2017

A Statistical Approach to Computed Tomography Perfusion

Seth Thomas Lirette
University of Alabama at Birmingham

A STATISTICAL APPROACH TO COMPUTED TOMOGRAPHY PERFUSION

by

SETH T. LIRETTE

INMACULADA B. ABAN, COMMITTEE CHAIR
MARGUERITE IRVIN
ANDREW SMITH
HEMANT TIWARI
NENGJUN YI

A DISSERTATION

Submitted to the graduate faculty of The University of Alabama at Birmingham,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

BIRMINGHAM, ALABAMA

2017

A STATISTICAL APPROACH TO COMPUTED TOMOGRAPHY PERFUSION

SETH T. LIRETTE

BIOSTATISTICS

ABSTRACT

Stroke is, and will continue to be, a pervasive problem in the both the United States and across the world, and computed tomography (CT) perfusion scanning will continue to be a first-line diagnostic tool to quantify where and how much blood flow occlusion is present in the brain. CT perfusion maps displaying several scalar perfusion parameters for each brain voxel will continue to be used by radiologists and other clinicians. The goal of this dissertation is to examine and improve upon current standards of practice concerning CT perfusion. We first exhibit how software that constructs perfusion maps can be implemented and distributed completely within an open-source environment, alleviating either the cost or accessibility hindrances that often accompany access to such. Although challenging and far from perfect, we prove that such a product can be delivered to researchers wishing to use it. We then seek to answer whether using the entire residue function rather than distilling it down to a few scalar quantities would prove to be a useful and enlightening approach. But before we answered that question, we first determine which modeling framework we need to employ. We then, through an extensive simulation study, go on to show that the Functional Linear Regression That's Interpretable (FLiRTI) method produced results that have the best predictive accuracy while also not producing too much of a computational burden. Finally, we take on the analysis of 93 participants with validated stroke/no stroke to assess the accuracy of cerebral blood flow, cerebral blood volume, mean transit time, the brain tissue time-attenuation curve, and the residue function to predict the presence of stroke. Logistic regression was used for the three scalar quantities and the FLiRTI method was used for the two functional curves. It was shown that the residue function did, in fact, offer the most in terms of predictive ability.

Keywords: CT perfusion, perfusion maps, R, functional data analysis, FLiRTI,

DEDICATION

This dissertation is dedicated to my wife, Ashley, and children, Jackson, Maddox, and Hannah, and any future children brought into our family.

I also dedicate this document to the triune God, who bestows upon his creation the ability to quantify a broken world and provides them the means with which to be redeemed from it.

ACKNOWLEDGMENTS

I have to begin by thanking my advisor, Dr. Inmaculada Aban, for her priceless support, encouragement, constructive criticism, and advice even during the midst of recovering from open heart surgery. You are truly a wonderful mentor, Chichi.

I would also like to thank my committee: Dr. Ryan Irvin, for her expertise in stroke epidemiology; Dr. Nengjun Yi, for adding a Bayesian perspective to things; Dr. Hemant Tiwari, for giving feedback and my only "B" at UAB; and Dr. Andrew Smith, for providing the data, being a great collaborator for many years, and leaving me stranded in Mississippi.

I would be remiss to not thank the Department of Mathematics and Statistics at Mississippi State University for instilling in me the love and wonder of mathematics and statistics and for teaching me how to adequately write a proof, and the Department of Biostatistics at UAB for augmenting my knowledge in this ever-intriguing field.

I owe incredible gratitude to the University of Mississippi Medical Center and specifically the Department of Data Science and the Center of Biostatistics and Bioinformatics for signing my paychecks and allowing me to be a guinea pig for a distance PhD. Thanks specifically to Dr. Micheal Griswold for being a mentor and taking a shot at a zero-experience graduate student six years ago.

I thank my brother, cousins, aunts, uncles, grandmother, co-workers, friends, and Grace Baptist Church of Jackson, MS, for supplying laughs, love, and endless support.

Finally, I thank my parents, Bob and Marilyn. Thank you for your financial support. Thank you for providing for me in the past and continuing to help me and my family. Most of all, thank you for instilling in me the importance of integrity, respect, and hard work.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table *Page*

THE PREDICTIVE ABILITY AND COMPUTATIONAL BURDEN OF FUNCTIONAL VS. LONGITUDINAL MODELS FOR SCALAR OUTCOMES

THE USE OF FUNCTIONAL FORMS AND SCALAR SUMMARIES OF WHOLE BRAIN PERFUSION IN PREDICTING ISCHEMIC STROKE

LIST OF FIGURES

Figure *Page*

REVIEW OF LITERATURE

A TOOL TO VISUALIZE AND ANALYZE PERFUSION DATA: DEVELOPMENT AND APPLICATION OF THE R PACKAGE "CTP"

THE PREDICTIVE ABILITY AND COMPUTATIONAL BURDEN OF FUNCTIONAL VS. LONGITUDINAL MODELS FOR SCALAR OUTCOMES

THE USE OF FUNCTIONAL FORMS AND SCALAR SUMMARIES OF WHOLE BRAIN PERFUSION IN PREDICTING ISCHEMIC STROKE

REVIEW OF LITERATURE

Background and Motivation for Research

Stroke

Stroke is the fifth leading cause of death in the United States, with approximately 800,000 people dying from strokes each year [1]. There are three broad types of stroke: hemorrhagic, ischemic, and transient ischemic attack (TIA). TIAs are often called "warnings" or "mini-strokes." Hemorrhagic strokes occur when an artery in the brain ruptures. These are easily detectable with non-enhanced Computed Tomography (CT). Our focus will be on ischemic strokes, which account for 85% of all US strokes [1]. During an ischemic stroke, an artery that supplies blood to the brain becomes blocked. Magnetic Resonance Imaging (MRI) is considered the gold-standard for diagnosing an ischemic stroke, but stroke protocols have to be done quickly in order to salvage as much brain tissue as possible. MRI scanners are not always available in such a readily manner. CT scanners, on the other hand, are almost always immediately available at most medical centers and hospitals. A problem arises in that ischemic strokes are very difficult to detect using nonenhanced CT. This is where CT Perfusion enters.

Computed Tomography Perfusion and Time Attenuation Curves

Computed Tomography Perfusion (CTP) is an imaging technique first described twenty-five years ago [2]. This was initially done with single section CT scanners and progressed into multisection CT scanners [3]. With the advent of 256 slice (and higher) CT scanners, it is now possible to perform CTP on the entire brain, but this is still relatively

rare due to the advanced machines needed [4]. During this procedure, a bolus of contrast is injected into a patient as a preselected area of tissue is repeatedly scanned while the bolus passes through the arteries and veins. As the contrast arrives to the brain tissue, it increases the "brightness" (often called "attenuation") of the tissue, measured in Hounsfield Units (HU). This is typically used to obtain one image per second or one image every two seconds [3]. This relationship between attenuation and time is plotted on a Time-Attenuation Curve (TAC), an example of which is shown in Figure 1.

Various statistics, called perfusion parameters, from these TACs are used to eval-

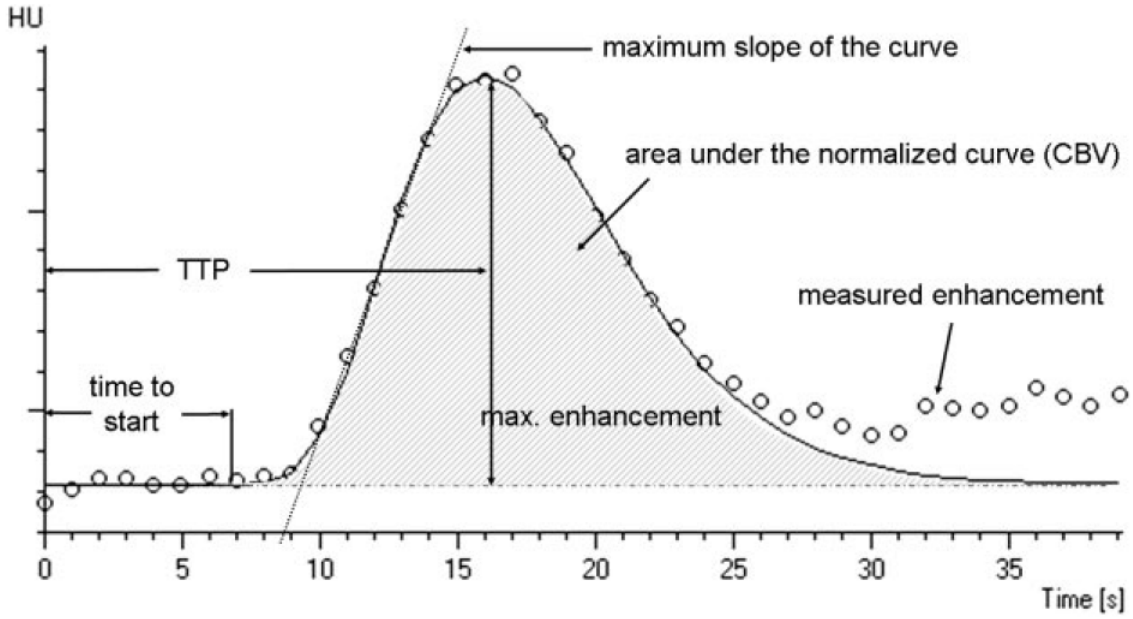

Figure 1: Example of Time-Attentuation Curve (TAC) in normal brain tissue
Note: From "Comprehensive imaging of ischemic stroke with multisection CT" by Tomandl, et al., 2003, *RadioGraphics*, 23: 565-592. Copyright 2003 by the Radiological Society of North America. Reprinted with permission.

uate the amount of blood available to be used by the brain. These are derived from an "inflow" TAC gathered from a major artery and a TAC from each voxel of brain tissue to be mapped. Essentially, the most basic measurements are flow and volume, denoted by Cerebral Blood Flow (CBF) and Cerebral Blood Volume (CBV). Two other additional measures are sometimes reported: Time To Peak (TTP) and Mean Transit Time (MTT). TTP is the time from the start of injection until maximum contrast in reached. TTP can be easily

calculated empirically without any estimation techniques. CBV is a fairly straightforward calculation consisting of the area under the curve (AUC) of the tissue TAC divided by the AUC of the arterial TAC. MTT is defined as the mean time between the arterial inflow and venous outflow and has a direct relationship to both CBV and CBF. This will be discussed later. MTT and CBF estimation vary tremendously depending on the method used for the estimation. The perfusion parameters can then be graphed regionally for every volume-pixel (voxel) and provided with a color map. Figure 2 shows an example of a nonenhanced CT, TTP color map, CBF color map, and CBV color map. Figure 3 shows the relationship between the tissue TAC and the various color maps. The perfusion parameters mapped in figures 2 and 3 were mapped using a technique called the "maximum slope method," which is just one of many methods to be discussed in the next section.

The estimation techniques depend on the vendor of the postprocessing software. The variability of the estimation of CBF and CBV are of great consequence. For example, at CBF 35 mL/100 g/min or less brain tissue can still survive. At CBF of 20 mL/100 g/min or less transmission between neurons is disturbed. At CBF of 10 mL/100 g/min or less, cell death occurs. So the estimation of these different perfusion parameters matter tremendously. Traditionally, the postprocessing software provided through commercial CT vendors evaluate TACs exclusively through the use of mathematical models.

Estimating Perfusion Parameters with Mathematical Models

Three of the most important measures when trying to determine the amount of salvageable brain tissue are the CBF, CBV, MTT. They provide clinicians with visually identifiable areas of the brain that are either fully dead, or potientially salvagable. For any particular voxel, this is given by the simple fraction

$$CBF = \frac{CBV}{MTT} \tag{1}$$

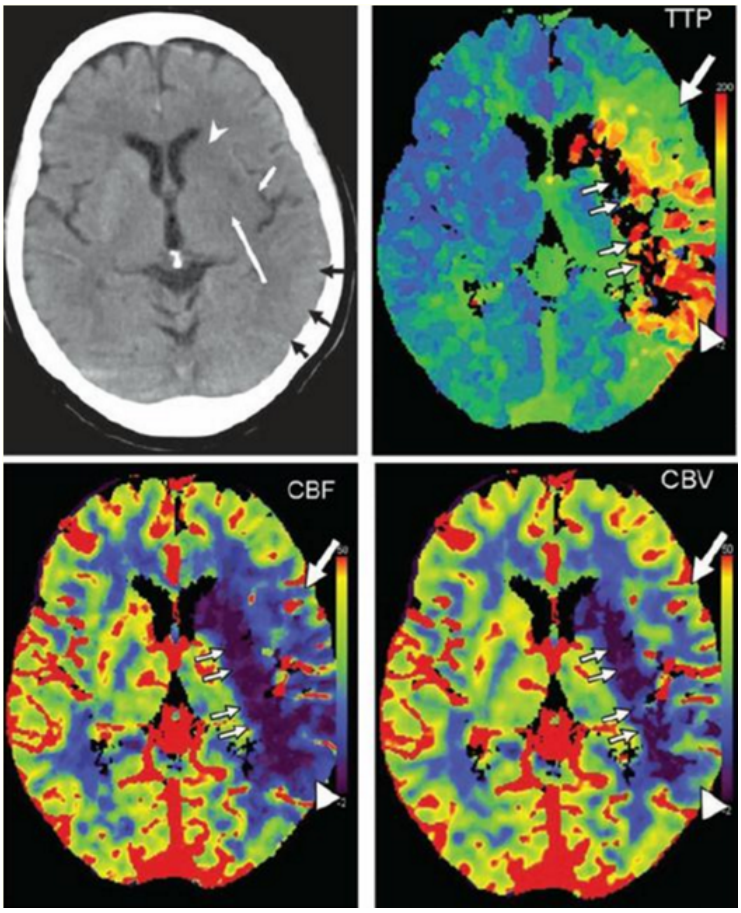

Figure 2: Example of Color Mapping a TAC. For TTP, red means longer TTP (bad), blue means shorter TTP (good). For CBF, red means faster flow (good), blue means slower flow (bad). For CBV, red means more volume (good), blue means less volume (bad).
Note: From "Comprehensive imaging of ischemic stroke with multisection CT" by Tomandl, et al., 2003, *RadioGraphics*, 23: 565-592. Copyright 2003 by the Radiological Society of North America. Reprinted with permission.

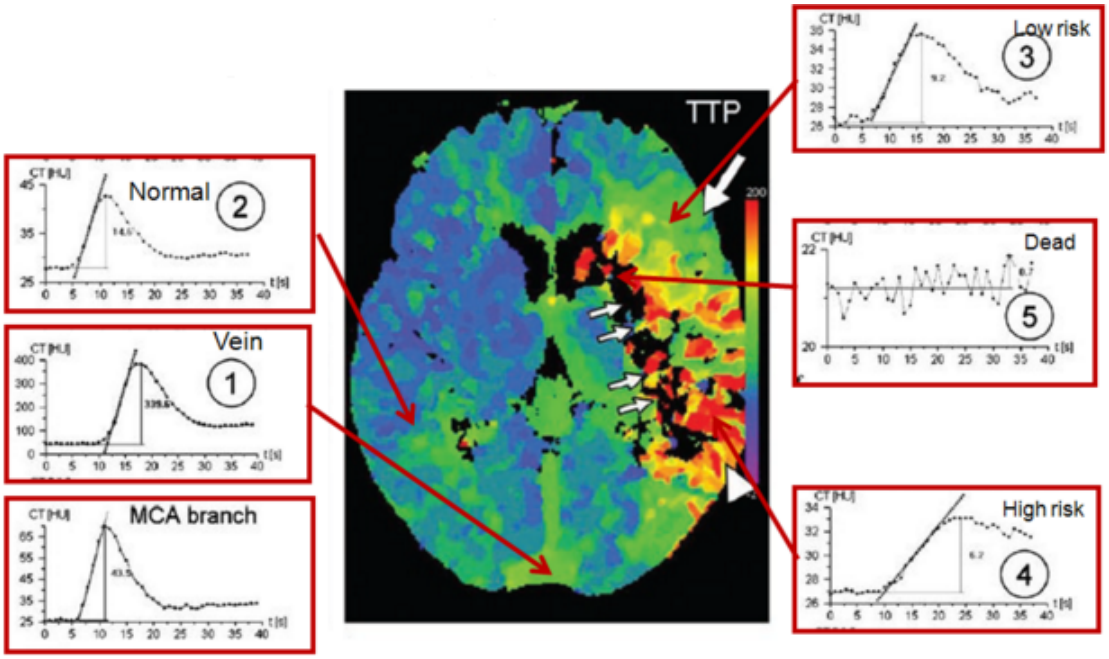

Figure 3: Example of relationship between Color Mapping and a TAC for TTP. The MCA branch is not pictured on the brain image. The vein (1) is slightly slower than the MCA. (2) depicts normal tissue, shown with a blue color map. (3) shows tissue at low risk. Note the less steep slope and longer TTP. (4) shows high risk tissue, mapped by the color orange. (5) shows dead brain tissue with a black color map. The TAC is completely noise because the contrast cannot get to that area of the brain.

Note: From "Comprehensive imaging of ischemic stroke with multisection CT" by Tomandl, et al., 2003, *RadioGraphics*, 23: 565-592. Copyright 2003 by the Radiological Society of North America. Reprinted with permission.

and CBV can be calculated by

$$CBV = \frac{\int_0^T C_t(t)dt}{\int_0^T C_a(t)dt} \tag{2}$$

where, T is the maximum time observed, $C_t(t)$ is the TAC for brain tissue and $C_a(t)$ is the TAC for the artery [5] [6]. The majority of the work involves calculating the CBF (or, conversely, the MTT), and there are two mathematical approaches in deriving the CBF: methods not based on deconvolution and methods based on convolution.

Non-deconvolution Methods

Methods not involving deconvolution – finding the solution to a convolving equation – were the first to be considered for calculating CBF. The main attraction is that these can be calculated quickly. At the core, these methods are based on the Fick principle of conservation of mass for cardiac output. In its strict form this requires TACs for arterial, venous, and tissue:

$$C_t(t) = CBF \cdot \int_0^T (C_a(t) - Cv(t))dt \tag{3}$$

This method is considered impractical due to the need for TACs for tissue, arterial, and venous phases – the physiological action of blood entering an organ through an artery, being dispersed to tissue via capillaries, and exiting through a vein. A simplification arises is to assume that venous concentration is zero during the time we are concerned with observing. This specification is called the Mullani-Gould formulation, the "No Venous Outflow Method," or the "Single Compartment Formulation." It is written as:

$$C_t(t) = CBF \cdot \int_0^T C_a(t)dt \tag{4}$$

Applying the Fundamental Theorem of Calculus and recognizing that the rate of contrast attenuation will peak when the arterial phase peaks gives us the final form of what is called

the "maximum slope method."

$$\left[\frac{dC_t(t)}{dt}\right]_{max} = CBF \cdot [C_a(t)]_{max} \qquad [5][7]. \tag{5}$$

To show what the left-hand side of this equation translates to in clinical practice, consider again Figure 3. The box labeled "Normal" has a very steep slope on the increasing of the TAC, shown by the black straight line. Moving to the "Low risk" box, we see the slope as somewhat flatter. In the "High risk" box, the slope is even more flat, and the slope for "Dead" is completely horizontal.

There are several documented disadvantages of the maximum slope method. A very high rate of contrast injection is required, rates almost never achieved in practice, due to patient safety reasons. The method is very sensitive to noise on the CT image. And the no venous outflow assumption is an oversimplification. The main advantage of the maximum slope method was that it allowed for calculation of the perfusion parameters sooner, where deconvolution was more computationally intensive. This was a real issue fifteen years ago, when CTP was first being established. Now this is typically not a problem with modern computing power. Deconvolution methods are now considered the gold standard for calculating CTP parameters, and, thus, will be our focus henceforth.

Deconvolution Methods

Methods involving deconvolution were first set forth by Østergaard et al. [8]. As previously stated, deconvolution methods are almost unanimously considered superior to non-deconvolution methods. All deconvolution methods start by defining two functions. The TAC for brain tissue ($C_t(t)$) can be defined by two functions: (1) the TAC for the artery ($C_a(t)$) and (2) the residue function $R(t)$. $R(t)$ is defined as the TAC "of the tissue due to an idealized instantaneous injection" [7]. In other words, $R(t)$ measures the fraction of contrast still present in a voxel at time t. $R(t)$ is unitless and $R(0) = 1$. It can be shown

that

$$C_t(t) = CBF \cdot [C_a(t) \otimes R(t)] = CBF \cdot \int_{\tau}^{t} C_a(\tau) R(t-\tau) d\tau \tag{6}$$

where $\otimes$ is the convolution operator, defined for continuous functions as:
$f(x) \otimes g(x) = \int_z^x f(z)g(x-z)dz$ [5] [7] [9].

The rest of the problem involves deconvolving equation (6), and there are two main non-parametric approaches to this. Equation (6) can be deconvolved using Fourier transforms, but this solution is very sensitive to noise on the CT. By far the most commonly used methods involve singular value decomposition.

Singular Value Decomposition Approaches

Let us begin by revisiting MTT. Recall that MTT is the Mean Transit Time. In statistical terms, this is simply the expected value of t for some function $h(t)$ that describes the distribution of contrast. Hence,

$$MTT = \int_0^{\infty} t\ h(t)\ dt. \tag{7}$$

The formulation given by Wang et al. [9] gives

$$MTT = 1 - \int_0^{\infty} R(t) dt \tag{8}$$

We now return to equation (6). Assuming brain tissue and arterial attenuation are measured at equally spaced time (Δt) points $t_0, t_1, ..., t_N$ (an assumption generally met), equation (6) can be discretized as

$$C_t(t_i) = CBF \cdot \int_{\tau}^{t_i} C_a(\tau) R(t_i - \tau) d\tau = \Delta t \cdot CBF \cdot \sum_{k=0}^{t_N} C_{a_i}(t) R(t_i - k) \tag{9}$$

where $t, k \in t_0, t_1, ..., t_N$. Using matrix notation, for each $t_i, i = 0, \ldots, N$, and using

equation (9), we get

$$\boldsymbol{C_t} = \Delta t \cdot CBF \cdot \boldsymbol{C_a R} \tag{10}$$

where $\boldsymbol{C_t} = \begin{bmatrix} C_t(t_0) \\ C_t(t_1) \\ \vdots \\ C_t(t_N) \end{bmatrix}$, $\boldsymbol{C_a} = \begin{bmatrix} C_a(t_0) & 0 & \dots & 0 \\ C_a(t_1) & C_a(t_0) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ C_a(t_N) & C_a(t_{N-1}) & \dots & C_a(t_0) \end{bmatrix}$, and $\boldsymbol{R} = \begin{bmatrix} R(t_0) \\ R(t_1) \\ \vdots \\ R(t_N) \end{bmatrix}$

Now we can solve for matrix $\mathbf{R}$. Using Singular Value Decomposition (SVD), $\boldsymbol{C_a}$ can be decomposed into $\boldsymbol{C_a} = \boldsymbol{USV}^T$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices and $\boldsymbol{S}$ is a diagonal matrix with singular values of matrix $\boldsymbol{C_a}$. The inverse is given by $\boldsymbol{C_a^{-1}} = \boldsymbol{V\bar{S}^{-1}U}^T$. The analytical inverse of $\boldsymbol{S}$ can be found; however, this inverse gives a solution that is not very smooth and no information of interest can be gathered from it. Here $\boldsymbol{\bar{S}^{-1}}$ approximates the inverse of $\boldsymbol{S}$ and is given by

$$\bar{S}_{ii}^{-1} = \begin{cases} 1/\mathrm{S}_{ii} & \text{if } \mathrm{S}_{ii} > P_{SVD} \\ 0 & \text{otherwise} \end{cases}, i = 0, ..., N$$

P_{SVD} is a cut-off set to some percentage of the maximum singular value of $\boldsymbol{C_a}$. Smaller P_{SVD} give more accurate MTT values but $\boldsymbol{R}$ will be less smooth. P_{SVD} is usually set between 5% and 20% of the maximum singular value of $\boldsymbol{C_a}$. We can now solve for an estimate of $\boldsymbol{R}$, given by

$$\boldsymbol{\hat{R} = C_a^{-1} C_t \approx V\bar{S}^{-1}U^T C_t} \tag{11}$$

From which we get our final estimate for MTT and CBF as

$$\widehat{MTT} = \frac{\Delta t \cdot \sum_{i=0}^{N} \hat{R}(t_i)}{max\hat{R}(t_i)} \tag{12}$$

$$\widehat{CBF} = max(\hat{R}(t_i)) \quad [9] \tag{13}$$

Different SVD algorithms

It was generally inferred from the first development of SVD deconvolution algorithms that the accuracy could be improved upon. What was described in the previous section is the most basic of the SVD deconvolution methods, termed either SVD or standard SVD (sSVD). Modifications exist that improve upon sSVD. Two documented algorithms are the block-circulant SVD (bSVD) and the oscillation minimization bSVD (oSVD) [9]. The bSVD algorithm extends the length of the TACs by adding zeros to the end giving a convolution matrix

$$\boldsymbol{C_B} = \begin{bmatrix} \boldsymbol{C_a} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{C_a} \end{bmatrix} \text{ where } \boldsymbol{C_a} \text{ is defined as in the previous section and}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & C_a(t_N) & C_a(t_{N-1}) & \dots & C_a(t_1) \\ 0 & 0 & C_a(t_N) & \dots & C_a(t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The system is then solved the same way as sSVD. Still the P_{SVD} cut-off must be chosen arbitrarily. To eliminate this the oSVD algorithm adaptively chooses singular value thresholds for each voxel, depending on the smoothness of $R(t)$ [10].

