Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Mingrui Liang Date

An exploration of time series models and their application to functional magnetic resonance imaging

By

Mingrui Liang
MSPH

Department of Biostatistics and Bioinformatics

_____

Benjamin Risk
Committee Chair

Ying Guo
Committee Member

An exploration of time series models and their application to functional magnetic resonance imaging

By

Mingrui Liang

Bachelor of Science
Sun Yat-Sen University
2016

Thesis Committee Chair: Benjamin Risk, PhD
Thesis Committee Member: Ying Guo, PhD

An abstract of
A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science in Public Health
in Department of Biostatistics and Bioinformatics
2018

Abstract

An exploration of time series models and their application to functional magnetic resonance imaging

By Mingrui Liang

The general linear model is a popular tool in functional magnetic resonance imaging (fMRI) data analysis. One of the major problems in fMRI data analysis is that the fMRI blood-dependent oxygen level dependent (BOLD) time course is serially correlated. Some of the mainstream neuroimaging software packages use overly simplified models of time series errors, such as AR(1), which can lead to invalid inference due to the in- ability to account for serial correlation. There has been renewed interested in this issue with recent developments in acquisition protocols leading to much shorter time to repetition (TR), or the time between acquisition of brain images. We compared different modeling methods in this article in order to explore the factors that contribute to inflated type I error rates in fMRI time series data analysis. We introduce the application of autoregressive moving average models (ARMA) to the analysis of single-subject fMRI data, where the order of the AR and MA components are chosen using Akaike's information criterion corrected for small sample size (AICc). Simulations were used to examine type I error rates. When the true model has an AR(1) structure, more flexible ARMA(p, q) models generally lowered the type one error rates relative to ordinary least squares (OLS) and the AR(6) model, but were often still higher than nominal levels. We also estimated spatially specific time series models for thirty subjects in a motor task from the Human Connectome Project, where control variables orthogonal to the conventional covariate matrix were introduced to gain insight into type one errors. The value of the autocorrelation function is downwardly biased when using OLS residuals, which would select the incorrect time series model. We also suggest that the length of the time series data and model complexity may affect the accuracy of inference.

An exploration of time series models and their application to functional magnetic resonance imaging

By

Mingrui Liang

Bachelor of Science
Sun Yat-Sen University
2016

Thesis Committee Chair: Benjamin Risk, PhD
Thesis Committee Member: Ying Guo, PhD

A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science in Public Health
in Department of Biostatistics and Bioinformatics
2018

Acknowledgements

I want to thank the faculty, advisors, and staff of the Biostatistics and Bioinformatics Department at Rollins School of Public Health for my unforgettable two years of study. I had a wonderful time here at Emory because of their patience, knowledge and caring. I would especially like to thank Dr. Benjamin Risk for all of his advice and support to help me write this thesis. Also, a special thanks to Dr. Ying Guo for taking the time to read my thesis. Lastly and most importantly, I would like thank my parents and friends for their care and support that motivated and encouraged me all the way.

Contents

1 Introduction

One of the tasks of neuroscience is to map the functions of brain regions. Functional magnetic resonance imaging (fMRI) is one of the important techniques related to this task, where the variation of the BOLD signal in the brain is captured in a rapid and sequential pattern[1]. The goal of fMRI data analysis is to see whether the BOLD signal changes in response to some outside manipulation in each brain voxel.

A popular tool to detect this variation is the general linear model (GLM), where the BOLD signal variation plays the role of the outcome variable. The body functions that we are interested in, such as the movement of a subject's left leg, are reflected by the independent variables in the model. However, because of the autocorrelation of the BOLD signal response, the inference using Ordinary Least Squares (OLS) estimate would be biased, considered that the assumption of the GLM is violated, where the data should be uncorrelated with equal variance. The source of the biased inference are the errors. Since the errors are not independent in time, although the estimate of the coefficients themselves are still unbiased, using OLS to estimate the parameters without considering the correlation structure, or failing to use a correct one, can cause bias when calculating the covariance of the coefficients, and hence inferences would be invalid.

One simple solution is to consider the first order autoregressive model, also known as the AR(1) structure, for the error term, where we assume the intervals of each scan in the scanning session are the same and the value from the previous scan impacts the current scan. The AR(1) model is able to capture the correlation between observations. The correlation between observations k time points apart is $\rho = \phi^k$, where ϕ is the correlation coefficient in the AR(1) model. Given the lag 1 observation, an observation is conditionally independent of other observations higher than lag 2 , which means that

$$cor(X_t, X_{t-k}|X_{t-1}) = 0, \qquad k = 2, 3, ..., t-1$$

This strategy can improve the validity of inferences for fMRI time series with repetition times (TR) above 2 seconds[2].

Similarly, a significant decrease of false positives is observed in the SPM software package, which uses a global AR(1) model in which the same autocorrelation is assumed for all locations[3]. However, Eklund et al. (2012) found the global AR(1) structure in SPM failed to adequately model the serial correlation in the residuals and hence caused a high error rate when analyzing a large fMRI dataset, particularly in datasets with short TRs[4].

There are several other software packages that offer alternatives to SPM, although little is known about their suitability for shorts TRs. In the FMRISTAT software package, Worsley et al.

(2002) propose a method where a spatially varying AR(p) structure is considered, which extends the AR(1) model to p previous time points [3]. There is also the autoregressive moving-average (ARMA) model structure. Specifically, the ARMA(1,1), which has one autoregressive parameter like in AR(1) with one additional moving average parameter, is applied in the software package AFNI[5]. Woolrich et al., in the FSL software package, choose to estimate a flexible AR model in each voxel, and then use a Tukey-Taper to smooth the model in the frequency domain[6].

Recent advances in simultaneous multislice (SMS) echo-planar imaging (EPI) have led to large decreases in the TR, where the TR for whole-brain fMRI can be less than one second[7]. These shorts TRs have led to concerns that the current methods, which were developed for traditional fMRI data with TRs of 2-3 seconds, may not adequately account for the serial correlation in the new sensorimotor rhythm (SMR) fMRI series.

In this paper, we used four different modeling methods on simulated and actual time series data, in order to find out factors that affect the validity of the inference of time series data. For the simulated time series data, we compared the type I error rates using different modeling methods and different covariate structures. For the actual time series data, we examined the autocorrelation function (ACF), partial autocorrelation function (PACF) and the model selection results in order to discover the actual time series structure. We also compared the type I error rates for actual data. Finally, activation plots were made to highlight the functional region on the cortical surface, also to check the bias of the inference.

2 ARMA Overview

The fundamental modeling technique that we applied in this article is the autoregressive moving-average (ARMA) model. As the name suggests, the model is a combination of an autoregressive (AR) part and a moving average (MA) part. The AR part predicts an outcome based on its previous values, whereas the MA part claims that the outcome depends on its current or past values of the noise term. An ARMA model with a set of covariates is written as:

$$Y_t - X_t^T\beta = \epsilon_t + \sum_{i=1}^{p} \phi_i (Y_{t-i} - X_{t-i}^T\beta) + \sum_{i=1}^{q} \theta_i \epsilon_{t-i},$$

where Y_{t-i} is the outcome response for time $t-i$, X_t are the covariates, β are their coefficients, ϵ_{t-i} is the white noise error term at time $t-i$, $\phi_1, ..., \phi_p$ are the AR coefficients, and $\theta_1, ..., \theta_q$ are the MA coefficients.

