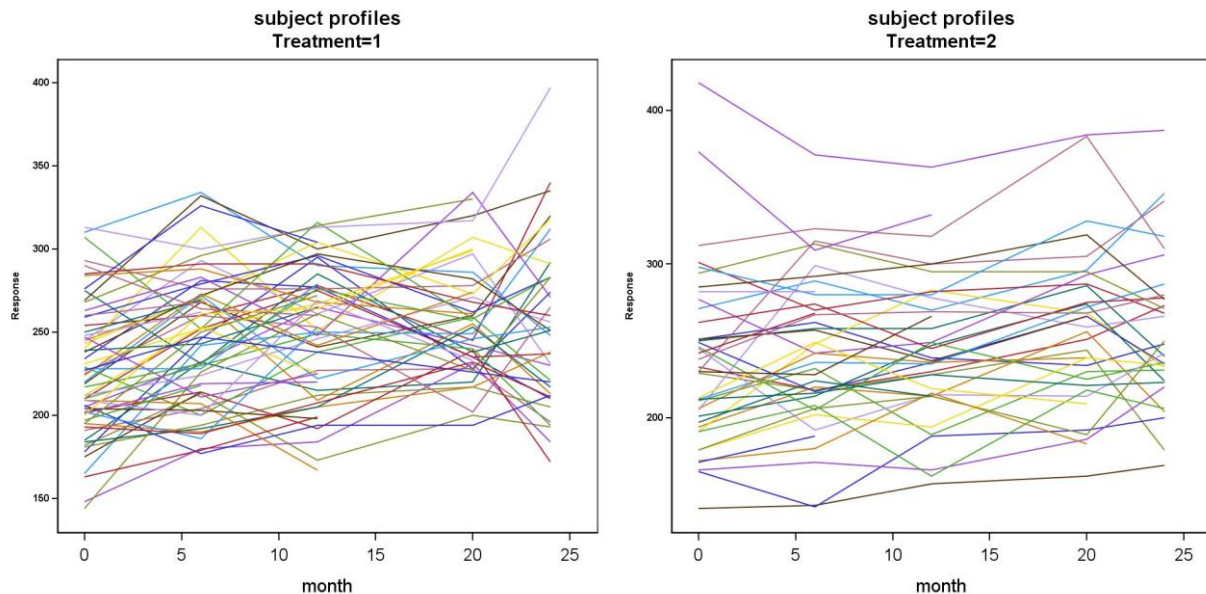


Figure 1. Subject profiles plots by treatment for the gallstone data

The subject profiles appear to show wide variability due to subject differences, whereas within subject the up and down patterns appear more like random noise. Also, from these subject profiles plots, the slopes are not very variable across subjects. Thus, we focus on a random intercept model since the profile plots do not do much to support the manner of handling repeated measures data.

The SAS MIXED procedure syntax for estimating parameters in our LMM.

PROC MIXED

```
DATA/*specifies input data set*/
METHOD/*specifies that estimation method */
NOCLPRINT /*suppresses Class Level Information completely or in parts */
CL; /*requests confidence limits for covariance parameter estimates*/
CLASS /*specifies the classification variables */
MODEL /* specifies a single dependent variable and the fixed effects*/
CHISQ /*requests chi-square tests*/
S /*displays fixed-effects parameter estimates*/
CL; /*Request 95% CL for the fixed-effects parameter estimates*/
RANDOM /*defines the random effects */
SUBJECT/*identifies the subjects in the model*/
RUN;
```

The example of SAS coding of LMM is as follows. This SAS program obtains ML parameter estimates of the gallstone data. To get the REML parameter estimates, the **METHOD=** option in the **PROC MIXED** statement is set to **reml**.

```
ODS OUTPUT SolutionF=FixedEffects
            CovParms=RandomCov;
PROC MIXED DATA=mwsug2018 METHOD=ml noclprint CL;
            CLASS ID treatment;
            MODEL y= treatment month treatment*month/CHISQ S CL;
            RANDOM intercept/ SUBJECT=ID;
RUN;
ODS OUTPUT CLOSE;
```

BOOTSTRAP METHODS

Bootstrap as the most popular resampling method is mainly used for estimating the sampling distribution of a statistic of interest. The basic bootstrap method typically has three straightforward steps:

1. Repeatedly resample n points with replacement from the original dataset of size n .
2. Compute $\hat{\theta}_i^*$ on each resample.
3. Compute ξ based on these bootstrap replications of $\hat{\theta}_i^*$ as our estimate of ξ for $\hat{\theta}_i$

The principle of the bootstrap method is illustrated in the following diagram.