A problem arises from the fact that there is no standardized procedure for generating perfusion maps. There are not even standardized acronyms for the different algorithms. In addition, most vendors keep their software algorithms proprietary and do not release information about them. Kudo et al. [11] studied the maps generated from 13 different software algorithms distributed by 7 different vendors. No two packages created the same maps. Other studies have documented the differences in different software implementations [12] [13]. Furthermore, can statistical models augment these mathematical models? We will consider four statistical approaches.

Statistical Approaches To Augment the Mathematical Models

Once we use one or more of the methods described in the previous section to construct $R(t)$, we must ask the question: instead of distilling $R(t)$ into one single measure such as CBF, Tmax, MTT, etc., how can we best use every piece of information contained within $R(t)$? This is where we will take a statistical stance. We will present four potential statistical modeling techniques: generalized linear mixed models, generalized estimating equations, a two stage model based on the best linear unbiased predictor, and functional linear regression that's interpretable. In all models, let $Y = P(\text{confirmed stroke } |\mathbf{x})$.

Generalized Linear Mixed Models

When starting to construct a model to predict either a continuous or binary outcome and applying Occam's Razor, one of the most natural places to begin is the generalized linear model (GLM). A common formulation of the GLM is given in equation 14.

$$g(\mu) = \beta_0 + \beta_1^T \mathbf{X} \tag{14}$$

where $\mu = E(Y|\mathbf{x}))$ and g is the link function.

Although a variety of distributional families for Y and link functions for g can be employed, the most common are $Y \sim$ Gaussian with identity link function when the outcome is continuous and $Y \sim$ Bernoulli with logit link for a binary outcome. We see no need to deviate from these canonical models.

Using the most basic model, the GLM, we immediately encounter a problem. One of the assumptions – arguably the most important – of the GLM is the assumption of independence or exchangeability across the rows of the dataset. This assumption is obviously violated in the data structures we are interested in studying. We will now present three models that account for this.

Generalized Estimating Equations

Generalized Estimating Equations (GEE) were first described by Liang and Zeger as a longitudinal data analysis method [14]. It is a general method for analyzing repeated measure data where the subject-level observations are correlated, but observations across subjects are independent. GEE fits marginal or population-average models, where the goal is to make inference and prediction about the population while accounting for within-subject correlation. This is distinct from the subject-specific models such as generalized linear mixed models. Given that we are looking at repeated measures for the predictor set with a scalar outcome, we are only concerned with population average models in this paper.

Let i independent observations index outcome $Y_i = (Y_{i1}, ..., Y_{ij}, ..., Y_{in_i})$, then $\mathrm{E}(Y_{ij} = \mu_{ij})$ is related to predictor set x by

$$g(\mu_{ij}) = x_{ij}^T \beta \tag{15}$$

Let R_i denote the working correlation matrix and let

$$V_i = A_i^{1/2} R_i A_i^{1/2} \tag{16}$$

where A_i is a diagonal matrix with elements $Var(Y_{ij})$. The estimate $\hat{\beta}$ is then the solution to the equation

$$\sum_i \frac{\partial \mu_i^T}{\partial \beta} V_i^{-1} (Y_i - \mu_i) = 0. \tag{17}$$

Many different correlation structures can be specified for R_i, but given how close together our time points are sampled (2 seconds, equally spaced) an autoregressive of order one correlation structure is our choice. This is specified as

$$R = \begin{bmatrix} 1 & \rho & \dots & \rho^{N-1} \\ \rho & 1 & \dots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \dots & 1 \end{bmatrix}.$$

Like GLM, we can specify family and link functions for Y and g and will continue with the choices described in that section.

A Two-Stage Model Using the Best Linear Unbiased Predictor

Since we have an outcome that is time-invariant, we are relegated to using marginal or population average models. However, our predictor set is time-varying. This begs the question: Can we port the information contained in the subject-specific values into a population average model? Stanek et al. asked this first, formulating the question as: "Why not routinely use best linear unbiased predictors (BLUPs) as estimates of cholesterol, per cent fat from kcal and physical activity?" [15]. They argue that the best linear unbiased predictor (BLUP) should routinely be used in such situations. Morrell et al. then applied the same strategy to improve estimation for logistic models [16]. The model most closely resembling our two-stage model was constructed by Shetty et al. [17]. The BLUPs are the subject-specific predictions using both fixed and random effects from a linear mixed model. These predicted values are then used as the predictors in a marginal model. More specifically the modeling procedure, using the motivating data, is:

1. Fit a random intercept-slope model with attenuation as the outcome and time as the predictor and obtain subject-specific parameter estimates from the model
2. Fit GEE model using intercept and slope estimates from Step 1 using autoregressive of order 1 correlation with appropriate family and link functions

Functional Linear Regression That's Interpretable

In most instances when we think of data or the modeling of data, we usually envision a number of variables taken from a sample of individuals with one observation per subject in the cross-sectional setting and repeated measurements per subject, measured at "discrete" time points, in a longitudinal study. These would perhaps be modeled ordinarily by generalized linear mixed models or some marginal method such as GEE. Functional data analysis (FDA) encompasses a different way to conceptualize a datum. In FDA, the most basic unit of information on a subject is an entire observed curve or function. Functional regression and its analogue, the generalized functional linear model (GFLM), are two very useful tools in FDA. Ramsay and Silverman (2002) and Muller and Stadtmuller (2005) provide a variety of examples and applications [18] [19]. Each technique under the GFLM umbrella falls into one of three categories: (1) function-on-scalar regression (2) function-on-function regression or (3) scalar-on-function regression. Our focus will be on the last.

Scalar-on-function regression is typically expressed by the following model

$$Y_i = \beta_0 + \int \beta(t) X_i(t) dt + \epsilon_i. \tag{18}$$

The coefficient function $\beta(t)$ is usually represented in one of three ways. The first method represents $\beta(t)$ as some *p*-dimensional basis function. The second method attempts to shrink the variability in $\beta(t)$ via some least squares penalty. The third method, and our method of choice takes a variable selection approach to solving the problem. James et al. have named this method "Functional Linear Regression That's Interpretable" (FLiRTI) [20]. FLiRTI is an attractive choice because it gives accurate predictions, is flexible, has nice theoretical properties, is computationally efficient, and unlike the first two methods, gives an estimate for the coefficient function $\beta(t)$ that is interpretable. Regarding the last advantage, because FLiRTI is a shrinkage procedure, it effectively "zeros" the time points

that do not play a significant role in predicting the outcome while retaining time points that have significant effects. James et al. display this feature well when applying FLiRTI to the Canadian Weather Data in the original FLiRTI paper [20]. This paper focuses on the predictive accuracy of FLiRTI.

We will give a brief overview of the FLiRTI model and model fit. Full details can be seen in James et al. [20]. The FLiRTI model borrows an idea from the basis approach but allows errors in the $\beta(t)$ specification. Specifically

$$Y_i = \beta_0 + \mathbf{X}_i^T \eta + \epsilon_i^*. \tag{19}$$

where $\mathbf{X}_i = \int X_i(t)\mathbf{B}(t)dt$, $\epsilon_i^* = \epsilon_i + \int X_i(t)e(t)dt$, and the p-dimensional basis $\mathbf{B}(t) = [b_1(t), b_2(t), ..., b_p(t)]^T$. If we define

$$\boldsymbol{\gamma} = A\eta \tag{20}$$

where A is a vector combining finite difference operators with the basis functions, we can combine (6) and (7) to get the FLiRTI model

$$\mathbf{Y} = XA^{-1}\boldsymbol{\gamma} + \boldsymbol{\epsilon}^*. \tag{21}$$

The model is then fit using either the LASSO selector via the LARS algorithm or the Dantzig selector via the DASSO algorithm to obtain $\hat{\boldsymbol{\gamma}}$ [21] [22] [23] [24]. The FLiRTI estimate for $\beta(t)$ is then given by

$$\hat{\beta}(t) = \mathbf{B}(t)^T A^{-1}\hat{\boldsymbol{\gamma}}. \tag{22}$$

FLiRTI has tuning parameters that correspond to tuning parameters in any smoothing situation. The most common approach to choosing these is to choose the combination that produce the lowest cross-validated residual sum of squares, and this is the approach we take in our simulation study. For inference, one can obtain bootstrapped confidence inter-

vals for $\beta(t)$, and James et al. describe a permutation test to test for statistical significance of the relationship between $X(t)$ and Y [20].

Software for CT Perfusion

As of this writing, there are two major classes of software for CTP. The first is the proprietary software developed and distributed by the five major CT manufacturers (General Electric Healthcare, Hitachi Medical Systems, Philips Medical Systems, Siemens Healthcare, and Toshiba Medical Systems). The second class consists of two academic programs, Perfusion Mixmatch Analyzer (PMA) from the Acute Stroke Imaging Standardization Group [25], Japan, and Stroketool from Digital Image Solutions, Germany [26].

Both of these classes have drawbacks. The commercial software is designed to only work on each particular company's CT hardware. Further, assuming one has compatible software, one is incredibly hard-pressed to discover the purchasing price of the software. It is assumed the cost is prohibitive to many who would like to use the data output from these machines. The PMA is somewhat restrictive in its use. One is actually required to take a quiz proving that you have sufficient knowledge of imaging techniques before you are given an access code. Although the software is free and is expressly restricted for use by researchers, not all researchers will be able to even download the software. Stroketool has not been updated since 2011 and does not support any operating systems newer than Windows XP.

This makes the R computing environment very attractive [27]. It is completely open source and, therefore, can be used by any researcher. It lends itself to statistical applications very seamlessly, and software updates are easily distributed.

The Motivating Problem: The UMMC CeCT-CED Study

The Contrast-enhanced CT Color Enhanced Detection (CeCT-CED) study at the University of Mississippi Medical Center (UMMC) provided the motivation for this project. This in an IRB approved, HIPAA compliant, retrospective study and included patients pre-

sented to the Emergency Department at UMMC between January 1, 2010 and April 15, 2014 with new onset of acute stroke symptoms, triggering a code gray protocol that includes Nonenhanced CT (NECT), CT Angiography, and CTP imaging of the head. Included patient were adults with available NECT images (N = 423). Patients without a confirmatory brain MRI (gold standard for ischemic infarcts) or a second NECT exam documenting hemorrhagic strokes were excluded (N = 133). There were 290 subjects in the final inclusion cohort. For patients with ischemic strokes, the confirmatory MRI was used to categorize the type of stroke. Since CTP was only a minor part of the study not all were included for assessment of CTP. Of the 290 subjects, 93 were randomly chosen for inclusion into the CTP portion of the study.

There are two different data objects that are of interest to us. The first is, for each of the 93 participants, we have a series of fourteen grayscale images measured at one second intervals in the usual fashion of CTP acquisition. We wish to use these images to construct a parametric maps of CBF using self written software implementing the previously described algorithms in the R computing environment [27].

The second data object we are interested in is described as followed. A designated region of interest on each patient's CTP preprocessed images was then examined by a trained neuroradiologist, and data was recorded. Of the variables collected, there are 6 in which we are interested. First, for each patient's region of interest, the attenuation value (in Hounsfield units) was recorded along with the time at which the attenuation value occurred. Using this data, we can then construct $C_t(t)$. Also collected was the time at which the contrast entered the artery, the maximum attenuation during this arterial phase, and the duration of this arterial phase. Finally, each patient received a confirmatory MRI indicating the presence or absence of an ischemic stroke.

Research Goals

Paper 1

As shown in the section detailing the available software for perfusion mapping, there does not exist a truly open source academic tool to construct a CT perfusion map. Paper 1 details the construction of such a program and showcases its implementation.

We give a review of the process used to create the program and the mathematics underpinning it. We perform a study on applying to program to a simulated digital phantom, with details as to how the phantom in generated, to test the accuracy of the program and describe the results. We then give details and worked examples of how to download and call the program while showcasing its implementation on two subjects from our data. We then discuss best practices for use and detail some of the limitations of the program.

This paper is currently under review in *Computer Methods and Programs in Biomedicine*.

Paper 2

When presented with data on dozens, if not hundreds of individuals, each with dozens of time points, what modeling technique is both most appropriate and best utilizes computational resources? Paper 2 sheds light on this question.

Paper 2 will consist of a simulation study of GLMM vs. GEE vs. two-stage vs. functional regression. We first describe our motivating problem and possibly generalizations to other datasets. We then detail the four considered modeling frameworks listing above. We move on to describe the simulation study for both continuous outcomes and binary outcomes and then present results of the simulation studies of model performance, as quantified by root mean squared error, mean absolute error, and the Brier score, mostly through a series of graphs. Finally, we offer a general discussion of the results, implications, and plans for future research.

Paper 3

The primary aim of the CeCT-CED study described in the previous section was to develop and improve the color enhanced detection for the CT angiography modality. No one using these data has investigated the effect that the CTP data may have on predicting an MRI confirmed ischemic stroke. Given the richness of the data at hand, this seems like a waste. This paper demonstrates the effect of perfusion parameters and the use of the full $R(t)$ on the probability of an ischemic stroke. Papers 1 and 2 help build the foundation for the crescendo that is Paper 3.

We show how the scalar values of CBF, CBV, and MTT relate to the probability of ischemic stroke. For each participant, we calculate the three perfusion parameters. Then, using the information gleaned from Paper 2 (FLiRTI models are useful for binary outcome prediction), we employ this specific functional data analysis technique demonstrate how the full $R(t)$ curve relates to stroke probability. We demonstrate how $R(t)$ will offer more predictive value than any one of the scalar quantities. We then, through a series of statistical graphs, offer insight into how the residue function is useful in determining stroke probability.

Future Research

Finally, we end this dissertation with a summary of conclusions and limitations and propose ideas for future research.

A TOOL TO VISUALIZE AND ANALYZE PERFUSION DATA: DEVELOPMENT AND APPLICATION OF THE R PACKAGE "CTP"

SETH T. LIRETTE, ANDREW SMITH, AND INMACULADA B. ABAN

Submitted to *Computer Methods and Programs in Biomedicine*

Format adapted for dissertation

1 INTRODUCTION

Of the two major classifications of stroke (hemorrhagic and ischemic), ischemic is the most prevalent [1]. During an ischemic stroke, blood supply to the brain is either temporarily or permanently reduced or occluded. If the location of an ischemic stroke can be detected early, a treatment plan can be quickly developed to break the clot using thrombolytic drugs. It is therefore important to visually identify where a stroke is occurring and the extent of the damage to the brain tissue. This is where brain perfusion is of great use.

Perfusion is defined as the process of delivering blood to a tissue via a capillary bed. Despite this straightforward definition, perfusion itself is difficult to measure, especially in the brain. Thus we are limited to estimating perfusion. This is also not an easy endeavor. Radiological brain imaging provides two tools to help estimate the delivery of blood to the brain: dynamic susceptibility contrast perfusion-weighted imaging, taken from an MRI, or computed tomography (CT) perfusion. There is variability in both the image acquisition parameters across different scanners and in the mathematical models for calculation perfusion, which are typically vendor specific; however, the image acquisitions are standardized, and the input data are constant across machine vendors. The method used to estimate various parameters of perfusion does not depend on the type of machine used to acquire the data, therefore from this point forward we will focus the discussion around CT perfusion.

Computed Tomography Perfusion (CTP) is an imaging technique first described twenty-five years ago [2]. This was initially done with single section CT scanners and progressed into multisection CT scanners [3]. With the advent of 256 slice (and higher) CT scanners, it is now possible to perform CTP on the entire brain, but this is still relatively

rare due to the advanced machines needed [4]. During this procedure, a bolus of contrast is injected into a patient while a preselected area of tissue is repeatedly scanned while the bolus passes through the arteries and veins. As the contrast arrives to the brain tissue, it increases the "brightness" (often called "attenuation") of the tissue, measured in Hounsfield Units (HU). This is typically used to obtain one image per second [3]. This relationship between attenuation and time can be plotted on a Time-Attenuation Curve (TAC).

Various quantities, called perfusion parameters, from these TACs are used to evaluate the amount of blood available to be used by the brain. These are derived from an "inflow" TAC gathered from a major artery and a TAC from each voxel of brain tissue to be mapped. Essentially, the most basic measurements are flow and volume, denoted by Cerebral Blood Flow (CBF) and Cerebral Blood Volume (CBV). Two other additional measures are sometimes reported: Time To Peak (TTP) and Mean Transit Time (MTT). TTP is the time from the start of injection until maximum contrast in reached. TTP can be easily calculated empirically without any estimation techniques. CBV is a fairly straightforward calculation consisting of the area under the curve (AUC) of the tissue TAC divided by the AUC of the arterial TAC. MTT is defined as the mean time between the arterial inflow and venous outflow and has a direct relationship to both CBV and CBF. This will be discussed later. MTT and CBF estimation vary tremendously depending on the method used for the estimation. The perfusion parameters can then be graphed regionally for every voxel and provided with a color map. The perfusion parameter most widely applied and the most easily understood, conceptually, is CBF. It is, arguably, the most important. If we can map where blood flow decreases or stops, we can see where an infarct (or stroke) occurs.

The estimation techniques vary across vendors of the postprocessing software [5] [6] [7] [8] [7]. The variability of the estimation of CBF and CBV are of great consequence. For example, a decrease of 10-15 mL/100 g/minute can signal that blood is having difficulty perfusing an area, implying there is a blockage and giving information to the clinician on where intervention should occur. So the estimation of these different perfusion parame-

ters matters tremendously. In addition, most of the software are proprietary and expensive ($50,000 as a conservative estimate) making it difficult for researchers to evaluate and further improve estimation methods. Thus there is a need for a learning and analysis tool for these perfusion data.

In this paper, we present an open source tool in the R computing environment [9] that we have developed which may be used to help clinicians and other researchers explore the perfusion results from their data. We illustrate this novel tool for evaluating different methods for perfusion estimation and showcase the application of the tool. This paper is organized as follows: a brief introduction to different perfusion parameter estimation techniques, a description of the simulation (called a phantom in imaging studies), software implementation and usage, results from both the simulation study and in vivo CT data, and finally a discussion of limitations and future directions of the software.

2 METHODS

2.1 Techniques for Estimating Perfusion Parameters

The essence of the problem of estimating the perfusion parameters can be distilled down to this question: how can we arrange the series of images shown in Figure 1, consisting of 27 sequential grayscale CT images measured at one second intervals, to represent CBF? We will briefly discuss the different methods for accomplishing this with the goal of comparing and contrasting the accuracy and performance of the methods.

Three of the most important measures when trying to determine the amount of salvageable brain tissue are the CBF, CBV, MTT. They provide clinicians with visually identifiable areas of the brain that are either fully dead, or potientially salvagable. CBF is, arguably, the most important as it provides information to help answer whether blood is flowing to the brain. For any particular pixel/voxel, this is defined mathematically as the simple fraction

$$CBF = \frac{CBV}{MTT} \tag{1}$$

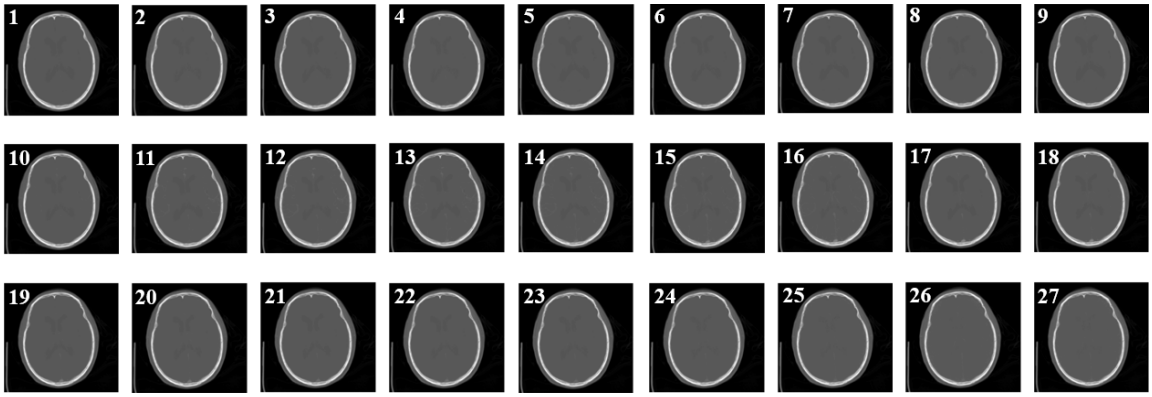

Figure 1: Original grayscale images of single slice CTP data taken at 27 one second intervals from a 72 year-old male. These are the raw data images used to construct the CBF map shown in Figure 3A.

and CBV can be calculated by

$$CBV = \frac{\int_0^T C_t(t)dt}{\int_0^T C_a(t)dt} \tag{2}$$

where, T is the maximum time observed, $C_t(t)$ is the TAC for brain tissue and $C_a(t)$ is the TAC for the artery [10] [11]. The majority of the work involves calculating the CBF (or, conversely, the MTT), and there are two mathematical approaches in deriving the CBF: methods not based on deconvolution and methods based on convolution.

Non-deconvolution Methods

Methods not involving deconvolution – finding the solution to a convolving equation – were the first to be considered for calculating CBF and first described by Miles et al. [2]. The main attraction is that these can be calculated quickly. At the core, these methods are based on the Fick principle of conservation of mass for cardiac output. In its strict form this requires TACs for arterial (C_a), venous (C_v), and tissue (C_t):

$$C_t(t) = CBF \cdot \int_0^T (C_a(t) - Cv(t))dt \tag{3}$$

This method is considered impractical due to the need for TACs for arterial, tissue (and capillary), and venous phases – the physiological action of blood entering an organ through an artery, being dispersed to tissue via capillaries, and exiting through a vein. A simplification arises is to assume that venous concentration is zero during the time we are concerned with observing. This specification is called the Mullani-Gould formulation, the "No Venous Outflow Method," or the "Single Compartment Formulation" [12]. It is written as:

$$C_t(t) = CBF \cdot \int_0^T C_a(t)dt \tag{4}$$

Applying the Fundamental Theorem of Calculus and recognizing that the rate of contrast attenuation will peak when the arterial phase peaks gives us the final form of what is called the "maximum slope method."

$$\left[\frac{dC_t(t)}{dt}\right]_{max} = CBF \cdot [C_a(t)]_{max} \qquad [10][13]. \tag{5}$$

There are several documented disadvantages of the maximum slope method. A very high rate of contrast injection is required, rates almost never achieved in practice, due to patient safety reasons. The method is very sensitive to noise on the CT image and is an oversimplification. The main advantage of the maximum slope method was that it allowed for calculation of the perfusion parameters sooner, where deconvolution was more computationally intensive. This was a real issue fifteen years ago, when CTP was first being established. Now this is typically not a problem with modern computing power. Deconvolution methods are now considered the gold standard for calculating CTP parameters, and, thus, will be our focus henceforth.

Deconvolution Methods

Methods involving deconvolution were first set forth by Østergaard et al. [14]. As previously stated, deconvolution methods are almost unanimously considered superior to

non-deconvolution methods. All deconvolution methods start the same. The TAC for brain tissue ($C_t(t)$) can be defined by two functions: (1) the TAC for the artery ($C_a(t)$) and (2) the residue function $R(t)$. $R(t)$ is defined as the TAC "of the tissue due to an idealized instantaneous injection" [13]. In other words, $R(t)$ measures the fraction of contrast still present in a voxel at time t. $R(t)$ is unitless and $R(0) = 1$. It can be shown that

$$C_t(t) = CBF \cdot [C_a(t) \otimes R(t)] = CBF \cdot \int_{\tau}^{t} C_a(\tau) R(t-\tau) d\tau \tag{6}$$

where $\otimes$ is the convolution operator, defined for continuous functions as:
$f(x) \otimes g(x) = \int_z^x f(z)g(x-z)dz$ [10] [13] [15].

The rest of the problem involves deconvolving equation (6), and there are two main non-parametric approaches to this. Equation (6) can be deconvolved using Fourier transforms, but this solution is very sensitive to noise on the CT. By far the most commonly used methods involve singular value decomposition.