If we use the backshift notation B, where $B^l Y_t = Y_{t-l}$, then our model becomes:

$$(1 - \sum_{i=1}^{p} \phi_i B^i)(Y_t - X_t^T \beta) = (1 + \sum_{i=1}^{q} \theta_i B^i)\epsilon_t.$$

An ARMA(p, q) model refers to an ARMA model with p autoregressive terms plus q moving-average terms. A simplified version of the ARMA(p, q) model is the AR(p) model. It can be written as

$$Y_t - X_t^T \beta = \sum_{i=1}^{p} \phi_i B^i (Y_t - X_t^T \beta) + \epsilon_t.$$

The AR model assumes the partial autocorrelation function to be 0 after lag p, which captures long dependence in the autocorrelation function. On the other hand, the MA model has 0 dependence in the autocorrelation function after lag q, and captures long dependence in the partial autocorrelation function. Therefore a combination of these two helps to capture more complex serial correlations.

3 Simulation

3.1 Simulation Setting

We conducted a simulation study in order to explore 3 factors that may lead to invalid inference of the time series data using existing methods. The 3 factors are time series structure of our response variables, level of complexity of the covariate matrix that we regress against and the model structure that we choose to fit our data.

In order to explore the impact of different time series structure, we simulated 4 different sets of time series as outcome variables in our study, namely AR(1) with $\phi = 0, \phi = 0.4, \phi = 0.9$, and ARMA(2,2) with $\phi_1 = 0.9, \phi_2 = -0.4, \theta_1 = 0.9, \theta_2 = 0.4$. We generated 5000 sets of time series as our outcome variable for each setting, each with a length of 284 time points. For each setting of time series, we used 4 sets of different covariate matrix, namely 1) a 284×1 random Gaussian vector, 2) a structured control variable, 3) a 284×22 random Gaussian matrix, and 4) a covariate matrix from a block-design motor task of subject 105014 in the Human Connectome Project (HCP) plus the attached control variable. More detail about the variables in the covariate matrix is discussed in section 4.1.

The true coefficients were set equal to zero in all settings. In the first setting, we examine the validity of inference when the covariate has no structure or dependence. In the second setting, the purpose of creating this structured control variable is to demonstrate how the structure of the covariate impacts the type I error rate. The structured control variable is a fixed 284×1 vector which is orthogonal with the covariate matrix. It is used in the real data analysis where

it has no relationship with other covariates and it is expected to have no relationship with the response, hence the value of its coefficient should equal to 0. In the third and fourth setting, the purpose is to set up a way to measure the level of validity of inference when modeling with numerous covariates. For valid inference, the distribution of the p-values of the covariate we examine should be approximately uniform.

Finally, we used 4 different models to fit our simulated data, which are the OLS, AR(1), AR(6) and the auto.arima() function in R. We applied the auto.arima() function here to return the best ARIMA model according to AICc after searching over possible ARMA(p, q) models[8].

3.2 Simulation Results

We summarize the simulation results using histograms to illustrate the distribution of the p-values for the coefficient of the regressors of interest as we described above. Results are categorized according to the setting of the covariate matrix. In the figures below, each column represents different setting of time series structure of the data, and each row represents different models that we use to fit the data.

3.2.1 Random Gaussian vector

As we can see from the result (Figure 1) using a random Gaussian vector as a regressor, the only severe issue appeared when the time series data has an ARMA(2,2) structure. The AR(1) model is able to capture the lag 1 partial correlation but fails to capture the lag 2 partial correlation. Since the AR(2) coefficient is negative, overall the model overestimates the variance of the coefficient, which lead to an overly conservative type I error rate.

When using a random Gaussian covariate, the general type I error rate is similar to our nominal error rate 0.05. The overall p-value distribution is close to uniform distribution.

3.2.2 Structured control variable

In the second result (Figure 2), an obvious issue of inflated type I error rate was observed when using OLS to fit our time series data. For the ARMA(2,2) outcome structure, the AR(1) model showed less severe conservative type I error rate, which appears to be due to the particular form of the structured covariate matrix. Overall, our type I error rates are around the nominal error rate for AR(1), AR(6) and auto.arima() models, but severely inflated in OLS. Note that the AR(1) structure was closest to 0.05 for $\phi = 0$, $\phi = 0.4$, and $\phi = 0.9$, auto.arima() was the second closest, and AR(6) tended to have more inflated type I error rate.

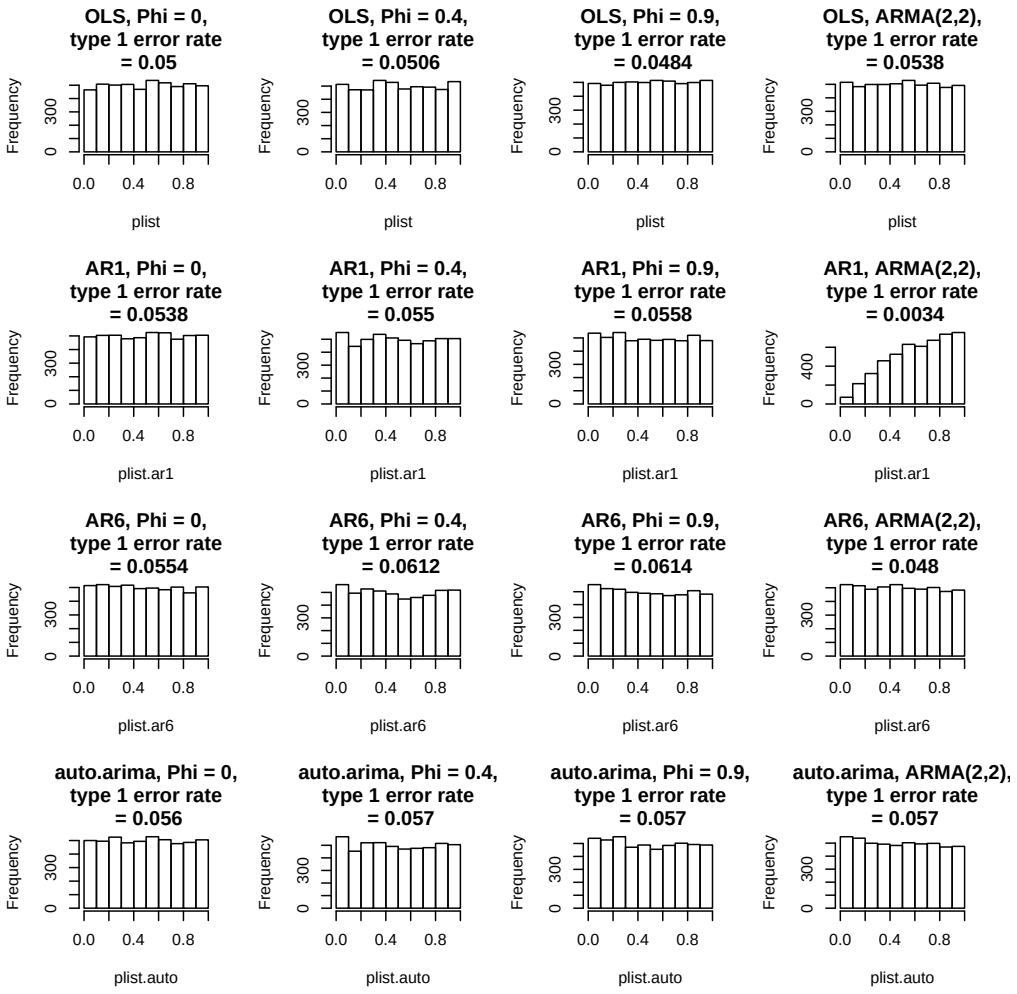

Figure 1: Simulation results using univariate regression with a random Gaussian vector as the covariate.