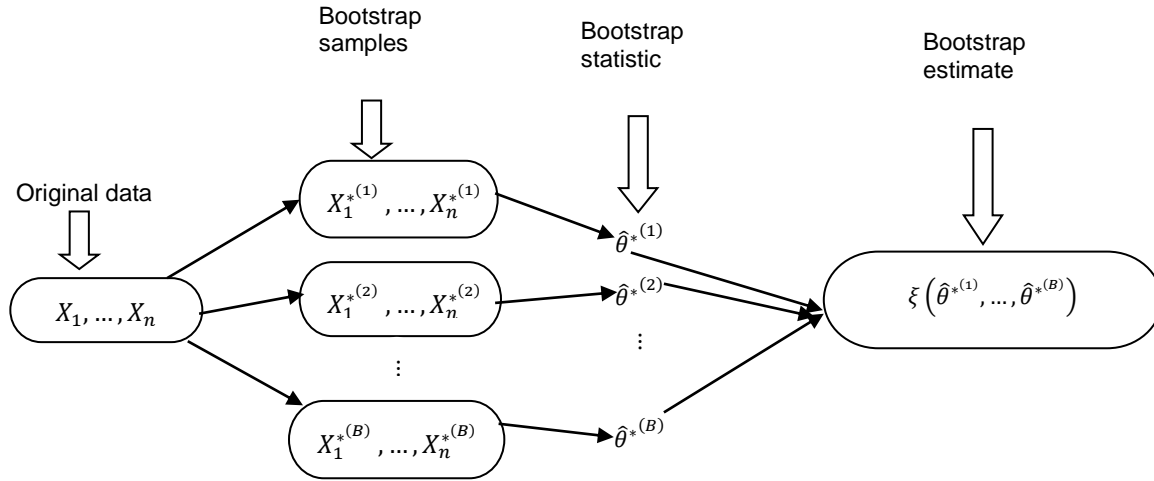


Figure 2. Schematic diagram of the bootstrap method

Let $\hat{\theta}_i^*$ be the parameter estimated for the i^{th} bootstrap sample. Given a data set, the expected value of the bootstrap estimator over the bootstrap distribution is calculated as the average of the parameter estimates from the B bootstrap samples

$$\hat{\theta}_B = \frac{\sum_{i=1}^B \hat{\theta}_i^*}{B}$$

The bootstrap standard error is obtained as the sample standard deviation of the $\hat{\theta}_i^*$

$$\widehat{se}_B = \sqrt{\frac{\sum_{i=1}^B (\hat{\theta}_i^* - \hat{\theta}_B)^2}{B - 1}}$$

A 95% bootstrap confidence interval can be constructed by calculating the 2.5th and 97.5th percentile of the bootstrap distribution

$$\hat{\theta}_{(0.025, B)} \leq \theta \leq \hat{\theta}_{(0.975, B)}$$

Non-parametric bootstrap

This method consists of resampling with replacement of the entire subjects, that is, the joint vector of the design variables and corresponding responses (y_i, X_i, Z_i) from the original data before modeling. The bootstrap is obtained as follows:

1. Sample (y_i^*, X_i^*, Z_i^*) for $i \in \{1, \dots, n\}$ with replacement from $\{(y_1, X_1, Z_1), \dots, (y_n, X_n, Z_n)\}$ in the original data
2. Fit the mixed model to the data to obtain the ML estimates
3. Repeat 1–2 a total of B times

The **SURVEYSELECT** procedure allows one to generate samples of many kinds from any an impute dataset. However, the same feature can be used to generate bootstrap samples. The basic **PROC SURVEYSELECT** syntax is given below.

```
*Basic syntax;
PROC SURVEYSELECT
  DATA=/*original data*/
  OUT=/*bootstrap data*/
  SEED=/*Specify random number seed*/
  METHOD=/*specify the type of random sampling*/
  SAMPRATE=/*Specify sampling rate*/
  OUTHITS /*give the times a record chosen*/
  REP=; /*specify the number of replicates */
RUN;
```

