

**An Iterative Approach to Hemodynamic Response Function
Temporal Derivatives in Statistical Parametric Mapping for
Functional Neuroimaging**

BY
JOHN MUSCHELLI III

A thesis submitted to Johns Hopkins University in conformity with the
requirements for the degree of Master of Science

Baltimore, Maryland
January, 2011

**Copyright 2011 by John Muschelli III
All rights reserved**

ABSTRACT

In this thesis, I proposed a method for estimating the shift parameter (δ) of hemodynamic response function (hrf) in functional MRI (fMRI) data. Using a Taylor series approximation around $\delta = 0$, the method fits a least squares estimate using the hrf and its temporal derivative and an estimate of δ is obtained. The regression is re-run with the Taylor expansion centered at the new estimate. This process iterates until convergence and regresses only the hrf with the estimated δ to approximate the true model. Simulations were run to review the method's testing performance and estimation, comparing to other common methods in use. A real-world application was done to show computational cost and plausibility of this method. Overall, the method estimates δ well under certain circumstances and has good estimation properties when iterated.

ACKNOWLEDGEMENTS

I would like to acknowledge first and foremost my advisor Dr. Brian Caffo. This work would not have been possible without his guidance, training, and enthusiastic spirit for statistics. I would also like to thank Dr. Graham Redgrave for providing the data for analysis in this work and serving as a thesis reader. Thanks also to Drs. Ciprian Craiceanu and Michelle Carlson for serving as thesis readers. I would also like to thank Carol Thompson for helping me with problems whenever they arose.

I would also express my gratitude for the Department of Biostatistics, where the education and atmosphere of collaboration has been nothing short of spectacular. Finally, thanks goes to my family and friends, whose support was crucial in my motivation and determination in my program.

Contents

Table of Contents	iii
List of Tables	vi
List of Figures	vii
1 Scope of Work	1
2 Introduction	2
3 Background	3
3.1 Use of the Hemodynamic Response Function	3
3.2 Questions for Investigator	5
3.3 Goals of Analysis with the HRF	6
3.4 Different Inter-stimulus Times	10
3.5 Preprocessing	13
3.6 Reviewing Purposes of an Analysis	13
3.6.1 Estimating Delta vs. Detecting Activation	13
3.6.2 Estimating the Shift within a Region of Activation	14
4 Methods	15
4.1 Modeling Approaches	15
4.1.1 Non-linear Models	15
4.1.2 The Linear Regression Model	16

4.1.3	Taylor Approximation of the True Model	16
4.1.4	Block Maximized Iterated Estimator for the Hemodynamic Lag	18
4.1.5	Iterated Ratio Estimators for the Hemodynamic Lag	18
4.1.6	Variance Shrinkage in the Ratio Estimator	20
4.2	A Review of Testing for Activation with a Known HRF	21
4.2.1	Standard T-tests for β_1	21
4.2.2	T-test for both parameters	21
4.2.3	Calhoun's one-sided F-statistic	22
5	Simulations	24
5.1	Convergence	25
5.2	Performance Diagnostics	27
5.3	Estimation of δ and Performance of Scenarios	29
5.3.1	High SNR	30
5.3.2	Medium SNR	36
5.3.3	Low SNR	42
5.3.4	Non-activated Voxels	48
5.3.5	Conclusion	49
5.3.6	When to use the IRE or SRE	50
5.4	Computational Cost	51
6	Application	53
6.1	Description of the Subjects	53
6.2	Iowa Gambling Task	53
6.3	Scope of Analysis and Preprocessing	54
6.4	Analysis Details	55
6.5	Results	56

7 Conclusion	59
7.1 Questions Revisited	59
7.2 Simulation Results Regarding Testing for Activation	60
8 Further Areas of Research	61
8.1 Comparing $\hat{\delta}$'s across Methods	61
8.2 Using δ in Approximating the HRF	61
8.3 Going to 3D Space	62
8.4 Different Convergence Limits	62
8.5 Use of δ as a Grouping Factor	63
9 Summary	64
Bibliography	64
10 Appendix	68
10.1 R code	69
10.1.1 HRF and Design Functions	69
10.1.2 Simulations	71
10.1.3 Data Compression	74
10.1.4 Analysis of Subjects in Application	75
10.2 Maps and Histograms for All Subjects	81

List of Tables

5.1	Convergence Rates for Active Voxels, High SNR	26
5.2	Convergence Rates for Active Voxels, Medium SNR	26
5.3	Convergence Rates for Active Voxels, Low SNR	26
5.4	Average Number of Iterations for Converged Active Voxels	27
5.5	Convergence Rates for Non-active Voxels	27
5.6	AUC for Lag = 1, SNR = 0.5	31
5.7	AUC for Lag = 2, SNR = 0.5	32
5.8	AUC for Lag = 4, SNR = 0.5	33
5.9	AUC for Lag = 1, SNR = 0.25	37
5.10	AUC for Lag = 2, SNR = 0.25	37
5.11	AUC for Lag = 4, SNR = 0.25	37
5.12	AUC for Lag = 1, SNR = 0.125	43
5.13	AUC for Lag = 2, SNR = 0.125	43
5.14	AUC for Lag = 4, SNR = 0.125	43

List of Figures

3.1	Parameters of the HRF	5
3.2	Hemodynamic Shift	8
3.3	HRF and the Temporal Derivative	9
3.4	Change from baseline with short inter-stimulus time	11
3.5	Change from baseline with sufficient inter-stimulus time	12
5.1	ROC for Short Lag, High SNR	31
5.2	ROC for Medium Lag, High SNR	32
5.3	ROC for Long Lag, High SNR	33
5.4	Box Plot for Bias of β_1 for High SNR	35
5.5	Box Plot for Bias of δ for High SNR	36
5.6	ROC for Short Lag, Medium SNR	38
5.7	ROC for Medium Lag, Medium SNR	39
5.8	ROC for Long Lag, Medium SNR	40
5.9	Box Plot for Bias of β_1 for Medium SNR	41
5.10	Box Plot for Bias of δ for Medium SNR	42
5.11	ROC for Short Lag, Low SNR	44
5.12	ROC for Medium Lag, Low SNR	45
5.13	ROC for Long Lag, Low SNR	46
5.14	Box Plot for Bias of β_1 for Low SNR	47
5.15	Box Plot for Bias of δ for Low SNR	48
5.16	Box Plot for Bias of β_1 for Non-Activated Voxels	49

6.1	IGT Paradigm	54
6.2	Spatial Maps and Histograms of Lags for Subject 10	58
10.1	Spatial Maps and Histograms of Lags for Subject 9	81
10.2	Spatial Maps and Histograms of Lags for Subject 10	82
10.3	Spatial Maps and Histograms of Lags for Subject 11	82
10.4	Spatial Maps and Histograms of Lags for Subject 12	83
10.5	Spatial Maps and Histograms of Lags for Subject 14	83
10.6	Spatial Maps and Histograms of Lags for Subject 15	84
10.7	Spatial Maps and Histograms of Lags for Subject 16	84
10.8	Spatial Maps and Histograms of Lags for Subject 17	85
10.9	Spatial Maps and Histograms of Lags for Subject 18	85
10.10	Spatial Maps and Histograms of Lags for Subject 19	86
10.11	Spatial Maps and Histograms of Lags for Subject 20	86
10.12	Spatial Maps and Histograms of Lags for Subject 21	87
10.13	Spatial Maps and Histograms of Lags for Subject 22	87
10.14	Spatial Maps and Histograms of Lags for Subject 23	88
10.15	Spatial Maps and Histograms of Lags for Subject 24	88
10.16	Spatial Maps and Histograms of Lags for Subject 25	89
10.17	Spatial Maps and Histograms of Lags for Subject 26	89
10.18	Spatial Maps and Histograms of Lags for Subject 27	90
10.19	Spatial Maps and Histograms of Lags for Subject 28	90
10.20	Spatial Maps and Histograms of Lags for Subject 29	91

Chapter 1

Scope of Work

The purpose of this thesis is to review common tests for activation in the brain and propose a new method for estimating the hemodynamic shift with an iterative use of the temporal derivative. The models use a first order Taylor series expansion proposed by Friston (Friston, 2006). The main discussion points will be how well the method estimates the shift and how biased the regression estimates are when using an estimated shift. Also, this thesis will cover points about the next steps in analysis after the shift has been estimated. It will then focus on a testing paradigm and potential problems with certain methods. Sensitivity will be defined as the ability to identify truly activated voxels, whereas specificity is the ability to identify non-activated voxels. The section on testing will describe how sensitive these tests are and how the methods perform using certain tests as criteria for activation.

The results from Liao et al. (2002) have inspired much of the work. Although the method described in that work uses a set of basis functions for the hrf and its temporal derivative, it is an implementation of the ratio estimator in method. That method uses a more sophisticated choice of the hrf, called the spectral estimator, but still use ratio estimators to estimate δ . The method discussed here is an alternative and different use of the temporal derivative to estimate δ and ultimately estimate activation with the use of an assumed hrf.

Chapter 2

Introduction

Functional magnetic resonance imaging (fMRI) measures blood-oxygen-level dependent (BOLD) responses to estimate neuronal activation within the brain (Friston, 2006). Data for fMRI is taken at a volumetric pixel (voxel) -wise level. Each voxel represents a small portion of the brain, typically between a cubic centimeter and millimeter in size depending on acquisition parameters. Neuronal activity consumes oxygen and deoxygenates hemoglobin and causes arterioles to dilate, causing blood flow to increase the amount oxygenated hemoglobin in the capillaries (Kruggel and Von Cramon, 1999). The deoxygenated hemoglobin has different magnetic properties and the MRI detects these differences (Lorberbaum et al., 2010).

The change from oxygenated to deoxygenated hemoglobin follows a stereotyped pattern relating to blood flow and volume. This change in BOLD responses is called the hemodynamic response and is modeled by the hemodynamic response function (hrf), as seen in Figure 3.1 (Glover, 1999; Kruggel and Von Cramon, 1999; Rosen et al., 1998).

Chapter 3

Background

The choice of the hrf can vary when analyzing different event-related paradigms. There are a number of considerations and steps taken before any data collection is done which are choice of the hrf, reviewing the analysis goals, and carefully designing the event-related paradigm. In principle, one would have an analysis plan mapped out *a priori*. In practice, a model is commonly chosen after the data is collected. This chapter discusses the use of the hrf in analysis as well as things to consider before running any analyses.

3.1 Use of the Hemodynamic Response Function

In fMRI analysis, the hrf is widely used in regression to form models for testing and estimation. Some analyses vary in the choice of the hrf, but the methodology is fundamentally the same in a regression framework: preprocess the data, choose a response function, and analyze the data with a model. Thus, it is important to consider the choice and use of the hrf and how the hrf itself can vary.

There is not one set form of the hrf and it varies across subjects, sometimes due to non-physiological factors (Aguirre et al., 1998; Handwerker et al., 2004). It also varies within subjects over time due to growth, aging, disease processes, stimulants etcetera and across tissue due to variation in physiology. The response for similar types of tissues have been shown to be comparable in some cases (Buckner et al.,

1998). One explanation for the varying responses is the differences in vasculature in these areas (Friston, 2006).

Multiple models for the shape of the hrf have been proposed to approximate the true response. The assumed hrf for this review is the one pictured in Figure 3.1, which is a mixture of two gamma functions (Friston et al., 1998; Lange and Zeger, 1997). Being the default model for the hrf in the popular SPM software, the hrf used is perhaps the most widely assumed shape. Other functions proposed have been based on the normal or Poisson distributions (Friston et al., 1998). The hrf has been empirically estimated to be similar to the shape of the difference of two gamma functions, but with slightly different parameters than the assumed hrf here (Glover, 1999). A set of basis functions, which have shape similar to that of Figure 3.1, have been used but use different procedures in analysis (Liao et al., 2002).

The hrf is defined by a set of parameters, shown in Figure 3.1. The hrf used is from the SPM8 package (<http://www.fil.ion.ucl.ac.uk/spm/>) and one similar to the ones discussed in Friston (2006). The onset is the time when a stimulus occurs. The lag is the time from onset until the highest activation, or the peak, and occurs typically 5-8 seconds after stimulation. The dispersion parameter is how much time occurs during the response, which is typically 3-4 seconds (Kruggel and Von Cramon, 1999). For example, a larger dispersion would result in a peak more spread across time, having a high response for more time. The shift is the time from the onset until the beginning of activation (Figure 3.2). The undershoot occurs after the peak and is where the signal slightly drops below the baseline activation. There are a number of reasons why this may occur (Schroeter et al., 2006). Essentially, the undershoot occurs because tissues replenish oxygen stores after blood flow has returned to the baseline level, causing a drop in oxygenated blood (Buxton et al., 1998).

Estimation of these parameters is commonly of interest. With a known hrf, a typical way of estimating these parameters is with models using the temporal derivative.

Varying any of these parameters can change the shape and timing of the peaks and undershoots. The choice of these parameters may change depending on the type of events being studied. Thus, choosing these parameters and the analysis goals must be considered by the investigator before analyzing the data.

Anatomy of the Hemodynamic Response Function

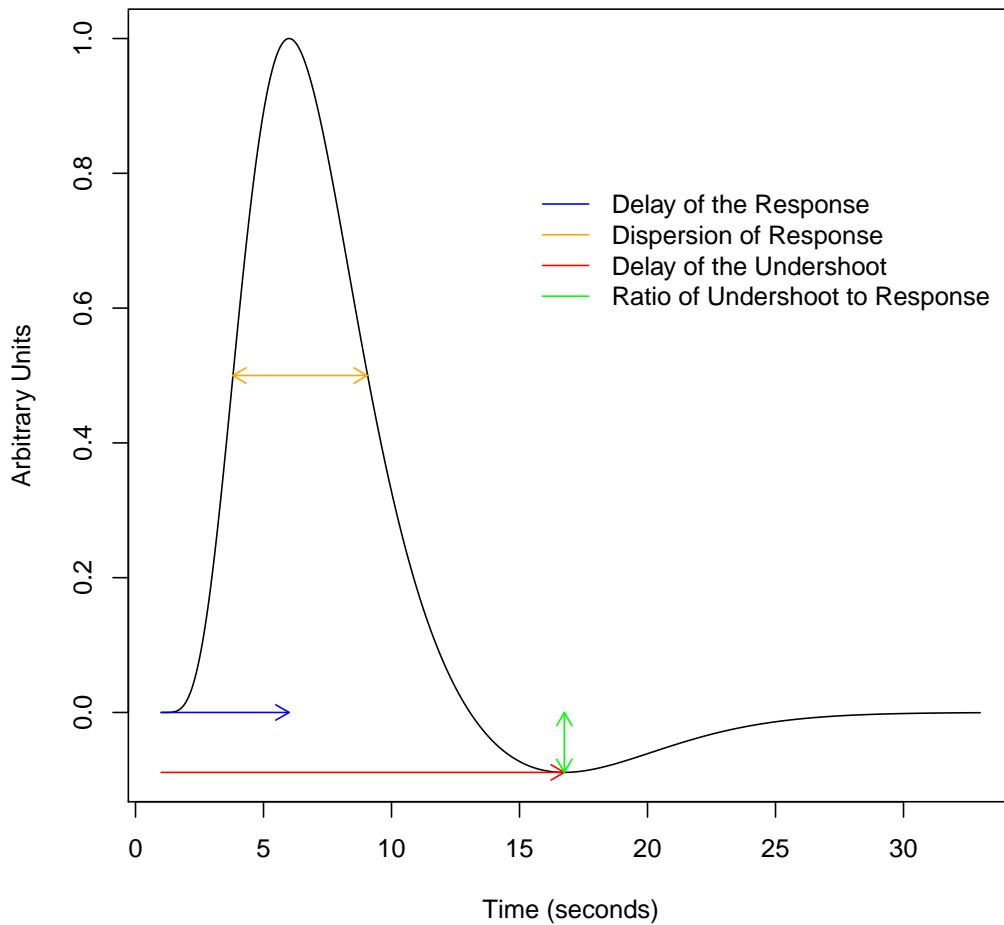


Figure 3.1: Parameters of the HRF

3.2 Questions for Investigator

Before beginning any analysis with the hrf or its derivatives, one must ask: “what is the goal of including the temporal derivative in the model of fMRI analysis”?

Most important is whether estimating parameters of the hrf is of interest, or whether they are simply nuisance parameters that need to be dealt with to determine regions of activation. As we outline below, the issue of use of the temporal derivative of the hrf in testing has been considered in the literature (see Section 3.6.1 below for more discussion), and it is a standard option in most fMRI software. Though used commonly, entirely satisfactory solutions of how and why to include the hrf temporal derivative still elude researchers.

In practice, more specific questions of the analysis goals need to be answered. Is the shift itself intrinsically of interest? Can the estimate of the shift be used as a plug-in estimate for subsequent testing of activation? Can the shift be assumed to be constant over the whole brain? Does using the temporal derivative improve the meaningfulness and/or utility of the inferences we draw about the data? These questions will be revisited in the conclusion section.

3.3 Goals of Analysis with the HRF

One typical goal when using the hrf is to correctly model the fMRI response to detect localized activation associated with an experimental paradigm performed in the scanner. In fMRI event-related studies, the investigator has a series of times of stimuli and the subsequent activation values from MRI output. Ideally, the hrf will accurately model the trend in the data and the effects of concurrent stimuli are additive. Because the effects are assumed to be additive, a convolution with the stimulus times and the hrf is used to produce a design vector that should approximate the actual response. In this case, linear regression should estimate the unknown parameter well. Our principal discussion will relate to such event-related studies where there is sufficient “temporal resolution” of this trend. This resolution refers to the hrf returning to a baseline activation level before another stimulus occurs. Different inter-stimulus times (IST) will be discussed in section 3.4 below.

One common yet difficult problem in activation studies in fMRI is that the hrf is usually unknown. Moreover, as we will discuss more thoroughly below in Section 4.1.1, non-linear estimation of its shape is inherently an ill posed problem and adds a non-trivial computational burden from non-linear maximization. Hence, as is standard in fMRI analysis, we assume that the shape of the hrf is known.

However, with a known shape, a remaining issue is identifying the hemodynamic shift or the hemodynamic lag. Notice that, if one assumes that the response occurs immediately after the stimulus (i.e. a shift of zero) and one wants to estimate the lag, then the model can not use a fixed hrf shape and must use different estimation techniques than described here to attempt to identify the time from stimuli when activation is largest. In this context, if one correctly identifies the shape of the hrf, estimating the hemodynamic shift is equivalent to estimating the hemodynamic lag (Figure 3.2).

Hemodynamic Response Function Shift

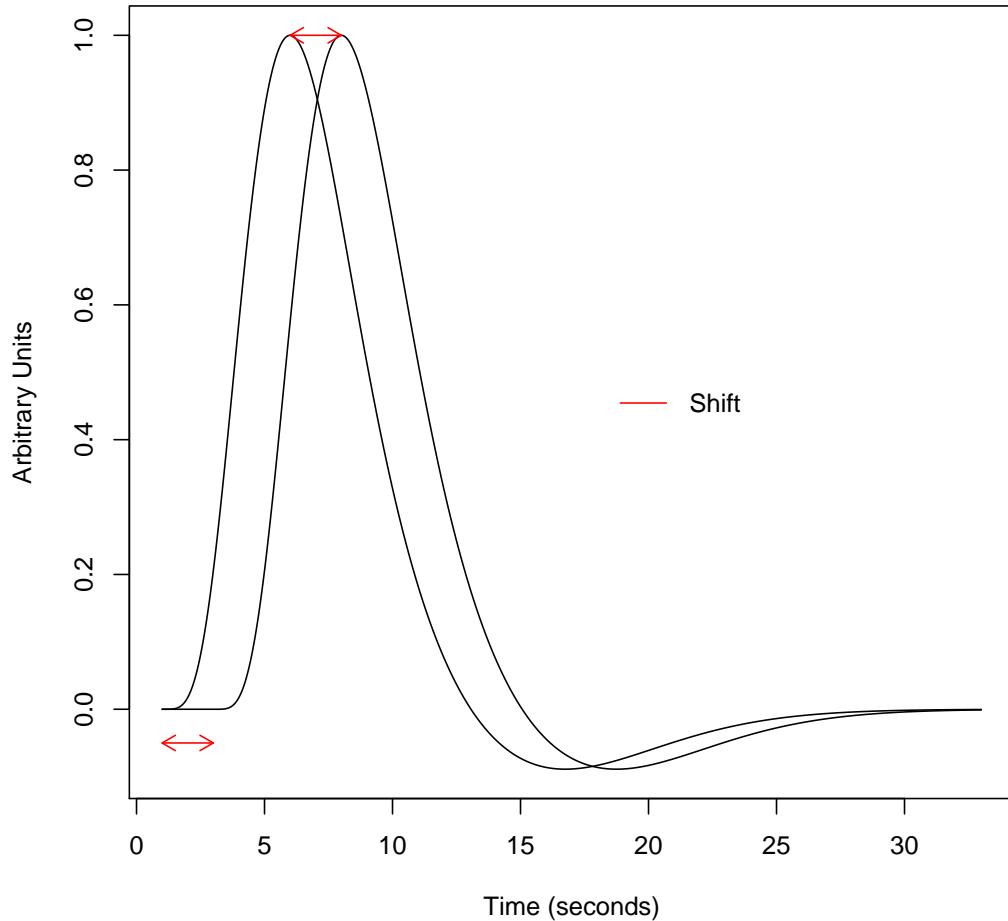


Figure 3.2: Hemodynamic Shift

From this point, the terms shift, lag, and δ will be used interchangeably. One of the main methods of estimating the hemodynamic shift is by the use of temporal derivatives. The derivative of the canonical hrf is shown in Figure 3.3. The use of temporal derivatives in linear regression models will be the main focus of our method.

Before a description of methods, we will discuss the form of the stimulus times, preprocessing of the data, and some investigative questions of analysis.

Hemodynamic Response Function and Derivative

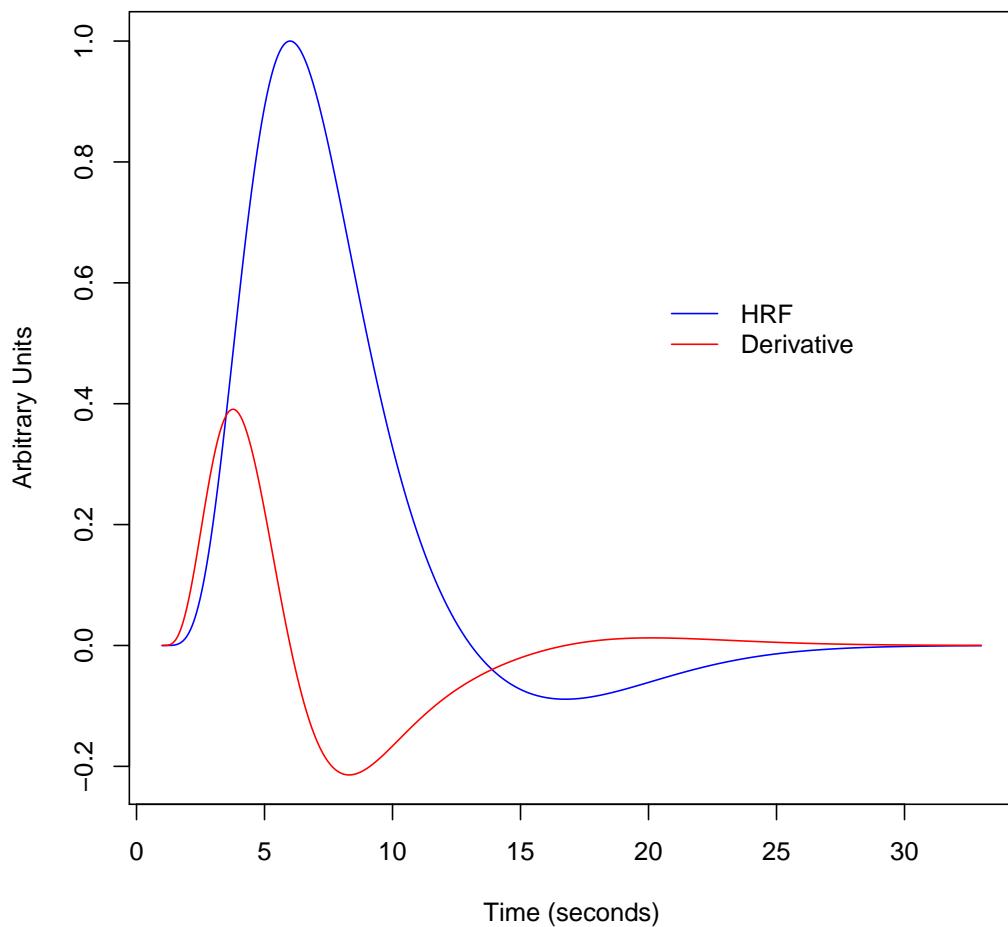


Figure 3.3: HRF and the Temporal Derivative

3.4 Different Inter-stimulus Times

Inter-stimulus times were considered because the convolution has an additive effect if stimulus events are too close to each other. The methods may not do as well if the stimuli are too rapid, since they are no longer estimating an event-related design. The rapid stimuli do not allow the function to return to baseline. If the convolved function does not return to baseline, the results may give a biased estimate of the baseline, which can affect estimation of β parameters. Figure 3.4 shows a hrf convolved with a stimulus that had an event every 10 seconds. Figure 3.5 has a convolved function with an IST of 30 seconds. Note that using a 10-second IST does not allow for sufficient time to return to a baseline, which was zero in this figure. The models in the simulations assume that there is sufficient time for the activity to return to a baseline level after the undershoot. The method may do poorly for close inter-stimulus times and further study will be done.