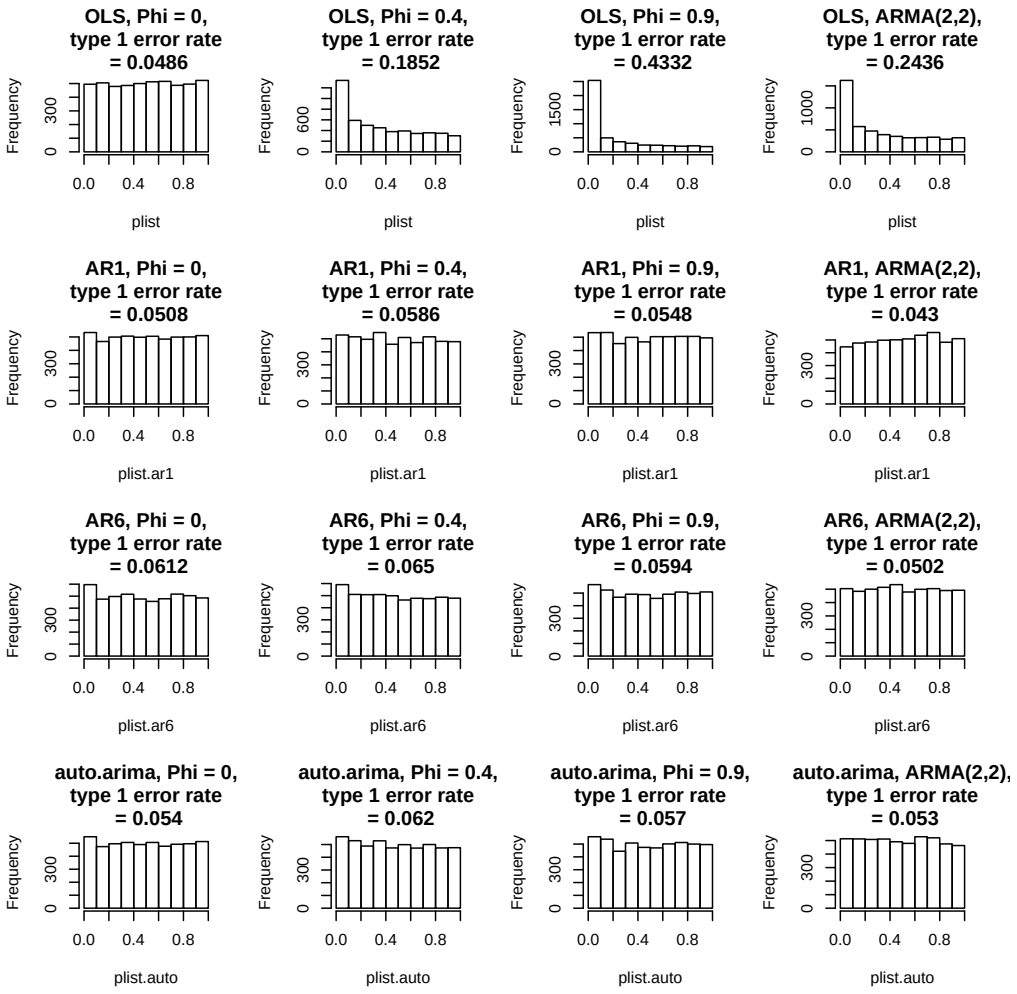

Figure 2: Simulation results using univariate regression with a structured control variable as the covariate.

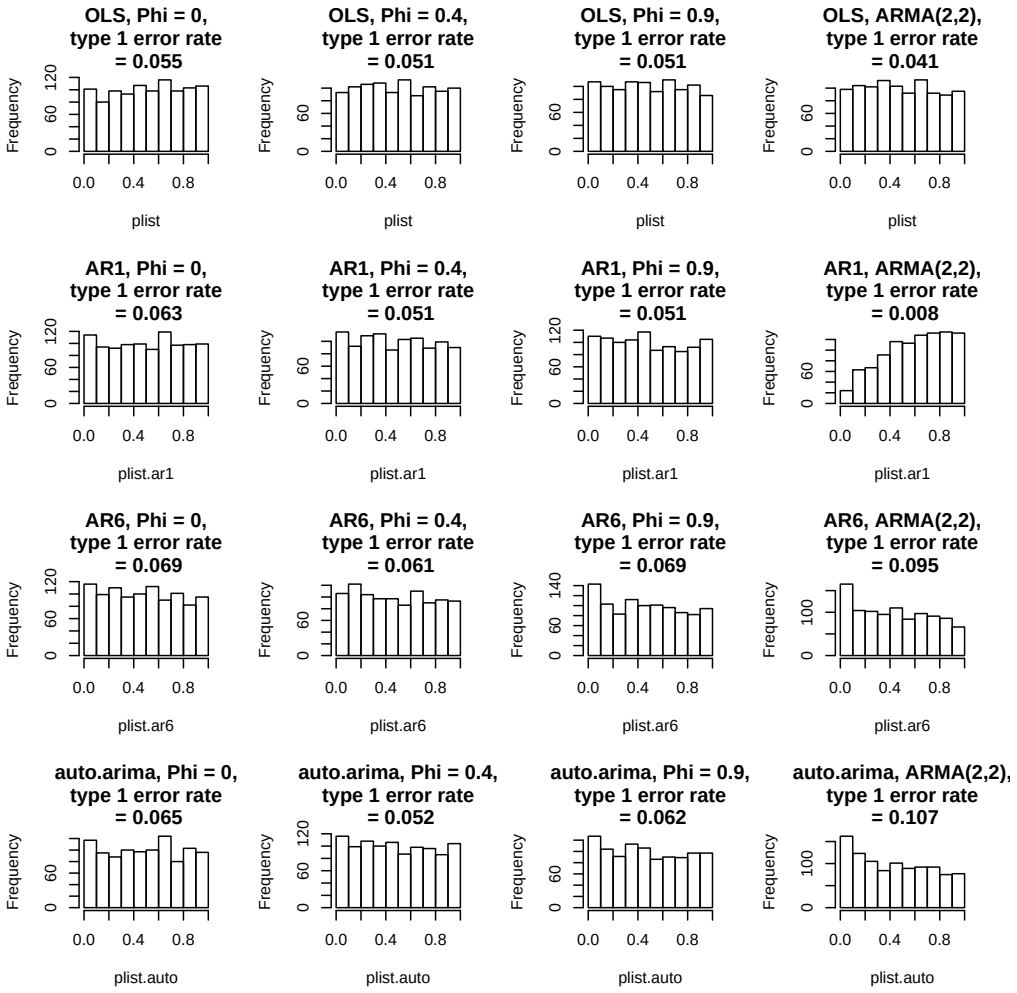

Figure 3: Simulation results using multiple regression with a random Gaussian matrix with twenty-two covariates.

