Comparing Methods for Analyzing Longitudinal Data in the Presence of Nonignorable Drop-out

Abstract This simulation study examines how several methods perform when analyzing data with nonignorable drop-out. Two methods, generalized estimating equations and mixed effect models, do not take the nonignorability of the data into account. The third method, a mixture model, is used to adjust for the appropriate missing data mechanism. When the data is moderately correlated over time, the mixture model performs best for nonignorable data highly dependent on the unobserved response, while the other two methods do better as the missingness depends more on the observed responses. When the data is highly correlated, generalized estimating equations and the mixed effects model do quite well even when the missingness is entirely dependent on the unobserved response. For real-world analysis, a sensitivity analysis is recommended since the missing data mechanism is not verifiable.

1 Introduction

Many types of missing data occur in longitudinal studies. Missing data that is missing completely at random (MCAR) occurs randomly and does not depend on the subjects' observed or unobserved measurements. Missing at random (MAR) data is missing dependent only on measurements that are observed. When the missing data mechanism is dependent on unobserved responses or measurements, this is called informative or non-ignorable (NI) missingness.

Analyzing data without accounting for the missing data mechanism can produce biased results. This simulation paper will show the results of analyzing nonignorable drop-out with a mixture model which properly accounts for this structure, a mixed effects model (MEM) which accounts

for MAR, and generalized estimating equations (GEE) which are appropriate for MCAR. Since it is not possible to test the difference between NI and MAR data with what is observed in real-world experiments, it is important to see how these methods perform for various levels of nonignorability.

2 Data Generation

Assume there are $N = 500$ subjects and five times, $t = 1, 2, 3, 4, 5$ at which each subject is scheduled to be observed. There are 250 subjects that are from group 1 ($G_i = 1$) and 250 from group 2 ($G_i = 2$). The response for subject i at time t_j is denoted as Y_{ij}. Similar to a randomized clinical trial where the groups should have similar responses at baseline, it is assumed the groups share a common intercept, but have different slopes to give the model

$$Y_{ij} = \beta_0 + \beta_1^{(1)} I(G_i = 1) t_j + \beta_1^{(2)} I(G_i = 2) t_j + \epsilon_{ij}. \tag{2.1}$$

For the data generation, $\beta_0 = 10$, $\beta_1^{(1)} = 1$, and $\beta_1^{(2)} = 1.2$, so there is a 20% difference in the rate of change between the two groups.

The responses for subject i are assumed to be normally distributed and independent between subjects, but correlated with AR(1) structure across time. This correlation model is used in longitudinal analyses to reflect the fact that measurements at closer time points are thought to be more correlated than time points that are further away. To induce these attributes, ϵ_{ij} is distributed normal with mean 0 and the following correlation structure across time j,

$$\begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \rho^4 \\ \rho & 1 & \rho & \rho^2 & \rho^3 \\ \rho^2 & \rho & 1 & \rho & \rho^2 \\ \rho^3 & \rho^2 & \rho & 1 & \rho \\ \rho^4 & \rho^3 & \rho^2 & \rho & 1 \end{pmatrix} \tag{2.2}$$

for situations where $\rho = 0.75$ and $\rho = 0.5$ to simulate both strongly correlated and moderately correlated data. The variance of ϵ_{ij} is taken to be $\sigma^2 = 1$ so the correlation matrix is equivalent to the covariance matrix.

It is assumed that there is only monotone missingness which means that if a subject is not observed at one time point, then there are no observations for that patient for any subsequent time

points. The nonresponse mechanism is modeled with logistic regression dependent only on the previous response observed and the current, possibly unobserved, response. This is written as

$$\text{logit}\left[P(R_{ij}=1 \mid \bar{R}_{i,j-1}=1)\right] = \alpha_1 + \alpha_2 Y_{i,j-1} + \alpha_3 Y_{ij} \tag{2.3}$$

where R_{ij} is the indicator that subject i responds at time t_j and $\bar{R}_{i,j} = (R_{i1}, ..., R_{ij})$ is the history of all previous response indicators up through time t_j. Another variable to express drop-out is D_i which is the drop-out time of subject i. For completers, or those who don't drop out, $D_i = 6$ since there are five observation times.

This drop-out model is the same for both treatment groups and only dependent on the previous and current response. This is a common model in the literature. Drop-out could vary by treatment, but since the responses are different for the the different treatment groups (growth rate differs by 20%), the assumption is that the responses reflect the treatment group in the way that is necessary for modeling the drop-out.

For the mixture model, the parameters are estimated separately for each missingness pattern. The slope for those that drop out at $t = 2$ is not estimable since at least two observations are needed for estimating a slope. Therefore, every subject is observed at the first two times to get around this model identifiability issue. This means $R_{i1} = R_{i2} = 1$ and (2.3) is the model for missing data at times $t = 3 - 5$. Different values of $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ will be used as discussed in the next section.

3 Simulation Design

For the simulation, both method and the drop-out model will vary. GEE for MCAR data, a MEM assuming MAR data, and a mixture model assuming nonignorable drop-out will all be used to analyze the data.