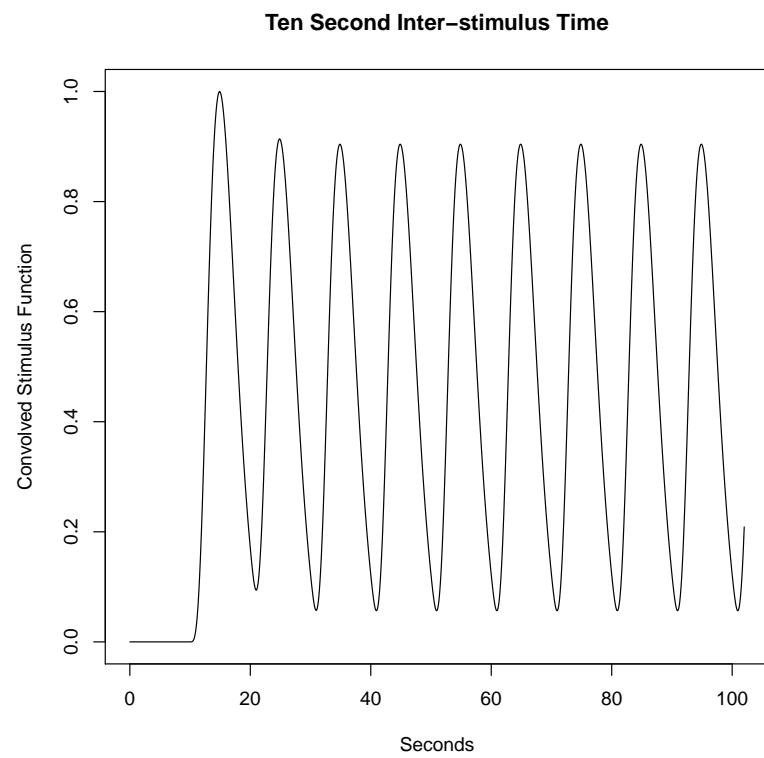


Figure 3.4: Change from baseline with short inter-stimulus time

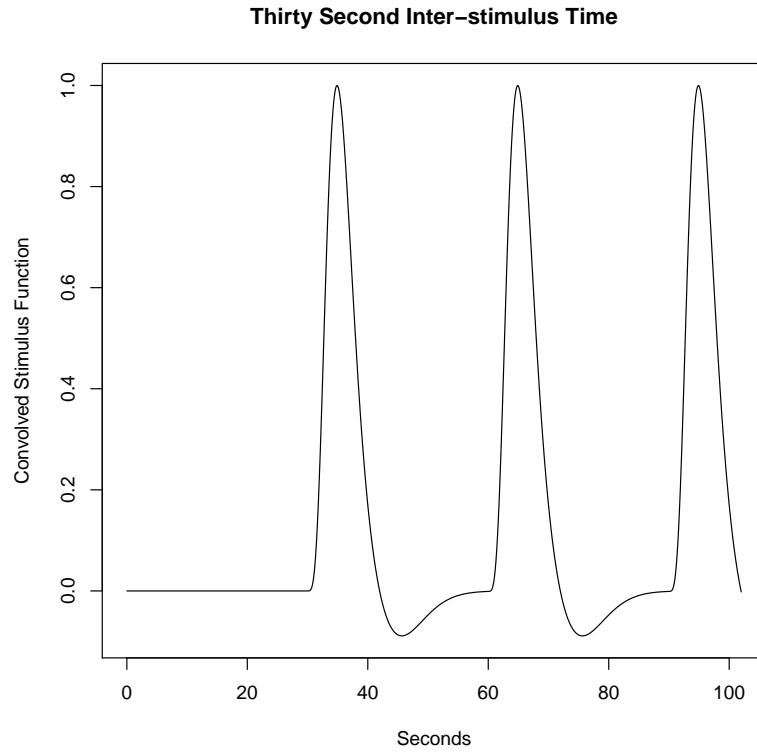


Figure 3.5: Change from baseline with sufficient inter-stimulus time

The convolved design function does not reach the maximum peak of activation when the paradigm has a short IST because the undershoot in the previous stimulus is convolved with the peak of the current stimulus. One method that may solve this problem is to only regress the convolution to a portion of the response after the first peak of activation. One of the ramifications of using a longer IST time is that the number of scans to maintain the same number of events increases, which may introduce other difficulties, like participant fatigue and/or lack of attention to the task. Again, this analysis assumes that the IST is sufficiently long for a return to baseline as in figure 3.5. Friston (2006) describes how the convolution model should do well to estimate parameters even when IST are small, but the long IST was chosen to prove our method could estimate δ well.

3.5 Preprocessing

Before any analysis, the fMRI data must be preprocessed in order to attempt remove any measurement effects from the data collection process. The goal of the preprocessing is that after an analysis is done, the analysis should detect true signal and not an artifact.

The description of methodology here assumes the data has been motion corrected, realigned, smoothed, normalized and transformed to a template coordinate system such as the Montreal Neurological Institute (MNI) or Talarach and Tournoux space, or equivalent (Evans et al., 1992; Friston, 2006; Talairach and Tournoux, 1988). The proof of method neglects the spatial component of the data in that there is no correlation between voxels in the simulations. Further analysis can use common methods for analyzing correlated data along with methods here to estimate δ more effectively in actual fMRI data.

3.6 Reviewing Purposes of an Analysis

In this section, we will discuss some aspects of analysis that need to be decided by the investigator. Whether the investigator primarily wants to detect activation or parameters affects the analysis plan. Also, having a region with known activation available, that can be used for estimation and comparison of methods, will affect the use of a voxel-specific or clustered estimation of δ .

3.6.1 Estimating Delta vs. Detecting Activation

The estimation of δ is not a fundamentally necessary procedure if the primary purpose of an analysis is to detect activation. Methods such as the cone T statistic maximize the T statistics over a range of plausible shifts, $[\Delta_1, \Delta_2]$ (Friman et al., 2003). Worsley and Taylor (2006) propose a method to find p-values for these T-statistics to quantify regions of activation. One could identify the $\delta \in [\Delta_1, \Delta_2]$ that maximized this function

as an estimate of δ , but the estimate may not be useful if the primary purpose is activation detection.

One identifiable problem in this method is that the true shift may not be in the “plausible interval”. One way to avoid this would be to expand $[\Delta_1, \Delta_2]$ over a larger interval and repeat the maximization process. However, this increases computation time and the complexity of the process and the method is not inherently an estimation process.

Many applications have the primary purpose of locating and identifying areas of activation within the brain. If one correctly knows the shift, the testing procedure should be relatively straightforward, as long as the assumptions of the linear regression hold of course. Thus, we show that with an accurate estimate of δ , using the plug-in estimate can yield a close approximation to the true model. Using the temporal derivative in the model can give different statistics to detect activation, which will be discussed below.

3.6.2 Estimating the Shift within a Region of Activation

In some instances, an investigator may know of a region of activation or cluster of similar voxels a priori. In this case, if the investigator assumes a common shift for these areas of the brain, then averaging δ over this area of activation will result in a better estimate than estimating each voxel separately. The rationale here is that if these voxels have the same shift, then a measure of central tendency (i.e. mean or median) should estimate δ better than one δ_i alone from one voxel. The methods below are applicable whether assuming a common lag or not (Section 4.1.3), and can act as a diagnostic for comparison of lags between regions.

Chapter 4

Methods

4.1 Modeling Approaches

In this chapter, we focus on approximate methods based on temporal derivatives and linear models. These models will be used to answer the analysis questions posed earlier. Other methods described below are non-linear models and block maximization. The main focus is on using ratio estimators with a Taylor series approximation. Each method assumes a known hrf.

4.1.1 Non-linear Models

In our discussion, we use linear models. In contrast, one could parameterize the hrf with a non-linear model and has been used in other analysis methods (Calhoun et al., 2004). These methods are becoming more common and this section acknowledges their use but explains why they were not employed here. A non-linear model estimates the parameter, in this case δ , and powers of the parameter, i.e. δ^2, δ^3 , and so on. Liao et al. (2002) point out that iterative non-linear methods are computationally time consuming. These models can also add unnecessary complexity to the analysis.

There is also the issue of identifiability. The signal-to-noise ratio in fMRI is typically very weak. Moreover, only relatively small clusters of voxels tend to be activated. In the non-activated voxels, there are no hrf parameters to estimate. The problem of estimating the hrf is one of estimating a non-linear function that is only

present in a small unknown percentage of thousands of voxels. Hence, we simplify the problem and focus on a parameter of particular importance (the lag) while assuming the shape of the hrf is known. This has the added benefit of keeping computation cost relatively low and model fitting relatively simple.

The methods proposed were also kept within a regression framework. Other methods, such as independent component analysis (Calhoun et al., 2003) are not discussed here. The rationale is that regression is arguably the most common form of analysis, and methods using this framework would likely have more applications in practice.

4.1.2 The Linear Regression Model

The argument behind the use of the temporal derivative lies in a first order Taylor series expansion of the convolution of the hemodynamic response function (hrf) and the stimulus function (s), which will be called the design function (x). The overall model that we are trying to approximate in a linear regression analysis is described here: let $Y(t)$ be the fMRI intensities for times t , and δ_i is the hemodynamic shift for each voxel i , then the model is:

$$Y_i(t) = \beta_{0i} + \beta_i x(t - \delta_i) + \varepsilon_i \quad (4.1)$$

where $\varepsilon_i \sim N(0, \sigma_i^2)$ (independent across voxels). This model will be referred to as the true model.

4.1.3 Taylor Approximation of the True Model

Since we do not know δ and want to estimate it, we use a Taylor series expansion. By the mean value theorem or equivalently a first order Taylor expansion, for values of δ_i near 0, $x(t - \delta) \approx x(t) - x'(t)\delta$. Hence Equation 4.1 becomes

$$Y_i(t) \approx \beta_{0i} + \beta_i x(t) - \beta_i \delta_i x'(t) + \varepsilon_i. \quad (4.2)$$

This approximation works well if $\delta \approx 0$. Because Model (4.2) is difficult to fit, a related model is often fit

$$Y_i(t) \approx \beta_{0i} + \beta_{1i}x(t) - \beta_{2i}x'(t) + \varepsilon_i. \quad (4.3)$$

It is important to emphasize the difference between Models 4.2 and 4.3. Model 4.2 is exact up to the mean value approximation. Model 4.3 contains Model 4.2 as a special case when $\beta_{1i} = \beta_i$ and $\beta_{2i} = \beta_i\delta$. However, without further restrictions it also contains a much broader parameter space. For example, β_{1i} and β_{2i} could have different signs, which cannot happen if Model (4.2) is correct.

As such, fitting Model (4.3) with ordinary least squares (OLS) at each voxel does not honor parameter constraints, a fact that we will revisit later. Let $\hat{\beta}_{1i}$ and $\hat{\beta}_{2i}$ be the OLS estimates of β_{1i} and β_{2i} obtained by fitting Model 4.3. Then (again up to the mean value approximation) $\hat{\beta}_{2i}$ should approximate $\beta_{1i}\delta$, and $\hat{\delta}_i = \frac{\hat{\beta}_{2i}}{\hat{\beta}_{1i}}$. The obvious step taken in this analysis was to refit the model with the estimated δ from the model in order to approximate the true model. This refitting with the estimated δ will be referred to as the simple ratio estimator (SRE).

Voxel-wise vs. Clustered Estimation

With these approximations, the temporal derivative can be used to estimate a voxel-wise or cluster-specific estimate of δ . Each voxel has different attributes such as proximity to vasculature, vascular density, and vascular dilation compliance (Biswal et al., 2007). These different properties of voxels can affect the temporal shift of activation. Therefore, the assumption that voxels have a common lag time may be frequently violated. The difference in estimation is that some measure (i.e. mean, median) will be used to describe the common δ over similar voxels. The difference between the two models is illustrated below.

$$\text{Common shift } Y_i(t) = \beta_{i0} + \beta_{i1}x(t - \delta) + \varepsilon_i; \quad \varepsilon_i \sim N(0, \sigma_i^2)$$

$$\text{Voxel-specific shift } Y_i(t) = \beta_{i0} + \beta_{i1}x(t - \delta_i) + \varepsilon_i; \quad \varepsilon_i \sim N(0, \sigma_i^2)$$

Using this method assuming a voxel-wise or cluster-specific lag time can allow for comparison between estimates for regions under both conditions. Thus, comparing the voxel-wise estimates to the clustered ones can check if the common lag assumption is reasonably upheld. Thus, a ratio estimator can act as a comparative tool for δ between a voxel and a cluster if the estimate of δ is accurate.

4.1.4 Block Maximized Iterated Estimator for the Hemodynamic Lag

The SRE outlined previously has some theoretical flaws, which we sought to improve upon. First, the mean value approximation is centered at 0. Secondly, no global maximization is attempted. Thirdly, the estimator does not honor the parameter constraints of $\delta > 0$ and $\text{sign}(\beta_1) = \text{sign}(\beta_2)$. We sought to improve on some of these flaws with an iterated estimator *while still remaining in a regression framework*. Our first inclination was to consider OLS by minimizing

$$\sum_t ||Y_i(t) - \{\beta_i x(t) - \beta_i \delta_i x'(t)\}||^2$$

for both β_i and δ . If one separately estimates δ_i and β_i via block maximization, two regression equations are obtained: one for β_i with δ_i fixed at the previous estimate then one for δ_i with β_i fixed. However, though very appealing to write down, this approach did so poorly in simulations that we do not mention it further here.

4.1.5 Iterated Ratio Estimators for the Hemodynamic Lag

Since block maximization did not perform well and we still sought to improve upon the SRE, we derived a more general approach using an iterated ratio estimator (IRE). This approach aims to account for the Taylor series approximation being poor when δ_i is not close to zero. The iterated approach of this ratio estimator re-centers the Taylor expansion around the estimate of δ and iterates. The expansion of $x(t - \delta)$

around δ_0 then becomes:

$$x(t - \delta) \approx x(t - \delta_0) - x'(t)(\delta - \delta_0). \quad (4.4)$$

Under Approximation (4.4), Model (4.1) becomes

$$Y_i(t) \approx \beta_{0i} + \beta_i x(t - \delta_0) - \beta_i(\delta_i - \delta_0)x'(t - \delta_0) + \varepsilon_i. \quad (4.5)$$

The regression analog to Equation (4.5) is then:

$$Y_i(t) \approx \beta_{0i} + \beta_{1i}x(t - \delta_0) - \beta_{2i}x'(t - \delta_0) + \varepsilon_i. \quad (4.6)$$

If one fits Model (4.5) using Model (4.6), then $\hat{\beta}_{1i}$ estimates $\hat{\beta}_i$, $\hat{\beta}_{2i}$ estimates $\hat{\beta}_i(\delta_i - \delta_0)$.

Hence, a the associated ratio estimator is $\hat{\delta} = \frac{\hat{\beta}_{2i}}{\hat{\beta}_{1i}} + \delta_0$.

Hence, our ratio estimator is as follows. Initiate $\delta_0 = 0$.

1. Fit Model (4.6) using OLS;
2. Set $\hat{\delta}_\Delta = \frac{\hat{\beta}_{2i}}{\hat{\beta}_{1i}}$
3. Estimate $\hat{\delta}_i = \tilde{\delta}_\Delta + \delta_0$ (note changing $\hat{\delta}_\Delta$ to $\tilde{\delta}_\Delta$ and see Equation (4.7));
4. Set $\delta_0 = \hat{\delta}_i$
5. Go to Step 1.

A more explicit expansion on this method is in section 10. Note that if this algorithm is run for one iteration, the standard ratio estimator is obtained. Otherwise, the Taylor series is iteratively re-centered to obtain a better estimate.

The goal is for the model with the temporal derivative to converge to the true model. This should occur if the estimate for the shift is close to δ . The algorithm is considered to converge if the estimate of $\frac{\hat{\beta}_{2i}}{\hat{\beta}_{1i}} < 10^{-8}$. Hence the change from one iteration to the next was essentially zero if the algorithm converged. Since we constrain that $\delta > 0$, any estimate of δ that is negative is estimated to be zero and

the iteration ended, which was noted as “zeroing out”. Similarly, δ was constrained to be in “plausible” lag times which were defined as less than 6 seconds. Six seconds was chosen as the cutoff since any shifts of 6 seconds were noted as “late” by Buckner et al. (1998), which is referred to as “maxing out” of the iteration. A review of other papers had a maximum of 4 seconds for the shift. The maximum shift in our parameter space was higher so that the method may converge above or below the maximum lag estimated (4 s in the simulations below) (Calhoun et al., 2004; Liao et al., 2002; Worsley and Taylor, 2006). The maximum number of iterations was set to be 100; if a voxel reached the maximum number of iterations, they were referred to as non-convergent.

Thus, this approach attempts to obtain a more accurate approximation of the true model. It also upholds the $\delta > 0$ constraint. This method does not require $\text{sign}(\beta_1) = \text{sign}(\beta_2)$ since $\hat{\delta}$ may slightly over or under approximate δ , since the difference from the true δ is what the iterated ratio estimator is approximating. Thus, this method attempts to optimally estimate δ over the parameter space, while withholding the constraints desired.

4.1.6 Variance Shrinkage in the Ratio Estimator

When actually estimating δ in practice, we use a weighted ratio estimator. Liao et al. (2002) described how the ratio estimate of δ could perform badly if the voxel was not activated (i.e. a small value for T_1). They describe that a better estimate of δ is shrunk by a transformation of T_1 where T_1 is the t-statistic of the β_1 parameter. The overall estimate of δ , called $\tilde{\delta}$ is:

$$\tilde{\delta} = \frac{\hat{\delta}}{1 + \frac{1}{T_1^2}} \quad (4.7)$$

where $\hat{\delta} = \frac{\beta_2}{\beta_1}$. With $\tilde{\delta}$, one can compute the new $\hat{\delta}$ from Step 3 in the iteration and fit the model:

$$Y_i(t) = \beta_{0i} + \beta_i x(t - \delta_0) + \varepsilon_i$$

Essentially, the weighting is used so that for voxels with very small T statistics, it is assumed they are not activated and shrinks the estimate of δ to zero. This shrinkage should allow for better estimation of δ by variance shrinkage.

4.2 A Review of Testing for Activation with a Known HRF

From the above models there are different plausible testing procedures for activation after a regression is run. If an unbiased estimate of δ is obtained, then the model fit with the hrf and no derivative terms should closely approximate the true model. The standard t-tests were done with these models. For models with both the hrf and its derivative included, the “composite test” (Section 4.2.2 below for definition) and modified F-test are appropriate and were done in the simulations. The description of these tests and hypotheses are below.

4.2.1 Standard T-tests for β_1

A simple method of testing for activation is using a standard t-test for the hypothesis $\beta_1 = 0$. If the $\delta - \delta_0 \rightarrow 0$, $\beta_2 \rightarrow 0$ in Equation (4.6). Thus, this method should perform well if δ is correctly specified and the derivative terms are excluded (see Section 5.2). For models including the temporal derivative, this test neglects the lag β_2 parameter and can perform badly if δ is incorrectly specified (Calhoun et al., 2004). Thus, testing both parameters, β_1 and β_2 , may better detect activation. Recall that if a voxel is not activated, then the β parameter should be 0 and thus β_1 and β_2 should be zero.

4.2.2 T-test for both parameters

If both x and x' are included in the model then one can test both parameters β_1 and β_2 simultaneously to detect activation. The method allows testing of the composite null hypothesis: $\beta_1 = 0$ and $\beta_2 = 0$. The natural alternative hypothesis is $\beta_1 > 0$ or

$\beta_2 > 0$. Ideally, this test detects whether there is evidence of activation accounting for shifts. This will be referred to the composite test. This test compares $\max\{T_1, T_2\}$ to a critical t-statistic, T_c . If either T_1 or T_2 are greater than T_c , then the voxel is categorized as activated. This was motivated by the fact that for longer lags or activated voxels with misspecified lags, the β_2 parameter should be larger in activated voxels even if the β parameter is small.

One method for further research may be to look at this composite test, but constrain the testing to only when $\hat{\beta}_1 > 0$ and $\hat{\beta}_2 > 0$ for the SRE. This constraint would only look at cases where either the shift term or the activation term was sufficiently large, but the estimate of $\delta > 0$. For the IRE, since the estimate may converge from above or below δ , the $\hat{\beta}_2 > 0$ constraint would not be appropriate, but the $\hat{\beta}_1 > 0$ constraint could still apply.

4.2.3 Calhoun's one-sided F-statistic

Again, if both the hrf and its derivative are included in the model, one can use an F-statistic to test activation in order to take into account the β_2 coefficient. The F statistic tests whether a linear combination of the T_1 and T_2 statistics, for the corresponding β 's, is sufficiently large to categorize a voxel as activated. Worsley and Taylor (2006) discuss that the normal F-statistic $\frac{T_1^2 + T_2^2}{2}$, where T_i is the t-statistic for β_i , is insensitive to inactive voxels with negative T_1 values. A rationale of why the estimator is divided by 2 is as follows.

Let σ be the variance of the activation parameter β and let $T = \frac{\beta}{\sigma}$. In Model (4.3), we observed $\beta_1 \approx \beta$ and $\beta_2 \approx \beta\delta$. Then $T_1^2 \approx \frac{\beta^2}{\sigma^2}$. Also, $\text{Var}(\beta_2) \approx \text{Var}(\beta\delta) = \delta^2\sigma^2$, so $T_2^2 \approx \frac{\beta^2\delta^2}{\delta^2\sigma^2} = \frac{\beta^2}{\sigma^2}$. Thus, the F-statistic of $T_1^2 + T_2^2 \approx 2T^2$, and thus needs to be divided by 2 to compare to the appropriate F distribution.

Calhoun et al. (2004) proposed incorporating the sign of T_1 so that the statistic, noted as F_+ , equals $\text{sign}(T_1)\frac{T_1^2 + T_2^2}{2}$. This “one-sided” statistic should allow for a more

sensitive categorization than the normal F statistic (Worsley and Taylor, 2006). This will be referred to the one-sided F-statistic or moderated F-statistic. The statistic F_+ would be compared to a $F_{2,n-3}$ distribution where n is the number of time points for a voxel.

Again, if one correctly can identify the lag, the problem of whether to test $\beta_1 = 0$ or $\beta_1 = 0, \beta_2 = 0$ is eliminated since the derivative term is dropped from the model. If the method can approximate δ sufficiently, the testing problem becomes much simpler. Hence, estimation of δ is an important process in testing for activation.

Chapter 5

Simulations

Simulations were run to investigate the properties of the IRE, which will be referred as the iterated approach or the algorithm interchangeably. The performance of the IRE will be compared to that of the SRE, to decide whether the iterative approach improves performance. Performance will be measured using ROC curves, area under the curve, and the overall bias of the estimates. The setup of the simulations is outlined below.