3.2.3 Random Gaussian matrix

In our third scenario (Figure 3), the type I error rate increase for AR(1) data was observed when using the AR(1), AR(6) and auto.arima() models. The maximum likelihood estimation (MLE) method was used here to estimate the test statistics. The standard error used in Wald statistics are underestimated when including many covariates and a relatively short time series.

Surprisingly, when the covariates are independent and identically Gaussian distributed, OLS works the best although the outcome variable has a time series structure because the standard error is approximately unbiased (see Risk et al. 2016, formula S.3). Moreover, inference is based on a T-distribution which accounts for a finite sample size and the number of covariates.

For ARMA(2,2) data, we can also see the overly conservative type I error rate for AR(1) model. OLS is close to the nominal level, following by AR(6). auto.arima() has the highest type I error rate.

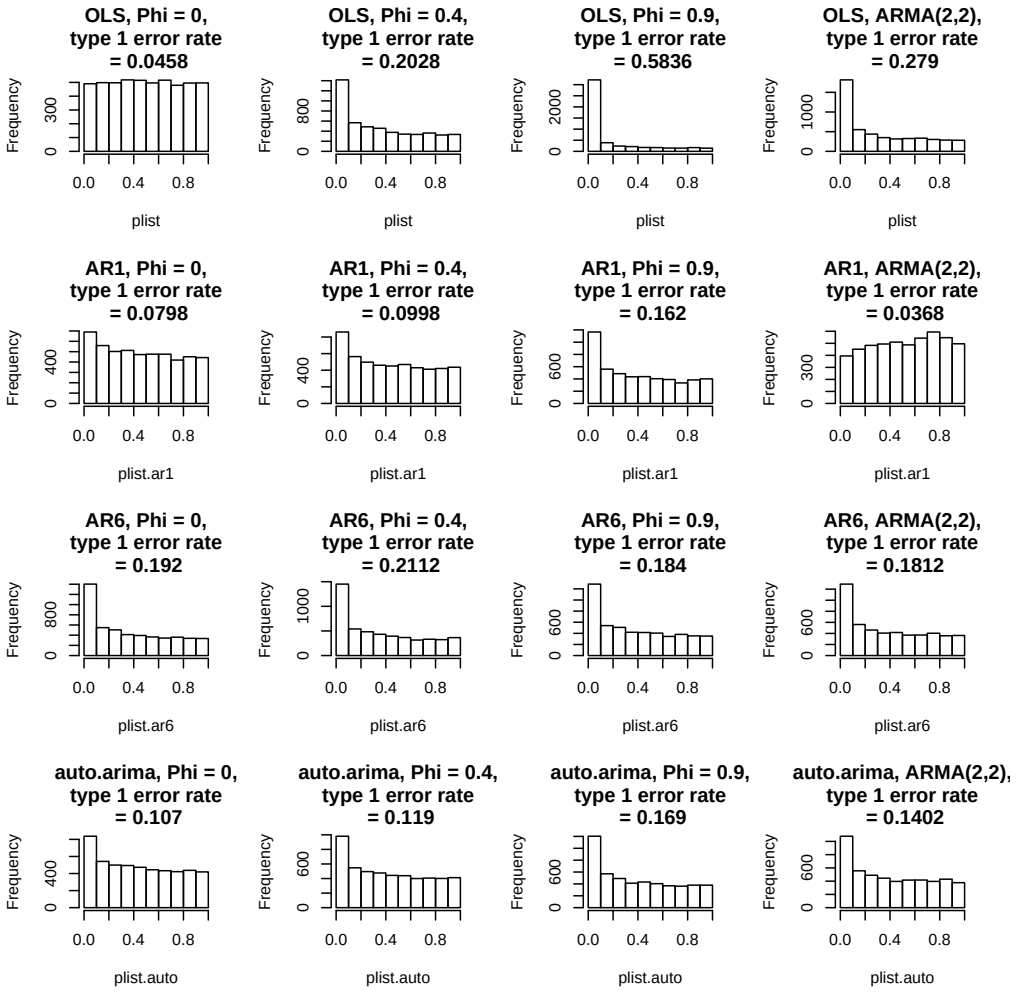

Figure 4: Simulation results using multiple regression with twenty-two covariates based on a block-design motor task; p-values for the structured control covariate shown.

3.2.4 Motor-task covariate matrix with control variable

In this covariate matrix setting (Figure 4), a severe inflated type I error rate was observed when using the OLS to fit our simulated AR(1) data. This inflation happened because the variance of our parameter of interest is underestimated. Also, as ϕ increased, the number of independent observations decreased, which caused the expansion of the effect of model over-specification and small sample size when using AR(1) to fit our data. When using the AR(6) model, more parameters were estimated, which led to more serious overspecification problems and severely inflated type 1 error rates. The type I error rate when using auto.arima() was better than using AR(6) model but worse than using AR(1) model because auto.arima() selected flexible models every time, but generally the model it selected would be more complex than the AR(1) model.

For ARMA(2,2) data, we can also see the overly conservative type I error rate in AR(1) model. auto.arima() works the best. AR(6) and OLS have higher type I error rates.

To sum up the result of our simulation study, OLS has an inflated type I error rate when the covariate is structured, and AR(1) has a incorrect type I error rate modeling ARMA(2,2) data. auto.arima(), on the other hand, is the most robust modeling approach when there is a single structured covariate, however the type I error rate can be inflated when there are a large number of covariates.

4 Methods

4.1 The Human Connectome Project

The data we used in our analysis is from a motor task involving thirty subjects in the Human Connectome Project (HCP). The HCP is a project to map human brain structure and function within single individuals or across subjects. They provide a resource for large amounts of publicly available unprocessed and preprocessed data using multiple imaging methods combined together with a large amount of behavioral and genetic data [9].

In the HCP motor task, experimental subjects were required to move their tongue to specific areas, squeeze their toes, or tap their fingers according to the visual cues presented to them [10]. Two sessions of scans with 284 time points, $2 \times 2 \times 2$ mm voxels, and 0.72 s repetition time were performed. Locations corresponding to the right cortex were extracted as in the paper by Risk et al. (2016) [11], resulting in 29,716 time series for one session. The vertex indices of the cortical surface mesh correspond to locations matched across subjects [12]. In this experiment, we used thirty subjects from the HCP. Among the 2 sessions, we chose the first (right-left) session of the motor task for analysis. Linear and quadratic time terms were used to capture scanner drift, and they were used as regressors in our analysis together with an intercept, the six task parameters described above (cue, left and right fingers, left and right toes, tongue), their temporal derivatives, and the six parameters from the rigid body motion correction, resulting in twenty-one covariates. Also, we set up a structured control variable that is orthogonal to the design matrix as suggested in Eklund's paper [4]. This control variable is independent to the rest of the regressors in the design matrix, hence we can use it as an method to measure the type I error rate – its theoretical coefficient should equal to 0.

