ABSTRACT

A Beta Regression Approach to Nonparametric Longitudinal Data Classification in Clinical Trials

Roberto Sergio Hernandez, Ph.D.

Chairperson: Jack D. Tubbs, Ph.D.

Classification is an important topic in statistical analysis. For example, in applications involving clinical trials, an often seen objective is to determine whether or not novel medicines and treatments differ from existing standards of care. There are numerous methods and approaches in the literature for this problem when the endpoint of interest is normally distributed or can be approximated by an asymptotic Normal distribution, yet, the approaches when using a non-normally distributed endpoint are limited. This is especially true when these endpoints are correlated across time. In this dissertation, we investigated several techniques for use with longitudinal, repeated measures data where there is a special interest in adapting some recent results found in the literature on Beta regression. The proposed methods provided a nonparametric, with regard to the design endpoint, model that can be used in the repeated measures problem.

A Beta Regression Approach to Nonparametric Longitudinal Data Classification in Clinical Trials

by

Roberto Sergio Hernandez, B.S., M.S.

A Dissertation

Approved by the Department of Statistical Science

James D. Stamey, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of
Doctor of Philosophy

Approved by the Dissertation Committee

Jack D. Tubbs, Ph.D., Chairperson

James D. Stamey, Ph.D.

Dean M. Young, Ph.D.

Philip D. Young, Ph.D.

Accepted by the Graduate School
May 2022

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to acknowledge my advisor Dr. Jack D. Tubbs for guiding and believing in me throughout this dissertation process, you encouraged me to keep learning and growing and I could not have asked for a better person in my corner. I would also like to acknowledge Dr. Dennis A. Johnston for helping me embark on my initial research journey and letting me know I could do this, I cannot thank you enough for taking me on as a student and helping me achieve this goal.

DEDICATION

To my parents and sister. Without you, none of this would have been possible.

CHAPTER ONE

Introduction and Literary Review

1.1 Introduction and Motivation

The statistical literature contains many procedures for modeling repeated data in the presence of time dependent covariates when the dependent response variable is normally distributed. More recently, the likelihood methods have allowed one to consider dependent variables with distributions only limited to the exponential family (Agresti, 2013).

The first objective of this dissertation is to investigate an approach for modeling proportions in the situation as described above. Our approach will be to consider Generalized Estimating Equations (GEE) and Generalized Linear Mixed Models (GLMM) when the distribution of the dependent variable has a Beta distribution.

The second objective of this dissertation is to use some recent results found in the Receiver Operating Characteristic (ROC) curve literature when one has multiple groups for which the repeated measurement model is appropriate. This approach will allow one to relax any distributional assumptions on the dependent response variable.

1.2 Dissertation Plan and Aims

In this chapter, we review the literature for repeated measures models while mainly focusing on the methods considered in (Hein, 2019). In addition, we introduce several motivating examples that will be used in the remainder of the dissertation. In Chapter Two, we provide a brief introduction for the concepts used in our research. These include: the Beta distribution, longitudinal data, and placement values as used in conjunction with the ROC. We present the methods and applications that were the focus of this dissertation in Chapter Three. These include: Beta regression, Beta

reconstruction, and ROC curve classification. In Chapter Four, we present the results of our methods for our three repeated measures datasets and how they compare and contrast. In Chapter Five, we extend our methods to the multiple active treatment model. Finally, Chapter Six contains a summary and discussion of possible future work.

1.3 Literary Review

The genesis for this dissertation were the ideas presented in an unpublished dissertation, (Hein, 2019), which investigated several univariate models and multivariate approaches for modeling longitudinal data when the endpoints were proportions. He considered the Beta distribution when modeling these endpoints. However, we decided to proceed solely with the univariate approaches mentioned in the literature: Generalized Estimating Equations (GEE), introduced by (Liang and Zeger, 1986) with modifications by (Hu et al., 1998), (McDaniel et al., 2013), and (Wang, 2014), and Generalized Linear Mixed Models (GLMM) (Zimprich, 2010).

The GEE and GLMM models have similarities to the Generalized Linear Model (GLM) as introduced by (Lachenbruch et al., 1990). From there, (Ferrari and Cribari-Neto, 2004) and (Cribari-Neto and Zeileis, 2010) expanded this model to accommodate data from the Beta distribution.

In this dissertation we adapted an approach considered by (Stanley, 2018), whereby the placement values [(Cai, 2004), (Sullivan Pepe and Cai, 2004), and (Hajian-Tilaki, 2013)] of the data can be modeled using Beta regression.

1.4 Hein Review

As mentioned above, (Hein, 2019) and his investigation in using the Beta regression model when applied to longitudinal data was formative for the work found in this dissertation. He considered two univariate and two bivariate Beta regression models. We reproduced much of the simulation results given in his thesis and agreed

with his conclusion that the univariate methods outperform the bivariate approaches in the cases that he considered, such as type-I error rate and RMSE. As a result, we restricted our attention to the GEE and GLMM models when the response variable has a Beta distribution.

1.4.1 Vorechovsky's Method and Nataf Transformation

Both (Liu and Der Kiureghian, 1986) and (Nataf, 1962) were used to generate the correlation structure for our repeated measures models using a method suggested by (Vořechovský, 2008). These methods allowed us to adjust all of the parameters for data generation, including the strength and structure of the correlation. These datasets were then used to compare our classification method across a variety of situations.

1.4.1.1 Application. In order to simplify the data creation step in the fully generated data examples, we modified and added to some of the functions in (Hein, 2019) that used these concepts. An illustration of this is shown in Appendix A.

1.4.2 Notable Outcomes

We saw in (Hein, 2019) various ways to approach longitudinal Beta data analysis. Included were two models, one-treatment and two-treatment, which we could use and apply to our research.

Here, we show the longitudinal model for a single treatment. Starting at time 1, we have four time periods included, with β_1 as the intercept and regression parameter for the starting point and β_2 through β_4 accounting for the rest of the time periods. The single-treatment model is:

$$\log(\frac{\mu_{ij}}{1-\mu_{ij}}) = \beta_1 + \beta_2 * time_{i2} + \beta_3 * time_{i3} + \beta_4 * time_{i4}$$

where

$$Y_{ij} \sim Beta(\mu_{ij}, \phi).$$

A similar model can also be used for various treatments. Here, as an example, two treatments and four time periods are included. The two-treatment model is as follows:

$$\log(\frac{\mu_{ij}}{1-\mu_{ij}}) = \beta_1 + \beta_2 * time_{i2} + \beta_3 * time_{i3} + \beta_4 * time_{i4} + \beta_5 * trt_i + \beta_6 * trt_i * time_{i2} + \beta_7 * trt_i * time_{i3} + \beta_8 * trt_i * time_{i4}$$

for subjects $i = 1, \ldots, n$ and measurements $j = 1, \ldots, 4$ where $n = 12, 30, 50,$ and 100 equally distributed across two treatment groups ($trt_i = 0$ or 1).

1.4.2.1 Takeaways. After exploring and analyzing both of these types of models, we decided the former can be applied to a two-treatment clinical-trial situation as long as some alterations are made to the data beforehand. This kept our analysis simpler and gave us more freedom to experiment with a number of examples.

1.5 Motivating Datasets

We decided to have three datasets in our research:

(1) a fully generated datset where we can see how well the methods perform in a "controlled environment".

(2) a partially generated dataset, which we attained from making some distributional modifications on real data.

(3) a real world dataset, namely, the unchanged data from the previous example, which has been used outside of our research for repeated measures analysis.

Our completely generated data parameters were partially taken from (Hein, 2019), with some adjustments being made to fit our methods. The partially generated and real world Treadmill datasets were found in (Walker and Shostak, 2010) and were initally used for SAS analysis. Here, we proceeded to analyze all of the data using R.

1.5.1 Fully Generated Data

For the fully generated data, we used the methods proposed in (Hein, 2019). We used Vorechovsky's method (Vořechovský, 2008) to generate beta responses from different correlation structures and parameters. Within this method, Nataf's transformation (Nataf, 1962) was used to initially make the correlation matrix used.

We tried different correlation matrix structures (AR1, CS) to see if an "early" or "late" jump in the data gives way to a better model fit (in terms of accuracy in predicting the correct time a jump occurs). The data generation parameters were:

- Correlation structures: AR1, CS
- $n = 15, 100$ (per treatment)
- $\mu = 0.03, 0.5$
- $\rho = 0.1, 0.5$

We included four examples/situations, with each varying in sample size for comparison:

(1) We began with data generated by the parameters $\mu = 0.03$ and $\rho = 0.1$, where there was an "early" jump in the active treatment (time 2). We compared having $n = 15$ to $n = 100$. For both of these, we generated data with an AR1 correlation structure and with an exchangeable (CS) structure.

(2) We, then, used those same parameters ($\mu = 0.03, \rho = 0.1$) and incorporated a "late" jump (time 4) into the data, while also comparing when $n = 15$ and $n = 100$. Again, in both instances, we used an AR1 and an exchangeable (CS) correlation structure for the generation of said data.

(3) Now, we went back to an "early" jump (time 2) but with $\mu = 0.5$ and $\phi = 0.5$ as our parameters. However, again, we compared with $n = 15$ and $n = 100$

while generating data through an AR1 and exchangeable correlation structure.

(4) Lastly, we followed suit from the previous example, using $\mu = 0.5, \phi = 0.5, n = 15, 100$, and generating with AR1 and exchangeable correlation structures, yet we created a "late" jump (time 4).

1.5.2 Partially Generated Data

Generating the data consisted of splitting up the active and placebo treatments of the original Treadmill data, finding the means and standard deviations of the split data, and using a Normal distribution to create new data points. Only these new, normally distributed responses were used for analysis and considered "partially generated" because information from the real world dataset was used to create it. (The standard deviations were scaled down by 0.5 for the simulation and rounded values were used for uniformity with original data.)

The data generated is summarized in Tables 1.1, 1.2 and Figure 1.1 below.

Table 1.1. Partially Generated Treadmill Data, Means

Treatment/Time	WD0	WD1	WD2	WD3	WD4
Placebo	171.94	167.83	174.56	180.89	182.72
Active	170.05	185.05	200.50	201.15	208.00

Table 1.2. Partially Generated Treadmill Data, Standard Deviations

Treatment/Time	WD0	WD1	WD2	WD3	WD4
Placebo	42.52	41.25	44.16	46.28	46.51
Active	42.05	33.80	47.11	33.80	40.71

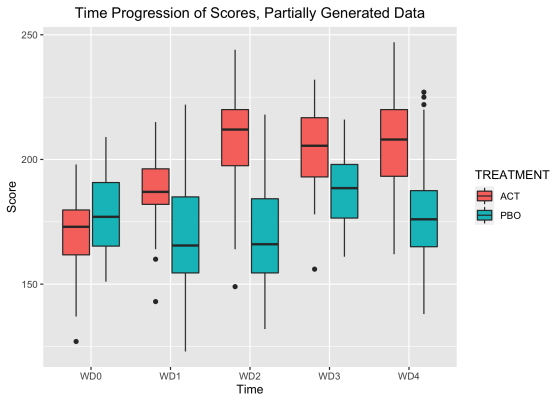

Figure 1.1. Time Progression of Scores, Partially Generated Data

We used a sample of size $n = 30$ for both active and placebo, giving a total of 60 patients. We see that at time 0, both treatments are very similar, with the placebo group with even slightly larger mean values. From there, however, we see an upward trend with the active and placebo group throughout, but the active treatment means increased about 300% more.

The variation trend is not a clear as the mean is. For the placebo, the standard deviations increased and level off after time 2. For the active treatment, they oscillated up, down, and back up slightly. Overall, variation in the active group ranged more than the placebo.

We see the scores for both treatments start out relatively equal, with medians at around 175, with the placebo treatment actually having larger scores than the active treatment. As time progresses, the placebo treatment stays hovering at around the same value, 165 for times 1 and 2, 185 and 175 for times 3 and 4. The active treatment scores, however, increase relatively quickly for the first two times, up to 215 by time 2, with it leveling off after that. We see the most separation between the treatments at time 2, with the medians differing by a value of about 50.

1.5.3 Real World Data

Our motivating dataset came from *Common Statistical Methods for Clinical Research with SAS Examples* (Walker and Shostak, 2010). Patients were randomly assigned to receive either the new drug Novafylline, thought to reduce the symptoms of intermittent claudication, or a placebo in a 4-month double-blind study. The primary measurement of efficacy is the walking distance on a treadmill until discontinuation due to claudication pain. A total of 38 patients underwent treadmill testing at baseline (month 0) and at each of 4 monthly, follow-up visits. The treadmill walking distances (in meters) were recorded. Patients were stratified by sex. The experiment was conducted to see if there was any distinction in exercise tolerance profiles.

Scores for the real world data are seen in Figure 1.2 below.

Figure 1.2. Time Progression of Scores, Real World Data

We see the scores for both treatments start out relatively equal, with medians at around 175, with the placebo treatment having larger scores than the active treatment, again, but its spread much more similar to the active treatment than in the partially generated data. As time progresses, the placebo treatment stays hovering at around the same value. The active treatment scores, differing from the previous example, increase more slowly and with less magnitude for the first two times, up to only 190 by time 2, with it leveling off to slightly over 200 after that. We see the most

separation between the treatments at time 2 and 4, it appears, but the spread shows more separation at time 4, with the active treatment having a much more compact distribution than the placebo.

1.5.4 Conclusion

Ultimately, we had completely controlled Beta distributed data, partially controlled normally distributed data, and uncontrolled non-distributional (in the sense that we did not know or care to know the distribution) data. These datasets had sufficient variety amongst them for our methods to be applied, information to be gathered, and meaningful conclusions to be given for multiple scenarios.

CHAPTER TWO

Research Concepts

2.1 Introduction and Discussion of Notation

In this chapter, we introduce the concepts we needed for our research methods. These will include an overview of the Beta distribution and its utility when considering longitudinal, repeated measures data. In addition, we present how the concept of placement values in conjunction with ROC curves and the AUC can be utilized in a clinical trial setting.

2.2 Beta Distribution

The Beta distribution is derived from a pair of independent Gamma random variables. Specifically, let X_1 and X_2 be two independent Gamma random variables having the joint PDF

$$h(x_1, x_2) = \frac{1}{\Gamma(a)\Gamma(b)} x_1^{a-1} x_2^{b-1} e^{-x_1 - x_2}, \quad 0 < x_1 < \infty, \quad 0 < x_2 < \infty,$$

zero elsewhere, where $a > 0$ and $b > 0$.

Let $Y_1 = X_1 + X_2$ and $Y_2 = X_1/(X_1 + X_2)$ when X_1 and X_2 are independent it follows that the marginal PDF of Y_2 is

$$g_2(y_2) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} y_2^{a-1}(1 - y_2)^{b-1},$$

which is the PDF of the Beta distribution with parameters a and b when $y_2 \in (0, 1)$ (Hogg et al., 2013).

The Beta function is

$$B(a, b) = \int_0^1 x^{a-1}(1 - x)^{b-1} dx$$

and is related to the Gamma function as

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}.$$

The Beta distribution, $X \sim B(a,b)$, n_{th} moments are simplified to

$$E(X^n) = \frac{B(a+n,b)}{B(a,b)} = \frac{\Gamma(a+n)\Gamma(a+b)}{\Gamma(a+b+n)\Gamma(a)},$$

therefore, the mean and variance are

$$E(X) = \frac{a}{a+b}, \quad Var(X) = \frac{ab}{(a+b)^2(a+b+1)}.$$

A brief introduction to the Beta generalized linear model given in (Ferrari and Cribari-Neto, 2004) is presented here. By letting $\mu = \frac{a}{a+b}$ and $\phi = a+b$, we obtain the reparameterized Beta distribution with mean and variance

$$E(Y) = \mu, \text{ and } Var(Y) = \frac{\mu(1-\mu)}{1+\phi}.$$

Let $y_1, \ldots, y_n$ be independent random variables from a Beta density with mean μ_t, $t = 1, \ldots, n$ and scale parameter ϕ. Then the Beta regression model can be written as

$$g(\mu_t) = \sum_{i=1}^{k} x_{ti}\beta_i = \eta_t,$$

where β is a vector of regression parameters, $x_{t1}, \ldots, x_{tk}$ are observations on k covariates, and g is a monotonic link function. Using the logit link, we have $\mu_t = \dfrac{1}{1+e^{-x_t'\beta}}$.

Estimates of the original parameters a and b are then

$$\hat{a} = \frac{\hat{\phi}}{1+e^{-x_t'\hat{\beta}}} \text{ and } \hat{b} = \hat{\phi}\left(1 - \frac{1}{1+e^{-x_t'\hat{\beta}}}\right).$$

This type of model is part of the foundation of what we proceeded to use for modeling and the concept of Beta reconstruction.

2.2.1 Motivation

The Beta distribution is very flexible, which is one of the reasons we were able to use it for our research. It can take on almost any shape, allowing us to use it to model practically any data that is encompassed by its range.