The drop-out model will vary in terms of the parameters α. For NI1, the coefficients are $\alpha_1 = 6$, $\alpha_2 = 0$, $\alpha_3 = -0.36$. The choice of $\alpha_2 = 0$ means that the data is only dependent on the unobserved response and not on any observed responses, or strongly nonignorable. NI2 has

Table 1: Response probabilities at each time

		NI1	NI2	NI3	MAR
G1	t_3	0.84	0.83	0.83	0.82
	t_4	0.79	0.75	0.77	0.76
	t_5	0.72	0.66	0.69	0.69
G2	t_3	0.82	0.80	0.81	0.81
	t_4	0.75	0.70	0.73	0.74
	t_5	0.66	0.58	0.63	0.65

$\alpha_1 = 7$, $\alpha_2 = -0.1$, $\alpha_3 = -0.36$ and NI3 has $\alpha_1 = 6.2$, $\alpha_2 = -0.2$, and $\alpha_3 = -0.2$. These two settings are varying degrees of dependence on the observed response and unobserved response. The last case, $\alpha_1 = 5.5$, $\alpha_2 = -0.36$, $\alpha_3 = 0$, gives MAR data, or missingness dependent only on the observed response. The values of α_1 vary to control the overall missing percentages at each time point.

Table 1 shows how the different choices of α affect the probability of an observed response at each time point assuming the previous response was observed. It is assumed all subjects respond at the first two time points and their probability of response is 1, so probabilities are only shown for $t = 3 - 5$. Since these probabilities depend on the value of the responses, two sets are examined in the table. The first, $\mathbf{Y} = (10, 11, 12, 13, 14)$, is the mean trajectory of group 1 with intercept=10 and slope=1 and the second, $\mathbf{Y} = (10, 11.2, 12.4, 13.6, 14.8)$, is consistent with group 2 having intercept=10 and slope=1.2. The table shows that there is a similar response probability across the methods, so any difference in results that are observed can be attributed to the method and not the amount of missing data.

4 Methods

Three methods will be used on all of the data sets produced in the simulation. Generalized estimating equations should produce unbiased response for MCAR data. The linear mixed effects model is a method to control for MAR data, and the mixture model adjusts for the nonignorable missing

data mechanism.

4.1 Generalized Estimating Equations

Generalized estimating equations are commonly used to analyze complete longitudinal data. Parameter estimates are obtained by solving the following system of equations introduced by Liang and Zeger (1986):

$$\sum_{i=1}^{N} \frac{\partial \mu_i}{\partial \beta'} \mathbf{V}_i^{-1} \left[\mathbf{Y}_i - \mu_i(\beta) \right] = 0 \tag{4.1}$$

where $\mathbf{V}_i$ is the working covariance matrix of $\mathbf{Y}_i$. An unstructured working covariance matrix was used since this would be done in practice when one does not know the structure. For our data, the mean model is $\mu_i(\beta) = \beta_0 + \beta_1^{(1)} I(G_i = 1) t_j + \beta_1^{(2)} I(G_i = 2) t_j$.

This method should be appropriate for MCAR data, but Davidian (2005) points out that since this method is not based on maximum likelihood, the results could be biased for MAR or NI missing data. PROC GENMOD in SAS fits the GEE using all available pairs of data, thus improving efficiency over a complete case estimator.

4.2 Linear Mixed Effects Model

Another model used is the linear mixed effects model,

$$Y_{ij} = \beta_0 + b_{0i} + \beta_1^{(1)} I(G_i = 1) t_j + \beta_1^{(2)} I(G_i = 2) t_j + b_{1i} t_j + \mathbf{e}_i \tag{4.2}$$

where $\mathbf{b}_i = (b_{0i}, b_{1i})$ are random subject-specific effects for both intercept and slope. Given that $\mathbf{b}_i \sim N(0, \mathbf{D})$ and $\mathbf{e}_i \sim N(0, \mathbf{R})$, then $\mathbf{Y}_i \sim N(\mathbf{X}_i \beta, \mathbf{Z}_i \mathbf{D} \mathbf{Z}_i' + \mathbf{R})$ where $\mathbf{Z}_i = (\mathbf{1}, \mathbf{t})$ and $\mathbf{X}_{ij} = (1, I(G_i = 1), I(G_i = 2))$. Hogan et al. (2004) states that a correctly specified model should give unbiased estimates under MAR when the observed likelihood is maximized. This model correctly specifies the mean, but doesn't correctly specify the autoregressive structure of the covariance. Jacqmin-Gadda et al. (2006) indicates that the large sample size and five measurement

times should still lead to a good approximation with this model. The mixed effects model can be done using PROC MIXED in SAS and using the sandwich variance estimate.

4.3 Mixture Model

The mixture model described in Fitzmaurice and Laird (2000) is used to account for nonignorable drop-out. Because the data is assumed to be normal, thus having a linear link, the mean model is

$$\begin{aligned} E\left[\mathbf{Y}_{ij} \mid D_i, \mathbf{X}_{ij}\right] &= \mathbf{X}_{ij}'\nu \\ &= \sum_{d=3}^{6} \nu_0^{(d)} I(D_i = t_d) + \nu_1^{(d,1)} I(D_i = t_d) I(G_i = 1) t_j + \nu_1^{(d,2)} I(D_i = t_d) I(G_i = 2) t_j. \end{aligned} \tag{4.3}$$