The example of SAS coding of the non-parametric bootstrap method of LMM is as follows. The **METHOD=** option allows us to specify the type of random sampling. For a bootstrap, we need unrestricted random sampling (URS) in the **METHOD=** option, and **SAMPRATE=**option set to 100% (i.e., obtain the sample to be of the same size as the original data set).

```
PROC SURVEYSELECT DATA=mwsug2018 OUT=outboot_NonParm
  SEED=123
  METHOD=URS
  SAMPRATE=1 /*get the sample of the same size as the original dataset*/
  OUTHITS
  REP=10000;
RUN;
```

We use the variable REPLICATE as the by-variable in the **PROC MIXED**. The **ODS OUTPUT** statement is invoked to build the datasets we want. In this case, we are interested in both fixed effects estimates (i.e., SolutionF=FixedEffects_NonParm) and random effects estimates (i.e., CovParms=RandomCov_NonParm).

```
ODS OUTPUT SolutionF=FixedEffects_NonParm
           CovParms=RandomCov_NonParm;
PROC MIXED DATA=outboot_NonParm METHOD=ml;
  BY replicate;
  CLASS id treatment;
  MODEL y= treatment month treatment*month/CHISQ S CL;
  RANDOM intercept/ SUBJECT=ID;
RUN;
ODS OUTPUT CLOSE;
```

Once the bootstrap statistics have been obtained, one may use the **UNIVARIATE** procedure to get the bootstrap estimates. The following SAS program computes non-parametric bootstrap estimates, standard errors, and the 95% confidence interval of fixed effects parameters. These bootstrap estimates are built into NonParm_BootEst_fixed SAS dataset. The random effect bootstrap estimates can be obtained similarly.

```
PROC UNIVARIATE DATA=FixedEffects_NonParm NOPRINT;
  CLASS effect;
  VAR estimate;
  OUTPUT OUT=NonParm_BootEst_fixed PCTLPTS=2.5, 97.5
  PCTLPRE=ci MEAN=mean STD=stderror;
```

RUN;

Semi-parametric bootstrap

The algorithm for semiparametric bootstrapping involves the residuals that adjust for only the fixed effects (i.e., the fitted mean structure). Once the MLEs are obtained, these residuals (marginal residuals) can be easily found via

$$\hat{\epsilon}_i = \hat{y}_i - X_i \hat{\beta} \quad i = 1, \dots, n$$

The semiparametric bootstrap procedure for the mixed model can be outlined as follows.

1. Fit the mixed model to the data and obtain ML estimates and the marginal residuals $\hat{\epsilon}_1, \dots, \hat{\epsilon}_n$
2. Sample $(\hat{\epsilon}_1^*, \dots, \hat{\epsilon}_n^*)$ with replacement from $(\hat{\epsilon}_1, \dots, \hat{\epsilon}_n)$ for $i \in \{1, \dots, n\}$ these residuals.
3. Construct the bootstrap data set using the fitted model

$$y_i^* = X_i \hat{\beta} + \hat{\epsilon}_i^*, \quad i = 1, \dots, n$$

4. Fit the mixed model to the bootstrap data to obtain the ML estimates
5. Repeat 2-4 B times

The example of SAS coding of the semi-parametric bootstrap method of LMM regression is as follows. The following SAS program fits LMM and obtains the marginal residuals via **OUTPM=** option in the **MODEL** statement and build the dataset called For_Semiboot .

```
PROC MIXED DATA=mwsug2018 METHOD=ml NOCLPRINT=1;
  CLASS id treatment;
  MODEL y = treatment month treatment*month/S CHISQ
  OUTPM=For_Semiboot RESIDUAL; /*obtain marginal residuals*/
  RANDOM intercept/ SUBJECT=ID;
```

run;

The challenge we face is that the **PROC SURVEYSELECT** was not designed in a way to resampling residuals. The hope is that we can still use this procedure to aid us to implement the semi-parametric bootstrap algorithm. To proceed, we first split the *For_Semiboot* dataset into two; one containing only the residuals to draw samples from and the other with remaining covariates. The following SAS program splits the *For_Semiboot* into two: *For_Semiboot1* and *Marginal_resid*.

```
DATA for_Semiboot1(KEEP=ID treatment month y pred counter)
  Marginal_resid(KEEP=resid);
SET for_Semiboot;
counter+1;
RUN;
```