In the context of this dissertation, the Beta distribution allowed us to take a non-traditional approach to classification, giving us the option to model non-normally distributed endpoints. We were able to arrive at a Beta distributed response variable and use Beta regression for further analysis. All of this to say, the flexibility of the Beta distribution is one of the reasons all of this work could be brought together, connected, and used for analysis of longitudinal, repeated-measures data.

2.3 Longitudinal, Repeated Measures Data

Longitudinal, repeated measures data are characteristically unique in that there are multiple observations, in our case across time, for each subject. This phenomena induces within subject correlation. Since clinical studies typically have a relatively small number of repeated measures with a limited number subjects, the analysis used for this data needs to account for subject-specific correlation.

In this dissertation, we focus on a clinical trial setting with two groups; an active treatment and a placebo control.

2.3.1 Correlation Structures

The goal of specifying a correlation structure when modeling longitudinal data is to be more efficient in estimating β. If done incorrectly, the efficiency of the model parameter estimates can be affected, which is one of the complexities of analyzing this type of data.

It is stated in (Agresti, 2013) that "a chosen model in practice is never exactly correct, but carefully choosing a working correlation structure can help with efficiency of the estimates", and (Agresti, 2015) also mentions "GEE estimators of β are con-

sistent even if the correlation structure is misspecified". In this research setting, we still get reasonable regression parameters and standard error estimates even if we incorrectly specify the correlation structure.

We decided to include various correlation structures in our research, such as independent, AR1, and CS, to have a more thorough body of work and more complete results. However, since we are not worried about inconsistent results, we did not focus on the differences of these correlation structures in terms of any specific measure. More details on these structures are included in Appendix A.

2.4 ROC Curve and AUC

The ROC curve plots the True Positive rate (sensitivity) against the False Positive rate (1 - specificity) for all possible cutpoints based on Y. This is shown as

$$\text{ROC}(\cdot) = \{(\text{FPR}(y),\ \text{TPR}(y)),\ y \in (\infty, \infty)\}.$$

From here, we can represent the ROC curve in terms of survival functions as

$$\text{ROC}(t) = S_A(S_P^{-1}(t)),\ t \in (0, 1),$$

where $S_A(\cdot)$ is the survival function for the active treatment population (A) and corresponding random variable Y_A, $S_P(\cdot)$ is the survival function for the placebo treatment population (P) and corresponding random variable Y_P, and t is the FPR.

The most common utility of the ROC curve is comparing the area under the curve, the AUC (Fubini, 1907). The AUC is defined as

$$\text{AUC} = \int_0^1 \text{ROC}(t)dt.$$

It is the probability that a randomly chosen active treatment subject has an observation higher than a randomly chosen placebo subject (Hanley and Mcneil, 1982). Therefore, a perfect test or classification method would yield an AUC equal to 1, and an uninformative test would give an AUC of 0.5 (the same as flipping a coin). An illustration of this is seen in Figures 2.1 and 2.2 below.

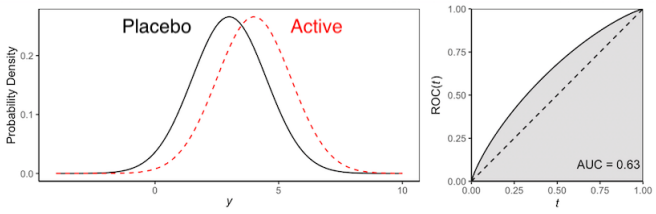

Figure 2.1. ROC curve with smaller AUC value

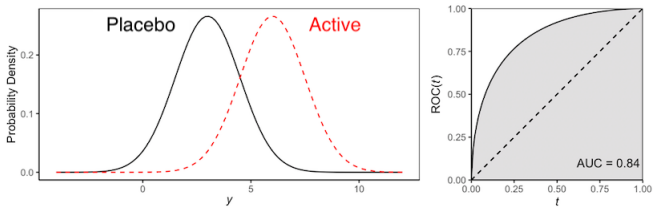

Figure 2.2. ROC curve with larger AUC value

In short, the more area under this curve, the more separation there is in the two samples being compared. A diagonal line would mean we correctly classified all of the samples from one population, yet we also incorrectly classified all of the samples from the other population. Therefore, for classification purposes, we want to see as much area under the curve as possible [(Hanley and Hajian-Tilaki, 1997), (Alonzo and Sullivan Pepe, 2002)].

2.5 Placement Values

Placement values are standardized versions of raw measurements. They are the proportion of the reference population with values larger than Y, and this standardizes Y to the distribution in that population, quantifying the separation between populations.

ROC curve analysis, in general, can be seen as the analysis of standardized measures (Hajian-Tilaki, 2013). In our research, those measures were placement values, more specifically, the values resulting from comparing the active treatment's endpoints to the control treatment's.

The placement values of a set of data, by definition, equal 1 minus the CDF of that data, also known as the survival curve. The notation is shown as

$$\text{PV}_A = 1 - F_P(Y_A) = S_P(Y_A),$$

where PV_A are the placement values for the active treatment subjects, $F_P(Y_A)$ represents the CDF of the placebo treatment population relative to the active treatment population, and $S_P(Y_A)$ represents the survival curve of the placebo treatment population relative to the active treatment population. This is the same as saying the placement values of Y, PV_Y, are the proportion of placebo population with values greater than Y. In other words,

$$S_P(y) = P[Y \geq y | A = 0] \Rightarrow \text{PV}_A = P[Y \geq Y_A | A = 0],$$

and by this definition, we see populations with a larger separation yield placement values close to 0 and populations with less separation have placement values close to 1. An illustration of this is seen in Figures 2.3 and 2.4 below.

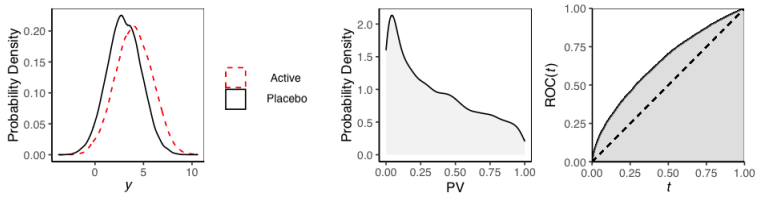

Figure 2.3. Placement Values with smaller AUC value

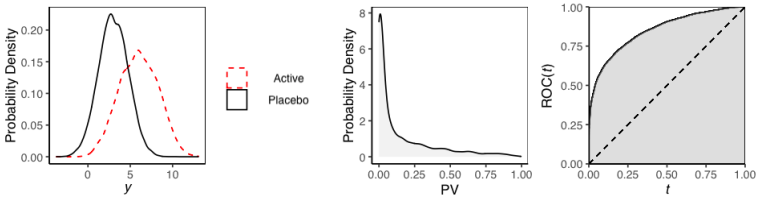

Figure 2.4. Placement Values with larger AUC value

We now have one set of scores (in the 0-1 range) that are dependent on the placebo treatment raw scores. Since they are in the interval (0,1), it is natural that we model them with a Beta distribution (Sullivan Pepe and Cai, 2004).

We see the CDF of the placement values, $\psi_{\text{PV}}(\cdot)$, is equal to the ROC(t) as

$$\begin{aligned}
\psi_{PV}(\cdot) &= P(\text{PV}_A < t) \\
&= P(S_P(Y_A)) \leq t \\
&= P(S_P^{-1}(t) < Y_A) \\
&= S_A(S_P^{-1}(t)) \\
&= \text{ROC}(t),\ t \in (0,1)
\end{aligned}$$

where t is the FPR and $\text{PV}_P = S_P(Y_P)$ is distributed as $U(0,1)$.

We also know the expected value of a random variable is the area under its survival function. Therefore,

$$\text{E}[1 - F_P(Y_A)] = \text{AUC}$$

and we see the mean of the placement value distribution is equal to the AUC.

We apply the non-parametric placement value methodology (Alonzo and Sullivan Pepe, 2002) from (Stanley, 2018), where she expands upon several ROC regression techniques, including a parametric (Sullivan Pepe and Cai, 2004) and semi-parametric (Cai, 2004) approach, as well [(Pepe, 1997), (Pepe, 1998), (Pepe, 2000)]. Because of this placement value idea, we are able to convert any exponential family distributed endpoints into Beta distributed and use Beta regression for analysis.

For our research, we found the AUC at every time period to see where the active treatment had more separation from the placebo, which we did by using placement values and the relationship their CDF has with the ROC curve.

CHAPTER THREE

Methods and Applications

3.1 Introduction

In this chapter, we discuss the methods used for our research. The overall concept of these methods is based upon a simple idea: The placement values are treated as the dependent variable in Beta regression models that allow for correlated, repeated measures data. The regression coefficients are then used to reconstruct a Beta distribution whose CDF is the ROC curve of the original response data. This ROC curve is used for the needed inference concerning the active and placebo groups at each time period within the repeated measure study.

3.2 Beta Regression

Beta regression models, as introduced by (Ferrari and Cribari-Neto, 2004), are a great way of modeling continuous dependent variables in the (0, 1) unit interval. In (Meaney and Moineddin, 2014), it is stated that, if the mean and dispersion are correctly specified, there seems to be much promise in using Beta regression instead of a traditional linear regression technique to model (0,1) response data. Where a possible bias can be the problem is misspecifying the dispersion, the type-1 error and power is comparable in linear and Beta regression, where the latter has the potential to be more powerful at detecting true non-zero differences between groups in a two-sample design. As we are using Beta regression in a classification problem, better detection potential was one of the reasons why we proceeded with this technique and clear motivation for using Beta regression moving forward.

Here, we introduce the various regression techniques we explored to analyze our data. We explain and present the "backbones" of our methods, namely, the

Generalized Linear Model (GLM) and the Linear Mixed Model (LMM), and how they are built upon to arrive at the the Beta regression techniques we proceeded to do the analysis with, namely, Generalized Estimating Equations (GEE) and the Generalized Linear Mixed Model (GLMM) (Hunger et al., 2012).

3.2.1 Generalized Linear Model (GLM)

The generalized linear model (Lachenbruch et al., 1990) is an extension of the ordinary linear model where normality is not required of the response data. The problem of interest is to model a function, $g(\cdot)$, of $E(Y) = \mu$ as a linear function of explanatory variables X. The GLM model consists of three components:

(1) the random component associated with the response variable.

(2) the systematic component associated with the explanatory variables.

(3) a link function that specifies the function, $g(\cdot)$.

Different link functions can be used to model data with various different distributional assumptions. For the purposes of our research, we used the logit link to model Beta distributed data, however, we used a more applicable modeling technique than the GLM, seeing as correlation and dependence needed to be included and accounted for.

3.2.2 Linear Mixed Model (LMM)

A linear mixed model is an extension of an ordinary linear model where the data are not completely independent, introducing random effects to the already present fixed effects. In our case, there is correlation in the response data over a fixed number of time periods.

The standard linear mixed model is

$$y = X\beta + Z\gamma + \varepsilon,$$

where the fixed-effect parameters β, known design matrix X, and normally distributed unobserved vector of random errors ε are carried over from the ordinary linear model. The differing components in the LMM are the unknown vector of random effects parameters γ, the known design matrix Z, and, although the random errors ε are present in the ordinary linear model, they are no longer required to be independent and/or homogeneous.

This model served as a foundation as well, like the GLM explained before, for our more appropriate modeling procedures that we used in the dissertation. However, since we did not restrict ourselves to normally distributed response variables, this type of model was not be used for our analysis. Now, the comparison becomes the use of GEE vs. GLMM as the longitudinal Beta regression method of choice (Carrière and Bouyer, 2002).

3.2.3 Generalized Estimating Equations (GEE)

Generalized Estimating Equations establish a predictive model for a dependent variable as a function of independent variables while simultaneously allowing for correlation in the repeated measures of these variables. The GEE model does not require specification of the distribution for the response variable. Instead, only the mean and variance of the dependent variable need be specified as long as the density function is a member of the exponential family of density functions. The results of this regression have a population-averaged interpretation for the regression coefficients (Hardin and Hilbe, 2003).

This method can be seen as an extension of the GLM when the correlation in the response variable is modeled. However, as a consequence of this difference, the GLM uses maximum likelihood (ML), whereas the GEE uses a quasi-likelihood (QL) method. This method allows us to not make assumptions about the distribution of the response observations (Wedderburn, 1974). In this case, the usual model selec-

tion techniques that rely on the likelihood function, such as, the Akaike information criterion (AIC), Bayesian information criterion (BIC), or Likelihood-ratio test (LRT), cannot be used (Cui and Quian, 2007). Even so, QL estimators have properties similar to those of ML estimators: $\hat{\beta}$ is asymptotically normal and is consistent as long as the link function and linear predictor function are correctly specified, even if the variance function is misspecified. Therefore, using QL instead of ML is not a huge tradeoff to be able to analyze, with reasonable statistical and computational efforts, correlated data that can take on any exponential family distribution. A major limitation of QL is missing data. The GEE method requires a stronger assumption about missing data than ML, in order for the QL estimates to be consistent. For GEE, the data must be "missing completely at random" (MCAR), which means that the probability an observation is missing is independent of that observation's value and the values of other variables in the entire data file. For ML, the data need only be "missing at random" (MAR), with what caused the data to be missing not depending on their values [(Agresti, 2015), (Hu et al., 1998), (Wang, 2014)].

In (Liang and Zeger, 1986), Generalized Estimating Equations are introduced as a marginal model conditioned on covariates. This procedure simultaneously models each marginal distribution, but takes into account the correlation structure when computing the standard errors for the estimating equation parameters (Agresti, 2015). The dependence in this model is not specified, yet treated as a nuisance parameter. GEEs were developed in order to produce regression estimates when analyzing repeated measures with non-normal response variables.

GEEs have a similar form to GLMs but since there is no full specification of the joint distribution, there is no likelihood function, as mentioned earlier. The GEE general form is

$$g(\mu_{it}) = x'_{it}\beta + \varepsilon_i,$$

representing the marginal distribution at each t, where $g(\cdot)$ is the link function.

The GEE and GLM provide identical results when the working correlation matrix is the identity matrix.

3.2.4 Generalized Linear Mixed Models (GLMM)

The Generalized Linear Mixed Model (GLMM) is a subject-specific model. Through an unobserved heterogeneity in the conditional mean, dependence is imposed. Basically, adding random effects to the Generalized Linear Model (GLM) yields the GLMM [(Zimprich, 2010), (Agresti, 2015)]. In longitudinal studies, repeated measurements are nested within a cluster that is a person observed over time, introducing dependence (within-subject correlation), to the data. This is where random effects come into play (Agresti, 2013). Since we use a non-linear link function in this model, only subject-specific interpretation of the parameters can be done.

The GLMM model is given by

$$g(\mu_i) = x_i'\beta + z_i'\gamma + \varepsilon_i$$

where $g(\cdot)$ is the link function, β is a $b \times 1$ vector of fixed regression coefficients for subject i, γ is a $h \times 1$ vector of random effects for subject i, and ε_i is a $t \times 1$ vector of within-subject random errors.

In the simplest of mixed models situations, where we assume the only random error is the between-subject variability, the model takes the form

$$g(\mu_i) = x_i'\beta + u_i + \varepsilon_i,$$

where u is a single random effect, subject-specific variability in this case, and ε are the random errors. This is called the intercept-only GLMM.

3.3 Beta Reconstruction

By performing Beta regression with the placement values using the GEE and GLMM models, we obtained estimates for the β's and ϕ parameters at each time

period. These estimates can then be reconstructed into the usual the Beta distribution notation (Ferrari and Cribari-Neto, 2004) given by

$$\mu = 1/(1 + \exp^{-\beta}), \quad a = \phi\mu, \quad b = \phi(1 - \mu),$$

where the expected value of the Beta distribution is $a/(a+b)$ and the AUC is $b/(a+b)$.

3.4 Classification

With a reconstructed Beta distribution from the active treatment placement values, we turned to (Sullivan Pepe and Cai, 2004) and used the fact that the cumulative distribution function of the placement values is equal to the ROC curve. From this reconstruction we have

$$ROC_{time=t} = B(a, b)$$

where $B(\cdot, \cdot)$ is the Beta function. The $ROC_{time=t}$ is used as a measure of the separation between the active treatment and the control groups at time t. Therefore, we now modeled the ROC curve to classify between active and placebo treatments at the various time periods to see where more separation is present.

This is where we used placement values to go from a two-treatment model to a single-group model, which allowed us to see separation between treatments without actually modeling both of them and without having to use more complex methods. We used Beta regression and Beta reconstruction because, by definition, placement values are in the (0,1) range. With the a and b parameter estimates we found for each time period, we modeled a Beta distribution CDF and, using (Sullivan Pepe and Cai, 2004), compared each respective time period's ROC curve against one another. We used these curves to find the AUC values and saw at which time period they were maximized, therefore finding the time with largest increase in separation between active treatment and placebo control.