There are 8 drop-out patterns since patients can drop out at times $t = 3 - 5$ or be completers and there are two groups. There is an intercept term for each drop-out pattern and separate slope estimates for each combination of group and drop-out pattern. The parameters ν can be obtained by using PROC GENMOD in SAS and including as model terms drop-out time and drop-out time by group by time interaction. These estimates can then be averaged over drop-out probabilities, $\pi_l = P(D_i = t_d, G_i = g)$ where l indexes the set $\{(d, g), d = 3, 4, 5, 6 \text{ and } g = 1, 2\} = \{(3,1), (4,1), (5,1), (6,1), (3,2),\ (4,2),\ (5,2), (6,2)\ \}$, to get the marginal expectation and parameter estimates, β, that are desired,

$$E(Y_{ij} \mid \mathbf{X}_{ij}) = \mu_{ij} = \sum_{l=1}^{8} \pi_l \mathbf{X}_{ij}'\nu. \tag{4.4}$$

Now the intercept and slope estimates are

$$\begin{aligned} \hat{\beta}_0^{(1)} &= \hat{\pi}_1 \nu_0^{(3)} + \hat{\pi}_2 \nu_0^{(4)} + \hat{\pi}_3 \nu_0^{(5)} + \hat{\pi}_4 \nu_0^{(6)} \\ \hat{\beta}_0^{(2)} &= \hat{\pi}_5 \nu_0^{(3)} + \hat{\pi}_6 \nu_0^{(4)} + \hat{\pi}_7 \nu_0^{(5)} + \hat{\pi}_8 \nu_0^{(6)} \\ \hat{\beta}_1^{(1)} &= \hat{\pi}_1 \nu_1^{(3,1)} + \hat{\pi}_2 \nu_1^{(4,1)} + \hat{\pi}_3 \nu_1^{(5,1)} + \hat{\pi}_4 \nu_1^{(6,1)} \\ \hat{\beta}_1^{(2)} &= \hat{\pi}_5 \nu_1^{(3,2)} + \hat{\pi}_6 \nu_1^{(4,2)} + \hat{\pi}_7 \nu_1^{(5,2)} \hat{\pi}_8 \nu_1^{(6,2)}. \end{aligned} \tag{4.5}$$

The drop-out probabilities, π_l, are estimated nonparametrically by using the proportion of subjects in each drop-out and group pattern. For example, if 20 patients in group 1 dropped out at $t = 3$,

then $\hat{\pi}_1 = 20/250$.

The SAS program does not give the proper variance estimates, so the delta method is used as suggested by Fitzmaurice and Laird (2000). The variance of π_l is independent for each group so we use the partition $\pi_l = (\pi_l^{(1)}, \pi_l^{(2)})$. Then $\text{Var}(\pi_l^{(1)}) = \text{diag}(\pi_l^{(1)}) - \pi_l^{(1)}\pi_l^{(1)'}$ since $\pi_l^{(1)}$ is multinomial and similarly for $\pi_l^{(2)}$. The estimated covariance matrix of the parameters ν can be obtained from the empirical estimate in SAS, so the variance of all of the parameters is

$$\mathbf{V} = \begin{pmatrix} \text{Var}(\pi_l^{(1)}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \text{Var}(\pi_l^{(2)}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \text{Var}(\hat{\nu}) \end{pmatrix}. \tag{4.6}$$

Then the Jacobian, $\mathbf{J} = \frac{\partial \beta}{\partial \theta}$, is calculated where $\theta = (\pi_l', \nu')$. The variance of our desired parameters, β, is $\mathbf{JVJ}'$.

5 Results

The tables below show the results for each method and missing data structure. The Monte Carlo experiment was done with $s = 1000$ replications. Bias for each parameter in the model is given. Note that the mixture model gives separate estimates for the intercept, β_0, since the drop-out probabilities are different for each group. The average standard error is the mean of the standard error estimates across the simulations, while the Monte Carlo standard deviation is the standard deviation of the parameter estimates across simulations. A 95% Wald confidence interval was constructed using the standard errors for each simulation, so ideally coverage should be 0.95.

Standard errors for each of these Monte Carlo estimates are summarized in the captions below the tables. Bias and average standard error are averages across simulations, so the Monte Carlo standard error is $\sqrt{\text{bias}/s}$ and (Avg SE)$/\sqrt{s}$ respectively. Boos and Stefanski (2013) note that the standard error for a variance estimate is $\sqrt{2/s}(\text{MC SD})^2$, and $\sqrt{p(1-p)/s}$ for coverage, where $p =$ coverage.

When $\rho = 0.5$ and the missingness depends heavily on the unobserved response, it is clear from

Table 2: Analysis of parameters when $\rho = 0.5$

		NI1				NI2			
Parm	Method	Bias	Avg SE	MC SD	Coverage	Bias	Avg SE	MC SD	Coverage
β_0	GEE	0.05	0.057	0.058	0.86	0.05	0.058	0.061	0.83
	MEM	0.05	0.061	0.062	0.89	0.05	0.061	0.063	0.88
	Mix $\beta_0^{(1)}$	-0.01	0.063	0.064	0.94	-0.03	0.064	0.066	0.90
	Mix $\beta_0^{(2)}$	-0.01	0.065	0.067	0.94	-0.03	0.067	0.069	0.91
$\beta_1^{(1)}$	GEE	-0.035	0.021	0.021	0.62	-0.041	0.023	0.025	0.57
	MEM	-0.032	0.026	0.025	0.77	-0.036	0.026	0.026	0.72
	Mixture	0.005	0.031	0.031	0.95	0.021	0.032	0.033	0.90
$\beta_1^{(2)}$	GEE	-0.040	0.022	0.023	0.54	-0.046	0.024	0.026	0.51
	MEM	-0.037	0.026	0.027	0.70	-0.041	0.027	0.028	0.66
	Mixture	0.003	0.033	0.033	0.95	0.019	0.035	0.036	0.92

Average Monte Carlo Standard Errors Bias- intercept: 0.002, slopes: 0.0009; Avg SE- intercept: 1.1×10^{-5}, slopes: 3.8×10^{-6}; MC SD- intercept: 1.8×10^{-4}, slopes: 3.5×10^{-5}; Coverage- intercept: 0.01, slopes: 0.01;

the results in Tables 2 and 3 that the mixture model performs much better than the methods that don't account for NI missing data. As the missingness increases its dependence on the observed response, the GEE and MEM methods begin working better. When the data is truly MAR, the mixture model performs much worse than both the GEE and MEM methods. This only further demonstrates that a sensitivity analysis based on different assumptions should be performed since choosing an incorrect model can give very poor results.