Second, when we set **METHOD=SRS** option to request a simple random sampling and also set the **SAMPRATE=1**, the **PROC SURVEYSELECT** is smart enough to return the copies of the original data set, say 10,000 times in original order. The following SAS program replicates the original data set B number of times.

```
PROC SURVEYSELECT DATA=for_Semiboot1 OUT=outFor_Semiboot1
METHOD=SRS SAMPRATE=1
REP=&B;
RUN;
```

We resort to DATA step to bootstrap the marginal residuals B times with the following SAS program.

```
DATA out_marginal_resid;
CALL STREAMINIT (122333998);
DO replicate = 1 TO &B; /*repeat the process B times*/
  DO counter = 1 TO num; /*up to the sample size in the dataset*/
```

```

rownum= ceil(num * ranuni(122333998));
SET Marginal_resid NOBS=num POINT=rownum;
  OUTPUT;
  END;
END;
STOP;
RUN;

```

We merge the two data sets using the BY variables, REPLICATE and COUNTER, giving us y_star variable made up of the marginal predicted values and a randomly selected marginal residual.

```

DATA for_mixed;
MERGE outFor_Semiboot1 out_marginal_resid;
BY replicate counter;
y_star=pred+resid;
RUN;

```

Once we have obtained the bootstrap data, the process to get the semi-parametric bootstrap estimates follows similarly to that of the non-parametric bootstrap with the help of **PROC MIXED** with **BY** statement and the **PROC UNIVARIATE**.

Parametric bootstrap

This method resamples residuals by simulating from an estimated distribution after model fitting. The bootstrap sample is obtained as follows:

1. Fit the mixed-effect model to the data to obtain ML estimates
2. ϵ^* are samples drawn from $N(0, \hat{R} = \hat{\sigma}^2 I_n)$ distributions
3. Generate the bootstrap responses

$$y_i^* = X_i \hat{\beta} + Z \hat{\gamma}_i + \epsilon_i^*$$

4. Fit the mixed-effect model to the bootstrap data and obtain the ML estimates
5. Repeat steps 2-4 B times

The example of SAS coding of the parametric bootstrap method of LMM is as follows. The **OUTP**=option request the predicted conditional means and built into the *For_Parmboot* dataset.

```

ODS OUTPUT CovParms=RandomCovParm;
PROC MIXED DATA=mwsug2018 METHOD=ML NOCLPRINT;
  CLASS ID treatment;
  MODEL y = treatment month treatment*month/S CHISQ
  OUTP=For_Parmboot; /*obtain conditional predicted values*/
  RANDOM intercept/ SUBJECT=ID
;
run;
ODS OUTPUT CLOSE;

```

We draw samples from a normal distribution with mean 0 and variance the estimated variance from our model with following SAS program.

```

DATA Temp;
CALL STREAMINIT (122333998);
DO replicate = 1 TO &B; /* repeat many times*/
  DO counter = 1 TO 515; /* up to the size in the original data*/
    error=rand("Normal", 0, sqrt(&Sigmahat2));
    OUTPUT;
  END;
END;

```

STOP;
 RUN;

The process to obtain the parametric bootstrap estimates is similar to the semi-parametric bootstrap method. That is, use the **PROC SURVEYSELECT** to replicate the data set containing the conditional predicted values and merge with the random samples drawn from the normal distribution. The bootstrap response y_star is obtained by adding conditional predicted values and the random samples from the normal distribution. Then the estimation of the parametric bootstrap statistics and estimates are obtained by **PROC MIXED** with **BY** statement and the **PROC UNIVARIATE** respectively.

RESULTS

The results of the bootstrap method applied to the gallstone data to estimate fixed and random effects parameters are summarized in Table 1 and Table 2 respectively. The quality of the LM and REML estimators were assessed using the standard errors and the confidence interval estimates obtained from the bootstrap techniques. The uncertainties estimated by the parametric bootstrap method are the smallest compared to the ones obtained by the semi-parametric and non-parametric methods as well as the ML and REML approaches when applied to the gallstone data. This indicates that the parametric bootstrap method is more precise. The ML and REML SE estimates are pretty close to the parametric bootstrap estimates. Thus, the present LMM does not suffer much from uncertainty parameter estimation.

Table 1. Comparison of different estimation methods of fixed effects for the gallstone data.