Nine scenarios were chosen to test performance: 1 (short), 2 (medium or moderate), or 4 (long) second lags, each using 0.125 (small), 0.25 (medium or moderate), 0.5 (high) signal to noise ratios (SNR) defined as $\frac{\beta}{\sigma}$. The variance of the response, σ was set at 1, so a change in the SNR was a change in the β parameter used for simulation. These scenarios try to see how well the SRE and IRE estimate δ and how well the tests used can detect activation as the lag increases and signal decreases.

Each scenario used 10,000 replicates of activated and 10,000 non-activated voxels. The number of scans was chosen to be 501, similar to that of the application below, and an assumed rate (TR) of 1 scan per one second. Thus, the data represents events over 500 seconds since the first scan is time 0. Note that when using real fMRI data, the response is in TR units, so for every mention of seconds below, there is a corollary to TR units. The case here used a TR unit is 1 second to simplify the programming. The algorithm was run on a voxel-specific level to obtain estimates for β and δ for each

voxel. The inter-stimulus time used was 30 seconds, to allow for sufficient temporal resolution. The non-activated voxels were pooled from each scenario to give a more accurate assessment of the bias of parameters and convergence rates for non-activated voxels.

Linear models as noted above were fit using least squares. The code used for the simulations is in Section 10.1.2 of the Appendix. The maximum number of iterations was set at 100 and the tolerance was 10^{-8} as stated above. The overall goal was to see whether, and in which scenarios, the IRE outperforms the SRE in bias and which testing procedure is optimal. Overall, the composite test and standard t-test for the IRE perform better than the SRE for the longer lags. Also, the IRE has better estimation of δ than the SRE for longer lags, but both do poorly if the SNR is too small.

5.1 Convergence

Recall that a voxel is said to converge if $\hat{\delta}_\Delta$, or the chance from one estimate of δ to the next is less than 10^{-8} . This convergence is important since the estimate of δ has stabilized around some value, ideally the true value. The convergence results for the activated voxels are in Tables 5.1, 5.2, and 5.3 below for each SNR, where NC denotes non-converged voxels.

As the SNR decreases in any of the lags, the rate of convergence decreases. The percentage of those zeroing out increases, as would be expected since the data has larger variation and a less pronounced form. Holding SNR constant, moderate lag times have the most convergence. This relationship makes sense because the estimates in the first iteration for the moderate lag time should be larger than in the first iteration of the short lag time. Thus, the estimates are less apt to zero out for moderate lags compared to short lags, as seen in the tables below. Also, the longer lag time may not have as accurate estimates of the first iteration since the first iteration

has the Taylor expansion around zero and may not be accurate for the longer lag time.

	Converged % (n)		
Lag	1	2	4
SNR	0.5	0.5	0.5
Converge	89 (8919)	97 (9661)	83 (8322)
Zeroed	9.7 (969)	2 (202)	15 (1546)
Maxed	0.0 (1)	0.1 (9)	0.5 (51)
NC	1.1 (111)	1.3 (128)	0.8 (81)

Table 5.1: Convergence Rates for Active Voxels, High SNR

	Converged % (n)		
Lag	1	2	4
SNR	0.25	0.25	0.25
Converge	68 (6760)	75 (7475)	61 (6083)
Zeroed	29 (2882)	21 (2078)	35 (3481)
Maxed	0.5 (53)	0.7 (69)	1.9 (191)
NC	3 (305)	3.8 (378)	2.4 (245))

Table 5.2: Convergence Rates for Active Voxels, Medium SNR

	Converged % (n)		
Lag	1	2	4
SNR	0.125	0.125	0.125
Converge	53 (5347)	55 (5546)	50 (5006)
Zeroed	42 (4199)	39 (3907)	45 (4518)
Maxed	1.2 (115)	1.1 (107)	1.5 (146)
NC	3.4 (339)	4.4 (440)	3.3 (330)

Table 5.3: Convergence Rates for Active Voxels, Low SNR

The average number of iterations for convergence for activated voxels are given in Table 5.4. The smaller the SNR, the more iterations the algorithm needs to converge on average, as expected. The number of iterations for a zeroed out case is 2 iterations for activated voxels. All non-converged voxels reached the limit of 100 iterations. The average number of iterations needed to max out was around 18 iterations, which was consistent across SNR for activated voxels. Thus, the algorithm does not need an unreasonable amount of iterations to converge.

Lag	SNR	Average Iterations to Convergence
1	0.5	18
2	0.5	18
4	0.5	19
1	0.25	22
2	0.25	22
4	0.25	23
1	0.125	24
2	0.125	25
4	0.125	25

Table 5.4: Average Number of Iterations for Converged Active Voxels

Table 5.5 shows the convergence rates for the non-activated voxels. The average number of iterations for convergence for non-activated voxels was 26 iterations. This is important since a large portion of typical fMRI data has non-activated voxels. Thus, the computation time for non-activated voxels is reasonable for this method if each iteration is short. As we will see in the computation time in the method application (Section 6.3), the analysis time is reasonable. The tables above and Table 5.5 give more information when looking at them in conjunction with the tables of bias in the next section. For example, the convergent voxels are relatively unbiased for activated voxels, and a larger percentage will converge with a higher SNR.

Result	% (n)
Converge	46 (41105)
Zeroed	49 (44507)
Maxed	0.04 (36)
NC	4.8 (4352)

Table 5.5: Convergence Rates for Non-active Voxels

5.2 Performance Diagnostics

To detect how well each method detected activation, each level of SNR and lag time compared the standard t-test, composite test, and one-sided F-tests for the IRE and SRE models. The models are shown below for clarification. Let $\hat{\delta}_0$ be the estimate from the first iteration and $\hat{\delta}_c$ be the estimate from the last iteration, regardless of whether the voxel converged. The “Derivative IRE” is the model of the last iteration

for the IRE, regardless of convergence, with the regressors of $x(t - \hat{\delta})$ and $x'(t - \hat{\delta})$, where $\hat{\delta}$ is from the last iteration. The “Derivative SRE” uses the initial model with regressors $x(t)$ and $x'(t)$.

Derivative IRE

$$Y(t) = \beta_0 + \beta_1 x(t - \hat{\delta}_c) - \beta_2 x'(t - \hat{\delta}_c) + \varepsilon$$

Derivative SRE

$$Y(t) = \beta_0 + \beta_1 x(t) - \beta_2 x'(t) + \varepsilon$$

Each derivative model uses the moderated F-statistic to obtain p-values for β coefficients for the tables. The degrees of freedom were the number of time points (N) subtracted by the number of predictors (including an intercept) (p). The F-statistics for the derivative models had degrees of freedom $v_1 = 2$, and $v_2 = N - p = 501 - 3 = 498$.

The “No Derivative” models use $\hat{\delta}$ from the iteration before, either the converged estimate or last iteration, with the regressor $x(t - \hat{\delta})$ in the model. These models should approximate the true model well if $\hat{\delta} \approx \delta$.

No Derivative IRE

$$Y(t) = \beta_0 + \beta_1 x(t - \hat{\delta}_c) + \varepsilon$$

No Derivative SRE

$$Y(t) = \beta_{0i} + \beta_1 x(t - \hat{\delta}_0) + \varepsilon$$

The non-derivative models use standard t-tests for β_1 , with degrees of freedom, v , $501-2 = 499$ to obtain p-values. The true model, noted by “True” in the tables, had regressor $x(t - \delta)$, and used t-tests with 499 degrees of freedom for β_1 . Type I error, α , or false positive rates will be defined as categorizing a voxel as activated given the voxel is truly non-activated. Power will be defined as defined as categorizing a voxel

as activated given the voxel is truly activated. ROC curves and AUC will be used as measures of performance.

The receiver operating characteristic (ROC) curve allows the comparison of the two methods with varying type I error rates. The optimum ROC curve has 100% power and 0% false positive rate. In comparing ROC curves then, a better ROC curve has high power at low type I error rates. The curves allow a comparison of power for a set α for different tests.

There are multiple curves plotted for each ROC curve, corresponding to different tests. The black line marks the true model. The red solid line is the t-test for the No Derivative IRE model and the dotted red line is the t-test on the subset of voxels in this model that converged. The orange line is the t-test for the No Derivative SRE. The blue lines use the Derivative SRE models, with the solid line representing the moderated F-statistic and the dotted line representing the composite test. The green lines use the Derivative IRE models and use the same statistics respective to the Derivative SRE models. The comparisons should be made between the No Derivative SRE and IRE models and in reference to the true model. The ROC for the true model (black line) shows how well estimator could perform if δ was specified exactly correct. The “t’s” denotes the composite test was done.

The area under the curve (AUC) is another diagnostic used to measure how well a statistic operates. The “curve” in AUC refers to the ROC curve, so a higher AUC notes better performance for a statistic.

5.3 Estimation of δ and Performance of Scenarios

Overall, the results are the same as when analyzing ROC curves or the AUC: the IRE tests outperform the SRE tests as lag increases and the composite test does the best when lag is long. Also, the bias of $\hat{\delta}$ for the SRE is small when the lag is short, but gets larger as the lag increases. The bias of $\hat{\delta}$ for the IRE is smaller than that

of the SRE as lag increases. The overall bias of $\hat{\beta}_1$ is smaller as lag increases for the IRE, but must be looked at with respect to convergent and non-convergent voxels. The sections are broken into the high, medium, and low SNR groups and lags across these SNRs are compared.

5.3.1 High SNR

The data in this section are based on a “high” SNR, which is defined as 0.5. The three levels of lag, short, medium and long (1 s, 2 s and 4 s, respectively) are analyzed and are discussed below. Tables 5.6, 5.7, and 5.8 show the tables for the area under the curve (AUC) for lags 1, 2, and 4s. The ROC curves show that each method has relatively high power for a small false positive rate. The composite test does well for a lag of 4 seconds and comparatively well for the other lags as seen in the AUC tables. The SRE does slightly better than the IRE models for the short lag, but the IRE does slightly better for the moderate and long lag settings. The standard t-tests for the non-derivative models do slightly better than the F-tests for the derivative models, but does comparably well or worse to the composite tests.

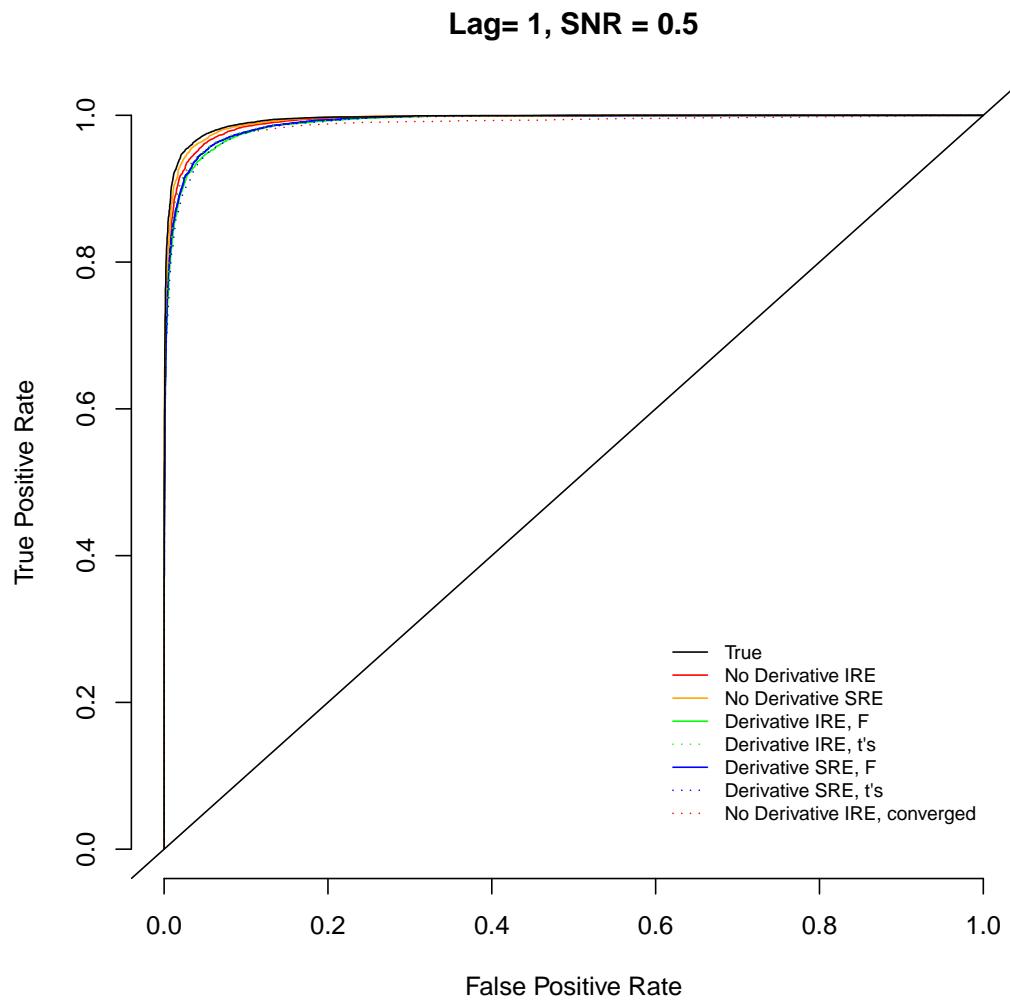


Figure 5.1: ROC for Short Lag, High SNR

Model	AUC
TRUE	0.995
No Derivative IRE	0.992
No Derivative SRE	0.994
No Derivative IRE, converged	0.987
Derivative SRE F	0.991
Derivative IRE F	0.990
Derivative SRE t's	0.989
Derivative IRE t's	0.991

Table 5.6: AUC for Lag = 1, SNR = 0.5

Lag= 2, SNR = 0.5

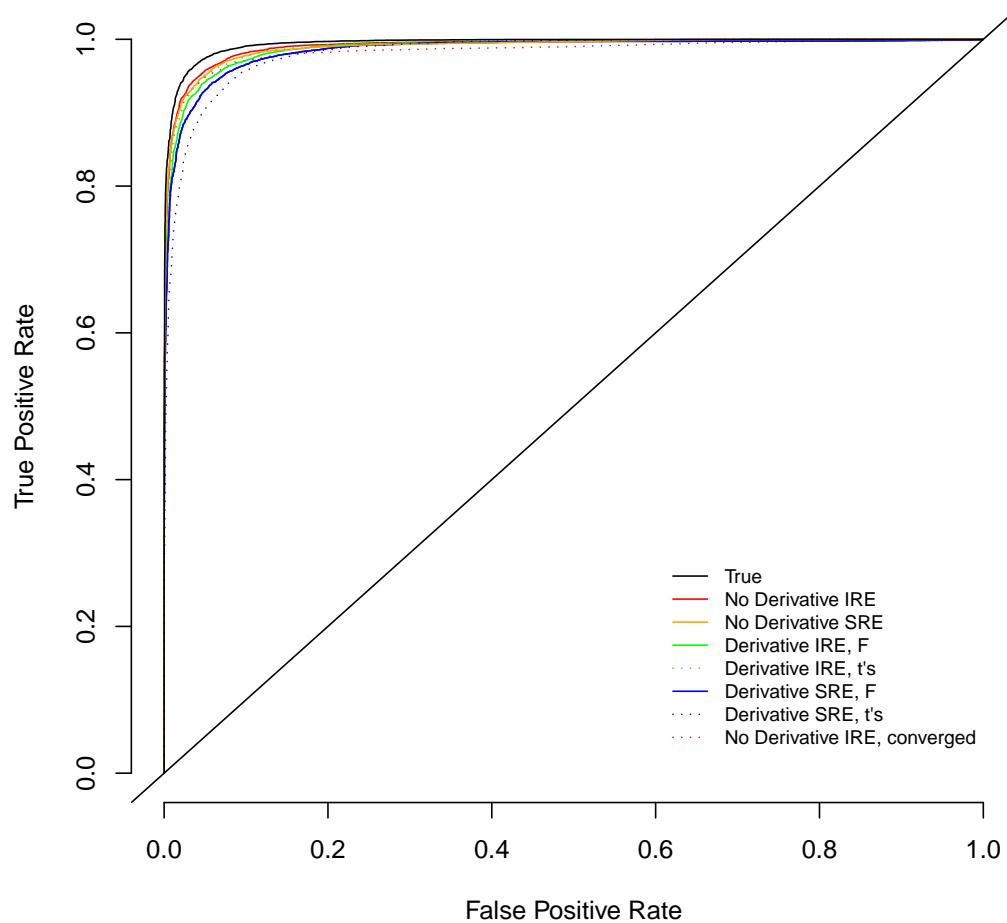


Figure 5.2: ROC for Medium Lag, High SNR

Model	AUC
TRUE	0.995
No Derivative IRE	0.991
No Derivative SRE	0.989
No Derivative IRE, converged	0.985
Derivative SRE F	0.985
Derivative IRE F	0.987
Derivative SRE t's	0.982
Derivative IRE t's	0.990

Table 5.7: AUC for Lag = 2, SNR = 0.5

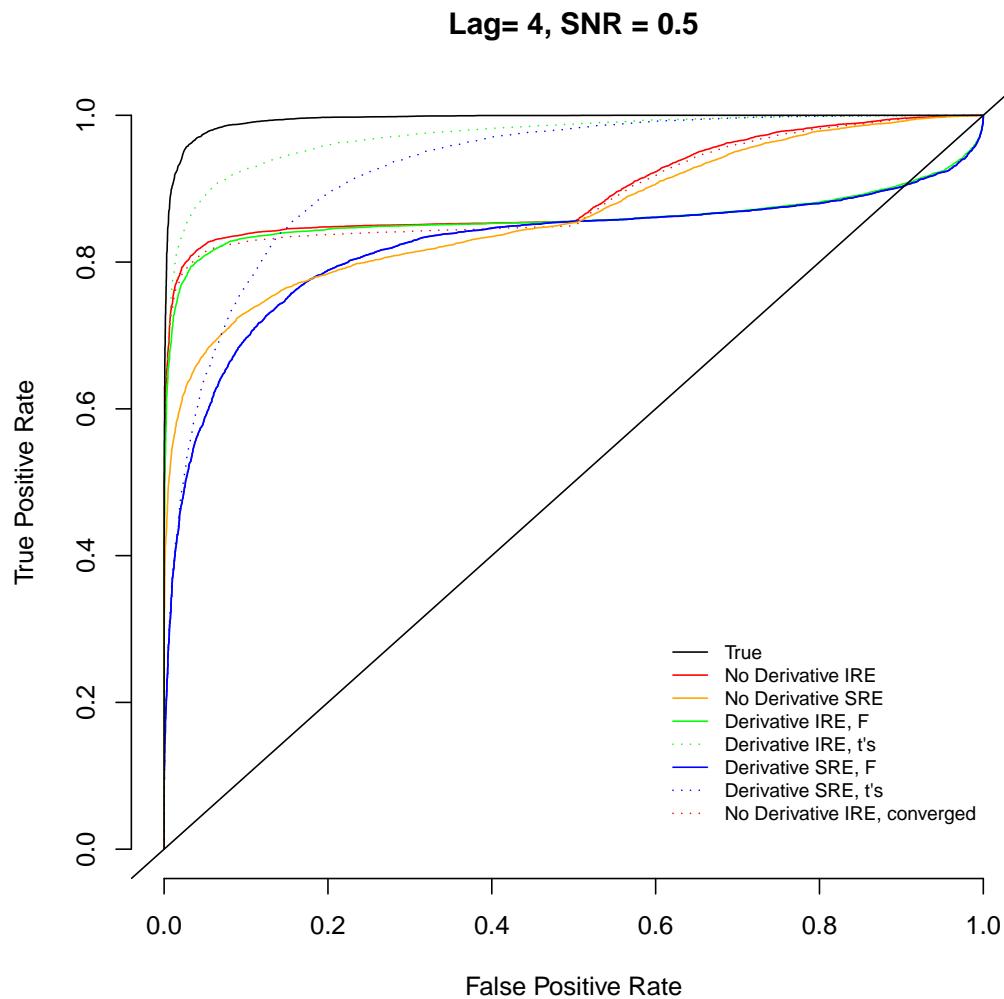


Figure 5.3: ROC for Long Lag, High SNR

Model	AUC
TRUE	0.995
No Derivative IRE	0.901
No Derivative SRE	0.864
No Derivative IRE, converged	0.894
Derivative SRE F	0.822
Derivative IRE F	0.860
Derivative SRE t's	0.927
Derivative IRE t's	0.972

Table 5.8: AUC for Lag = 4, SNR = 0.5

Below are box plots where the point in the middle is the median, the whiskers are the lines shown and box for the interquartile range is the interior distance between

lines around the median point. One desirable property of an estimator is low bias, which is the difference of the estimator from the true value. The estimates of bias for β_1 and δ are shown in Figures 5.4 and 5.5, respectively. If the focus of analysis is estimation of δ , one would want the most unbiased estimator of δ . Along with low bias, a good estimator will not have large variance. The spread of the data can be gauged by how far the interquartile range is on the box plots. Thus, a good estimator will have a bias close to 0 and a small box around the median.

The converged estimates of the IRE have consistently positively biased values for β_1 , which may be accounted for since an estimate of δ was used, which has variability. The estimates of β_1 are negatively biased for the non-converged estimates. The zeroed out estimates are increasingly negatively biased and the maxed out estimates are decreasing negatively biased. This is expected since the true δ is getting further away 0s in the zero case and getting closer $\Delta_{max}=6s$ in the maxed out case as the lag increases and the model is misspecified. One thing to note is that the converged IRE estimates are relatively homogeneous.

Bias for Beta 1, SNR = 0.5

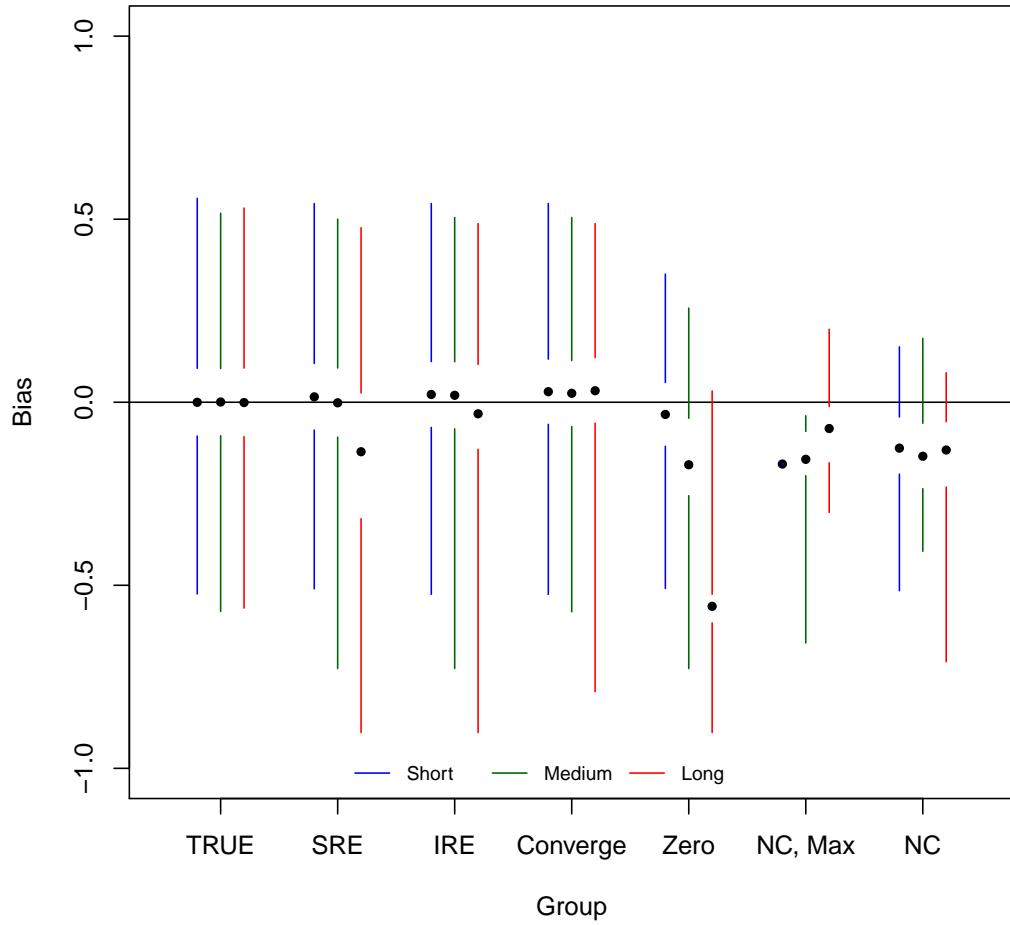


Figure 5.4: Box Plot for Bias of β_1 for High SNR

The true model, the zeroed out and maxed out estimates were included in Figure 5.5 for constancy across figures. This figure gives evidence that taking averages voxels with a common lag time should give a relatively accurate estimate of δ when the ire converges. As the lag increases, the estimate of δ for the SRE gets more biased, whereas the bias for the converged IRE does not have large differences in bias.

