University of Pennsylvania
UPenn Biostatistics Working Papers

Year 2016 *Paper* 45

Maximum Likelihood Based Analysis of Equally Spaced Longitudinal Count Data with Specified Marginal Means, First-order Antedependence, and Linear Conditional Expectations

Victoria Gamerman[*] Matthew Guerra[†]
Justine Shults[‡]

[*] Division of Biostatistics, University of Pennsylvania Perelman School of Medicine; Boehringer-Ingelheim Pharmaceuticals, Inc., vica@mail.med.upenn.edu

[†] Division of Biometrics III, OB, OTS, CDER, FDA, matthew.guerra@fda.hhs.gov

[‡] Division of Biostatistics, University of Pennsylvania Perelman School of Medicine, jshults@mail.med.upenn.edu

Maximum Likelihood Based Analysis of Equally Spaced Longitudinal Count Data with Specified Marginal Means, First-order Antedependence, and Linear Conditional Expectations

Victoria Gamerman, Matthew Guerra, and Justine Shults

Abstract

This manuscript implements a maximum likelihood based approach that is appropriate for equally spaced longitudinal count data with over-dispersion, so that the variance of the outcome variable is larger than expected for the assumed Poisson distribution. We implement the proposed method in the analysis of two data sets and make comparisons with the semi-parametric generalized estimating equations (GEE) approach that incorrectly ignores the over-dispersion. The simulations demonstrate that the proposed method has better small sample efficiency than GEE. We also provide code in R that can be used to recreate the analysis results that we provide in this manuscript.

Maximum Likelihood Based Analysis of Equally Spaced Longitudinal Count Data with Specified Marginal Means, First-order Antedependence, and Linear Conditional Expectations[☆☆]

Victoria Gamerman[a,b], Matthew Guerra[c], Justine Shults[a,*]

[a]*Department of Biostatistics, University of Pennsylvania, 423 Guardian Drive, Philadelphia, PA 19104*
[b]*Boehringer-Ingelheim Pharmaceuticals, Inc., 900 Ridgebury Road, Ridgefield, CT 06877*
[c]*Division of Biometrics III, OB, OTS, CDER, FDA, Silver Spring, MD 20993*

Abstract

This manuscript implements a maximum likelihood based approach that is appropriate for equally spaced longitudinal count data with over-dispersion, so that the variance of the outcome variable is larger than expected for the assumed Poisson distribution. We implement the proposed method in the analysis of two data sets and make comparisons with the semi-parametric generalized estimating equations (GEE) approach that incorrectly ignores the over-dispersion. The simulations demonstrate that the proposed method has better small sample efficiency than GEE. We also provide code in R that can be used to recreate the analysis results that we provide in this manuscript.

Keywords:
Count Data, First-Order Antedependence, Generalized Estimating Equations, Markov Property, Maximum Likelihood Estimation, Over-dispersion

[*] Corresponding author

Email addresses: vica@mail.med.upenn.edu (Victoria Gamerman), Matthew.Guerra@fda.hhs.gov (Matthew Guerra), jshults@mail.med.upenn.edu (Justine Shults)

URL: https://dbe.med.upenn.edu/biostat-research/JustineShults (Justine Shults)

March 25, 2016

1. Introduction

Longitudinal count data are often encountered in scientific studies. For example, Thall and Vail (1990) analyzed repeated seizure counts on subjects in a clinical trial. Common features of serial count data include intra-subject correlation, due to similarity between the repeated measurements on each participant, and over-dispersion, which occurs when the variance is larger than expected for the assumed distribution of the outcome variable (Efron, 1992). Poisson regression is often applied for analysis of count data, but is usually not appropriate for longitudinal studies because it ignores intra-subject correlations and over-dispersion. Generalized Poisson regression (Consul and Famoye, 1992) allows for both over- and under- dispersion, but assumes independence of measurements.

In this paper we implement a maximum-likelihood based method for the analysis of longitudinal count data with over-dispersion induced by the serial correlation of measurements. Key assumptions of the approach include the first-order Markov property and linearity of the expectations for the conditional distributions, which are assumed to be Poisson. In addition, we assume that the correlation between adjacent measurements on a subject is constant.

The assumptions of the first-order Markov property, linearity in the conditional expectations, and constant adjacent correlations have been shown to induce a first-order autoregressive AR(1) correlation structure for the repeated outcomes on each subject (Guerra and Shults, 2014). The AR(1) structure forces a decline in the intra-subject correlations with increasing separation in time. Our method is therefore most appropriate for analysis of equally spaced longitudinal count data with over-dispersion.

Other approaches for analysis of over-dispersed longitudinal count data include semi-parametric approaches such as generalized estimating equations (GEE) (Liang and Zeger, 1986). GEE is widely used because it does not require specification of the full likelihood that can be quite complex for longitudinal discrete data. However, GEE does not account for over-dispersion. In addition, the relative ease of application of GEE for discrete data can also be a potential limitation for the approach. When only the first two moments of the distribution of the outcome variable are estimated, as they are for GEE, it is possible to obtain estimates that are not compatible with any valid parent distribution. As cautioned by Molenberghs and Kenward (2010), "the parent provides a natural description of the framework into which the semi-parametrically specified parameters fit. The implication is that such semi-parametric methods as GEE1, GEE2, ALR, etc. can always be applied because there is always a valid parent, and hence a probabilistic basis."

We make comparisons with GEE because GEE is widely used for analysis of longitudinal discrete data. We also use GEE to obtain starting values for estimation. However, we confirm that the GEE based starting values satisfy constraints that are sufficient to ensure the existence of a valid parent distribution. We conduct simulations for moderately sized samples to demonstrate that when the likelihood is correctly specified, we have improved efficiency in estimation of the regression and correlation parameters for our approach relative to GEE.

Another model for longitudinal count data is the class of generalized linear mixed-effects models that incorporate random effects in the linear predictor. However, the implementation of likelihood based methods that involve random effects can be computationally challenging (p. 75, Fitzmaurice et al. 2008). In addition, in contrast to GEE, for mixed models it is not straightforward to specify a particular working correlation structure for the repeated measurements on subjects. For example, the AR(1) correlation structure is not among the covariance models that were suggested by Thall and Vail (1990). Mixed-effects models are typically employed when the goal is to estimate effects that are subject specific, because the analysis results are conditional on the random effects (Gardiner, Lou, and Roman, 2009).

In general, likelihood based approaches like the one we implement in this paper enjoy several general advantages. Unlike semi-parametric approaches, they yield an estimated likelihood that can be used to conduct likelihood ratio tests and to compare the fit of nested models using criteria such as the Akaike information criterion (AIC) (Akaike, 1974) and Bayesian information criterion (BIC) (Schwarz, 1978). Maximum likelihood estimators are also most (asymptotically) efficient among a wide class of estimators (Serfling, 2011) when the distribution is correctly specified. Our method in particular, allows for specification of the usual model for the marginal mean for Poisson data, while also accounting for over-dispersion and serial correlation in the data via the induced AR(1) correlation structure.

In Section 2 we discuss the notation, model assumptions, the likelihood and likelihood equations. In Section 3 we discuss an application of the methods followed by simulation studies in Section 4. We conclude with a discussion in Section 5.

2. Methods

2.1. Notation and Model Assumptions

The data comprise realizations y_{ij} of ordered discrete random variables Y_{ij} that are measured on subject i at time t_{ij} ($i = 1, \ldots, m$ and $j = 1, \ldots, n_i$). Associated with each y_{ij} is a vector of explanatory variables (covariates) $x_{ij} = (x_{ij1}, \ldots, x_{ijp})'$. The expected value of measurement Y_{ij} on subject i is given by

$$E(Y_{ij}) = \mu_{ij} = \lambda_{ij}, \tag{1}$$

and the variance by $\text{var}(Y_{ij}) = \sigma_{ij}^2$.

We assume that observations on different subjects are independent. Further, the measurements within subjects are correlated with a structure that depends on parameter α. Let $\text{cov}(Y_{ij}, Y_{ik})$ represent the covariance and $\text{corr}(Y_{ij}, Y_{ik})$ represent the correlation between Y_{ij} and Y_{ik}.

We make three assumptions. First, we assume first-order antedependence, such that each Y_{ij}, given the immediate antecedent Y_{ij-1}, is independent of all further preceding variables (Gabriel, 1962). The joint probability mass function of $Y_{i1}, \ldots, Y_{in_i}$ can then be expressed as

$$\begin{aligned} &\mathrm{P}(Y_{i1} = y_{i1}, Y_{i2} = y_{i2}, \ldots, Y_{in_i} = y_{in_i}) = \\ &\qquad \mathrm{P}(Y_{i1} = y_{i1})\mathrm{P}(Y_{i2} = y_{i2}|Y_{i1} = y_{i1}) \cdots \mathrm{P}(Y_{in_i} = y_{in_i}|Y_{in-1} = y_{in-1}). \end{aligned} \tag{2}$$

First-order antedependence is also referred to as the first-order Markov property in the literature (Feller, 1968, p. 419).

Second, we assume that the correlation between adjacent measurements on a subject is constant, implying that

$$\text{corr}(Y_{ij}, Y_{ij-1}) = \alpha$$

where $i = 1 \ldots, m$ and $j = 2, \ldots, n_i$. Third, we assume that the conditional expectation of Y_{ij} given Y_{ij-1} is a linear function of Y_{ij-1}, such that

$$\mathrm{E}(Y_{ij} \mid Y_{ij-1}) = a_{ij} + b_{ij} Y_{ij-1},$$

for $i = 1 \ldots, m$ and $j = 2, \ldots, n_i$.

These three assumptions imply the following results. From Theorem 2.1 of Guerra and Shults (2014), the conditional expectation is given by

$$\mathrm{E}\left(Y_{ij}|Y_{ij-1}\right) = \mu_{ij} + \alpha\sigma_{ij}/\sigma_{ij-1}\left(Y_{ij-1} - \mu_{ij-1}\right), \tag{3}$$

where $\mu_{ij} = \mathrm{E}(Y_{ij})$, $\alpha = \mathrm{corr}(Y_{ij-1}, Y_{ij})$, ${\sigma_{ij}}^2 = \mathrm{var}(Y_{ij})$, and

$${\sigma_{ij}}^2 \quad = \quad \frac{1}{1-\alpha^2}\mathrm{E}(\mathrm{var}(Y_{ij}\,|\,Y_{ij-1})), \tag{4}$$

where $i = 1, \ldots, m$ and $j = 2, \ldots, n_i$.

Next, from Theorem 2.2 of Guerra and Shults (2014), the correlation $\mathrm{corr}(Y_{ij}, Y_{ij+t})$ between Y_{ij} and Y_{ij+t} for $t > 0$ can be expressed as

$$\begin{aligned}\mathrm{corr}(Y_{ij}, Y_{ij+t}) &= \prod_{k=j}^{j+t-1} \mathrm{corr}(Y_{ij}, Y_{ij+1}) \\ &= \prod_{k=j}^{j+t-1} \alpha \\ &= \alpha^t.\end{aligned}$$

The induced correlation structure for $(Y_{i1}, \ldots, Y_{in_i})'$ is therefore an AR(1) structure.

This AR(1) structure is plausible for longitudinal data because it requires the correlation between measurements on a subject to decline with increasing separation in time. For example, if $\alpha = 0.5$, then the correlation between the 1st and 2nd measurements is 0.5, while the correlation between 1st and 3rd measurements is $(0.5)^2 = 0.25$.

2.2. Poisson Likelihood

We assume Poisson distributions for the marginal and conditional distributions in Equation 2. For each $i = 1, \ldots, m$, the distribution of Y_{i1} is Poisson with $\mu_{i1} = \lambda_{i1} = \exp\left(x'_{i1}\beta\right)$ and ${\sigma_{i1}}^2 = \lambda_{i1}$, where β is a $p \times 1$ vector of regression parameters. Then, for $j = 2, \ldots, n_i$, the *conditional* distribution of Y_{ij} given Y_{ij-1} is Poisson with conditional mean $\mathrm{E}\left(Y_{ij}|Y_{ij-1}\right) = {\lambda_{ij}}^*$ given by Equation 3, with

$$\mu_{ij} = \lambda_{ij} = \exp\left(x'_{ij}\beta\right), \tag{5}$$

and

$${\sigma_{ij}}^2 = \lambda_{ij}/(1-\alpha^2), \tag{6}$$

for $j = 2, \ldots, n_i$ and $i = 1, \ldots, m$. The Y_{ij} are over-dispersed relative to the Poisson distribution if $j \geq 2$ and $\alpha \neq 0$, because in this case ${\sigma_{ij}}^2 = \phi_{ij}\lambda_{ij}$, where $\phi_{ij} > 1$.