Method		Intercept $\hat{\beta}_1 \pm \widehat{SE}$ (95% CL)	Treatment $\hat{\beta}_2 \pm \widehat{SE}$ (95% CL)	Month $\hat{\beta}_3 \pm \widehat{SE}$ (95% CL)	Treatment*Month $\hat{\beta}_4 \pm \widehat{SE}$ (95% CL)
ML		236.51±5.66 (225.42, 247.46)	-3.95±6.30 (-16.29, 8.39)	0.95±0.33 (0.31, 1.59)	0.22±0.43 (-0.63, 1.07)
REML		236.51± 5.69 (225.36, 247.66)	-3.95±6.32 (-16.34, 8.44)	0.95±0.33 (0.30, 1.60)	0.22±0.43 (-0.64, 1.07)
Bootstrap	Non-parametric	236.52±5.70 (225.29, 247.76)	-1.89±5.38 (-15.61, 7.79)	0.95± 0.36 (0.25, 1.65)	0.10±0.34 (-0.57, 0.96)
	Semi-parametric	236.56±5.74 (256.53, 247.89)	-1.99±5.63 (-16.19, 8.18)	0.95±0.38 (0.20, 1.67)	0.11± 0.36 (-0.58, 1.03)
	Parametric	236.27±4.76 (227.00, 245.74)	-1.88± 4.69 (-13.79, 6.25)	0.95±0.31 (0.35, 1.56)	0.11±0.30 (-0.44, 0.86)

ML =Maximum likelihood, REML=Restricted Maximum Likelihood, SE=standard error, CL =confidence limit.

Table 2. Comparison of different estimation methods of random effects for the gallstone data

Method		Intercept $\hat{\gamma}_0 \pm \widehat{SE}$ (95% CL)	Residual $\hat{\sigma}^2 \pm \widehat{SE}$ (95% CL)
ML		495.53±131.34 (313.13, 901.17)	1485.49±106.85 (1296.46, 1719.34)
REML		507.21± 134.93 (320.02, 924.78)	1496.70±108.05 (1305.60, 1733.30)
Bootstrap	Non-parametric	674.37±114.80 (471.15, 918.60)	1294.77± 104.27 (1091.45, 1506.22)
	Semi-parametric	18.77±30.51 (0.00, 104.15)	1943.61±147.62 (1667.32, 2239.98)
	Parametric	331.12±70.89 (199.61, 477.42)	1476.08±97.24 (1291.53, 1673.14)

ML=Maximum likelihood, REML=Restricted Maximum Likelihood, SE=standard error,CL=confidence limit.

CONCLUSION

This paper describes SAS® programs to implement various bootstrap methods, which could be used for evaluating parameter uncertainty in LMM. Comparison of the results from the parametric, non-parametric, and semi-parametric bootstrap techniques to the ML and REML were made. There are other variant versions of bootstrapping LMM that have not been discussed in this paper. For example the wild bootstrap, the double bootstrap, m-out-of-n bootstrap, etc. However, with slight modification to the SAS program, one could implement their favourite versions of the bootstrap. For future works, it would be interesting to create a SAS macro program to perform all the variant bootstrap methods available in the literature for bootstrapping LMM. It would also be interesting to extend the LMM to possibly include censoring and discuss the bootstrap methods and implementation in SAS.

REFERENCES

- Cassell, David L. "Don't Be Loopy: Re-Sampling and Simulation the SAS® Way". SAS Institute Inc., 2007. Proceedings of the 2007 SAS Global Forum. Cary, NC: SAS Institute Inc.
- Shang, Junfeng, and Joseph E. Cavanaugh. "An assumption for the development of bootstrap variants of the Akaike information criterion in mixed models." *Statistics & Probability Letters* 78.12 (2008): 1422-1429.
- Thai, Hoai-Thu, et al. "A comparison of bootstrap approaches for estimating uncertainty of parameters in linear mixed-effects models." *Pharmaceutical statistics* 12.3 (2013): 129-140.
- Wei, L.J. and Lachin, J.M. (1984). Two-Sample Asymptotically Distribution-Free Tests for Incomplete Multivariate Observations. *Journal of the American Statistical Association*, 79, 653-661