4.2 Experiment Setting

We performed an analysis where we used four modeling methods (OLS, AR(1), AR(6), auto.arima() function in R) to fit our HCP data for thirty subjects.

For each fitting process, we measured the value of the ACF and PACF for each modeling methods. ACFs and PACFs are essential for examining the structure of time series models. We captured the ACFs and PACFs from lag 1 to lag 5 for the OLS residuals, AR1 model, AR6 model and the flexible model chosen by the auto.arima() function. The ACFs and PACFs were measured for all 30 subjects, and we also analyzed the average of the ACFs and PACFs among these subjects.

We were also interested in the corresponding ARMA model that the auto.arima() function chose at every location. The results for 30 subjects were recorded. We also recorded the model

selection result when we used auto.arima() to fit the averaged response time series across the 30 subjects, while the averaged motion measurement and their derivatives were used as regressors. Moreover, we recorded the type I error rate using these four modeling methods on all 30 subjects. The type I error rate using these four modeling methods were also saved on the averaged responses and covariates. The type I error rate was calculated as the percentage of the p-values of the control variables that were less than 0.05 .

Finally, we obtained the test statistics for the contrast variable, namely the measurement of left finger tapping versus the average of five other motion parameters, as well as for the control variable. Using the test statistics, we can highlight the area of the brain that is responsible for the motion of the left fingers by making an activation plot, also called a statistical parametric map. We can also visualize the map of our test statistics for the control variable mapped on the brain.

For the purpose of display, we only included the ACF-PACF result of the subject that has the median type I error rate (subject 100307), and the result for the averaged ACF and PACF. For the activation plot, we chose the subject that has the median and the worst type I error rate (subject 149741).

5 Results

5.1 ACFs And PACFs

The ACFs and PACFs provide insight into the time series structure. For a single subject (Figure 5), we plotted the PACF and ACF resulting from the fitted ARMA(p, q) model selected using auto.arima() (Figure 5(a)) and the sample PACF and ACF based on the OLS residuals (Figure 5(b)). In the upper plot, the grey area shows how the auto.arima() function selected no time series structure in some of the locations. Both auto.arima() and OLS show lag 1 correlations in similar areas. However, auto.arima() was still able to capture more lag 1 correlation than OLS if we look at the mean of the ACF (0.0328 vs 0.03), the variance of the ACF (0.0971 vs 0.0874), and the maximum value of the ACF (0.846 vs 0.62). In general, OLS estimates of the ACF and PACF are downwardly biased[3]. This suggests that pre-whitening using estimates of the PACF from the OLS residuals, which has been suggested elsewhere[13] may underestimate the temporal dependence, which could inflate type 1 error rates and also select the incorrect time series model. Also, the level of correlation decreased for lag greater than 2.

For average values of the ACFs and PACFs (Figure 6), the spatial patterns are more apparent, and the differences between the OLS and auto.arima() estimates are more prominent. In general, correlations estimated using the auto.arima() function (Figure 6(a)) were higher than using OLS

(Figure 6(b)). The negative bias caused by using the residuals from OLS to estimate the ACF and PACF is particularly prominent in lags three and four, which show extensive negative values in the OLS figures but not the auto.arima() figures. Plus, we observed particularly prominent correlation in the inferior parietal lobule, the juncture of medial orbito-frontal and the superior frontal lobule, and the lateral occipital lobule. We also saw obvious correlation in the middle and inferior temporal lobule.

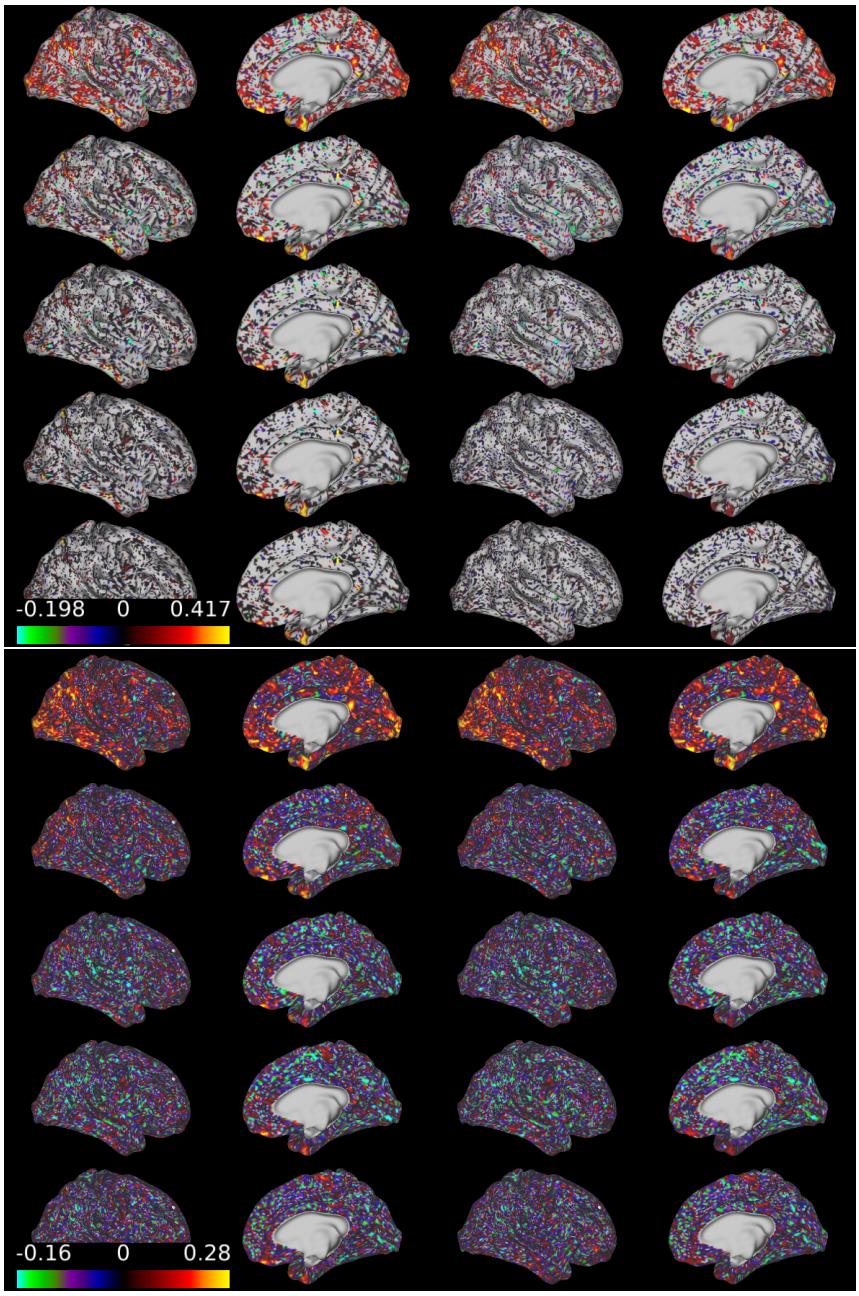

Figure 5: ACFs (2 plots on the left) and PACFs (2 plots on the right) of 100307, lag 1 to lag 5 (from top to bottom). (a) The upper plot: used the auto.arima() function in R, (b) the lower plot: PACF and ACF from OLS residuals.