Singular Value Decomposition Approaches

Let us begin by revisiting MTT. Recall that MTT is the Mean Transit Time. In statistical terms, this is simply the expected value of t for some function $h(t)$ that describes the distribution of contrast. Hence,

$$MTT = \int_0^{\infty} t \; h(t) \; dt. \tag{7}$$

The formulation given by Wang et al. [15] gives

$$MTT = 1 - \int_0^{\infty} R(t) dt \tag{8}$$

We now return to equation (6). Assuming brain tissue and arterial attenuation are measured at equally spaced time (Δt) points $t_0, t_1, ..., t_N$ (an assumption generally met),

equation (6) can be discretized as

$$C_t(t_i) = CBF \cdot \int_{\tau}^{t_i} C_a(\tau) R(t_i - \tau) d\tau = \Delta t \cdot CBF \cdot \sum_{k=0}^{t_N} C_{a_i}(t) R(t_i - k) \qquad (9)$$

where $t, k \in t_0, t_1, ..., t_N$. Using matrix notation, for each $t_i, i = 0, \ldots, N$, and using equation (9), we get

$$\boldsymbol{C_t} = \Delta t \cdot CBF \cdot \boldsymbol{C_a R} \qquad (10)$$

where $\boldsymbol{C_t} = \begin{bmatrix} C_t(t_0) \\ C_t(t_1) \\ \vdots \\ C_t(t_N) \end{bmatrix}$, $\boldsymbol{C_a} = \begin{bmatrix} C_a(t_0) & 0 & \ldots & 0 \\ C_a(t_1) & C_a(t_0) & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ C_a(t_N) & C_a(t_{N-1}) & \ldots & C_a(t_0) \end{bmatrix}$, and $\boldsymbol{R} = \begin{bmatrix} R(t_0) \\ R(t_1) \\ \vdots \\ R(t_N) \end{bmatrix}$

Now we can solve for matrix $\mathbf{R}$. Using Singular Value Decomposition (SVD), $\boldsymbol{C_a}$ can be decomposed into $\boldsymbol{C_a} = \boldsymbol{U S V}^T$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices and $\boldsymbol{S}$ is a diagonal matrix with singular values of matrix $\boldsymbol{C_a}$. The inverse is given by $\boldsymbol{C_a^{-1}} = \boldsymbol{V \bar{S}^{-1} U}^T$. The analytical inverse of $\boldsymbol{S}$ can be found; however, this inverse gives a solution that is not very smooth and no information of interest can be gathered from it. Here $\boldsymbol{\bar{S}^{-1}}$ approximates the inverse of $\boldsymbol{S}$ and is given by

$$\bar{S}_{ii}^{-1} = \begin{cases} 1/\mathrm{S}_{ii} & \text{if } \mathrm{S}_{ii} > P_{SVD} \\ 0 & \text{otherwise} \end{cases}, i = 0, ..., N$$

P_{SVD} is a cut-off set to some percentage of the maximum singular value of $\boldsymbol{C_a}$. Smaller P_{SVD} give more accurate MTT values but $\boldsymbol{R}$ will be less smooth. P_{SVD} is usually set between 5% and 20% of the maximum singular value of $\boldsymbol{C_a}$. We can now solve for an estimate of $\boldsymbol{R}$, given by

$$\boldsymbol{\hat{R} = C_a^{-1} C_t \approx V \bar{S}^{-1} U^T C_t} \qquad (11)$$

From which we get our final estimate for MTT and CBF as

$$\widehat{MTT} = \frac{\Delta t \cdot \sum_{i=0}^{N} \hat{R}(t_i)}{max\hat{R}(t_i)} \tag{12}$$

$$\widehat{CBF} = max(\hat{R}(t_i)) \quad [15] \tag{13}$$

Different SVD algorithms

It was generally inferred from the first development of SVD deconvolution algorithms that the accuracy could be improved upon. What was described in the previous section is the most basic of the SVD deconvolution methods, termed either SVD or standard SVD (sSVD). Modifications exist that improve upon sSVD. Two documented algorithms are the block-circulant SVD (bSVD) and the oscillation minimization bSVD (oSVD) [15]. The bSVD algorithm extends the length of the TACs by adding zeros to the end giving a convolution matrix

$\boldsymbol{C_B} = \begin{bmatrix} \boldsymbol{C_a} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{C_a} \end{bmatrix}$ where $\boldsymbol{C_a}$ is defined as in the previous section and

$$\boldsymbol{B} = \begin{bmatrix} 0 & C_a(t_N) & C_a(t_{N-1}) & \dots & C_a(t_1) \\ 0 & 0 & C_a(t_N) & \dots & C_a(t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The system is then solved the same way as sSVD. Still the P_{SVD} cut-off must be chosen arbitrarily. To eliminate this the oSVD algorithm adaptively chooses singular value thresholds for each voxel, depending on the smoothness of $R(t)$ [16].

A problem arises from the fact that there is no standardized procedure for generating perfusion maps. There are not even standardized acronyms for the different algorithms. In addition, most vendors keep their software algorithms proprietary and do not release

information about them. Kudo et al. [7] studied the maps generated from 13 different software algorithms distributed by 7 different vendors. No two packages created the same maps. Other studies have documented the differences in different software implementations [5] [6]. How can we improve the transparency of the software and provide an open source tool for generating perfusion maps? The answer is the R package "CTP." But first, we need to verify the software maps the true values for CBF. We do so by first simulating a digital phantom.

2.2 The Digital Phantom (Simulation Details)

In order to evaluate the ability of the software to accurately estimate CBF, we performed a simulation study using an imaging phantom. Typically in radiological imaging studies, an actual physical phantom is constructed to represent physiological characteristics, such as CBF. This is very difficult, if not impossible, to do for blood flow. Instead, we chose to construct a digital phantom following the methods of Kudu et al. and Ostergaard et al. [7] [14]. The phantom was constructed entirely within R by the authors.

TACs for each pixel of the phantom were defined by the following equation:

$$C_t(t) = \frac{CBV}{MTT} \cdot [C_a(t) \otimes R(t)] - \tau \tag{14}$$

where CBV was set equal to 5 mL/100 g. Seven MTT values (3.4, 4.0, 4.8, 6.0, 8.0, 12.0, and 24.0 seconds) and seven τ values (0.0, 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0 seconds) were used, where τ represents arterial input delay. The ranges chosen for MTT span the range of brain tissue with very good blood flow (MTT = 3.4s) to dead tissue (MTT = 24.0s). The values for τ all represent physiologically valid arterial delays for human brains. A box-shaped $R(t)$ was used, and $C_a(t)$ was from a gamma-variate function, defined as:

$$R(t) = \begin{cases} 1 & \text{if t} \leq \text{MTT} \\ 0 & \text{if t} > \text{MTT} \end{cases}$$

$$C_a(t) = \begin{cases} 0 & \text{if t} \leq t_0 \\ \text{C}_0(t - t_0)^a e^{-(t-t_0)/b} & \text{if t} > \text{t}_0 \end{cases}$$

with constants C_0 equal to 1, t_0 equal to 12 seconds, a equal to 3, and b equal to 1.5. Gaussian noise was then added to the phantom to simulate a signal-to-noise ratio (SNR) of 5, a typical SNR for CT perfusion imaging scanned at 80 kVp and 200 mA, both typical scanner settings.

A visual representation of the phantom can be seen in Figure 2. Colors closer to black/purple/navy hues indicate lower blood flow; whereas, colors closer to orange and red hues indicate higher blood flow. It is arranged into 49 tiles each corresponding to a different combination of the seven MTTs and seven arterial delays. Decreasing MTT (corresponding to higher CBF), ranging from 24 seconds on the far left to 3.4 seconds on the far right, are displayed across the columns. Increasing arterial input delay (τ) are shown down the rows, with the shortest (0 seconds) shown on the top row and the longest (3 seconds) shown on the bottom row. Each tile contains 32 x 32 = 1024 pixels.

3 SOFTWARE

The software has been implemented and contained fully within the R computing environment. An open-source version is available and can be downloaded through the GitHub repository "SethLirette/CTP" and is licensed through the GNU General Public License v3.0 [17]. It requires five other R packages to be first installed: (1) "fields" (2) "oro.dicom" (3) "MASS" (4) "zoo" (5) "caTools." Once the "CTP" package is installed, the function "ctp_cbf" can be called to draw the perfusion map. We would like to stress that the program should be used only for research an illustrative purposes and should not be used to diagnose or treat any disease.

The function "ctp_cbf" allows 9 arguments, at least 4 of which are required to be specified. (1) *folder* points the function to the directory containing the set of CT images

(in DICOM, the industry standard, format) to be combined. The images must be named in alphanumerical order according to the order in which they were collected. This is a required argument. (2) *method* is a character string indicating estimation technique to be used ("maxslope","sSVD","bSVD", or "oSVD"). This is a required argument. (3) *aif* is a character string indicating the arterial input function to be used ("auto", "max", "gv", or "user"). This is a required argument. (4) *t0* is a parameter to be specified when aif="gv". It is the time in which contrast enters. It has no default value and is required if aif="gv". (5) *C0* is a parameter to be specified when aif="gv". It is a constant for the gamma variate (gv) function and defaults to 1. (6) *a* is a parameter to be specified when aif="gv". It is the shape parameter for the gamma variate (gv) function and defaults to 3. (7) *b* is a parameter to be specified when aif="gv". It is the rate parameter for the gamma variate (gv) function and defaults to 1.5. (8) *aif.user* denotes a user-defined vector for the arterial input function. It is required if aif="user". (9) *lag* is the time interval in seconds for collection of CT data. Usually set to 1 or 2. This is a required argument.

We will now demonstrate the call of the "ctp_cbf" function. This first call is pointed to a folder containing the images for hypothetical patient "001" at slice location 12. We declare that we wish to employ the max slope algorithm with a gamma variate input function and the contrast arriving at 10 seconds. We also declare that the images where taken at one second time intervals (lag=1).

```
>   ctp_cbf(folder="C:/.../r001/loc12", method="maxslope",
+   aif="gv", t0=10, lag=1)
```

The second example uses the images from slice location 3 of patient "083" and the oSVD algorithm. A gamma variate input function is called with the contrast arriving at 10 seconds and multiplying a constant value of 1.13. One again, we declare the images to have been collected at one second intervals.

```
>   ctp_cbf(folder="C:/.../r083/loc3", method="oSVD",
+   aif="gv", t0=10, C0=1.13, lag=1)
```

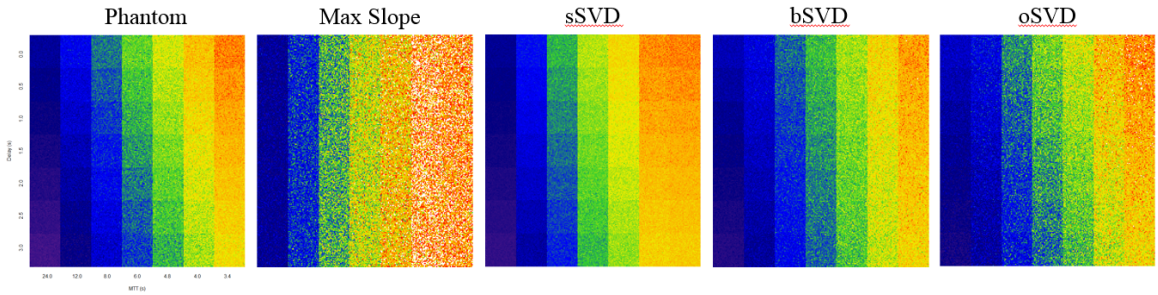

Figure 2: Results from simulation study. The section labeled "Phantom" represents the truth values for various delays and MTTs. The phantom is divided into 49 blocks each with a different combination for arterial delay and MTT. The arterial delays on the rows are (from top to bottom, in seconds) 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0. The MTTs on the columns are (from left to right, in seconds) 24.0, 12.0, 8.0, 6.0, 4.8, 4.0, and 3.4. The upper-rightmost block is indicative of the highest blood flow, and the lower-leftmost block is indicative of the lowest blood flow. To the right are the results of applying the four different perfusion estimation algorithms to the phantom.

Our final example is identical to the previous example, except it demonstrates how a user would declare his or her own arterial input function ("svec"), a vector of length 27. This vector named "svec" is then used in the call of "ctp_cbf."

```
>   svec = c(rep(50,11),51,211,
+  365,445,446,396,323,247,181,127,87,58,rep(50,4))
>   ctp_cbf(folder="C:/.../r083/loc3", method="oSVD",
+   aif="user", aif.user=svec, lag=1)
```

4 SIMULATION RESULTS

The results of the simulation study are shown in Figure 2. All four of the software algorithms produced a map that is subjectively similar to the phantom. To reiterate, the phantom is organized to reflect varying MTTs and arterial delays, with decreasing MTT (improved flow) going from left to right and increasing arterial input delays going from top to bottom.

As expected, the max slope algorithm produced a map that was completely insensitive to arterial delay as shown by its uniformity when looking down the rows. Further, the max slope algorithm tended to overestimate blood flow, especially at MTT less than 12

seconds. It also gave a much more noisy image.

The SVD algorithms had a marked improvement over the max slope algorithm. All three SVD algorithms displayed the ability to account for arterial delay, with bSVD and oSVD performing marginally better than sSVD. Similar to max slope, sSVD tended to overestimate blood flow at MTT less than 12. Also similarly to max slope, sSVD failed to discriminate the difference between an MTT of 4 seconds and an MTT of 3.4 seconds, as displayed in the uniformity of the two right most columns. Both bSVD and oSVD gave very good estimations at all levels of delay and MTT. If anything, bSVD tends to very slightly underestimate blood flow at MTT less than 6 seconds. The oSVD algorithm gave very good estimation results, with almost no distinguishable differences than the phantom. The tradeoff was a run time three times as long as bSVD (10.6 seconds vs. 3.3 seconds). Max slope had a run time of 1.1 seconds, and sSVD had a run time of 0.6 seconds.

5 IN VIVO APPLICATION

The software was then applied to in vivo CT data. Two illustrative examples are shown in Figure 3. Both patients were admitted to the emergency room at the Univesity of Mississippi Medical Center with stroke-like symptoms, triggering a code grey protocol, during which CT perfusion and MRI data were collected.

Patient A was a 72 year-old male who triggered the code grey protocol at 12:47 PM. As the CBF map shows (Figure 3A), he had good perfusion throughout the entire brain. MRI confirmed the absence of stroke. Patient B was a 62 year-old male who triggered the code grey protocol at 2:45 PM. From Figure 3B, we see reduced CBF, particularly on the patient's right side. The map is drawn using the neurological convention (the left side of the image is the patient's right side of the body). MRI confirmed that this patient had a large vessel stroke located in the right anterior cerebral artery. Both of these maps were drawn using the oSVD algorithm. Using the proposed R program, the computational time for Patient A was 54.9 seconds and for Patient B was 54.4 seconds.

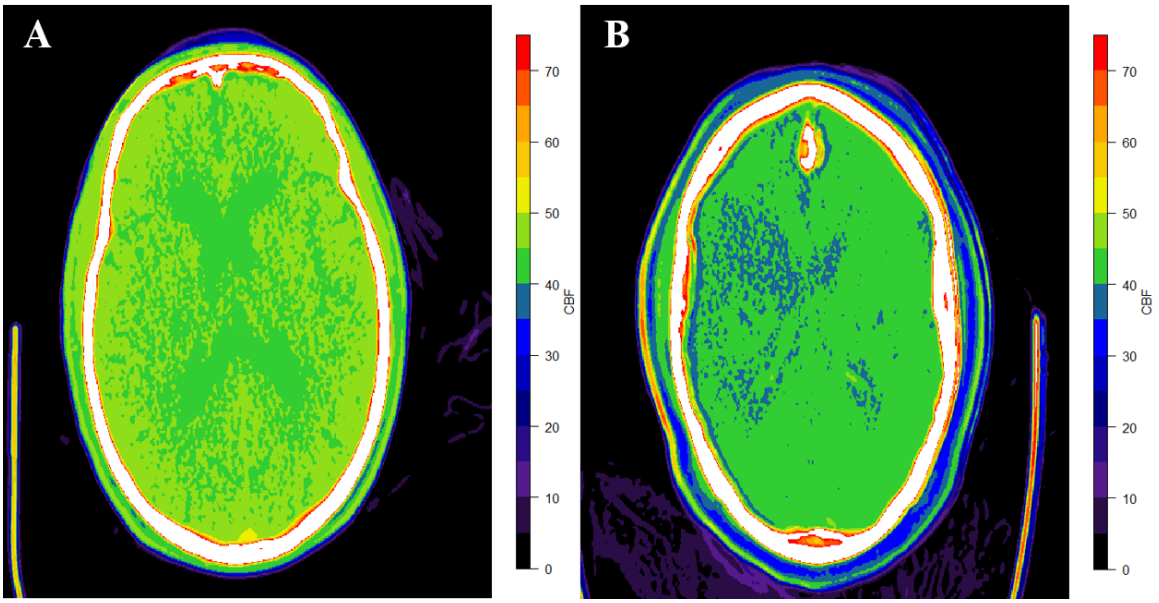

Figure 3: Results from in vivo CTP data. Patient A has normal blood flow as confirmed by no evidence of stroke on MRI. Patient B had a confirmed large vessel stroke located in the anterior cerebral artery.

6 DISCUSSION

We have demonstrated that the it is not only possible, but practical to create an open-source piece of software that can accomplish the goal of creating a CBF perfusion map. With this tool, we were able to compare the different methods of estimating perfusion and compare with a digital phantom to evaluate which method seems to perform the best. This tool is, and will continue to be, free to the public for research and illustrative purposes. Anyone with the ability to arrange a sequence of images into alphanumerical order and can download R can now create a perfusion map without having to purchase expensive software and with complete transparency as to the algorithm used.

But this supposes that the user has a most basic understanding of R. How can we go further to alleviate even that problem? In the future iterations of the software, we would like to create a Shiny application of the software [18]. Shiny is a framework to turn R functions and scripts into web applications. Running a Shiny application that uses the functions available in "CTP", anyone that can use a web browser will be able to create a perfusion

map. This is truly exciting!

It is worth mentioning some limitations of the software in its current form. While the "ctp_cbf" function allows options for the max slope, sSVD, and bSVD algorithms, the authors recommend that only the oSVD algorithm be employed in practice. While the other three worked well in the phantom study, during in vivo applications, we have observed that only the oSVD algorithm gives satisfactory results. The other algorithms typically do not provide much discriminant ability across the various regions of brain tissue (as represented by the pixels of the perfusion map). Of course, the function can be used to display the differences between actual applications of the four different methods. This brings to the second potential improvement. The oSVD algorithm takes approximately one minute to process on a standard laptop computer. We would like to get this processing time down and think it can be done with more efficient coding. As a final limitation, and probably the most profound, is the importance of selecting a good arterial input function, which plays an important role as an input into the deconvolution equation used to estimate the perfusion parameters. The difficulty of doing this has been echoed by leaders in the field [19]. For example, arteries are currently selected by hand by highly trained technicians, although more automated methods have been more recently proposed. An attempt was made to include the methods proposed by Mouridsen et al., which attempts to automatically choose an arterial input by subsetting the pixels and using multiple levels of k-means clustering [20]. This first of all made the program extremely slow, multiplying the processing time by a factor of five. It also failed to automatically choose an artery with high precision. We chose to leave the implementation in the software (it can be used by declaring `aif="auto"` in the function call) as an illustrative tool; however, we recommend the user defines an arterial input function either by using the gamma variate option or the pure user-defined option.

All of the methods used to create the perfusion maps in this paper are common methods. They all consist of distilling the information contained within $R(t)$ into one single measure, in our case CBF. This leads us to ask the question: how can we best use every

piece of information contained within $R(t)$? We will answer this question in the next paper and take a decidedly statistical stance. We will present four potential statistical modeling techniques: generalized linear mixed models, generalized estimating equations, a two stage model based on empirical Bayes, and generalized functional linear models.

REFERENCES

[1] CDC, "Stroke." `http://www.cdc.gov/stroke`. Accessed February 22, 2016.

[2] K. Miles, M. Hayball, and A. Dixon, "Colour perfusion imaging: a new application of computed tomography," *The Lancet*, vol. 337, no. 8742, pp. 643–645, 1991.

[3] B. F. Tomandl, E. Klotz, R. Handschu, B. Stemper, F. Reinhardt, W. J. Huk, K. Eberhardt, and S. Fateh-Moghadam, "Comprehensive imaging of ischemic stroke with multisection ct," *Radiographics*, vol. 23, no. 3, pp. 565–592, 2003.

[4] J. J. S. Shankar, C. Lum, *et al.*, "Whole brain ct perfusion on a 320-slice ct scanner," *Indian Journal of Radiology and Imaging*, vol. 21, no. 3, p. 209, 2011.

[5] K. Kudo, M. Sasaki, K. Yamada, S. Momoshima, H. Utsunomiya, H. Shirato, and K. Ogasawara, "Differences in ct perfusion maps generated by different commercial software: Quantitative analysis by using identical source data of acute stroke patients," *Radiology*, vol. 254, no. 1, pp. 200–209, 2009.

[6] K. Kudo, M. Sasaki, K. Ogasawara, S. Terae, S. Ehara, and H. Shirato, "Difference in tracer delay–induced effect among deconvolution algorithms in ct perfusion analysis: Quantitative evaluation with digital phantoms," *Radiology*, vol. 251, no. 1, pp. 241–249, 2009.

[7] K. Kudo, S. Christensen, M. Sasaki, L. Østergaard, H. Shirato, K. Ogasawara, M. Wintermark, and S. Warach, "Accuracy and reliability assessment of ct and mr

perfusion analysis software using a digital phantom," *Radiology*, vol. 267, no. 1, pp. 201–211, 2013.

[8] M. Sasaki, K. Kudo, T. Boutelier, F. Pautot, S. Christensen, I. Uwano, J. Goodwin, S. Higuchi, K. Ito, and F. Yamashita, "Assessment of the accuracy of a bayesian estimation algorithm for perfusion ct by using a digital phantom," *Neuroradiology*, vol. 55, no. 10, pp. 1197–1203, 2013.

[9] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[10] A. Konstas, G. Goldmakher, T.-Y. Lee, and M. Lev, "Theoretic basis and technical implementations of ct perfusion in acute ischemic stroke, part 1: theoretic basis," *American Journal of Neuroradiology*, vol. 30, no. 4, pp. 662–668, 2009.

[11] R. Wirestam, L. Andersson, L. Ostergaard, M. Bolling, J.-P. Aunola, A. Lindgren, B. Geijer, S. Holtås, and F. Ståhlberg, "Assessment of regional cerebral blood flow by dynamic susceptibility contrast mri using different deconvolution techniques," *Magnetic resonance in medicine*, vol. 43, no. 5, pp. 691–700, 2000.

[12] N. A. Mullani and K. L. Gould, "Re: First-pass measurements of regional blood flow with external detectorsreply," *Journal of Nuclear Medicine*, vol. 25, no. 7, pp. 831–836, 1984.

[13] K. Miles and M. Griffiths, "Perfusion ct: a worthwhile enhancement?," *The British journal of radiology*, vol. 76, pp. 220–231, 2003.

[14] L. Østergaard, R. M. Weisskoff, D. A. Chesler, C. Gyldensted, and B. R. Rosen, "High resolution measurement of cerebral blood flow using intravascular tracer bolus passages. i. mathematical approach and statistical analysis," *Magnetic Resonance in Medicine*, vol. 36, no. 5, pp. 715–725, 1996.

[15] J. Wang, O. Masoud, and B. Frake, "Delay insensitive svd algorithm for perfusion analysis," Apr. 11 2008. US Patent App. 12/082,695.

[16] O. Wu, L. Østergaard, R. M. Weisskoff, T. Benner, B. R. Rosen, and A. G. Sorensen, "Tracer arrival timing-insensitive technique for estimating flow in mr perfusion-weighted imaging using singular value decomposition with a block-circulant deconvolution matrix," *Magnetic resonance in medicine*, vol. 50, no. 1, pp. 164–174, 2003.

[17] "Gnu general public license, version 3." `https://www.gnu.org/licenses/gpl-3.0.en.html`, June 2007. Accessed October 31, 2016.

[18] RStudio, Inc, *Easy web applications in R.*, 2013. URL: `http://www.rstudio.com/shiny/`.

[19] F. Calamante, "Arterial input function in perfusion mri: a comprehensive review," *Progress in nuclear magnetic resonance spectroscopy*, vol. 74, pp. 1–32, 2013.

[20] K. Mouridsen, S. Christensen, L. Gyldensted, and L. Østergaard, "Automatic selection of arterial input function using cluster analysis," *Magnetic resonance in medicine*, vol. 55, no. 3, pp. 524–531, 2006.

THE PREDICTIVE ABILITY AND COMPUTATIONAL BURDEN OF FUNCTIONAL VS. LONGITUDINAL MODELS FOR SCALAR OUTCOMES

SETH T. LIRETTE, INMACULADA B. ABAN, AND COAUTHORS

In preparation for *Computational Statistics and Data Analysis*

Format adapted for dissertation

1 INTRODUCTION

Depending upon who one chooses to believe, either Legendre in his work *Nouvelles méthodes pour la détermination des orbites des comètes* or Gauss in his work *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss* invented/first described the method of least squares [1] [2] [3]. These two titans of science were concerned with predicting the orbits of the heavenly bodies, but they unleashed a tool that fellow practitioners of science would be using for prediction for centuries to come. The linear regression model, which is, in its most basic form, fit by the method of least squares, has been used for modeling almost any natural phenomena imaginable. The work of Galton, Edgeworth, Yule, and Pearson later generalized the method and provided the assumptions for today's ordinary least squares (OLS) model [3].