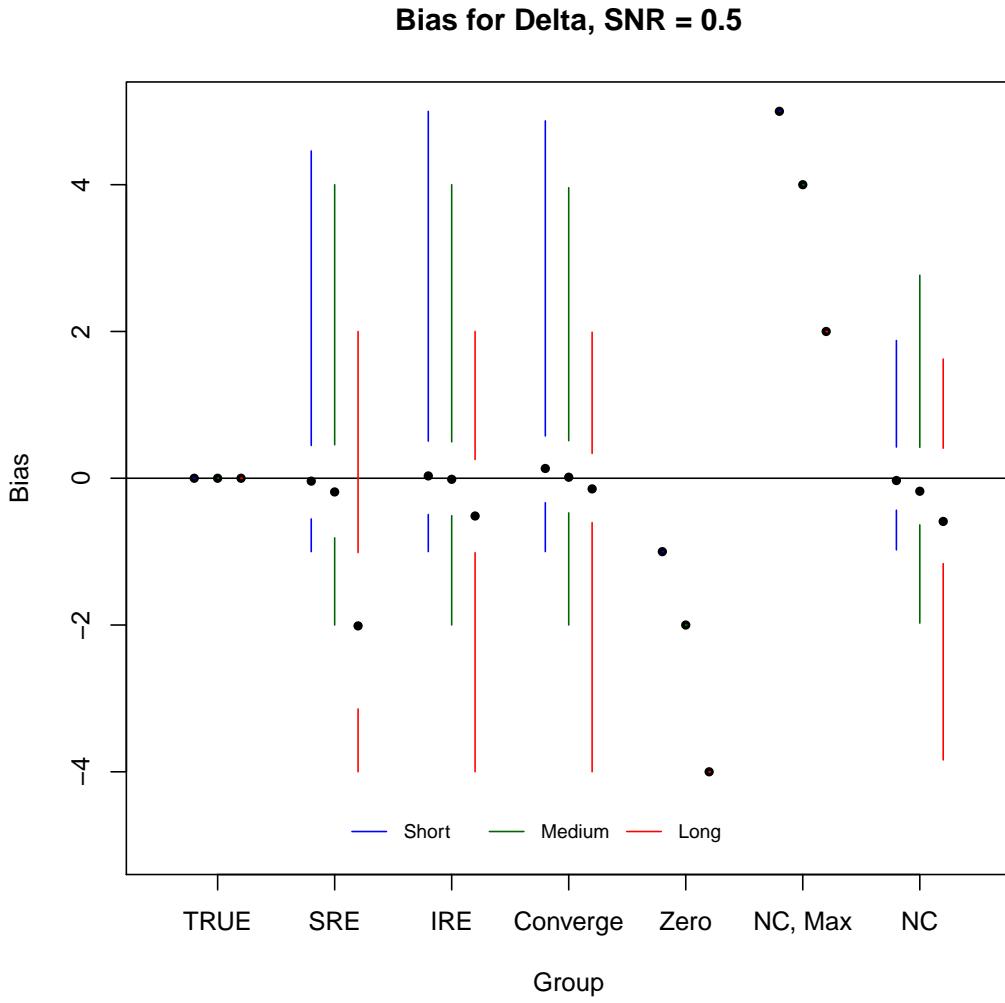


Figure 5.5: Box Plot for Bias of δ for High SNR

5.3.2 Medium SNR

A similar analysis of differing lag times is below with an SNR of 0.25. One thing to note is that the performance of the SRE again decreases as lag increases and that the composite test does as well, if not better than, the single t-test for the no derivative IRE model. One reasonable explanation is that the composite test is detecting the activated voxels that may have zeroed or maxed out. The composite test seems to outperform the other tests, specifically the one-sided F-statistic, as the lag increases regardless of SNR.

Model	AUC
TRUE	0.902
No Derivative IRE	0.892
No Derivative SRE	0.895
No Derivative IRE, converged	0.879
Derivative SRE F	0.875
Derivative IRE F	0.873
Derivative SRE t's	0.865
Derivative IRE t's	0.869

Table 5.9: AUC for Lag = 1, SNR = 0.25

Model	AUC
TRUE	0.896
No Derivative IRE	0.868
No Derivative SRE	0.859
No Derivative IRE, converged	0.854
Derivative SRE F	0.833
Derivative IRE F	0.839
Derivative SRE t's	0.853
Derivative IRE t's	0.868

Table 5.10: AUC for Lag = 2, SNR = 0.25

Model	AUC
TRUE	0.900
No Derivative IRE	0.740
No Derivative SRE	0.693
No Derivative IRE, converged	0.725
Derivative SRE F	0.652
Derivative IRE F	0.692
Derivative SRE t's	0.765
Derivative IRE t's	0.811

Table 5.11: AUC for Lag = 4, SNR = 0.25

The ROC curves show similar information as the AUC tables in terms of performance, but have interesting patterns. As the false positive rate reaches 50%, the figures start to level out and then increase again for certain statistics below. This may be an artifact due to using half activated and half non-activated voxels in the simulation.

Lag= 1, SNR = 0.25

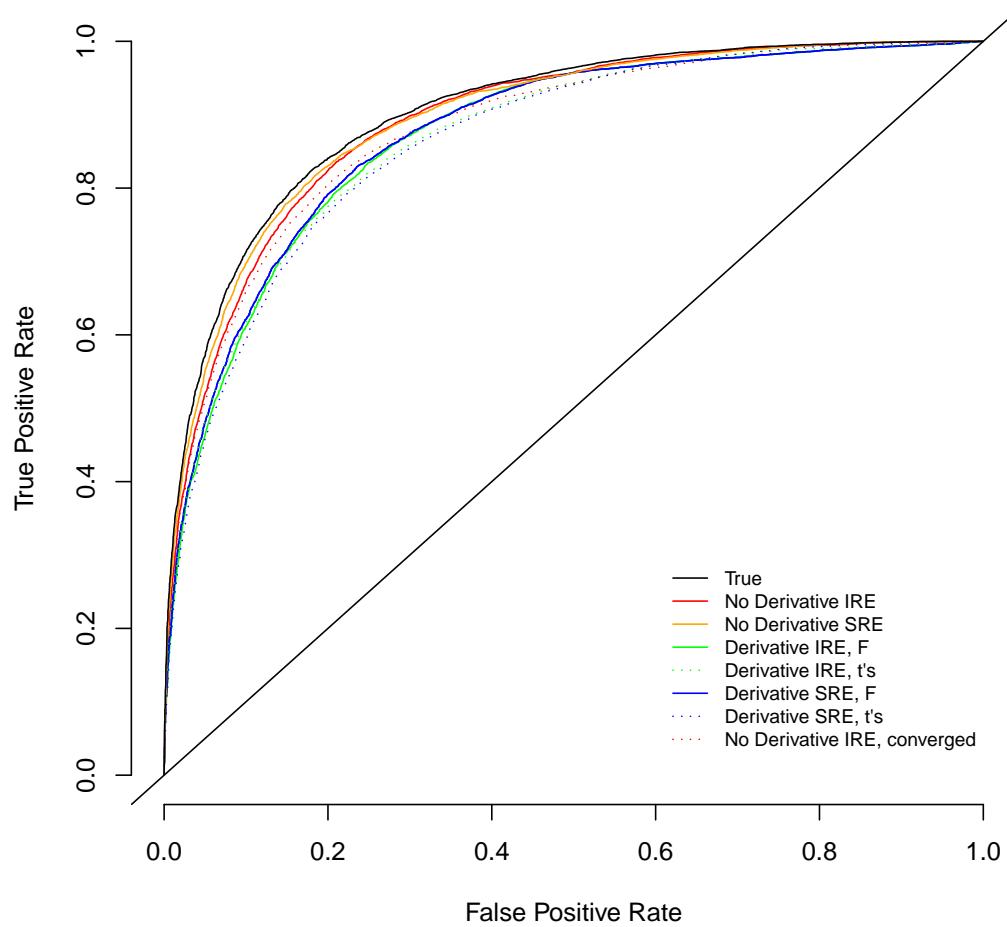


Figure 5.6: ROC for Short Lag, Medium SNR

Lag= 2, SNR = 0.25

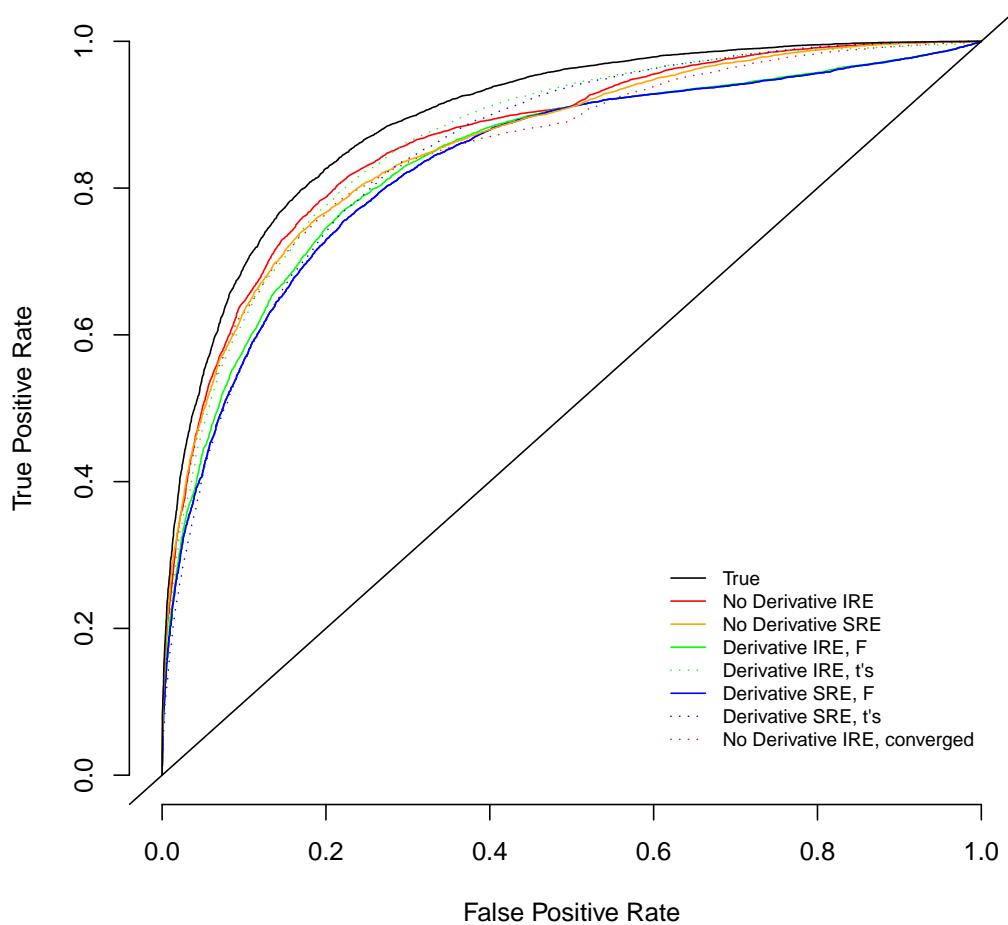


Figure 5.7: ROC for Medium Lag, Medium SNR

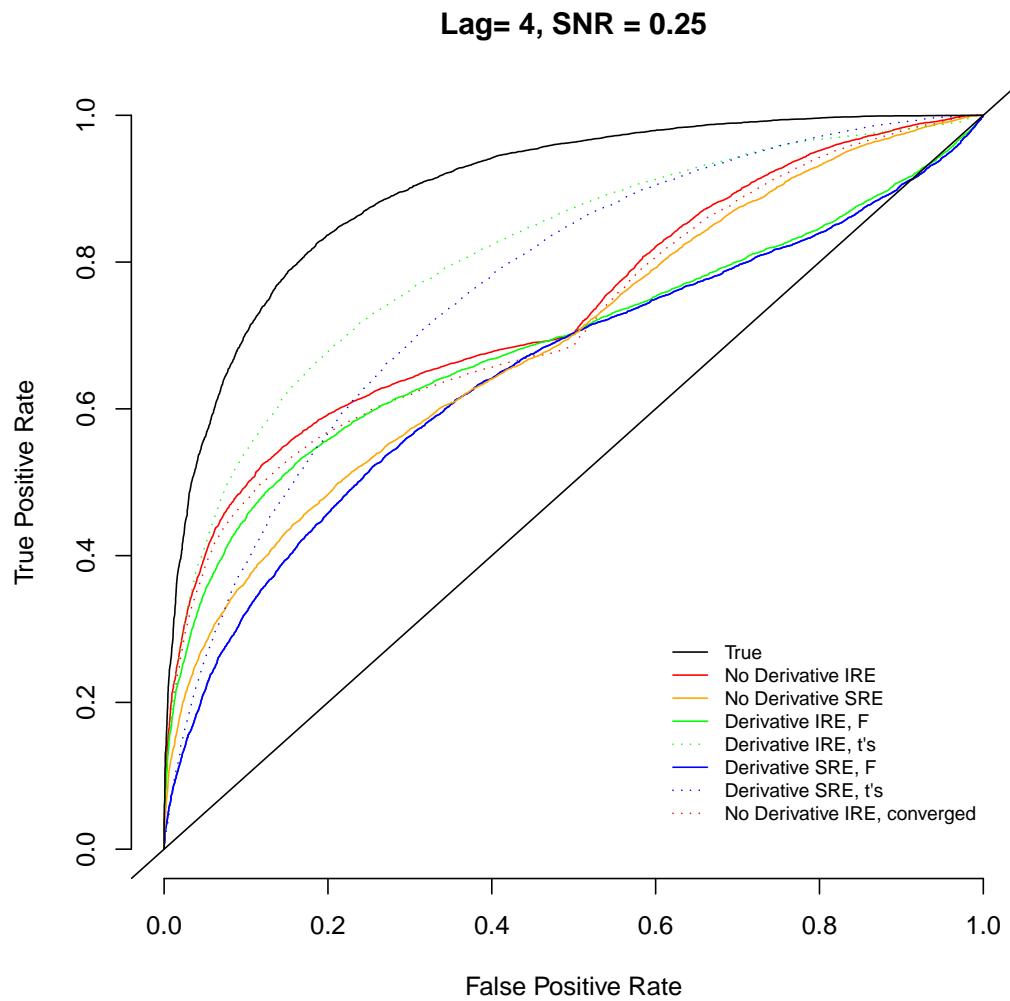


Figure 5.8: ROC for Long Lag, Medium SNR

Again, the estimates of β_1 have a similar pattern as with high SNR with respect to bias. Also, as expected, the estimates of δ become more biased and variable as the SNR decreases.

Bias for Beta 1, SNR = 0.25

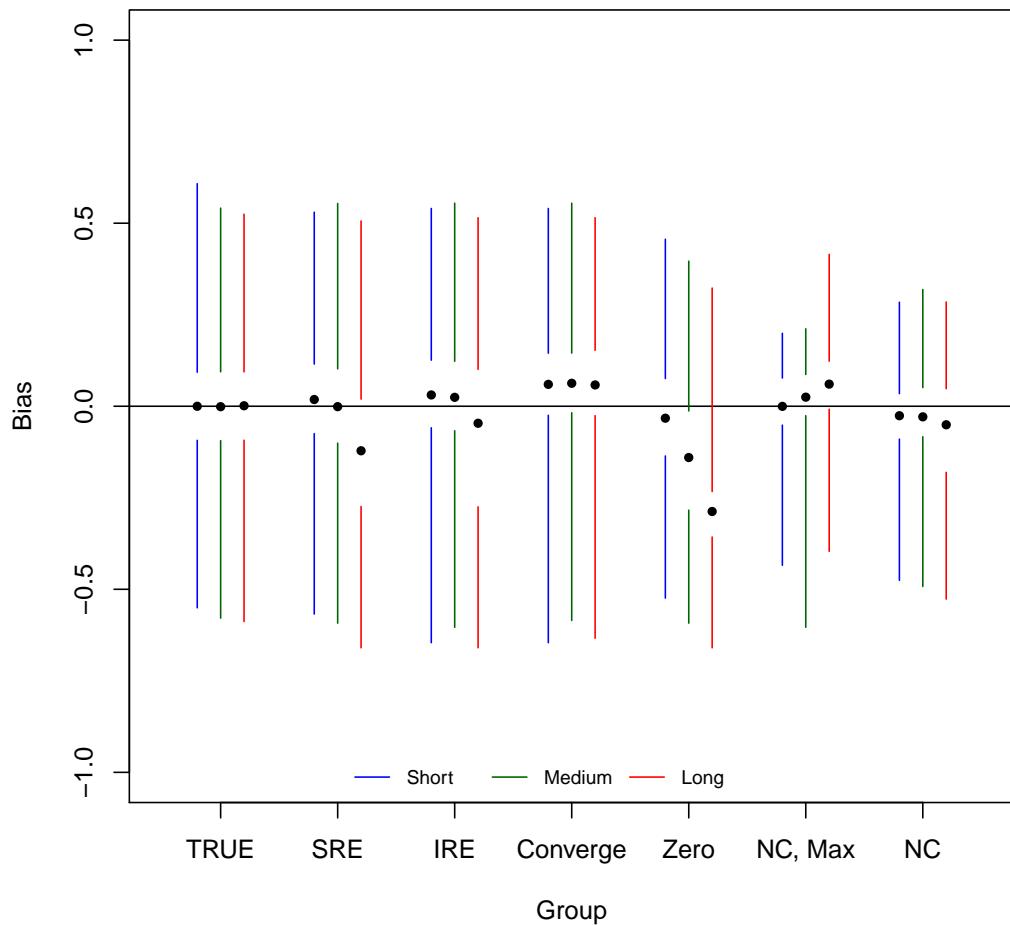


Figure 5.9: Box Plot for Bias of β_1 for Medium SNR

Bias for Delta, SNR = 0.25

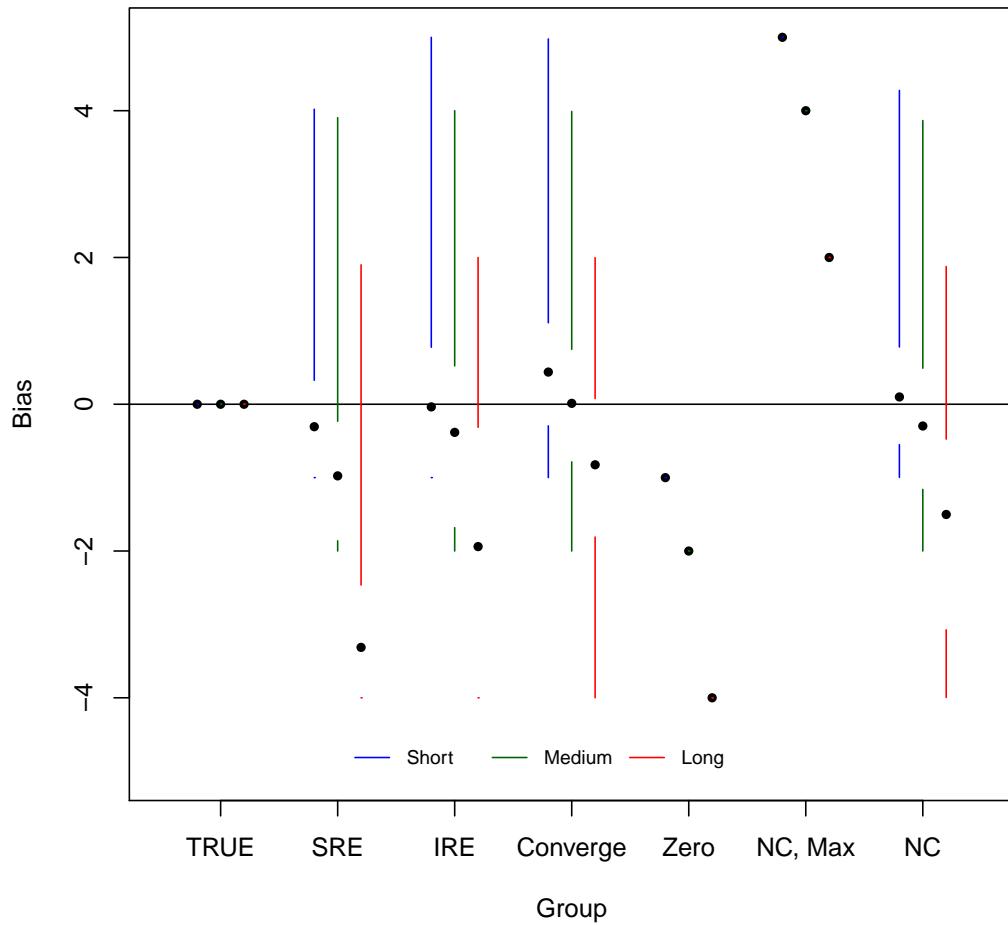


Figure 5.10: Box Plot for Bias of δ for Medium SNR

5.3.3 Low SNR

The results for the low SNR of 0.125 are similar to those above. The SRE model still has best performance when the lag is short (albeit slightly). The no derivative IRE model t-test has good performance for longer lags and the derivative IRE model composite test seems to have better performance for longer lags. Overall, the IRE models perform better compared to the SRE models for voxels with longer lag times.

Model	AUC
TRUE	0.738
No Derivative IRE	0.729
No Derivative SRE	0.730
No Derivative IRE, converged	0.721
Derivative SRE F	0.708
Derivative IRE F	0.708
Derivative SRE t's	0.703
Derivative IRE t's	0.704

Table 5.12: AUC for Lag = 1, SNR = 0.125

Model	AUC
TRUE	0.738
No Derivative IRE	0.710
No Derivative SRE	0.700
No Derivative IRE, converged	0.697
Derivative SRE F	0.678
Derivative IRE F	0.681
Derivative SRE t's	0.701
Derivative IRE t's	0.706

Table 5.13: AUC for Lag = 2, SNR = 0.125

Model	AUC
TRUE	0.733
No Derivative IRE	0.615
No Derivative SRE	0.587
No Derivative IRE, converged	0.607
Derivative SRE F	0.567
Derivative IRE F	0.586
Derivative SRE t's	0.635
Derivative IRE t's	0.655

Table 5.14: AUC for Lag = 4, SNR = 0.125

The differences in ROC curves for the long lag (Figure 5.15), have similar patterns as the ROC curves above. The same pattern in the medium SNR was observed at 50% false positive rate.