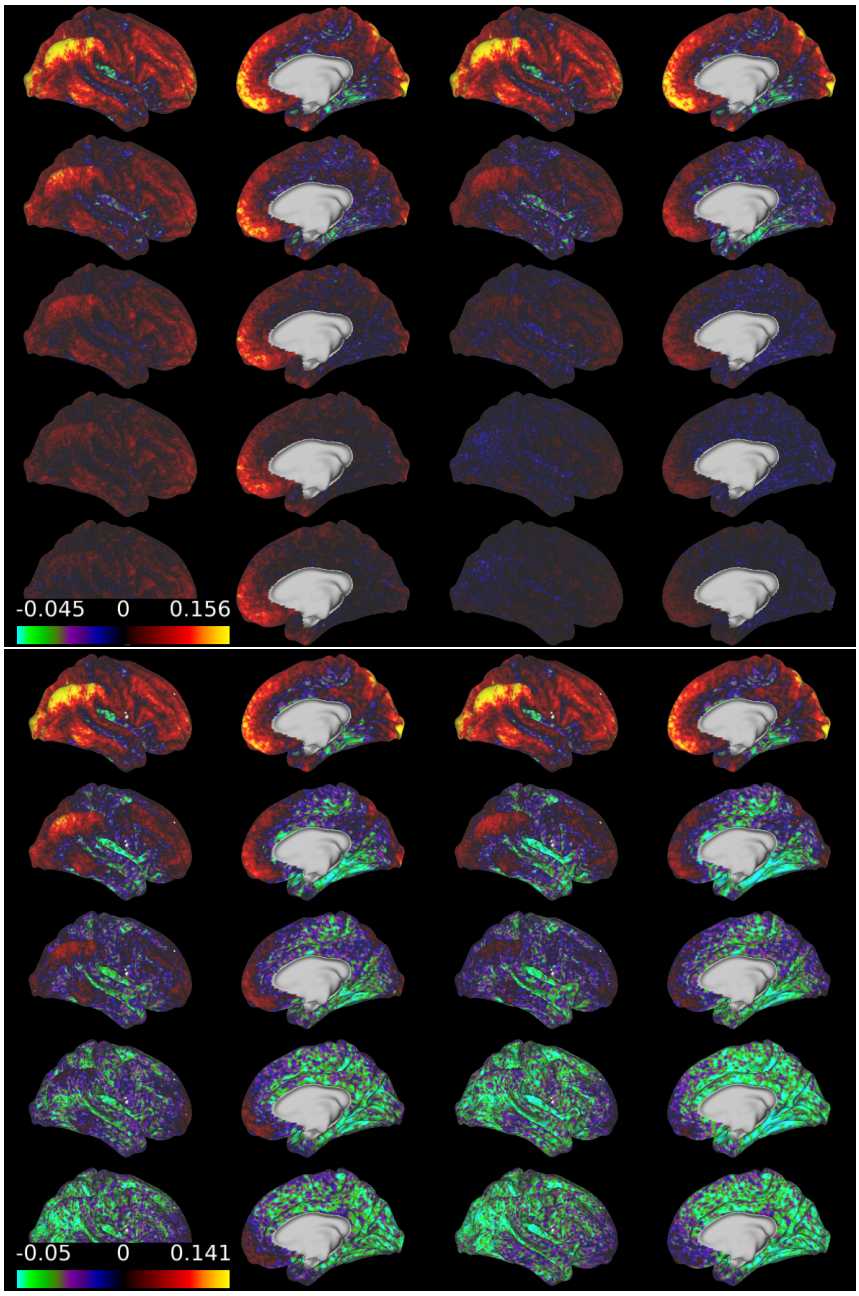

Figure 6: Averaged ACFs (2 plots on the left) and PACFs (2 plots on the right), lag 1 to lag 5 (from top to bottom). (a) The upper plot: used the auto.arima() function in R, (b) the lower plot: PACF and ACF from OLS residuals.

5.2 Model Selection

The function auto.arima() selected a variety of models. For single subjects, ARMA(0,0) was selected most often, followed by ARMA(1,0), ARMA(0,1) and ARMA(1,1). It is worthwhile to mention that some of the more complicated models, such as ARMA(2,2), were also observed in the result of model selection over 1000 times.

For averaged response and covariate matrix, the most common structure was ARMA(1,0) (selected 9340 times), followed by ARMA(0,0) (selected 7958 times), ARMA(2,0) (selected 2806 times) and ARMA(1,1) (selected 2253 times).

Table 1: Result of model selection by the auto.arima() function

Subjects	ARMA(0,0)	ARMA(0,1)	ARMA(0,2)	ARMA(0,3)	ARMA(1,0)
Averaged	7958	2079	308	71	9340
100307	15127	2945	586	193	4575
100408	12903	3017	556	141	4632
101107	16452	2820	1031	275	3326
103818	15511	3291	835	273	4499
105014	15855	2757	775	253	3745
105115	13684	2315	908	177	4588
111716	13615	2285	832	275	5083
113619	14560	2736	734	219	4087
118528	16303	3038	1536	250	3053
118730	15514	2836	749	255	5012
123925	15675	3096	996	240	3670
124422	14831	2898	886	243	4528
128632	15844	2840	901	300	4392
129028	15795	2479	1219	350	3760
133928	11352	2705	727	138	5655
135225	15736	2825	856	241	4948
144832	14015	3215	645	162	4723
146432	15168	2655	960	242	3320
149741	12033	2628	565	125	7546
151223	14181	2775	867	240	5215
159340	16343	2827	1865	615	3505
160123	14710	2552	921	176	4314
188347	15695	3060	975	226	3496
189450	16440	3066	906	251	4459
201111	14584	3135	929	253	4073
208226	9082	2531	368	68	4224
239944	14118	3099	795	181	4670
245333	12093	2829	824	247	3704
499566	14052	2643	870	164	3792
654754	12246	2789	585	117	5425

ARMA(1,1)	ARMA(1,2)	ARMA(2,0)	ARMA(2,1)	ARMA(2,2)	ARMA(2,3)
2253	566	2806	964	1462	166
1854	358	1267	516	1144	119
2309	448	1817	817	1341	138
1665	314	1176	468	1074	118
976	235	1224	393	1333	99
1730	378	1376	441	1235	133
2079	435	2032	742	1210	159
1716	404	2045	633	1077	118
2010	454	1555	501	1322	141
1053	283	1179	455	1236	146
1103	187	1480	348	1071	95
1505	380	1282	407	1208	152
1486	342	1430	473	1312	123
1003	342	1526	352	1178	103
1370	312	1531	437	977	135
2150	747	2276	522	1542	218
1022	279	1407	395	988	80
1821	474	1647	552	1196	131
1938	397	1301	562	1471	178
1390	351	2224	537	1125	126
1235	319	1666	410	1276	127
838	235	1356	348	742	89
2030	378	1520	615	1245	141
1882	377	1263	488	1199	117
849	283	1312	336	1028	65
1894	530	1213	640	1150	142
5490	981	1884	1561	1413	159
1562	367	1904	524	1211	128
3013	747	1731	670	1739	236
1853	749	1694	665	1696	211
2699	566	1890	692	1374	142