The likelihood can then be expressed as

$$\begin{aligned} L(\beta, \alpha) &= \prod_{i=1}^{m} \mathrm{P}(Y_{i1} = y_{i1})\mathrm{P}(Y_{i2} = y_{i2}|Y_{i1} = y_{i1}) \cdots \mathrm{P}(Y_{in_i} = y_{in_i}|Y_{in-1} = y_{in-1}) \\ &= \prod_{i=1}^{m} \frac{\exp(-\lambda_{i1}){\lambda_{i1}}^{y_{i1}}}{y_{i1}!} \prod_{j=2}^{n_i} \frac{\exp(-{\lambda_{ij}}^*)({\lambda_{ij}}^*)^{y_{ij}}}{y_{ij}!} \\ &= \prod_{i=1}^{m} \exp\left(y_{i1}\ln(\lambda_{i1}) - \lambda_{i1} - \ln(y_{i1}!)\right) \prod_{j=2}^{n_i} \exp\left(y_{ij}\ln({\lambda_{ij}}^*) - {\lambda_{ij}}^* - \ln(y_{ij}!)\right). \end{aligned}$$

Taking the natural logarithm then yields the log-likelihood,

$$\ln\left(L(\beta, \alpha)\right) = \sum_{i=1}^{m} \left(y_{i1}\theta_{i1} - \exp(\theta_{i1}) - \ln(y_{i1}!)\right) + \sum_{j=2}^{n_i} \left(y_{ij}{\theta_{ij}}^* - \exp({\theta_{ij}}^*) - \ln(y_{ij}!)\right),$$

where $\theta_{i1} = \ln(\lambda_{i1}) = x'_{i1}\beta$ and ${\theta_{ij}}^* = \ln({\lambda_{ij}}^*)$.

The following constraints must be satisfied in order for the constructed likelihood to be valid: (1) $\lambda_{ij} > 0$ ($j = 1, \ldots, n_i$); (2) $-1 < \alpha < 1$ ($j = 2, \ldots, n_i$), in order to achieve a positive-definite correlation matrix; and (3) $\lambda_{ij} - \alpha\sigma_{ij}/\sigma_{ij-1}(\lambda_{ij-1}) > 0$ ($j = 2, \ldots, n_i$) (Guerra and Shults, 2014).

2.3. Likelihood Equations

To obtain maximum likelihood estimates of β and α, we need to obtain simultaneous solutions to the following estimating equations for β and α, respectively:

$$\begin{aligned} \frac{\partial \ln\left(L(\beta, \alpha)\right)}{\partial \beta} &= \sum_{i=1}^{m} \left(y_{i1} - \exp(\theta_{i1})\right) \frac{\partial \theta_{i1}}{\partial \beta} + \sum_{j=2}^{n_i} \left(y_{ij} - \exp({\theta_{ij}}^*)\right) \frac{\partial {\theta_{ij}}^*}{\partial \beta} \quad (7) \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \ln\left(L(\beta, \alpha)\right)}{\partial \alpha} &= \sum_{i=1}^{m} \left(y_{i1} - \exp(\theta_{i1})\right) \frac{\partial \theta_{i1}}{\partial \alpha} + \sum_{j=2}^{n_i} \left(y_{ij} - \exp({\theta_{ij}}^*)\right) \frac{\partial {\theta_{ij}}^*}{\partial \alpha} \quad (8) \\ &= 0. \end{aligned}$$

The derivatives are provided in Appendix A. We maximized the likelihood using an adaptive barrier algorithm as implemented in the `constrOptim` function in R (R Core Team, 2014). We applied the Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization method by Broyden (1970), Feltcher (1970), Goldfarb (1970), and Shanno (1970a, 1970b), which is implemented in `constrOptim` when the gradient is provided.

The following algorithm summarizes our estimation procedure for a particular model:

1. Choose initial estimates (starting values) of α and β. Starting values can be obtained using GEE to fit a Poisson model with an AR(1) correlation structure; however, we should check that the starting values satisfy the constraints (Section 2.2). If the estimates violate the constraints, change the starting values by choosing a value for α that is closer to zero or by applying Poisson regression, which is equivalent to assuming that $\alpha = 0$.
2. Obtain solutions to the likelihood equations 7 and 8 using the adaptive barrier algorithm that is implemented in the R package `constrOptim`. R code for the log likelihood function and for the gradient function, both of which are implemented in the Application, are provided in Appendix B.

2.4. Asymptotic Distribution of the Estimators

If the model is correctly satisfied and standard regularity conditions are satisfied, the ML approach described here will yield estimates that are consistent and asymptotically normal. Let $\theta = (\beta, \alpha)^T$ and the maximum likelihood estimators $\hat{\theta} = (\hat{\beta}, \hat{\alpha})^T$. We estimated the asymptotic covariance matrix of $\hat{\theta}$ with the observed information $(i(\hat{\theta}))^{-1}$ that we estimated using the inverse of the negative Hessian matrix, which is defined in Appendix A and implemented in Appendix B.

3. Application

3.1. Doctor visits data

Here we consider an analysis of a subset of data from the German Socio-Economic Panel data (Winkelmann, 2004) that we obtained within Stata (`http://www.stata-press.com/data/r14/drvisits`) and then exported for analysis in R (StataCorp, 2013). Here we compare the results of an analysis using the proposed ML approach with the results obtained using Poisson regression and GEE.

The goal of the analysis was to assess the impact of the 1997 health reform on the reduction of government expenditures. A sample of 1518 women who were employed full time in the year before or after the reform was implemented were evaluated. The outcome we considered was the self-reported number of doctor visits in the three months prior to the interview. The main covariate of interest was an indicator variable that took value 1 if the interview took place after the reform was implemented (or took value 0 otherwwise). Additional covariate information was available on each participant's age, education, marital status, self-reported health status, and the logarithm of household income. Of the 1518 women in the dataset, 709 were interviewed both before and after the reform was implemented; 391 were only interviewed before; and 418 were only interviewed after the reform went into effect. This resulted in a total of 2227 observations available for the analysis.

We assumed Equation 5 with the following linear predictor:

$$x_{ij} = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3ij} + \beta_4 x_{4ij} + \beta_5 x_{5ij} + \beta_6 x_{6ij},$$

where x_{ij1} was the indicator variable for health care reform (1 if after implementation; 0 if before), x_{ij2} was age in years, x_{ij3} was education in years, x_{ij4} was marital status (1 if married; 0 if not married), x_{ij5} was self-reported health status (1 if bad; 0 if not bad), and x_{ij6} was the logarithm of household income.

We first fit the above model using Poisson regression as implemented in the `glm` function in R; the results are provided in Table 1. Among women with the same household income, marital status, self-reported health, and education, there was a reduction in the log count of doctor visits of 0.140 after health care reform was implemented ($p < 0.0001$).

Next, we used the `geeglm` function in R to implement GEE with an assumed AR(1) working correlation structure; the results are shown in Table 1. As for Poisson regression, there was a significant reduction in the log count of doctor visits ($\hat{\beta}_1 = -0.123, p = 0.0202$). The estimated correlation parameter was 0.213.

When we fit the GEE model we assumed that the scalar parameters $\phi_{ij} = \phi = 1 \; \forall \; i, \, j$. After fitting GEE, we assessed the adequacy of this assumption by obtaining an estimate of ϕ based on the final GEE estimates of β:

$$\widehat{\phi} = \frac{1}{m} \sum_{i=1}^{m} \frac{Z_i(\widehat{\boldsymbol{\beta}})' Z_i(\widehat{\boldsymbol{\beta}})}{n_i},$$

where $Z_i(\widehat{\boldsymbol{\beta}})$ is the $n_i \times 1$ vector of Pearson residuals $z_{ij}(\widehat{\boldsymbol{\beta}})$ with $z_{ij}(\widehat{\boldsymbol{\beta}}) = \frac{y_{ij} - \widehat{\lambda_{ij}}}{\sqrt{\widehat{\lambda_{ij}}}}$. The estimated ϕ was $\hat{\phi} = 4.33$, which is much greater than 1 and was therefore suggestive of over-dispersion in the data.

Lastly, we fit the proposed ML approach using the algorithm for estimation described in Section 2.3. We obtained starting values for our approach using GEE, after first confirming that $\hat{\alpha}$ satisfied the necessary constraint to guarantee a valid parent distribution, which in this case was $\hat{\alpha} < 0.4494$.

Table 1 shows the results for the ML approach. The estimated correlation parameter was 0.313 with a 95% confidence interval of (0.272, 0.354). Although not customary for longitudinal data, a likelihood ratio test of the null hypothesis $\alpha = 0$ resulted in a p-value < 0.0001. After adjusting for the correlation among the counts of doctors visits, for over-dispersion, and for the other covariates, we again found that there was a significant impact of initiation of health care reform on the number of doctor visits ($\hat{\beta}_1 = -0.113, p < 0.0001$).

Overall, the parameter estimates were similar for the proposed ML approach, GEE, and the Poisson regression. While the impact of age was similar across the approaches, it was significant in both the ML and Poisson approaches but not significant in the GEE model (ML $p = 0.0005$, GEE $p = 0.1182$, and Poisson $p = 0.0008$). Similarly, the logarithm of household income was significant in both the ML and Poisson approaches but not significant in the GEE model (ML $p < 0.0001$, GEE $p = 0.0809$, and Poisson $p < 0.0001$).

With estimates of the log-likelihood for Poisson regression and the proposed ML approach, it was possible to calculate the AIC and BIC criteria as measures of the relative quality of the models for this set of data. Both BIC and AIC incorporate a penalty term for the number of parameters used in the model because it is possible to increase the numerical value of the likelihood solely by including additional parameters in the model, which may result in over-fitting the model to the data. This penalty term is larger in the BIC as compared to the AIC. For the Poisson regression model, the AIC and BIC values were 11899 and 11939, which were both greater than the AIC and BIC values for the ML approach (AIC = 11707 and BIC = 11746), which indicates that the ML approach had improved model fit over Poisson regression.

3.2. Epilepsy seizure data

Here we implement the proposed ML method and GEE for analysis of the epilepsy seizure data (Thall and Vail, 1990; Farewell and Farewell, 2013). We do not demonstrate the application of Poisson regression as we did in the previous section. However, results for Poisson regression confirmed the selection of the more general model that we obtained for the proposed ML approach.

We assumed Equation 5 with the following linear predictor:

$$x_{ij}'\beta = \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3} + \beta_4 x_{ij4}, \tag{9}$$

where x_{ij1} represents an indicator for treatment, x_{ij2} represents baseline seizure count (number of seizures in the 3 month time period prior to the start of the study), x_{ij3} represents subject age in years, and x_{ij4} represents two-week time period (coded as 1,2,3,4). We initially included a time period by treatment interaction term, but the interaction term was not significant for the proposed approach or for GEE (both p-values > 0.05); we therefore initially focused on the simpler model 9 for this demonstration.

Table 2 shows the sample mean and variance of seizure counts at baseline and the four subsequent two-week periods (denoted as Y1 through Y4) for the placebo and drug groups for the seizure counts; it also displays the sample mean and variance of age at baseline. From the table, the sample variance for the outcome variables, Y1 through Y4, were greater than their respective means, which suggested that there was over-dispersion in the seizure counts.

Table 3 shows the results of the analysis. The estimates were similar for the proposed ML method and GEE. The estimate of treatment was negative for both approaches, which suggested that the number of seizures was lower for subjects in the treatment group. However, treatment only differed significantly from 0 for the proposed ML approach ($p = 0.0127$ for ML versus $p = 0.3014$ for GEE). In addition, time period only differed significantly from 0 for the proposed ML approach ($p = 0.0031$ for ML versus $p = 0.0580$ for GEE).