Lag= 1, SNR = 0.125

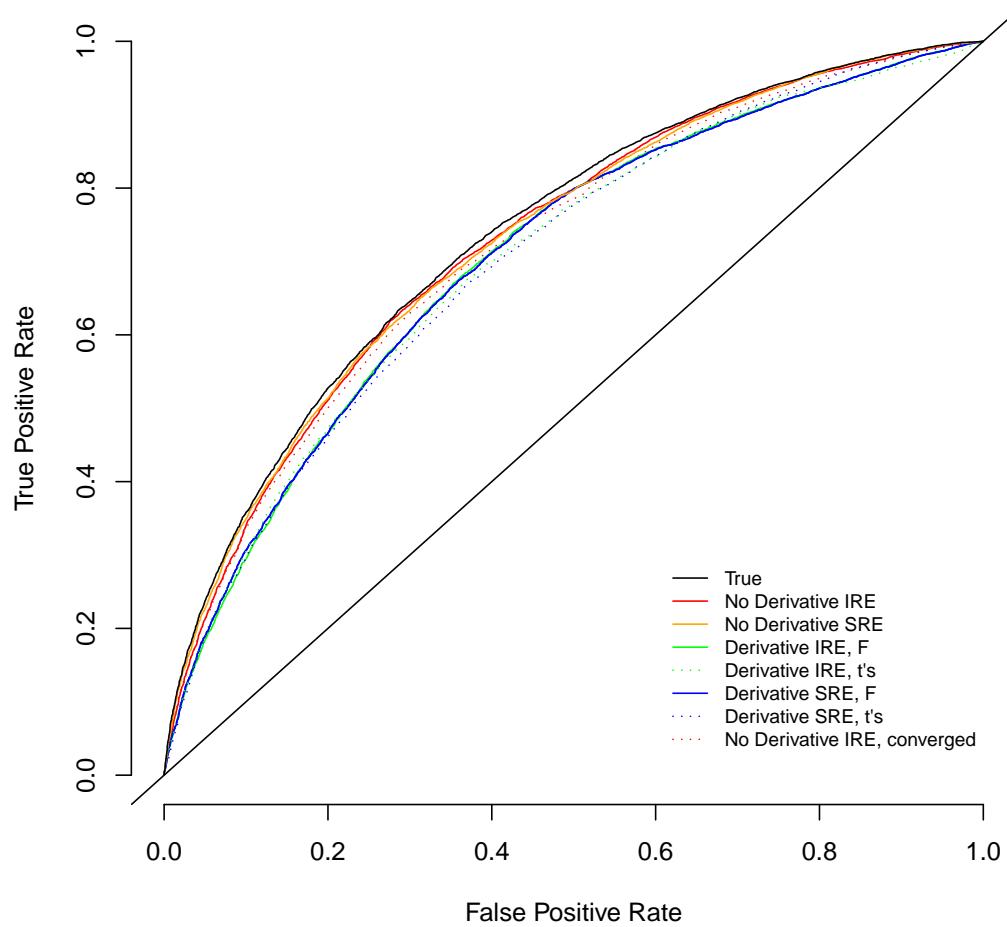


Figure 5.11: ROC for Short Lag, Low SNR

Lag= 2, SNR = 0.125

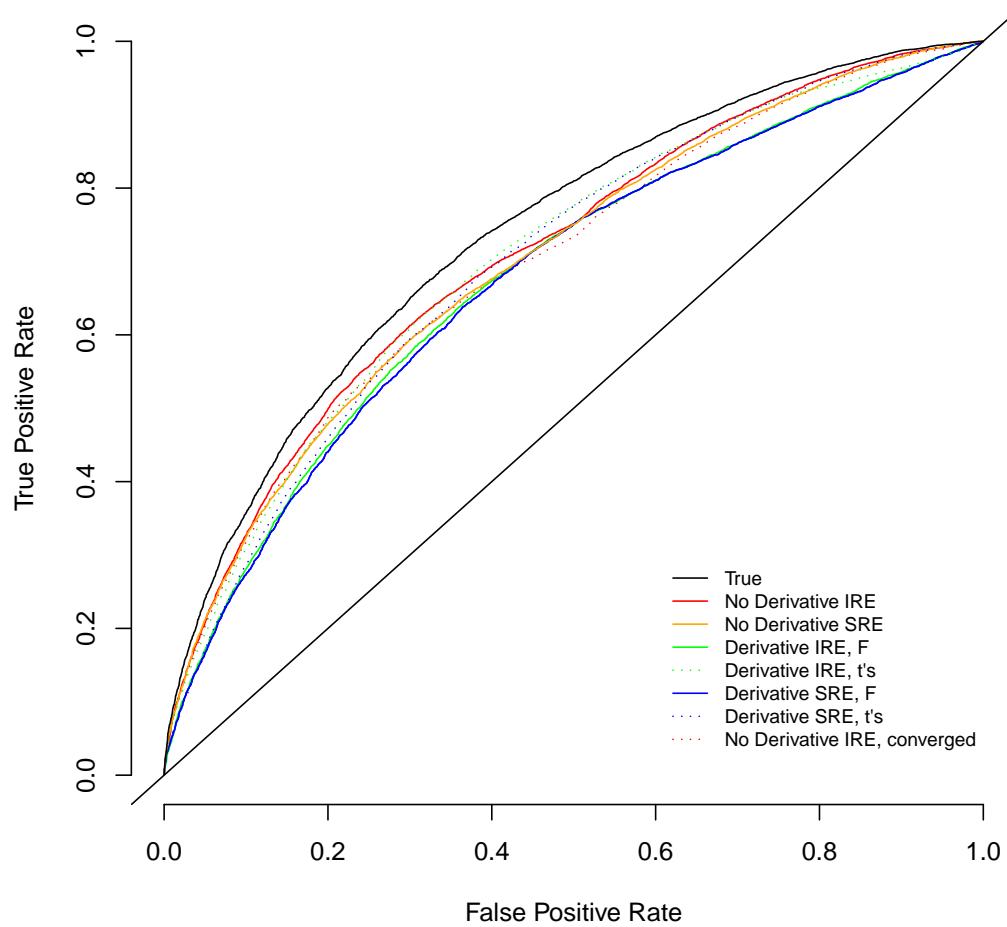


Figure 5.12: ROC for Medium Lag, Low SNR

Lag= 4, SNR = 0.125

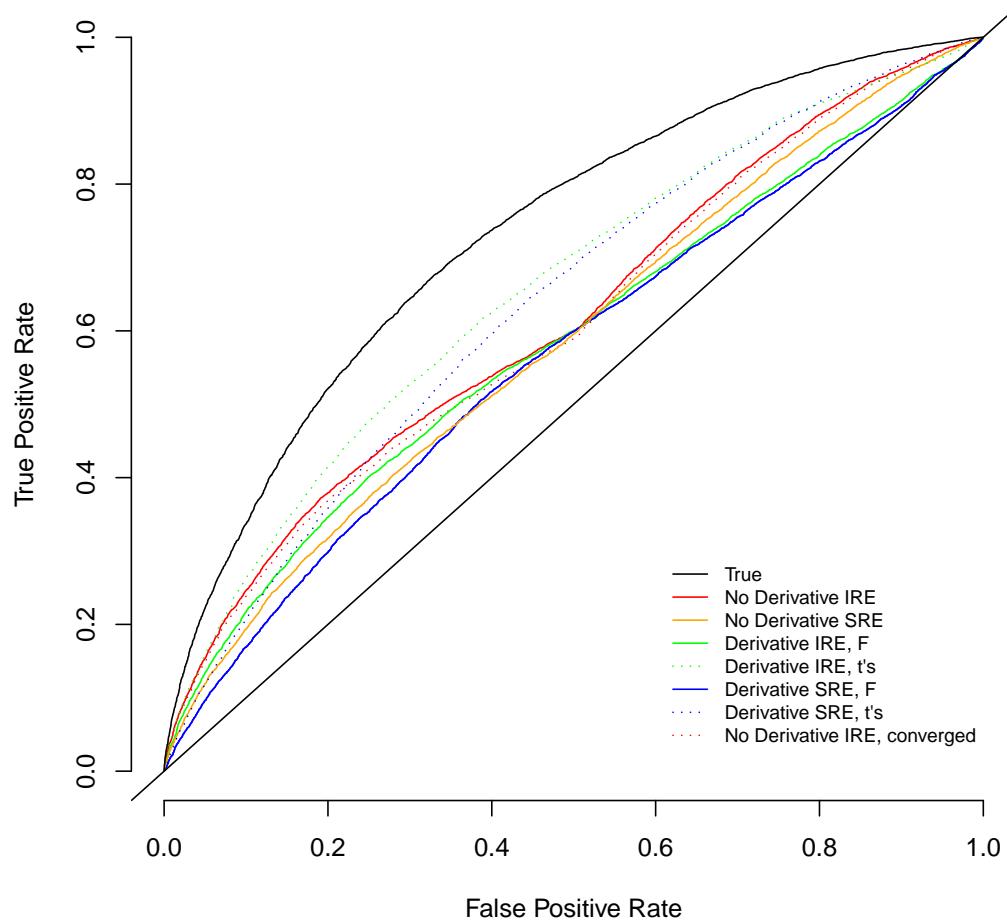


Figure 5.13: ROC for Long Lag, Low SNR

The bias of β_1 does not change greatly from that of the medium SNR. The bias of δ varies much more greatly here and is worse than the other SNR. Figure 5.15 displays that with a small enough signal to noise ratio, accurately estimating δ is difficult. As noted that this method may work for other parameters of the hrf, this should apply to those parameters as well.

Bias for Beta 1, SNR = 0.125

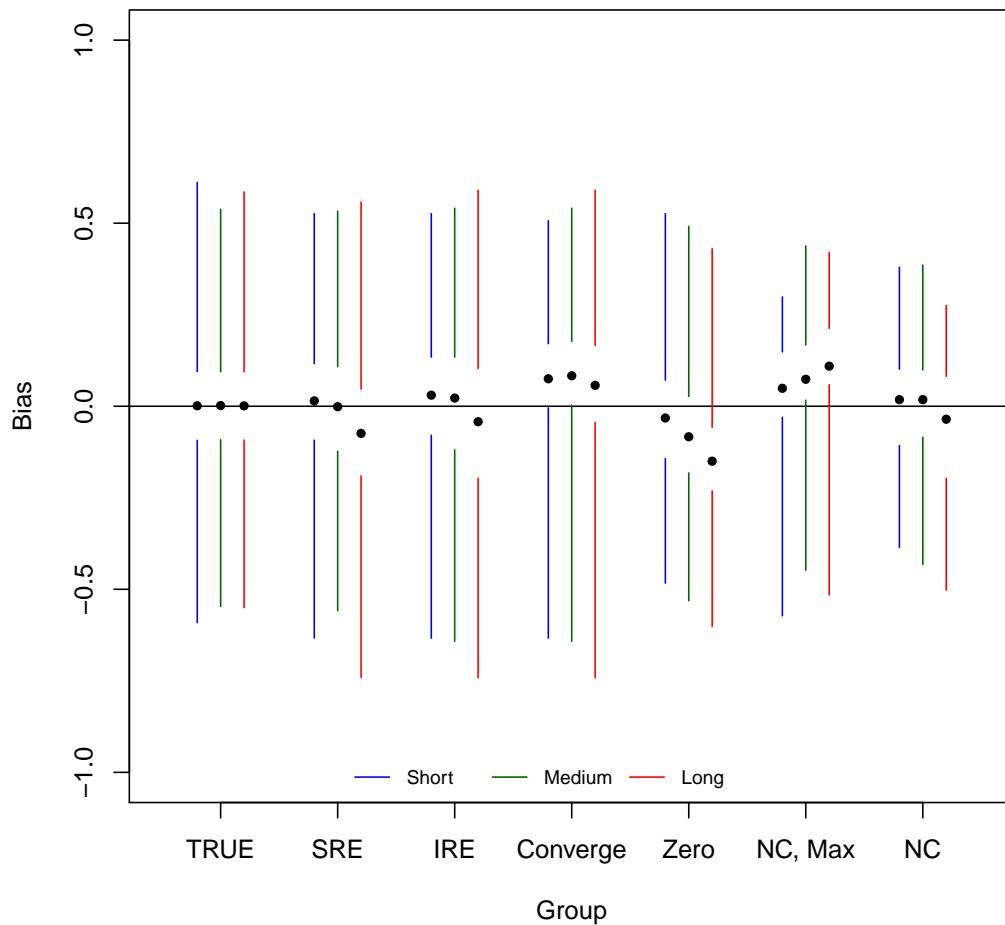


Figure 5.14: Box Plot for Bias of β_1 for Low SNR

Bias for Delta, SNR = 0.125

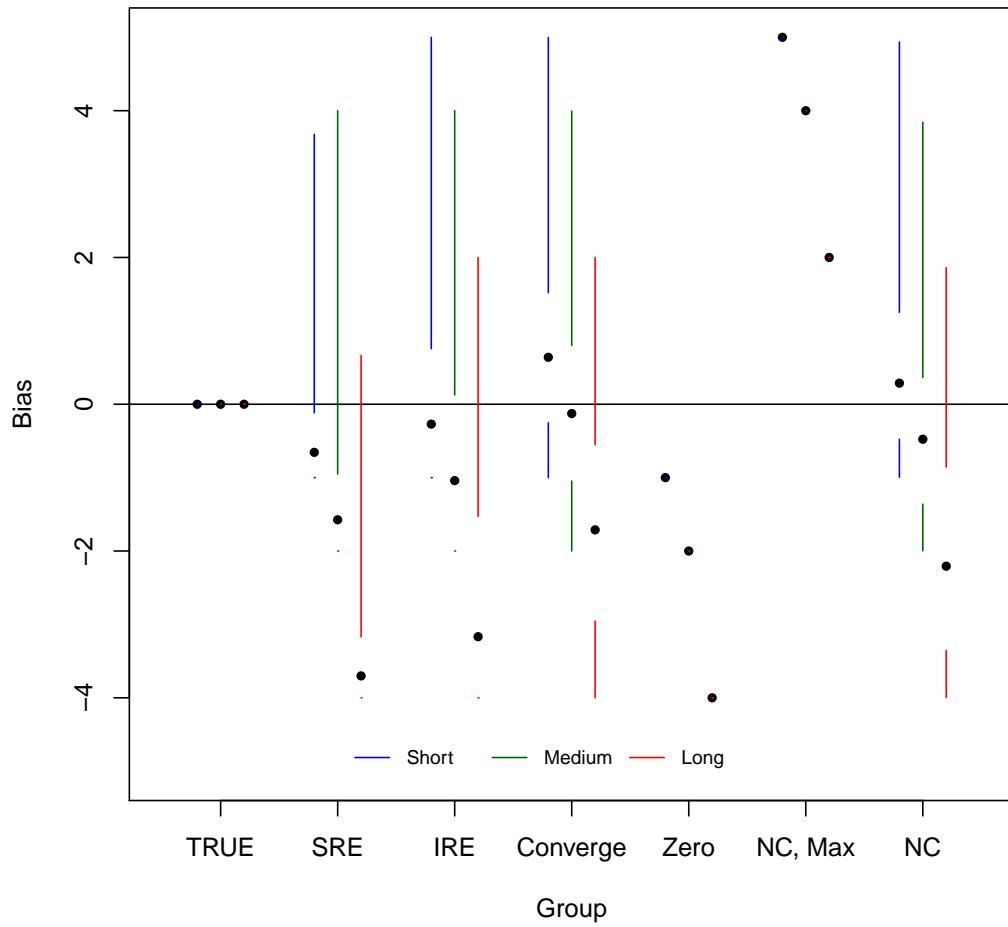


Figure 5.15: Box Plot for Bias of δ for Low SNR

5.3.4 Non-activated Voxels

Recall a non-activated voxel has no hemodynamic response, and can be thought of having a SNR of 0. Figure 5.16 show box plots for non-activated voxels and that the median bias is approximately zero for the methods. The converged voxels for the IRE are slightly positively biased whereas the maxed out voxels are slightly negatively biased. These results show that on average, each method should perform well on non-activated voxels. Since a large portion of fMRI data are non-activated voxels, it shows that these methods do well on average.

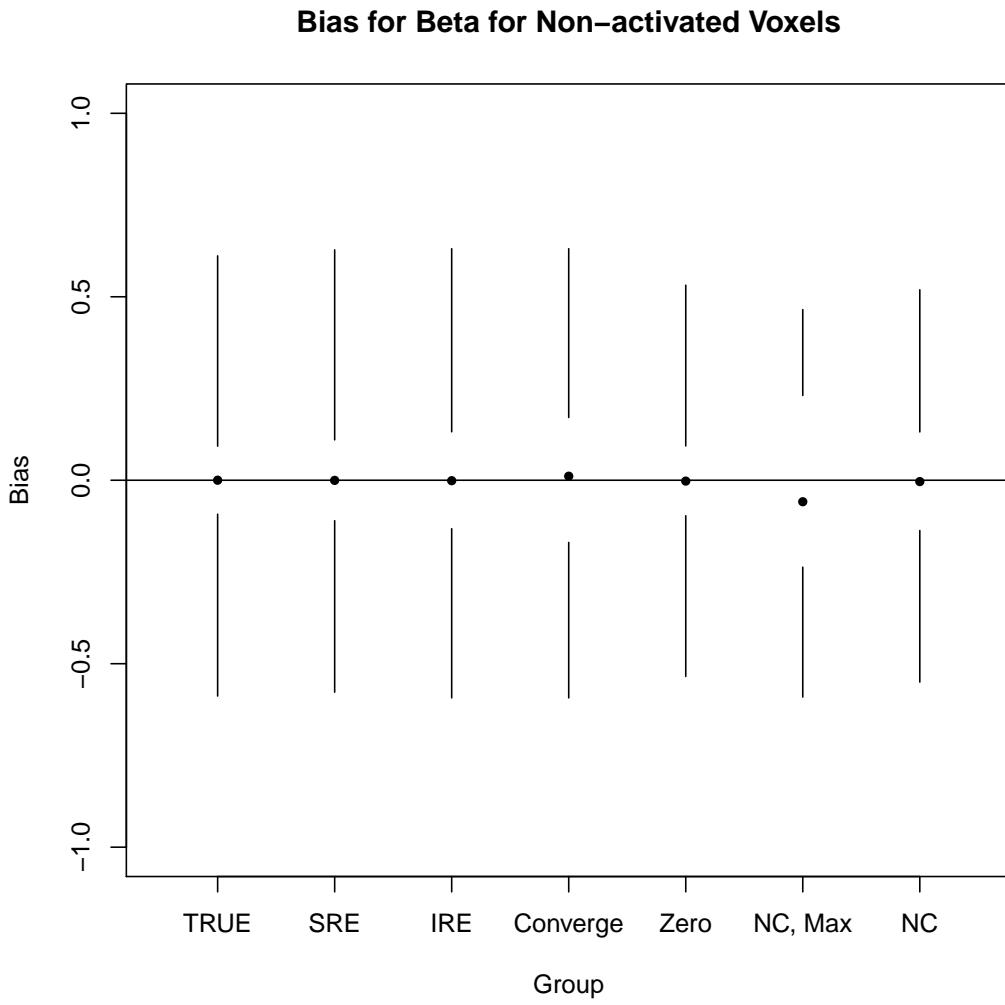


Figure 5.16: Box Plot for Bias of β_1 for Non-Activated Voxels

5.3.5 Conclusion

Overall, the SRE model performed better for shorter lags and the IRE model for longer lags. The IRE model t-test and IRE model composite test were comparable for medium lags and the IRE model composite test performed best when the lag was long. Thus, the composite test should be an area of future research in fMRI detection.

When looking at the bias of β_1 , the IRE is positively biased in converged estimates. Estimates that zeroed out in the IRE had consistently negatively biased estimates, which would affect the power of the method. The overall median bias was low for

each method, but the SRE had less bias for β_1 for the short and moderate lag times. Each method was relatively unbiased for β_1 in non-activated voxels.

Estimation for δ was much more variable. For high SNR, the converged and non-converged estimates had small median bias for the IRE. The SRE had small median bias for short lags, but had increasingly more bias as the lag was longer. The bias of each method gets worse as the SNR decreases, but the converged and non-converged estimates of the IRE are consistently less biased than that of the SRE.

Using the shrinkage estimator for δ_Δ was investigated to see its influence on the algorithm and its use in the estimation of β_1 . The method was run using $\hat{\delta} = \hat{\delta}_\Delta + \delta_0$ (see step 3 of iteration) rather than using $\tilde{\delta}_\Delta + \delta_0$, which resulted in poorer estimates of δ and β_1 . Therefore, $\tilde{\delta}_\Delta$ was included in the algorithm.

In conclusion, the results indicate that the SRE works well in estimating δ and β_1 with shorter lag times. The IRE has better performance in estimating δ when the method converges, and even moderately well when it stays within the parameter space. This is to be expected since the Taylor series approximation should be a better approximation as the lag diverges from zero. The performance of the composite test is high when the lags are longer and the IRE models outperform the SRE models as the lag increases.

5.3.6 When to use the IRE or SRE

The method of the IRE works as a diagnostic for δ if the voxel is activated and the iteration converges as was shown above. Therefore, if estimation of δ is a primary purpose of analysis, one should look at the estimate from the IRE. If there is a known short or no lag, then using the SRE would result in less biased estimates of β and better detection performance. If there is an unknown lag or a moderate lag is known, the IRE should be used for detection and estimation. A more pragmatic approach would be to compare estimates of δ from the SRE and IRE and see how different they

are. If there is a common lag, using measure of central tendency (mean or median) for each method could help compare estimates of δ .

For testing activation, comparing the sets of voxels categorized activated from one method but not the other may prove useful in how the methods differentially detect. One may denote activated voxels by the subset that both methods categorize as activated to be highly conservative or the subset that either method call activated to be more liberal. These choices are dependent on the goals of analysis and the cost of false positives.

5.4 Computational Cost

When using iterations in a method where many units are analyzed, computational cost must be weighed. The computational cost increases exponentially with the number of voxels and number of scans in a given analysis. The prediction matrix of fitted values computed from the hat matrix can be quite large depending on the number of voxels and scans. The true computational cost compared to that of the non-iterated ratio estimator is the number of iterations the method must undergo. Each iteration must again compute the variance/covariance matrix $\sigma(X'X)^{-1}$, where X is the design matrix. The tables of average number of iterations show that the estimate of δ converges quickly with a low tolerance. Setting a less conservative tolerance would then allow an estimate to converge more readily and can reduce computation time. Though this method may save time, setting too high a tolerance will not give as accurate results.

Another computational consideration is the assumption of a common shift across voxels. If a common lag was assumed, one could use a mean or median lag for each cluster and compute cluster-specific shifted hrf matrices for analysis during each iteration. With a relatively small number of clusters, this would have small computational cost. Also, the overall estimate of the shift outperforms the voxel-wise estimator if

there truly is a common lag across voxels since the mean of the estimate should be unbiased.

If no common lag is assumed, one would estimate a voxel-wise shift and use this shifted hrf to estimate activation in the brain. A voxel-specific design matrix must be computed with the estimated voxel-specific shift. As the number of voxels increase, the computational cost increases greatly. Couple this with the iterative process, the time to analyze one subject can be quite long.

Some ways to save computation time could be to group voxels during each iteration with a “common shift” that is within a certain tolerance level (i.e. 0.1 seconds) and use one design matrix to get estimates. This is different than clustering the voxels or having a cluster a priori. These voxels are not assumed to have any similarity, including spatial proximity. The use of parallel computing could also greatly reduce computation time since the process would be common for all voxels.

Pre-processing the design matrix, hat matrices, and covariance matrix for each $\delta \in [\Delta_1, \Delta_2]$ reduces computation time dramatically. This method was not employed in the simulations since the computation time was reasonable. As the size of the analysis increased, the method was employed to decrease time in the Application analysis in section 6.3.

Chapter 6

Application

This chapter discusses using the SRE and the IRE and the testing methods described above on actual fMRI data. The purpose of the application is to show that these methods can be used on actual fMRI data and have reasonable computation time in practice.

6.1 Description of the Subjects

The data was collected on 10 women with anorexia nervosa and 10 age- and education-matched, non-dieting women at Johns Hopkins Medical Institutions. Participants were asked to perform the Iowa Gambling Task (IGT) (Bechara et al., 1995) while MRI images were taken. There were 501 scans taken for each subject at a sampling rate (TR) of 2 seconds per scan, totaling a scanning time of around 17 minutes per subject.

6.2 Iowa Gambling Task

The IGT consists of a screen where 4 decks are present, typically labeled A, B, C, D. Two of the decks have high reward but high penalties, called “high risk” decks and two decks have low reward/low penalty, called “low risk” decks. The goal for the subject is to maximize the amount of money won. Figure 6.1 shows the study paradigm alongside the view of the task the subject would have during each phase of

the task. Bechara et al. (1995) provides an example of a subject's series of choices. If a subject draws 10 times from high risk decks and receives \$100 each time, the subject will receive \$1000, but then again draws from these decks and loses \$1250, for a net loss of \$250. If drawing from the low risk decks 10 times and receives \$50 each time, but then draws and loses \$250, for a net gain of \$250. This just an example, but the overall gain should be higher if consistently choosing the low risks decks.