ARMA(3,0)	ARMA(3,1)	ARMA(3,2)	ARMA(3,3)	ARMA(4,0)	others
642	235	260	104	196	306
309	157	183	72	79	232
419	249	265	67	154	443
350	123	212	46	74	192
319	81	190	57	145	255
361	99	216	69	79	214
484	146	251	60	150	296
548	194	229	46	228	388
398	170	265	75	114	375
367	99	288	43	122	265
414	95	160	29	151	217
335	123	262	66	76	243
338	110	287	71	99	259
269	100	184	61	89	232
487	147	273	43	105	296
437	248	413	118	103	365
335	85	147	47	110	215
370	159	197	62	113	234
362	187	421	78	94	382
379	121	199	59	110	198
545	120	288	64	150	238
332	90	147	21	128	235
319	138	270	57	89	241
259	123	233	52	47	224
225	66	132	35	106	157
255	178	249	68	88	335
477	390	323	121	164	480
400	137	198	70	142	210
499	243	438	140	129	434
306	208	319	96	67	331
400	187	231	79	72	222

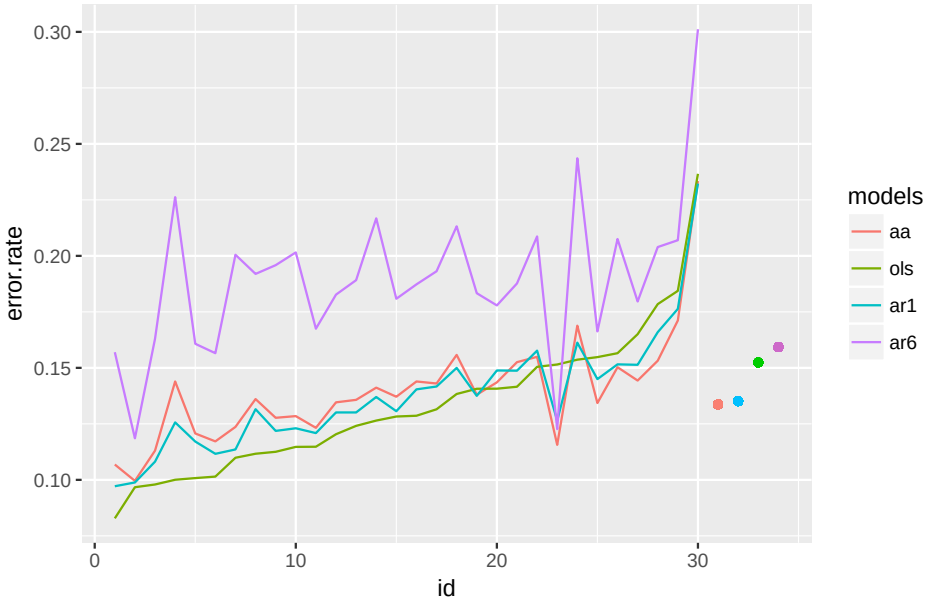

Figure 7: Type I error rate for 30 subjects (line) and the averaged time series (dot)

5.3 Type I Error Rates

As shown in the result (Figure 7), the type I error rates were pretty close when using OLS, AR(1) and auto.arima(). OLS performed slightly better than other methods for about 20 subjects. The type I error rates when using AR(1) and the auto.arima() function were similar with each other, and the type I error rates when using AR(6) were the biggest. This result can be explained because the auto.arima() selected no time series structure in around half of the locations, and OLS has the lowest type I error rate when there is no time series structure, particularly when there are many covariates. On the other hand, the model selection showed that there are more time series structure for the averaged outcome and covariates, due to the lower measurement errors. Here we observed that the type I error rate was the smallest when using the auto.arima() function, following by AR(1), OLS and AR(6). However, overall the type I error rates were substantially higher than the nominal 0.05 rate.

5.4 Activation Plots

The activation plots (Figure 8) for contrast variables included areas associated with the primary motor cortex. We chose to show the results using OLS and the auto.arima() function. This highlight was obvious when we looked at the subject that has a median OLS type I error rate (subject 100307). The area of the highlight was more vague for the subject that has the worst OLS type I error rate (subject 149741).

For the control variables, the ideal statistics on the plot should be asymptotically normally distributed around 0 with no spatial structure. However, our results, especially for subject

149741, showed that the statistics drifted from 0, which indicated bias. Although we also used the auto.arima() method, the bias was not fixed. We didn't see much improvement in type I error rate probably because we were modeling with too many covariates (22 covariates in total) and each time series is relatively short (284 time points).

6 Discussion

In this study, we explored the effect of different modeling methods on time series data using both simulations and an actual data analysis. We observed that failing to use the correct model leads to inflated type I error rate and biased inference. This might be a problem for some current mainstream neuroimaging software. A global AR(1) structure is used in SPM, and AFNI uses an ARMA(1,1) structure. They are overly simplified models, hence their inference might be inaccurate. FSL uses an alternative modeling approach for inference, which is beyond the scope of this study.

In the simulation study, although it wasn't very obvious when using one random Gaussian vector as regressor, difference was noticed between different modeling methods when using the structured control variable as a regressor. The AR(1) model performed the best when the time series itself is simulated with an AR(1) structure. The auto.arima() function, followed by the AR(6) model, had the second and third lowest type I error rate. OLS provided a severely biased result, especially when the correlation was increasing. This trend was also observed when using the motor-task covariate matrix. Surprisingly, type 1 error rates using the MLE estimates were inflated when using twenty-two covariates, even when the AR(1) model was estimated and the true generating process followed the AR(1) model. This may be because we have a large number of covariates and a relatively short time series.

In the actual data analysis, we first examined the ACFs and PACFs when using different models. We saw that ACFs and PACFs for individual subjects tend to be noisier than the averaged ACFs and PACFs. Besides, PACF plots were observed to drop to zero faster than the ACF plots. The reason is that our model selection result suggested that the AR structure is common, and theoretically, for an AR structure, it is more likely to see zero values in PACF plots than in ACF plots. For example in an AR(1) structure, started from ϕ, the correlation would drop to $\phi^2, \phi^3, ...$ as the lag increases. The PACF, on the other hand, would drop to zero for lag greater than 1. At the same time, it can also be observed that for some locations of the cortical surface the partial correlation still exists after lag 1, which indicates a more complicated time series structure than AR(1) in the actual data set. This result also matched with the outcome of the model selection process, where in general more than one third of the locations