The likelihood ratio test of the hypothesis that the regression parameter for time period is 0 also suggested that time period should be retained in the model for the proposed ML approach (p = 0.0030.) However, since the GEE analysis suggested that time period might not be important, we removed time period from the model for both GEE and the proposed ML approach. As shown in Table 4, treatment differed significantly from 0 for the proposed ML approach, but was not significant for GEE ($p = 0.0121$ for ML versus $p = 0.2977$ for GEE).

We next compared the AIC and BIC for the models that included and excluded time period. As shown in the Tables, both the AIC and BIC values were smaller for the larger model that included time period. The respective AIC and BIC values were 1566 and 1579 for the larger model, versus 1573 and 1583 for the smaller model. The AIC and BIC values indicated that the fit was superior for the larger model, which lent additional support for the larger model with its significant treatment and time period effects.

4. Simulation Studies

In the previous section we identified significant treatment effects for the proposed ML approach that were not observed for GEE. Since the results depended on choice of approach, it was of interest to compare the performance of the methods for finite samples. We therefore performed simulations to assess the properties of the estimators of α and β for the proposed ML approach and GEE.

4.1. Set-up

We compared the performance of the ML and GEE estimators for

$$x'_{ij}\beta = \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3}, \tag{10}$$

where the x_{ijk} were defined in the previous section.

The results shown here are based on $R = 1000$ simulation runs, equal group sizes $m/2$, $\beta = (0.4467, -0.1659, 0.0232, 0.0258)'$ (based on GEE estimates), and $n_i = 4$ measurements per subject. For this scenario, the correlation must satisfy the constraint $\alpha < 0.707$ (see Section 2.2) to ensure the existence of a valid parent distribution. We specified values of $\alpha \in \{0.2, 0.4, 0.6, 0.7\}$.

Covariates were simulated based on the observed epilepsy seizure data in the previous section. Treatment was specified as present (equal to 1) for one group and as absent (equal to 0) for the other group. Baseline seizure count was simulated from a Poisson distribution with a random seed and mean

= 31.22 based on the mean baseline age from the epilepsy data. Similarly, age was simulated from a normal distribution based on the epilepsy data for which the minimum age was 18, the mean was 28.3, and the standard deviation was 6.261. Simulated age values below 18 were discarded and the next simulated age value was assigned. Age was then rounded to a whole number, as it was recorded in the epilepsy data.

The approach proposed by Guerra and Shults (2014) was used to simulate the correlated Poisson seizure counts with specified means, over-dispersion, and AR(1) correlation structure.

4.2. Assessments

We wrote code in R to evaluate mean square error (MSE), percent bias, small sample efficiency, and 95% coverage probabilities using the observed information matrix. The mean square error (MSE) for estimator $\hat{\theta}$ is defined as

$$\frac{1}{R}\sum_{i=1}^{R}(\theta - \hat{\theta}_i)^2,$$

where θ is the true value. The percent bias for estimator $\hat{\theta}$ is defined as

$$\left\{\frac{1}{R}\sum_{i=1}^{R}(\theta - \hat{\theta}_i)/\theta\right\} * 100.$$

Lastly, to evaluate the coverage probabilities, a 95% confidence interval was computed for each parameter estimate within each simulation run. The coverage probabilities represent the proportion of the R simulation runs in which the true parameter fell within the 95% confidence bounds. GEE coverage probabilities were computed similarly using the naïve variance estimates obtained from `geeglm` in R.

4.3. Results

Table 5 displays the MSE and Table 6 displays the percent bias for the simulations. For the ML method, the MSE for $\hat{\beta}$ and $\hat{\alpha}$ and the percent bias for $\hat{\alpha}$ decreased as m increased.

As compared to GEE, the ML approach had lower MSE and percent bias for all sample sizes for $\hat{\alpha}$. For $\hat{\beta}$, the percent bias was similar for ML and GEE; however, the MSE was slightly smaller for ML than for GEE. For scenarios with high correlation ($\alpha = 0.6$ or 0.7), the intercept and treatment estimates, $\hat{\beta}_0$ and $\hat{\beta}_1$, had smaller MSE and percent bias for the proposed ML approach than for GEE, for all samples sizes.

Table 7 then displays the estimated coverage probabilities. With respect to $\widehat{\beta}$, the coverage probabilities were similar for the ML and GEE approach and were close to the nominal 95% level. With respect to $\widehat{\alpha}$, the ML approach model-based coverage probabilities were close to the nominal 95%, which outperformed the GEE approach, whose model-based coverage probabilities were below the nominal 95% level. Coverage probabilities for α were better for the ML based approach than GEE across all sample sizes and correlations ($\alpha = 0.2, 0.4, 0.6, 0.7$).

5. Discussion

We proposed an ML approach for analysis of equally spaced longitudinal count data that accounts for intra-subject correlation of measurements and over-dispersion. Our application of the ML approach and GEE demonstrated that the results of the analysis differed between approaches, with significant treatment differences observed for some models for the ML approach, but not for GEE. The availability of the AIC and BIC criteria for the ML approach was useful for selecting between nested models. The interested reader can replicate our analyses using code in R that we provided in Appendix B.

Our simulations demonstrated that the ML approach was similar to or slightly outperformed GEE with respect to MSE, bias, and coverage probabilities, especially for higher values of the correlation (for $\hat{\beta}$). That the ML approach outperformed GEE for larger values of the correlation was not surprising. We assumed over-dispersion that was induced by α and that was greater for larger values of α. For $\alpha = 0$ the assumed models for the marginal means and correlations would have been identical for the ML approach and GEE. That the differences between the two approaches were greatest for larger values for the correlation was therefore to be expected.

However, there are some limitations to the proposed ML approach that should be acknowledged. First, we assumed that the adjacent correlations on subjects are constant, which may not be plausible for data that are unequally spaced in time. In addition, although the proposed approach accounts for over-dispersion in the distribution of Y_{ij} for $j = 2, \ldots, n_i$, it assumes that Y_{i1} is distributed as Poisson. The proposed approach therefore does not account for over-dispersion in the first measurements on each subject.

6. References

Akaike, H. (1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control *19,* 719-723.

Broyden, C.G. (1970). The convergence of a class of double-rank minimization algorithms. Journal of the Institute of Mathematics and Its Applications *6,* 76-90.

Consul, P.C., and Famoye, F. (1992). Generalized Poisson regression model. Communications in Statistics - Theory and Methods *21,* 89–109.

Efron, B. (1992). Poisson overdispersion estimates based on the method of asymmetric maximum likelihood. Journal of the American Statistical Association *87,* 98–107.

Farewell, D.M., and Farewell, V.T. (2013). Dirichlet negative multinomial regression for overdispersed correlated count data. Biostatistics *14,* 395–404.

Feller, W. (1968). An introduction to probability theory and its applications, Volume I. New York: John Wiley and Sons, Inc.

Fitzmaurice, G, Davidian, M, Verbeke, G, and Molenberghs, G (2008). Longitudinal Data Analysis. CRC Press.

Fletcher, R. (1970). A New Approach to Variable Metric Algorithms. Computer Journal *13,* 317-322.

Gabriel, KR (1962). Ante-dependence analysis of an ordered set of variables. The Annals of Mathematical Statistics *33(1),* 201212.

Gardiner, J.C., Luo, Z., and Roman, L.A. (2009). Fixed effects, random effects and GEE: what are the differences?. Statistics in medicine, *28(2),* 221–239.

Goldfarb, D. (1970). A Family of Variable Metric Updates Derived by Variational Means. Mathematics of Computation *24,* 23-26.

Guerra, M.W., Shults, J., Amsterdam, J., and Ten-Have, T. (2012). The analysis of binary longitudinal data with time-dependent covariates. Statistics in medicine *31,* 931–948.

Guerra, M.W. and Shults, J. (2014). A note on the simulation of overdispersed random variables with specified marginal means and product correlations. The American Statistician *68,* 104–107.

Liang, KY and Zeger, SL (1986). Longitudinal data analysis using generalized linear models. Biometrika *73,* 1322.

Molenberghs, G. and Kenward, M.G. (2010). Semi-parametric marginal models for hierarchical data and their corresponding full models. Computational Statistics and Data Analysis *54,* 585–597.

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. `http://www.R-project.org/`.

Schwarz, G. (1978). Estimating the dimension of a model. Annals of Statistics *6,* 461-464.

Serfling, R. (2011). Asymptotic relative efficiency in estimation. In International encyclopedia of statistical science (pp. 68-72). Springer Berlin Heidelberg.

Shanno, D.F. (1970a). Conditioning of quasi-Newton methods for function minimization. *Math. Comput.* **24,** 647-656.

Shanno, D.F. and Kettler, P.C. (1970b). Optimal conditioning of quasi-Newton methods. Math. Comput. *24,* 657-664.

StataCorp LP. (2013). STATA Multilevel Mixed-Effects Reference Manual 13. College Station, TX: Stata Press.

Thall, P.F., and Vail, S.C. (1990). Some covariance models for longitudinal count data with overdispersion. Biometrics *46 (3)* , 657–671.

Winkelmann, R. (2004). Health care reform and the number of doctor visitsan econometric analysis. Journal of Applied Econometrics *19 (4),* 1099–1255.

Table 1: Estimated parameters from the ML, GEE, and Poisson models in the analysis of the doctor visits data.

ML Approach ($AIC = 11707; BIC = 11750$)

Coefficients:

Parameter	Estimate	Std.err	Wald	$Pr(>\|W\|)$
(Intercept)	-0.461	0.2811	2.69	0.1008
Reform	-0.113	0.0241	21.99	< 0.0001
Age	0.005	0.0014	12.22	0.0005
Education	-0.008	0.0064	1.54	0.2153
Marital Status	0.026	0.0294	0.75	0.3855
Health Status	1.100	0.0313	1238.28	< 0.0001
Log Income	0.150	0.0376	15.83	< 0.0001

Correlation Parameters:

Parameter	Estimate	Std.err
alpha	0.313	0.0208

GEE Approach

Coefficients:

Parameter	Estimate	Std.err	Wald	$Pr(>\|W\|)$
(Intercept)	-0.381	0.5766	0.44	0.5083
Reform	-0.123	0.0529	5.40	0.0202
Age	0.005	0.0033	2.44	0.1182
Education	-0.009	0.0118	0.61	0.4349
Marital Status	0.038	0.0698	0.30	0.5822
Health Status	1.105	0.0873	160.23	< 0.0001
Log Income	0.139	0.0798	3.05	0.0809

Correlation Parameters:

Parameter	Estimate	Std.err
alpha	0.213	0.0238

Poisson Regression ($AIC = 11899; BIC = 11942$)

Coefficients:

Parameter	Estimate	Std.err	z value	$Pr(>\|z\|)$
(Intercept)	-0.414	0.2691	-1.54	0.1242
Reform	-0.140	0.0265	-5.28	< 0.0001
Age	0.004	0.0013	3.35	0.0008
Education	-0.011	0.0060	-1.78	0.0743
Marital Status	0.041	0.0278	1.49	0.1375
Health Status	1.133	0.0303	37.40	< 0.0001
Log Income	0.149	0.0360	4.14	< 0.0001

Table 2: Mean and variance for the placebo and treatment groups.

Variable	Placebo†	Drug†	Total†
	(n=28)	(n=31)	(n=59)
Y1	9.86 (102.8)	8.58 (332.7)	8.95 (220.2)
Y2	8.29 (66.6)	8.42 (140.7)	8.36 (103.8)
Y3	8.79 (215.2)	8.13 (192.9)	8.44 (200.2)
Y4	7.96 (58.2)	6.71 (126.8)	7.31 (93.1)
Baseline	30.79 (681.2)	31.61 (782.9)	31.22 (722.5)
Age	29.00 (36.0)	27.74 (43.6)	28.34 (39.7)

† Values in the table represent the mean (variance).

Table 3: Estimated parameters from the GEE and ML approaches for analysis of the epilepsy data when Period is included in the models.