When $\rho = 0.75$, only the results for NI1 and MAR are shown in Table 4 for brevity. Even in the NI1 case, the GEE and MEM methods perform much better than when $\rho = 0.5$. It is intuitive that these methods work better when the data is highly correlated since the observed response is strongly correlated with the unobserved response which drives the missing data mechanism.

An interesting observation is that the mixture model method seems to consistently underestimate the intercept, while the other two methods overestimate it. This pattern is reversed for estimating the slopes. Note that the methods do better for estimating the intercept than the slopes

Table 3: Analysis of parameters when $\rho = 0.5$

		NI3				MAR			
Parm	Method	Bias	Avg SE	MC SD	Coverage	Bias	Avg SE	MC SD	Coverage
β_0	GEE	0.03	0.057	0.059	0.93	0	0.060	0.061	0.94
	MEM	0.02	0.061	0.063	0.94	0	0.061	0.062	0.94
	Mix $\beta_0^{(1)}$	-0.06	0.063	0.065	0.85	-0.09	0.064	0.065	0.66
	Mix $\beta_0^{(2)}$	-0.05	0.066	0.068	0.87	-0.09	0.065	0.067	0.70
$\beta_1^{(1)}$	GEE	-0.021	0.022	0.023	0.84	0.002	0.022	0.025	0.94
	MEM	-0.017	0.026	0.026	0.90	0.005	0.026	0.026	0.94
	Mixture	0.038	0.032	0.032	0.76	0.066	0.032	0.032	0.44
$\beta_1^{(2)}$	GEE	-0.024	0.023	0.024	0.82	0.001	0.023	0.024	0.93
	MEM	-0.020	0.027	0.028	0.88	0.004	0.027	0.027	0.94
	Mixture	0.037	0.034	0.034	0.86	0.066	0.034	0.033	0.50

Average Monte Carlo Standard Errors Bias- intercept: 0.002, slopes: 0.0009; Avg SE- intercept: 1.1×10^{-5}, slopes: 3.8×10^{-6}; MC SD- intercept: 1.8×10^{-4}, slopes: 3.5×10^{-5}; Coverage- intercept: 0.01, slopes: 0.01;

across all data types. This can be attributed to the fact that there are no missing observations for the first time point.

6 Conclusion

Three different methods with four different missing data mechanisms were used to evaluate nonignorable drop-out data. The missing data mechanisms varied in how strongly they were dependent on the unobserved response. When the data is highly correlated, classical methods that don't take the nonignorability into account perform quite well, but are still a little worse than the mixture model when the data is highly nonignorable. However, when the data is truly MAR, the mixture model performs very poorly while the MEM and GEE models perform very well as expected. In this situation, not much would be lost by assuming the missingness is ignorable.

When the data is only moderately correlated and missingness is highly dependent on the unobserved value, both the generalized estimating equations and mixed effects model perform poorly,

Table 4: Analysis of parameters when $\rho = 0.75$

		NI1				MAR			
Parm	Method	Bias	Avg SE	MC SD	Coverage	Bias	Avg SE	MC SD	Coverage
β_0	GEE	0.02	0.056	0.060	0.92	-0.02	0.065	0.089	0.93
	MEM	0.02	0.056	0.057	0.93	-0.01	0.056	0.057	0.94
	Mix $\beta_0^{(1)}$	-0.02	0.058	0.060	0.92	-0.06	0.059	0.064	0.81
	Mix $\beta_0^{(2)}$	-0.01	0.059	0.061	0.94	-0.04	0.059	0.063	0.86
$\beta_1^{(1)}$	GEE	-0.020	0.022	0.030	0.77	0	0.029	0.036	0.92
	MEM	-0.017	0.022	0.022	0.87	0.006	0.023	0.022	0.94
	Mixture	0.011	0.025	0.027	0.92	0.041	0.027	0.029	0.66
$\beta_1^{(2)}$	GEE	-0.024	0.024	0.027	0.76	0	0.030	0.036	0.92
	MEM	-0.021	0.023	0.023	0.85	0.005	0.023	0.024	0.94
	Mixture	0.007	0.027	0.027	0.94	0.038	0.028	0.029	0.73

Average Monte Carlo Standard Errors Bias- intercept: 0.002, slopes: 0.0008; Avg SE- intercept: 5.3×10^{-5}, slopes: 1.9×10^{-5}; MC SD- intercept: 1.9×10^{-4}, slopes: 3.5×10^{-5}; Coverage- intercept: 0.009, slopes: 0.010;

while the mixture model does very well. As the missing data mechanism becomes more dependent on the observed responses, again the GEE and MEM methods perform much better, while the mixture model performs worse.