We now show how the placement values and ROC curve are used for classification with real data as an introduction for the actual application of these methods.

Here, we illustrate what the partially generated and real world datasets look like in terms of both of these concepts and make conclusions in terms of some of their summary statistics, such as medians and IQR.

3.4.1 Placement Values

3.4.1.1 Partially Generated Data Placement Values. Placement values and their densities for the partially generated data are seen in Figures 3.1 and 3.2 below.

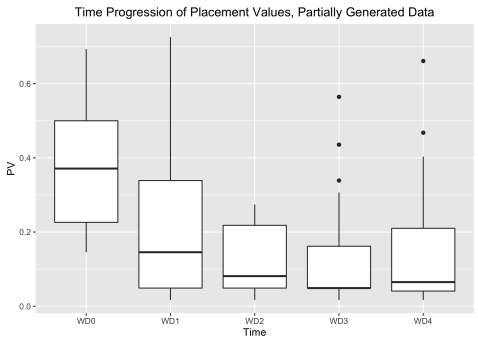

Figure 3.1. Time Progression of Placement Values, Partially Generated Data

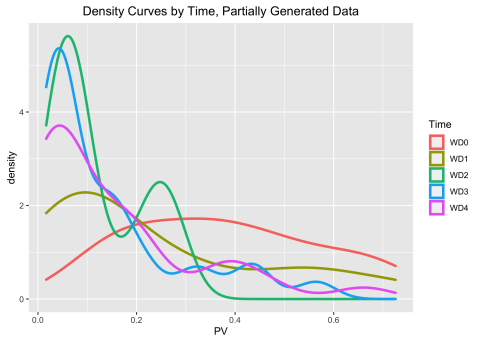

Figure 3.2. Density Curves by Time, Partially Generated Data

We see the placement values at time 0 with a median at about 0.4, not showing much separation between the two treatments. Quickly, the values decrease to under 0.2 at time 1 and to less than 0.1 by time 2 and stay at that level. The spread also begins decreasing after time 1, with the smallest happening at time 3. Including

outliers, it appears time 2 has the smallest spread and, overall, the smallest placement values.

The placement value density curves confirm previous conclusions. Time 0 has the mean closer to 0.4, showing not much separation between treatments. Then, the curves begin having peaks of increasing heights, with these peaks between 0 and 0.2, and the highest one appearing at time 2. Like in Figure 3.1, times 2 and 3 are very close, but it seems outliers cause time 2 to have higher separation of the treatments.

3.4.1.2 Real World Data Placement Values. Placement values and their densities for the real world data are seen in Figures 3.3 and 3.4 below.

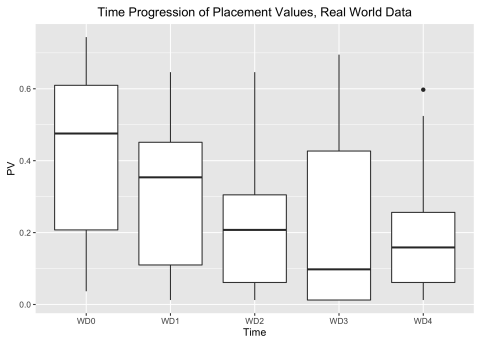

Figure 3.3. Time Progression of Placement Values, Real World Data

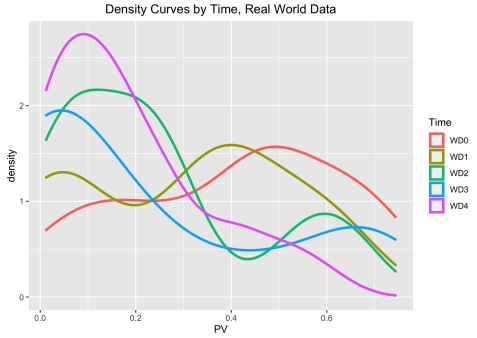

Figure 3.4. Density Curves by Time, Real World Data

The placement values for this dataset begin at about 0.5 with a very large spread. They decrease slowly from time 1 to time 3, where they bounce back up

slightly from a value of 0.1 at time 3 to 0.15 at time 4. The spread, however, is very clearly smaller at time 4 in comparison to the rest. This makes time 4 appear to be the moment with largest separation between the active and placebo treatments.

The placement value density curves show time 0 with the mean at around 0.5, showing less separation between treatments than the partially generated dataset. As time progresses, the curves begin having peaks of increasing heights, with these peaks between 0 and 0.2, with the highest one appearing at time 4, convincingly.

3.4.2 ROC Curve and AUC

3.4.2.1 Partially Generated Data, ROC Curve. Empirical ROC curves for the partially generated data are seen in Figure 3.5 below.

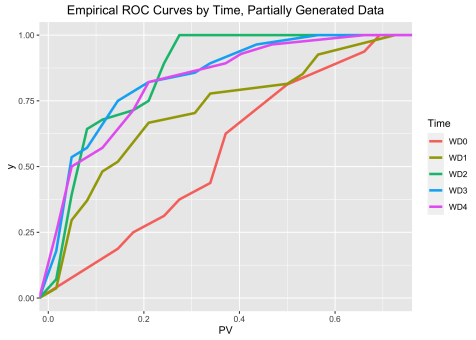

Figure 3.5. Empirical ROC Curves by Time, Partially Generated Data

We see the ROC curves progressing from almost a diagonal line $y = x$ at time 0 to slight separation at time 1, the largest separation at time 2, and time 3 and time 4 almost overlapping one another. Therefore, we are expecting to see our methods catch the maximum treatment separation at time 2.

3.4.2.2 Real World Data, ROC Curve. Empirical ROC curves for the real world data are seen in Figure 3.6 below.

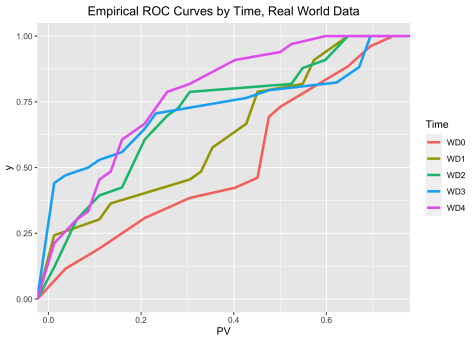

Figure 3.6. Empirical ROC Curves by Time, Real World Data

We see the ROC curves progressing from almost a diagonal line $y = x$ at time 0, again, to a slightly higher curve at time 1, a larger separation at time 2, and time 3 and time 4 being very similar, with time 4 appearing to be higher up, overall. Therefore, we are expecting to see our methods have close separation conclusions for time 3 and time 4, with the maximum treatment separation at time 4.

3.4.3 Design

The procedure for our classification experiment is as follows:

(1) Compute the placement values for the n treated subjects at time $t = 0$ using the control subjects at time $t = 0$. Therefore, we have $y_{i0} = \text{pv}_i$ for subject i, where $i = 1, \ldots, n$. So, at time $t = 0$, we get the placement values $\underline{y}_0 = (y_{10}, \ldots, y_{n0})'$.

(2) Compute the placement values for the n treated subjects at time $t = 1$ using the control subjects at time $t = 0, 1$. We now have $y_{i1} = \text{pv}_i$ for subject i, where $i = 1, \ldots, n$. So, at time $t = 1$, we get the placement values $\underline{y}_1 = (y_{11}, \ldots, y_{n1})'$.

(3) Repeat the previous steps with time $t = 2, 3, 4$ using the control group at time $t = 0, 1, 2$, $t = 0, 1, 2, 3$, and $t = 0, 1, 2, 3, 4$, respectively. This will yield the placement values $\underline{y}_2 = (y_{20}, \ldots, y_{n2})'$ for time $t = 2$, $\underline{y}_3 = (y_{13}, \ldots, y_{n3})'$ for time $t = 3$, and $\underline{y}_4 = (y_{14}, \ldots, y_{n4})'$ for time $t = 4$.

(4) Create a data frame with variables PAT (patient number) and PV (placement value), having $4n$ values per column. In our case, we used $n = 30$ patients, so we have a data frame with dimensions 120×2.

This same procedure was followed for our fully generated dataset, yet our times were $t = 1, 2, 3, 4$ instead of $t = 0, 1, 2, 3, 4$ and the sample size varied at either $n = 15$ or $n = 100$, depending on the scenario and example.

The resulting GEE model is:

$$g(PV) = \hat{\alpha} + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + \varepsilon_{ij},$$

where g is the logit link and ε_{ij} are the random errors.

The resulting GLMM model is:

$$g(PV) = \hat{\alpha} + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + u_i + \varepsilon_{ij},$$

where g is the logit link, u_i is the subject-specific random effect (random intercept model), and ε_{ij} are the random errors.

The same ϕ is used for all parameter estimates in their respective model for the Beta reconstruction process. (e.g. we have one ϕ and 5 βs for each model.)

We end up with a design matrix $\mathbf{X}\boldsymbol{\beta}$,

$$\mathbf{X} = \begin{bmatrix} j_n' & 0_n' & 0_n' & 0_n' & 0_n' \\ 0_n' & j_n' & 0_n' & 0_n' & 0_n' \\ 0_n' & 0_n' & j_n' & 0_n' & 0_n' \\ 0_n' & 0_n' & 0_n' & j_n' & 0_n' \\ 0_n' & 0_n' & 0_n' & 0_n' & j_n' \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix},$$

where $\mathbf{X}$ is $n \times 5$, $\boldsymbol{\beta}$ is 5×1, and $j_n' = (1, 1, \ldots, 1)'$.

Our model becomes

$$z = logit(\underline{pv}) = \mathbf{X}\boldsymbol{\beta} + \varepsilon,$$

where ε is a $5n \times 1$ vector containing the random Beta errors. Now, we can compute $(\hat{\alpha}, \hat{\beta})$ in the Beta regression using

$$E[\hat{z}] = e^{(\mathbf{X}\boldsymbol{\beta})}/(1 + e^{(\mathbf{X}\boldsymbol{\beta})}(1 + \phi)).$$

For example, with time $t = 0$,

$$E[\hat{z}] = e^{(\hat{\boldsymbol{\beta}}_0)}/(1 + e^{(\hat{\boldsymbol{\beta}}_0)}(1 + \phi)),$$

we can get (α_0, β_0) and compute the CDF at that specific time. With time $t = 4$,

$$E[\hat{z}] = e^{(\hat{\boldsymbol{\beta}}_4)}/(1 + e^{(\hat{\boldsymbol{\beta}}_4)}(1 + \phi)),$$

we can get (α_4, β_4) and compute the CDF.

We used this analysis at all time periods to see where separation between treatments started being more apparent. Then, we looked for the quantified departure of the active treatment from the placebo at each time point. (We have the GEE and the GLMM regression estimates at each of the 5 stages.)

All placebo values were kept at each respective time, yet only the new 30 active observations were used to get placement scores. Basically, we had 30 active vs. 30 placebo observations at time 0. At time 4, we had 30 active vs. 150 placebo observations. We did this exercising the assumption that the placebo treatment endpoints remain constant across time.

CHAPTER FOUR

Results

4.1 Introduction

Now, after we applied our methods to all three of our dataset examples (fully generated, partially generated, real world), we obtained these results. We describe the Beta regression, Beta reconstruction, and Classification results separately and by dataset.

Our overall conclusion was that our classification technique, including the intermediate steps of regression and reconstruction, did what it was supposed to do. While we saw incorrect classification in one of the situations posed with the fully generated datasets ("early" jump, $n = 100$), further analysis of the data was done to investigate this disparity. We discovered that at least a fourth of the data points were less than or equal to 0.005, making the distribution very susceptible to incorrect estimation from Beta regression methods. Since Beta regression is the first step after obtaining the placement values, the rest of our method did not correctly classify between the active and placebo treatments in this scenario.

In terms of our method being applied to real data, we arrived at an even higher level conclusion. Our research correctly classified at time periods where separation is seen in a "moderate" amount, yet struggled where the treatments almost completely overlapped or did not overlap at all. How did we arrive at this? Again, Beta regression does not work as well with values close to 0 or 1, which represent the placement values in the scenarios mentioned where there is too much or not enough separation between treatments. Bottom-line, we can clearly see times 0 and 1 are not where we have maximum separation between treatments, yet it is more difficult to decide on which time period is the one with this trait.

4.2 Beta Regression Applications

We start with an already transformed Beta response variable, placement values for the active treatment which take the placebo into account. From here, we applied both the GEE and GLMM modeling techniques. For GEE, we specified the correlation structure as we wrote the model. For GLMM, the correlation was already included in the random effects we are modeling. We also specified which distribution family the response follows, which we did by inputting a custom function developed by (Hein, 2019). This allowed us to model Beta distributed dependent data. For both methods, we set Time as the independent variable, for which we got a regression coefficient, along with getting a scale parameter.

4.2.1 Treadmill Data (Partially Generated & Real World)

We chose to do the modeling and analysis with both the GEE and GLMM models. Here, we specified "Beta" as the distributional assumption for the response variable, namely, the placement values of the treadmill walking distances (in meters). We also tried two of the correlation structures mentioned: auto-regressive(1) and exchangeable/compound-symmetric (independence was not included because we assumed, given that we have longitudinal data, there was dependence of some sort across time). Also, a transformation was done to the placement values to be able to deal with extremes at 0 and 1. This transformation was

$$\frac{(y(n-1)+0.5)}{n},$$

where y are the placement values and n is the sample size. This was introduced in (Smithson and Verkuilen, 2006) and mentioned in (Cribari-Neto and Zeileis, 2010). The logit link was used as default for both models and parameter estimates were obtained for times 1 through 4, using time 0 as reference.

4.2.2 Beta Data (Fully Generated)

For the Beta generated data, we started with Beta distributed data and found the placement values using treatment 1 as a placebo and treatment 2 as the active treatment. Times 2 through 4 were treated as the levels of the Time independent variable, having time 1 as the reference. Again, the logit link was used for both models. We varied the strength of correlation using recommendations from (Cohen, 1988) (small = 0.1, large = 0.5).

4.2.2.1 Correlation Structure Specification. When fitting the GEE models, we focused on using the correct models for the data, meaning we fit a GEE with AR1 specified correlation structure to data generated with an AR1 correlation, and did the same with the CS correlation structure. When fitting a GLMM model, we did not make any correlation structure assumptions when modeling, however, the data was generated using either an AR1 or CS correlation structure. Therefore, both of these were modeled for comparison.

4.3 Beta Regression Results and Conclusion

While we can make clear conclusions with each of our examples in terms of the regression coefficient behavior across time periods and models, the actual interpretation of these parameters is not of much use to our applications. We can say the odds of the placement values increase by e^{β_t} or decrease by $1 - e^{\beta_t}$ (in the negative β case) amount if the measurement of the original response variable comes from time t. Without proper context or knowledge of where these placement values came from or how we got them, that interpretation does not mean much. However, we used these Beta coefficients as a step in the direction of our classification method and saw a summary of how the data (in terms of placement values) was behaving across time and what that meant in terms of the placebo and active treatments being analyzed.

4.3.1 Fully Generated Data

4.3.1.1 $n = 15, 100$, $\mu = 0.03$, $\rho = 0.1$, "early" jump (time = 2).

- $n = 15$: With a small sample size, everything runs accordingly: we see the regression parameter β make the largest jump and have the largest value at time 2, exactly when the jump is made in the data.

- $n = 100$: Using a larger sample size, we get surprising results: the regression parameter keeps increasing as time goes on, showing the largest at time 3 for the GEE models and at time 4 for GLMM. However, the "early" jump is still present at time 2 for all models, which is when it happens in the actual data.

Table 4.1. Beta β Regression Coefficients, Fully Generated Data, Example 1 (small parameters, "early" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.19 / 0.37	0.19 / 0.37	0.21 / 0.43	0.20 / 0.43
Time 2	-2.64 / -2.10	-2.64 / -2.10	-1.92 / -1.76	-1.90 / -1.75
Time 3	-0.91 / -2.49	-0.89 / -2.46	-1.11 / -1.86	-1.06 / -1.85
Time 4	-1.42 / -2.38	-1.37 / -2.35	-1.48 /-1.88	-1.44 / -1.89

Table 4.1 is referenced in Figure 4.1 below.

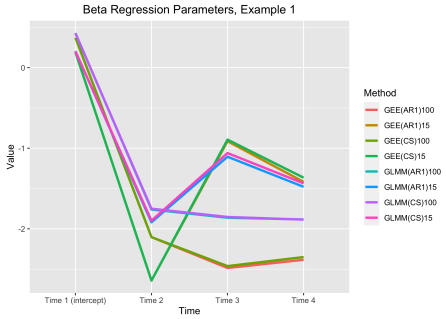

Figure 4.1. Beta β Regression Coefficients, Fully Generated Data, Example 1 (small parameters, "early" jump)

4.3.1.2 $n = 15, 100$, $\mu = 0.03$, $\rho = 0.1$, "late" jump (time = 4).