ML Approach ($AIC = 1566; BIC = 1579$)				
Coefficients:				
Parameter	Estimate	Std.err	Wald	$Pr(> \lvert W \rvert)$
(Intercept)	0.6569	0.1958	11.26	0.0008
Treatment	-0.1668	0.0667	6.26	0.0124
Baseline	0.0232	0.0007	1111.24	< 0.0001
Age	0.0238	0.0056	17.94	< 0.0001
Period	-0.0634	0.0215	8.72	0.0032
Correlation Parameters:				
Parameter	Estimate	Std.err		
alpha	0.416	0.0334		
GEE Approach				
Coefficients:				
Parameter	Estimate	Std.err	Wald	$Pr(> \lvert W \rvert)$
(Intercept)	0.5855	0.3491	2.81	0.0936
Treatment	-0.1642	0.1589	1.07	0.3014
Baseline	0.0232	0.0012	350.97	< 0.0001
Age	0.0263	0.0118	4.95	0.0261
Period	-0.0644	0.0340	3.59	0.0580
Correlation Parameters:				
Parameter	Estimate	Std.err		
alpha	0.551	0.0656		

Table 4: Estimated parameters from the GEE and ML approaches for analysis of the epilepsy data when Period is not included in the models.

ML Approach ($AIC = 1573; BIC = 1583$)				
Coefficients:				
Parameter	Estimate	Std.err	Wald	$Pr(>\|W\|)$
(Intercept)	0.5072	0.1894	7.17	0.0074
Treatment	-0.1673	0.0667	6.30	0.0121
Baseline	0.0232	0.0007	1113.57	$< .0001$
Age	0.0238	0.0056	17.99	$< .0001$
Correlation Parameters:				
Parameter	Estimate	Std.err		
alpha	0.423	0.0342		
GEE Approach				
Coefficients:				
Parameter	Estimate	Std.err	Wald	$Pr(>\|W\|)$
(Intercept)	0.4467	0.3621	1.52	0.2174
Treatment	-0.1659	0.1593	1.09	0.2977
Baseline	0.0232	0.0012	353.32	$< .0001$
Age	0.0258	0.0117	4.86	0.0275
Correlation Parameters:				
Parameter	Estimate	Std.err		
alpha	0.544	0.0639		

Table 5: Small sample efficiencies for evaluating the AR(1) correlation structure for varying values of α and sample size per group.

Mean squared error using ML

m	α	R*	$\hat{\beta}_0$	$\hat{\beta}_1^{[1]}$	$\hat{\beta}_2^{[2]}$	$\hat{\beta}_3^{[2]}$	$\hat{\alpha}^{[1]}$
60	0.2	1000	0.056	0.355	0.297	0.291	0.609
	0.4	1000	0.088	0.503	0.427	0.445	0.505
	0.6	1000	0.127	0.803	0.642	0.619	0.308
	0.7	998	0.132	0.908	0.716	0.656	0.171
120	0.2	1000	0.029	0.176	0.138	0.137	0.305
	0.4	1000	0.040	0.254	0.203	0.194	0.236
	0.6	1000	0.054	0.381	0.291	0.294	0.124
	0.7	1000	0.067	0.489	0.349	0.325	0.067
300	0.2	1000	0.010	0.071	0.057	0.054	0.111
	0.4	1000	0.016	0.101	0.084	0.078	0.080
	0.6	1000	0.025	0.153	0.121	0.118	0.047
	0.7	1000	0.029	0.174	0.144	0.140	0.023

Mean squared error using GEE

m	α	R*	$\hat{\beta}_0$	$\hat{\beta}_1^{[1]}$	$\hat{\beta}_2^{[2]}$	$\hat{\beta}_3^{[2]}$	$\hat{\alpha}^{[1]}$
60	0.2	1000	0.057	0.355	0.300	0.290	0.668
	0.4	1000	0.089	0.516	0.427	0.450	0.701
	0.6	1000	0.137	0.852	0.703	0.653	0.571
	0.7	1000	0.160	1.133	0.883	0.795	0.424
120	0.2	1000	0.029	0.176	0.139	0.138	0.340
	0.4	1000	0.040	0.260	0.204	0.198	0.334
	0.6	1000	0.062	0.415	0.327	0.325	0.240
	0.7	1000	0.083	0.595	0.435	0.402	0.178
300	0.2	1000	0.010	0.072	0.058	0.054	0.128
	0.4	1000	0.017	0.103	0.085	0.079	0.124
	0.6	1000	0.027	0.162	0.132	0.129	0.093
	0.7	1000	0.036	0.211	0.182	0.176	0.066

Note: The true correlation structure is AR(1).
There are equal sample sizes of $\frac{m}{2}$ per group and
$\beta = (\beta_0, \beta_{drug}, \beta_{baseline}, \beta_{age})' = (0.4467, -0.1659, 0.0232, 0.0258)'$;
[1]True value by a factor of 10^2; [2]True value by a factor of 10^4;

Table 6: Percent bias for evaluating the AR(1) correlation structure for varying values of α and sample size per group.

			Percent bias using ML				
m	α	R*	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\alpha}$
60	0.2	1000	2.57	0.53	-0.61	-0.53	9.41
	0.4	1000	6.33	-0.42	-1.15	-2.31	5.15
	0.6	1000	1.95	0.05	-0.95	0.27	2.65
	0.7	998	-2.21	2.71	0.72	1.21	0.69
120	0.2	1000	-0.04	0.14	0.07	0.20	5.30
	0.4	1000	2.25	-0.52	-0.57	-0.65	2.74
	0.6	1000	0.43	-0.79	0.08	-0.13	1.39
	0.7	1000	2.00	0.15	-0.01	-1.04	0.17
300	0.2	1000	0.68	-0.18	-0.57	0.22	2.83
	0.4	1000	0.85	-0.29	-0.16	-0.25	1.31
	0.6	1000	1.91	-0.38	-0.30	-0.75	0.53
	0.7	1000	1.47	-0.29	-0.14	-0.63	-0.03

			Percent bias using GEE				
m	α	R*	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\alpha}$
60	0.2	1000	2.48	0.54	-0.60	-0.49	10.94
	0.4	1000	6.26	-0.34	-1.10	-2.29	6.06
	0.6	1000	1.88	0.64	-0.90	0.45	4.86
	0.7	1000	0.60	1.87	-0.28	0.84	4.60
120	0.2	1000	-0.22	0.07	0.13	0.24	6.19
	0.4	1000	1.95	-0.51	-0.40	-0.64	2.89
	0.6	1000	0.16	-1.05	0.34	-0.21	2.18
	0.7	1000	1.74	-0.22	0.14	-0.83	2.55
300	0.2	1000	0.65	-0.23	-0.57	0.23	2.87
	0.4	1000	0.72	-0.32	-0.15	-0.18	0.98
	0.6	1000	2.03	-0.33	-0.23	-0.88	0.86
	0.7	1000	2.83	-0.20	-0.58	-0.91	1.64

Note: The true correlation structure is AR(1).

There are equal sample sizes of $\frac{m}{2}$ per group and

$\beta = (\beta_0, \beta_{drug}, \beta_{baseline}, \beta_{age})' = (0.4467, -0.1659, 0.0232, 0.0258)'$;

Table 7: Coverage probabilities for the ML and GEE approaches with the AR(1) correlation structure for varying values of α and sample size per group.

				Coverage Probability				
m	α	Method	R	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\alpha}$
60	0.2	ML	1000	94.7	95.2	95.5	95.5	93.8
		GEE	1000	94.4	95.0	94.8	95.1	91.1
	0.4	ML	1000	93.8	94.6	95.9	93.0	94.6
		GEE	1000	93.2	94.3	95.5	92.7	86.1
	0.6	ML	1000	93.8	93.9	94.3	94.0	93.4
		GEE	1000	94.1	93.6	95.1	93.1	83.2
	0.7	ML	998	95.4	95.3	95.4	95.5	92.3
		GEE	1000	95.0	94.9	94.0	95.7	84.6
120	0.2	ML	1000	94.7	95.2	95.2	94.8	92.9
		GEE	1000	94.2	95.1	94.9	94.5	91.3
	0.4	ML	1000	95.1	96.1	95.6	94.7	95.1
		GEE	1000	95.2	96.0	95.5	94.5	85.4
	0.6	ML	1000	95.9	94.5	95.3	94.9	95.5
		GEE	1000	95.5	95.5	95.5	94.9	84.5
	0.7	ML	1000	95.3	94.2	94.7	96.2	92.9
		GEE	1000	95.3	94.2	95.0	95.9	87.2
300	0.2	ML	1000	95.2	95.0	94.7	94.7	94.5
		GEE	1000	95.6	95.3	94.8	94.6	91.5
	0.4	ML	1000	93.5	95.4	94.2	93.9	96.5
		GEE	1000	93.7	96.0	94.9	94.3	86.2
	0.6	ML	1000	93.2	95.4	94.9	94.0	95.2
		GEE	1000	93.8	95.6	94.6	94.9	85.9
	0.7	ML	1000	94.5	95.1	94.1	94.4	92.4
		GEE	1000	94.8	95.9	94.6	94.8	88.0

Note: The true correlation structure is AR(1).
There are equal sample sizes of $\frac{m}{2}$ per group and
$\beta = (\beta_0, \beta_{drug}, \beta_{baseline}, \beta_{age})' = (0.4467, -0.1659, 0.0232, 0.0258)'$

Appendix A. Derivatives

The partial derivative with respect to β (Equation 7) is given by

$$\begin{aligned}
\frac{\partial}{\partial\beta}\ell(\beta,\alpha) = \sum_{i=1}^{m} & y_{i1}x_{i1} - \lambda_{i1}x_{i1} \\
& + \left(x_{i2}\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}}\frac{\sqrt{\lambda_{i2}}}{2}\left(\frac{y_{i1}}{\sqrt{\lambda_{i1}}}(x_{i2}-x_{i1}) - \sqrt{\lambda_{i1}}(x_{i2}+x_{i1}) \right) \right) \\
& \quad \times \left(y_{i2}\left(\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1}) \right)^{-1} - 1 \right) \\
& + \sum_{j=3}^{n_i} \left(x_{ij}\lambda_{ij} + \frac{\alpha\sqrt{\lambda_{ij}}}{2}\left(\frac{y_{ij-1}}{\sqrt{\lambda_{ij-1}}}(x_{ij}-x_{ij-1}) - \sqrt{\lambda_{ij-1}}(x_{ij}+x_{ij-1}) \right) \right) \\
& \quad \times \left(y_{ij}\left(\lambda_{ij} + \alpha\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1}) \right)^{-1} - 1 \right).
\end{aligned} \tag{Appendix A.1}$$

The partial derivative with respect to α (Equation 8) is given by

$$\begin{aligned}
\frac{\partial}{\partial\alpha}\ell(\beta,\alpha) = \sum_{i=1}^{m} & \left((1-\alpha^2)^{-3/2}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1}) \right) \\
& \quad \times \left(y_{i2}\left(\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1}) \right)^{-1} - 1 \right) \\
& + \sum_{j=3}^{n_i} \left(\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1}) \right) \\
& \quad \times \left(y_{ij}\left(\lambda_{ij} + \alpha\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1}) \right)^{-1} - 1 \right).
\end{aligned} \tag{Appendix A.2}$$