ACKNOWLEDGMENTS

The author thanks Robin High of University of Nebraska Medical Center for his review and helpful suggestions that greatly improved this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Adams Kusi Appiah
Graduate Student
University of Nebraska Medical Center Omaha
Adams.kusiappiah@unmc.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX 1. SAS CODE

```
data mwsug2018;
input ID Treatment $ Y1 Y2 Y3 Y4 Y5;
cards;
1      1      178      246      295      228      274
2      1      254      260      278      245      340
3      1      185      232      215      220      292
4      1      219      268      241      260      320
5      1      205      232      265      242      230
6      1      182      213      173      200      193
7      1      310      334      290      286      248
```

8	1	191	204	227	228	196
9	1	245	270	209	255	213
10	1	229	200	238	259	221
11	1	245	293	261	297	231
12	1	240	313	251	307	291
13	1	234	281	277	235	210
14	1	210	252	275	235	237
15	1	275	231	285	238	251
16	1	269	332	300	320	335
17	1	148	180	184	231	184
18	1	181	194	212	217	205
19	1	165	242	250	249	312
20	1	293	276	276	278	306
21	1	195	190	205	217	238
22	1	210	230	249	240	194
23	1	212	224	246	271	256
24	1	243	271	304	273	318
25	1	259	279	296	262	283
26	1	202	214	192	239	172
27	1	184	192	205	253	217
28	1	238	272	297	282	251
29	1	263	283	248	334	271
30	1	144	226	261	227	283
31	1	220	272	222	246	253
32	1	225	260	253	202	265
33	1	224	273	242	274	.
34	1	307	252	316	258	283
35	1	313	300	313	317	397
36	1	231	252	267	299	.
37	1	206	177	194	194	212
38	1	285	291	291	268	260
41	1	250	269	.	.	.
42	1	175	214	.	.	.
43	1	201	219	220	.	.
44	1	268	296	314	330	.
45	1	202	186	253	.	.
46	1	260	268	.	.	.
48	1	209	207	167	.	.
49	1	197	218	.	.	.
50	1	248	262	.	.	.
51	1	212	253	225	.	.
52	1	276	326	304	.	.
53	1	163	179	199	.	.
54	1	239	243	265	.	.
55	1	204	203	198	.	.
56	1	247	211	225	.	.
57	1	195	250	272	.	.
58	1	228	228	279	.	.
59	1	290	264	260	.	.
60	1	284	288	268	261	.
61	1	217	231	276	257	.
62	1	209	200	269	233	.
63	1	200	261	264	300	.
64	1	227	247	.	.	220
65	1	193	189	.	232	211
1	2	251	262	239	234	248
2	2	233	218	230	251	273

3	2	250	258	258	286	240
4	2	141	143	157	162	169
5	2	418	371	363	384	387
6	2	229	218	228	244	179
7	2	271	289	270	296	346
8	2	312	323	318	383	310
9	2	194	220	214	256	204
10	2	211	232	189	230	231
11	2	205	299	278	259	266
12	2	191	248	283	268	233
13	2	249	217	236	266	235
14	2	301	270	282	287	268
15	2	201	214	247	274	224
16	2	251	257	237	266	.
17	2	277	242	249	293	306
18	2	294	313	295	295	271
19	2	212	236	235	272	287
20	2	230	315	300	305	341
21	2	206	242	236	239	.
22	2	246	205	249	225	236
23	2	245	192	215	214	242
24	2	179	202	194	239	234
25	2	165	142	188	192	200
26	2	262	274	245	275	278
27	2	212	216	228	221	223
28	2	285	292	300	319	277
29	2	166	171	166	186	220
30	2	179	206	214	189	250
31	2	298	280	280	328	318
32	2	238	267	269	268	280
33	2	172	180	216	183	.
34	2	191	208	162	218	206
37	2	282	282	.	.	.
38	2	213	249	219	209	.
44	2	171	188	.	.	.
45	2	242	268	.	.	.
46	2	197	224	214	.	.
47	2	230	228	266	.	.
48	2	373	309	332	.	.