- $n = 15$: Again, everything works as expected using the small sample size. We see a jump in the regression parameter as the jump happens in the data, at time 4. The largest β is also present at time 4 for all models.

- $n = 100$: This time, with the large sample size, we see a correct estimation of the data. The larger β and the largest jump is seen at time 4, when the largest separation in the treatments happens.

Table 4.2. Beta β Regression Coefficients, Fully Generated Data, Example 2 (small parameters, "late" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.19 / 0.37	0.19 / 0.37	0.19 / 0.41	0.19 / 0.41
Time 2	-0.43 / 0.02	-0.43 / 0.02	-0.46 / 0.07	-0.45 / 0.07
Time 3	0.87 / 0.01	0.83 / 0.03	0.65 / 0.01	0.62 / 0.05
Time 4	-1.42 / -2.38	-1.37 / -2.35	-1.38 / -1.82	-1.35 / -1.81

Table 4.2 is referenced in Figure 4.2 below.

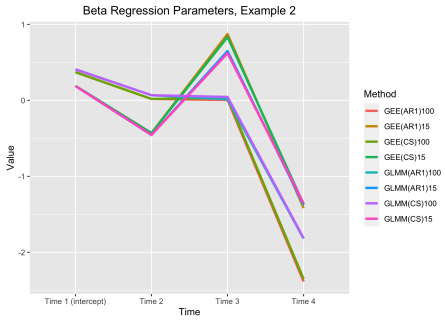

Figure 4.2. Beta β Regression Coefficients, Fully Generated Data, Example 2 (small parameters, "late" jump)

4.3.1.3 $n = 15, 100$, $\mu = 0.5$, $\rho = 0.5$, *"early" jump (time = 2).*

- $n = 15$: With the larger parameters and small sample size, we see similar results to the smaller parameters: largest β separation and magnitude is seen where the largest separation exists in the data, time 2.

- $n = 100$: Here, similarly to the previous "early" jump using smaller parameters, we see contradicting model results: the β parameters continue growing as time progresses, peaking at time 4 for GEE and GLMM models. However, again, we still see the largest jump in β size at time 2.

Table 4.3. Beta β Regression Coefficients, Fully Generated Data, Example 3 (large parameters, "early" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.19 / 0.37	0.19 / 0.37	0.24 / 0.47	0.23 / 0.46
Time 2	-3.37 / -2.50	-3.37 / -2.50	-3.16 / -1.96	-3.19 / -2.08
Time 3	-2.86 / -3.01	-2.30 / -2.90	-2.80 / -2.19	-2.47 / -2.39
Time 4	-2.03 / -3.32	-2.03 / -3.17	-2.25 / -2.40	-2.16 / -2.51

Table 4.3 is referenced in Figure 4.3 below.

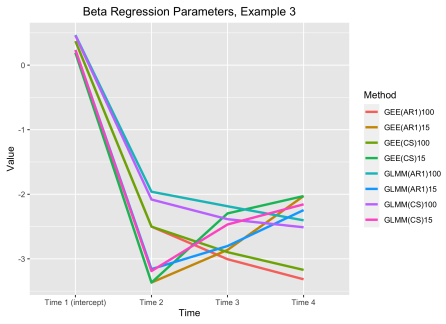

Figure 4.3. Beta β Regression Coefficients, Fully Generated Data, Example 3 (large parameters, "early" jump)

4.3.1.4 $n = 15, 100$, $\mu = 0.5$, $\rho = 0.5$, *"late" jump (time = 4).*

- $n = 15$: With the small sample size and late jump, the larger parameters yield similar results to the smaller ones: a larger separation and larger magnitude in the regression parameters at time 4 for GEE and GLMM in both the AR1 and CS models.

- $n = 100$: In similar fashion to the previous large sample size and late jump example, the models estimate correctly. We see the jump and the largest values for the β parameters at time 4.

Table 4.4. Beta β Regression Coefficients, Fully Generated Data, Example 4 (large parameters, "late" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.19 / 0.37	0.19 / 0.37	0.21 / 0.43	0.22 / 0.43
Time 2	-0.34 / 0.14	-0.34 / 0.14	-0.42 / 0.18	-0.45 / 0.18
Time 3	0.63 / 0.07	0.54 / 0.16	0.54 / 0.09	0.51 / 0.19
Time 4	-2.03 / -3.32	-2.03 / -3.17	-1.88 / -2.28	-1.94 / -2.33

Table 4.4 is referenced in Figure 4.4 below.

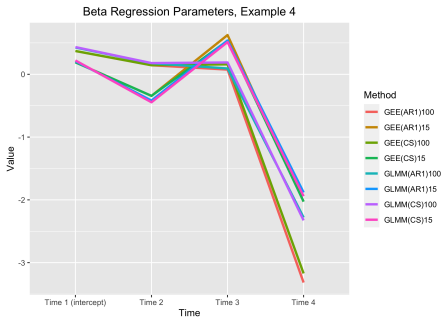

Figure 4.4. Beta β Regression Coefficients, Fully Generated Data, Example 4 (large parameters, "late" jump)

4.3.1.5 Scale (ϕ) Regression Coefficients, Examples 1-4. Here, we see a summary of the scale coefficients for all of the examples.

Table 4.5. Scale ϕ Regression Coefficients, Fully Generated Data, Examples 1-4

Ex/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
1	2.95 / 3.84	3.06 / 3.89	1.34 / 0.86	1.29 / 0.85
2	3.47 / 3.56	3.50 / 3.66	1.02 / 0.77	1.01 / 0.75
3	1.03 / 2.18	0.95 / 2.21	3.15 / 1.31	3.22 / 1.52
4	2.75 / 3.08	2.27 / 3.07	1.61 / 1.07	2.05 / 1.20

4.3.2 Partially Generated Data

For the partially generated data, we see a quickly declining β coefficient for all of the models from time 0 to time 2, with the GEE models declining slightly more than the GLMM. This means the largest jump in the data, as observed by GEE, would be time 2, as opposed to time 1, as shown by GLMM. We then see a sharp jump up for the GEE models at time 3 and a slight level off after that, making time 2 the time with the lowest (and largest in absolute value) β value. For the GLMM model, we see a very very slight decline after time 2, with a 0.01 increase at time 4. This makes time 3 the lowest β coefficient for GLMM.

The scale values are pretty similar across models, with GLMM having the smallest by around a 0.06 difference from GEE scale values.

Table 4.6. Beta β Regression Coefficients, Partially Generated Data

Time/Method	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
Time 0 (Int)	-0.48	-0.50	-0.51	-0.47
Time 1	-1.17	-1.17	-1.17	-1.26
Time 2	-2.00	-2.00	-2.00	-1.73
Time 3	-1.83	-1.82	-1.82	-1.76
Time 4	-1.73	-1.73	-1.73	-1.75

Table 4.6 is referenced in Figure 4.5 below.

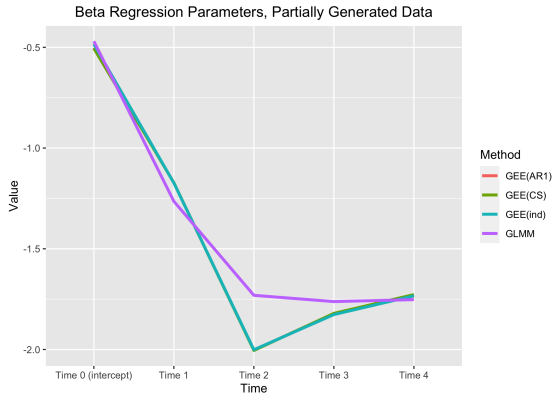

Figure 4.5. Beta β Regression Coefficients, Partially Generated Data

Table 4.7. Scale ϕ Regression Coefficients, Partially Generated Data

Scale/Method	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
ϕ	1.82	1.82	1.82	1.76

4.3.3 Real World Data

The real world data also presents varying conclusions across model types as before, but here we see the opposite of what was seen in the partially generated data. Again, we see quickly declining β regression coefficients, this time from time 0 to time 3 for the GEE and GLMM models. This seems to show the largest jump in the data happening at time 1 for both the GEE and GLMM models. However, the GLMM models have a quick jump back up at time 4 whereas the GEE models decline even further, making the lowest β values time 3 and time 4 for GLMM and GEE, respectively.

The scale coefficients are more distinct than before, with the GLMM ϕ more than 1.1 less than the GEE models.

Table 4.8. Beta β Regression Coefficients, Real World Data

Time/Method	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
Time 0 (Int)	-0.39	-0.28	-0.30	-0.34
Time 1	-0.82	-0.78	-0.82	-0.97
Time 2	-1.14	-1.10	-1.12	-1.08
Time 3	-1.22	-1.21	-1.20	-1.49
Time 4	-1.48	-1.46	-1.44	-1.37

Table 4.8 is referenced in Figure 4.6 below.

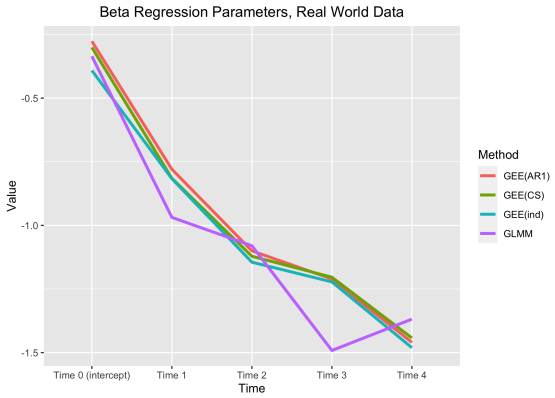

Figure 4.6. Beta β Regression Coefficients, Real World Data

Table 4.9. Scale ϕ Regression Coefficients, Real World Data

Scale/Method	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
ϕ	2.54	2.51	2.51	1.36

4.4 Beta Reconstruction Applications

Once we obtained the regression coefficients and scale parameter from the Beta regression, we were able to reconstruct our Beta distribution. This distribution was created from the sample of placement values relating the active treatment to the placebo, which therefore allowed us to model the full distribution they came from. With these distributions, we found their respective expected values for varying times and methods.

4.5 Beta Reconstruction Results and Conclusion

Again, as with our regression results, we use these as a way to proceed through our method and not as a standalone procedure. Therefore, we make conclusions about the reconstructed Beta parameters and expected values, but we are more concerned with using these results as a mean to an end in the larger scale of classification.

4.5.1 Fully Generated Data

4.5.1.1 Example 1. For our first example, we see all of the expected values quickly and drastically decreasing from time 1 to time 2, around a 0.5 decrease for both methods in all correlation structures. From time 2 to time 3, the situations where $n = 15$ have an expected value rise of about 0.2 whereas the $n = 100$ situations decrease even further, even if very slightly. Between time 3 and time 4, the former values decrease by a value of 0.1, making the lowest point of the $n = 15$ situations be time 2. In the same time frame, the $n = 100$ GEE models increase, again, ever so slightly, causing their lowest points to be at time 3. The $n = 100$ GLMM models, however, continue decreasing and, by a very small margin, have their lowest points occur at time 4. This shows the same disparity present with the β estimates: inconsistency between GEE and GLMM in the maximum separation prediction with a larger sample size. However, all models in both situations still consistently show the jump from no separation to large separation at time 2.

Table 4.10. Regression Reconstruction Expected Values, Fully Generated Data, Example 1 (small parameters, "early" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.55 / 0.59	0.55 / 0.59	0.55 / 0.61	0.55 / 0.61
Time 2	0.07 / 0.11	0.07 / 0.11	0.13 / 0.15	0.13 / 0.15
Time 3	0.29 / 0.08	0.29 / 0.08	0.25 / 0.13	0.26 / 0.14
Time 4	0.20 / 0.08	0.20 / 0.09	0.19 / 0.13	0.19 / 0.13

Table 4.10 is referenced in Figure 4.7 below.

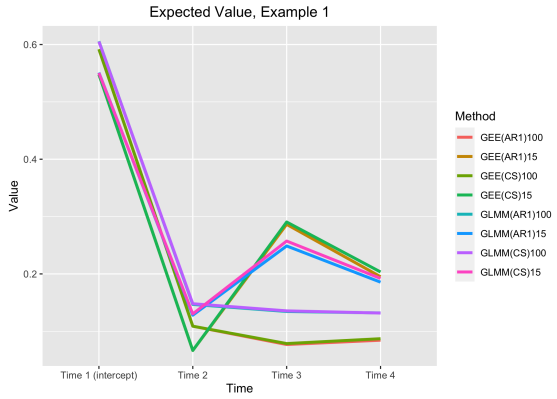

Figure 4.7. Regression Reconstruction Expected Values, Fully Generated Data, Example 1 (small parameters, "early" jump)

4.5.1.2 Example 2. In our second example, we see all methods reflect a relatively small decrease from time 1 to time 2. At time 3, the $n = 15$ trials increase 0.3 while the $n = 100$ stay hovering at about the 0.5 value. Then, from time 3 to time 4, all methods and sample sizes decrease immensely to values between 0.1 and 0.2, making time 4 the lowest point of all of the situations, as expected. This example shows consistency across methods and sample sizes, pointing out time 4 as the time when the largest jump and the largest separation between treatments is present.

Table 4.11. Regression Reconstruction Expected Values, Fully Generated Data, Example 2 (small parameters, "late" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.55 / 0.59	0.55 / 0.59	0.55 / 0.60	0.55 / 0.60
Time 2	0.39 / 0.50	0.39 / 0.50	0.39 / 0.52	0.39 / 0.52
Time 3	0.71 / 0.50	0.70 / 0.51	0.66 / 0.50	0.65 / 0.51
Time 4	0.20 / 0.08	0.20 / 0.09	0.20 / 0.14	0.21 / 0.14

Table 4.11 is referenced in Figure 4.8 below.

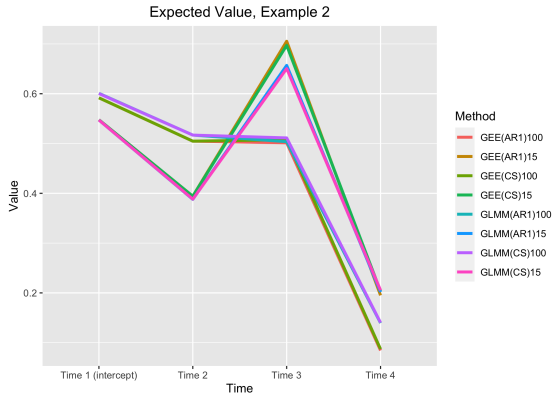

Figure 4.8. Regression Reconstruction Expected Values, Fully Generated Data, Example 2 (small parameters, "late" jump)

4.5.1.3 Example 3. In this third example, as seen in the first example with an "early" jump, there is a huge decrease from time 1 to time 2, from values at 0.6 to 0.1 and less than 0.05 for the $n = 15$ situations. Then, consistently, values continue to decrease slowly for the $n = 100$ situations while the $n = 15$ sample size situations slowly increase from time 2 to time 4. This makes the lowest point equal to time 2 for $n = 15$ and time 4 for $n = 100$ sample sizes. Inconsistency is seen across methods when measuring the largest separation, however, the large jump at time 2 is correctly seen by all methods in both sample sizes.

Table 4.12. Regression Reconstruction Expected Values, Fully Generated Data, Example 3 (large parameters, "early" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.55 / 0.59	0.55 / 0.59	0.56 / 0.61	0.56 / 0.61
Time 2	0.03 / 0.08	0.03 / 0.08	0.04 / 0.12	0.04 / 0.11
Time 3	0.05 / 0.05	0.09 / 0.05	0.06 / 0.10	0.08 / 0.08
Time 4	0.12 / 0.04	0.12 / 0.04	0.10 / 0.08	0.10 / 0.08

Table 4.12 is referenced in Figure 4.9 below.

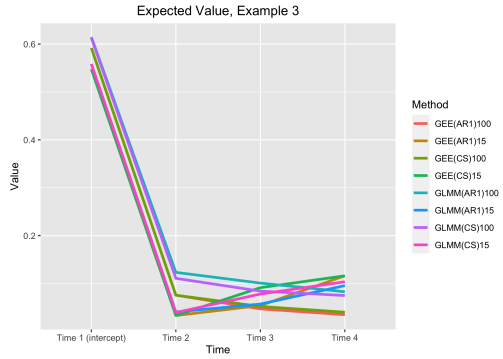

Figure 4.9. Regression Reconstruction Expected Values, Fully Generated Data, Example 3 (large parameters, "early" jump)

4.5.1.4 Example 4. In this example, we see almost the exact results as with the second example except with slightly less variation, which the larger parameters might explain. We see a slight decrease from time 1 to 2 with all models, with the $n = 15$ models decreasing more. Then, these same models have a large jump between time 2 and time 3, whereas the $n = 100$ situations hover at around the 0.5 value they were at in time 2. Finally, we see an extremely large decrease in expected value at time 4 for all models, making this the lowest point and the jump time for all models, correlation structures, and sample sizes in this final example.