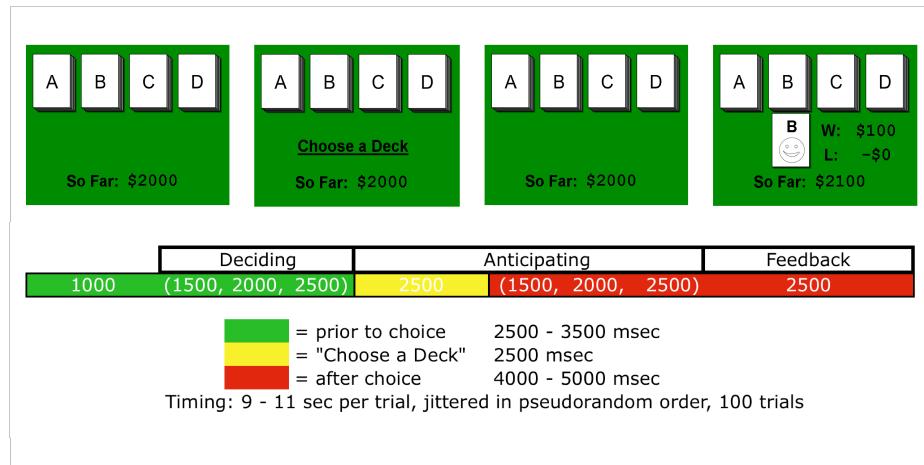


Figure 6.1: IGT Paradigm

6.3 Scope of Analysis and Preprocessing

The scope of this analysis is to use actual stimulus and response vectors from an fMRI study and analyze it using the SRE and IRE as stated above. In the overall data set, were three phases of stimulation that were recorded, called decision, anticipation, and feedback. The decision phase consisted of the time directly after the screen was renewed and the subject was asked to pick a deck. This phase was used as the stimulus vector for this analysis. The other phases could have been analyzed similarly.

The data was slice time corrected, realigned, and normalized to the MNI template. The data was smoothed using a Gaussian 8 mm^3 kernel. This preprocessing was not done by the thesis candidate. A “high pass filter” was used by regressing each voxel on time and time^2 and subtracting off the response by $\hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 t^2$ where t is time.

This was done to try to reduce scanner drift and non-linearity. This preprocessing and the subsequent analysis was done by the thesis candidate.

The overall purpose of the study was to examine differences in BOLD signal while decision-making in subjects with anorexia and controls, particularly with respect to subjects' insensitivity to the consequence of decisions. This purpose is close to the original purpose of the task, which was used to study impaired decision-making after injury to the ventromedial prefrontal cortex. The analysis provided for the thesis was to show a real-world application of the method and did not analyze the data to compare the groups studied. After analyzing the data and testing for activation using the methods here on each subject, an inter-group comparison could be made.

6.4 Analysis Details

The analysis was done in R statistical software (<http://cran.r-project.org/>) and had code motivated by SPM8 (referenced above). The code used for the analysis is in Section 10.1.3 in the Appendix. Below is an outline of the methods used that were implemented in R.

When implementing the actual method of convolution, the stimulus and hrf functions are zero-padded. This is done so that each vector had the sum of the length of the hrf function and the length of the stimulus function subtracted by 1 so that there was no “wraparound effect” caused when using fast Fourier transforms (Friston, 2006) (see Section 10.1.1 in the Appendix for details). Also, when using data that are in TR units, care must be taken to ensure that the convolution is in TR units when the analysis is done. The sets for tolerance and maximum number of iterations were arbitrary and can be changed to reflect desired precision.

The method of pre-processing referred from above in section 5.4 was implemented in analyzing the data to reduce computation time. Using a “grid” of 0.01s, each $\delta \in [0, 6]$, the design, hat, and covariance matrices were pre-made. The estimated

$\hat{\delta}$ was rounded to the hundredth of a second and the appropriate matrix was drawn from the preprocessed library of matrices.

The design matrix is referred to as X , which is 501 by p , where p is the number of predictors, either 2 or 3. The predictors were a vector of 1's (for the intercept), the hrf convolved with the stimulus, and the temporal derivative convolved with the stimulus if included in the model. The hat matrices were defined as $(X'X)^{-1}X'$ and $X(X'X)^{-1}X'$, since they put the “hat” on β and Y , to get $\hat{\beta}$ and \hat{Y} , respectively. The pre-made covariance matrix was defined as $(X'X)^{-1}$, where the true covariance matrix $\sigma(X'X)^{-1}$ was computed for each iteration. Thus, 601 of the matrices described above were made for each corresponding δ . The preprocessing time was around 15 minutes, but resulting in a data file over 2 gigabytes of memory. This is noted to show the preprocessing can reduce overall computation time but has a slight and trivial tradeoff in hard disk usage, especially if the number of scans increases dramatically.

Parallel computing was also employed, which allowed a sample of approximately 200,000 voxels to be analyzed in about 2 hours, after being into 4 parts. The one drawback of this method is that the estimation of δ and tolerance is dependent on the choice of the grid. For example, if the preprocessed matrices are made for 0s, 0.1s, 0.2s,..., 6s, then estimating δ within 0.01s is not possible. Thus, a smaller grid allows for more precise estimation of δ , but requires increased non-trivial pre-processing time and memory, since the precision is on the log scale. The grid used was 0.01s, which is considered a fine grid for estimation. This grid was used to show that even with high precision, the method was not unreasonably computationally intense.

6.5 Results

The results show that the computation time for analysis is reasonable and that actual fMRI data can be analyzed with these methods. Figure 6.2 shows the results for one subject (subject 10) from an analysis. This subject was a control subject. The results

show that there are many voxels zero out as by the spike in the histogram of the lag estimates. The shading on the brain template displays how large the lag estimates are. Note the scales are not the same for the IRE estimates (upper portion) and the SRE estimates (lower portion).

These maps and histograms show that the IRE took a larger range of values than the SRE. Also, it seems that the IRE detects lags in areas that the SRE does not. These figures imply that spatial maps of δ can be processed using this method. Also, if the lags are restricted to only voxels categorized activated, this may provide a useful tool to see clustering. The histogram of the estimates from the IRE also show a clustering of estimates that is not present in the histogram from the SRE. This may indicate the IRE can detect longer shifts in real data compared to the SRE. Overall, the application results show that this method can be reasonably applied to actual fMRI data.

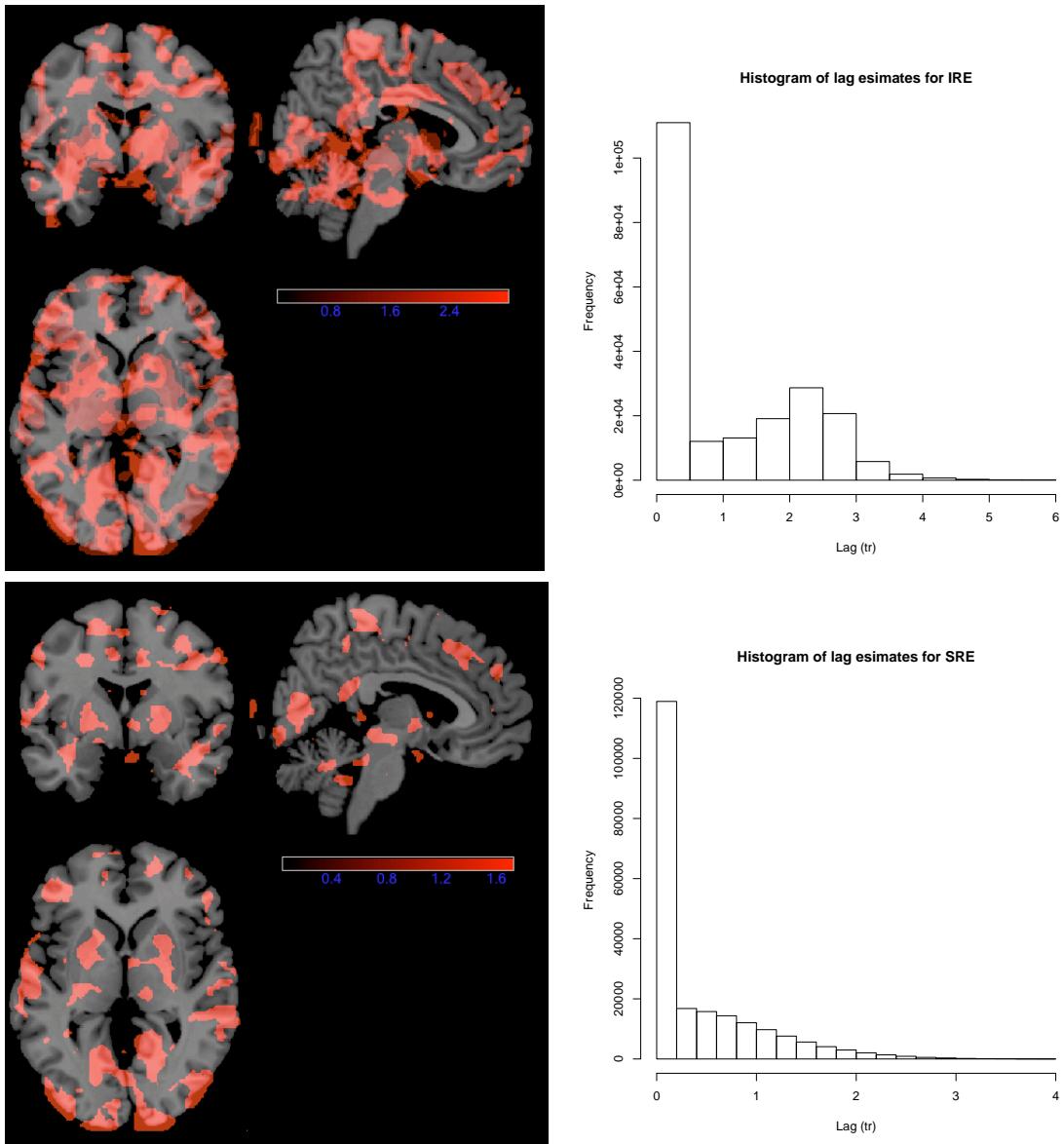


Figure 6.2: Spatial Maps and Histograms of Lags for Subject 10. The upper portion of the figure is a spatial map and histogram of the estimate of δ , respectively, for the IRE. The lower portion is the same figures for the SRE. Note the two scales are not identical for the spatial map.

Chapter 7

Conclusion

7.1 Questions Revisited

After looking at the results from simulations and the application, one can address the questions as posed before. **Is the lag itself intrinsically of interest?** This question is not essentially a statistical question but rather a biological one. As the analysis in Buckner et al. (1998) shows, different areas of the brain may show activation when using hrfs that have different δ parameters. The difference in these activations may be random noise or a reflection of the vasculature or some other physiological component of the brain. The simulations shows that one can obtain a good approximation of δ . This may have implications of using the estimated δ as a surrogate for measuring some biological process. Also, one can detect activation better with a better estimate of δ , which implies that estimation of the shift is intrinsically of interest.

Can the estimate of the lag be used as a plug-in estimate for subsequent testing of activation? The results of the bias of $\hat{\delta}$ from the simulations show that the estimates obtained by the IRE and SRE can be used as reasonably accurate estimates for δ . If the lag is considered to be short, the SRE does reasonably well. For longer lags, converged and non-converged estimates of the IRE may prove more useful and close to the true value. Investigation of maxed and zeroed out estimates may prove more complex. Overall, the SRE performs better with short lags and the IRE performs better with longer lags and the plug-in estimate can be used to

approximate the true model.

Can the lag be assumed to be constant over the whole brain? Again, this is a more a biological question than a statistical one. Preliminary analysis of subject 10, as well as the other subjects shown in the Appendix, shows that different areas of the brain probably have varying lags. So, a cluster-specific lag may be reasonable in some situations, but a common lag for the whole brain is probably not appropriate.

7.2 Simulation Results Regarding Testing for Activation

The simulations showed that the composite test may perform well in detecting activation, especially for voxels with longer lags. The one-sided F-statistic performs moderately well, but not as well as the standard t-test when the lag can be estimated accurately. The tests performed show that for longer lags, tests using the IRE performs more accurately. These results seem to imply that with a better estimate of δ , testing procedures do better. Thus, the estimation of δ should be of interest, even when the overall analysis goal is testing for activation.

Chapter 8

Further Areas of Research

The results from the simulations and application show that this method can be used for fMRI data. The next step is to compare analyses from these methods with those commonly done. Also, using these methods in conjunction with current methods may prove useful in practice. This chapter describes some ways to compare and join the IRE with other methods and slight changes to the IRE procedure.

8.1 Comparing $\hat{\delta}$'s across Methods

Comparing the estimated δ 's from the SRE or IRE to the δ that maximizes the activation function within the cone T-statistic may be of interest. This would allow a researcher to have essentially multiple estimates of a shift. Methods of combining these shifts in a weighted estimate may be worthwhile. Also, using a known area of activation with a known shift would allow researchers to directly compare methods for estimation. Although the δ from the cone T statistic is not intrinsically found by estimation, it may show how different the estimated t-statistic maps are between the IRE model and the cone statistic.

8.2 Using δ in Approximating the HRF

One must need fMRI data taken at a fine time scale to get an accurate approximation of the hrf and its parameters. Henson et al. (2002) describes that the estimated δ

parameter can be used post hoc in higher order Taylor approximations of the true hrf, which may more accurately estimate the real hrf. The use of the estimated δ to estimate the true hrf was not done here since a common hrf was assumed over all voxels. If one area has a known form, as in Glover (1999), then looking at an area with a suspected shift would shed light on how the methods compare practically.

To compare methods, discordant voxels, voxels that one method deemed active an another did not or vice versa, would be of high interest. If one can understand why certain methods are differing in these voxels, one may be able to revise current methods to gain predictive accuracy.

8.3 Going to 3D Space

Since this data was looking at a voxel-wise analysis of δ , using spatial data to find voxels with a similar shift close to each other may be used to define clusters. Pooling estimates of δ from neighborhood voxels may allow for a better estimation of the shift or the level of activation in a voxel. Also, common methods for correlated data may give better estimates for voxels close in proximity. These results were not presented here since the proof of method was for an estimation of δ . Using some metric of centrality such as the mean or median for a cluster of voxels can be employed and a common design matrix can be used to estimate estimates for the cluster. This method would be more computationally efficient as well.

8.4 Different Convergence Limits

The iteration algorithm here is set to converge if the change in the estimate of δ from one iteration to the next is essentially zero. One can slightly change the algorithm to extend the iteration until the estimate for β_2 in Equation (4.5) is less than some tolerance. If $\beta_2 \rightarrow 0$, then Equation (4.5) should converge to Equation (4.1). Friston (2006) describes that using a set of basis functions that have a range of δ can be

used with the SRE so that functions with δ close to the true shift would be able to approximate well since the expansion should be valid. This method is essentially minimizing β_2 over values of $\delta \in [\Delta_1, \Delta_2]$. This choice of δ is similar to that of the cone T statistic, except using a minimizing procedure.

8.5 Use of δ as a Grouping Factor

Determining whether δ can truly act as a biologically meaningful measure is of much interest. If an investigator thinks this is true, a technique could be developed to group subject by δ . This would be useful if one was interested in investigating how vulnerability factors affect cognitive function. For example, lags in certain areas of the brain may correspond to a cognitive defects, which would be useful in grouping subjects by cognitive similarity in a certain area. This purpose of δ needs to be investigated further.

Chapter 9

Summary

Overall, the review of testing shows that there are post-hoc approaches for the IRE that perform well. When methods for estimating δ do well, the models used should approximate the true model and the tests on these models should perform better than the models with the derivative included. If evidence suggests a long lag time, the IRE may outperform the SRE in estimating δ . One should compare activated voxels in both methods, as well as other methods, to have more confidence in categorizing areas as activated.

When testing, even if the estimation of δ is not of intrinsic/biologic interest, using a good estimate of δ can improve detection of activation. Therefore, estimating δ and testing for activation are intertwined and must be considered when analyzing any BOLD response-related data. With the iterated ratio estimator, we think that it can estimate δ with reasonable accuracy.

Thus, using the temporal derivative in a linear regression framework can help estimate the true response with a known hrf. Its use in analysis can be very helpful in detecting activation as well as estimating parameters of the hrf. In conclusion, including the hrf's temporal derivative in fMRI analysis can improve parametric mapping and testing procedures by improving approximation of the response.

Bibliography

- Aguirre, G., Zarahn, E., and D'esposito, M. (1998). The variability of human, BOLD hemodynamic responses. *Neuroimage*, 8(4):360–369.
- Bechara, A., Damasio, A., Damasio, H., and Anderson, S. (1995). Insensitivity to future consequences following damage to human prefrontal cortex. *Cognition on cognition*, 1.
- Biswal, B., Kannurpatti, S., and Rypma, B. (2007). Hemodynamic scaling of fMRI-BOLD signal: validation of low-frequency spectral amplitude as a scalability factor. *Magnetic resonance imaging*, 25(10):1358–1369.
- Buckner, R., Koutstaal, W., Schacter, D., Dale, A., Rotte, M., and Rosen, B. (1998). Functional-anatomic study of episodic retrieval:: Ii. selective averaging of event-related fMRI trials to test the retrieval success hypothesis. *Neuroimage*, 7(3):163–175.
- Buxton, R., Wong, E., and Frank, L. (1998). Dynamics of blood flow and oxygenation changes during brain activation: the balloon model. *Magnetic Resonance in Medicine*, 39(6):855–864.
- Calhoun, V., Adali, T., Pekar, J., and Pearlson, G. (2003). Latency (in) sensitive ICA Group independent component analysis of fMRI data in the temporal frequency domain. *NeuroImage*, 20(3):1661–1669.
- Calhoun, V., Stevens, M., Pearlson, G., and Kiehl, K. (2004). fMRI analysis with the

- general linear model: removal of latency-induced amplitude bias by incorporation of hemodynamic derivative terms. *Neuroimage*, 22(1):252–257.
- Evans, A., Collins, D., and Milner, B. (1992). An MRI-based stereotactic atlas from 250 young normal subjects. *Soc. neurosci. Abstr*, 18:408.
- Friman, O., Borga, M., Lundberg, P., and Knutsson, H. (2003). Adaptive analysis of fMRI data. *NeuroImage*, 19(3):837–845.
- Friston, K., Fletcher, P., Josephs, O., Holmes, A., Rugg, M., and Turner, R. (1998). Event-related fMRI: characterizing differential responses. *Neuroimage*, 7(1):30–40.
- Friston, K. J. (2006). *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 1 edition.
- Glover, G. (1999). Deconvolution of impulse response in event-related BOLD fMRI. *NeuroImage*, 9(4):416–429.
- Handwerker, D., Ollinger, J., and D'Esposito, M. (2004). Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *Neuroimage*, 21(4):1639–1651.
- Henson, R., Price, C., Rugg, M., Turner, R., and Friston, K. (2002). Detecting latency differences in event-related BOLD responses: application to words versus nonwords and initial versus repeated face presentations. *Neuroimage*, 15(1):83–97.
- Kruggel, F. and Von Cramon, D. (1999). Temporal properties of the hemodynamic response in functional MRI. *Human Brain Mapping*, 8(4):259–271.
- Lange, N. and Zeger, S. (1997). Non-linear fourier time series analysis for human brain mapping by functional magnetic resonance imaging. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 46(1):1–29.

- Liao, C., Worsley, K., Poline, J., Aston, J., Duncan, G., and Evans, A. (2002). Estimating the delay of the fMRI response. *NeuroImage*, 16(3PA):593–606.
- Lorberbaum, J., Bohning, D., Shastri, A., Nahas, Z., and George, M. (2010). Functional Magnetic Resonance Imaging (fMRI) for the Psychiatrist.
- Rosen, B., Buckner, R., and Dale, A. (1998). Event-related functional mri: past, present, and future. *Proceedings of the National Academy of Sciences*, 95(3):773.
- Schroeter, M., Kupka, T., Mildner, T., Uludağ, K., and von Cramon, D. (2006). Investigating the post-stimulus undershoot of the BOLD signals a simultaneous fMRI and fNIRS study. *Neuroimage*, 30(2):349–358.
- Talairach, J. and Tournoux, P. (1988). *Co-planar stereotaxic atlas of the human brain: 3-dimensional proportional system: an approach to cerebral imaging*. Thieme.
- Worsley, K. and Taylor, J. (2006). Detecting fMRI activation allowing for unknown latency of the hemodynamic response. *Neuroimage*, 29(2):649–654.

Chapter 10

Appendix

The first iteration assumes $\delta_0 = 0$, so $x(t - \delta_0) = x(t)$ and $x'(t - \delta_0) = x'(t)$ and is simply the Liao et al. (2002) estimator.

$$\begin{aligned}\frac{\beta_{2i1}}{\beta_{1i1}} &= \delta - \delta_0 \\ \frac{\beta_{2i1}}{\beta_{1i1}} + \delta_0 &= \delta \\ \delta_0 &= 0 \\ \frac{\beta_{2i1}}{\beta_{1i1}} &= \hat{\delta}_1\end{aligned}$$

The process is then iterated with $\delta_{01} = \tilde{\delta}_1$:

$$\begin{aligned}Y_i(t) &\approx \beta_{0i} + \beta_i[x(t - \delta_{01}) - (\delta - \delta_{01})x'(t - \delta_{01})] + \varepsilon_i \\ &= \beta_{0i} + \beta_{1i}x(t - \delta_{01} - \beta_{2i}(\delta - \delta_{01}))x'(t - \delta_{01}) + \varepsilon_i \\ &\Rightarrow\end{aligned}$$

$$\begin{aligned}\frac{\beta_{2i2}}{\beta_{1i2}} + \delta_{01} &= \hat{\delta}_2 \\ \tilde{\delta}_2 &= \frac{\hat{\delta}_2}{1 + \frac{1}{T_{1i2}^2}} \\ \delta_{02} &= \tilde{\delta}_2\end{aligned}$$

Overall, the general form is:

$$\begin{aligned}
 Y_i(t) &\approx \beta_{0ij} + \beta_{1ij}x(t - \delta_{0j}) - \beta_{2ij}(\delta - \delta_{0j})x'(t - \delta_{0j}) + \varepsilon_i \\
 \frac{\beta_{2ij}}{\beta_{1ij}} + \delta_{0(j-1)} &= \hat{\delta}_j \\
 \tilde{\delta}_j &= \frac{\hat{\delta}_j}{1 + \frac{1}{T_{1ij}^2}} \\
 \delta_{0j} &= \tilde{\delta}_j
 \end{aligned}$$

Let β_{kij} denote the i^{th} voxel, and j^{th} iteration for β_k , $k = 1, 2$.