Since investigators can't test whether missing data is ignorable or nonignorable based off of what they observe, this simulation is important to show the repercussions of analyzing data under the incorrect assumption. This validates many researchers' advice to use a sensitivity analysis when analyzing data of this type.

References

Boos, D. D. and L. A. Stefanski, 2013: *Essential Statistical Inference: Theory and Methods*. Springer.

Davidian, M., 2005: St 732: Applied longitudinal data analysis. Lecture Notes.

Fitzmaurice, G. M. and N. M. Laird, 2000: Generalized linear mixture models for handling nonignorable dropouts in longitudinal studies. *Biostatistics*, **1**, 141–156.

Hogan, J. W., J. Roy, and C. Korkontzelou, 2004: Tutorial in biostatistics: Handling drop-out in longitudinal studies. *Statistics in Medicine*, **23**, 1455–1497.

Jacqmin-Gadda, H., S. Sibillot, C. Proust, J. Molina, and R. Thiébaut, 2007: Robustness of the linear mixed model to misspecified error distribution. *Computational Statistics and Data Analysis*, **51 (10)**, 5142–5154.

Liang, K. Y. and S. L. Zeger, 1986: Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 1322.

Little, R., 2009: Selection and pattern-mixture models. *Longitudinal Data Analysis*, G. Fitzmaurice, M. Davidian, G. Verbeke, and G. Molenberghs, Eds., Chapman and Hall/CRC Press.