The elements of the matrix of second-order partial derivatives of the log likelihood, called the Hessian matrix, are given by,

$$
\begin{aligned}
\frac{\partial^2 \ell(\beta,\alpha)}{\partial\beta\partial\beta'} &= \sum_{i=1}^{m} (-x_{i1}x_{i1}\lambda_{i1}) \\
&+ \Bigg(x_{i2}x_{i2}\lambda_{i2} + \bigg(\frac{\alpha}{\sqrt{1-\alpha^2}} \frac{\sqrt{\lambda_{i2}}}{2} \frac{x_{i2}}{2} \bigg) \times \bigg(\frac{y_{i1}(x_{i2}-x_{i1})}{\sqrt{\lambda_{i1}}} - (x_{i2}+x_{i1})\sqrt{\lambda_{i1}} \bigg) \\
&\qquad - \bigg(\frac{y_{i1}(x_{i2}-x_{i1})x_{i1}}{2\sqrt{\lambda_{i1}}} + \frac{(x_{i2}+x_{i1})x_{i1}\sqrt{\lambda_{i1}}}{2} \bigg) \times \bigg(\frac{\alpha}{\sqrt{1-\alpha^2}} \frac{\sqrt{\lambda_{i2}}}{2} \bigg) \Bigg) \\
&\times \Bigg(y_{i2} \bigg(\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}} \sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}} (y_{i1}-\lambda_{i1}) \bigg)^{-1} - 1 \bigg) \Bigg) \\
&- \Bigg(y_{i2} \bigg(\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}} \sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}} (y_{i1}-\lambda_{i1}) \bigg)^{-2} \\
&\qquad \times \bigg(x_{i2}\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}} \Big(\frac{1}{2} \sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}} (x_{i2}-x_{i1})(y_{i1}-\lambda_{i1}) - x_{i1}\lambda_{i1} \sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}} \Big) \Big) \bigg) \\
&\times \Bigg(\bigg(x_{i2}\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}} \frac{\sqrt{\lambda_{i2}}}{2} \bigg(\frac{y_{i1}}{\sqrt{\lambda_{i1}}} (x_{i2}-x_{i1}) - \sqrt{\lambda_{i1}} (x_{i2}+x_{i1}) \bigg) \bigg) \Bigg) \\
&+ \sum_{j=3}^{n_i} x_{ij}x_{ij}\lambda_{ij} + \bigg(\alpha \frac{\sqrt{\lambda_{ij}}}{2} \frac{x_{ij}}{2} \bigg) \times \bigg(\frac{y_{ij-1}(x_{ij}-x_{ij-1})}{\sqrt{\lambda_{ij-1}}} - (x_{ij}+x_{ij-1})\sqrt{\lambda_{ij-1}} \bigg) \\
&- \bigg(\frac{y_{ij-1}(x_{ij}-x_{ij-1})x_{ij-1}}{2\sqrt{\lambda_{ij-1}}} + \frac{(x_{ij}+x_{ij-1})x_{ij-1}\sqrt{\lambda_{ij-1}}}{2} \bigg) \times \bigg(\alpha \frac{\sqrt{\lambda_{ij}}}{2} \bigg) \\
&\times \bigg(y_{ij} \Big(\lambda_{ij} + \alpha \sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}} (y_{ij-1}-\lambda_{ij-1}) \Big)^{-1} - 1 \Big) \bigg) \\
&- \bigg(y_{ij} \Big(\lambda_{ij} + \alpha \sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}} (y_{ij-1}-\lambda_{ij-1}) \Big)^{-2} \\
&\qquad \times \bigg(x_{ij}\lambda_{ij} + \alpha \bigg(\frac{1}{2} \sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}} (x_{ij}-x_{ij-1})(y_{ij-1}-\lambda_{ij-1}) - x_{ij-1}\lambda_{ij-1} \sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}} \bigg) \bigg) \bigg) \\
&\times \bigg(x_{ij}\lambda_{ij} + \alpha \frac{\sqrt{\lambda_{ij}}}{2} \bigg(\frac{y_{ij-1}}{\sqrt{\lambda_{ij-1}}} (x_{ij}-x_{ij-1}) - \sqrt{\lambda_{ij-1}} (x_{ij}+x_{ij-1}) \bigg) \bigg),
\end{aligned}
\tag{Appendix A.3}
$$

$$
\begin{aligned}
\frac{\partial^2 \ell(\beta,\alpha)}{\partial\beta\partial\alpha} = \sum_{i=1}^{m} &\Biggl((1-\alpha^2)^{-3/2} \left(\frac{1}{2}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(x_{i2}-x_{i1})(y_{i1}-\lambda_{i1}) - x_{i1}\lambda_{i1}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}} \right) \Biggr) \\
&\times \Biggl(y_{i2} \left(\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1}) \right)^{-1} - 1 \Biggr) \\
&- \Biggl(y_{i2} \left(\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1}) \right)^{-2} \\
&\qquad \times \Biggl(x_{i2}\lambda_{i2} + \frac{\alpha}{\sqrt{1-\alpha^2}} \left(\frac{1}{2}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(x_{i2}-x_{i1})(y_{i1}-\lambda_{i1}) - x_{i1}\lambda_{i1}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}} \right) \Biggr) \Biggr) \\
&\times \Biggl((1-\alpha^2)^{-3/2}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1}) \Biggr) \\
&+ \sum_{j=3}^{n_i} \Biggl(\left(\frac{1}{2}\sqrt{\frac{\lambda_{i2j}}{\lambda_{ij-1}}}(x_{ij}-x_{ij-1})(y_{ij-1}-\lambda_{ij-1}) - x_{ij-1}\lambda_{ij-1}\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}} \right) \Biggr) \\
&\times \Biggl(y_{ij} \left(\lambda_{ij} + \alpha\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1}) \right)^{-1} - 1 \Biggr) \\
&- \Biggl(y_{ij} \left(\lambda_{ij} + \alpha\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1}) \right)^{-2} \\
&\qquad \times \Biggl(x_{ij}\lambda_{ij} + \alpha \left(\frac{1}{2}\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(x_{ij}-x_{ij-1})(y_{ij-1}-\lambda_{ij-1}) - x_{ij-1}\lambda_{ij-1}\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}} \right) \Biggr) \Biggr) \\
&\times \Biggl(\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1}) \Biggr),
\end{aligned}
\qquad \text{(Appendix A.4)}
$$

and

$$\begin{aligned}\frac{\partial^2 \ell(\beta,\alpha)}{\partial\alpha^2} = \sum_{i=1}^{m}&\left(\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1})\frac{3\alpha}{(1-\alpha^2)^{5/2}}\right)\\
&\times\left(y_{i2}\left(\lambda_{i2}+\frac{\alpha}{\sqrt{1-\alpha^2}}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1})\right)^{-1}-1\right)\\
&-y_{i2}\left(\lambda_{i2}+\frac{\alpha}{\sqrt{1-\alpha^2}}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1})\right)^{-2}\\
&\qquad\times\left(\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1})(1-\alpha^2)^{-3/2}\right)\\
&\times\left((1-\alpha^2)^{-3/2}\sqrt{\frac{\lambda_{i2}}{\lambda_{i1}}}(y_{i1}-\lambda_{i1})\right)\\
&+\sum_{j=3}^{n_i}\left(-y_{ij}\left(\lambda_{ij}+\alpha\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1})\right)^{-2}\right.\\
&\qquad\left.\times\left(\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1})\right)\right)\\
&\times\left(\sqrt{\frac{\lambda_{ij}}{\lambda_{ij-1}}}(y_{ij-1}-\lambda_{ij-1})\right).\end{aligned}\tag{Appendix A.5}$$