Of course, over the last few decades the framework has been generalized even further. First, Nelder and Wedderburn formulated a beautiful set of methods for no longer requiring the response variable to be Gaussian distributed [4]. Somewhat distinctly and simultaneously, mixed models for correlated or clustered data were developed over many decades. West et al. give a wonderful brief history into the development of the mixed model [5]. Semiparametric methods for clustered data then arose [6] [7].

More recently, there has been much interest in the field of functional data analysis, the definition of which we will undertake in section 3 of this paper. Whereas a typical regression problem has a scalar predictor and a scalar outcome, or scalar-on-scalar regression, functional regression can take any of three forms: (1) function-on-scalar regression (2) function-on-function regression or (3) scalar-on-function regression. We were

presented with data of the third form, but before we analyzed the data, we first needed to know what technique would be most appropriate. This paper seeks to address that concern while also investigating the computational burden of each method. The paper is arranged as follows. In section 2, we outline the motivating problem and the potential to generalized to other data. We then describe potential modeling frameworks in section 3. Then we set forth our simulation framework and present simulation results in sections 4 and 5. And finally, we offer a general discussion about the results, implications, and plans for future research in section 6.

2 MOTIVATING PROBLEM AND GENERALIZATION TO OTHER DATASETS

The motivation for this paper was initiated when a clinical investigator presented a set of data that was structured in a peculiar way. In an IRB approved, HIPAA compliant, retrospective study, 93 patients had received computed tomography perfusion (CTP) studies due to a suspected stroke. "Whole" brain attenuation (measured in Hounsfield Units) was gathered at 27 time points. Figure 1A shows one patient's time/attenuation trajectory, and figure 1B plots the same for all 93 patients. The data also contained the absence or presence of an ischemic stroke, validated by MRI. The clinical question was essentially this: Can we use all of the information contained in the 27 attenuation values for each participant to construct a probability of ischemic stroke? This paper will explore the statistical techniques used to answer this question. An alternative continuous example is using ambulatory blood pressure measured every 10 minutes over a 24 hour period as the functional predictor for predicting some continuous outcome such as glomerular filtration rate. We will address the scenario for continuous outcomes as well.

Like many other instances in the scientific process, especially those involving statistical modeling, we had to first choose a modeling technique to apply to these data. Given the scalar outcome and longitudinal (or functional) form of the predictor, a population average model might be a good initial choice. But there have been many advances in recent

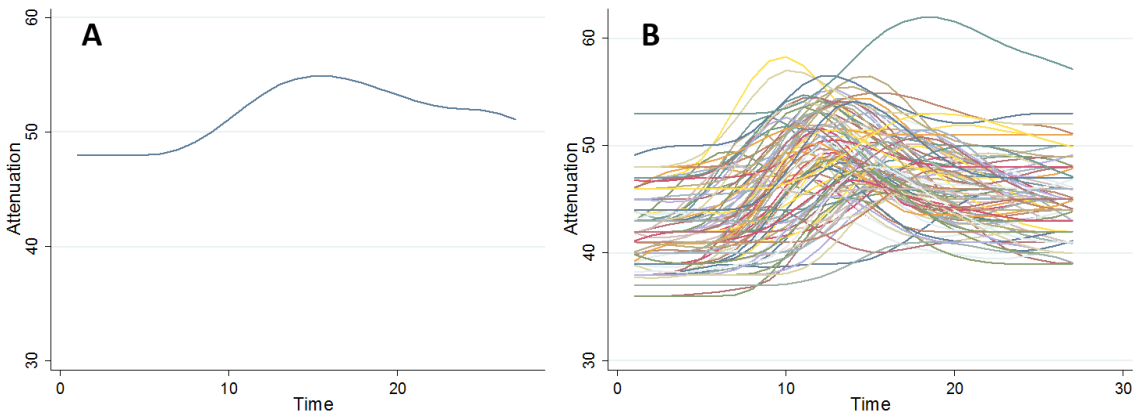

Figure 1: Time-attenuation curve for (A) one patient (B) all 93 patients

years in the field of functional data analysis, starting with Ramsay and Silverman [8] [9]. More recently, the functional generalized linear model has been developed [10] [11] [12] [13]. This leads us to ask the question: When presented with scalar outcome data on individuals, each with dozens of time points on the predictor, what modeling technique is both most appropriate and best utilizes computational resources? This paper seeks to answer that question for both binary outcomes and continuous outcomes.

3 MODELS

3.1 Generalized Linear Models

When starting to construct a model to predict either a continuous or binary outcome and applying Occam's Razor, one of the most natural places to begin is the generalized linear model (GLM). A common formulation of the GLM is given in equation 1.

$$g(\mu) = \beta_0 + \beta_1^T \mathbf{X} \tag{1}$$

where $\mu = E(Y|\mathbf{x}))$ and g is the link function.

Although a variety of distributional families for Y and link functions for g can be employed, the most common are $Y \sim$ Gaussian with identity link function when the outcome is continuous and $Y \sim$ Bernoulli with logit link for a binary outcome. We see no

need to deviate from these canonical models.

Using the most basic model, the GLM, we immediately encounter a problem. One of the assumptions – arguably the most important – of the GLM is the assumption of independence or exchangeability across the rows of the dataset. This assumption is obviously violated in the data structures we are interested in studying. We will now present three models that account for this.

3.2 Generalized Estimating Equations

Generalized Estimating Equations (GEE) were first described by Liang and Zeger as a longitudinal data analysis method [6]. It is a general method for analyzing repeated measure data where the subject-level observations are correlated, but observations across subjects are independent. GEE fits marginal or population-average models, where the goal is to make inference and prediction about the population while accounting for within-subject correlation. This is distinct from the subject-specific models such as generalized linear mixed models. Given that we are looking at repeated measures for the predictor set with a scalar outcome, we are only concerned with population average models in this paper.

Let i independent observations index outcome $Y_i = (Y_{i1}, ..., Y_{ij}, ..., Y_{in_i})$, then $\mathrm{E}(Y_{ij} = \mu_{ij})$ is related to predictor set x by

$$g(\mu_{ij}) = x_{ij}^T \beta \tag{2}$$

Let R_i denote the working correlation matrix and let

$$V_i = A_i^{1/2} R_i A_i^{1/2} \tag{3}$$

where A_i is a diagonal matrix with elements $Var(Y_{ij})$. The estimate $\hat{\beta}$ is then the solution to the equation

$$\sum_i \frac{\partial \mu_i^T}{\partial \beta} V_i^{-1} (Y_i - \mu_i) = 0. \tag{4}$$

Many different correlation structures can be specified for R_i, but given how close together our time points are sampled (2 seconds, equally spaced) an autoregressive of order one correlation structure is our choice. This is specified as

$$R = \begin{bmatrix} 1 & \rho & \dots & \rho^{N-1} \\ \rho & 1 & \dots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \dots & 1 \end{bmatrix}.$$

Like GLM, we can specify family and link functions for Y and g and will continue with the choices described in that section.

3.3 A Two-Stage Model Using the Best Linear Unbiased Predictor

Since we have an outcome that is time-invariant, we are relegated to using marginal or population average models. However, our predictor set is time-varying. This begs the question: Can we port the information contained in the subject-specific values into a population average model? Stanek et. al asked this first, formulating the question as: "Why not routinely use best linear unbiased predictors (BLUPs) as estimates of cholesterol, per cent fat from kcal and physical activity?" [14]. They argue that the best linear unbiased predictor (BLUP) should routinely be used in such situations. Morrell et. al then applied the same strategy to improve estimation for logistic models [15]. The model most closely resembling our two-stage model was constructed by Shetty et. al [16]. The BLUPs are the subject-specific predictions using both fixed and random effects from a linear mixed model. These predicted values are then used as the predictors in a marginal model. More specifically the modeling procedure, using the motivating data, is:

1. Fit a random intercept-slope model with attenuation as the outcome and time as the predictor and obtain subject-specific parameter estimates from the model
2. Fit GEE model using intercept and slope estimates from Step 1 using autoregressive of order 1 correlation with appropriate family and link functions

3.4 Functional Linear Regression That's Interpretable

In most instances when we think of data or the modeling of data, we usually envision a number of variables taken from a sample of individuals with one observation per subject in the cross-sectional setting and repeated measurements per subject, measured at "discrete" time points, in a longitudinal study. These would perhaps be modeled ordinarily by GLMM or some marginal method such as GEE. Functional data analysis (FDA) encompasses a different way to conceptualize a datum. In FDA, the most basic unit of information on a subject is an entire observed curve or function. Functional regression and its analogue, the generalized functional linear model (GFLM), are two very useful tools in FDA. Ramsay and Silverman and Muller and Stadtmuller provide a variety of examples and applications [8] [11]. Each technique under the GFLM umbrella falls into one of three categories: (1) function-on-scalar regression (2) function-on-function regression or (3) scalar-on-function regression. Our focus will be on the last.

Scalar-on-function regression is typically expressed by the following model

$$Y_i = \beta_0 + \int \beta(t) X_i(t) dt + \epsilon_i. \tag{5}$$

The coefficient function $\beta(t)$ is usually represented in one of three ways. The first method represents $\beta(t)$ as some *p*-dimensional basis function. The second method attempts to shrink the variability in $\beta(t)$ via some least squares penalty. The third method, and our method of choice takes a variable selection approach to solving the problem. James et. al have named this method "Functional Linear Regression That's Interpretable" (FLiRTI) [12]. FLiRTI is an attractive choice because it gives accurate predictions, is flexible, has nice theoretical properties, is computationally efficient, and unlike the first two methods, gives an estimate for the coefficient function $\beta(t)$ that is interpretable. Regarding the last advantage, because FLiRTI is a shrinkage procedure, it effectively "zeros" the time points that do not play a significant role in predicting the outcome while retaining time points

that have significant effects. James displays this feature well when applying FLiRTI to the Canadian Weather Data in the original FLiRTI paper [12]. This paper focuses on the predictive accuracy of FLiRTI.

We will give a brief overview of the FLiRTI model and model fit. Full details can be seen in James et. al [12]. The FLiRTI model borrows an idea from the basis approach but allows errors in the $\beta(t)$ specification. Specifically

$$Y_i = \beta_0 + \mathbf{X}_i^T \eta + \epsilon_i^*. \tag{6}$$

where $\mathbf{X}_i = \int X_i(t)\mathbf{B}(t)dt$, $\epsilon_i^* = \epsilon_i + \int X_i(t)e(t)dt$, and the p-dimensional basis $\mathbf{B}(t) = [b_1(t), b_2(t), ..., b_p(t)]^T$. If we define

$$\boldsymbol{\gamma} = A\eta \tag{7}$$

where A is a vector combining finite difference operators with the basis functions, we can combine (6) and (7) to get the FLiRTI model

$$\mathbf{Y} = XA^{-1}\boldsymbol{\gamma} + \boldsymbol{\epsilon}^*. \tag{8}$$

The model is then fit using either the LASSO selector via the LARS algorithm or the Dantzig selector via the DASSO algorithm to obtain $\hat{\boldsymbol{\gamma}}$ [17] [18] [19] [20]. The FLiRTI estimate for $\beta(t)$ is then given by

$$\hat{\beta}(t) = \mathbf{B}(t)^T A^{-1}\hat{\boldsymbol{\gamma}}. \tag{9}$$

FLiRTI has tuning parameters that correspond to tuning parameters in any smoothing situation. The most common approach to choosing these is to choose the combination that produce the lowest cross-validated residual sum of squares, and this is the approach we take in our simulation study. For inference, one can obtain bootstrapped confidence intervals for $\beta(t)$, and James et. al describe a permutation test to test for statistical significance

of the relationship between $X(t)$ and Y [12].

4 SIMULATION DETAILS

To better understand which model would exhibit the best predictive performance under a variety of circumstances, we simulated 500 datasets for each combination of sample sizes of 15, 30, 50, 100, 500, 1000, and 5000 and number of time points equal to 3, 5, 10, 25, 50, 100, and 250. Each simulated dataset was constructed as follows:

1. For subject i, draw one J-variate sample from a multivariate normal distribution with mean zero and covariance matrix Σ, where Σ is defined by covariance function

$$\text{cov}(t_p, t_q) = \exp(-\tfrac{1}{2}|t_p - t_q|^2)$$

   with t_p and t_q being the elements of the vector of time points **t**

2. Repeat Step 1, using a different random seed, for all N subjects

3. Generate an observation level random intercept $u_i \sim N(0, 2)$

4. Generate scalar outcome

   (a) For continuous outcomes: $y_i = S[x_i] + u_i$

   (b) For binary outcomes: $Y \sim Bern(p), p = \dfrac{1}{1 + \exp[-(-2 + 0.005S[x_i] + u_i)]}$

   where $S[x_i] = \text{Mean}[x_i] + \text{SD}[x_i] + \text{Skewness}[x_i] + \text{Kurtosis}[x_i]$.

Essentially, we are projecting each (t_p, t_q) pair onto a spline basis. This corresponds to the reason for choosing the covariance function as stated in step 1. It can be shown that the squared exponential covariance function is equivalent to a regression model with infinite basis functions [21], and this assures that we draw from a smooth Gaussian process. Additionally, the covariance between adjacent cells is almost 1 and decreases as distance between cells increases, similar to an autoregressive structure. The choice for the outcome formula was chosen to give each datum a subject-specific nonlinear trajectory while also

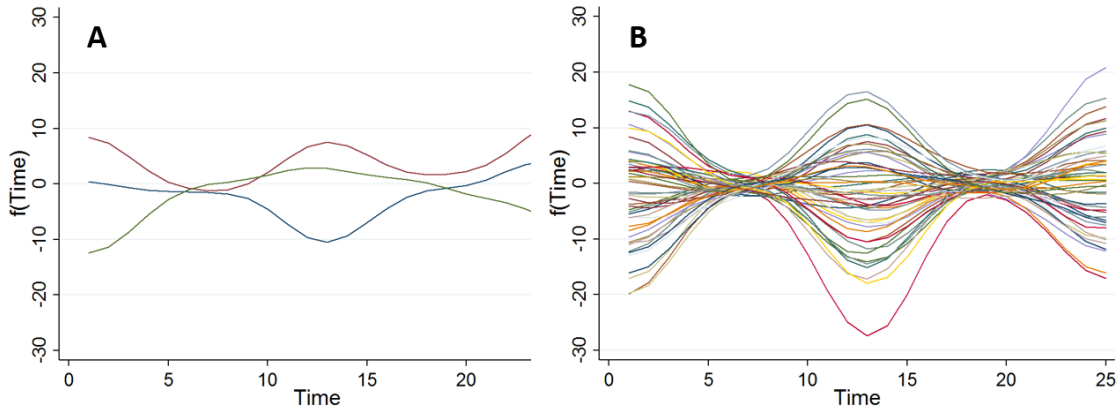

Figure 2: Results of simulation from a random sample of 25 time points for (A) N=3 functions and (B) N=50 functions

providing some spread in the outcome of interest. This approach is similar to the simulation approach taken by Gertheiss et al. and Chen et al. [22] [23]. The R code for generating these data can be found in Appendix B.

Figure 2A shows the results of the simulating process for three randomly selected simulated functions for 25 observed time points. As you can see, these functional forms are particularly smooth. Figure 2B shows every function for sample size equal to 50 with 25 time points each. Tables 1 and 2 display summary statistics for the simulated continuous and binary outcomes, respectively. Table 1 shows means with standard deviation in parentheses. The first row of Table 2 shows the mean probability of a positive outcome with the minimum and maximum probability in brackets. The second row of each cell displays the number of positives with the percentage.

Table 1: Summary statistics for continuous outcomes

Sample Size (N)	Number of Time Points (J)						
	3	5	10	25	50	100	250
15	7.4 (7.9)	7.4 (5.4)	6.6 (6.0)	8.3 (5.1)	10.4 (4.1)	7.3 (2.9)	9.7 (4.5)
30	9.6 (7.3)	9.7 (6.6)	9.3 (5.1)	9.1 (4.9)	8.1 (5.1)	6.4 (3.6)	7.5 (4.8)
50	10.9 (7.7)	9.6 (5.8)	9.7 (5.0)	8.4 (4.2)	6.3 (4.4)	6.0 (4.4)	9.0 (5.2)
100	8.4 (6.3)	7.5 (5.0)	6.8 (4.0)	6.0 (4.0)	7.3 (4.5)	8.3 (5.1)	7.6 (4.9)
500	10.0 (7.7)	8.9 (5.6)	8.0 (4.8)	7.9 (4.9)	8.0 (4.8)	7.7 (4.8)	7.7 (4.8)
1000	9.4 (7.4)	8.5 (5.4)	7.7 (5.0)	7.6 (4.9)	7.5 (4.8)	7.5 (4.8)	7.6 (4.7)
5000	10.0 (7.4)	8.9 (5.5)	8.1 (5.0)	7.9 (4.9)	7.8 (4.8)	7.7 (4.8)	7.7 (4.7)

Table 2: Summary statistics for binary outcomes

Sample Size (N)	Number of Time Points (J)						
	3	5	10	25	50	100	250
15	0.25 [0.01, 0.80] 5 (33%)	0.14 [0.03, 0.53] 5 (33%)	0.33 [0.01, 0.97] 2 (13%)	0.29 [0.02, 0.89] 3 (20%)	0.28 [0.01, 0.95] 4 (27%)	0.17 [0.03, 0.73] 3 (20%)	0.27 [0.01, 0.61] 3 (20%)
30	0.25 [0.01, 0.89] 8 (27%)	0.24 [0.01, 0.86] 4 (13%)	0.33 [0.02, 0.96] 6 (20%)	0.27 [0.01, 0.93] 7 (23%)	0.16 [0.01, 0.80] 11 (37%)	0.20 [0.01, 0.87] 11 (37%)	0.23 [0.01, 0.89] 4 (13%)
50	0.28 [0.01, 0.94] 13 (26%)	0.27 [0.01, 0.94] 7 (14%)	0.26 [0.01, 0.98] 6 (12%)	0.25 [0.01, 0.79] 10 (20%)	0.20 [0.01, 0.87] 8 (16%)	0.18 [0.01, 0.88] 14 (28%)	0.24 [0.01, 0.96] 9 (18%)
100	0.25 [0.01, 0.92] 16 (16%)	0.22 [0.01, 0.89] 23 (23%)	0.26 [0.01, 0.97] 17 (17%)	0.19 [0.01, 0.92] 26 (26%)	0.24 [0.01, 0.92] 15 (15%)	0.23 [0.01, 0.99] 24 (24%)	0.21 [0.01, 0.96] 17 (17%)
500	0.24 [0.01, 0.97] 113 (23%)	0.23 [0.01, 0.95] 121 (24%)	0.24 [0.01, 0.99] 104 (21%)	0.21 [0.01, 0.97] 126 (25%)	0.24 [0.01, 0.99] 96 (19%)	0.22 [0.01, 0.96] 127 (25%)	0.23 [0.01, 0.99] 110 (22%)
1000	0.23 [0.01, 0.99] 260 (26%)	0.24 [0.01, 0.98] 257 (26%)	0.23 [0.01, 0.98] 239 (24%)	0.23 [0.01, 0.99] 250 (25%)	0.22 [0.01, 0.98] 246 (25%)	0.22 [0.01, 0.99] 224 (22%)	0.24 [0.01, 0.99] 229 (23%)
5000	0.23 [0.01, 0.99] 1166 (23%)	0.23 [0.01, 0.99] 1126 (23%)	0.23 [0.01, 0.99] 1113 (22%)	0.23 [0.01, 0.99] 1176 (24%)	0.24 [0.01, 0.99] 1135 (23%)	0.23 [0.01, 0.99] 1145 (23%)	0.23 [0.01, 0.99] 1107 (22%)

5 EVALUATION METRICS

Given that we are concerned with prediction, our reporting of the simulation results focuses on prediction accuracy. Furthermore, we are also interested in the practical application of the models, therefore we also examined the time taken to run each model in each of the scenarios. All models were run on a Dell Latitude E5570 with 16 GB RAM and Intel i7 quad-core at 2.60 GHz running 64-bit Windows 7. Models were fit in R and more specifically using the `glm` function from base R, the `geeglm` function from the `geepack` library, the `lmer` function from the `lme4` library, and the `flrti` function from the `flrti` library [24] [25] [26] [27]. We now will discuss our measures of prediction accuracy.

For continuous outcomes, we consider two measures of prediction accuracy, the mean absolute error (MAE) and root mean standard error (RMSE). RMSE gives a general form of the loss function and assigns worse errors to predictions further away from the mean by squaring the residuals. MAE, on the other hand, considers all errors to be weighted the same and is most robust to outlying errors. Considering both loss functions did not change substantive conclusions. MAE and RMSE are mathematically represented as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} = (y_i - \hat{y}_i)^2} = \sqrt{(\text{Bias}(\hat{y}_i))^2 + \text{Var}(\hat{y}_i)} \tag{10}$$

For binary outcomes, most modeling techniques yield predicted probabilities, and a few options for defining predictive accuracy were considered. Perhaps the most common and intuitive option is setting some threshold for yes/no (say 0.50) and define any predicted probability below this as a 0 and above as a 1. We chose not to employ this option for multiple reasons. It depends on an arbitrary threshold choice, it requires strong assumptions of correct model fit, and it discards a lot of information. For instance, the probability comparison of 0.01 vs. 0.99 is equivalent to 0.49 vs. 0.51.

We also considered using area under the receiver operating characteristics curve

(AUC). While better than the previous option, we ultimately decided not to use the AUC because it quantifies the ordering of the predicted probabilities, not their accuracies. For instance, five observations with predicted probabilities 0.01, 0.03, 0.05, 0.07, and 0.09 would have the same AUC if their predicted probabilities were 0.1, 0.3, 0.5, 0.7, and 0.9 even though these two sets are very discrepant.

Ultimately, we decided to use the Brier score, given as

$$BS = \frac{1}{n}\sum_{i=1}^{n}(P(Y_i = 1) - y_i)^2 \quad [28]. \tag{11}$$

This score function is analogous to the RMSE for continuous outcomes. On this scale, a Brier score of zero is the best score possible (either a predicted probability of 1.0 and an observed outcome of 1 or a predicted probability of 0.0 and an outcome of zero). On the other hand, one is the worst score possible. Also notice that a predicted probability of 0.5 will yield a Brier score of 0.25 regardless of the observed outcome.

6 RESULTS

Continuous results evaluated by MAE and RMSE are displayed in figures 3 and 4. Binary results characterized by Brier scores are shown in figure 5. For MAE and RMSE, similar patterns are seen for both. As expected, increasing sample size makes for more precision, even if biased predictions. For each of GLM, GEE, and the two-stage model, within each sample size, increasing the number of sampled time point reduces error, this hold for the FLiRTI model at samples sizes greater than 30. There are two curious insights drawn from the FLiRTI model. First, for N>J (i.e. number of subjects is greater than the number of time points), FLiRTI behaves as expected, with a reduction in error for increasing numbers of sampled time points, but when N<J, the errors start to increase with an increase in sampled time points. Take for example sample size 15, we see first a decrease in error with an increase in sampled time points until the 25 time points simulation, then there is a slight uptick in error for number of time points 50, 100, and 250. Although less

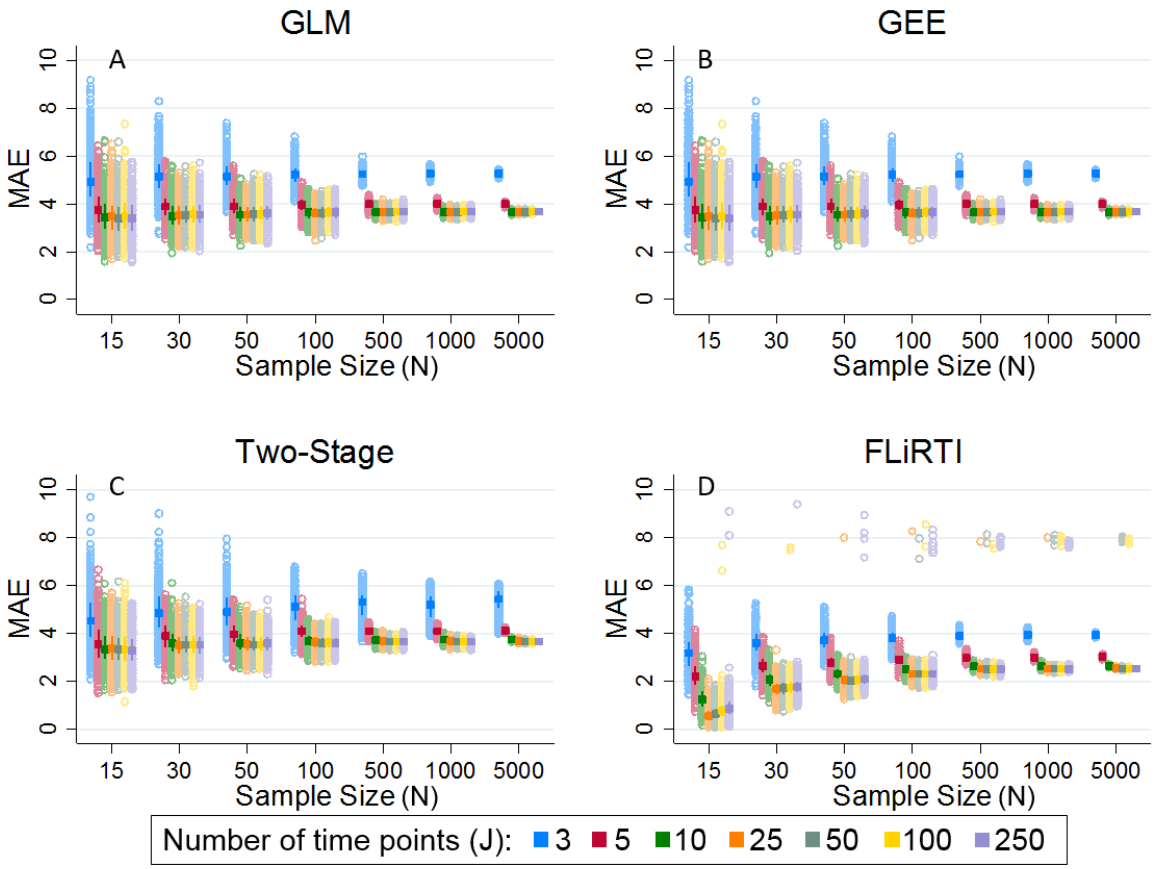

Figure 3: Simulation results showing mean absolute error (MAE) across sample sizes and number of time points for (A) generalized linear model (GLM) (B) generalized estimating equation (GEE) (C) two-stage model (D) functional regression model (FLiRTI). Each open circle represents the result from one model on one simulated dataset. Each N, J combination was simulated 500 times. Squares represent medians and bars connect the first and third quartile.

pronounced, this behavior is observed for N=30, J$\geq$50 and N=50, J$\geq$100. This behavior is expected, as FLiRTI essentially applies a variable selection procedure to the functional data, and while variable selection procedures such a LASSO and DASSO, can fit models where the sample size is less than the number of variables, the prediction error does increase in those cases. Second, increasing the sample size seems to slightly increase the median error. This also is to be expected. The authors of the original FLiRTI methodology paper state that choosing N<<J allows the FLiRTI algorithm to further minimize ϵ_i in equation 5. Nevertheless, as an overall conclusion, we can see that the FLiRTI model performed better than all other models in each scenario.

