University of Technology Sydney (UTS)
School of Mathematical and Physical Sciences
Faculty of Science

Aggregation in Regression Analysis of Very Large Time Series Datasets

by

Alan Malecki

A thesis submitted
in partial fulfilment of the
requirements for the degree

Doctor of Philosophy

Sydney, Australia

Certificate of original ownership

I, Alan Malecki declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Science, at the University of Technology Sydney (UTS).

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training program.

Signature:

Production Note:
Signature removed prior to publication.

Date: 30/04/2020

Acknowledgement

This thesis and my life as a statistician would not be possible without my supervisor Louise Ryan. You have moulded me into the researcher and person I am now. You took me on as an honours student, showed me the exciting possibilities of statistics, and developed an interest that cannot be satisfied. You have made this journey an enjoyable and rich experience, and furthermore, you have made an immeasurable contribution to my life. When we began this journey, I never thought I would have such an incredible mentor and friend. Scott Sisson, my co-supervisor, I have truly appreciated the time you have given me. It has been a pleasure coming to your office and discussing my thesis, as well as many other topics. To my partner, Georgia, who supported me at all times, and gave me hope and joy throughout, we've done it together. Lastly, I thank my parents, who have given me everything. Nothing would have been possible without your love and support.

Contents

List of Figures

List of Tables

Abstract

The focus of this thesis is on the analysis of large and complex data. Computer memory constraints can prohibit the analysis of large datasets, and this issue is further complicated when faced with complex data. We are motivated by an environmental dataset concerning air particulate measurements and the impact of passing coal transport trains. This dataset has over 600,000 observations and is complicated by it's long memory dependence. Current methods for long memory time series are limited to small datasets. To overcome these issues, we consider two approaches for the analysis of large and complex data:

1. transforming data such that its volume and complexity is reduced, and,
2. extending current statistical methods for big data to allow for complex data structures.

The use of temporal aggregation transforms the dataset to a more manageable size. This permits the use of an AutoRegressive Fractionally Integrated Moving Average (ARFIMA) process on our motivating dataset. We also consider transforming the data to a bivariate series to reduce the loss of information due to this temporal aggregation.

Divide and Recombine is a modern approach to analysing big data. This approach for big data analysis has not yet been extended to the time series setting. We explore this situation and extend the D&R process for long memory time series.

1 Introduction

Technological advances in the capture and storage of data have lead to larger and larger datasets. The analysis of such big datasets is aided by increasing computer power and availability. Theses advances have not been matched at the same rate by statistical methods. Therefore to effectively harness the power of big data, statistical theory must also be advanced. In a statistical analysis, it is no longer uncommon to be confronted with a dataset that is too large to be stored on a single computer. Software such as Hadoop[55] and MapReduce[12] are often utilised in these cases, where the data is stored on a number of seperate computers and analysed separately. However, such methods can be ineffective for complex data, particularly in the time series domain.

The focus of this thesis is to develop methods that will allow the analysis of big and complex data. We consider two approaches to this issue:

1. transforming data such that its volume and complexity is reduced, and,
2. extending current statistical methods for big data to allow for complex data structures.

This thesis is motivated by an environmental dataset that underpins all of our work. This particular dataset concerns the impact of passing trains on air particulate levels in the town of Newcastle, Australia. Initially our work with this dataset was an issue of public policy. Citizens in the town of Newcastle were worried that coal transport by rail was increasing air pollution levels. The Environmental Protection Authority (EPA), conducted a study with the goal of determining the impact of passing trains on air particulate levels. The analysis of this dataset has endured many difficulties. It is large, as it contains over 600,000 observations, and its time series dependence constrains its analysis.