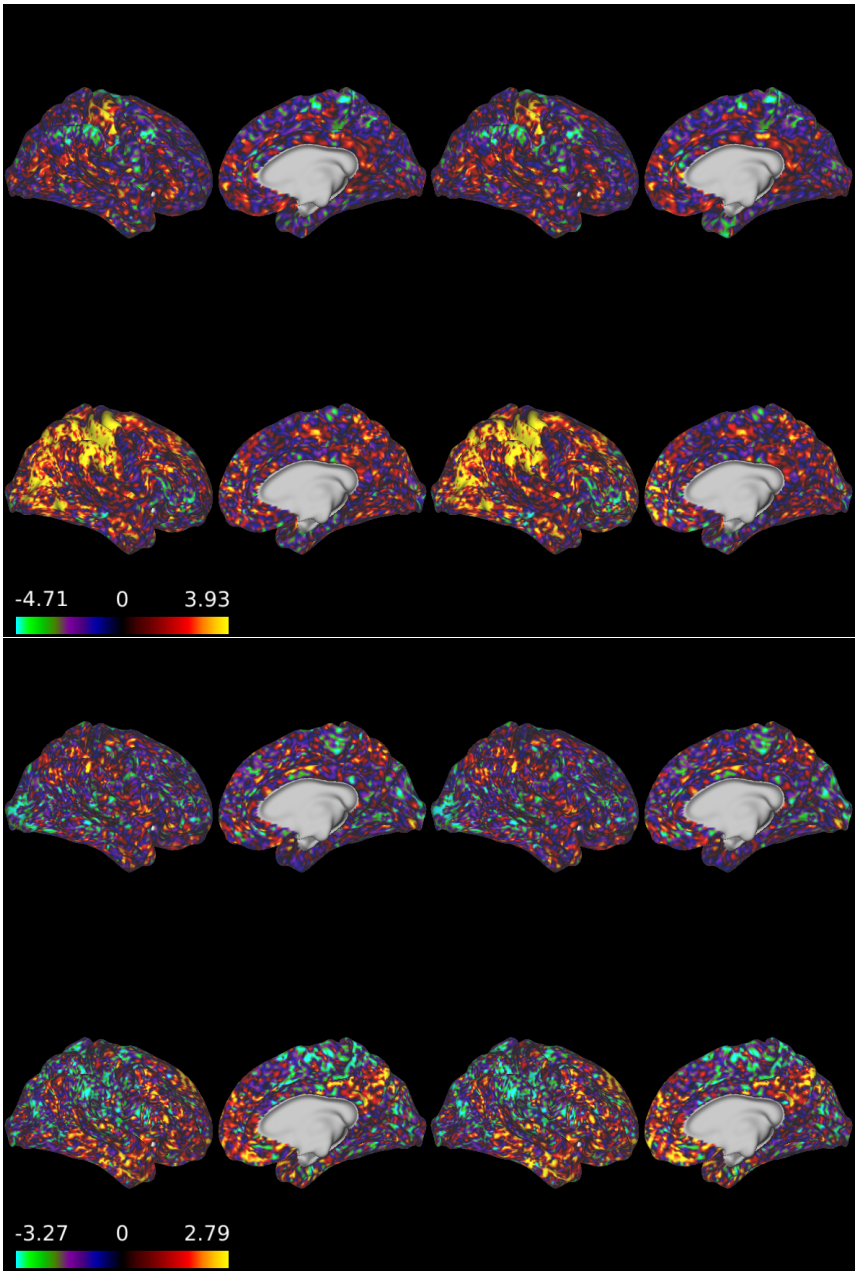

Figure 8: Activation plot for contrast variable for subject that has a median OLS type I error rate (subject 100307) and worst type I error rate (subject 149741) (2 upper plots) and control variable for subject 100307 and 149741 (2 lower plots) using OLS and the auto.arima() function.

have a structure that is more complex than an AR(1) structure. Additionally, the value of the autocorrelation function is downwardly biased when using OLS residuals, which would select the incorrect time series model.

There are a number of shortcoming of this study that should be noted. Even though auto.arima() has the ability to choose the most suitable time series model for each location, and it might be considered a very good method to solve the problems in inference created by serial correlation, it has its own limitations. A obvious drawback of auto.arima() is that the method it uses for estimation relies on the asymptotic theory, which might bring inaccuracy when the sample size is not large enough, or the response time series is not long enough. Also, auto.arima() is not going to bring a large amount of improvement in accuracy when there are too many variables in the covariate matrix. The performance of auto.arima() under such a situation is only slightly better than OLS, but the cost for such little improvement is that the auto.arima() process is over 30 times slower than the OLS model fitting. The auto.arima() method took almost 12 hours to finish our analysis process for all 30 subjects on a 40-core computing cluster, where the OLS only took 10 minutes on a 72-core computing cluster. Additionally, there is no guarantee that our structured variable has no relationships with the response variable.

References

[1] S.A. Huettel, A.W. Song, and G. McCarthy. *Functional Magnetic Resonance Imaging*. Oxford University Press, Incorporated, 2009.

[2] Torben E Lund, Kristoffer H Madsen, Karam Sidaros, Wen-Lin Luo, and Thomas E Nichols. Non-white noise in fmri: does modelling have an impact? *Neuroimage*, 29(1):54–66, 2006.

[3] Keith J Worsley, CH Liao, J Aston, V Petre, GH Duncan, F Morales, and AC Evans. A general statistical analysis for fmri data. *Neuroimage*, 15(1):1–15, 2002.

[4] Anders Eklund, Mats Andersson, Camilla Josephson, Magnus Johannesson, and Hans Knutsson. Does parametric fmri analysis with spm yield valid results?—an empirical study of 1484 rest datasets. *NeuroImage*, 61(3):565–578, 2012.

[5] Robert W Cox. Afni: software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical research*, 29(3):162–173, 1996.

[6] Mark W Woolrich, Brian D Ripley, Michael Brady, and Stephen M Smith. Temporal autocorrelation in univariate linear modeling of fmri data. *Neuroimage*, 14(6):1370–1386, 2001.

[7] David A Feinberg, Steen Moeller, Stephen M Smith, Edward Auerbach, Sudhir Ramanna, Matt F Glasser, Karla L Miller, Kamil Ugurbil, and Essa Yacoub. Multiplexed echo planar imaging for sub-second whole brain fmri and fast diffusion imaging. *PloS one*, 5(12):e15710, 2010.

[8] Rob J Hyndman, Mitchell O'Hara-Wild, Christoph Bergmeir, Slava Razbash, Earo Wang, and Maintainer Rob Hyndman. Package 'forecast'. 2017.

[9] David C Van Essen, Kamil Ugurbil, E Auerbach, D Barch, TEJ Behrens, R Bucholz, Acer Chang, Liyong Chen, Maurizio Corbetta, Sandra W Curtiss, et al. The human connectome project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231, 2012.

[10] Deanna M Barch, Gregory C Burgess, Michael P Harms, Steven E Petersen, Bradley L Schlaggar, Maurizio Corbetta, Matthew F Glasser, Sandra Curtiss, Sachin Dixit, Cindy Feldt, et al. Function in the human connectome: task-fmri and individual differences in behavior. *Neuroimage*, 80:169–189, 2013.

[11] Benjamin B Risk, David S Matteson, R Nathan Spreng, and David Ruppert. Spatiotemporal mixed modeling of multi-subject task fmri via method of moments. *NeuroImage*, 142:280–292, 2016.

[12] Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.

[13] Hernando Ombao, Martin Lindquist, Wesley Thompson, and John Aston. *Handbook of Neuroimaging Data Analysis*. CRC Press, 2016.