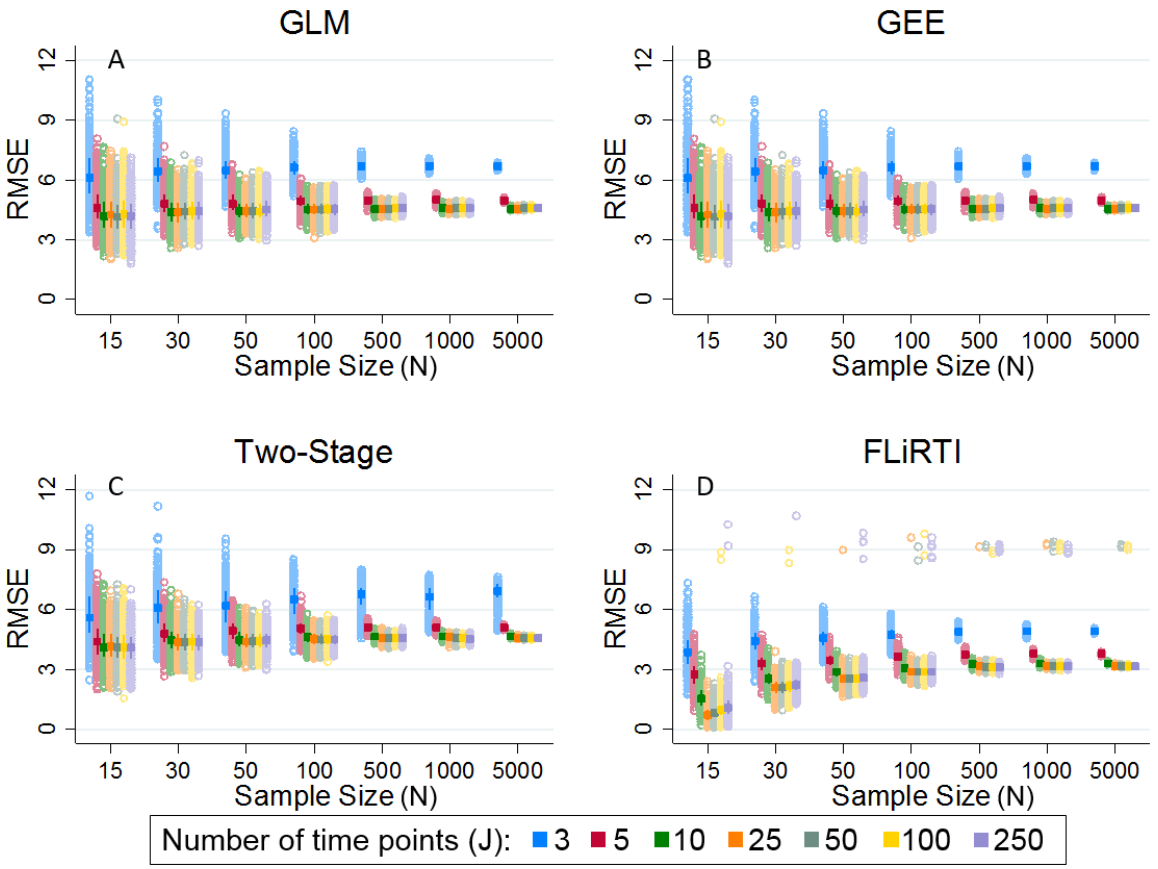

Figure 4: Simulation results showing root mean standard error (RMSE) across sample sizes and number of time points for (A) generalized linear model (GLM) (B) generalized estimating equation (GEE) (C) two-stage model (D) functional regression model (FLiRTI). Each open circle represents the result from one model on one simulated dataset. Each N, J combination was simulated 500 times. Squares represent medians and bars connect the first and third quartile.

Figure 5, similarly to figures 3 and 4 plots simulation results across sample sizes and number of time points with the Brier score on the y-axis. As expected, GLM performed the worst, in terms of both bias and precision. The two-stage model also performed as expected. GEE provided great improvement over GLM, but had many outliers for prediction models. The two-stage model, as expected, was slightly less more biased than GEE but more precise, especially at large sample sizes. And once, again the FLiRTI model outperformed all others and displayed the curious behavior at a sample size of 15.

Due to the potential for practical applications of these models, we also considered the time taken to run each. These results, recorded in seconds, are shown in figures 6 and

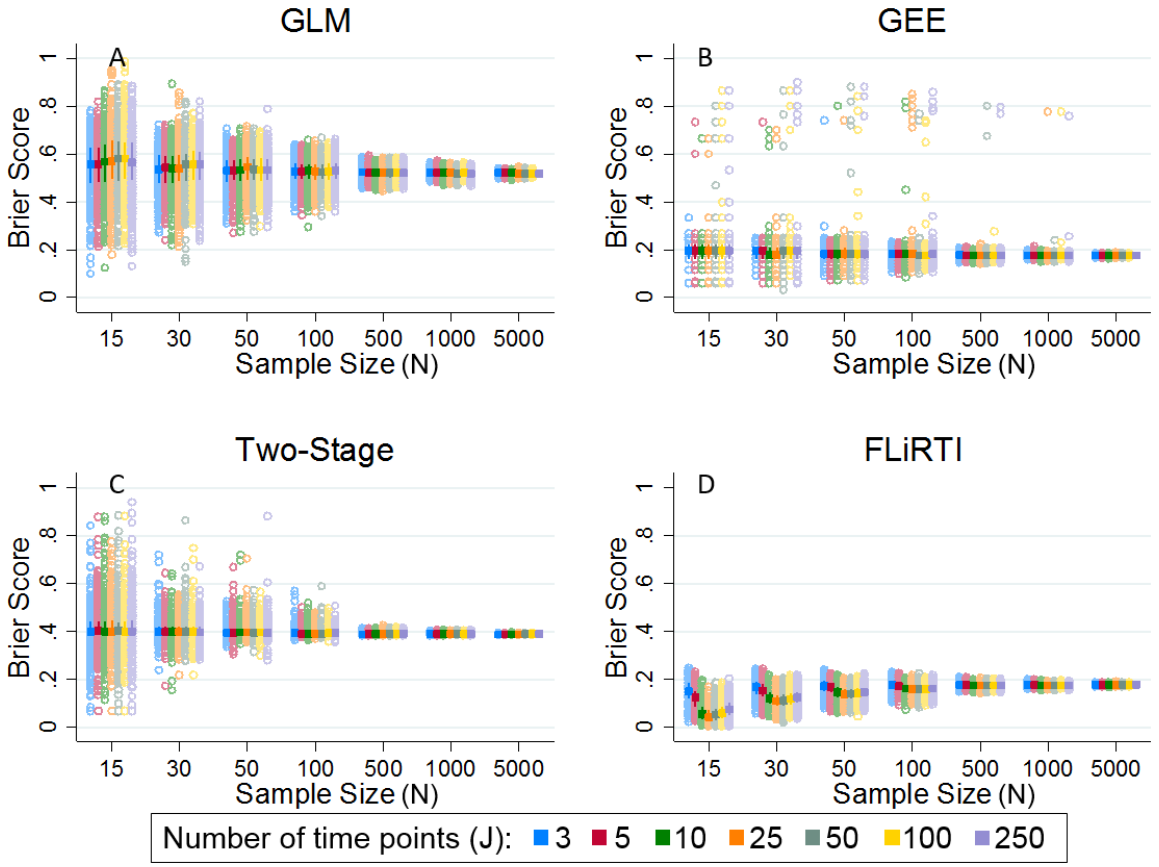

Figure 5: Simulation results showing Brier Scores (lower is more accurate) across sample sizes and number of time points for (A) generalized linear model (GLM) (B) generalized estimating equation (GEE) (C) two-stage model (D) functional regression model (FLiRTI). Each open circle represents the result from one model on one simulated dataset. Each N, J combination was simulated 500 times. Squares represent medians and bars connect the first and third quartile.

7. Figures 8 and 9 are replications of the data contained in figures 6 and 7, only with the y-axis restricted to 100 seconds. Obviously, the GLM is the quickest procedure, with a maximum time of 1.73 seconds in the continuous scenario and 4.21 in the binary scenario. The GEE model performed similarly until four-digit sample sizes were reached or the number of sample time points exceeded 100. The maximum time for GEE was 521 seconds for continuous and 982 seconds for binary. The same pattern was observed for the two-stage model with a continuous maximum time of 203 seconds and binary of 199 seconds. FLiRTI was quicker than GEE for sampled time points less than 250, especially at larger sample sizes but was slightly slower when the number of times points was 250. Figures 6D and

7D show some strange behavior for the FLiRTI model, with the data for the 250 sampled time points giving a reverse-U shape. We postulate that the computation time for FLiRTI, especially during the cross-validation procedure, stabilizes for N>30 and J>100, but are unsure why this occurs. It is worth mentioning in the FLiRTI model, the times shown are the combined times taken to select the tuning parameters by cross-validation and the time to fit the model. The FLiRTI model fit time is on the order of the GLM fit time. The maximum time for a continuous FLiRTI fit was 47.6 seconds and for binary fit was 2.66 seconds. Therefore, if one does not have to search the tuning parameter space and starts with reasonable tuning parameters, the FLiRTI procedure is actually very quick.

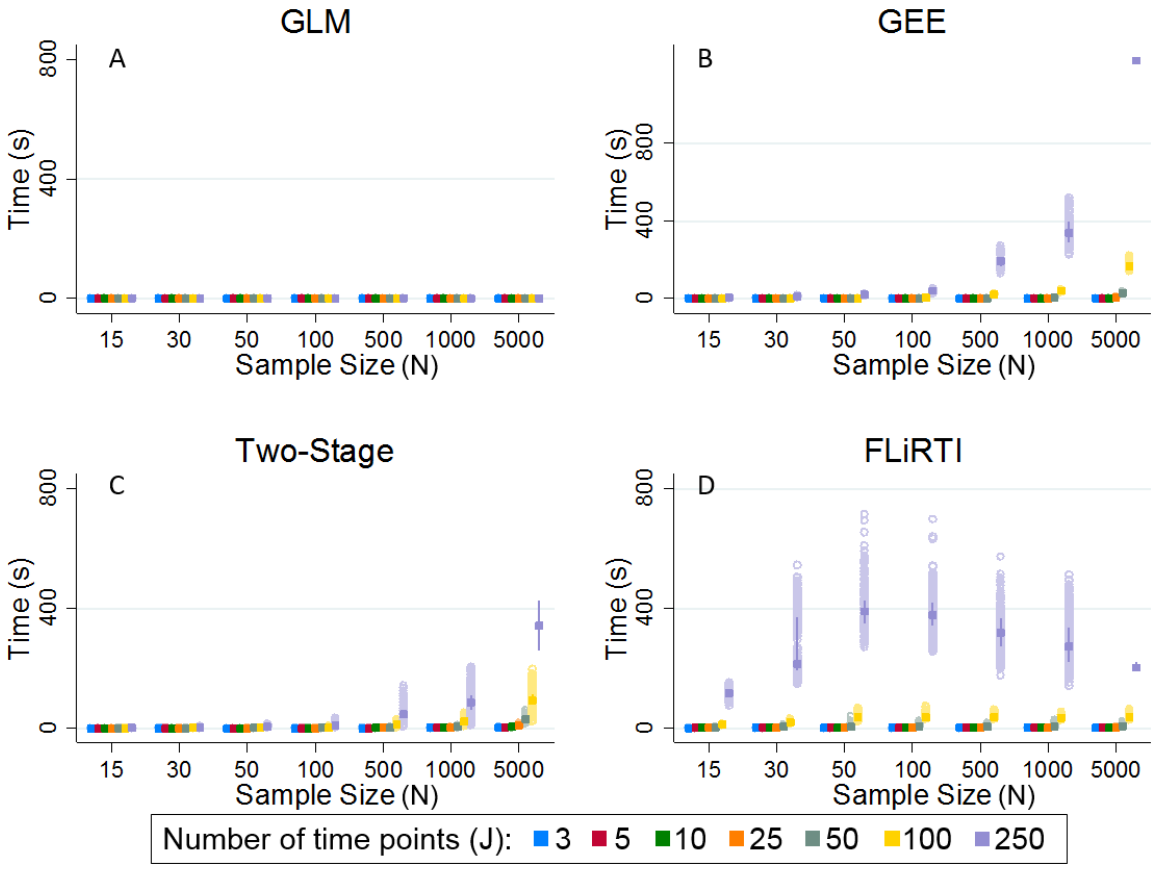

Figure 6: Simulation results showing time (seconds) taken to run each model for continuous outcomes across sample sizes and number of time points for (A) generalized linear model (GLM) (B) generalized estimating equation (GEE) (C) two-stage model (D) functional regression model (FLiRTI). Each open circle represents the result from one model on one simulated dataset. Squares represent medians and bars connect the first and third quartile.

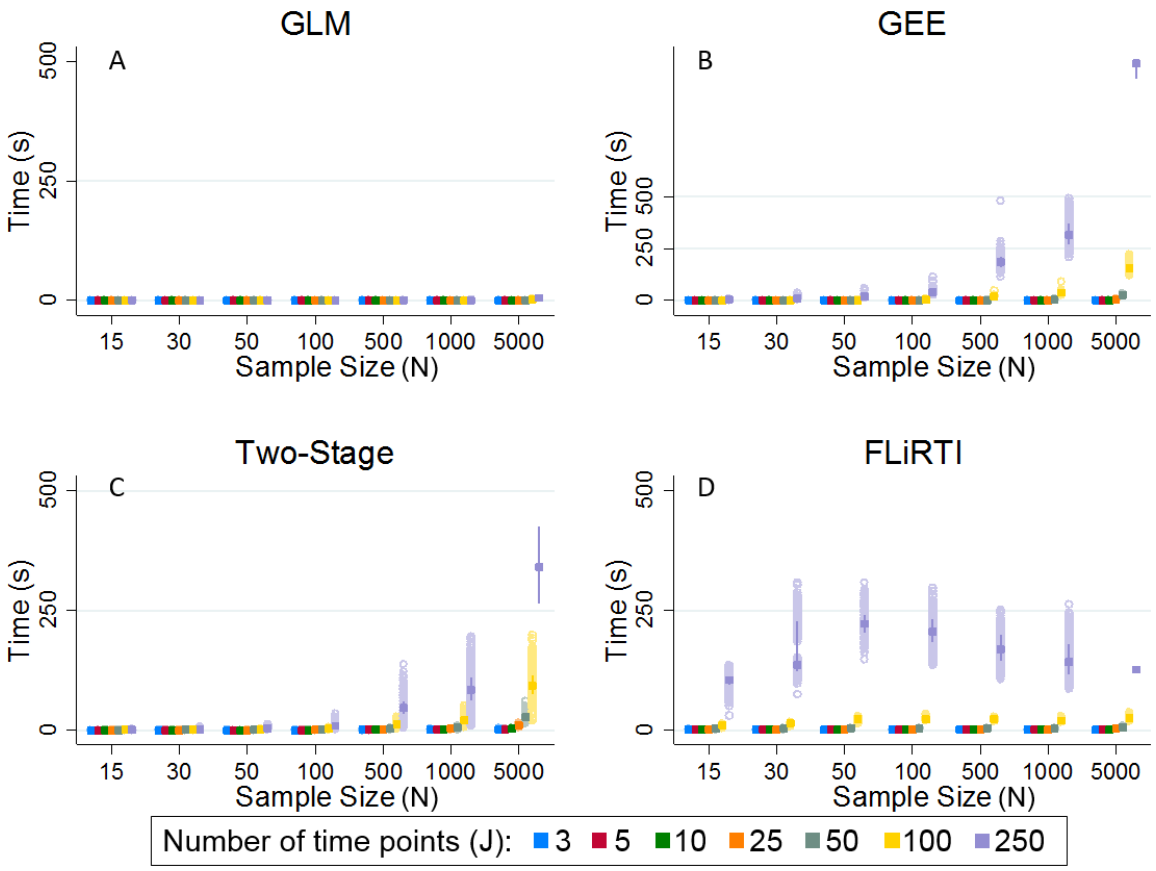

Figure 7: Simulation results showing time (seconds) taken to run each model for binary outcomes across sample sizes and number of time points for (A) generalized linear model (GLM) (B) generalized estimating equation (GEE) (C) two-stage model (D) functional regression model (FLiRTI). Each open circle represents the result from one model on one simulated dataset. Squares represent medians and bars connect the first and third quartile.

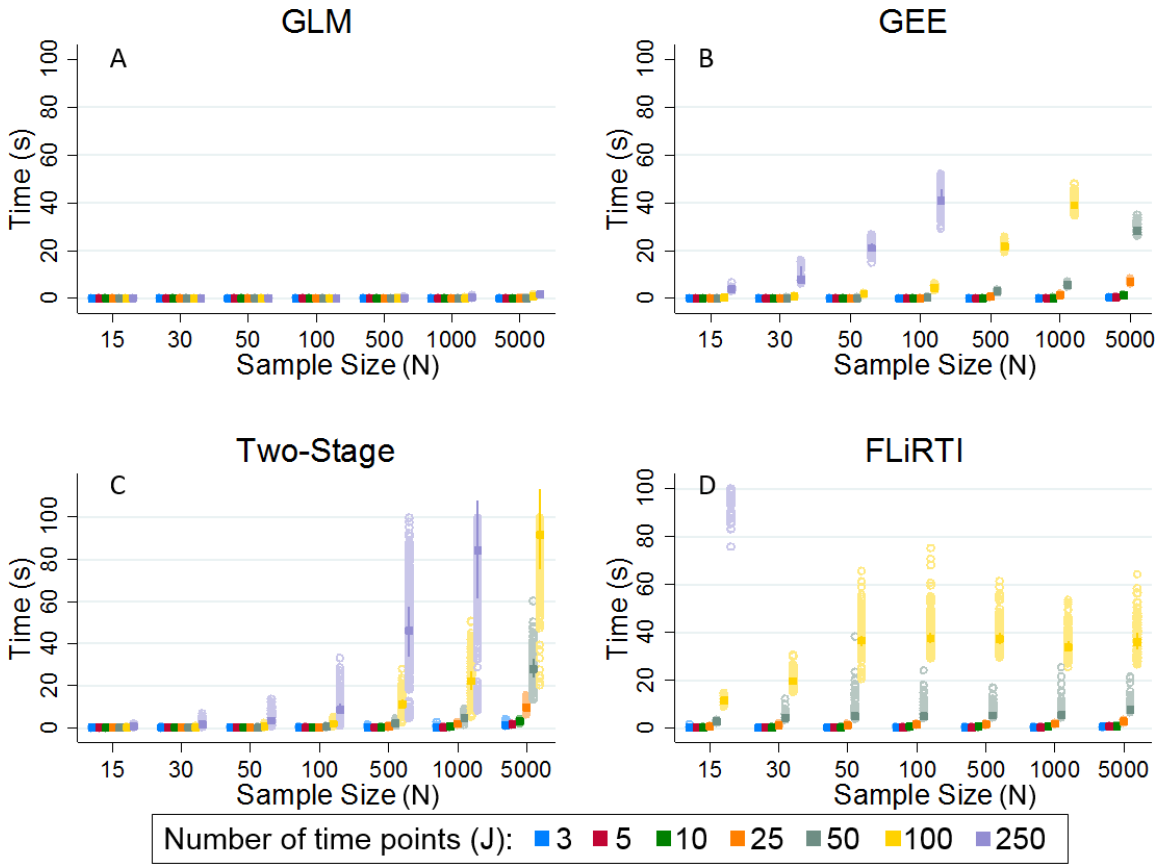

Figure 8: Same information as displayed in Figure 5 but with the y-axis restricted to less than 100 seconds

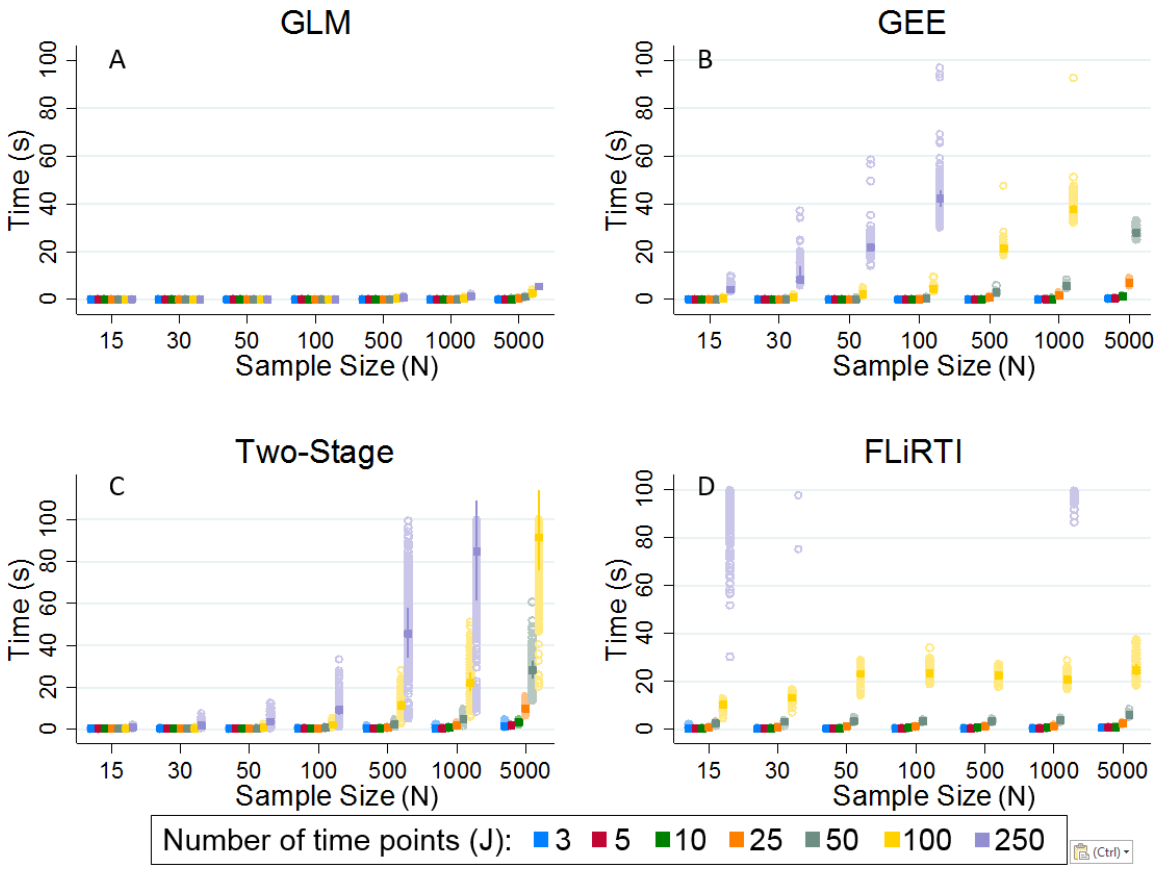

Figure 9: Same information as displayed in Figure 6 but with the y-axis restricted to less than 100 seconds

Table 3: Software options for fitting models described in this paper. Parentheses on the R row denote package from which the command originates.