Table 4.13. Regression Reconstruction Expected Values, Fully Generated Data, Example 4 (large parameters, "late" jump)

Time/Method	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.55 / 0.59	0.55 / 0.59	0.55 / 0.61	0.56 / 0.61
Time 2	0.42 / 0.54	0.42 / 0.54	0.40 / 0.54	0.39 / 0.54
Time 3	0.65 / 0.52	0.63 / 0.54	0.63 / 0.52	0.63 / 0.55
Time 4	0.12 / 0.04	0.12 / 0.04	0.13 / 0.09	0.13 / 0.09

Table 4.13 is referenced in Figure 4.10 below.

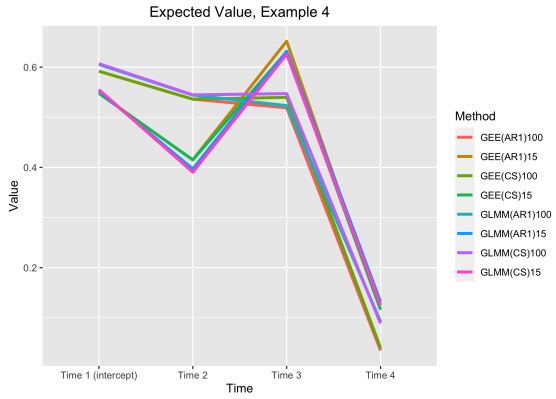

Figure 4.10. Regression Reconstruction Expected Values, Fully Generated Data, Example 4 (large parameters, "late" jump)

4.5.2 Partially Generated Data

With the partially generated data, we see different results for the different regression methods. Both start out the same, with a decrease in expected value of about 0.2 between time 0 and time 2. Then, the GLMM model declines slightly more at time 3, before coming back up at time 4, making the lowest GLMM expected value occur at time 3. The GEE models, regardless of correlation structure, all increase continually from time 2 to time 4, making their lowest expected values happen at time 2, instead. Despite the inconsistency in finding the largest separation between treatments, the largest jump in magnitude is seen at time 1 for all models.

Table 4.14. Regression Reconstruction Expected Values, Partially Generated Data

Time/Method	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
Time 0 (Int)	0.38	0.38	0.38	0.38
Time 1	0.24	0.24	0.24	0.22
Time 2	0.12	0.12	0.12	0.15
Time 3	0.14	0.14	0.14	0.15
Time 4	0.15	0.15	0.15	0.15

Table 4.14 is referenced in Figure 4.11 below.

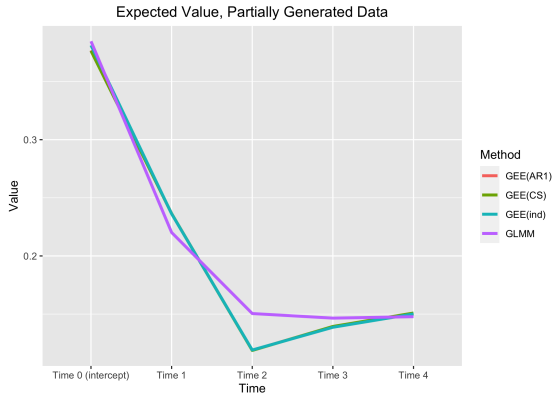

Figure 4.11. Regression Reconstruction Expected Values, Partially Generated Data

4.5.3 Real World Data

With our real world data, we see, again, contradicting results between the GEE and GLMM regression techniques, but this time at different time periods. We see consistent decline of the expected value from time 0 to time 3, with the GLMM method declining more than GEE, especially between time 2 and time 3. Then, GLMM increases in expected value from about 0.18 to 0.2, whereas the GEE methods decrease from 0.22 to about 0.19. Therefore, we have the GEE models with a expected value floor at time 4 and the GLMM model at time 3. Again, despite this inconsistency, the largest jump is seen at time 1 for all models.

Table 4.15. Regression Reconstruction Expected Values, Real World Data

Time/Method	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
Time 0 (Int)	0.40	0.43	0.43	0.42
Time 1	0.31	0.31	0.31	0.28
Time 2	0.24	0.25	0.25	0.25
Time 3	0.23	0.23	0.23	0.18
Time 4	0.19	0.19	0.19	0.20

Table 4.15 is referenced in Figure 4.12 below.

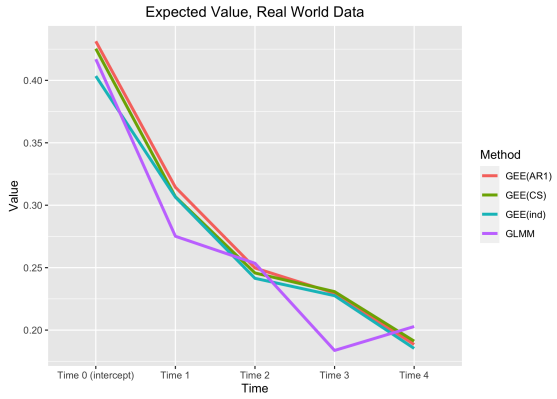

Figure 4.12. Regression Reconstruction Expected Values, Real World Data

4.6 Classification Applications

After we obtained our reconstructed Beta distributions for each dataset, we proceeded with our classification application by generating an ROC curve from the Beta parameters and calculating the AUC for each regression method and correlation structure at each time period and for each example.

4.7 Classification Results and Conclusion

Our classification method, in all scenarios, did what it was supposed to do: it correctly located which time period exhibited the largest jump in separation between the active treatment and the placebo.

4.7.1 Fully Generated Data

4.7.1.1 Example 1. In this example, we observe rapidly rising AUC values in the first time change, from time 1 to time 2, from an AUC of 0.4-0.45 to values at 0.82-0.86 for all models and sample sizes. Then, a sharp decline occurs for the $n = 15$ sample size situations at time 3, where a slight rise in AUC happens for the $n = 100$ models. At time 4, an extra slight rise in values is seen in the $n = 100$ GLMM models

and the $n = 15$ GEE and GLMM models, whereas the $n = 100$ GEE models show a tiny decrease, making the AUC peak be at time 2 for all $n = 15$ models, time 3 for the $n = 100$ GEE models, and time 4 for $n = 100$ GLMM models. Despite the disparity among models when gauging separation magnitude, all of them correctly show the jump in separation difference at time 2.

The fully generated data ROC curves by time are referenced in Figures 4.13 and 4.14 below. Each graph represents the GEE(AR1), GEE(CS), GLMM(AR1), and GLMM(CS) models, respectively.

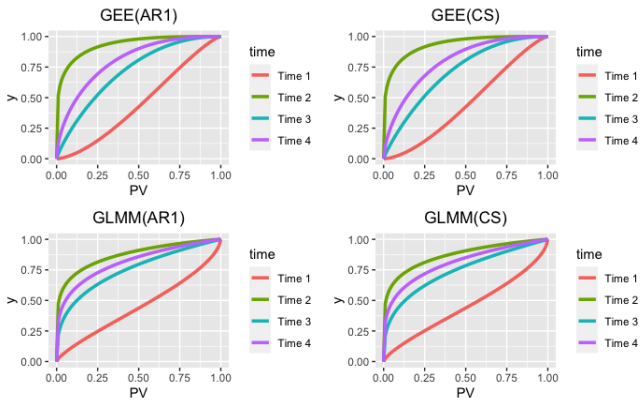

Figure 4.13. ROC Curves by Time, Fully Generated Data, Example 1 ($n = 15$)

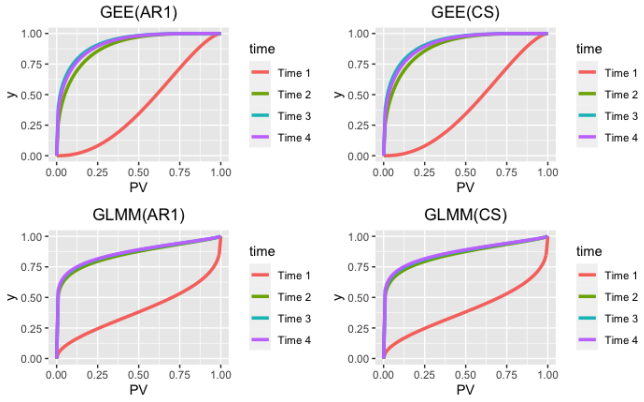

Figure 4.14. ROC Curves by Time, Fully Generated Data, Example 1 ($n = 100$)

Table 4.16. Area Under ROC Curve (AUC), Fully Generated Data, Example 1 (small parameters, "early" jump)

Time/Parameter	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.45 / 0.41	0.45 / 0.41	0.45 / 0.39	0.45 / 0.39
Time 2	0.93 / 0.89	0.93 / 0.89	0.87 / 0.85	0.87 / 0.85
Time 3	0.71 / 0.92	0.71 / 0.92	0.75 / 0.87	0.74 / 0.86
Time 4	0.80 / 0.92	0.80 / 0.91	0.81 / 0.87	0.81 / 0.87

Table 4.16 is referenced in Figure 4.15 below.

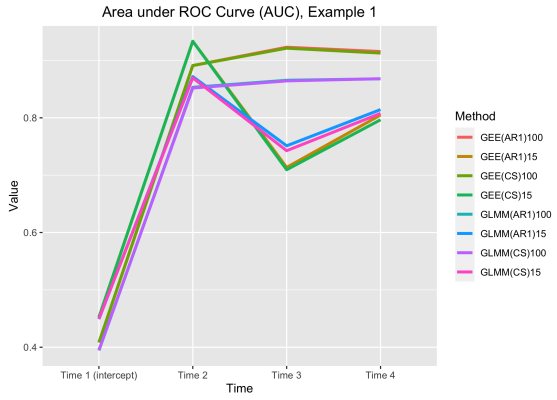

Figure 4.15. Area Under ROC Curve (AUC), Fully Generated Data, Example 1 (small parameters, "early" jump)

4.7.1.2 Example 2. This example's results proved to be more consistent than the previous example's in terms of AUC. We see an increase from time 1 to time 2 for all models and sample sizes, and then a large decrease at time 3 for the $n = 15$ situations and a hovering at 0.5 for the $n = 100$ sample sizes. Then, an extremely large increase in all methods is present at time 4, yielding the highest AUC and largest separation between treatments to be at time 4. Here, all models are consistent with both the separation maximum and the maximum jump time, which also occurs at time 4.

The fully generated data ROC curves by time are referenced in Figures 4.16 and 4.17 below. Each graph represents the GEE(AR1), GEE(CS), GLMM(AR1), and GLMM(CS) models, respectively.

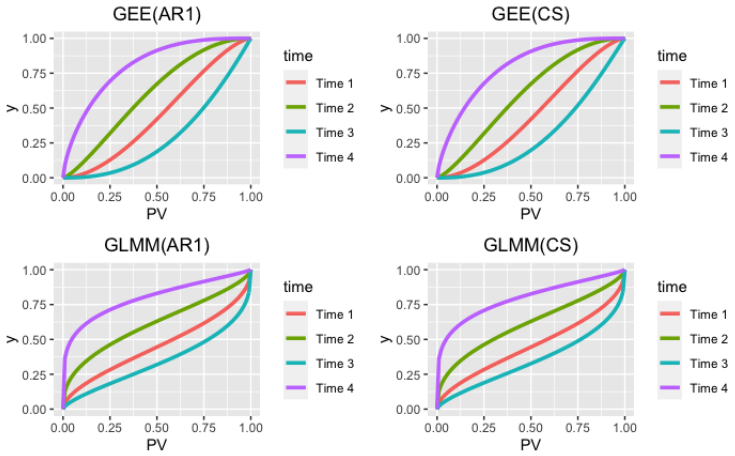

Figure 4.16. ROC Curves by Time, Fully Generated Data, Example 2 ($n = 15$)

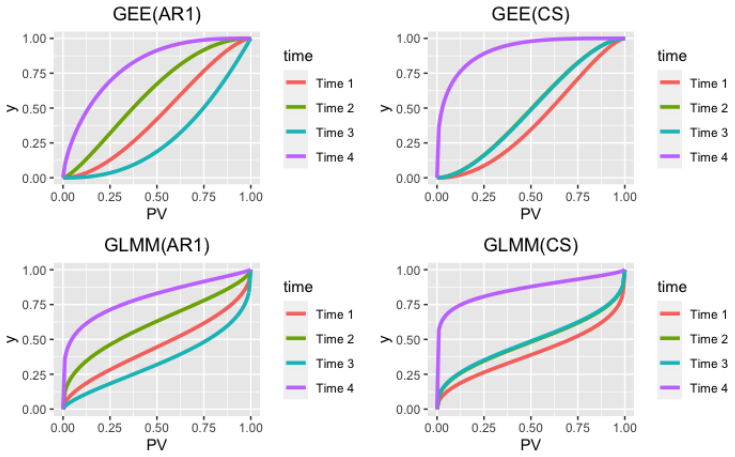

Figure 4.17. ROC Curves by Time, Fully Generated Data, Example 2 ($n = 100$)

Table 4.17. Area Under ROC Curve (AUC), Fully Generated Data, Example 2 (small parameters, "late" jump)

Time/Parameter	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.45 / 0.41	0.45 / 0.41	0.45 / 0.40	0.45 / 0.40
Time 2	0.61 / 0.50	0.61 / 0.50	0.61 / 0.48	0.61 / 0.48
Time 3	0.29 / 0.50	0.30 / 0.49	0.34 / 0.50	0.35 / 0.49
Time 4	0.80 / 0.92	0.80 / 0.91	0.80 / 0.86	0.79 / 0.86

Table 4.17 is referenced in Figure 4.18 below.

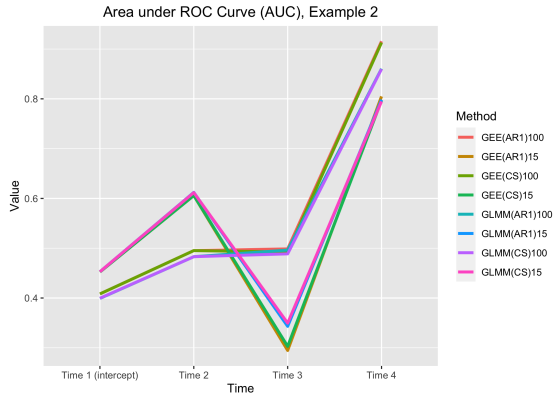

Figure 4.18. Area Under ROC Curve (AUC), Fully Generated Data, Example 2 (small parameters, "late" jump)

4.7.1.3 Example 3. In this example, we see a large, quick increase in AUC values between time 1 and time 2, from AUC values of 0.4 all the way to 0.85-0.95 for all methods and models. Then, the $n = 15$ situations decrease continually from time 2 to time 4, yielding the highest AUC at time 2. The $n = 100$ models, however, continually increase in that same time span, and yield the highest AUC value at time 4. Again, like the previous example handling an early jump in the data, the maximum separation between active treatment and placebo is estimated to be slightly different by different models. However, the jump in separation is seen at the correct time, time 2, with all methods.

The fully generated data ROC curves by time are referenced in Figures 4.19 and 4.20 below. Each graph represents the GEE(AR1), GEE(CS), GLMM(AR1), and GLMM(CS) models, respectively.

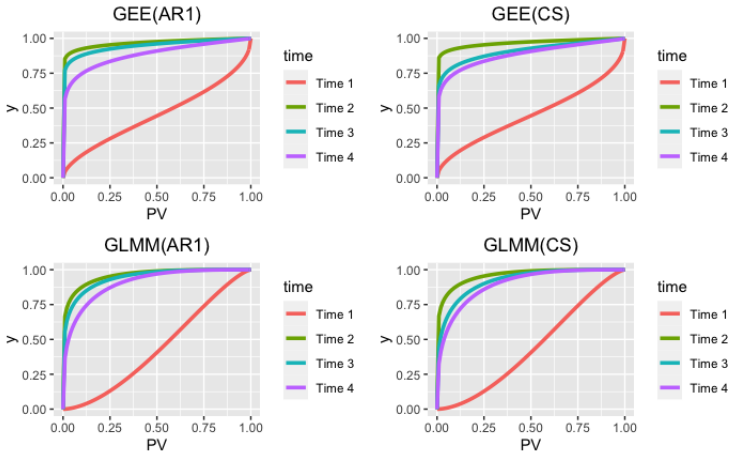

Figure 4.19. ROC Curves by Time, Fully Generated Data, Example 3 ($n = 15$)

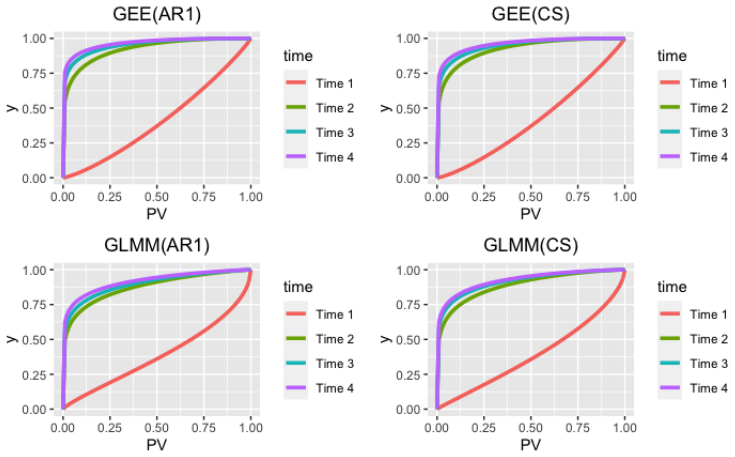

Figure 4.20. ROC Curves by Time, Fully Generated Data, Example 3 ($n = 100$)

Table 4.18. Area Under ROC Curve (AUC), Fully Generated Data, Example 3 (large parameters, "early" jump)

Time/Parameter	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.45 / 0.41	0.45 / 0.41	0.44 / 0.39	0.44 / 0.39
Time 2	0.97 / 0.92	0.97 / 0.92	0.96 / 0.88	0.96 / 0.89
Time 3	0.95 / 0.95	0.91 / 0.95	0.94 / 0.90	0.92 / 0.92
Time 4	0.88 / 0.97	0.88 / 0.96	0.90 / 0.92	0.90 / 0.92

Table 4.18 is referenced in Figure 4.21 below.