Time series data are often analysed using AutoRegressive Integrated Moving Average (ARIMA) models with differencing order d, where $d \in 0, 1, 2$. The differencing parameter is applied to obtain a stationary and invertible ARMA process. This process has been extended to the case where d is no longer an integer, under the guise of an AutoRegressive Fractionally Integrated Moving Average (ARFIMA) process. The need for this extension arose from the discovery of long memory dependence in certain time series datasets. Long memory dependence occurs when autocorrelations are non-summable. These long range correlations can be small, yet are important for correct statistical inference. As a result, it can occur that observations in long memory data are dependent for large distances between events. Currently the use of the ARFIMA model is limited to datasets of a small size. This is due to the computational demands of calculating the inverse of the covariance matrix which has a Toeplitz structure. A standard computer such as used by this author, is unable to analyse ARFIMA models for datasets larger than 12 to 15 thousand observations. Thus, not only for our dataset, but as available datasets become bigger, there is a significant need for improving the current models for data with long memory dependence.

This thesis consists of five chapters, each of which aims to improve the ability to analyse big and complex data. In chapter 2 we provide a detailed overview of the Hunter Valley Coal train dataset that is used throughout this thesis. We also present some previous analyses conducted on this data, as well as outlining the effect of temporal aggregation. In chapter 3, we consider temporal aggregation as a method to transform the data such that it's size and complexity is reduced, and we are able to fit a regression with ARFIMA errors to the aggregated data. Our results in chapter 3 suggest that we may have a misspecified model. Thus in chapter 4, we explore this effect. We hypothesize that there is a tail effect after a train has passed the air particulate monitor and we explore the possibility of this omitted variable. We are able to show that there is such an effect, and we suggest a method to determine the tail effect through a simulation study and the coal train application. Returning to the approach of transforming data such that it can be analysed using existing methods, in chapter 5 we continue with our study of temporal aggregation. However, it is clear that an issue with temporal aggregation is that there is a loss of information as a result of the data transformation. This information loss could be reduced by analysing two of the air particulate level time series concurrently. We utilise a hierarchical log likelihood to analyse this bivariate series. This accounts for the autocorrelation and cross-correlation in the two series. We also consider aggregating the data into a bivariate series of means and standard deviations, as a method to reduce this loss of information. While this strategy reduces the information loss due to temporal aggregation, it is extremely computationally demanding, and is limited to Autoregressive AR(1) processes and is therefore unable consider the long memory dependence in the data.

This leads us to our second approach for the analysis of big and complex data, the extension of current statistical methods. A relatively recent development in computation and statistics is the Divide and Recombine (D&R) process. This has been implemented in Hadoop and MapReduce. While successfully applied to big data, this approach suffers for time series data. In chapter 6 we explore how to use the D&R process for big time series data. While allowing us to analyse the unaggregated data, we face two issues. Firstly, by dividing the data into subsets, we lose information between subsets, and secondly, the D&R process for times series data is troubled by the difficulty of fitting time series models, which increases the chance of overfitting the models. To account for the loss of information between subsets, we compare the results of fitting the model to different subset lengths, which results in consistent outcomes. This subset length selection is further improved by dividing the data such that the same error structure of the full data is retained in each subset. We conclude this chapter with a comparison of our results for each modelling strategy used on the coal train dataset, as well as the computational timings of each model. To our best knowledge, our implementation of the D&R process on an ARFIMA model with a Toeplitz structured covariance matrix is the first of its kind.

2 Hunter Valley Coal Train Dataset and Previous Analyses

2.1 Introduction

The motivation behind this thesis has been an analysis of an environmental dataset. The size and complexity of this dataset, have limited its analysis. In this chapter we provide a thorough overview of its features. The local community of Newcastle, Australia was concerned with the passing of uncovered loaded coal trains through their neighbourhoods, to the local port, which is the world's largest coal export port. The fear of adverse health effects from this coal transport pollution was communicated to the NSW Environmental Protection Authority (EPA). The EPA and the Australian Rail Track Corporation (ARTC) then conducted a study on the air particulate levels associated with passing trains in the Newcastle region. The goal of this study was to determine:

1. Whether trains operating on the Hunter Valley rail network are associated with elevated particulate matter concentrations; and
2. Whether trains loaded with coal have a stronger association compared with unloaded coal trains or other trains on the network.