10.1 R code

10.1.1 HRF and Design Functions

The function my.hrf returned the hrf and temporal derivative of the hrf for a given stimulus vector in seconds (x) and a set of parameters for the hrf. The design function took parameters x, how fine the hrf should be made (by), the TR sampling, the number of scans, and the delay (onset). If the stimulus vector was given in seconds, then the every parameter is neglected. The hrf and stimulus vector is zero padded to avoid wraparound. The fast Fourier transform (fft) of the hrf and the temporal derivative with the stimulus vector, sampled at TR units, are returned. The result is then not in seconds, but TR, since the response is in TR units.

```

my.hrf <- function(x,
grid = 1,
delayR = 6,
          delayU = 16,
          dispR = 1,
          dispU = 1,
          ratioRU = 6,
          onset=0,
          klength = 32 ){

shapeR <- delayR/dispR
rateR <- 1/dispR
shapeU <- delayU/dispU
rateU <- 1/dispU
ratioRU <- ratioRU

```

```

delta_t <- 1/grid
# USE a lag and then see if derivative actually can give good diagnostic
#x needs to be in seconds (maybe TR if onset is in TR?)r==0)
x <- x - onset
x[which(x <= 0)] <- 0
hrf <- dgamma(x, shape = shapeR, rate = rateR) - dgamma(x, shape = shapeU,
rate = rateU) / ratioRU;
hrf <- hrf/max(abs(hrf))
driv <- vector(length=length(hrf)-1)
for(i in 1:length(hrf)-1){
driv[i] <- (hrf[i+1]- hrf[i])/(delta_t)
}
driv <- c(0, driv)
return(list(hrf=hrf, driv=driv))
}

design <- function(x= NULL, onset=0, scans=500, tr=1, by=0.1, every=32){
grid <- 1/by
u <- seq(0, 32, by = by)
seconds <- seq(0, tr*scans, by=by)
if (is.null(x)){
x <- rep(c(rep(0, length=every/by-1),1), length=scans/by +1)
}
else {
#x <- rep(c(rep(0, length=every/by-1),1), length=scans/by +1)
x <- x
}
hrf <- my.hrf(u, onset=onset, grid=grid)$hrf
driv <- my.hrf(u, onset=onset, grid=grid)$driv
pad <- length(hrf)-1
lenx <- length(x)
x<- c(x, rep(0, length=pad))
hrf <- c(hrf, rep(0, length=(length(x)-length(hrf))))
driv <- c(driv, rep(0, length=(length(x)-length(driv))))
res <- convolve(x, hrf, conj=FALSE)
dres <- convolve(x, driv, conj=FALSE)
res_true <- res[-c((lenx+1):(lenx+pad))]
x <- x[-c((lenx+1):(lenx+pad))]
hrf <- hrf[-c((lenx+1):(lenx+pad))]
dres <- dres[-c((lenx+1):(lenx+pad))]
samples <- which(seconds %% tr == 0)
res <- res_true[samples]
dres <- dres[samples]
if (!is.null(every)) x <- rep(c(rep(0, length=every-1),1),
length=scans +1)
else x <- x[samples]
return(list(res=res, dres=dres, x=x))
}

```

10.1.2 Simulations

The run function is the workhorse function and is called for every iteration and returns a data frame and estimate of δ . The checks data frame is the Derivative IRE model, checklasts the No Derivative IRE model, the check_methods the Derivative SRE, and the check_methodlasts the No Derivative SRE model, and the checktrue data frame is the model fit with the true lag.

```

setwd("~/gambling")
source("./programs/hrf_function_20100208.R")
run <- function(i, pred=3, delta){
  result.list <- design(onset=delta, every=every)
  result <- result.list$res
  dresult <- result.list$dres
  pred <- pred
  X <- matrix(NA,nrow=t, ncol=pred)
  X[,1] <- 1
  X[,2] <- result
  if (pred==3) X[,3] <- -dresult

  Hatb <- solve(t(X)%*%X)%*%t(X)
  if (class(Hatb)=="try-error") break
  Haty <- X%*%Hatb

  Betahat <- Hatb%*%Y_0[,i]
  Yhat <- Haty%*%Y_0[,i]

  resid <- Y_0[,i]-Yhat
  sum2 <- apply(resid, 2, function(x) sum(x^(2)))
  varhat <- sum2/(t-pred)

  Covar <- solve(t(X)%*%X)
  Varb1 <- varhat*Covar[2,2]
  if (pred==3) Varb2 <- varhat*Covar[3,3]

  if (pred==3) check <- data.frame(cbind(B0 = Beta[1,i], B1 = Beta[2,i],
  b0 = Betahat[1,], b1= Betahat[2,], b2 = Betahat[3,],
  t1 = Betahat[2,]/sqrt(Varb1),Varb1=Varb1,
  t2 = Betahat[3,]/sqrt(Varb2),Varb2=Varb2,
  lag=Betahat[3,]/Betahat[2,]) )
  if (pred==2) check <- data.frame(cbind(B0 = Beta[1,i], B1 = Beta[2,i],
  b0 = Betahat[1,], b1= Betahat[2,],
  t1 = Betahat[2,]/sqrt(Varb1),Varb1=Varb1) )
  df = t-pred

  check$p <- with(check, pt(t1, df=df, lower.tail=FALSE))
  if (pred==3) {
    check$F <- with(check, pf(sign(b1)*(t1^2+ t2^2)/2,
    df1=2, df2=df, lower.tail=FALSE))
    check$lag <- check$lag/(1+1/(check$t1^2))
    deltahat <- check$lag
  }
}

```

```

threshhold <- 6

}

if (pred==2) {
check$F <- with(check, pf(sign(b1)*(t1^2), df1=1,
df2=df, lower.tail=FALSE))
deltahat <- 0
}

delta <- deltaxhat + delta
check$sigp <- check$p < 0.05
check$sigF <- check$F < 0.05
if (delta > threshhold) delta <- threshhold
if (delta < 0){
  delta <- 0
  deltaxhat <- 0
}

check$delta <- delta
return(list(check=check, delta=delta, deltaxhat=deltahat))
}

for(thelag in c(1, 2, 4)){
for(thej in c(-1, -2,-3)){
every=30
lagger <- thelag
j <- thej
by=0.1
g <- lag <- lagger
t <- 501
vox <- 20000
# p is percent activated voxels
p <- .5
# USe a lag and then see if derivative actually can give good diagnostic
res_true <- design(onset=lag, by=by, every=every)$res
#Potential betas are in simbetas
pred <- 2
X_0 <- matrix(NA,nrow=t, ncol=pred)
X_0[,1] <- 1
X_0[,2] <- res_true
#Beta is 2 (predictors) by
Beta <- matrix(NA,ncol=vox, nrow=pred)
Beta[1,] <- 1
activated <- rep(1, length = vox*p)
for(i in 1:(vox*p)){
Beta[2,i] <- activated[i]*2^j
}
Beta[2,][is.na(Beta[2,])] <- 0

sd <-1
E <- matrix(rnorm(vox*t, mean = 0, sd=sd), nrow=t, ncol=vox)
# Y is from TR = 1
Y_0 <- Y <- X_0 %*% Beta +E
threshhold=6
limit <- 100

```

```

lone<- NULL
checkit <-run(1, delta=0)$check
checks <- matrix(ncol=ncol(checkit))
colnames(checks) <- c(colnames(checkit))
check_methods <- checks
checkit2 <-run(1, pred=2, delta=0)$check
checklasts <- check_methodlasts <- checktrues <- matrix(ncol=ncol(checkit2))
colnames(checklasts) <- colnames(check_methodlasts) <-
colnames(checktrues) <- c(colnames(checkit2))
deltas <- matrix(nrow=vox, ncol=limit)
deltas[,1] <- 0
for(i in c(1:vox)) {

delta0 <- 0
delta <- delta0

check <- data.frame(cbind(b2=1, F=0, p = 0))
deltahat <- 1
iter <- 1

onedeltas <- 0
while(abs(deltahat) > 1e-8){

out <-run(i, delta=delta)
if (iter == 1) {
check_method <- out$check
check_method$lag <- (check_method$b2/check_method$b1)/
(1+1/check_method$t1^2)+delta0
}
deltahat <- out$deltahat
delta <- out$delta
onedeltas <- c(onedeltas, delta)

if(abs(deltahat) <= 1e-8 | iter ==100) {
checks <- rbind(checks, out$check)
}

#if(delta < 0) check$b2 <- 0
iter <-iter + 1
if (iter > limit) {
print(paste("DID NOT CONVERGE, deltahat", deltahat))
break
}
#While loop killer
}
check_methods <- rbind(check_methods, check_method)
lone <- c(lone, length(onedeltas))
if (iter <=limit) onedeltas <- c(onedeltas,
rep(onedeltas[length(onedeltas)], length=limit-length(onedeltas) ) )
if (iter > limit) onedeltas <- onedeltas[-length(onedeltas)]

deltas[i,] <- onedeltas

df <- t - 2
outlast <- run(i, pred=2, delta=delta)
checklast <- outlast$check
checklast$p <- with(checklast, pt((t1), df=df, lower.tail=FALSE))
checklast$F <- with(checklast, pf((t1^2), df1=1, df2=df, lower.tail=FALSE))
checklasts <- rbind(checklasts, checklast)

##LOOKAT TRUE MODEL

```

```

outtrue <- run(i, pred=2, delta=lag)$check
checktrue <- outtrue
checktrues <- rbind(checktrues, checktrue)

outmethod <- run(i, pred=2, delta=check_method$delta)$check
check_methodlast <- outmethod
check_methodlasts <- rbind(check_methodlasts, check_methodlast)

print(i)
}
checks <- checks[-1,]
checktrues <- checktrues[-1,]
checklasts <- checklasts[-1,]
check_methods <- check_methods[-1,]
check_methodlasts <- check_methodlasts[-1,]

all <- list(chcks=checks, checktrues=checktrues, checklasts=checklasts,
check_methods=check_methods, check_methodlasts=check_methodlasts,
lag=lagger, deltas=deltas, lone=lone, Y_0=Y_0, j=j, Beta=Beta, delta0=delta0,
vox=vox, t=t, limit=limit, threshhold=threshhold, sd=sd)

b <- 2^thej
save(all, file=paste("./results/iter_", lagger, "_", b, "_cluster.rda", sep=""))
}
}

```

10.1.3 Data Compression

The NIFTI files were inputted, a mask was made by seeing if data was at every point within a subject for the number of time scans (501 in this case). This mask indexed points of the data set that had data at every point and was the map from the condensed 2-d structure to the 3-d spatial map. The patient.rda file made at the end of the program was the compressed 2-d data structure that was a voxel by time matrix.

```

rm(list=ls())
library(Rniftilib)

for (j in c("swra9009_10_1GamblingSENSE.nii",
"swra9020_9_1GamblingSENSE.nii",
"swra9010_10_1GamblingSENSE.nii",
"swra9021_9_1GamblingSENSE.nii",
"swra9011_9_1GamblingSENSE.nii",
"swra9022_9_1GamblingSENSE.nii",
"swra9012_9_1GamblingSENSE.nii",
"swra9023_080530_13_1GamblingSENSE.nii",
"swra9014_9_1GamblingSENSE.nii",
"swra9024_9_1GamblingSENSE.nii",
"swra9015_070907_10_1GamblingSENSE.nii",
"swra9025_9_1GamblingSENSE.nii",
"swra9026_9_1GamblingSENSE.nii",
"swra9017_9_1GamblingSENSE.nii",
"swra9027_9_1GamblingSENSE.nii",

```

```

"swra9018_071109_9_1GamblingSENSE.nii",
"swra9028_9_1GamblingSENSE.nii",
"swra9019_10_1GamblingSENSE.nii",
"swra9029_9_1GamblingSENSE.nii"
)) {
setwd("/nexsan/bst4/gambling/")
i <- substring(j, 7, 8)
x <- nifti.image.read(file=j, )
for(i in 1:dim(x)[4]) {
y <- summary(x[,,,1][which(x[,,,1] >10)])
binary <- (abs(x[,,,i]) > y["Mean"])
if (i == 1) mask <- binary
else mask <- mask * binary
}
rm(x)
save(mask, file=paste("~/gambling/masks/Patient", i,
"mask.rda", sep = ""))
}

for (j in c(16)) {
x <- nifti.image.read(file=paste("swra90", j,
"_9_1GamblingSENSE.nii", sep = ""))
for(i in 1:dim(x)[4]) {
binary <- (abs(x[,,,i]) > 200)
if (i == 1) mask <- binary
else mask <- mask * binary
}
rm(x)
save(mask, file=paste("~/gambling/masks/Patient", j,
"mask.rda", sep = ""))
}

for (i in c(9, 10:12, 14:29)){
load(file=paste("~/gambling/masks/Patient", i, "mask.rda", sep=""))
temp <- which(mask == 1)
rm(mask)
x <- nifti.image.read(file=
paste("swra90", i, "_9_1GamblingSENSE.nii", sep = ""))
dat <- matrix(NA, 501, length(temp))
for (j in 1 : 501){
dat[j,] <- x[dim1=1 : 79, dim2=1 : 95, dim3=1 : 69, dim4 = j][temp]
}
rm(x)
mask <- dat
rm(dat)
save(mask, file=paste("~/gambling/compressed/Patient", i, ".rda",
sep=""), compress=TRUE)
## Dat is 501 by 72000,
}

```

10.1.4 Analysis of Subjects in Application

This code is similar to the simulation code but with one large difference, the design matrices are already created in the list “stuff” from design_matrices.rda. The data

was also split into 4 parts (hence the split variable) to reduce run-time. The data was then merged below to give a Patient_done.rda file. The data was also transformed back into the 3-d structure for a t-statistic map (ts), significance map (sig) and lag map for each subject. The 4th dimension in the NIFTI file is No Derivative IRE, No Derivative SRE, Derivative IRE, Derivative SRE for the ts and sig files and No Derivative IRE, No Derivative SRE lags and another two images of No Derivative IRE, No Derivative SRE lags, but only looking at activated voxels as cutoff by $\alpha = 0.05$.

```

rm(list=ls())
setwd("~/gambling")
source("./programs/hrf_function_20100208.R")
pt <- 10
split <- 1
id <- 9000 + pt
load(file=paste("./compressed/Patient", pt,
"_filtered.rda", sep=""))
stimulus <- read.csv("stimulus_data.csv", header=TRUE)
vox <- 54000
mask <- mask[, (1+ vox*(split-1)):(vox*split)]
t <- nrow(mask)
stim <- stimulus[which(stimulus$ID==id),]
stim <- round(stim,1)
by=0.1
grid = 1/by
tr <- 2
scans=501
u <- round(seq(0, scans*tr, by=by), 2)
x <- rep(0, length=length(u))
x[!is.na(match(u, stim$Decision.TR*2))] <-1
delta <- 0
load("design.matrices.rda")

run_graham <- function(i, pred=3, delta, by = 0.1, x ){
  d <- round(delta, 2)*1/0.01+1

  if (pred==3) {
    Betahat <- stuff[[d]]$Hatb%*%mask[,i]
    Yhat <- stuff[[d]]$Haty%*%mask[,i]
    resid <- mask[,i]-Yhat
    varhat <- sum(resid^2)/(t-pred)
    Covar <- stuff[[d]]$Covar
    check <- matrix(c(b0 = Betahat[1,], b1= Betahat[2,],
    b2 = Betahat[3,],
    t1 = Betahat[2,]/sqrt(varhat*Covar[2,2]),
    Varb1= varhat*Covar[2,2],
    t2 = Betahat[3,]/sqrt( varhat*Covar[3,3]),
    Varb2= varhat*Covar[3,3],
    
```

```

lag=(Betahat[3,]/Betahat[2,])/(1+1/(Betahat[2,]^2/
(varhat*Covar[2,2]))),
delta=NA), ncol=9 )
colnames(check) <- c("b0", "b1", "b2", "t1", "Varb1", "t2", "Varb2", "lag", "delta")
deltahat <- check[, "lag"]

}
if (pred==2) {
Betahat <- stuff2[[d]]$Hatb%*%mask[,i]
Yhat <- stuff2[[d]]$Haty%*%mask[,i]
resid <- mask[,i]-Yhat
varhat <- sum(resid^2)/(t-pred)
Covar <- stuff2[[d]]$Covar
check <- matrix(c(b0 = Betahat[1,],
b1= Betahat[2,], t1 = Betahat[2,]/
sqrt(varhat*Covar[2,2]), Varb1= varhat*Covar[2,2],
delta=NA), ncol=5 )
colnames(check) <- c("b0", "b1", "t1", "Varb1", "delta")
deltahat <- 0
}
delta <- deltax + delta
if (delta > threshhold) delta <- threshhold
if (delta < 0){
  delta <- 0
  deltax <- 0
}

check[, "delta"] <- delta
return(list(check=check, delta=delta, deltax=deltahat))

}
tr <- nrow(mask)
vox <- ncol(mask)
threshhold <- 6
limit <- 100
lone<- NULL
checkit <- run_graham(1, x = x, by=by, pred=3, delta=0)$check
checks <- matrix(ncol=ncol(checkit), nrow=vox)
colnames(checks) <- c(colnames(checkit))
check_methods <- checks

checkit2 <- run_graham(1,x=x, pred=2, delta=0)$check
checklasts <- check_methodlasts <- matrix(ncol=ncol(checkit2),
nrow=vox)
colnames(checklasts) <- colnames(check_methodlasts) <-
c(colnames(checkit2))

d.conv <- matrix(ncol=ncol(mask), nrow=limit)

for(i in 1:vox){
  delta <- delta0 <- 0

#check <- data.frame(cbind(b2=1, F=0, p = 0))
deltahat <- 1
iter <- 1

```

```

onedeltas <- 0
while(abs(deltahat) > 1e-3){
  out <- run_graham(i, delta=delta, x=x, by=by)
  if (iter == 1) {
    check_method <- out$check
  }
  deltahat <- out$deltahat
  delta <- out$delta
  onedeltas <- c(onedeltas, delta)
  #if(delta < 0) check$b2 <- 0
  iter <- iter + 1
  if (iter >= limit) {
    #print(paste("DID NOT CONVERGE, deltahat", deltahat))
    break
  }
}
checks[i,] <- out$check

check_methods[i,] <- check_method
lone <- c(lone, length(onedeltas))
if (length(onedeltas) < limit) onedeltas <- c(onedeltas,
rep(onedeltas[length(onedeltas)], length = limit-
length(onedeltas)))
d.conv[,i ] <- onedeltas
onedeltas <-NULL

outlast <- run_graham(i, x=x, pred=2, delta=delta)
checklast <- outlast$check
checklasts[i,] <- checklast

check_methodlasts[i,] <- run_graham(i, x=x, pred=2,
delta=check_methods[, "delta"]$check
if (i %% 100==0) print(i)
}

#check_methods[,"lag"] <- (check_methods[, "b2"]/
#check_methods[, "b1"])/(1+1/check_methods[, "t1"]^2)+delta0

#checklasts[,"p"] <- pt(checklast[,"t1"], df=t-2,
lower.tail=FALSE)
#checklasts[,"F"] <- pf(checklast[,"t1"]^2, df1=1, df2=t-2,
lower.tail=FALSE)

all <- list(checks=checks, checklasts=checklasts, lone=lone,
check_methods=check_methods, check_methodlasts=
check_methodlasts)
save(all, compress=TRUE, file=paste("./compressed/graham_pt_",
pt, "_", split, ".rda", sep=""))

rm(list=ls())
setwd("~/gambling")
library(Rniftilib)
for(patient in c(9, 10, 11, 12, 14:29)){
  checks <- checklasts <- check_methods <-
  check_methodlasts <- iters <- NULL
}

```

```

setwd("/home/bst/student/jmuschel/gambling/")
#checks <- derivative IRE model
#checklasts <- no derivative IRE model
#check_methods <- derivative SRE model
#check_methodlasts <- no derivative SRE model
#lone # of iterations
for(split in 1:4){

  load(file=paste("./compressed/graham_pt_", patient,
  "_", split, ".rda", sep=""))
  checks <- rbind(checks, all$checks)
  checklasts <- rbind(checklasts, all$checklasts)
  check_methods <- rbind(check_methods,
  all$check_methods)
  check_methodlasts <- rbind(check_methodlasts,
  all$check_methodlasts)
  iters <- c(iters, all$lone)
}

dim(checks)
length(iters)
checks <- data.frame(checks)
checklasts <- data.frame(checklasts)
check_methods <- data.frame(check_methods)
check_methodlasts <- data.frame(check_methodlasts)

save(checks, checklasts, iters, check_methods,
check_methodlasts,
file=paste("./results/Patient", patient, "_analysis.rda", sep=""),
compress=TRUE)

ttimes <- 501
checklasts$tp <- pt(checklasts$t1, df=ttimes-2, lower.tail=FALSE)
checks$Fp <- with(checks, pf(sign(t1)*(t1^2 + t2^2)/2, df1=2,
df2= ttimes-3, lower.tail=FALSE))
checkst <- matrix(c(checks$t1, checks$t2), ncol=2)
colnames(checkst) <- c("t1", "t2")
checks$t <- apply(checkst, 1, max )
checks$tp <- pt(checks$t, df= ttimes-3, lower.tail=FALSE)

check_methodlasts$tp <- pt(check_methodlasts$t1,
df= ttimes-2, lower.tail=FALSE)
check_methods$Fp <- with(check_methods,
pf(sign(t1)*(t1^2 + t2^2)/2, df1=2, df2= ttimes-3, lower.tail=FALSE))
check_methodst <- matrix(c(check_methods$t1, check_methods$t2), ncol=2)
colnames(check_methodst) <- c("t1", "t2")
check_methods$t <- apply(check_methodst, 1, max )
check_methods$tp <- pt(check_methods$t, df= ttimes-3, lower.tail=FALSE)

sigit <- function(x, alpha=0.05){
  return(x <= alpha)
}

checklasts$sigtp <- sigit(checklasts$tp)
checks$sigFp <- sigit(checks$Fp)
checks$sigtp <- sigit(checks$tp, alpha=0.05)

check_methodlasts$sigtp <- sigit(check_methodlasts$tp)
check_methods$sigFp <- sigit(check_methods$Fp)
check_methods$sigtp <- sigit(check_methods$tp, alpha=0.05)

```

```

#load mask
load(file=paste("./masks/Patient", patient, "mask.rda", sep=""))

ire.ts <- sre.ts <- ire.d.ts <- sre.d.ts <- ire.sig <-
sre.sig <- sre.d.sig <- ire.d.sig
<- ire.lag <- sre.lag <- array(0, dim=dim(mask))
ire.d.Fps <- sre.d.Fps <- ire.d.tps <- sre.d.tps <-
array(1, dim=dim(mask))

ire.ts[which(mask==1)] <- round(checklasts$t1, 3)
sre.ts[which(mask==1)] <- round(check_methodlasts$t1, 3)
ire.d.ts[which(mask==1)] <- round(chcks$t, 3)
sre.d.ts[which(mask==1)] <- round(check_methods$t, 3)

ire.d.Fps[which(mask==1)] <- round(chcks$Fp, 3)
sre.d.Fps[which(mask==1)] <- round(check_methods$Fp, 3)

ire.d.tps[which(mask==1)] <- round(chcks$tp, 3)
sre.d.tps[which(mask==1)] <- round(check_methods$tp, 3)

ire.ts[which(ire.ts <0)] <- 0
sre.ts[which(sre.ts <0)] <- 0
sre.d.ts[which(sre.d.ts <0)] <- 0
ire.d.ts[which(ire.d.ts <0)] <- 0

ire.sig[which(mask==1)] <- checklasts$sigtp
sre.sig[which(mask==1)] <- check_methodlasts$sigtp
ire.d.sig[which(mask==1)] <- chcks$sigtp
sre.d.sig[which(mask==1)] <- check_methods$sigtp

ire.lag[which(mask==1)] <- checklasts$delta
sre.lag[which(mask==1)] <- check_methodlasts$delta

ire.sig.lag <- round(ire.lag*ire.sig, 3)
sre.sig.lag <- round(sre.sig*sre.lag, 3)

all <- list(ire.sig=ire.sig, ire.ts=ire.ts, ire.d.ts=ire.d.ts, ire.d.Fps=ire.d.Fps,
ire.d.tps=ire.d.tps, sre.sig=sre.sig, sre.ts=sre.ts, sre.d.ts=sre.d.ts,
sre.d.Fps=sre.d.Fps, sre.d.tps=sre.d.tps, ire.d.sig=ire.d.sig,
sre.d.sig=sre.d.sig, ire.lag=ire.lag, sre.lag=sre.lag,
ire.sig.lag=ire.sig.lag, sre.sig.lag=sre.sig.lag)

save(all, compress=TRUE, file=paste("./compressed/Patient_",
patient, "_done.rda", sep=""))

load( file=paste("./compressed/Patient_", patient, "_done.rda", sep=""))

setwd("/nexsan/bst4/gambling")
id <- patient + 9000
if (patient==9) img <- nifti.image.read(file=
paste("swra9009_10_1GamblingSENSE.nii", sep ""))
if (patient==10) img <- nifti.image.read(file=
paste("swra9010_10_1GamblingSENSE.nii", sep ""))
if (patient==18) img <- nifti.image.read(file=
paste("swra9018_071109_9_1GamblingSENSE.nii", sep ""))
if (patient==15) img <- nifti.image.read(file=
paste("swra9015_070907_10_1GamblingSENSE.nii", sep ""))
if (patient==19) img <- nifti.image.read(file=
paste("swra9019_10_1GamblingSENSE.nii", sep ""))