Appendix: Code

```
#data generation in R
library(MASS)
library(foreign)
set.seed(1989)

#------Model complete data----------
#number of subjects
N <- 500

#number of measurement points
n <- 5

#indicator matrix of groups-first half group 1, rest group 2
group <- cbind(c(rep(1,N/2),rep(0,N/2)),c(rep(0,N/2),rep(1,N/2)))
gvect <- c(rep(1,N/2),rep(2,N/2))

#AR correlation matrix- can change value of rho
rho <-0.75 #0.5
sig <- matrix(nrow=n,ncol=n)
sig <- cbind(c(1,rho,rho^2,rho^3,rho^4),c(rho,1,rho,rho^2,rho^3),c(rho^2,rho,1,rho,rho^2),
             c(rho^3,rho^2,rho,1,rho),c(rho^4,rho^3,rho^2,rho,1))

#function to get probability of response - input 1xn vector
#can change values of alpha
alpha1 <- 5.5
alpha2 <- -0.36
alpha3 <- 0
resp <- rep(-1,n)
presp <- function(y){
  for(i in 2:n){
    resp[i] <- exp(alpha1 + alpha2*y[i-1] + alpha3*y[i])/
      (1 + exp(alpha1 + alpha2*y[i-1] + alpha3*y[i]) )
  }
  return(resp)
}
```

```
#simulate all data sets- s= number of simulations
s <- 1000
out <- NA

for(i in 1:s){
  #simulate errors with mean 0 and AR(1) var
  eps <- mvrnorm(N,rep(0,5),sig)

  #generate data
  beta0 <- 10
  beta1 <- 1
  beta2 <- 1.2
  yfull <- beta0 + beta1*group[,1]%*%matrix(c(1,2,3,4,5),1,5) +
    beta2*group[,2]%*%matrix(c(1,2,3,4,5),1,5) + eps


  #calculate response probability matrix based on full data
  respprob <- t(apply(yfull,1,presp))
  respprob[,1] <-1
  respprob[,2] <-1

  #------Get observed data----------
  #response matrix
  R <- matrix(nrow=N, ncol=n)

  #each subject has at least two responses
  R[,1] <- 1
  R[,2] <- 1

  #generate uniform number between 0 and 1
  u1 <- runif(N,0,1)

  #respond at next time if u1 is less than response probability
  R[,3] <- (u1 < respprob[,3])

  #repeat and ensure monotonicity - no response if previous entry 0
  u2 <- runif(N,0,1)
  R[,4] <- (u2 < respprob[,4] & R[,3]==1)
```

```
  u3 <- runif(N,0,1)
  R[,5] <- (u3 < respprob[,5] & R[,4]==1)

  #observed data - 0 indicates missing
  yobs <- R*yfull

  #dropout time, d=6 means completers
  d <-  (yobs[,2]!=0 & yobs[,3]==0)*3 + (yobs[,3]!=0 & yobs[,4]==0)*4+
    (yobs[,3]!=0 & yobs[,4]!=0 & yobs[,5]==0)*5
  d <- d + (d==0)*6

  outi <- cbind(i,c(1:N),gvect,d,yobs)

  #output sim, subject, group, drop time, and observed responses (0 if missing)
  out <- rbind(out,outi)
}

#name columns
out <- out[-1,]
colnames(out) <- c("sim","subject","group","tdrop","time1","time2","time3","time4","time5")

#output data to sas data set
write.table(out, "C:/Users/Rebecca Hager/Desktop/Prelim Exam/Simulation/Data/rho75/mar.txt",
            row.names = FALSE, col.names = TRUE, sep="\t")

#change values in program to get different data sets
#5nmar1 = rho=0.5 alpha1=6 alpha2=0 alpha3=-0.36
#5nmar2 = rho=0.5 alpha1=7 alpha2=-0.1 alpha3=-0.36
#5nmar3 = rho=0.5 alpha1=6.2 alpha2=-0.2 alpha3=-0.2
#5mar = rho=0.5 alpha1=5.5 alpha2=-0.36 alpha3=0
#nmar1 = rho=0.75 alpha1=6 alpha2=0 alpha3=-0.36
#nmar2 = rho=0.75 alpha1=7 alpha2=-0.1 alpha3=-0.36
#nmar3 = rho=0.75 alpha1=6.2 alpha2=-0.2 alpha3=-0.2
#mar = rho=0.75 alpha1=5.5 alpha2=-0.36 alpha3=0

#clean up
rm(list=ls())
```

```
#check how the response probability changes for various alpha and responses
alpha1 <- 6
alpha2 <- -0.36
alpha3 <- 0
n=5
resp <- rep(1,n)
y <- c(10,11,12,13,14)
#y <- c(10,11.2,12.4,13.6,14.8)

#function takes in response vector, and outputs response probabilities
#first 2 probabilities always 1 since first 2 responses observed
for(i in 3:n){
  resp[i] <- exp(alpha1 + alpha2*y[i-1] + alpha3*y[i])/
    (1 + exp(alpha1 + alpha2*y[i-1] + alpha3*y[i]) )
}
resp

*GEE Analysis in SAS;
libname out 'C:\Users\Rebecca Hager\Desktop\Prelim Exam\Simulation\Output\GEE\rho75';

*change dset for each different dataset;
%let dset=mar;

*import data;
proc import datafile='C:\Users\Rebecca Hager\Desktop\Prelim Exam\Simulation\Data\rho75\mar.txt'
     out=data
     dbms=dlm
     replace;
     delimiter='09'x;
     getnames=YES;
run;

*make 0 into missing values;
data data;
set data;
if time1=0 then time1=.;
if time2=0 then time2=.;
if time3=0 then time3=.;
```

```
if time4=0 then time4=.;
if time5=0 then time5=.;
run;

*change data to 1 line for each observation instead of just 1 line for each subject;
data data2(keep=sim subject group tdrop t resp);
set data;
by sim subject;
if first.sim or first.subject;

array time {5} time1-time5;
do i = 1 to 5;
 t = i;
 resp=time{i};
 output;
end;
run;

*do GEE analysis for each simulation and output parameters and covariance matrix;
*repeated indicates covariance across subjects- assume unstructured;
ods output GEEEmpPEst=gmparms GEERCov=gecov;
proc genmod data= data2;
by sim;
class subject group;
model resp= group*t/ dist = normal;
repeated subject=subject / type=un corrw ecovb;
run;
ods output close;

*rename and output covariance and parameters;
data out.cov_mar_&dset;
set gecov;
run;

data out.parms_mar_&dset;
set gmparms;
run;
```

```
*MEM analysis in SAS;