Package	GLM	GEE	Two-Stage	FLiRTI
R	`glm (base)`	`geeglm (geepack)` `gee (gee)`	`lmer (lme4)` + GEE model	`flrti`
SAS	PROC GLM	PROC GENMOD	PROC MIXED + GEE model	NA
Stata	`glm`	`xtgee`	`mixed` + GEE model	NA
SPSS	GENLIN	GENLIN	MIXED + GEE model	NA

7 DISCUSSION

In this paper, we set out to answer the question "When presented with scalar outcome data on individuals, each with dozens of time points on the predictor, what modeling technique is both most appropriate and best utilizes computational resources?" We believe we have presented results that strongly favor the FLiRTI model in this scenario. FLiRTI displayed the best predictive accuracy in a variety of circumstances, routinely delivering the best MAE, RMSE, and Brier scores. The only downside to the FLiRTI method is the computational resources. However, the bulk of the time was actually taken to find the optimal tuning parameters via cross-validation. The actual model fit for FLiRTI was much quicker, with min, median, and max times of 0.0, 0.01, and 47.6 seconds for continuous outcomes and 0.0, 0.0, and 2.66 seconds for binary outcomes. FLiRTI is very easy to implement in R, but more work needs to be done especially regarding inferences for FLiRTI for generalized linear models. Also, while one can program one's own fitting algorithm, FLiRTI currently only has a native implementation in R, with the `flrti` package written by the inventor of FLiRTI, Gareth James. The other models can be fit in many other statistical software packages. Table 3 shows potential options for fitting these in the four most popular statistical packages.

One more discussion point worth illustrating is that our simulation is of a (not

obvious) hierarchical nature. At first glance, data structured this way ("one row per unit of measurement") appears to be cross-sectional and suitable for fitting with marginal models, such as the first three considered. But our simulated outcomes are random instead of fixed. This should almost always be an assumption when dealing with functional data. Thus, the three marginal models should have and did underperform when compared to FLiRTI. It was a priori assumed that the two-stage model would exhibit more robustness to model misspecification than what was observed, but this was not the case. GEE performed better than expected, though. If the outcomes had been simulated with a fixed population parameter, we would expect the marginal models to performed almost as well as FLiRTI.

The implications for the impact of this research are substantial. With the recent rise of wearable technology and other technologies recording data at very frequent intervals, FLiRTI and other functional data techniques can provide modeling frameworks that produce accurate results and don't require a ton of computing power. An immediate application comes to mind when we consider the studies that have collected ambulatory blood pressure data. Edwards and Simpson provided an attractive approach for model a participants nonlinear ambulatory blood pressure trajectories using orthonormal polynomials in a linear mixed model framework [29]. Their work focused on only modeling the ambulatory blood pressure and not using it as a predictor. So analyses using ambulatory blood pressure data as a predictor are often relegated to summary statistics, such as means, variances, or maximums, for each participant instead of using all information. It will undoubtedly be beneficial to apply FDA techniques to these data. We will do so in the next paper, where we will apply FLiRTI to the data described in our motivating problem and look to predict the probability of ischemic stroke in 93 patients, each with a brain attenuation curve consisting of 27 time points.

REFERENCES

[1] A. M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes*. No. 1, F. Didot, 1805.

[2] C. F. Gauss, *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss*. sumtibus Frid. Perthes et IH Besser, 1809.

[3] S. M. Stigler, *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press, 1986.

[4] J. A. Nelder and R. W. M. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society. Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[5] B. West, K. Welch, A. Galecki, and B. Gillespie, "At (2007) linear mixed models: A practical guide using statistical software."

[6] S. L. Zeger, K.-Y. Liang, and P. S. Albert, "Models for longitudinal data: a generalized estimating equation approach," *Biometrics*, pp. 1049–1060, 1988.

[7] T. J. Hastie and R. J. Tibshirani, *Generalized additive models*, vol. 43. CRC press, 1990.

[8] J. O. Ramsay and B. W. Silverman, *Applied functional data analysis: methods and case studies*. Springer, 2002.

[9] J. O. Ramsay and B. W. Silverman, *Functional data analysis*. Springer, 2005.

[10] G. M. James, "Generalized linear models with functional predictors," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 64, no. 3, pp. 411–432, 2002.

[11] H.-G. Müller and U. Stadtmüller, "Generalized functional linear models," *Annals of Statistics*, pp. 774–805, 2005.

[12] G. M. James, J. Wang, and J. Zhu, "Functional linear regression that's interpretable," *The Annals of Statistics*, pp. 2083–2108, 2009.

[13] J. Goldsmith, C. M. Crainiceanu, B. Caffo, and D. Reich, "Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 61, no. 3, pp. 453–469, 2012.

[14] E. J. Stanek, A. Well, and I. Ockene, "Why not routinely use best linear unbiased predictors (blups) as estimates of cholesterol, per cent fat from kcal and physical activity?," *Statistics in medicine*, vol. 18, no. 21, pp. 2943–2959, 1999.

[15] C. H. Morrell, L. J. Brant, J. D. Pearson, G. N. Verbeke, and J. L. Fleg, "Applying linear mixed-effects models to the problem of measurement error in epidemiologic studies," *Communications in Statistics-Simulation and Computation*, vol. 32, no. 2, pp. 437–459, 2003.

[16] V. Shetty, C. H. Morrell, and S. S. Najjar, "Modeling a cross-sectional response variable with longitudinal predictors: an example of pulse pressure and pulse wave velocity," *Journal of applied statistics*, vol. 36, no. 6, pp. 611–619, 2009.

[17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[18] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[19] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[20] G. M. James, P. Radchenko, and J. Lv, "Dasso: connections between the dantzig selector and lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 1, pp. 127–142, 2009.

[21] C. E. Rasmussen and C. K. Williams, "Gaussian processes for machine learning. 2006," *The MIT Press, Cambridge, MA, USA*, vol. 38, pp. 715–719, 2006.

[22] J. Gertheiss, J. Goldsmith, C. Crainiceanu, and S. Greven, "Longitudinal scalar-on-functions regression with application to tractography data," *Biostatistics*, p. kxs051, 2013.

[23] Y. Chen, J. Goldsmith, and R. T. Ogden, "Variable selection in function-on-scalar regression," *Stat*, 2016.

[24] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[25] D. Bates, M. Mchler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *Journal of Statistical Software*, vol. 67, no. 1, pp. 1–48, 2015.

[26] S. Hjsgaard, U. Halekoh, and J. Yan, "The r package geepack for generalized estimating equations," *Journal of Statistical Software*, vol. 15, no. 1, pp. 1–11, 2005.

[27] G. M. James and J. Zhu, *Documentation for the R-code to implement the FLRTI methodology in Functional Linear Regression Thats Interpretable*, 2009. URL: `http://www-bcf.usc.edu/~gareth/research/flrtidoc.pdf`.

[28] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[29] L. J. Edwards and S. L. Simpson, "An analysis of 24-hour ambulatory blood pressure monitoring data using orthonormal polynomials in the linear mixed model," *Blood pressure monitoring*, vol. 19, no. 3, p. 153, 2014.

THE USE OF FUNCTIONAL FORMS AND SCALAR SUMMARIES OF WHOLE BRAIN PERFUSION IN PREDICTING ISCHEMIC STROKE

SETH T. LIRETTE, INMACULADA B. ABAN, AND COAUTHORS

In preparation for *Journal of Applied Statistics*

Format adapted for dissertation

1 INTRODUCTION

Stroke is the fifth leading cause of death in the United States, with approximately 800,000 people dying from strokes each year [1]. With such a prevalent disease, there is an obvious need for an accurate method to predict who will, in the future, suffer from a stroke. The most prominent example of this is the Framingham Stroke Risk Score [2]. Many others have further developed methods and models to assess stroke risk. While many of these are used to assess future risk of stroke, there is also a need to accurately classify whether a person is experiencing a stroke in real time when a person presents with stroke-like symptoms. This is often done with advanced imaging techniques.

There are three broad types of stroke: hemorrhagic, ischemic, and transient ischemic attack (TIA). TIAs are often called "warnings" or "mini-strokes." Hemorrhagic strokes occur when an artery in the brain ruptures. These are easily detectable with nonenhanced Computed Tomography (CT). Our focus will be on ischemic strokes, which account for 85% of all strokes [1]. During an ischemic stroke, an artery that supplies blood to the brain becomes blocked. Magnetic Resonance Imaging (MRI) is considered the gold-standard for diagnosising an ischemic stroke, but stroke protocols have to be done quickly in order to salvage as much brain tissue as possible. MRI scanners are not always available in such a readily manner. CT scanners, on the other hand, are almost always immediately available at most medical centers and hospitals. A problem arises in that ischemic strokes are very difficult to detect using nonenhanced CT. This is where CT Perfusion (CTP) enters. We will expound on the role and methods of acquiring and estimating CTP parameters in the Methods section.

The data generated from a CTP protocol comes in the form of time-attenuation curves (TAC), to be discussed later. These data are then condensed into some sort of scalar value, such as cerebral blood flow (CBF), cerebral blood volume (CBV), or mean transit time (MTT). This would be akin to taking the data from a sample, condensing them to a few summary statistics, and disregarding the information contained in the remainder of the dataset. This is clearly an inefficient process. To our knowledge no one has examined the use of the full time curve data for assessing stroke probability. This paper will address that need. We will explore the predictive utility of whole-brain tissue time-attenuation curve as well as the predictive ability of the residue function.

2 METHODS

2.1 Computed Tomography Perfusion Data Acquisition

Computed Tomography Perfusion (CTP) is an imaging technique first described twenty-five years ago [3]. This was initially done with single section CT scanners and progressed into multisection CT scanners [4]. With the advent of 256 slice (and higher) CT scanners, it is now possible to perform CTP on the entire brain, but this is still relatively rare due to the advanced machines needed [5]. During this procedure, a bolus of contrast is injected into a patient as a preselected area of tissue is repeatedly scanned while the bolus passes through the arteries and veins. As the contrast arrives to the brain tissue, it increases the "brightness" (often called "attenuation") of the tissue, measured in Hounsfield Units (HU). This is typically used to obtain one image per second or one image every two seconds [4]. This relationship between attenuation and time can be plotted on a Time-Attenuation Curve (TAC), an example is shown for a random participant in Figure 1.

Various statistics, called perfusion parameters, from these TACs are used to evaluate the amount of blood available to be used by the brain. These are derived from an "inflow" TAC gathered from a major artery and a TAC from each voxel of brain tissue to be mapped. Essentially, the most basic measurements are flow and volume, denoted

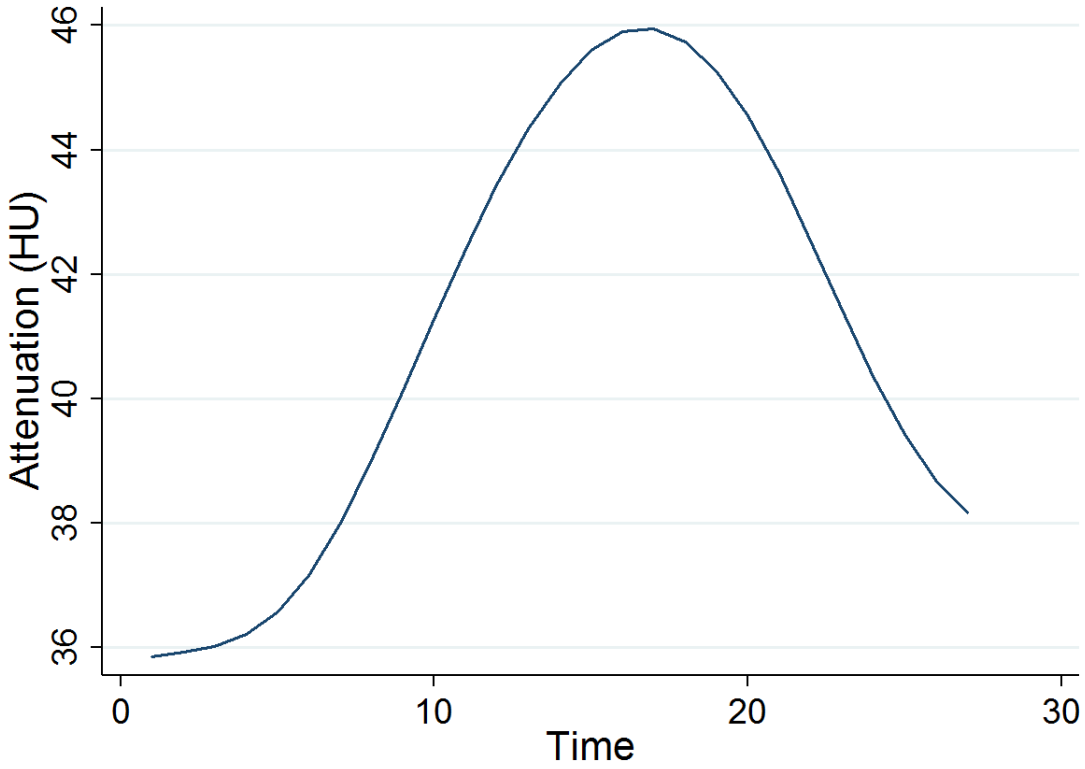

Figure 1: Time-Attenuation Curve (TAC) for brain tissue of a randomly chosen participant.

by Cerebral Blood Flow (CBF) and Cerebral Blood Volume (CBV). Two other additional measures are sometimes reported: Time To Peak (TTP) and Mean Transit Time (MTT). TTP is the time from the start of injection until maximum contrast in reached. TTP can be easily calculated empirically without any estimation techniques. CBV is a fairly straightforward calculation consisting of the area under the curve (AUC) of the tissue TAC divided by the AUC of the arterial TAC. MTT is defined as the mean time between the arterial inflow and venous outflow and has a direct relationship to both CBV and CBF. This will be discussed later. MTT and CBF estimation vary tremendously depending on the method used for the estimation. The perfusion parameters can then be graphed regionally for every volume-pixel (voxel) and provided with a color map.

2.2 Estimating the Residue Function and Perfusion Parameters

Three of the most important measures when trying to determine the amount of

salvageable brain tissue are the CBF, CBV, MTT. They provide clinicians with visually identifiable areas of the brain that are either fully dead, or potientially salvagable. For any particular voxel, CBF can be defined as the ratio of CBV to MTT and CBV can be calculated by dividing the integral of $C_t(t)$ by $C_a(t)$, the TACs for brain tissue and the artery, respectively. [6] [7]. The majority of the work involves calculating the CBF (or, conversely, the MTT), and there is generally a consensus on the best way to estimate CBF, namely deconvolution using singular value decomposition.

Methods involving deconvolution were first set forth by Østergaard et al. [8]. All deconvolution methods start the same. $C_t(t)$ can be defined by two functions: (1) $C_a(t)$ and (2) the residue function $R(t)$. $R(t)$ is defined as the TAC "of the tissue due to an idealized instantaneous injection" [9]. In other words, $R(t)$ measures the fraction of contrast still present in a voxel at time t. $R(t)$ is unitless and $R(0) = 1$. In essence, $C_t(t)$ is the convolution of $C_a(t)$ and $R(t)$ multiplied times CBF [6] [9] [10].

The rest of the problem involves deconvolving this equation, and there are two main non-parametric approaches to this. The equation can be deconvolved using Fourier transforms, but this solution is very sensitive to noise on the CT. By far the most commonly used methods involve singular value decomposition.

After the deconvolution algorithm is applied, we then get an estimate for $\hat{R}(t)$, from which we get our final estimate for MTT and CBF as

$$\widehat{MTT} = \frac{\Delta t \cdot \sum_{i=0}^{N} \hat{R}(t)}{max\hat{R}(t)} \tag{1}$$

$$\widehat{CBF} = max(\hat{R}(t)). \tag{2}$$

Full details can be seen in [6] [9] [10].

This question we seek to answer is this: "Can we get more predictive ability from the full $\hat{R}(t_i)$ curve, rather than reducing it to these scalar values?" We will provide some

insight into this after describing our data and statistical methods employed.

2.3 Patient Population and Measured Variables

Our sample of patients was drawn from a population presenting to the Emergency Department at the University of Mississippi Medical Center between January 1, 2010 and April 15, 2014 with new onset of acute stroke symptoms, triggering a code gray protocol that includes Nonenhanced CT (NECT), CT Angiography, and CTP imaging of the head. Included patients were adults with available NECT images (N = 423). Patients without a confirmatory brain MRI (gold standard for ischemic infarcts) or a second NECT exam documenting hemorrhagic strokes were excluded (N = 133). There were 290 subjects in the final inclusion cohort. For patients with ischemic strokes, a confirmatory MRI was used to categorize the type of stroke. Since CTP was only a minor part of the study, not all were included for assessment of CTP. Of the 290 subjects, 93 were randomly chosen for inclusion into the CTP portion of the study. A designated region of interest on each patient's CTP preprocessed images was then examined by a trained neuroradiologist, and data was recorded.

Of the variables collected, there are 6 in which we are interested. First, for each patient's region of interest, the attenuation value (in Hounsfield units) was recorded along with the time at which the attenuation value occurred. Using this data, we can then construct $C_t(t)$. Also collected was the time at which the contrast entered the artery, the maximum attenuation during this arterial phase, and the duration of this arterial phase. Finally, each patient received a confirmatory MRI indicating the presence or absence of an ischemic stroke.

Given that $C_a(t)$ was not directly measured, we were relegated to constructing it from a gamma variate function, an approach widely accepted [11]. Using the methods described above, we then estimated $\hat{R}(t_i)$, $\widehat{CBV}$, $\widehat{CBF}$, and $\widehat{MTT}$.

2.3 Statistical Methods

Methods for Scalar Predictors: Logistic Regression

The almost ubiquitous choice for relating any scalar predictor to a binary scalar outcome is logistic regression. Formulaically, given predictor X,

$$\text{logit}(p_i) = \log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1^T X \tag{3}$$

This will be the approach used when $\widehat{CBV}$, $\widehat{CBF}$, or $\widehat{MTT}$ is our predictor variable.

Methods for Functional Predictors: Functional Linear Regression That's Interpretable

If instead we were to consider the each patient's tissue TAC, $C_t(t)$, or residue function, $R(t)$, as our predictor for stroke probability, we will have to open our functional data analysis (FDA) toolkit. Functional regression and its analogue, the generalized functional linear model (GFLM), are two very useful tools in FDA. Ramsay and Silverman and Muller and Stadtmuller provide a variety of examples and applications [12] [13]. Each technique under the GFLM umbrella falls into one of three categories: (1) function-on-scalar regression (2) function-on-function regression or (3) scalar-on-function regression. Obviously we need to apply number (3).

Scalar-on-function regression for binary outcomes is typically expressed by

$$Y_i = \beta_0 + \int \beta(t) X_i(t) dt. \tag{4}$$

The coefficient function $\beta(t)$ is usually represented in one of three ways. The first method represents $\beta(t)$ as some *p*-dimensional basis function. The second method attempts to shrink the variability in $\beta(t)$ via some least squares penalty. The third method, and our method of choice takes a variable selection approach to solving the problem. James et. al have named this method "Functional Linear Regression That's Interpretable" (FLiRTI) [14]. FLiRTI is an attractive choice because it gives accurate predictions, is flexible, has

nice theoretical properties, is computationally efficient, and unlike the first two methods, gives an estimate for the coefficient function $\beta(t)$ that is interpretable.

We will give a brief overview of the FLiRTI model and model fit. Full details can be seen in James et. al [14]. The FLiRTI model borrows an idea from the basis approach but allows errors in the $\beta(t)$ specification. Specifically

$$Y_i = \beta_0 + \mathbf{X}_i^T \eta \tag{5}$$

where $\mathbf{X}_i = \int X_i(t)\mathbf{B}(t)dt$ and the p-dimensional basis $\mathbf{B}(t) = [b_1(t), b_2(t), ..., b_p(t)]^T$. If we define

$$\boldsymbol{\gamma} = A\eta \tag{6}$$

where A is a vector combining finite difference operators with the basis functions, we can combine (5) and (6) to get the FLiRTI model

$$\mathbf{Y} = XA^{-1}\boldsymbol{\gamma} \tag{7}$$

The model is then fit using either the LASSO selector via the LARS algorithm or the Dantzig selector via the DASSO algorithm to obtain $\hat{\boldsymbol{\gamma}}$ [15] [16] [17] [18]. The FLiRTI estimate for $\beta(t)$ is then given by

$$\hat{\beta}(t) = \mathbf{B}(t)^T A^{-1}\hat{\boldsymbol{\gamma}} \tag{8}$$

FLiRTI has tuning parameters that correspond to tuning parameters in any smoothing situation. The most common approach to choosing these is to choose the combination that produce the lowest cross-validated residual sum of squares, and this is the approach we will take. For inference, one can obtain pointwise bootstrapped confidence intervals for $\beta(t)$ or write a permutation test to test for statistical significance of the relationship between $X(t)$ and Y [14].

Evaluating Predictive Ability: Brier Score

For binary outcomes, most modeling techniques yield predicted probabilities, and a few options for defining predictive accuracy were considered. Perhaps the most common and intuitive option is setting some threshold for yes/no (say 0.50) and define any predicted probability below this as a 0 and above as a 1. We chose not to employ this option for multiple reasons. It depends on an arbitrary threshold choice, it requires strong assumptions of correct model fit, and it discards a lot of information. We also considered using area under the receiver operating characteristics curve (AUC). While better than the previous option, we ultimately decided not to use the AUC because it quantifies the ordering of the predicted probabilities, not their accuracies.

Ultimately, we decided to use the Brier score [19]. A Brier score of zero is the best score possible (either a predicted probability of 1.0 and an observed outcome of 1 or a predicted probability of 0.0 and an outcome of zero). On the other hand, one is the worst score possible. Also note that a predicted probability of 0.5 will yield a Brier score of 0.25 regardless of the observed outcome.

From the Brier score, we can also develop a permutation test. The steps go as such: (1) Calculate the Brier score, BS, for the FLiRTI method applied to the stroke data (2) Permute the response variable B times and for each permutation compute the new Brier score (3) The "p-value" for rejecting no relationship between $X(t)$ and and Y will then be the percentage of permutations that fall below BS.

All calculations were performed within the R computing environment and FLiRTI models were fit using the R package `flrti` written by James and Zhu [20] [21].

3 RESULTS

Of the 93 participants 66 (71%) had an MRI confirmed stroke. The sample contained 50 (54%) males and of those who had a stroke, 42 (64%) were male, with 24 (36%) being female (Fisher's exact p=0.005). The mean age for the cohort was 59 years with a

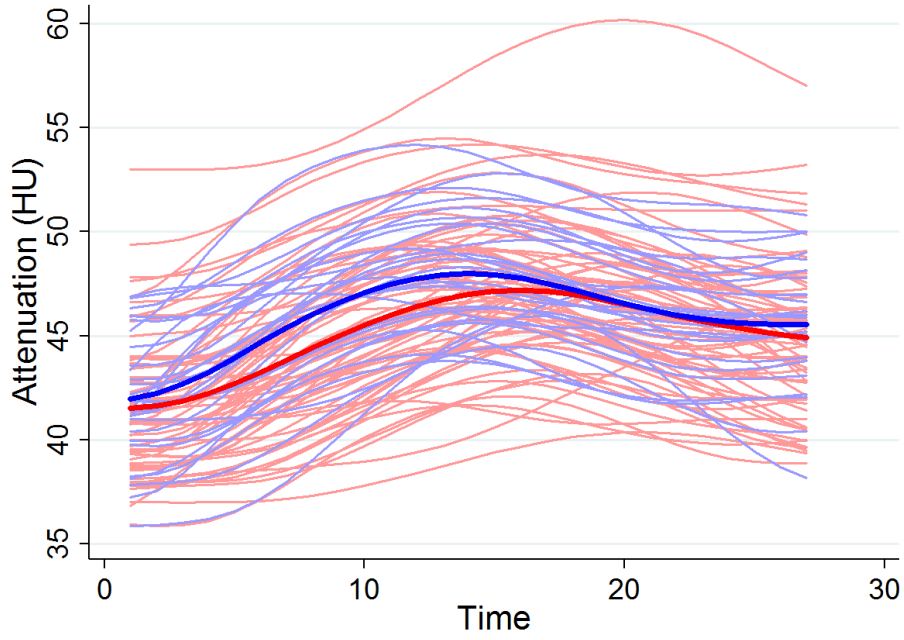

Figure 2: Brain Tissue Time-Attenuation Curves (TACs) for all 93 participants. Blue curves represent those without stroke. Red curves represent those with stroke. Thick lines show the average curve for each group.

standard deviation (SD) of 15.4 years and range of 21 to 92 years. The mean age for those with stroke was 62 years (SD=13.5) while the mean age for those without stroke was 54 years (SD=18.3) (Mann-Whitney U p=0.055).