;

run;

proc print data=mwsug2018;

run;

data mwsug2018;

set mwsug2018;

month=0; y=Y1; **output;**

month=6; y=Y2; **output;**

month=12; y=Y3; **output;**

month=20; y=Y4; **output;**

month=24; y=Y5; **output;**

drop Y1 Y2 Y3 Y4 Y5;

run;

```

/*Subject Profile Plots*/

ODS GRAPHICS ON / reset = all height= 12 in width=5 in border= off ;
ods listing image_dpi=200 gpath="C:\Users\adams\OneDrive - University of
Nebraska Medical Center\MWSUG 2018\data";
ods graphics / imagename = "profile" imagefmt=jpg ;

PROC SGPLOT data=mwsug2018 noautolegend;
by treatment;
Series x=month y=y / group=id lineattrs=(pattern=1 thickness=1);
XAXIS offsetmin=0.05 offsetmax=0.05 ;
YAXIS offsetmin=0.05 offsetmax=0.05
      label='Response' labelattrs=(weight=bold size=6)
      valueattrs=(weight=bold size=6);
TITLE 'subject profiles';
run;

ods graphics off;

proc mixed data=/*specifies input data set*/
method=reml /*specifies that estimation method */
noclprint /*suppresses "Class Level Information" completely or in parts */
CL; /*requests confidence limits for covariance parameter estimates*/
class ;/*specifies the classification variables */
model /* specifies a single dependent variable and the fixed effects*/
chisq /*requests chi-square tests*/
s /*displays fixed-effects parameter estimates*/
CL; /*request 95% CL for fixed-effects parameter estimates*/
random /*defines the random effects */
subject=/*identifies the subjects in the model*/
run;

*Obtain Maximum likelihood (ML) Estimates;
ods output SolutionF=FixedEffects
           CovParms=RandomCov;
proc mixed data=mwsug2018 method=reml noclprint cl;
class id treatment;
model y= treatment month treatment*month/chisq ddfm=kr cl;
random intercept month/ subject=id;
run;
ods output close;

* output ML fixed effects estimates;
proc print data=FixedEffects;
run;

* output ML random effects estimates;
proc print data=Randomcov;
run;

*Obtain Restricted Maximum likelihood (REML) Estimates;
ods output SolutionF=FixedEffects_Reml
           CovParms=RandomCov_Reml;
proc mixed data=mwsug2018 method=reml noclprint cl;
class id treatment;
model y= treatment month treatment*month/chisq s ddfm=kr cl;
random intercept / subject=id;

```

```

run;
ods output close;

* output REML fixed effects estimates;
proc print data=FixedEffects;
run;
proc print data=FixedEffects_Reml;run;

*output REML random effects estimates;
proc print data=RandomCov_Reml;
run;

/*##### Bootstrap Methods #####*/

*PROC SURVEYSELECT to generate Bootstrap Samples;

*Basic syntax;
PROC SURVEYSELECT
DATA=/*original data*/
OUT=/*bootstrap data*/
SEED=/*Specify random number seed*/
METHOD=/*specify the type of random sampling*/
SAMPRATE=/*Specify sampling rate*/
OUTHITS /*give the times a record chosen*/
REP=; /*specify the number of replicates */
RUN;

%macro ODSOff(); /* Call prior to BY-group processing */
ods graphics off;
ods exclude all;
ods noresults;
%mend;

%macro ODSOn(); /* Call after BY-group processing */
ods graphics on;
ods exclude none;
ods results;
%mend;

%let B=10000;
/*## Non-Parametric Bootstrap ##*/
*generate bootstrap samples (case-by-case);
proc surveyselect data=mwsug2018 out=outboot_NonParm
seed=123
method=urs
samprate=1/*get the sample of the same size as the original dataset*/
outhits
rep=&B;
run;

%ODSOff;

ods output SolutionF=FixedEffects_NonParm
CovParms=RandomCov_NonParm;
proc mixed data=outboot_NonParm method=ml;