Appendix B. R Code

Appendix B.1. R code for the Doctor Visits data

```
# Application of methods described in "Maximum Likelihood Based
# Analysis of Equally Spaced Longitudinal Count Data with Specified
# Marginal Means, First-order Antedependence, and Linear Conditional
# Expectations

# Data: Doctor visits data

####################################################################
# The following appendix contains additional information to        #
# reproduce the analysis in the Application Section for the        #
# doctor visits data, which is a subset of data from the German    #
# Socio-Economic Panel data (Winkelmann, 2004) that we obtained    #
# within Stata (StataCorp, 2013). A sample of 1518 women who were  #
# employed full time in the year before or in the year after the   #
# 1997 health reform to reduce government expenditures was used    #
# to assess the impact on the number of doctor visits. The outcome #
# was the self-reported number of doctor visits in the most recent #
# three months prior to the interview. Covariate include the       #
# indicator of whether the interview was before the reform or      #
# after the reform women’s age, education, marital status, self-   #
# reported health status, and the logarithm of the household       #
# income.                                                          #
#
# Here we compare the results of an analysis using the proposed    #
# ML approach with the results obtained using Poisson regression   #
# and GEE.                                                         #
####################################################################


####################################################################
#                                                                  #
# Table of Contents
# 1. Supporting functions (run these FIRST)
# 2. Functions for ML (run these SECOND)
# 2.1 Log likelihood
# 2.2 Gradient
# 3. Load the data
# 4. Poisson approach
# 5. GEE approach
# 6. ML approach
#                                                                  #
####################################################################
```

```
####################################################################
#                                                                  #
######################### Options ##################################
#                                                                  #
####################################################################

# Clear workspace
ls()
rm(list=ls())
ls()

# Optional code for more decimal places
options(digits=10)

####################################################################
#                                                                  #
######### 1. Supporting functions for ML Approach ##################
#                                                                  #
####################################################################

# Getting Info from the Data
# This function will remain all the subjects
# This function will not help order the subjects
# This function was written by Matt Guerra

cluster.size<- function(id){
  clid<- unique(id)
  m<- length(unique(id))
  n<- rep(0,m)
  autotime<- rep(0,0)
  for(i in 1:m){
    n[i]<- length(which(id==clid[i]))
    autotime<- c(autotime,1:n[i])
  }
  id<- rep(1:m,n)
  return(list(m=m,n=n,id=id,autotime=autotime))
}


# Data Process: This function will delete subjects with less or
# equal to #=del.n observations
# This function was written by Matt Guerra
```

```
data.proc<- function(data,formula,time=NULL,id,del.n){

  dat<- data.frame(data)
  col.name<- names(dat)

  cluster<- cluster.size(id)
  m<- cluster$m
  n<- cluster$n
  id<- cluster$id
  if(length(time)==0){
    time<- cluster$autotime
  }
  autotime<- cluster$autotime
  index<- order(id,time)
  if(ncol(dat)==1){
    dat<- dat[index,]
  }else{
    dat<- dat[index,]
  }
  dat<- data.frame(dat)
  names(dat)<- col.name


  del<- which(n<=del.n)
  if(length(del)>0){
    n<- n[-del]
    m<- length(n)
    mtch<- match(id,del)
    del.id<- which(mtch!="NA")
    dat<- dat[-del.id,]
    dat<- data.frame(dat)
    names(dat)<- col.name
    row.names(dat)<- 1:nrow(dat)
    time<- time[-del.id]
    autotime<- autotime[-del.id]
    id<- rep(1:m,n)
  }

  formula<- as.formula(formula)
  fml<- as.formula(paste("~",formula[3],"+",formula[2],sep=""))
  dat<- model.matrix(fml,data=dat)

  return(list(data=dat,time=time,autotime=autotime,id=id,m=m,n=n))
}
```

```
####################################################################
#                                                                  #
##################### 2. Functions for ML ##########################
#                                                                  #
####################################################################



####################################################################
#                                                                  #
##################### 2.1 Log likelihood ###########################
#                                                                  #
####################################################################

# Log Likelihood function
# This function was written by Victoria Gamerman

drv.logl <- function(start.values){
 alpha <- start.values[1]
 beta <- start.values[2:length(start.values)]

#to be updated by user:
 formula <- numvisit ~ reform + age + educ + married + badh + loginc
 id <- drvisits$id
 time <- drvisits$visit
 d <- dim(drvisits)
 k <- length(all.vars(formula))-1
 dt.fm<- data.frame(drvisits)

dataset<- data.proc(data=dt.fm,formula=formula,time=time,id=id,del.n=0)
  m<- dataset$m
  n<- dataset$n
  id<- dataset$id
  time<- dataset$time

l_beta_a <- 0
l_beta_b <- 0
l_beta_c <- 0
for (i in 1:m){
        data_i <- matrix(NA, nrow=n[i], ncol=dim(dataset$data)[2])
        data_i[1:n[i],1:dim(dataset$data)[2]] <- dataset$data[which(id==i),]
        data.end<- ncol(data_i)
        x_i <- matrix(NA, nrow=n[i], ncol=k+1)
        x_i[1:n[i],1:(k+1)] <- data_i[,-data.end]
        y_i<- data_i[,data.end]
        n_i <- nrow(data_i)

for (j in 1:n_i){
   if (j == 1){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      l_beta_a <- l_beta_a + y_i[j]*log(lam_ij) - exp(log(lam_ij))
          - log(factorial(y_i[j]))
   }
   if (j == 2){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
      lam_ij_1 <- lam_ij_1[1]
      lamdot_i2 <- lam_ij + (alpha / sqrt(1-alpha^2))*sqrt(lam_ij
         / lam_ij_1)*(y_i[j-1] - lam_ij_1)
      #constraint
      constr <- sqrt(lam_ij / (lam_ij_1 + lam_ij))
         #print(constr)
         if(is.finite(constr) == FALSE){ constr <- 0.2}
         if(is.finite(lamdot_i2) == FALSE){ lamdot_i2 <- 0.5*constr}
         if(lamdot_i2 < 0){lamdot_i2 <- 0.5*constr}
      l_beta_b <- l_beta_b + y_i[j]*log(lamdot_i2)
         - exp(log(lamdot_i2)) - log(factorial(y_i[j]))
   }
   if (j > 2){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
      lam_ij_1 <- lam_ij_1[1]
      lamdot_ij <- lam_ij + alpha *sqrt(lam_ij / lam_ij_1)
         *(y_i[j-1] - lam_ij_1)
      constr <- sqrt(lam_ij / lam_ij_1)
         #print(constr)
         if(is.finite(constr) == FALSE){ constr <- 0.2}
         if(is.finite(lamdot_ij) == FALSE){ lamdot_ij <- 0.5*constr}
         if(lamdot_ij < 0){lamdot_ij <- 0.5*constr}
      l_beta_c <- l_beta_c + y_i[j]*log(lamdot_ij)
         - exp(log(lamdot_ij)) - log(factorial(y_i[j]))
   }
}
}
loglik <- l_beta_a + l_beta_b + l_beta_c
```

```
return(loglik)
}


####################################################################
#                                                                  #
######################## 2.2 Gradient ##############################
#                                                                  #
####################################################################

# Gradient function: It should take arguments matching those of
# f and return a vector containing the gradient.
# This function was written by Victoria Gamerman

drv.grad <- function(start.values){
 alpha <- start.values[1]
 beta <- start.values[2:length(start.values)]

#to be updated by user:
data<-drvisits
 formula <- numvisit ~ reform + age + educ + married + badh + loginc
 id <- drvisits$id
 time <- drvisits$visit
 d <- dim(drvisits)
 k <- length(all.vars(formula))-1
 dt.fm<- data.frame(drvisits)

dataset<- data.proc(data=dt.fm,formula=formula,time=time,id=id,del.n=0)
  m<- dataset$m
  n<- dataset$n
  id<- dataset$id
  time<- dataset$time
  autotime<- dataset$autotime

l_beta_a <- matrix(0,nrow=k+1, ncol=1)
l_beta_b <- matrix(0,nrow=k+1, ncol=1)
l_beta_c <- matrix(0,nrow=k+1, ncol=1)
l_alpha_a <- matrix(0,nrow=1, ncol=1)
l_alpha_b <- matrix(0,nrow=1, ncol=1)

for (i in 1:m){
   data_i <- matrix(NA, nrow=n[i], ncol=dim(dataset$data)[2])
   data_i[1:n[i],1:dim(dataset$data)[2]] <- dataset$data[which(id==i),]
   data.end<- ncol(data_i)
   x_i <- matrix(NA, nrow=n[i], ncol=k+1)
   x_i[1:n[i],1:(k+1)] <- data_i[,-data.end]
   y_i<- data_i[,data.end]
   n_i <- nrow(data_i)

if (n_i>=1){
   for (j in 1:n_i){
      if (j == 1){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         l_beta_a <- l_beta_a + y_i[j]*x_i[j,]-x_i[j,]*lam_ij
      }
      if(j==2){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
         lam_ij_1 <- lam_ij_1[1]
         l_alpha_a <- l_alpha_a + y_i[j]*(lam_ij
            + (alpha/sqrt(1-alpha^2))*sqrt(lam_ij/lam_ij_1)*(y_i[j-1]
               -lam_ij_1))^(-1)*
            (sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1)*((1-alpha^2)^(-3/2)))
            -(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1)*((1-alpha^2)^(-3/2)))
         l_beta_b <- l_beta_b + y_i[j]*(lam_ij+(alpha/sqrt(1-alpha^2))
            *sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))^(-1)*
            (x_i[j,]*lam_ij+(alpha/sqrt(1-alpha^2))*(0.5*sqrt(lam_ij/lam_ij_1)
            *(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)-x_i[j-1,]*lam_ij_1
            *sqrt(lam_ij/lam_ij_1)))-
            (x_i[j,]*lam_ij+(alpha/sqrt(1-alpha^2))*(0.5*sqrt(lam_ij/lam_ij_1)
            *(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)-x_i[j-1,]*lam_ij_1
            *sqrt(lam_ij/lam_ij_1)))
      }
      if(j>2){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
         lam_ij_1 <- lam_ij_1[1]
         l_alpha_b <- l_alpha_b + y_i[j]*(lam_ij + alpha*sqrt(lam_ij/lam_ij_1)
            *(y_i[j-1]-lam_ij_1))^(-1)*(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))
            -(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))
         l_beta_c <- l_beta_c + y_i[j]*(lam_ij + alpha*sqrt(lam_ij/lam_ij_1)
            *(y_i[j-1]-lam_ij_1))^(-1)*(x_i[j,]*lam_ij+alpha
            *(0.5*sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)
            -x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))-(x_i[j,]*lam_ij+alpha
            *(0.5*sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)
            -x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))
      }
   }
}
}
l_alpha <- l_alpha_a+l_alpha_b
l_beta <- l_beta_a+l_beta_b+l_beta_c
out<-t(t(c(l_alpha,l_beta)))

return(out)
}




####################################################################
#                                                                  #
###################### 3. Load the data ############################
#                                                                  #
####################################################################

# Data Dictionary
# id = person id
# numvisit = number of dr visits in last 3 mo before interview
# age = age in years
# educ = education in years
# married = 1 if married, 0 else
# badh = self-reported health status, 1 if bad, 0 else
# loginc = log of household income
# reform = 0 if interview before reform, 1 if after reform

# outcome variable = numvisit
# covariates = reform age educ married badh loginc
# identification variable = id
# timing variable = visit

# Download the data from StataCorp:
# http://www.stata-press.com/data/r14/drvisits.dta

# Library to read Stata files
# Install Package: readstata13
library(readstata13)
drvisits <- read.dta13("C:/Users/Victoria/Downloads/drvisits.dta",
   convert.factors = TRUE, generate.factors = FALSE,
   encoding = NULL, fromEncoding = NULL, convert.underscore = FALSE,
   missing.type = FALSE, convert.dates = TRUE, replace.strl = FALSE,
   add.rownames = FALSE)
```

```
names(drvisits)
attach(drvisits)

####################################################################
#                                                                  #
#################### 4. Poisson approach ###########################
#                                                                  #
####################################################################

drv_poi <- glm(numvisit ~ reform + age + educ + married + badh
   + loginc, data=drvisits, family = poisson)
summary(drv_poi)

# Use AIC to compute the -2logl to be used in BIC calculation
poi.m2logl <- drv_poi$aic - 2*(length(drv_poi$coefficients)+1)
poi.BIC <- log(length(unique(id)))*(length(drv_poi$coefficients)+1)
   +poi.m2logl
poi.BIC

####################################################################
#                                                                  #
######################## 5. GEE approach ###########################
#                                                                  #
####################################################################

library(geepack)
drv_gee <- geeglm(numvisit ~ reform + age + educ + married + badh
   + loginc, data=drvisits, id = id,
   family = poisson(link = "log"), corstr = "ar1")
summary(drv_gee)


####################################################################
#                                                                  #
########################### 6. ML approach #########################
#                                                                  #
####################################################################

# Assign starting values from GEE
beta.start <- drv_gee$geese$beta
alpha.start <- drv_gee$geese$alpha
start.values <- t(t(c(alpha.start,beta.start)))

####################################################################
```

```
# Enforce the constraint: -1 < alpha < 1
# vector is defined as (alpha, beta parameters)
ui <- rbind(c(1,0,0,0,0,0,0,0), c(-1,0,0,0,0,0,0,0))
ci <- c(-1,-1)

####################################################################

# Call the constrOptim function
full.ml <- constrOptim(start.values, drv.logl, grad=drv.grad,
   ui = ui, ci = ci, mu = 1e-04, control=list("fnscale"=-1),
   outer.iterations = 100, outer.eps = 1e-05, hessian = TRUE)

####################################################################

# Organize the output
mle.beta <- full.ml$par[2:8]
mle.alpha <- full.ml$par[1]
mle.full <- full.ml$value #log likelihood
mle.cov <- solve(-full.ml$hessian) #covariance matrix

####################################################################

# Compute the AIC and BIC
AIC <- 2*(length(mle.beta)+1)-2*(mle.full)
BIC <- log(length(unique(id)))*(length(mle.beta)+1)-2*(mle.full)

####################################################################

# Observed information
#ob = observed information = 1/i(hat(theta))
std.err <- "ob"
if (std.err=="ob"){
   mle_cov <- mle.cov
}

####################################################################

# Hypothesis testing
formula <- numvisit ~ reform + age + educ + married + badh + loginc
pp <- length(all.vars(formula))

Stderr <- matrix(NA, nrow=pp, ncol=1)
Wald <- matrix(NA, nrow=pp, ncol=1)
pval <- matrix(NA, nrow=pp, ncol=1)
for (p in 1:pp){
   Stderr[p,] <- sqrt(mle_cov[(p+1),(p+1)])
   Wald[p,] <- (mle.beta[p]/sqrt(mle_cov[(p+1),(p+1)]))^2
   pval[p,] <- 1-pchisq(Wald[p,1] , df=1, lower.tail = TRUE,
      log.p = FALSE)
}
results <- cbind(mle.beta,Stderr, Wald, pval)
alpha_results <- cbind(mle.alpha,sqrt(mle_cov[1,1]))
fit_stats <- rbind(mle.full,AIC,BIC)

#format output
rownames(fit_stats) <- c("Log-Likelihood:", "AIC:", "BIC:")
colnames(fit_stats) <- c("")
colnames(results) <- c("Estimate", "Std.err", "Wald", "Pr(>|W|)")
rownames(results) <- c("(Intercept)", "reform", "age", "educ",
   "married", "badh", "loginc")
colnames(alpha_results) <- c("Estimate", "Std.err")
rownames(alpha_results) <- c("alpha")

####################################################################

# Print the output in a nice format
print(fit_stats)
cat("\n Coefficients:\n")
print(results)
cat("\n Estimated Correlation Parameters: \n")
print(alpha_results)

####################################################################
########################### THE END ################################
####################################################################
```