libname out 'C:\Users\Rebecca Hager\Desktop\Prelim Exam\Simulation\Output\MRM\rho75';

*change dset for each different dataset;
%let dset=nmar3;

*import data;
proc import datafile='C:\Users\Rebecca Hager\Desktop\Prelim Exam\Simulation\Data\rho75\nmar3.txt'
     out=data
     dbms=dlm
     replace;
     delimiter='09'x;
     getnames=YES;
run;

*make 0 into missing values;
data data;
set data;
if time1=0 then time1=.;
if time2=0 then time2=.;
if time3=0 then time3=.;
if time4=0 then time4=.;
run;

*change data to 1 line for each observation instead of just 1 line for each subject;
data data2(keep=sim subject group tdrop t resp);
set data;
by sim subject;
if first.sim or first.subject;

array time {4} time1-time4;
do i = 1 to 4;
 t = i;
 resp=time{i};
 output;
end;
run;
```

```
*Mixed effects model. Random intercept and slope term;
*use empirical for empirical covariance, not model-based;
ods output SolutionF=sf;
proc mixed data=data2 empirical noclprint;
by sim;
class subject group;
model resp = group*t /s cl covb;
random intercept t / subject=subject type=un g gcorr;
run;
ods output close;

*rename and output estimates and variance;
data out.mrm_&dset;
set sf;
run;



*Mixture model in SAS;
libname out 'C:\Users\Rebecca Hager\Desktop\Prelim Exam\Simulation\Output\Mixture model\rho75';

*change dset for each different dataset;
%let dset=mar;
*import data;
proc import datafile='C:\Users\Rebecca Hager\Desktop\Prelim Exam\Simulation\Data\rho75\mar.txt'
     out=data
     dbms=dlm
     replace;
     delimiter='09'x;
     getnames=YES;
run;

*make 0 into missing values;
data data;
set data;
if time1=0 then time1=.;
if time2=0 then time2=.;
if time3=0 then time3=.;
```

```
if time4=0 then time4=.;
if time5=0 then time5=.;
run;

*change data to 1 line for each observation instead of just 1 line for each subject;
data data2(keep=sim subject group tdrop td6 td3 td4 td5 t resp);
set data;
by sim subject;
if first.sim or first.subject;

*dropout time indicators;
if tdrop=6 then td6=1; else td6=0;
if tdrop=3 then td3=1; else td3=0;
if tdrop=4 then td4=1; else td4=0;
if tdrop=5 then td5=1; else td5=0;

array time {5} time1-time5;
do i = 1 to 5;
 t = i;
 resp=time{i};
 output;
end;
run;

*get percentage of subjects missing for each drop-out and group level;
ods output CrossTabFreqs=ctf (keep=sim group tdrop Frequency colPercent);
proc freq data=data;
by sim;
tables group*tdrop;
run;
ods output close;

*data manip - get pattern for each group;
data out.ctf_&dset;
set ctf;
where group ^=. and tdrop^=.;
run;
```

```
*Mixture model by including drop-out time and interaction;
*use empirical covariance over model, repeated give covariance structure across subject;
ods output GEEEmpPEst=gmparms GEERCov=gecov;
proc genmod data= data2;
by sim;
class subject group tdrop;
model resp= tdrop group*tdrop*t/ noint dist = normal;
repeated subject=subject / type=un corrw ecovb;
run;
ods output close;

*rename and output covariance and parameters;
data out.cov_mix_&dset;
set gecov;
run;

data out.parms_mix_&dset;
set gmparms;
if parm ^= "Intercept";
run;



#Get GEE results in R using SAS output
library(sas7bdat)

#true parms
int <- 10
s1 <- 1
s2 <- 1.2

#number of simulations
s <- 1000

#number of measurement points
n <- 5

#number of patients
N <- 500
```

```
#read in SAS data sets
#cols = sim, parm (parm1-parm4), parm1, parm2, parm3, parm4
covm <- read.sas7bdat("C:/Users/Rebecca Hager/Desktop/Prelim Exam/Simulation/Output/GEE/rho75/cov_mar_mar.sas7bda

#cols = sim parm level estimate sterr lcl ucl z p
estm <- read.sas7bdat("C:/Users/Rebecca Hager/Desktop/Prelim Exam/Simulation/Output/GEE/rho75/parms_mar_mar.sas7b

#initiate variables for loop over simulation
betahat <- matrix(nrow=s,ncol=3)
coveri <- rep(0,s)
covers1 <- rep(0,s)
covers2 <- rep(0,s)
bias <- matrix(nrow=s,ncol=3)
varf <- matrix(nrow=s,ncol=3)
cover <- matrix(nrow=s,ncol=3)

#loop over simulation and calculate bias, variance, and coverage
for(i in 1:s){

  #get bias
  betahat[i,] <- estm[(estm[,1]==i),4]
  bias[i,] <- betahat[i,] - c(int,s1,s2)

  varf[i,] <- diag(as.matrix(covm[(covm[,1]==i),3:5]))

  #get upper and lower limits for the CI
  up <- betahat[i,] + 1.96*sqrt(varf[i,]) #estm[(estm[,1]==i),7]
  low <- betahat[i,] - 1.96*sqrt(varf[i,]) #estm[(estm[,1]==i),6]

  #for each parameter, set cover=1 if estimate in CI
  if(low[1]<= int & int <= up[1]){ coveri[i]=1 }
  if(low[2]<= s1 & s1 <= up[2]){ covers1[i]=1 }
  if(low[3]<= s2 & s2 <= up[3]){ covers2[i]=1 }
  cover[i,]<- c(coveri[i],covers1[i],covers2[i])
}

#Bias & MC SE
```

```
colMeans(bias)
sqrt(apply(bias,2,var)/s)

#SD & MC SE
sqrt(colMeans(varf))
sqrt(apply(varf,2,var)/s)

#MC SE (variance of bias estimates) & MC SE
sqrt(apply(bias,2,var))
sqrt(2/s)*apply(bias,2,var)

#Coverage & MC SE
colSums(cover)/s
sqrt(colSums(cover)/s * (1-colSums(cover)/s)/s)

#clean up
rm(list=ls())



#Get MEM results in R using SAS output
library(sas7bdat)

#true parms
int <- 10
s1 <- 1
s2 <- 1.2

#number of simulations
s <- 1000