```

```

by replicate;
class id treatment;
model y= treatment month treatment*month/chisq s;
random intercept/ subject=id;
run;
ods output close;

%ODSON;

*Fixed effect bootstrap estimates;
proc univariate data=FixedEffects_NonParm noprint;
class effect;
var estimate;
output out=NonParm_BootEst_fixed pctlpts=2.5, 97.5
pctlpre=ci mean=mean std=stderror;
run;

proc print data=NonParm_BootEst_fixed;
run;

*Random effect bootstrap estimates;
proc univariate data=RandomCov_NonParm noprint;
class CovParm;
var estimate;
output out=NonParm_rand pctlpts=2.5, 97.5
pctlpre=ci mean=mean std=stderror;
run;
proc print data=NonParm_rand;
run;

/*## Semi-Parametric Bootstrap ##*/
proc mixed data=mwsug2018 method=ml noclprint=1;
class id treatment;
model y = treatment month treatment*month/s chisq
outpm=For_Semiboot residual; /*obtain marginal residuals*/
random intercept/ subject=id;
run;

*seperate the data into resid only and other variables;
data for_Semiboot1(keep=ID treatment month y pred counter)
Marginal_resid(keep=resid);
set for_Semiboot;
counter+1;
run;

* Replicate for_Semiboot1 B times;
proc surveyselect data=for_Semiboot1 out=outFor_Semiboot1
method=srs samprate=1
rep=&B;
run;

*resample marginal residuals;
data out_marginal_resid;
CALL STREAMINIT(122333998);

```

```

do replicate = 1 to &B; /*repeat the process B times*/
do counter = 1 to num; /*up to the sample size in the dataset*/
  rownum= ceil(num * ranuni(122333998));
set Marginal_resid nobs=num point=rownum;
output;
end;
end;
stop;
run;

data for_mixed;
merge outFor_Semiboot1 out_marginal_resid;
by replicate counter;
y_star=pred+resid;
run;

%ODSOFF;
ods output SolutionF=FixedEffects_SemiParmboot
            CovParms=RandomCov_SemiParmboot;
proc mixed data=for_mixed method=ml;
by replicate;
class id treatment;
model y_star= treatment month treatment*month/chisq s;
random intercept/ subject=id;
run;
ods output close;
%ODSON;

*Fixed effect bootstrap estimates;
proc univariate data=FixedEffects_SemiParmboot noprint;
class effect;
var estimate;
output out=SemiParmBootEst_fixed pctlpts=2.5, 97.5
pctlpre=ci mean=mean std=stderror;
run;
proc print data=SemiParmBootEst_fixed;
run;

*Random effect bootstrap estimates;
proc univariate data=RandomCov_SemiParmboot noprint;
class CovParm;
var estimate;
output out=SemiParm_rand pctlpts=2.5, 97.5
pctlpre=ci mean=mean std=stderror;
run;
proc print data=SemiParm_rand;
run;

/*##### Parametric Bootstrap #####*/
ods output CovParms=RandomCovParm;
proc mixed data=mwsug2018 method=ml noclprint;
class id treatment;
model y = treatment month treatment*month/s chisq
      outp=For_Parmboot; /*obtain conditional predicted values*/
random intercept/ subject=id;
run;

```

```

ods output close;

*Replicate For_Parmboot1;
data For_Parmboot1(keep=ID treatment month y pred counter);
set For_Parmboot;
counter+1;
run;

proc surveysselect data=For_Parmboot1 out=outFor_Parmboot1
method=srs samprate=1
rep=&B;
run;

*creat marco to get residual variance;
proc sql noprint;
select Estimate
into :Sigmahat1-:Sigmahat2
from RandomCovParm;
quit;

*draw samples from normal distribution;
data Temp;
CALL STREAMINIT(122333998);
do replicate = 1 to &B; /* repeat many times*/
do counter = 1 to 515; /* up to the size in the original data*/
error=rand("Normal", 0, sqrt(&Sigmahat2));
output;
end;
end;
stop;
run;

data for_mixedParm;
merge outFor_Parmboot1 temp;
by replicate counter;
y_star=pred+error;
run;

%ODSOFF;
ods output SolutionF=FixedEffects_bootParm
CovParms=RandomCov_bootParm;
proc mixed data=for_mixedParm method=ml;
by replicate;
class id treatment;
model y_star= treatment month treatment*month/chisq s;
random intercept/ subject=id;
run;
ods output close;
%ODSON;

*Fixed effect bootstrap estimates;
proc univariate data=FixedEffects_bootParm noprint;
class effect;
var estimate;
output out=BootEst_fixedParm pctlpts=2.5, 97.5

```



```
pctlpre=ci mean=mean std=stderror;
run;
proc print data=BootEst_fixedParm;
run;

*Random effect bootstrap estimates;
proc univariate data=RandomCov_bootParm noprint;
  class CovParm;
  var estimate;
  output out=randParm pctlpts=2.5, 97.5
  pctlpre=ci mean=mean std=stderror;
run;
proc print data=randParm;
run;
```