The goal of this chapter is to introduce the reader of this thesis to the coal train dataset used throughout this thesis. We firstly present a detailed description of the data. We follow this description with a review of the previous analyses conducted on this dataset. These have been conducted by a private company, and by Professors Ryan, Wand and myself. We conclude this chapter with a review of some of the difficulties faced by these analyses, which have lead to our methods in chapters 3 to 6 of this thesis.

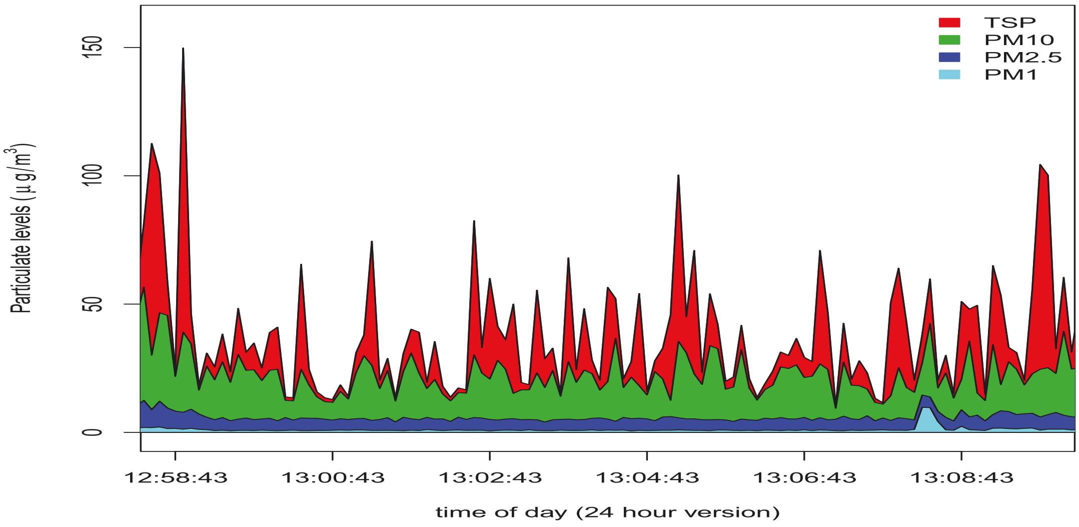
2.2 Hunter Valley Coal Train dataset

To analyse the impact of passing trains on air particulate levels, an OSIRIS air pollution monitor was placed beside the rail tracks at a location in Metford, a nearby town to Newcastle, along the rail corridor between the coal mines and the port. A severe issue with this study was that the monitor is placed in a single location. It would be preferable to utilise a number of monitors to reduce the bias that can result of only one recording location.

The OSIRIS monitor records particulate measurements (in micrograms per cubic metre) in three size groups. Particulate Matter (PM) of less than 1, 2.5, and 19 micrometers in diameter, and an aggregate sum as Total Suspended Particulate (TSP). The monitor also recorded if a train was present or not, as well as the train type and how fast it was travelling. These observations were recorded at a 6 second interval. The length of the study was 61 days, however there were times with missing observations. We thus restricted our analysis to days which had over 1,000 observations. This resulted in 55 days for our analysis, corresponding to slightly more than 600,000 observations in total.

In figure 2.1, we present a ten minute section on the first day of the study, of the four air particulate measurements, PM1, PM2.5, PM10 and TSP. The data, especially for TSP, are very jumpy and prone to extremes. For this reason we considered a log transform (with a logarithm of base e) on the data, which we utilised for all of our following analyses in this thesis.

Figure 2.1: Recorded levels of air particulates PM1, PM2.5, PM10 and TSP for a ten minute period on 30 November 2012.



There were 5,601 individual trains that passed the monitor during the study period, with a median of 137 per day. As we can see from table 2.1, there is some variation in the train types. The passenger trains were the most frequent, with 3576 passing the monitor during the study. They were also the fastest, with a median speed of 24.9 m/s. In comparison, the loaded coal trains were the slowest. This is to be expected due to their increased weight and length over the other train types. In the case that a passing object, whether it was a train or possibly a maintenance vehicle, was not identified, then it was recorded as unknown.