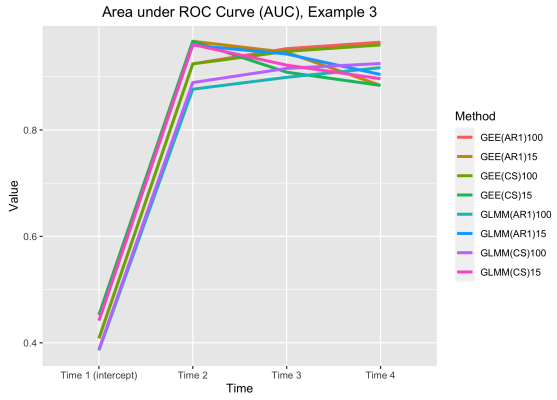

Figure 4.21. Area Under ROC Curve (AUC), Fully Generated Data, Example 3 (large parameters, "early" jump)

4.7.1.4 Example 4. In our last example of fully generated data, we see very similar results to our second example. A slight increase in AUC is seen for all methods from under 0.5 in the span between time 0 and time 1. Then, a large decrease to back under 0.4 for the $n = 15$ sample sizes is present at time 3, where the $n = 100$ models stay hovering at the previous time's value of about 0.5. Finally, we see a huge increase in AUC values up to over 0.8 for all models and sample sizes, showing the most separation between treatments to be at time 4. Here, the jump in treatment difference is also at its maximum and is correctly gauged by all models.

The fully generated data ROC curves by time are referenced in Figures 4.25 and 4.23 below. Each graph represents the GEE(AR1), GEE(CS), GLMM(AR1), and GLMM(CS) models, respectively.

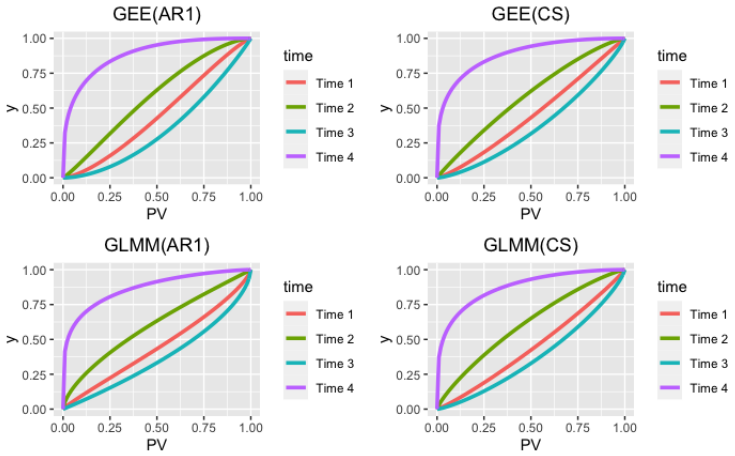

Figure 4.22. ROC Curves by Time, Fully Generated Data, Example 4 ($n = 15$)

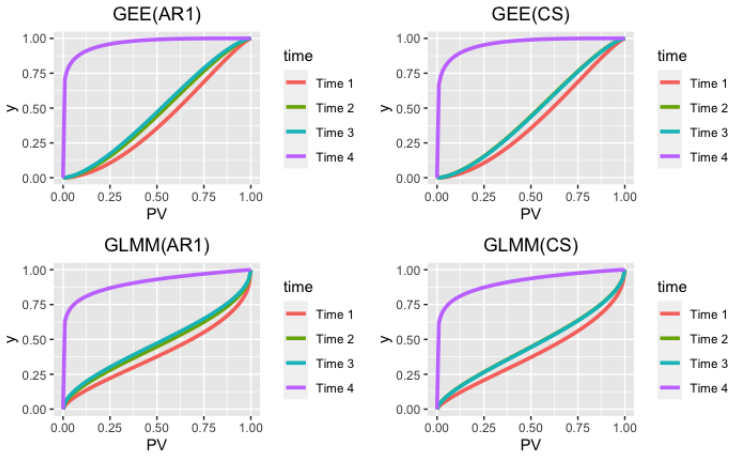

Figure 4.23. ROC Curves by Time, Fully Generated Data, Example 4 ($n = 100$)

Table 4.19. Area Under ROC Curve (AUC), Fully Generated Data, Example 4 (large parameters, "late" jump)

Time/Parameter	GEE(AR1) (n=15/n=100)	GEE(CS) (n=15/n=100)	GLMM(AR1) (n=15/n=100)	GLMM(CS) (n=15/n=100)
Time 1 (Int)	0.45 / 0.41	0.45 / 0.41	0.45 / 0.39	0.45 / 0.39
Time 2	0.59 / 0.46	0.59 / 0.46	0.60 / 0.46	0.61 / 0.46
Time 3	0.35 / 0.48	0.37 / 0.46	0.37 / 0.48	0.37 / 0.45
Time 4	0.88 / 0.97	0.88 / 0.96	0.87 / 0.91	0.87 / 0.91

Table 4.19 is referenced in Figure 4.24 below.

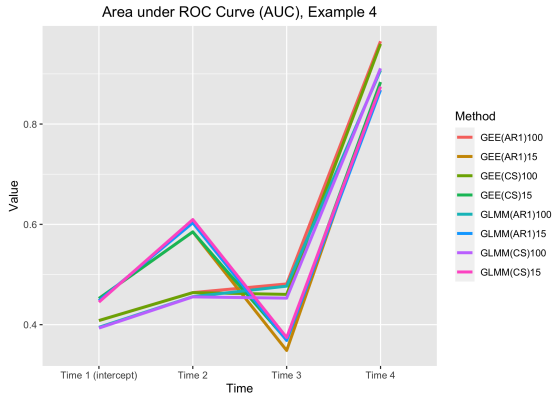

Figure 4.24. Area Under ROC Curve (AUC), Fully Generated Data, Example 4 (large parameters, "late" jump)

4.7.2 Partially Generated Data

With the partially generated data we see a contradicting result: GEE models show the highest AUC at time 2 yet the GLMM model shows it at time 3, by a very small margin. Initially, the AUC values start at about 0.6, with not much separation between treatments. Then, from time 0 to time 2, there is a large increase in values, to 0.85 for GLMM and almost 0.9 for GEE models. At time 3, the AUC increases very slightly for GLMM where, at the same time, the AUC decreases for the GEE models. Then, both decrease at time 4, leaving their highest AUC values at time 2 and time 3 for GEE and GLMM, respectively. This contradiction in the magnitude of separation between active treatment and placebo is not present when analyzing the time of largest jump in the separation. Across all models, the largest change in treatment difference happens at time 1.

The partially generated data ROC curves by time are referenced in Figure 4.25 below. Each graph represents the GEE(AR1), GEE(CS), GEE(ind), and GLMM models, respectively.

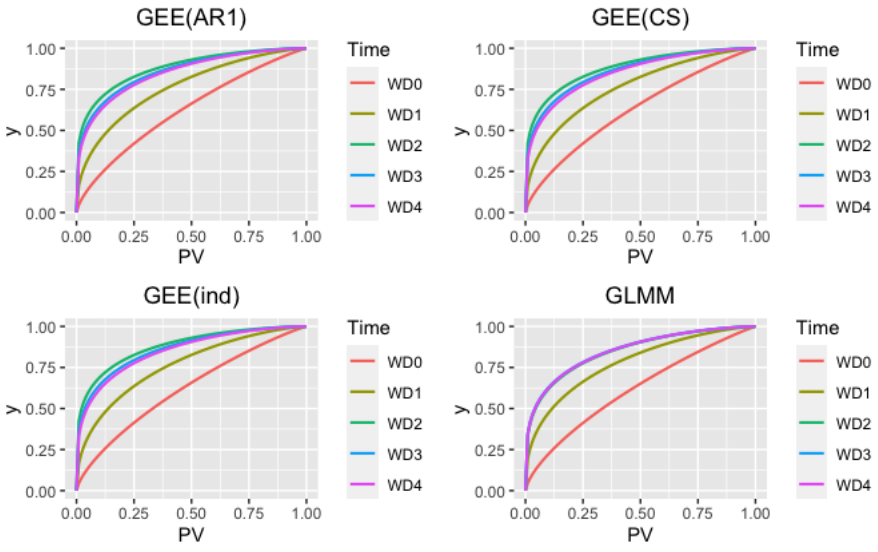

Figure 4.25. ROC Curves by Time, Partially Generated Data

Table 4.20. Area Under ROC Curve (AUC), Partially Generated Data

Time/Parameter	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
Time 0 (Int)	0.62	0.62	0.62	0.62
Time 1	0.76	0.76	0.76	0.78
Time 2	0.88	0.88	0.88	0.85
Time 3	0.86	0.86	0.86	0.85
Time 4	0.85	0.85	0.85	0.85

Table 4.20 is referenced in Figure 4.26 below.

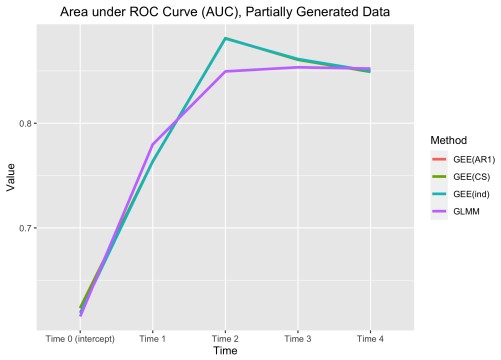

Figure 4.26. Area Under ROC Curve (AUC), Partially Generated Data

4.7.3 Real World Data

In the application of our method to the real world data, we see a very minor difference in AUC results, particularly between GEE and GLMM models as a whole. From time 0 up to time 2 they perform almost identically, with the GLMM model showing an increase even higher in AUC at time 3 as opposed to the GEE models. At time 4, we see a decrease in the GLMM model AUC value, therefore peaking at time 3. The GEE model AUCs, however, keep rising and reach around the same value the GLMM had at the previous time, peaking at time 4. Even though the maximum separation between treatments is not consistent across models, the largest jump in treatment difference is seen at time 1 consistently. This result is what we see in all fully generated and partially generated examples, as well.

The real world data ROC curves by time are referenced in Figure 4.27 below. Each graph represents the GEE(AR1), GEE(CS), GEE(ind), and GLMM models, respectively.

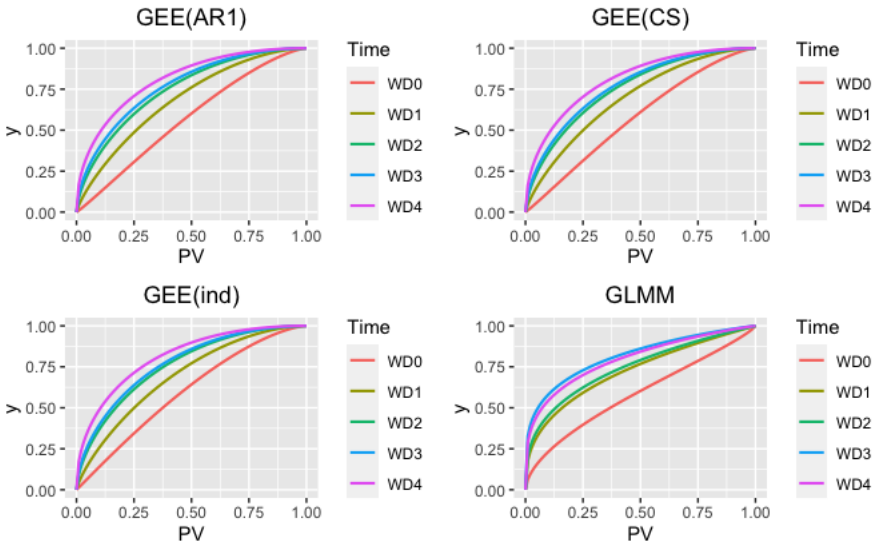

Figure 4.27. ROC Curves by Time, Real World Data

Table 4.21. Area Under ROC Curve (AUC), Real World Data

Time/Parameter	GEE(ind)	GEE(AR1)	GEE(CS)	GLMM
Time 0 (Int)	0.60	0.57	0.57	0.58
Time 1	0.69	0.69	0.69	0.72
Time 2	0.76	0.75	0.75	0.75
Time 3	0.77	0.77	0.77	0.82
Time 4	0.81	0.81	0.81	0.80

Table 4.21 is referenced in Figure 4.28 below.

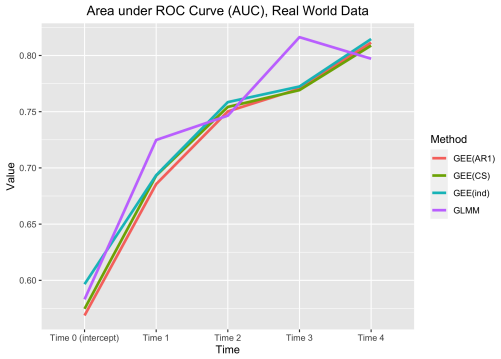

Figure 4.28. Area Under ROC Curve (AUC), Real World Data

CHAPTER FIVE

Multiple Active Treatment Classification

5.1 Introduction

Previously, when analyzing the data, we had a placebo and active treatment group. From there, we were able to get down to the one-dimension case by transforming the original response variable into a placement value that included information from both treatment types. Then, we were able to perform the Beta regression, Beta reconstruction, and classification as we would when having only one group. Now, we considered the case when there is a placebo and more than one active treatment in an experiment. For simplicity, we focused on the two active treatment and placebo case.

With this approach, we can take our method a step further, not only being able to find optimal time periods for an active treatment, but also comparing between more than one active treatment at those periods to see if there is a significant difference and significant separation.

5.2 Methods

5.2.1 Two-Sample Kolmogorov-Smirnov (KS) Test

The Kolmogorov-Smirnov goodness-of-fit test is a nonparametric test to compare a distribution, either to a hypothesized one, or to another distribution. Here, we use it to compare two placement value distributions to each other, namely the distributions resulting from comparing two active treatments to the placebo control. As opposed to similar tests like Wilcoxon and the Mann-Whitney, here we compare full distribution functions and not just medians or means (Dodge, 2008).

The use of the KS test is advised when

- distribution means or medians are similar but differences in variance/symmetry are suspected.
- sample sizes are small.
- differences between distributions are suspected to affect only the upper or lower end of distributions.
- the shift between two distributions is hypothesized to be small but systematic.
- two samples are of unequal size. (Engmann and Cousineau, 2011)

5.2.2 Two-Sample Anderson-Darling (AD) Test

The Anderson-Darling test was first introduced in (Anderson and Darling, 1952) and generalized to a k-sample version in (Scholz and Stephens, 1987). A very similar test to the KS, this is a "weighted" version of it.

One of the motivations for using this test instead of the KS test would be if we have fatter tails in our placement value distributions. Another scenario where we would include this method for our analysis would be when there are very small differences between the distributions of the populations being compared, especially if sample sizes are large (Engmann and Cousineau, 2011).

5.3 Applications

5.3.1 Kolmogorov-Smirnov Test

Applying the KS test is relatively straightforward: we have two samples, in this case those would be set of placement values coming from each active treatment, which get compared to see at which point there exists maximum separation between their respective CDFs. This separation is quantified into the KS statistic, Δ, which is

$$\Delta_{m,n} = \sqrt{\frac{mn}{m+n}} \sup_x |F_m(x) - F_n(x)|.$$

The null hypothesis that $F_m(x)$ and $F_n(x)$ come from the same distribution is rejected if Δ is larger than the critical value Δ_α at a given α.