Figures 2 and 3 show the raw data for each participant's tissue TAC and residue function, respectively. Even before modeling, we can gain some insights with exploratory data analysis. As displayed in figure 2, there is not much of a distinction between stroke/no stroke for the average tissue TACs. The blue line is slightly higher, but there is not much different in either shape nor magnitude. Contrastingly, consider figure 3. The thick blue line is a very different shape than the thick red line. The blue line is almost monotonically decreasing, representing a clearing of the contrast in a way that an impeded brain should. However, the red line shows something different. The contrast does not seem to be clearing in a way similar to those without stroke.

Modeling results are shown in Table 1. The scalar values did not offer much in

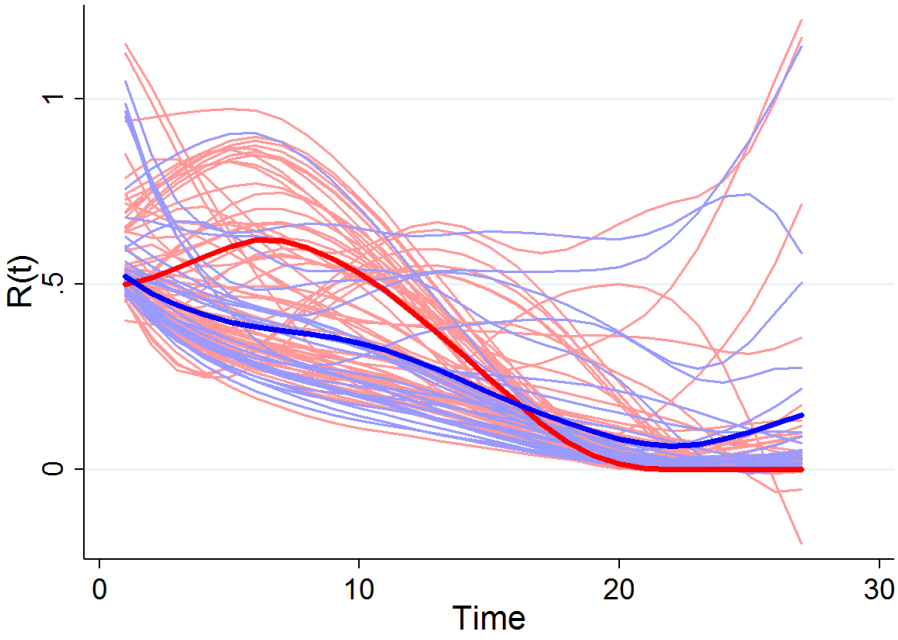

Figure 3: Brain tissue residue functions ($R(t)$) for all 93 participants. Blue curves represent those without stroke. Red curves represent those with stroke. Thick lines show the average curve for each group.

terms of predictive utility, with all p>0.4 and all Brier Scores greater than 0.2. The models using functional predictors $C_t(t)$ and $R(t)$ were both significant using the permutation test. As hypothesized, the residue function offered the most predictive ability, with the lowest Brier Score. Figure 4 displays the permutation results for both functional predictors. We see that no Brier score permutation for either predictor falls below the observed Brier score from the true data, with all permutations for the $R(t)$ predictor falling well above the true value of 0.139, indicating strong evidence of a relationship between $R(t)$ and stroke probability.

Figures 5-9 display the ordered predicted probabilities for stroke for each of the predictors for each participant. The scalar predictors do not discriminate between stroke/no stroke very well, with all predicted probabilities lying in the 0.6-0.85 range. Beginning with figure 8 and the $C_t(t)$ predictor, we start to see a bit more separation in terms of the predicted probabilities, but there is still a lot of mixing of the blue and red dots, showing

Predictor	Odds Ratio	p-value*	Brier Score
CBF	0.998	0.621	0.205
MTT	1.124	0.406	0.205
CBV	1.023	0.721	0.206
$C_t(t)$	NA	<0.001	0.183
$R(t)$	NA	<0.001	0.139

Table 1: Modeling results for both scalar and functional predictors
CBF=Cerebral Blood Flow; MTT=Mean Transit Time; CBV=Cerebral Blood Volume
*From logistic regression models for CBF, MTT, and CBV and permutation test for FLiRTI models for $C_t(t)$ and $R(t)$

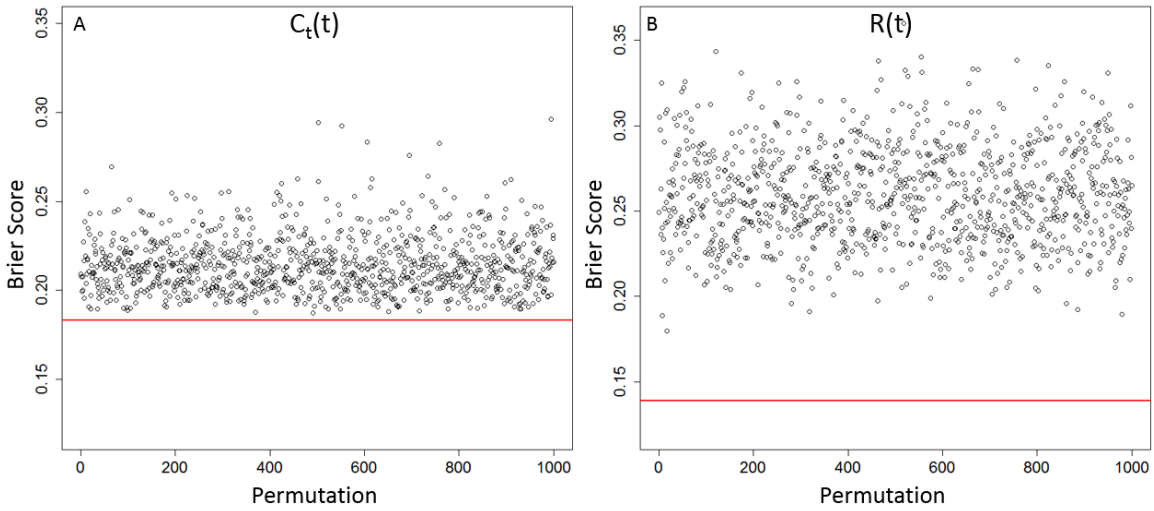

Figure 4: Brier score (BS) permutation test results for brain tissue time-attenuation curve, $C_t(t)$, and residue function, $R(t)$. The red line shows the observed BS from the true data and each dot represents a different permutation.

a lack of ordering and discrimination. Examining figure 9, things improve drastically. The lowest eight predicted probabilities all had absence of stroke, and only one lone predicted probability above 0.75 (a 52 year-old female) was not a stroke. Sensitivity was high for this model (0.91) while specificity was moderately low (0.44) (Table 2).

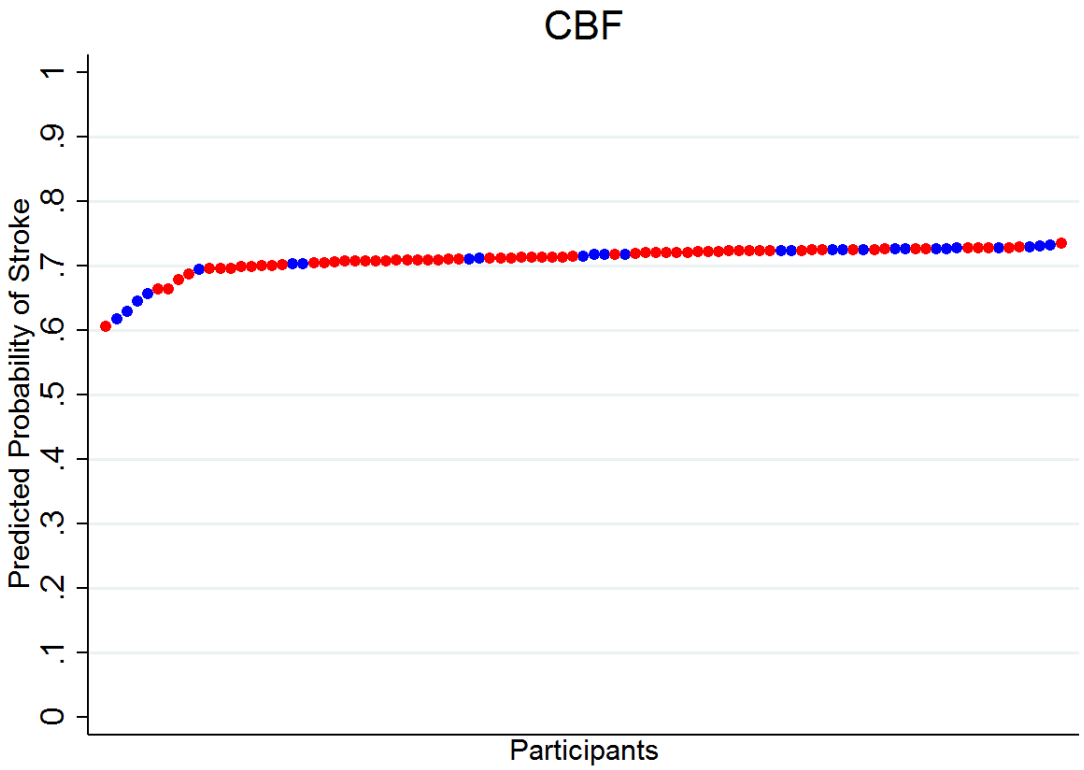

Figure 5: Ordered predicted probabilities for stroke for cerebral blood flow for each participant. Blue dots represent those without stroke. Red dots represent those with stroke.

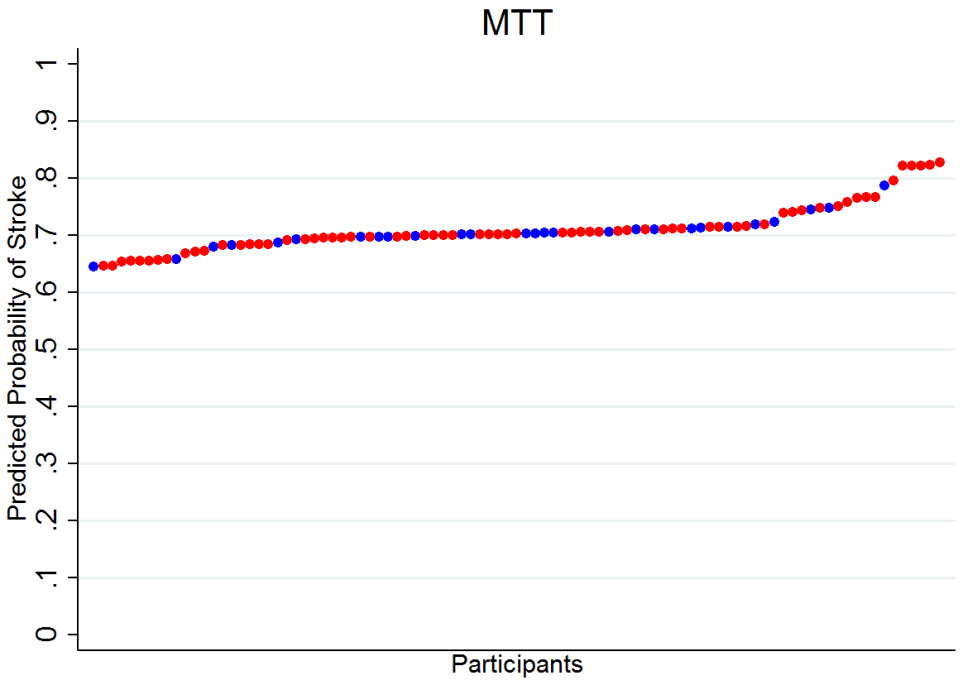

Figure 6: Ordered predicted probabilities for stroke for mean transit time for each participant. Blue dots represent those without stroke. Red dots represent those with stroke.

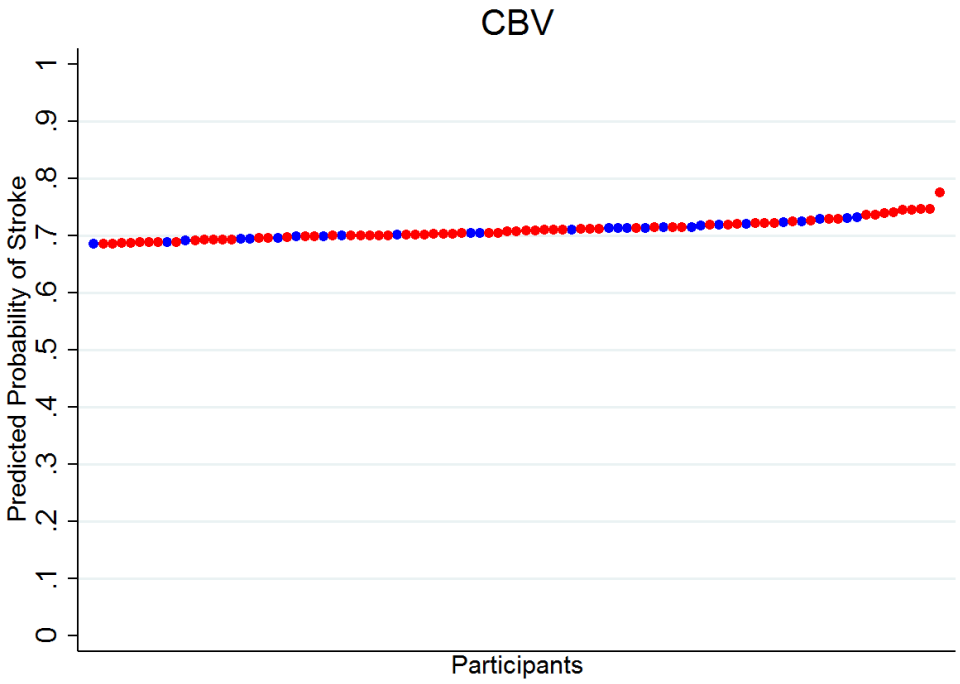

Figure 7: Ordered predicted probabilities for stroke for cerebral blood volume for each participant. Blue dots represent those without stroke. Red dots represent those with stroke.

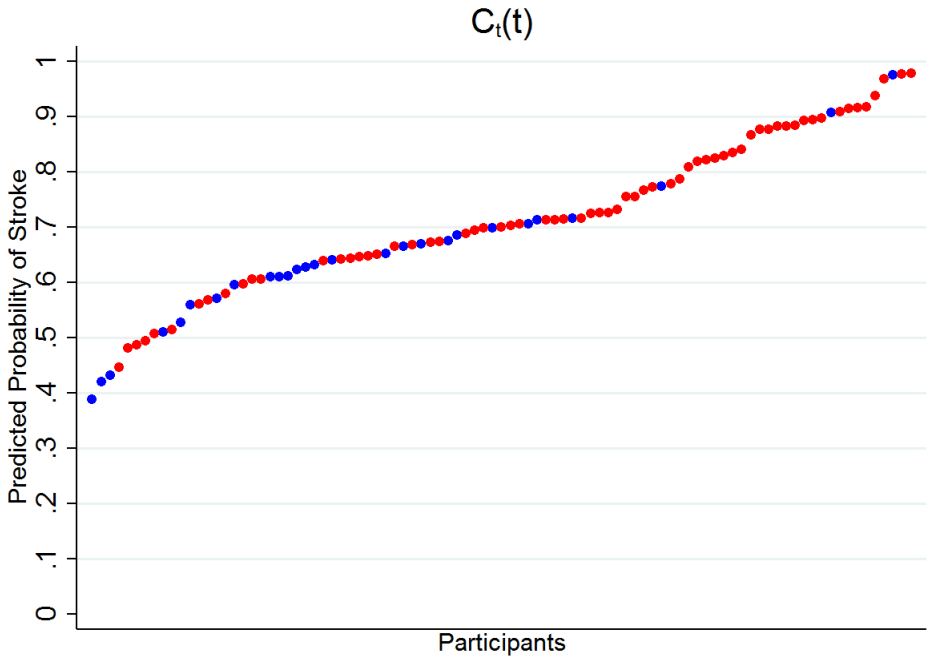

Figure 8: Ordered predicted probabilities for stroke for brain tissue time-attenuation curves for each participant. Blue dots represent those without stroke. Red dots represent those with stroke.

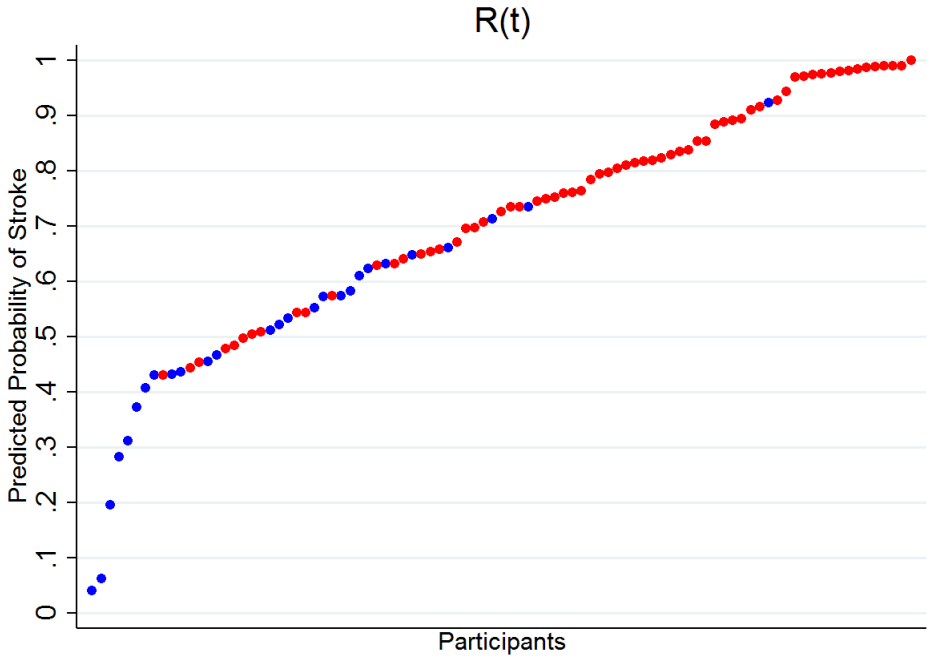

Figure 9: Ordered predicted probabilities for stroke for residue functions for each participant. Blue dots represent those without stroke. Red dots represent those with stroke.

Let us return to the plot of each participant's residue function. Figure 10 shows the same information plotted in figure 3, but only for those without stroke. Also highlighted in green are the five participants with the lowest predicted probability from the FLiRTI model. These five curves are visually distinct and provide us with some insights. These participant's seem to be displaying proper physiology. The contrast is quickly exiting the brain. Figure 11 shows the analogous curves for those participants with stroke. Highlighted in magenta are those five participants with the highest predicted probabilities of stroke. The distinction here is less apparent than the distinction for the no stroke curves.

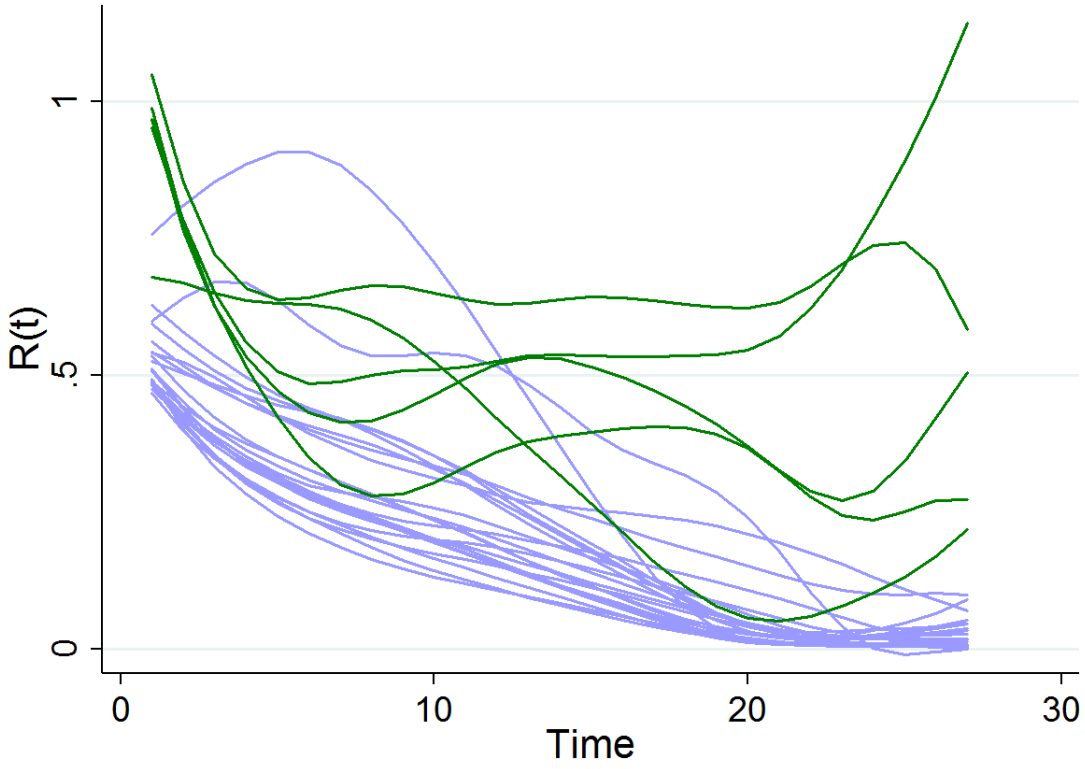

Figure 10: Brain tissue residue functions ($R(t)$) for all 27 participants without stroke. Highlighted in green are the five participants with the lowest predicted probability for stroke from the FLiRTI model.

Currently, there does not exist any effective means for adjusting for covariates when using a FLiRTI model. To somewhat circumvent this problem, we conducted a series of stratified analyses for men, women, less than 65 years old, and 65 years or greater. The

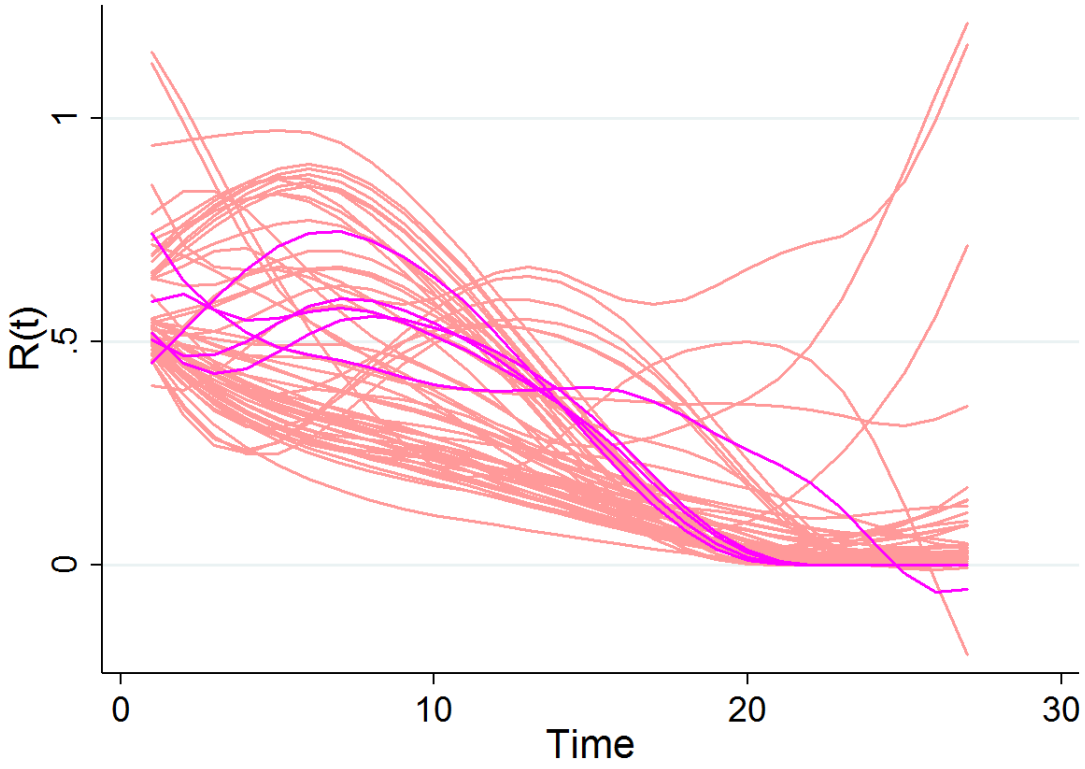

Figure 11: Brain tissue residue functions ($R(t)$) for all 66 participants with stroke. Highlighted in magenta are the five participants with the highest predicted probability for stroke from the FLiRTI model.

results are shown in Table 2. For the ROC analysis, a predicted probability of $>$0.5 was defined as the model predicting a positive stroke. As you can see, within each strata, there was a decrease in Brier score and increases in every parameter of the ROC analysis. Specifically, there was a marked improvement in sensitivity, especially for males, and $\geq$65 years. Also, there was a drastic improvement in specificity. This provides some evidence that if we were able to adjust for sex and age in our model, the predictive ability would improve even more. Returning to the 52 year-old female identified as the lone predicted probability from the overall model over 0.75 but without a stroke: the female-only model gave her a predicted probability of 0.58 and the $<$65-only model gave her a predicted probability of 0.92, leading us to believe that sex adjustment is more important than age adjustment, but this is only conjecture.