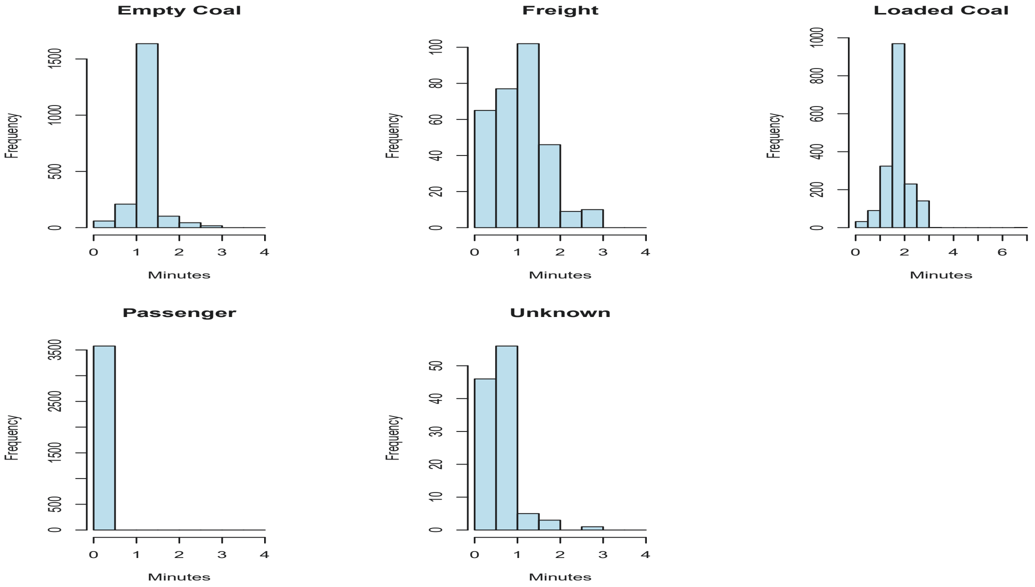
Table 2.1: Information about trains passing during the study period. Columns show median value, along with lower and upper quartiles (LQ, UQ).

Train Type	Trains per Day	Speed (m/s)	Number of Trains
	Median(LQ,UQ)	Median(LQ,UQ)	Total
Empty Coal	40(27,47)	19.8(18.5,20.7)	2066
Freight	6(2,8)	18.2(14.7,28.5)	309
Loaded Coal	36(24,41)	13.9(11.7,15.1)	1788
Passenger	60(55,92)	24.9(24.0,28.9)	3576
Unknown	3(2,3)	19.1(15.7,21.0)	111

The duration of a passing train was of key interest to our analysis. In figure 2.2 we show the distribution of the time it took each train type to pass the monitor. The passenger trains were the quickest and of shortest length, as a consequence they passed the monitor almost instantaneously.