The KS statistic, in terms of our research methods, is the "maximum distance" the empirical CDF of the active placement values, or the observed ROC curve, is from the diagonal line where the AUC = 0.5. This distance is also known as Youden's J statistic, given by J = sensitivity + specificity − 1 (Youden, 1950). An illustration of this is seen in Figure 5.1 below, where the red line represents the empirical ROC curve and the dashed line represents the ROC when the AUC = 0.5.

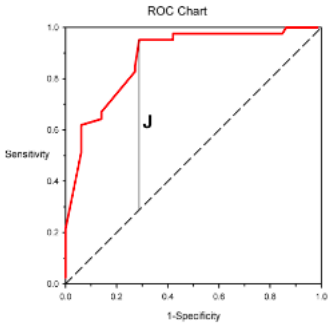

Figure 5.1. Youden's J statistic

5.3.2 Anderson-Darling Test

The AD statistic is

$$AD = \frac{1}{mn} \sum_{i=1}^{m+n} M_i (Z_{(m+n-mi)})^2 \frac{1}{iZ_{(m+n-i)}}$$

where $Z_{(m+n)}$ represents the combined and ordered samples $X_{(m)}$ and $Y_{(n)}$, of size m and n respectively, and M_i represents the number of observations in $X_{(m)}$ that are equal to or smaller than the ith observation in $Z_{(m+n)}$. The null hypothesis that $X_{(m)}$ and $Y_{(n)}$ come from the same distribution is rejected if AD is larger than the corresponding α-level critical value.

5.4 Extension of Two Active Treatment Classification

There are k-sample versions of the Kolmogorov-Smirnov and the Anderson-Darling tests, as well. If we were presented with data with more than two active treatments being tested against a placebo, we could, potentially, use our method in conjunction with these multiple sample tests. Ideally, we would find placement values for all treatments in relation to the placebo control, then compare CDFs for all of the k active treatments to see if there is any separation. If there is, we would then use the two-sample analogous methods of these tests to see where the separation lies.

5.5 Conclusion

In conclusion, we believe our classification method can be very useful to many different applications in placebo-controlled clinical trials. We extensively explored the placebo-active treatment situation and developed a specific way of finding separation between the treatments, however, we also found a way to generalize our approach to more than one active treatment, if necessary. Depending on the specific nature of the data and situation, various tests and specifications can be applied to carry out this analysis.

CHAPTER SIX

Conclusions, Limitations, and Future Research

6.1 Conclusions

This dissertation was done to connect the concepts of placement values and ROC curves with Beta regression to create a method that can be used in clinical trials when longitudinal data is present and classification is needed between treatments. Additionally, we did not set any restrictions on the distribution of the endpoints analyzed to make this classification technique flexible and applicable to more scenarios.

The usefulness of our research lies in giving a biostatistician an extra visualization tool for their arsenal when presented with longitudinal data in a clinical trial. In short, we provided a classification method for multiple treatments that can be applied to a wide range of endpoints which will account for across-time correlation and help distinguish at which time point(s) a treatment is most effective.

The results seen from our analysis were promising. Classification resulted as expected in all fully generated dataset examples, while our real-world and partially generated datasets gave us a chance to apply the method to more realistic data and arrive at similar conclusions.

The method had trouble in the "early" jump, large ($n = 100$) scenario posed with the fully generated data, incorrectly classifying where the maximum separation between treatments was present with all four Beta regression methods. However, Beta regression has certain limitations which this scenario exploited, so that result was expected (Hunger et al., 2011). In terms of finding the correct time point at which the jump in separation occurred, however, our method proved to correctly classify in all scenarios.

When applied to the real datasets, the same discrepancy was seen. In the early time periods where the different treatment observations had more overlap with each other, separation was relatively easy to detect. However, where more separation between the treatments started being seen, almost to the point of no overlap, our method struggled to be consistent detecting maximum separation across models. However, in this scenario, visually inspecting the data initially would stop us from ever using our method given the separation between treatments is so apparent. In terms of finding the time at which a jump in treatment separation occurs, as with the fully generated data, our method was able to perform consistently.

This conclusion led us to believe that, if we develop a transformation of the placement values such that they are even more separated from the (0,1) endpoints (say, $(PV + 0.5)/2$, where we would limit all values to the (0.25, 0.75) range), we can strengthen our classification method, in terms of finding the time of maximum separation between treatments, and it can be more widely applied to longitudinal data. That being said, we saw very promising results finding the point of largest jump in separation between groups with our method's present performance.

6.2 Limitations

6.2.1 Data Availability and Variability

In searching for appropriate data for our research, we found it a bit difficult and time consuming to locate useful datasets that fit our needs. Mostly, data was small in terms of the number of patients or it only had a couple of observations per patient (pre and post, usually). For this reason, we decided to add patients for our partially generated and real world data. Ideally, we would have preferred to have more data options from where to choose, especially if we could have had "small" and "large" real datasets for comparison of our method in both scenarios. We could have

also varied the amount of observations per patient to see if that had an effect on the accuracy of our method's classification (e.g. three vs. five observations).

6.3 Future Research

6.3.1 Different Transformations on Placement Values

We saw a limitation in our research was Beta regression not being able to adequately handle placement values at the (0,1) endpoints or near them. Therefore, a future project would include trying various transformation techniques, either from the literature or created by us, to see if using them creates a more accurate classification method.

6.3.2 k-Treatment Analysis

While touched upon in this dissertation and our research, a k-treatment classification analysis was not able to be fully done. With what we have presented, however, it seems like a natural progression of what could be done next by building on what we showed here. Although not as common as an active treatment vs. placebo longitudinal comparison, it would be beneficial to the advancement of clinical trials to have more research done on the multiple active treatment vs. placebo longitudinal study scenario (Kiefer, 1959).

6.3.3 Response Distribution Variation

In this dissertation, we applied our classification method to three different situations: Beta data (fully generated), Normal data (partially generated), and non-specified data (real world). Ideally, given our nonparametric placement value approach, we would like to apply our method in multiple controlled scenarios where we can test if and how it compares across all possible response distributions (Exponential family).

6.3.4 Classification Accuracy Comparison

After seeing our classification method in action, we can now turn to other methods to compare how "well" ours performed. Since we use both GEE and GLMM, a quasi-likelihood and maximum likelihood method, we can conduct these comparisons in various ways. To do this, some of the already included literature must be further analyzed while new topics must be researched, as well.

6.3.5 Bayesian Analysis using Youden's J statistic

Used as a way to quantify how much separation is present between two placement value distributions, Youden's J can also be used in a Bayesian setting. We can set its probability distribution as a prior, treating it as a random variable, and create credible intervals with this information. This can be a way to connect our research to the Bayesian side of statistics and, possibly, use more information for classifying between treatments.

APPENDICES

APPENDIX A

Additional Details

A.1 Abbreviations

- PDF: Probability density function
- CDF: Cumulative distribution function
- SE: Standard error
- LMM: Linear Mixed Model
- GLM: Generalized Linear Model
- GEE: Generalized Estimating Equations
- GLMM: Generalized Linear Mixed Model
- LNMVB: Libby and Novick Multivariate Beta
- SLMVB: Sarmanov-Lee Multivariate Beta
- AIC: Akaike Information Criterion
- BIC: Bayesian Information Criterion
- AR1: Auto-regressive(1)
- CS: Compound Symmetric; Exchangeable
- ind: Independent
- UNSTR: Unstructured
- ROC: Receiver Operating Characteristic [curve]

- AUC: Area under the [ROC] curve
- PV: Placement Value
- Ex: Example
- Int: Intercept
- ML: Maximum Likelihood
- QL: Quasi-Likelihood
- KS: Kolmogorov-Smirnov [test]
- AD: Anderson-Darling [test]

A.2 Beta Distribution Examples

Here, we mention just some of the useful properties and shapes this distribution can take depending on its parameter values and their magnitude, both independently and in relation to each other.

- If $\alpha > 1$ and $\beta = 1$, the pdf is strictly increasing, however, if $\alpha = 1$ and $\beta > 1$, it is now strictly decreasing.
- When $\alpha, \beta < 1$, the pdf is U-shaped.
- When $\alpha, \beta > 1$, the pdf is unimodal.
- Now, if $\alpha = \beta$, the pdf is centered at 1/2, with it becoming more concentrated about the mean as α increases. The special case $\alpha = \beta = 1$ yields the Uniform(0, 1) distribution.

Examples of Beta distribution flexibility are seen in Figure A.1 below, where the following are shown:

- $\alpha = 1, \beta = 3$ (Strictly decreasing)
- $\alpha = \beta = 1$ (Uniform)
- $\alpha = \beta = 10$ (Centered with "large" parameter values)
- $\alpha = \beta = 2$ (Centered with "small" parameter values)
- $\alpha = 3, \beta = 1$ (Strictly increasing)
- $\alpha = 0.3, \beta = 0.7$ (U-shaped)
- $\alpha = 3, \beta = 7$ (Unimodal)

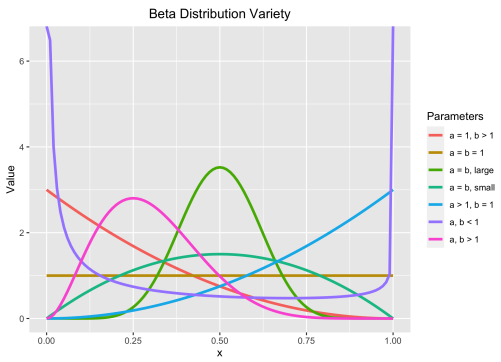

Figure A.1. Beta Distribution variation

A.3 Correlation Structures

We highlight some properties of the correlation structures used for our research.

A.3.1 Independence

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Under this structure, the estimates are identical to the ML estimates obtained by treating all observations within and between groups as independent (Agresti, 2015).

A.3.2 Auto Regressive (AR1)

$$\begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$$

This correlation structure takes proximity into account: the closer to each other (in time) the observations are, the more correlated. Despite its wide use, the AR1 structure often poorly gauges within-subject correlations that decay at a slower or faster rate than required by this specific model (Simpson et al., 2010).

A.3.3 Exchangeable / Compound Symmetry (CS)

$$\begin{bmatrix} 1 & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho \\ \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & 1 \end{bmatrix}$$

With this correlation structure, the lag is not taken into account: the same correlation is assumed no matter how close or far from each other the observations are. This is a useful structure for dealing with clustered data with no time ordering, such as when using ANOVA for analysis.

A.3.4 Untructured (UNSTR)

$$\begin{bmatrix} 1 & \rho_{12} & \rho_{13} & \rho_{14} \\ \rho_{12} & 1 & \rho_{23} & \rho_{24} \\ \rho_{13} & \rho_{23} & 1 & \rho_{34} \\ \rho_{14} & \rho_{24} & \rho_{34} & 1 \end{bmatrix}$$

where $\rho_{jk} = Corr(Y_{ij}, Y_{ik})$ for the i_{th} subject at times j and k.

This type of correlation structure makes sense when repeated measurements are unequally spaced and variances differ in no recognizable pattern. Since all of the variances and covariances are unique, this is the structure that uses up the most degrees of freedom, so it should only be used if any other fails in fitting the data.

A.4 Nataf and Vorechovsky Example

Here we show several of the R functions from (Hein, 2019) that we used for data creation. They were as follows:

(1) *alpha.beta()*: reparameterized a Beta distribution. When given μ and ϕ parameters, it outputs a and b. This function is used in *p.tilda()*.

(2) *p.tilda()*: given the means and standard deviation of both Beta distributions used, it solves the non-linear equation

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{H_i - \mu_i}{\sigma_i}\frac{H_j - \mu_j}{\sigma_j}\varphi(\tilde{H}_i, \tilde{H}_j, \tilde{\rho}_{ij})d\tilde{H}_i d\tilde{H}_j - \rho_{ij} = 0$$

for $\tilde{\rho}_{ij}$.

(3) *make.corr()*: given the correlation coefficient, the number of repeated measures, the means and standard deviation, and the correlation structure type, this function creates the correlation matrix for the dataset we want to generate using the *p.tilda()* function.

(4) *Vorechovsky()*: this function simulates the Beta responses using Vorechovsky's method, which includes using Cholesky decomposition on the correlation matrix created with *make.corr()*.

(5) *long()*: this function converts the data into long format and makes it easier to analyze.

(6) ***data_creation()***: we created this function to aggregate all of the steps above and simplify our data generation process. Given the sample size, mean of each treatment, standard deviation, correlation coefficient, time of jump (early for $t = 2$, late for $t = 4$), and correlation structure (AR1, CS, or Unstructured), a dataset with active vs. placebo treatments having four repeated measures is created.

A simple example is illustrated here. When the code *data_creation(n=50, mu1=0.1, mu2=0.2, phi=0.1, sd=0.01, jump="early", cor_str="CS")* is used, Figure A.2 is the output:

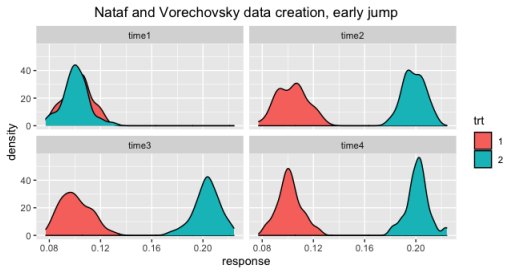

Figure A.2. Vorechovsky's Method for $n = 50$ and CS correlation structure with a jump at $t = 2$

Now, when "late" is specified in the code for *jump*, Figure A.3 is the result:

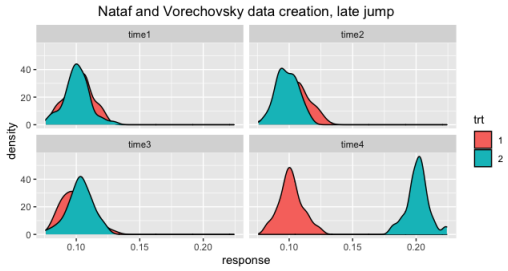

Figure A.3. Vorechovsky's Method for $n = 50$ and CS correlation structure with a jump at $t = 4$

The complete code for these functions is included in Appendix B.