	Overall	Males	Females	<65 years	≥65 years
Brier Score	0.139	0.059	0.084	0.099	0.058
ROC Area	0.68	0.75	0.88	0.94	0.83
Sensitivity	0.91	1.00	0.92	0.93	1.00
Specificity	0.44	0.50	0.84	0.94	0.67
PPV	0.80	0.91	0.88	0.97	0.89
NPV	0.67	1.00	0.89	0.85	1.00

Table 2: Overall and stratified models with Brier scores and ROC analysis
PPV=positive predictive value. NPV=negative predictive value

4 DISCUSSION

Through this paper, we have demonstrated that the full functional form of the residue function offers superior predictive ability as compared to the brain tissue TAC and the scalar alternatives – CBF, MTT, and CBV. The FLiRTI model using $R(t)$ most aptly separated those with stroke from those without.

With this being said, however, FLiRTI is not a panacea for these types of data. While the acronym stands for "Functional Linear Regression That's Interpretable," we view this as a bit of a misnomer. The FLiRTI model mimics a variable selection procedure applied to functional data and thus gives a beta estimate for each time point. In our view, these betas do not offer much in terms of translational value. Furthermore, there is not currently a mechanism in which to adjust for covariates in FLiRTI models. We would, in the future, like to examine the addition of even simple covariates such as age, sex, and smoking. Currently, this is not possible, as it is with other modeling procedures such as logistic regression. This is a desired addition and more work needs to be done in this area. We also performed a 10-fold cross-validation to assess model overfitting which returned a Brier score of 0.240, indicating that some overfitting might be occurring.

As another note of caution, we must stress that no algorithmic procedure should supplant the well-trained eye of a radiologist, neurologist, or neuroradiologist. Human intuition and reasoning will always offer something that machines cannot attain.

This paper shows that the residue function can be used at the population level, but

more work needs to be done to show and translate how this function can be used at the individual level for a new person presenting stroke-like symptoms. It is very easy to make a perfusion map of CBF, MTT, or CBV for each pixel of a brain image. In future work, we plan on devising a method to visualize $R(t)$ in this way. Much work needs to be done to accomplish this. Nevertheless, we believe we have made a useful addition to the understanding that the role of the full functional form of the residue function provides in predicting stroke and have shown that it can feasibly be used in practice.

REFERENCES

[1] CDC, "Stroke." `http://www.cdc.gov/stroke`. Accessed February 22, 2016.

[2] P. A. Wolf, R. B. D'agostino, A. J. Belanger, and W. B. Kannel, "Probability of stroke: a risk profile from the framingham study.," *Stroke*, vol. 22, no. 3, pp. 312–318, 1991.

[3] K. Miles, M. Hayball, and A. Dixon, "Colour perfusion imaging: a new application of computed tomography," *The Lancet*, vol. 337, no. 8742, pp. 643–645, 1991.

[4] B. F. Tomandl, E. Klotz, R. Handschu, B. Stemper, F. Reinhardt, W. J. Huk, K. Eberhardt, and S. Fateh-Moghadam, "Comprehensive imaging of ischemic stroke with multisection ct," *Radiographics*, vol. 23, no. 3, pp. 565–592, 2003.

[5] J. J. S. Shankar, C. Lum, *et al.*, "Whole brain ct perfusion on a 320-slice ct scanner," *Indian Journal of Radiology and Imaging*, vol. 21, no. 3, p. 209, 2011.

[6] A. Konstas, G. Goldmakher, T.-Y. Lee, and M. Lev, "Theoretic basis and technical implementations of ct perfusion in acute ischemic stroke, part 1: theoretic basis," *American Journal of Neuroradiology*, vol. 30, no. 4, pp. 662–668, 2009.

[7] R. Wirestam, L. Andersson, L. Ostergaard, M. Bolling, J.-P. Aunola, A. Lindgren, B. Geijer, S. Holtås, and F. Ståhlberg, "Assessment of regional cerebral blood flow by dynamic susceptibility contrast mri using different deconvolution techniques," *Magnetic resonance in medicine*, vol. 43, no. 5, pp. 691–700, 2000.

[8] L. Østergaard, R. M. Weisskoff, D. A. Chesler, C. Gyldensted, and B. R. Rosen, "High resolution measurement of cerebral blood flow using intravascular tracer bolus

passages. i. mathematical approach and statistical analysis," *Magnetic Resonance in Medicine*, vol. 36, no. 5, pp. 715–725, 1996.

[9] K. Miles and M. Griffiths, "Perfusion ct: a worthwhile enhancement?," *The British journal of radiology*, vol. 76, pp. 220–231, 2003.

[10] J. Wang, O. Masoud, and B. Frake, "Delay insensitive svd algorithm for perfusion analysis," Apr. 11 2008. US Patent App. 12/082,695.

[11] K. Kudo, M. Sasaki, K. Ogasawara, S. Terae, S. Ehara, and H. Shirato, "Difference in tracer delay–induced effect among deconvolution algorithms in ct perfusion analysis: Quantitative evaluation with digital phantoms," *Radiology*, vol. 251, no. 1, pp. 241–249, 2009.

[12] J. O. Ramsay and B. W. Silverman, *Applied functional data analysis: methods and case studies*. Springer, 2002.

[13] H.-G. Müller and U. Stadtmüller, "Generalized functional linear models," *Annals of Statistics*, pp. 774–805, 2005.

[14] G. M. James, J. Wang, and J. Zhu, "Functional linear regression that's interpretable," *The Annals of Statistics*, pp. 2083–2108, 2009.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[16] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[17] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[18] G. M. James, P. Radchenko, and J. Lv, "Dasso: connections between the dantzig selector and lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 1, pp. 127–142, 2009.

[19] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[20] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[21] G. M. James and J. Zhu, *Documentation for the R-code to implement the FLRTI methodology in Functional Linear Regression Thats Interpretable*, 2009. URL: `http://www-bcf.usc.edu/~gareth/research/flrtidoc.pdf`.

CONCLUSION

Summary

The overarching goal of this dissertation was to apply statistical methodology to CT perfusion data, where statistics have been mostly ignored. We first exhibited how software that constructs perfusion maps can be implemented and distributed completely within an open-source environment, alleviating either the cost or accessibility hindrances that often accompany access to such. We then moved on to demonstrate that the FLiRTI model is a good choice when seeking to relate a functional form to a scalar outcome. Finally, we showed the the residue function had the best predictive ability in the analysis of 93 code grey participants.

Functional data analysis and scalar-on-function regression, in particular, are quickly becoming more widely used in research, especially in the clinical and public health sectors. With the rapid expansion of wearable technology and more prolific data collection at the individual level, statisticians need to be able to properly utilize the storm of data quickly approaching. We, especially in Paper 3, have shown that this can be done in a practical way and in Paper 2 demonstrated that conclusions drawn from Paper 3 are more appropriate since a more appropriate model was chosen than those typically considered. Once more research is done, we should be able to apply the principles set forth in Paper 1 to publicly disseminate tools to help the general public have a better understanding of their individual health. Although not personalized medicine, *per se*, this will help individuals quantify themselves better and perhaps seek out interventions sooner.

Future Research

To continue this research, we can make some additions and improvements to the software detailed in Paper 1. First, we believe that, through more efficient programming, the processing and construction time for the perfusion maps, and in particular those using the oSVD algorithm, can be lowered. Further, the current iteration of the software only constructs a CBF map. It should not be too much of a burden to add additional functions to create maps for CBV, MTT, and possibly Tmax and other perfusion parameters. Finally, the software currently presupposes a basic understanding of R in order to access and run the program. While R is open-source, this could be an unnecessary hurdle for a lot of people that could wish to use the software. Therefore, we look forward to developing a Shiny application that runs the software. This would enable anyone with access to a web browser to create a perfusion map.

Relating to the topics covered in Paper 2, we believe we could extend this research further by examining other functional data analysis models in addition to FLiRTI. The methods competing against FLiRTI in Paper 2 are very common methods, but more sophisticated options are also available. We can consider the same simulation approach while examining the use of functional regression techniques that utilize principle components reductions or based the functional predictors on spline bases. Additionally, it would be interesting to examine the use of joint models (sometimes called shared parameter models) to model data of this form as well.

Finally, in Paper 3, we demonstrated how using the residue function, rather than distilling it down to scalar quantities gives better predictive accuracy for stroke probabilities. While this is a useful result, some improvements can be made. First, while FLiRTI is "interpretable," we do not see it as particularly translational to clinicians. It would be useful to develop a modeling framework that accomplishes this goal. Further, work needs to be done to be able to add covariates to the FLiRTI model. This is a non-trivial, but incredibly important issue. Finally, Paper 3 shows that the residue function is useful at the population

level, but more work needs to be done to show and translate how this function can be used at the individual level for a new person presenting stroke-like symptoms. It is very easy to make a perfusion map of CBF, MTT, or CBV for each pixel of a brain image. In future work, we plan on devising a method to visualize $R(t)$ in this way. Much work needs to be done to accomplish this.

In conclusion, this dissertation has been both an exciting and enlightening endeavor, but much more work needs to be done, and we look forward to the task at hand.

GENERAL LIST OF REFERENCES

[1] CDC, "Stroke." `http://www.cdc.gov/stroke`. Accessed February 22, 2016.

[2] K. Miles, M. Hayball, and A. Dixon, "Colour perfusion imaging: a new application of computed tomography," *The Lancet*, vol. 337, no. 8742, pp. 643–645, 1991.

[3] B. F. Tomandl, E. Klotz, R. Handschu, B. Stemper, F. Reinhardt, W. J. Huk, K. Eberhardt, and S. Fateh-Moghadam, "Comprehensive imaging of ischemic stroke with multisection ct," *Radiographics*, vol. 23, no. 3, pp. 565–592, 2003.

[4] J. J. S. Shankar, C. Lum, *et al.*, "Whole brain ct perfusion on a 320-slice ct scanner," *Indian Journal of Radiology and Imaging*, vol. 21, no. 3, p. 209, 2011.

[5] A. Konstas, G. Goldmakher, T.-Y. Lee, and M. Lev, "Theoretic basis and technical implementations of ct perfusion in acute ischemic stroke, part 1: theoretic basis," *American Journal of Neuroradiology*, vol. 30, no. 4, pp. 662–668, 2009.

[6] R. Wirestam, L. Andersson, L. Ostergaard, M. Bolling, J.-P. Aunola, A. Lindgren, B. Geijer, S. Holtås, and F. Ståhlberg, "Assessment of regional cerebral blood flow by dynamic susceptibility contrast mri using different deconvolution techniques," *Magnetic resonance in medicine*, vol. 43, no. 5, pp. 691–700, 2000.

[7] K. Miles and M. Griffiths, "Perfusion ct: a worthwhile enhancement?," *The British journal of radiology*, vol. 76, pp. 220–231, 2003.

[8] L. Østergaard, R. M. Weisskoff, D. A. Chesler, C. Gyldensted, and B. R. Rosen, "High resolution measurement of cerebral blood flow using intravascular tracer bolus

passages. i. mathematical approach and statistical analysis," *Magnetic Resonance in Medicine*, vol. 36, no. 5, pp. 715–725, 1996.

[9] J. Wang, O. Masoud, and B. Frake, "Delay insensitive svd algorithm for perfusion analysis," Apr. 11 2008. US Patent App. 12/082,695.

[10] O. Wu, L. Østergaard, R. M. Weisskoff, T. Benner, B. R. Rosen, and A. G. Sorensen, "Tracer arrival timing-insensitive technique for estimating flow in mr perfusion-weighted imaging using singular value decomposition with a block-circulant deconvolution matrix," *Magnetic resonance in medicine*, vol. 50, no. 1, pp. 164–174, 2003.

[11] K. Kudo, S. Christensen, M. Sasaki, L. Østergaard, H. Shirato, K. Ogasawara, M. Wintermark, and S. Warach, "Accuracy and reliability assessment of ct and mr perfusion analysis software using a digital phantom," *Radiology*, vol. 267, no. 1, pp. 201–211, 2013.

[12] K. Kudo, M. Sasaki, K. Yamada, S. Momoshima, H. Utsunomiya, H. Shirato, and K. Ogasawara, "Differences in ct perfusion maps generated by different commercial software: Quantitative analysis by using identical source data of acute stroke patients," *Radiology*, vol. 254, no. 1, pp. 200–209, 2009.

[13] K. Kudo, M. Sasaki, K. Ogasawara, S. Terae, S. Ehara, and H. Shirato, "Difference in tracer delay–induced effect among deconvolution algorithms in ct perfusion analysis: Quantitative evaluation with digital phantoms," *Radiology*, vol. 251, no. 1, pp. 241–249, 2009.

[14] S. L. Zeger, K.-Y. Liang, and P. S. Albert, "Models for longitudinal data: a generalized estimating equation approach," *Biometrics*, pp. 1049–1060, 1988.

[15] E. J. Stanek, A. Well, and I. Ockene, "Why not routinely use best linear unbiased predictors (blups) as estimates of cholesterol, per cent fat from kcal and physical activity?," *Statistics in medicine*, vol. 18, no. 21, pp. 2943–2959, 1999.

[16] C. H. Morrell, L. J. Brant, J. D. Pearson, G. N. Verbeke, and J. L. Fleg, "Applying linear mixed-effects models to the problem of measurement error in epidemiologic studies," *Communications in Statistics-Simulation and Computation*, vol. 32, no. 2, pp. 437–459, 2003.

[17] V. Shetty, C. H. Morrell, and S. S. Najjar, "Modeling a cross-sectional response variable with longitudinal predictors: an example of pulse pressure and pulse wave velocity," *Journal of applied statistics*, vol. 36, no. 6, pp. 611–619, 2009.

[18] J. O. Ramsay and B. W. Silverman, *Applied functional data analysis: methods and case studies*. Springer, 2002.

[19] H.-G. Müller and U. Stadtmüller, "Generalized functional linear models," *Annals of Statistics*, pp. 774–805, 2005.

[20] G. M. James, J. Wang, and J. Zhu, "Functional linear regression that's interpretable," *The Annals of Statistics*, pp. 2083–2108, 2009.

[21] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[22] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[23] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[24] G. M. James, P. Radchenko, and J. Lv, "Dasso: connections between the dantzig selector and lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 1, pp. 127–142, 2009.

[25] ASIST, "Pma." `http://asist.umin.jp/index-e.htm`. Accessed August 2, 2016.

[26] D. I. Solutions, "Stroketool." http://www.digitalimagesolutions.de/. Accessed August 2, 2016.

[27] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

APPENDIX A

IRB APPROVAL

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

Institutional Review Board for Human Use

Exemption Designation
Identification and Certification of Research
Projects Involving Human Subjects

UAB's Institutional Review Boards for Human Use (IRBs) have an approved Federalwide Assurance with the Office for Human Research Protections (OHRP). The Assurance number is FWA00005960 and it expires on January 24, 2017. The UAB IRBs are also in compliance with 21 CFR Parts 50 and 56.

Principal Investigator:	Lirette, Seth T
Co-Investigator(s):	
Protocol Number:	**E160720001**
Protocol Title:	*A Statistical Approach to Computed Tomography Perfusion*

The above project was reviewed on 8/18/14. The review was conducted in accordance with UAB's Assurance of Compliance approved by the Department of Health and Human Services. This project qualifies as an exemption as defined in 45CFR46.101(b), paragraph 4.

This project received EXEMPT review.

Date IRB Designation Issued: 8/18/14

Margie Lawson, CIP

Designated Reviewer
Chair Designee

Investigators please note:

Any modifications in the study methodology, protocol and/or consent form/information sheet must be submitted for review to the IRB prior to implementation.

470 Administration Building
701 20th Street South
205.934.3789
Fax 205.934.1301
irb@uab.edu

The University of
Alabama at Birmingham
Mailing Address:
AB 470
1720 2ND AVE S
BIRMINGHAM AL 35294-0104

APPENDIX B

PAPER 2 DATA SIMULATION R CODE

```
## Loading Libraries
library(MASS)
library(lpSolve)
library(geepack)
library(lme4)
library(moments)
source(".../flrti_s.txt")

### Function used for Covariance Matrix
calcSigma<-function(X1,X2,l=1){
 Sigma<-matrix(rep(0,length(X1)*length(X2)),nrow=length(X1))
  for(i in 1:nrow(Sigma)) {
    for (j in 1:ncol(Sigma)) {
      Sigma[i,j]<-exp(-1/2*(abs(X1[i]-X2[j])/l)^2)
    }
  }
  return(Sigma)
}

outmatc = matrix(data=NA, nrow=24500, ncol=16)
outmatb = matrix(data=NA, nrow=24500, ncol=12)
m=1
```

```
for(a in c(15,30,50,100,500,1000,5000)) {
for(b in c(3,5,10,25,50,100,250)) {
for(c in 1:500) {

############################
### Generating Data #######
############################

tryCatch({
n.samples=a
n.draws=b
simnum=c
seednum = n.samples+n.draws+simnum*100
time<-seq(1,n.draws,len=n.draws)
values<-sapply(1:n.samples,function(i,j=seednum,D=n.draws){
  set.seed(j+i)
  x.star<-seq(-5,5,len=D)
  nval<-3
  f<-data.frame(x=seq(-5,5,l=nval),y=rnorm(nval,0,10))
  sigma.n<-0.2
  k.xx<-calcSigma(f$x,f$x)
  k.xxs<-calcSigma(f$x,x.star)
  k.xsx<-calcSigma(x.star,f$x)
  k.xsxs<-calcSigma(x.star,x.star)
  f.bar.star<-
    k.xsx%*%solve(k.xx+sigma.n^2*diag(1,ncol(k.xx)))%*%f$y
```

```
  cov.f.star<-k.xsxs-k.xsx%*%
     solve(k.xx+sigma.n^2*diag(1,ncol(k.xx)))%*%k.xxs
  mvrnorm(1,f.bar.star,cov.f.star)})
gammas<-t(values)
id = 1:n.samples
cont=rep(0,n.samples)
yesno=rep(0,n.samples)

for(k in 1:n.samples) {
  cont[k] = mean(values[,k]) + sd(values[,k])
          + skewness(values[,k]) + kurtosis(values[,k])
          + rnorm(1,0,2)
  yesno[k] = rbinom(1,1,(1/(1+exp(-(-2 + 0.005*
             (mean(values[,k]) + sd(values[,k])
            + skewness(values[,k]) + kurtosis(values[,k]))
            + rnorm(1,0,2))))))
}

yesno[1]=1
```

```
funcdat = data.frame(gammas, id, cont, yesno)


funcdat.long = reshape(funcdat[c("id", "cont", "yesno"
                       , grep("X",names(funcdat),value=T))]
                       , idvar = "id"
                       , varying=4:(ncol(gammas)+3)
                       , direction="long"
                       , timevar="time", v.names="gamma")


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

```
###############
### GLM #######
###############
time1 = proc.time()
cont.glm = glm(cont ~ gamma,data=funcdat.long
   [order(funcdat.long$id,funcdat.long$time),])
glm.time.c = proc.time() - time1

time1 = proc.time()
yesno.glm = glm(yesno ~ gamma,
  data=funcdat.long
    [order(funcdat.long$id,funcdat.long$time),]
    , family=binomial(link="logit"))
glm.time.b = proc.time() - time1

#MAE
glm.mae = mean(abs(resid(cont.glm)
  [seq(from=1,to=(n.samples*n.draws),by=ncol(gammas))]))

#RMSE
glm.rmse = sqrt(mean(resid(cont.glm)
  [seq(from=1,to=(n.samples*n.draws),by=ncol(gammas))]^2))

#brier score
glm.bs = mean((resid(yesno.glm)
 [seq(from=1,to=(n.samples*n.draws)
  ,by=ncol(gammas))] - funcdat$yesno)^2)
```

```
###############
### GEE #######
###############
time1 = proc.time()
cont.gee = geeglm(cont ~ gamma, id=id,
data=funcdat.long[order(funcdat.long$id,funcdat.long$time),]
 , family=gaussian, corstr="ar1")
gee.time.c = proc.time() - time1

time1 = proc.time()
yesno.gee = geeglm(yesno ~ gamma, id=id,
data=funcdat.long[order(funcdat.long$id,funcdat.long$time),]
 , family=binomial(link="logit"), corstr="ar1")
gee.time.b = proc.time() - time1

#MAE
gee.mae = mean(abs(cont.gee$residuals
 [seq(from=1,to=(n.samples*n.draws),by=ncol(gammas))]))

#RMSE
gee.rmse = sqrt(mean(cont.gee$residuals
 [seq(from=1,to=(n.samples*n.draws),by=ncol(gammas))]^2))

#brier score
gee.bs = mean((yesno.gee$fitted.values
 [seq(from=1,to=(n.samples*n.draws)
 ,by=ncol(gammas))] - funcdat$yesno)^2)
```

```
###################################
### 2-Stage Empirical Bayes #######
###################################
time1 = proc.time()
gamma.eb1 = lmer(gamma ~ time + (time | id),
 data=funcdat.long
   [order(funcdat.long$id,funcdat.long$time),])
tmpcoef = coef(gamma.eb1)$id
inter = NULL
slope = NULL


for(i in 1:length(funcdat$cont)) {
  inter[i] = tmpcoef[i,1]
  slope[i] = tmpcoef[i,2]
}
cont.eb = try(geeglm(cont~inter:slope, id=id, data=funcdat,
 family=gaussian, corstr="ar1"))
eb.time.c = proc.time() - time1

time1 = proc.time()
gamma.eb1 = lmer(gamma ~ time + (time | id),
 data=funcdat.long
   [order(funcdat.long$id,funcdat.long$time),])
tmpcoef = coef(gamma.eb1)$id
inter = NULL
slope = NULL
```

```
for(i in 1:length(funcdat$cont)) {
  inter[i] = tmpcoef[i,1]
  slope[i] = tmpcoef[i,2]
}
yesno.eb = try(geeglm(yesno ~ inter:slope, id=id,
 data=funcdat, family=binomial(link="logit"), corstr="ar1"))
eb.time.b = proc.time() - time1

#MAE
eb.mae = mean(abs(resid(cont.eb)))

#RMSE
eb.rmse = sqrt(mean(resid(cont.eb)^2))

#brier score
eb.bs = mean((resid(yesno.eb) - funcdat$yesno)^2)

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

```
###############
### FLiRTI ####
###############
time1 = proc.time()
contcv = flrti.cv(funcdat$cont,gammas,
 sigma=seq(0.0001,0.001,by=0.0001),
   weight=seq(0.02, 0.2, by=0.02))
flirti.time.c.cv = proc.time() - time1

time1 = proc.time()
contfit = flrti(funcdat$cont,gammas,
 sigma=contcv$minsigma,weight=contcv$minweight)
flirti.time.c.fit = proc.time() - time1

time1 = proc.time()
yesnocv = flrti.cv(funcdat$yesno,gammas,
 sigma=seq(0.0001,0.001,by=0.0001),
   weight=seq(0.02, 0.2, by=0.02))
flirti.time.b.cv = proc.time() - time1

time1 = proc.time()
ynfit = flrti(funcdat$yesno,gammas,
 sigma=yesnocv$minsigma,weight=yesnocv$minweight)
flirti.time.b.fit = proc.time() - time1

#MAE
flirti.mae = mean(abs(contfit$residuals))
```

```
#RMSE
flirti.rmse = sqrt(mean(contfit$residuals^2))

#brier score
flirti.bs = mean((ynfit$fitted.values - funcdat$yesno)^2)

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

}, error=function(e){})

outmatc[m,1]  = a
outmatc[m,2]  = b
outmatc[m,3]  = c
outmatc[m,4]  = glm.mae
outmatc[m,5]  = glm.rmse
outmatc[m,6]  = glm.time.c[3]
outmatc[m,7]  = gee.mae
outmatc[m,8]  = gee.rmse
outmatc[m,9]  = gee.time.c[3]
outmatc[m,10] = eb.mae
outmatc[m,11] = eb.rmse
outmatc[m,12] = eb.time.c[3]
outmatc[m,13] = flirti.mae
outmatc[m,14] = flirti.rmse
outmatc[m,15] = flirti.time.c.cv[3]
outmatc[m,16] = flirti.time.c.fit[3]
```

```
outmatb[m,1]  = a
outmatb[m,2]  = b
outmatb[m,3]  = c
outmatb[m,4]  = glm.bs
outmatb[m,5]  = glm.time.b[3]
outmatb[m,6]  = gee.bs
outmatb[m,7]  = gee.time.b[3]
outmatb[m,8]  = eb.bs
outmatb[m,9]  = eb.time.b[3]
outmatb[m,10] = flirti.bs
outmatb[m,11] = flirti.time.b.cv[3]
outmatb[m,12] = flirti.time.b.fit[3]

m=m+1
  print(a)
  print(b)
  print(c)
}
}
}

write.csv(outmatc, file="...\continuous")
write.csv(outmatb, file="...\binary")
```