Appendix B.2. R code for the Epilepsy Seizure data

```
# Application of methods described in "Maximum Likelihood Based
# Analysis of Equally Spaced Longitudinal Count Data with Specified
# Marginal Means, First-order Antedependence, and Linear Conditional
# Expectations

# Data:  Epilepsy seizure data

####################################################################
# The following appendix contains additional information to        #
# reproduce the analysis in the Application Section for the        #
# epilepsy data. Thall and Vail (1990) present data from a         #
# randomized, placebo-controlled study on 59 epileptic patients    #
# with seizure counts measured every 2 weeks over an 8 week period.#
# Patients were randomized to drug treatment or placebo alongside  #
# standard chemotherapy treatment and measured the outcome as the  #
# count of the number of seizures. Additional covariates include   #
# information on patient treatment (placebo or drug), baseline     #
# seizure counts, and age in years. Of the 59 patients, 28 were    #
# randomized to placebo and 31 were randomized to drug treatment.  #
####################################################################

####################################################################
# Table of Contents
# 1. Supporting functions (run these FIRST)
# 2. Functions for ML with period (run these SECOND)
#   2.1 Log likelihood
#   2.2 Gradient
# 3. Functions for ML without period (run these THIRD)
#   3.1 Log likelihood
#   3.2 Gradient
# 4. Load the data
# 5. GEE approach
#   5.1 GEE with period
#   5.2 GEE without period
# 6. ML approach
#   6.1 ML with period
#   6.2 ML without period
#   6.3 Likelihood ratio (LR) test
####################################################################

####################################################################
#                                                                  #
######################### Options ##################################
```

```
#                                                                  #
####################################################################

# Clear the workspace
rm(list = ls())

# Optional code for more decimal places
options(digits=10)

####################################################################
#                                                                  #
######### 1. Supporting functions for ML Approach ##################
#                                                                  #
####################################################################

# Getting Info from the Data
# This function will remain all the subjects
# This function will not help order the subjects
# This function was written by Matt Guerra


cluster.size<- function(id){
  clid<- unique(id)
  m<- length(unique(id))
  n<- rep(0,m)
  autotime<- rep(0,0)
  for(i in 1:m){
    n[i]<- length(which(id==clid[i]))
    autotime<- c(autotime,1:n[i])
  }
  id<- rep(1:m,n)
  return(list(m=m,n=n,id=id,autotime=autotime))
}


# Data Process
# This function will delete subjects with less or equal to #=del.n
#   observations.
# This function was written by Matt Guerra

data.proc<- function(data,formula,time=NULL,id,del.n){

  dat<- data.frame(data)
  col.name<- names(dat)
  cluster<- cluster.size(id)
  m<- cluster$m
  n<- cluster$n
  id<- cluster$id
  if(length(time)==0){
    time<- cluster$autotime
  }
  autotime<- cluster$autotime
  index<- order(id,time)
  if(ncol(dat)==1){
    dat<- dat[index,]
  }else{
    dat<- dat[index,]
  }
  dat<- data.frame(dat)
  names(dat)<- col.name


  del<- which(n<=del.n)
  if(length(del)>0){
    n<- n[-del]
    m<- length(n)
    mtch<- match(id,del)
    del.id<- which(mtch!="NA")
    dat<- dat[-del.id,]
    dat<- data.frame(dat)
    names(dat)<- col.name
    row.names(dat)<- 1:nrow(dat)
    time<- time[-del.id]
    autotime<- autotime[-del.id]
    id<- rep(1:m,n)
  }

  formula<- as.formula(formula)
  fml<- as.formula(paste("~",formula[3],"+",formula[2],sep=""))
  dat<- model.matrix(fml,data=dat)

  return(list(data=dat,time=time,autotime=autotime,id=id,m=m,n=n))
}

#####################################################################
#                                                                   #
##################### 2. Functions for ML ###########################
#                          WITH PERIOD                              #
#####################################################################
```

```
####################################################################
#                                                                  #
##################### 2.1 Log likelihood ###########################
#                                                                  #
####################################################################

# Log Likelihood function
# This function was written by Victoria Gamerman

ml.logl1 <- function(start.values){
 alpha <- start.values[1]
 beta <- start.values[2:length(start.values)]
 formula <- y ~ trt + base + age + period
 id <- epil$subject
 time <- epil$period
 d <- dim(epil)
 k <- length(all.vars(formula))-1
 dt.fm<- data.frame(epil)

dataset<- data.proc(data=dt.fm,formula=formula,time=time,id=id,del.n=0)
  m<- dataset$m
  n<- dataset$n
  id<- dataset$id
  time<- dataset$time

l_beta_a <- 0
l_beta_b <- 0
l_beta_c <- 0
for (i in 1:m){
        data_i <- matrix(NA, nrow=n[i], ncol=dim(dataset$data)[2])
        data_i[1:n[i],1:dim(dataset$data)[2]] <- dataset$data[which(id==i),]
        data.end<- ncol(data_i)
        x_i <- matrix(NA, nrow=n[i], ncol=k+1)
        x_i[1:n[i],1:(k+1)] <- data_i[,-data.end]
        y_i<- data_i[,data.end]
        n_i <- nrow(data_i)

for (j in 1:n_i){
   if (j == 1){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      l_beta_a <- l_beta_a + y_i[j]*log(lam_ij) - exp(log(lam_ij))
         - log(factorial(y_i[j]))
   }
   if (j == 2){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
      lam_ij_1 <- lam_ij_1[1]
      lamdot_i2 <- lam_ij + (alpha / sqrt(1-alpha^2))
         *sqrt(lam_ij / lam_ij_1)*(y_i[j-1] - lam_ij_1)
      #constraint
         constr <- sqrt(lam_ij / (lam_ij_1 + lam_ij))
         #print(constr)
         if(is.finite(constr) == FALSE){ constr <- 0.2}
         if(is.finite(lamdot_i2) == FALSE){ lamdot_i2 <- 0.5*constr}
         if(lamdot_i2 < 0){lamdot_i2 <- 0.5*constr}
      l_beta_b <- l_beta_b + y_i[j]*log(lamdot_i2) - exp(log(lamdot_i2))
         - log(factorial(y_i[j]))
   }
   if (j > 2){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
      lam_ij_1 <- lam_ij_1[1]
      lamdot_ij <- lam_ij + alpha *sqrt(lam_ij / lam_ij_1)*(y_i[j-1]
         - lam_ij_1)
      #constraint
         constr <- sqrt(lam_ij / lam_ij_1)
         #print(constr)
         if(is.finite(constr) == FALSE){ constr <- 0.2}
         if(is.finite(lamdot_ij) == FALSE){ lamdot_ij <- 0.5*constr}
         if(lamdot_ij < 0){lamdot_ij <- 0.5*constr}
      l_beta_c <- l_beta_c + y_i[j]*log(lamdot_ij) - exp(log(lamdot_ij))
         - log(factorial(y_i[j]))
   }
}
}
loglik <- l_beta_a + l_beta_b + l_beta_c
return(loglik)
}


####################################################################
#                                                                  #
####################### 2.2 Gradient ###############################
#                                                                  #
####################################################################
```

```
# Gradient function: It should take arguments matching those of
# f and return a vector containing the gradient.
# This function was written by Victoria Gamerman

ml.grad1 <- function(start.values){
 alpha <- start.values[1]
 beta <- start.values[2:length(start.values)]

 formula <- y ~ trt + base + age + period
 id <- epil$subject
 time <- epil$period
 d <- dim(epil)
 k <- length(all.vars(formula))-1
 dt.fm<- data.frame(epil)

dataset<- data.proc(data=dt.fm,formula=formula,time=time,id=id,del.n=0)
  m<- dataset$m
  n<- dataset$n
  id<- dataset$id
  time<- dataset$time
  autotime<- dataset$autotime

l_beta_a <- matrix(0,nrow=k+1, ncol=1)
l_beta_b <- matrix(0,nrow=k+1, ncol=1)
l_beta_c <- matrix(0,nrow=k+1, ncol=1)
l_alpha_a <- matrix(0,nrow=1, ncol=1)
l_alpha_b <- matrix(0,nrow=1, ncol=1)

for (i in 1:m){
   data_i <- matrix(NA, nrow=n[i], ncol=dim(dataset$data)[2])
   data_i[1:n[i],1:dim(dataset$data)[2]] <- dataset$data[which(id==i),]
   data.end<- ncol(data_i)
   x_i <- matrix(NA, nrow=n[i], ncol=k+1)
   x_i[1:n[i],1:(k+1)] <- data_i[,-data.end]
   y_i<- data_i[,data.end]
   n_i <- nrow(data_i)

if (n_i>=1){
   for (j in 1:n_i){
      if (j == 1){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         l_beta_a <- l_beta_a + y_i[j]*x_i[j,]-x_i[j,]*lam_ij
      }
      if(j==2){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
         lam_ij_1 <- lam_ij_1[1]
         l_alpha_a <- l_alpha_a + y_i[j]*(lam_ij + (alpha/sqrt(1-alpha^2))
            *sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))^(-1)*
            (sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1)*((1-alpha^2)^(-3/2)))
            -(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1)*((1-alpha^2)^(-3/2)))
         l_beta_b <- l_beta_b + y_i[j]*(lam_ij+(alpha/sqrt(1-alpha^2))
            *sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))^(-1)*
            (x_i[j,]*lam_ij+(alpha/sqrt(1-alpha^2))*(0.5*sqrt(lam_ij/lam_ij_1)
            *(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)-x_i[j-1,]*lam_ij_1
            *sqrt(lam_ij/lam_ij_1)))-(x_i[j,]*lam_ij+(alpha/sqrt(1-alpha^2))
            *(0.5*sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)
            -x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))
      }
      if(j>2){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
         lam_ij_1 <- lam_ij_1[1]
         l_alpha_b <- l_alpha_b + y_i[j]*(lam_ij + alpha*sqrt(lam_ij/lam_ij_1)
            *(y_i[j-1]-lam_ij_1))^(-1)*(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))
            -(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))
         l_beta_c <- l_beta_c + y_i[j]*(lam_ij + alpha*sqrt(lam_ij/lam_ij_1)
            *(y_i[j-1]-lam_ij_1))^(-1)*(x_i[j,]*lam_ij+alpha*(0.5
            *sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)
            -x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))-(x_i[j,]*lam_ij+alpha
            *(0.5*sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)
            -x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))
      }
   }
}
}
l_alpha <- l_alpha_a+l_alpha_b
l_beta <- l_beta_a+l_beta_b+l_beta_c
out<-t(t(c(l_alpha,l_beta)))

return(out)
}

####################################################################
#                                                                  #
##################### 3. Functions for ML ##########################
```

```
#                                                                  #
####################################################################

####################################################################
#                                                                  #
##################### 3.1 Log likelihood ###########################
#                                                                  #
####################################################################

# Log Likelihood function
# This function was written by Victoria Gamerman

ml.logl2 <- function(start.values){
 alpha <- start.values[1]
 beta <- start.values[2:length(start.values)]
 formula <- y ~ trt + base + age
 id <- epil$subject
 time <- epil$period
 d <- dim(epil)
 k <- length(all.vars(formula))-1
 dt.fm<- data.frame(epil)

dataset<- data.proc(data=dt.fm,formula=formula,time=time,id=id,del.n=0)
  m<- dataset$m
  n<- dataset$n
  id<- dataset$id
  time<- dataset$time


l_beta_a <- 0
l_beta_b <- 0
l_beta_c <- 0
for (i in 1:m){
        data_i <- matrix(NA, nrow=n[i], ncol=dim(dataset$data)[2])
        data_i[1:n[i],1:dim(dataset$data)[2]] <- dataset$data[which(id==i),]
        data.end<- ncol(data_i)
        x_i <- matrix(NA, nrow=n[i], ncol=k+1)
        x_i[1:n[i],1:(k+1)] <- data_i[,-data.end]
        y_i<- data_i[,data.end]
        n_i <- nrow(data_i)

for (j in 1:n_i){
   if (j == 1){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      l_beta_a <- l_beta_a + y_i[j]*log(lam_ij) - exp(log(lam_ij))
         - log(factorial(y_i[j]))
   }
   if (j == 2){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
      lam_ij_1 <- lam_ij_1[1]
      lamdot_i2 <- lam_ij + (alpha / sqrt(1-alpha^2))
         *sqrt(lam_ij / lam_ij_1)*(y_i[j-1] - lam_ij_1)
      #constraint
         constr <- sqrt(lam_ij / (lam_ij_1 + lam_ij))
         #print(constr)
         if(is.finite(constr) == FALSE){ constr <- 0.2}
         if(is.finite(lamdot_i2) == FALSE){ lamdot_i2 <- 0.5*constr}
         if(lamdot_i2 < 0){lamdot_i2 <- 0.5*constr}
      l_beta_b <- l_beta_b + y_i[j]*log(lamdot_i2) - exp(log(lamdot_i2))
         - log(factorial(y_i[j]))
   }
   if (j > 2){
      lam_ij <- exp(t(beta)%*%x_i[j,])
      lam_ij <- lam_ij[1]
      lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
      lam_ij_1 <- lam_ij_1[1]
      lamdot_ij <- lam_ij + alpha *sqrt(lam_ij / lam_ij_1)*(y_i[j-1]
         - lam_ij_1)
      #constraint
         constr <- sqrt(lam_ij / lam_ij_1)
         #print(constr)
         if(is.finite(constr) == FALSE){ constr <- 0.2}
         if(is.finite(lamdot_ij) == FALSE){ lamdot_ij <- 0.5*constr}