```

```

if (patient==23) img <- nifti.image.read(file=
paste("swra9023_080530_13_1GamblingSENSE.nii", sep ""))
if (!(patient %in% c(9, 10, 18, 15, 19, 23))) img <- nifti.image.read(file=paste("

setwd("/home/bst/student/jmuschel/gambling/nii/")

img$dim <- c(79,95, 69, 4)
img[,,,1] <- all$ire.sig
img[,,,2] <- all$sre.sig
img[,,,3] <- all$ire.d.sig
img[,,,4] <- all$sre.d.sig

nifti.set.filenames(img, paste("Patient_", patient, "_sig", sep=""))
nifti.image.write(img)

img[,,,1] <- all$ire.ts
img[,,,2] <- all$sre.ts
img[,,,3] <- all$ire.d.ts
img[,,,4] <- all$sre.d.ts

nifti.set.filenames(img, paste("Patient_", patient, "_ts", sep=""))
nifti.image.write(img)

img$dim <- c(79,95, 69, 4)
img[,,,1] <- all$ire.lag
img[,,,2] <- all$sre.lag
img[,,,3] <- all$ire.sig.lag
img[,,,4] <- all$sre.sig.lag

nifti.set.filenames(img, paste("Patient_", patient, "_lag", sep=""))
nifti.image.write(img)
}

```

10.2 Maps and Histograms for All Subjects

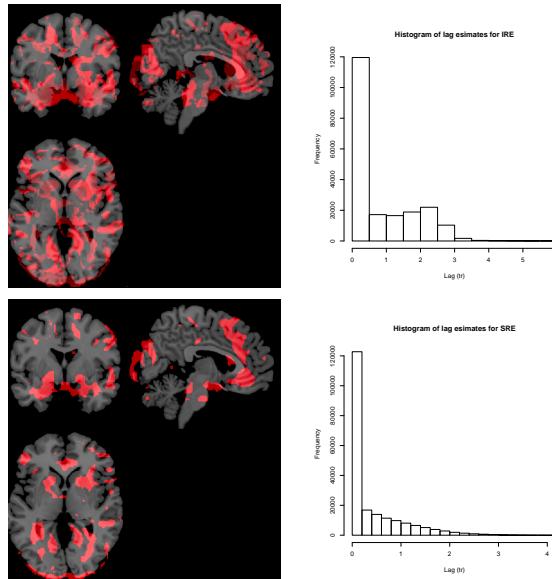


Figure 10.1: Spatial Maps and Histograms of Lags for Subject 9

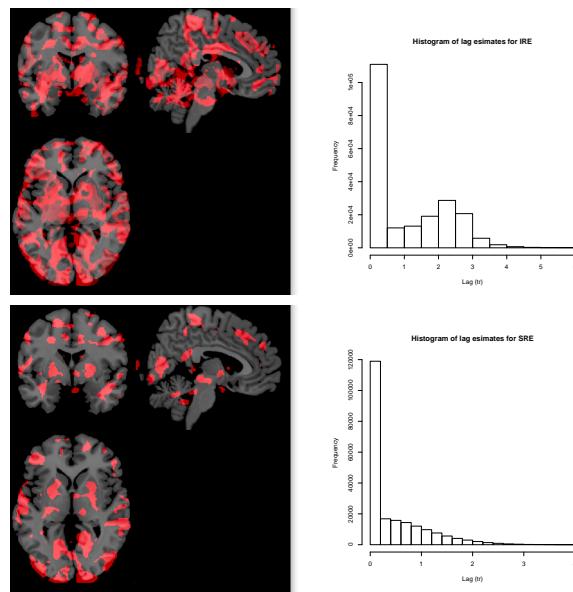


Figure 10.2: Spatial Maps and Histograms of Lags for Subject 10

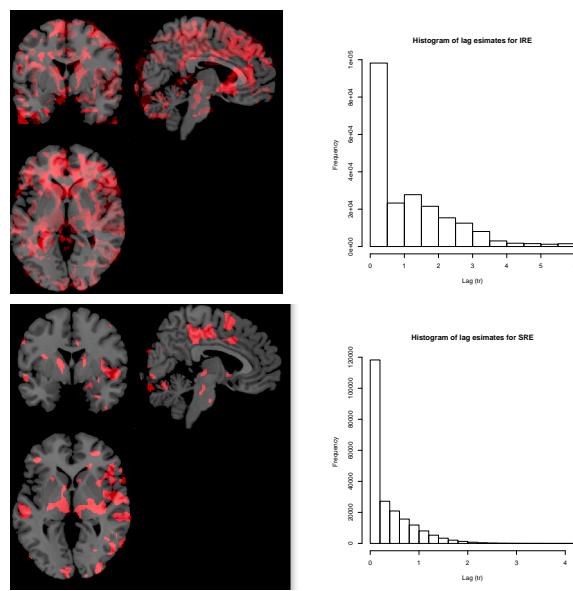


Figure 10.3: Spatial Maps and Histograms of Lags for Subject 11

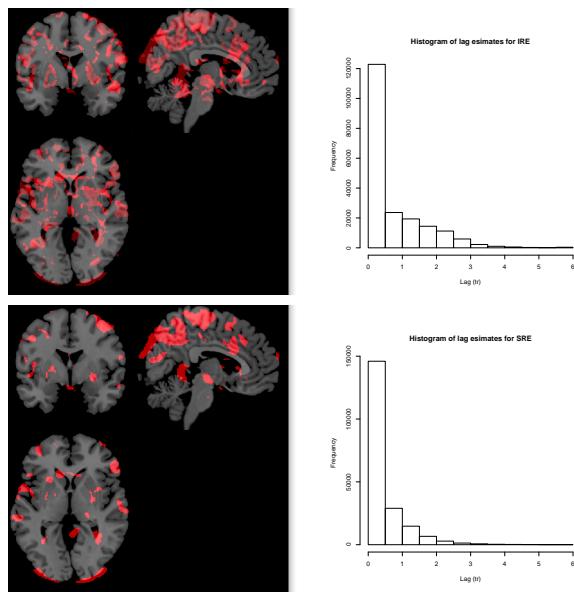


Figure 10.4: Spatial Maps and Histograms of Lags for Subject 12

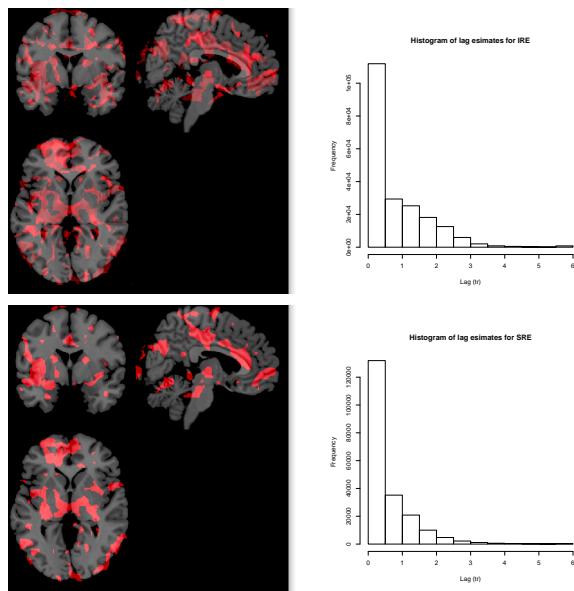


Figure 10.5: Spatial Maps and Histograms of Lags for Subject 14

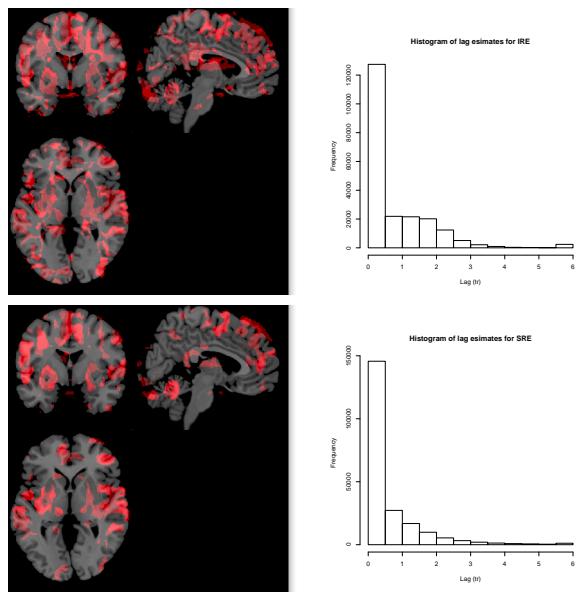


Figure 10.6: Spatial Maps and Histograms of Lags for Subject 15

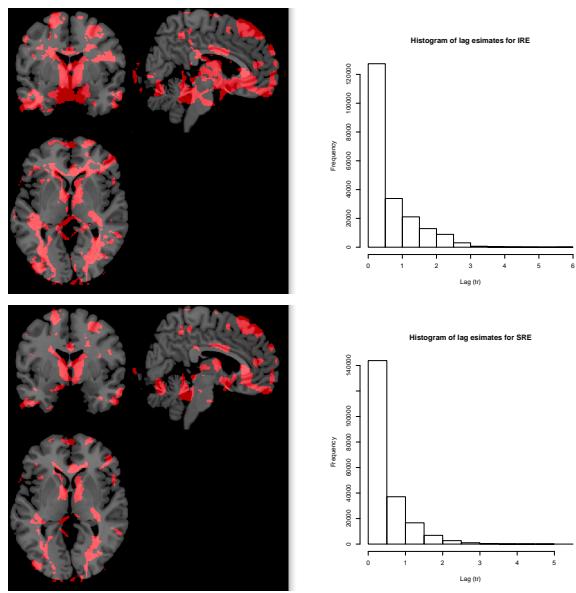


Figure 10.7: Spatial Maps and Histograms of Lags for Subject 16

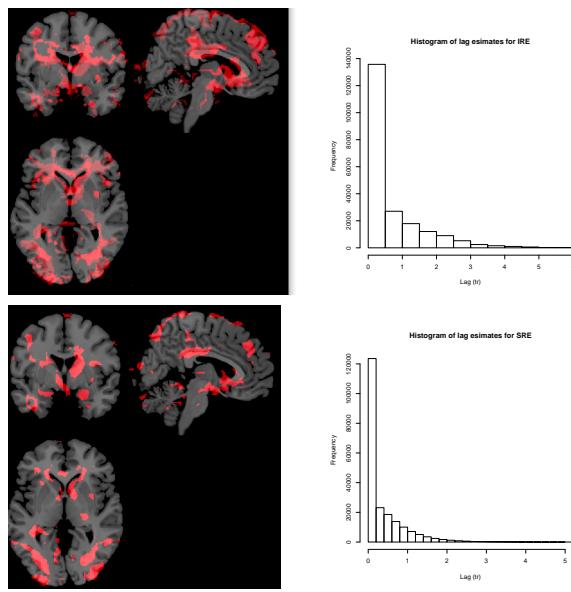


Figure 10.8: Spatial Maps and Histograms of Lags for Subject 17

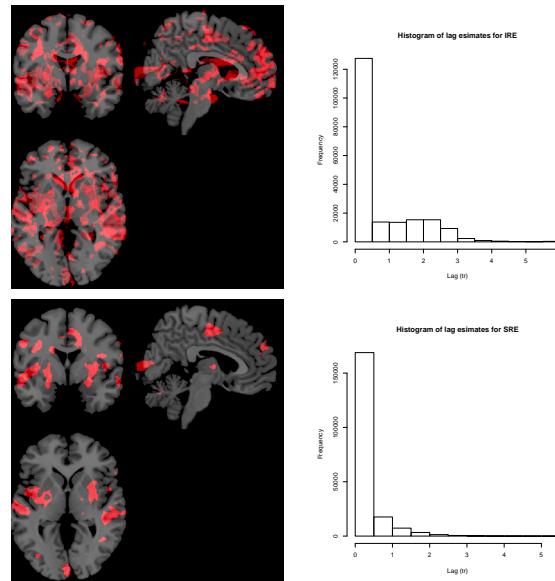


Figure 10.9: Spatial Maps and Histograms of Lags for Subject 18

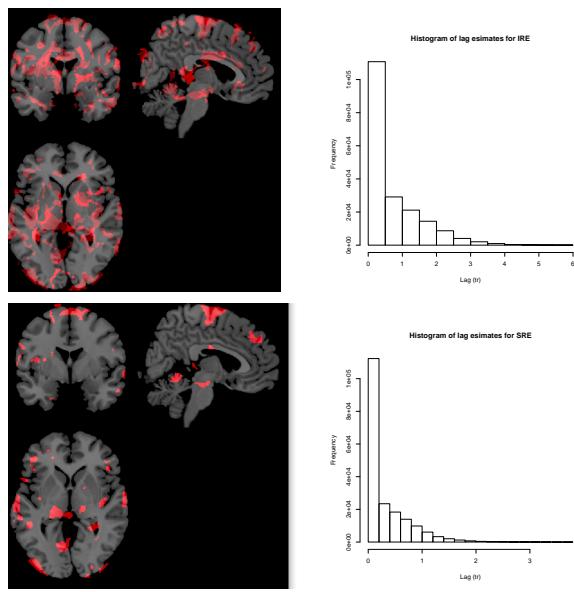


Figure 10.10: Spatial Maps and Histograms of Lags for Subject 19

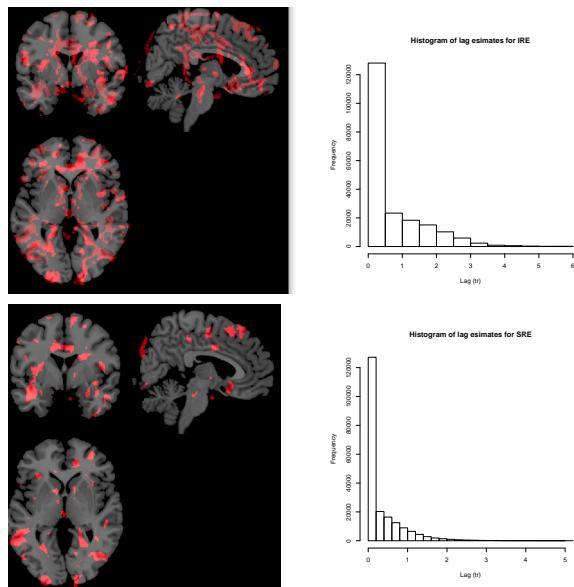


Figure 10.11: Spatial Maps and Histograms of Lags for Subject 20

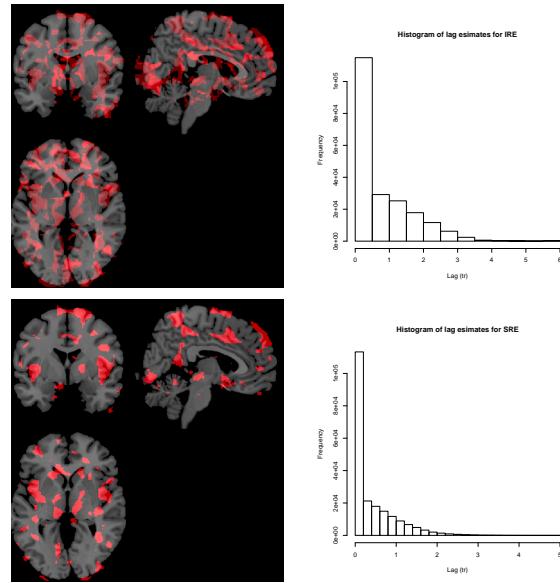


Figure 10.12: Spatial Maps and Histograms of Lags for Subject 21

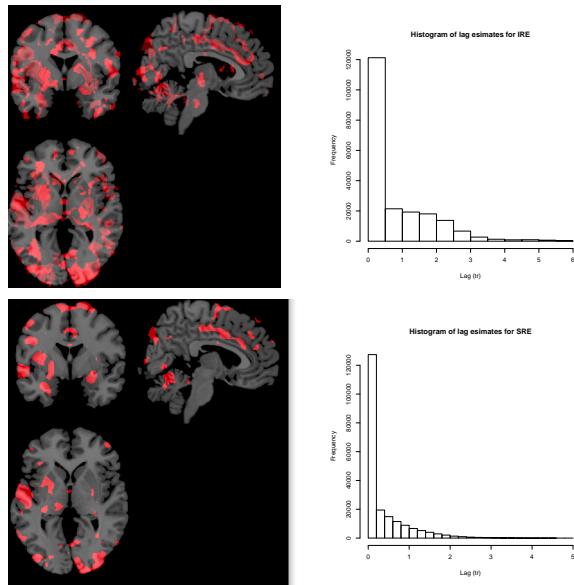


Figure 10.13: Spatial Maps and Histograms of Lags for Subject 22

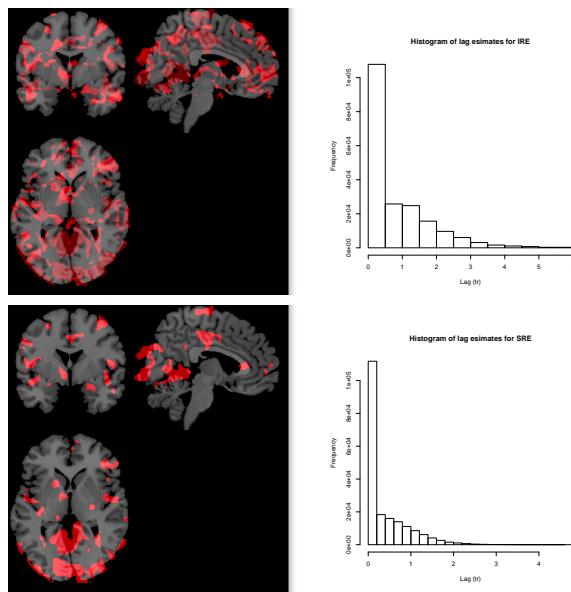


Figure 10.14: Spatial Maps and Histograms of Lags for Subject 23

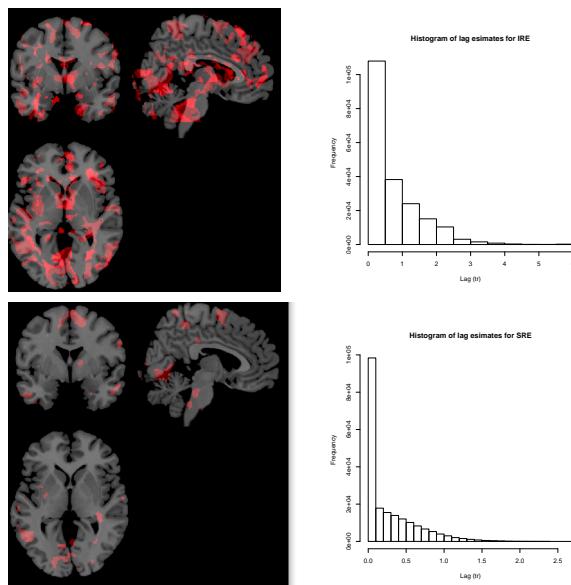


Figure 10.15: Spatial Maps and Histograms of Lags for Subject 24

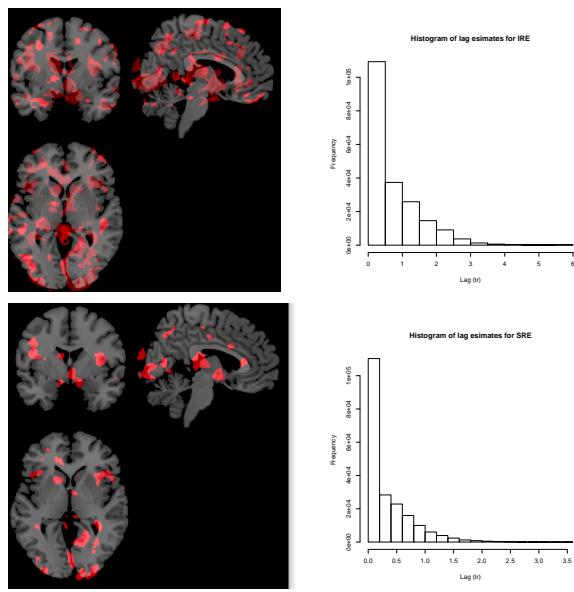


Figure 10.16: Spatial Maps and Histograms of Lags for Subject 25

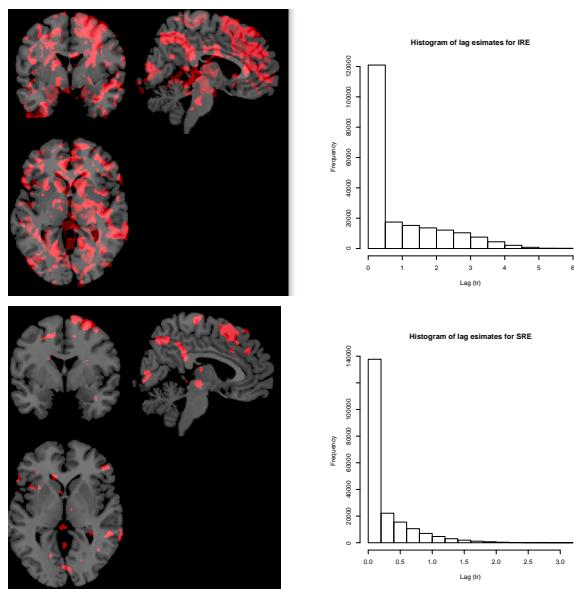


Figure 10.17: Spatial Maps and Histograms of Lags for Subject 26

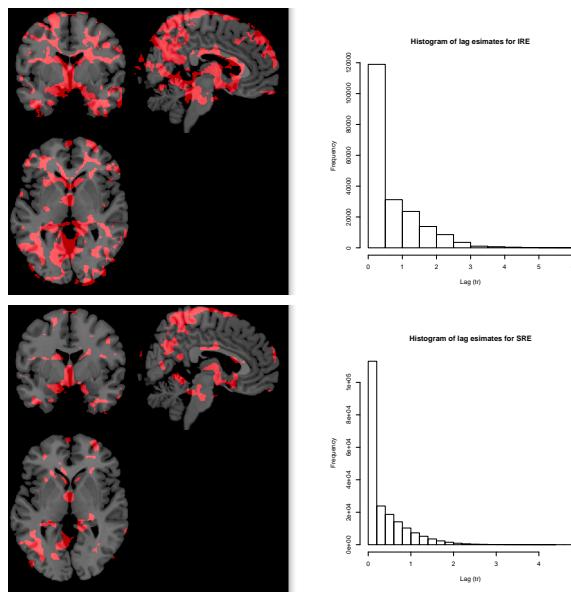


Figure 10.18: Spatial Maps and Histograms of Lags for Subject 27

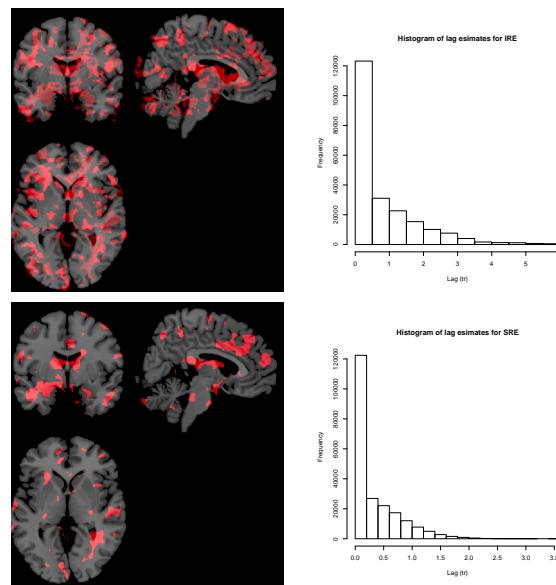


Figure 10.19: Spatial Maps and Histograms of Lags for Subject 28

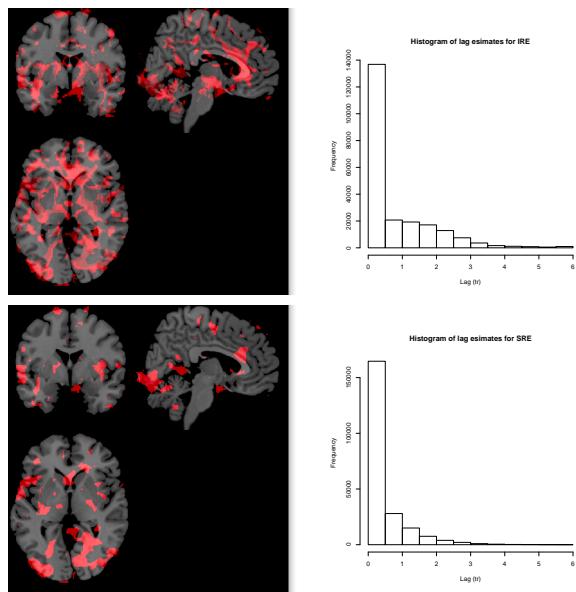


Figure 10.20: Spatial Maps and Histograms of Lags for Subject 29