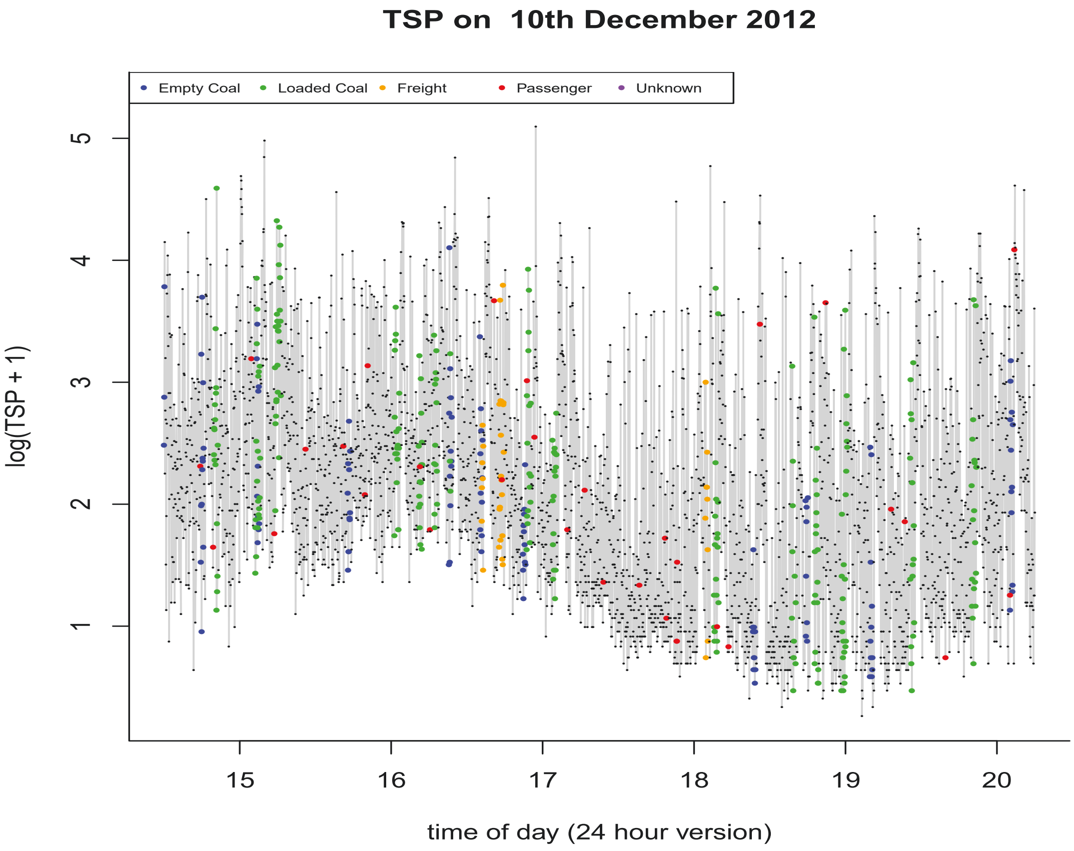
The empty coal trains took on average 1 minute to pass the monitor, while the loaded coal usually took 2 minutes. There were a few loaded coal trains that took up to 7 minutes to pass the monitor. The freight trains had the most varied distribution, with 1-2 minutes the most frequent passing time.

Figure 2.2: Time taken to pass the monitor for each train type (in minutes).



The data is highly complex, with a number of passing trains and variation in the air particle measurements. In figure 2.3, we present a 6 hour period of the log transformed TSP observations, coupled with passing train indicators. We have selected this subsection as it shows a number of features of the data. From figure 2.3 we can see that the data is highly correlated, and that there is a clear day/night effect. Here the mean air particulate levels shift downwards at 5pm. These are a few of the features of this data that have complicated our analyses.

Figure 2.3: Log transformed TSP data during a 6-hour period on 9 December 2012. Plotting symbols are colour-coded to indicate the presence of various train types.



We now consider some previous analyses of this dataset.

2.3 Previous analyses

This study was conducted by the EPA and ARTC, to determine:

1. Whether trains operating on the Hunter Valley rail network are associated with elevated particulate matter concentrations; and
2. Whether trains loaded with coal have a stronger association compared with unloaded coal trains or other trains on the network.

An initial analysis done by a private company was then reviewed by an independent peer review.

The outcome of this review was that there had been some methodological and analytical issues with this analysis. A re-analysis of this study was then conducted by Professors Ryan and Wand [52], and by Professor Ryan and myself in [41] and [53]. We present the methodologies and findings of these analyses here.

2.3.1 Initial analysis by a private company

The initial analysis of this study was conducted by a private company. Their process was to aggregate the air particulate measurements while each passing train type was passing the monitor and then compare these to the aggregate of particulate measurements when there were no trains passing the monitor.

For each particulate measurement PM1, PM2.5, PM10 and TSP, they compared the upper and lower 95% confidence limits for the mean of each train type against the time when no train was passing. The findings of this method were that there was a statistically significant difference for empty and loaded coal trains and the period of no passing trains for the TSP measures. For the smaller particulate measurements there were some train types that were found to be significantly different from the background mean of no train passing.