         if(lamdot_ij < 0){lamdot_ij <- 0.5*constr}
      l_beta_c <- l_beta_c + y_i[j]*log(lamdot_ij) - exp(log(lamdot_ij))
         - log(factorial(y_i[j]))
   }
}
}
loglik <- l_beta_a + l_beta_b + l_beta_c
return(loglik)
}

####################################################################
#                                                                  #
######################## 3.2 Gradient ##############################
#                                                                  #
```

```
####################################################################

# Gradient function: It should take arguments matching those of
# f and return a vector containing the gradient.
# This function was written by Victoria Gamerman

ml.grad2 <- function(start.values){
 alpha <- start.values[1]
 beta <- start.values[2:length(start.values)]

 formula <- y ~ trt + base + age
 id <- epil$subject
 time <- epil$period
 d <- dim(epil)
 k <- length(all.vars(formula))-1
 dt.fm<- data.frame(epil)

dataset<- data.proc(data=dt.fm,formula=formula,time=time,id=id,del.n=0)
  m<- dataset$m
  n<- dataset$n
  id<- dataset$id
  time<- dataset$time
  autotime<- dataset$autotime

l_beta_a <- matrix(0,nrow=k+1, ncol=1)
l_beta_b <- matrix(0,nrow=k+1, ncol=1)
l_beta_c <- matrix(0,nrow=k+1, ncol=1)
l_alpha_a <- matrix(0,nrow=1, ncol=1)
l_alpha_b <- matrix(0,nrow=1, ncol=1)

for (i in 1:m){
   data_i <- matrix(NA, nrow=n[i], ncol=dim(dataset$data)[2])
   data_i[1:n[i],1:dim(dataset$data)[2]] <- dataset$data[which(id==i),]
   data.end<- ncol(data_i)
   x_i <- matrix(NA, nrow=n[i], ncol=k+1)
   x_i[1:n[i],1:(k+1)] <- data_i[,-data.end]
   y_i<- data_i[,data.end]
   n_i <- nrow(data_i)

if (n_i>=1){
   for (j in 1:n_i){
      if (j == 1){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         l_beta_a <- l_beta_a + y_i[j]*x_i[j,]-x_i[j,]*lam_ij
      }
      if(j==2){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
         lam_ij_1 <- lam_ij_1[1]
         l_alpha_a <- l_alpha_a + y_i[j]*(lam_ij + (alpha/sqrt(1-alpha^2))
            *sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))^(-1)*
            (sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1)*((1-alpha^2)^(-3/2)))
            -(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1)*((1-alpha^2)^(-3/2)))
         l_beta_b <- l_beta_b + y_i[j]*(lam_ij+(alpha/sqrt(1-alpha^2))
            *sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))^(-1)*
            (x_i[j,]*lam_ij+(alpha/sqrt(1-alpha^2))*(0.5*sqrt(lam_ij/lam_ij_1)
            *(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)-x_i[j-1,]*lam_ij_1
            *sqrt(lam_ij/lam_ij_1)))-(x_i[j,]*lam_ij+(alpha/sqrt(1-alpha^2))
            *(0.5*sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]
            -lam_ij_1)-x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))
      }
      if(j>2){
         lam_ij <- exp(t(beta)%*%x_i[j,])
         lam_ij <- lam_ij[1]
         lam_ij_1 <- exp(t(beta)%*%x_i[j-1,])
         lam_ij_1 <- lam_ij_1[1]
         l_alpha_b <- l_alpha_b + y_i[j]*(lam_ij + alpha*sqrt(lam_ij/lam_ij_1)
            *(y_i[j-1]-lam_ij_1))^(-1)*(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))
            -(sqrt(lam_ij/lam_ij_1)*(y_i[j-1]-lam_ij_1))
         l_beta_c <- l_beta_c + y_i[j]*(lam_ij + alpha*sqrt(lam_ij/lam_ij_1)
            *(y_i[j-1]-lam_ij_1))^(-1)*(x_i[j,]*lam_ij+alpha*(0.5
            *sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)
            -x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))-(x_i[j,]*lam_ij+alpha
            *(0.5*sqrt(lam_ij/lam_ij_1)*(x_i[j,]-x_i[j-1,])*(y_i[j-1]-lam_ij_1)
            -x_i[j-1,]*lam_ij_1*sqrt(lam_ij/lam_ij_1)))
      }
   }
}
}
l_alpha <- l_alpha_a+l_alpha_b
l_beta <- l_beta_a+l_beta_b+l_beta_c
out<-t(t(c(l_alpha,l_beta)))

return(out)
}

####################################################################
#                                                                  #
```

```
###################### 4. Load the data ############################
#                                                                  #
####################################################################

# Data Dictionary
# subject = subject id
# y = count of the number of seizures
# trt = treatment placebo or drug
# base = baseline seizure counts
# age = subject age in years
# period = time variable

# outcome variable = y
# covariates = trt base age (period, where applicable)
# identification variable = subject
# timing variable = period

# Load the long data frame

library(MASS)
data(epil)
attach(epil)
names(epil)
summary(epil)

####################################################################
#                                                                  #
######################## 5. GEE approach ###########################
#                                                                  #
####################################################################

# Load the library
library(geepack)

####################################################################
#                                                                  #
#################### 5.1 GEE with Period ###########################
#                                                                  #
####################################################################

# Estimated parameters from the GEE approach for analysis of the
# epilepsy data when period is included in the models

epil_gee1 <- geeglm(y ~ trt + base + age + period, data=epil,
   id = epil$subject, family = poisson(link = "log"),
   corstr = "ar1")
summary(epil_gee1)


####################################################################
#                                                                  #
#################### 5.2 GEE without Period ########################
#                                                                  #
####################################################################

# Estimated parameters from the GEE approach for analysis of the
# epilepsy data when period is not included in the models

epil_gee2 <- geeglm(y ~ trt + base + age, data=epil,
   id = epil$subject, family = poisson(link = "log"),
   corstr = "ar1")
summary(epil_gee2)

####################################################################
#                                                                  #
############################ 6. ML approach ########################
#                                                                  #
####################################################################


####################################################################
#                                                                  #
######################## 6.1 ML with Period ########################
#                                                                  #
####################################################################

# Estimated parameters from the ML approach for analysis of the
# epilepsy data when period is included in the models

# Assign starting values from GEE
beta.start1 <- epil_gee1$geese$beta
alpha.start1 <- epil_gee1$geese$alpha
start.values1 <- t(t(c(alpha.start1,beta.start1)))

####################################################################

# Enforce the constraint: -1 < alpha < 1
# vector is defined as (alpha, beta parameters)
ui1 <- rbind(c(1,0,0,0,0,0), c(-1,0,0,0,0,0))
ci <- c(-1,-1)
```

```
####################################################################

# Call the constrOptim function
full.ml1 <- constrOptim(start.values1, ml.logl1, grad=ml.grad1,
   ui = ui1, ci = ci, mu = 1e-04, control=list("fnscale"=-1),
   outer.iterations = 100, outer.eps = 1e-05, hessian = TRUE)

####################################################################

# Organize the output
mle.beta1 <- full.ml1$par[2:6]
mle.alpha1 <- full.ml1$par[1]
mle.full1 <- full.ml1$value #log likelihood
mle.cov1 <- solve(-full.ml1$hessian) #covariance matrix

####################################################################

# Compute the AIC and BIC
AIC1 <- 2*(length(mle.beta1)+1)-2*(mle.full1)
BIC1 <- log(length(unique(subject)))*(length(mle.beta1)+1)-2*(mle.full1)

####################################################################

# Observed information
#ob = observed information = 1/i(hat(theta))
std.err <- "ob"
if (std.err=="ob"){
   mle_cov1a <- mle.cov1
}

####################################################################

# Hypothesis testing
formula1 <- y ~ trt + base + age + period
pp <- length(all.vars(formula1))

Stderr <- matrix(NA, nrow=pp, ncol=1)
Wald <- matrix(NA, nrow=pp, ncol=1)
pval <- matrix(NA, nrow=pp, ncol=1)
for (p in 1:pp){
   Stderr[p,] <- sqrt(mle_cov1a[(p+1),(p+1)])
   Wald[p,] <- (mle.beta1[p]/sqrt(mle_cov1a[(p+1),(p+1)]))^2
   pval[p,] <- 1-pchisq(Wald[p,1] , df=1, lower.tail = TRUE, log.p = FALSE)
}
```

```
results <- cbind(mle.beta1,Stderr, Wald, pval)
alpha_results <- cbind(mle.alpha1,sqrt(mle_cov1a[1,1]))
fit_stats <- rbind(mle.full1,AIC1,BIC1)

#format output
rownames(fit_stats) <- c("Log-Likelihood:", "AIC:", "BIC:")
colnames(fit_stats) <- c("")
colnames(results) <- c("Estimate", "Std.err", "Wald", "Pr(>|W|)")
rownames(results) <- c("(Intercept)", "trtprogabide", "base", "age", "period")
colnames(alpha_results) <- c("Estimate", "Std.err")
rownames(alpha_results) <- c("alpha")

####################################################################

# Print the output in a nice format
print(fit_stats)
cat("\n Coefficients:\n")
print(results)
cat("\n Estimated Correlation Parameters: \n")
print(alpha_results)

####################################################################
#                                                                  #
###################### 6.2 ML without Period #######################
#                                                                  #
####################################################################

# Estimated parameters from the ML approach for analysis of the
# epilepsy data when period is not included in the models

# Assign starting values from GEE
beta.start2 <- epil_gee2$geese$beta
alpha.start2 <- epil_gee2$geese$alpha
start.values2 <- t(t(c(alpha.start2,beta.start2)))

####################################################################

# Enforce the constraint: -1 < alpha < 1
# vector is defined as (alpha, beta parameters)
ui2 <- rbind(c(1,0,0,0,0), c(-1,0,0,0,0))
ci <- c(-1,-1)

####################################################################

# Call the constrOptim function
```

```
full.ml2 <- constrOptim(start.values2, ml.logl2, grad=ml.grad2, ui = ui2,
   ci = ci, mu = 1e-04, control=list("fnscale"=-1),
   outer.iterations = 100, outer.eps = 1e-05, hessian = TRUE)

#####################################################################

# Organize the output

mle.beta2 <- full.ml2$par[2:5]
mle.alpha2 <- full.ml2$par[1]
mle.full2 <- full.ml2$value #log likelihood
mle.cov2 <- solve(-full.ml2$hessian) #covariance matrix

#####################################################################

# Compute the AIC and BIC
AIC2 <- 2*(length(mle.beta2)+1)-2*(mle.full2)
BIC2 <- log(length(unique(subject)))*(length(mle.beta2)+1)-2*(mle.full2)

#####################################################################

# Observed information
#ob = observed information = 1/i(hat(theta))
std.err <- "ob"
if (std.err=="ob"){
   mle_cov2a <- mle.cov2
}

#####################################################################

# Hypothesis testing
formula2 <- y ~ trt + base + age
pp <- length(all.vars(formula2))

Stderr <- matrix(NA, nrow=pp, ncol=1)
Wald <- matrix(NA, nrow=pp, ncol=1)
pval <- matrix(NA, nrow=pp, ncol=1)
for (p in 1:pp){
   Stderr[p,] <- sqrt(mle_cov2a[(p+1),(p+1)])
   Wald[p,] <- (mle.beta2[p]/sqrt(mle_cov2a[(p+1),(p+1)]))^2
   pval[p,] <- 1-pchisq(Wald[p,1] , df=1, lower.tail = TRUE, log.p = FALSE)
}
results <- cbind(mle.beta2,Stderr, Wald, pval)
alpha_results <- cbind(mle.alpha2,sqrt(mle_cov2a[1,1]))
fit_stats <- rbind(mle.full2,AIC2,BIC2)
```

```
#format output
rownames(fit_stats) <- c("Log-Likelihood:", "AIC:", "BIC:")
colnames(fit_stats) <- c("")
colnames(results) <- c("Estimate", "Std.err", "Wald", "Pr(>|W|)")
rownames(results) <- c("(Intercept)", "trtprogabide", "base", "age")
colnames(alpha_results) <- c("Estimate", "Std.err")
rownames(alpha_results) <- c("alpha")

####################################################################

# Print the output in a nice format
print(fit_stats)
cat("\n Coefficients:\n")
print(results)
cat("\n Estimated Correlation Parameters: \n")
print(alpha_results)

####################################################################
#                                                                  #
################# 6.3 Likelihood ratio (LR) test ###################
#                                                                  #
####################################################################

# Test statistic: G = 2*(logL(reduced) - logL(full))
# Assuming reduced model (null) is correct, the sampling
# distribution of G is approximately Chi-Squared with df=1

G.period <- -2*( mle.full2 - mle.full1)
pval.period <- 1-pchisq(G.period, df=1, lower.tail = TRUE,
   log.p = FALSE)
G.period
pval.period

####################################################################
########################### THE END ################################
####################################################################
```