APPENDIX B

Selected R Code

B.1 Nataf and Vorechovsky Example

```
library(tidyverse)


#### Generating data
#Function for data creation and analysis
data_creation <- function(n, mu1, mu2, phi, sd, jump, cor_str) {
  ###################################
  # Functions extracted from dissertation
  ###################################

  # Given mu and variance of beta distribution,
    #parameters of beta distribution
  # input: mu and variance
  # output: alpha and beta
  alpha.beta <- function(m, v) {
    beta <- (m*(1 - m)^2 - v + v*m) / v
    alpha <- beta*m / (1 - m)
    return(c(alpha, beta))
  }

  # ------------------------------------------------------

  # Support function for Vorechovsky's method.
```

```
  #Determines correlation matrix using Nataf's transformation
if (!require("nleqslv")) install.packages("nleqslv")
library("nleqslv")

if (!require("pracma")) install.packages("pracma")
library(pracma)

p.tilda <- function(p.target, mu1, mu2, s) {
  a1 <- alpha.beta(mu1, s^2)[1]
  b1 <- alpha.beta(mu1, s^2)[2]
  a2 <- alpha.beta(mu2, s^2)[1]
  b2 <- alpha.beta(mu2, s^2)[2]
  p.tmp <- function(p) {
    f <- function(u1, u2) {
      tmp1 <- (qbeta(pnorm(u1), shape1 = a1, shape2 = b1) - mu1)/
          s
      tmp2 <- (qbeta(pnorm(u2), shape1 = a2, shape2 = b2) - mu2)/
          s
      tmp3 <- 1 / (2*pi*sqrt(1 - p^2))*exp(-1 / (2*(1 - p^2))*
          (u1^2 - 2*p*u1*u2 + u2^2))
      tmp1*tmp2*tmp3
    }
    integral2(f, -10, 10, -10, 10)$Q - p.target
  }
  nleqslv(0.5, p.tmp, method = 'Newton')$x
}
# ------------------------------------------------------
```

```
# Function to make correlation matrix for Vorechovsky's method
# input: rho, number of repeated measures,
  #type of correlation - CS, AR1, UNSTR
# output: t x t correlations matrix that has been
  #Nataf transformed
make.corr <- function(rho, t, mu.vector, sd, type) {
  mat <- diag(t)

  if(toupper(type) == 'CS') {
    for (i in 1:t) {
      for (j in i:t) {
        if (i != j) mat[i, j] <- mat[j, i] <- rho
      }
    }
  }
  if(toupper(type) == 'AR1') {
    for (i in 1:t) {
      for (j in i:t) {
        if (i != j) mat[i, j] <- mat[j, i] <- rho^(abs(i - j))
      }
    }
  }
  if(toupper(type) == 'UNSTR') {
    for (i in 1:t) {
      for (j in i:t) {
        if (i != j) mat[i, j] <- mat[j, i] <- runif(1)
```

```
      }
    }
  }
  for (i in 1:t) {
    for (j in i:t) {
      if (i != j) {
        mat[i, j] <- mat[j, i] <- p.tilda(mat[i, j],
        mu.vector[i], mu.vector[j], sd)
      }
    }
  }
  return(mat)
}

# ------------------------------------------------------

# Function to simulate beta responses for Vorechovsky's method
# inputs: n - num subjects; t - num repeated measures;
  #vector of means; common sd; target correlation matrix
# output: n x t matrix - rows are subjects and columns
  #are repeated measures
Vorechovsky <- function(n, t, mu.vector, sd, corr.mat) {
  U <- chol(corr.mat)
  tmp <- matrix(rnorm(n*t), nrow = n, ncol = t)
  mat <- tmp %*% U

  for (i in 1:t) {
```

```
    a <- alpha.beta(mu.vector[i], sd^2)[1]
    b <- alpha.beta(mu.vector[i], sd^2)[2]
    mat[, i] <- qbeta(pnorm(mat[, i]), shape1 = a, shape2 = b)
  }
  return(mat)
}

# ------------------------------------------------------

# Function to format simulated data into long format
# inputs: matrix - where each column is a repeated measure
#(groups need to be inputed as separate matrices)
# output: data frame in long format -
  #colnames <- subj, trt, time, response
if (!require("reshape2")) install.packages("reshape2")
library(reshape2)

long <- function(mat.1, mat.2) {
  df <- data.frame(mat.1)
  colnames(df) <- paste0('time', seq(1, length(mat.1[1, ]), 1))
  df$subj <- seq(1, length(mat.1[, 1]), 1)
  df$trt <- rep(1, length(mat.1[, 1]))
  long.df <- melt(df, id.vars = c('subj', 'trt'),
                  variable.name = c('time'),
                  value.name = c('response'))
  long.df <- long.df[order(long.df$subj), ]
```

```
  if(!missing(mat.2)) {
    df.2 <- data.frame(mat.2)
    colnames(df.2) <- paste0('time',
      seq(1, length(mat.2[1, ]), 1))
    df.2$subj <- seq(1, length(mat.2[, 1]), 1)
    df.2$trt <- rep(2, length(mat.2[, 1]))
    long.df.2 <- melt(df.2, id.vars = c('subj', 'trt'),
                      variable.name = c('time'),
                      value.name = c('response'))
    long.df.2 <- long.df.2[order(long.df.2$subj), ]
    long.df <- rbind(long.df, long.df.2)
  }
  long.df$trt <- factor(long.df$trt)

  return(long.df)
}



#---------------------- Added for simplicity -----------------
#-------------------------------------------------------------
# making correlation matrix
cor.mat <- make.corr(phi, 4, c(mu1, mu1, mu1, mu1), sd, cor_str)

# making dataset
set.seed(12345)

if (jump == 'early') {
```

```
    dat <- Vorechovsky(n, 4, c(mu1, mu1, mu1, mu1), sd, cor.mat)
    dat2 <- Vorechovsky(n, 4, c(mu1, mu2, mu2, mu2), sd, cor.mat)
  }

  if (jump == 'late') {
    dat <- Vorechovsky(n, 4, c(mu1, mu1, mu1, mu1), sd, cor.mat)
    dat2 <- Vorechovsky(n, 4, c(mu1, mu1, mu1, mu2), sd, cor.mat)
  }
  long.dat <- long(dat, dat2)

  long.dat
}

# n=50, early
hein_example <- data_creation(n=50, mu1=0.1, mu2=0.2, phi=0.1,
    sd=0.01, jump="early", cor_str="CS")

hein_example %>%
  ggplot(aes(response, fill = trt)) +
  geom_density() +
  facet_wrap(~ time) +
  ggtitle("Nataf and Vorechovsky data creation, early jump") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
# n=50, late
hein_example <- data_creation(n=50, mu1=0.1, mu2=0.2, phi=0.1,
    sd=0.01, jump="late", cor_str="CS")


hein_example %>%
  ggplot(aes(response, fill = trt)) +
  geom_density() +
  facet_wrap(~ time) +
  ggtitle("Nataf and Vorechovsky data creation, late jump") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
LinkFun <- function(arg) {log(arg / (1 - arg))}
InvLink <- function(arg) {exp(arg) / (1 + exp(arg))}
InvLinkDeriv <- function(arg) {exp(arg) / (1 + exp(arg))^2}
VarFun <- function(arg) {arg*(1 - arg)}
FunList <- list(LinkFun, VarFun, InvLink, InvLinkDeriv)


alpha.beta <- function(m, v) {
  beta <- (m*(1 - m)^2 - v + v*m) / v
  alpha <- beta*m / (1 - m)
  return(c(alpha, beta))
}

beta.glmm <- function(link = 'logit') {
  stats <- make.link(link)
  log_dens <- function(y, eta, mu_fun, phis, eta_zi) {
    phi <- exp(phis)
    mu <- mu_fun(eta)
    comp.1 <- lgamma(phi) - lgamma(mu*phi) - lgamma((1 - mu)*phi)
    comp.2 <- (mu*phi - 1)*log(y) + ((1 - mu)*phi - 1)*log(1 - y)
    out <- comp.1 + comp.2
    attr(out, "mu_y") <- mu
    out
  }
  structure(list(family = "Beta", link = stats$name,
  linkfun = stats$linkfun, linkinv = stats$linkinv,
```

```
  log_dens = log_dens), class = "family")
}

# input: parm_beta and Phi
# output: list with ROC
out_parm <- function(parm_beta, Phi, method_SE,
 plot_title = "ROC") {
  parm_beta <- parm_beta %>% unname()
  Phi <- Phi %>% unname()
  mu <- 1 /(1 + exp(-parm_beta))
  a0 <- Phi*mu
  b0 <- Phi * (1 - mu)

  expected_value = a0 / (a0 + b0)
  AUC = 1 - expected_value
  SE <- method_SE
  xx <- c(0:100)/100
  y <- pbeta(xx,a0, b0, lower.tail = TRUE)
  #plot(xx, y, "l", ylim = c(0, 1), main = plot_title)

  list("Beta Parameter" = parm_beta, "Standard Error" = SE,
  "Scale Parameter (phi)" = Phi, "a0" = a0, "b0" = b0,
  "Expected Value" = expected_value, "AUC" = AUC)
}
```

```
#Function for GEE and GLMM Beta regression
beta_regression <- function(data_PV, response, explanatory, id,
 cor_str = "independence") {
  ##############
  ### needed to use variables in geem and mixed_model,
  #creates "formula" objects
  ##############
  nullmodel <- reformulate("1", response = response)
  fullmodel <- reformulate(c("1", explanatory), response = response)
  remodel <- reformulate(paste("1", id, sep="|"))

  ##############
  ### creating beta regression output for both models
  ##############
  modALL_gee <- geem(fullmodel, id = data_PV[[id]], data = data_PV,
    family = FunList, corstr = cor_str)
  modALL_mix <- mixed_model(fullmodel, random = remodel, data_PV,
    beta.glmm, n_phis = 1, iter_EM = 0)

  summary_gee <- summary(modALL_gee)
  summary_mix <- summary(modALL_mix)

  ##############
  ### Beta reconstruction for all time points,
  #using both GEE and GLMM
  ##############
```

```
################# full gee output
#Test for time=time0 (Use the intercept)
parm_beta <- modALL_gee$beta[1]
Phi = modALL_gee$phi * 10
method_se <- summary_gee$se.model[1]
out_parm(parm_beta,Phi, method_se, "Time 0 ROC, GEE")
  -> gee_beta_phi_0

#Test for time = time1
parm_beta <- modALL_gee$beta[1] + modALL_gee$beta[2]
Phi = modALL_gee$phi * 10
method_se <- summary_gee$se.model[2]
out_parm(parm_beta,Phi, method_se, "Time 1 ROC, GEE")
  -> gee_beta_phi_1

#Test for time = time2
parm_beta <- modALL_gee$beta[1] + modALL_gee$beta[3]
Phi = modALL_gee$phi * 10
method_se <- summary_gee$se.model[3]
out_parm(parm_beta,Phi, method_se, "Time 2 ROC, GEE")
  -> gee_beta_phi_2

#Test for time = time3
parm_beta <- modALL_gee$beta[1] + modALL_gee$beta[4]
Phi = modALL_gee$phi * 10
method_se <- summary_gee$se.model[4]
out_parm(parm_beta,Phi, method_se, "Time 3 ROC, GEE")
```

```
  -> gee_beta_phi_3

#Test for time = time4
parm_beta <- modALL_gee$beta[1] + modALL_gee$beta[5]
Phi = modALL_gee$phi * 10
method_se <- summary_gee$se.model[5]
out_parm(parm_beta,Phi, method_se, "Time 4 ROC, GEE")
  -> gee_beta_phi_4



################## full glmm output
#Test for time = time0 (Use the intercept)
parm_beta <- modALL_mix$coefficients[1]
Phi = modALL_mix$phis
method_se <- summary_mix$coef_table[1,2]
out_parm(parm_beta,Phi, method_se, "Time 0 ROC, GLMM")
  -> glmm_beta_phi_0

#Test for time = time1
parm_beta <- modALL_mix$coefficients[1] + modALL_mix$coefficients[2]
Phi = modALL_mix$phis
method_se <- summary_mix$coef_table[2,2]
out_parm(parm_beta,Phi, method_se, "Time 1 ROC, GLMM")
  -> glmm_beta_phi_1

#Test for time = time2
parm_beta <- modALL_mix$coefficients[1] + modALL_mix$coefficients[3]
```

```
Phi = modALL_mix$phis
method_se <- summary_mix$coef_table[3,2]
out_parm(parm_beta,Phi, method_se, "Time 2 ROC, GLMM")
  -> glmm_beta_phi_2

#Test for time = time3
parm_beta <- modALL_mix$coefficients[1] + modALL_mix$coefficients[4]
Phi = modALL_mix$phis
method_se <- summary_mix$coef_table[4,2]
out_parm(parm_beta,Phi, method_se, "Time 3 ROC, GLMM")
  -> glmm_beta_phi_3

#Test for time = time4
parm_beta <- modALL_mix$coefficients[1] + modALL_mix$coefficients[5]
Phi = modALL_mix$phis
method_se <- summary_mix$coef_table[5,2]
out_parm(parm_beta,Phi, method_se, "Time 4 ROC, GLMM")
  -> glmm_beta_phi_4

gee_results <- list("Time 0" = gee_beta_phi_0,
  "Time 1" = gee_beta_phi_1,
                    "Time 2" = gee_beta_phi_2,
                  "Time 3" = gee_beta_phi_3,
                    "Time 4" = gee_beta_phi_4)
glmm_results <- list("Time 0" = glmm_beta_phi_0,
  "Time 1" = glmm_beta_phi_1,
                     "Time 2" = glmm_beta_phi_2,
```

```
                        "Time 3" = glmm_beta_phi_3,
                        "Time 4" = glmm_beta_phi_4)

  list("GEE Results" = gee_results, "GLMM Results" = glmm_results)
}
```

BIBLIOGRAPHY

Agresti, A. (2013). *Categorical Data Analysis*. John Wiley & Sons.

Agresti, A. (2015). *Foundations of Linear and Generalized Linear Models*. John Wiley & Sons.

Alonzo, T. A. and Sullivan Pepe, M. (2002). Distribution-free roc analysis using binary regression techniques. *Biostatistics*, 3(3):421–432.

Anderson, T. W. and Darling, D. A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193–212.

Cai, T. (2004). Semi-parametric ROC regression analysis with placement values. *Biostatistics*, 5(1):45–60.

Carrière, I. and Bouyer, J. (2002). Choosing marginal or random-effects models for longitudinal binary responses: application to self-reported disability among older persons. *BMC Med Res Methodol*, 2(15).

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates.

Cribari-Neto, F. and Zeileis, A. (2010). Beta regression in r. *Journal of Statistical Software*, 34(2):1–24.

Cui, J. and Quian, G. (2007). Selection of working correlation structure and best model in gee analyses of longitudinal data. *Communications in Statistics - Simulation and Computation*, 36(5):987–996.

Dodge, Y. (2008). *Kolmogorov–Smirnov Test*, pages 283–287. Springer New York, New York, NY.

Engmann, S. and Cousineau, D. (2011). Comparing distributions: The two-sample anderson-darling test as an alternative to the kolmogorov-smirnoff test. *Journal of Applied Quantitative Methods*, 6(3):1–17.

Ferrari, S. and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, 31(7):799–815.

Fubini, G. (1907). Sugli integrali multipli. *Roma Accademia Lincei Rendiconti*, 16:608–614.

Hajian-Tilaki, K. (2013). Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation. *Caspian J Intern Med*, 4(2):627–635.

Hanley, J. and Hajian-Tilaki, K. (1997). Sampling variability of nonparametric estimates of the areas under receiver operating characteristic curves: An update. *Academic Radiology*, 4(1):49–58.

Hanley, J. and Mcneil, B. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143:29–36.

Hardin, J. and Hilbe, J. (2003). *Generalized Estimating Equations*. Chapman and Hall.

Hein, N. A. (2019). Beta regression models for repeated-measures data analysis. *Theses and Dissertations*, page 381.

Hogg, R. V., Mckean, J. W., and Craig, A. T. (2013). *Introduction to Mathematical Statistics*. Pearson Education Inc.

Hu, F. B., Goldberg, J., Hedeker, D., Flay, B. R., and Pentz, M. A. (1998). Comparison of population-averaged and subject-specific approaches for analyzing repeated binary outcomes. *American Journal of Epidemiology*, 147(7):694–703.

Hunger, M., Baumert, J., and Holle, R. (2011). Analysis of sf-6d index data: is beta regression appropriate? *Value in health : the journal of the International Society for Pharmacoeconomics and Outcomes Research*, 14(5):759–767.

Hunger, M., Döring, A., and Holle, R. (2012). Longitudinal beta regression models for analyzing health-related quality of life scores over time. *BMC Med Res Methodol*, 12(144).

Kiefer, J. (1959). K-sample analogues of the kolmogorov-smirnov and cramer-v. mises tests. *The Annals of Mathematical Statistics*, 30(2):420–447.

Lachenbruch, P., McCullagh, P., and Nelder, J. (1990). Generalized linear models. *Biometrics*, 46:1231.

Liang, K. and Zeger, S. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22.

Liu, P. and Der Kiureghian, A. (1986). Multivariate distribution models with prescribed marginals and covariances. *Prob Eng Mech*, 1(2):105–111.

McDaniel, L. S., Henderson, N. C., and Rathouz, P. J. (2013). Fast pure r implementation of gee: Application of the matrix package. *The R journal vol. 5,1*, pages 181–187.

Meaney, C. and Moineddin, R. (2014). A monte carlo simulation study comparing linear regression, beta regression, variable-dispersion beta regression and fractional logit regression at recovering average difference measures in a two sample design. *BMC Med Res Methodol*, 14(14).

Nataf, A. (1962). Détermination des distributions de probabilités dont les marges sont donnés. *CR Acad Sci*, 225:42–43.

Pepe, M. (1997). A regression modeling framework for roc curves in medical diagnostic testing. *Biometrika*, 84:595–608.

Pepe, M. (1998). Three approaches to regression analysis of receiver operating characteristic curves for continuous test results. *Biometrics*, 54:124–35.

Pepe, M. (2000). An interpretation for the roc curve and inference using glm procedures. *Biometrics*, 56:352–359.

Scholz, F. W. and Stephens, M. A. (1987). K-sample anderson-darling tests. *Journal of the American Statistical Association*, 82(399):918–924.

Simpson, S. L., Edwards, L. J., Muller, K. E., Sen, P. K., and Styner, M. A. (2010). A linear exponent ar(1) family of correlation structures. *Statistics in medicine*, 29(17):1825–1838.

Smithson, M. and Verkuilen, J. (2006). A better lemon squeezer? maximum-likelihood regression with beta-distributed dependent variables. *Psychological Methods*, 11(1):54–71.

Stanley, S. (2018). Beta regression for modeling a covariate-adjusted roc. *PhD Dissertation, Baylor University*.

Sullivan Pepe, M. and Cai, T. (2004). The analysis of placement values for evaluating discriminatory measures. *Biometrics*, pages 528–535.

Vořechovský, M. (2008). Simulation of simply cross correlated random fields by series expansion methods. *Structural Safety*, 30(4):337–363.

Walker, G. A. and Shostak, J. (2010). *Common Statistical Methods for Clinical Research with SAS Examples*. SAS Institute Inc.

Wang, M. (2014). Generalized estimating equations in longitudinal data analysis: A review and recent developments. *Advances in Statistics*, 2014:1–11.

Wedderburn, R. W. M. (1974). Quasi-likelihood functions, generalized linear models, and the gauss-newton method. *Biometrika*, 61(3):439–447.

Youden, W. (1950). Index for rating diagnostic tests. *Cancer*, 3(1):32–35.

Zimprich, D. (2010). Modeling change in skewed variables using mixed beta regression models. *Research in Human Development*, 7(1):9–26.