#number of measurement points
n <- 5

#number of patients
N <- 500

#read in SAS dataset
#cols = sim effect group estimate sterr df tval p alpha lcl ucl
```

```
estm <- read.sas7bdat("C:/Users/Rebecca Hager/Desktop/Prelim Exam/Simulation/Output/MRM/rho75/mrm_mar.sas7bdat")

#initiate variables for loop over simulation
betahat <- matrix(nrow=s,ncol=3)
coveri <- rep(0,s)
covers1 <- rep(0,s)
covers2 <- rep(0,s)
bias <- matrix(nrow=s,ncol=3)
se <- matrix(nrow=s,ncol=3)
cover <- matrix(nrow=s,ncol=3)

#loop over simulation and calculate bias, variance, and coverage
for(i in 1:s){
betahat[i,] <- estm[(estm[,1]==i),4]
bias[i,] <- betahat[i,] - c(int,s1,s2)

se[i,] <- estm[(estm[,1]==i),5]

#get upper and lower limits for the CI
up <- betahat[i,] + 1.96*se[i,]
low <- betahat[i,] - 1.96*se[i,]

#for each parameter, set cover=1 if estimate in CI
if(low[1]<= int & int <= up[1]){ coveri[i]=1 }
if(low[2]<= s1 & s1 <= up[2]){ covers1[i]=1 }
if(low[3]<= s2 & s2 <= up[3]){ covers2[i]=1 }
cover[i,]<- c(coveri[i],covers1[i],covers2[i])
}

#Bias & MC SE
colMeans(bias)
sqrt(apply(bias,2,var)/s)

#SD & MC SE
colMeans(se)
apply(se,2,var)/sqrt(s)

#MC SE (variance of bias estimates) & MC SE
```

```
sqrt(apply(bias,2,var))
sqrt(2/s)*apply(bias,2,var)

#Coverage & MC SE
colSums(cover)/s
sqrt(colSums(cover)/s * (1-colSums(cover)/s)/s)

#clean up
rm(list=ls())



#Get Mixture model results in R using SAS output
#in paper nu=beta(code), and alpha=beta(code)
library(sas7bdat)

#number of simulations
s <- 1000

#number of measurement points
n <- 5

#number of patients
N <- 500

#true values (int, int, slope1, slope2)
true <- c(10,10,1,1.2)

#read in SAS datasets
#cols = sim, parm (parm1-parm4), parm1, parm2, parm3, parm4
cov <- read.sas7bdat("C:/Users/Rebecca Hager/Desktop/Prelim Exam/Simulation/Output/Mixture model/rho75/cov_mix_ma
est <- read.sas7bdat("C:/Users/Rebecca Hager/Desktop/Prelim Exam/Simulation/Output/Mixture model/rho75/parms_mix_

#cols=sim, group, tdrop, freq - has drop-out/group proportions
ctf <- read.sas7bdat("C:/Users/Rebecca Hager/Desktop/Prelim Exam/Simulation/Output/Mixture model/rho75/ctf_mar.sa

#set up holder variables
int1 <- rep(NA,s)
```

```
int2 <- rep(NA,s)
slope1 <- rep(NA,s)
slope2 <- rep(NA,s)
varfin <- matrix(nrow=s,ncol=4)
#holds all estimates
esta <- matrix(nrow=s,ncol=4)
cover <- matrix(0,nrow=s,ncol=4)

#loop over simulation
for(i in 1:s){

  #get betahat
  betahat <- est[(est[,1]==i),5]

  #get pihat
  pihat <- ctf[(ctf[,1]==i),4]/(N/2)

  #var of pihat - separate for each group
  v.pi1 <- diag(pihat[1:4]) - pihat[1:4]%*%t(pihat[1:4])
  v.pi2 <- diag(pihat[5:8]) - pihat[5:8]%*%t(pihat[5:8])
  zer <-  matrix(0,nrow=nrow(v.pi1),ncol=ncol(v.pi2))
  v.pi <- as.matrix(rbind(cbind(v.pi1,zer),cbind(zer,v.pi2)))/(N/2)

  #var of coefficients
  v.beta <- cov[(cov[,1]==i),]
  v.beta <- v.beta[,3:ncol(v.beta)]

  #matrix of 0
  z <- matrix(0,nrow=nrow(v.beta),ncol=ncol(v.pi))

  #variance of coeff and pi
  v1 <- cbind(v.beta,z)
  colnames(v1) <- paste("v", 1:20, sep="")
  v2 <- cbind(t(z),v.pi)
  colnames(v2) <- paste("v", 1:20, sep="")
  v <-as.matrix(rbind(v1,v2))

  #get estimates of alpha
```

```
int1[i] <- pihat[1]*betahat[1]+pihat[2]*betahat[2]+pihat[3]*betahat[3]+
  pihat[4]*betahat[4]

int2[i] <- pihat[5]*betahat[1]+pihat[6]*betahat[2]+pihat[7]*betahat[3]+
  pihat[8]*betahat[4]

slope1[i] <- pihat[1]*betahat[5]+pihat[2]*betahat[6]+pihat[3]*betahat[7]+
  pihat[4]*betahat[8]

slope2[i] <- pihat[5]*betahat[9]+pihat[6]*betahat[10]+pihat[7]*betahat[11]+
  pihat[8]*betahat[12]

esta[i,] <- c(int1[i],int2[i],slope1[i],slope2[i])

#Jacobian
j1 <- c(pihat[1],pihat[2],pihat[3],pihat[4],
        0,0,0,0,0,0,0,0,betahat[1],betahat[2],betahat[3],betahat[4],0,0,0,0)

j2 <- c(pihat[5],pihat[6],pihat[7],pihat[8],
        0,0,0,0,0,0,0,0,0,0,0,0,betahat[1],betahat[2],betahat[3],betahat[4])

j3 <- c(0,0,0,0,pihat[1],pihat[2],pihat[3],pihat[4],0,0,0,0,
        betahat[5],betahat[6],betahat[7], betahat[8], 0, 0, 0, 0)

j4 <- c(0,0,0,0,0,0,0,0,pihat[5],pihat[6],pihat[7],pihat[8],0,0,0,0,
        betahat[9],betahat[10],betahat[11], betahat[12])

J <- as.matrix(rbind(j1,j2,j3,j4))

#delta method for var of alpha
covfin <- J %*% v %*% t(J)
varfin[i,] <- diag(covfin)

#get upper and lower limits for the CI
up <- esta[i,] + 1.96*sqrt(varfin[i,])
low <- esta[i,] - 1.96*sqrt(varfin[i,])

#for each parameter, set cover=1 if estimate in CI
```

```
  if(true[1]>=low[1] & true[1]<=up[1]){ cover[i,1]=1}
  if(true[2]>=low[2] & true[2]<=up[2]){ cover[i,2]=1}
  if(true[3]>=low[3] & true[3]<=up[3]){ cover[i,3]=1}
  if(true[4]>=low[4] & true[4]<=up[4]){ cover[i,4]=1}


}


#Bias & MC SE
colMeans(esta)-true
sqrt(apply(esta,2,var)/s)

#SD & MC SE
sqrt(colMeans(varfin))
sqrt(apply(varfin,2,var)/s)

#MC SE (variance of bias estimates) & MC SE
sqrt(apply(esta,2,var))
sqrt(2/s)*apply(esta,2,var)

#Coverage & MC SE
colSums(cover)/s
sqrt(colSums(cover)/s * (1-colSums(cover)/s)/s)



#clean up
rm(list=ls())
```