The methodology of this analysis was flawed. The use of aggregation of all data points by train type or no train passing does not account for issues such as autocorrelation in the data, differences in each individual train or time of day differences. This analysis ignored many of the features of this dataset. Based on a independent peer review, it was suggested that a re-analysis be conducted.

2.3.2 Re-analyses of the the initial study.

Professors Ryan and Wand [52] and Professor Ryan and myself[41] and [53], implemented a generalized additive model (GAM)[66] from the 'mgcv' package[65] for the analyses of the coal train dataset. Utilising this advanced version of a linear regression overcame some of the issues ignored in the previous analysis. As shown in figure 2.3, there can be a day/night effect for the air particulate measurements, we account for this by including a smooth spline function for the time of the day. There is also indications that the data has a strong serial autocorrelation. A smooth spline function for the day of the study can account for this effect, and is included in our model.

We ran a number of exploratory analyses for the coal train data, and uncovered that the period after a train had passed was associated with an increase in air particulate levels. This effect can be considered as caused by the stirring up of air particles due to a passing train. It would be negligible to assume that the moment a train has passed that it's effect is complete. Anyone who has stood on a train platform when a train has been passing by, can be sure to feel a wind effect in the period after the train has passed. To determine the length of time to consider for the passing effect, we ran the model (1) for a number of feasible values, and picked the values that had the best fit according to the Akaike Information Criterion (AIC)[1]. This resulted in our use of the period of 5 minutes after each train had passed as another indicator variable. In our model (1), we set this to be the variable "EmptyCoalAfter" for the empty coal trains, and for all the other train types in same manner.

$$
\begin{aligned}
y_n =& \beta_0 + \beta_1 EmptyCoal_n + \beta_2 EmptyCoalAfter_n + \beta_4 Freight_n + \beta_5 FreightAfter_n \\
&+ \beta_6 LoadedCoal_n + \beta_7 LoadedCoalAfter_n + \beta_8 Passenger_n + \beta_9 PassengerAfter_n + \\
&\beta_{10} Unknown_n + \beta_{11} UnknownAfter_n + f_1(timeOfDay_n) + f_2(studyDayNumber_n) + \epsilon_n,
\end{aligned} \tag{1}
$$

where $n = 1, \ldots, N$ and $\epsilon_n \sim N(0, \sigma_\epsilon^2)$.

Unfortunately, the fit of model (1) in the 'mgcv' package in R would take a number of hours to work. Furthermore, we were unable to fit the extension of the gam model that adjusts for autocorrelation, as it was too computationally intensive. Thus, to remedy the strong autocorrelation effect in the data, we implemented a block bootstrap with 50 bootstrap samples. This maintains the dependence structure of the data, which we achieved by resampling by days. The use of bootstrapping adjusts the standard errors in our models, reducing the impact of autocorrelation. The results of these analyses are in table 2.2.

Table 2.2: Results of the generalized additive model from equation (1) for logged TSP values.

Variable	Estimate	Standard Error	t-value	p-value
Intercept	3.508	0.080	43.795	<0.001
Empty Coal	0.090	0.017	5.339	<0.001
Empty Coal within 5 min	0.113	0.012	9.675	<0.001
Freight	0.100	0.037	2.729	0.006
Freight within 5 min	0.100	0.030	3.316	0.001
Loaded Coal	0.073	0.017	4.196	<0.001
Loaded Coal within 5 min	0.081	0.015	5.387	<0.001
Passenger	0.049	0.018	2.694	0.007
Passenger within 5 min	0.044	0.010	4.542	<0.001
Unknown	0.124	0.041	3.044	0.002
Unknown within 5 min	0.074	0.044	1.710	0.087

The results in table 2.2 indicate that there is no significant difference between the coal (loaded and empty) and freight trains on TSP levels. The passenger trains account for about half of the pollution that the coal and freight trains. Furthermore, the period after each passing trains, selected to be 5 minutes, accounts for a similar increase in pollution as their respective train. Analysis of the other air pollution measurements, namely PM10, PM2.5 and PM1, all produced similar results. These can be found in the technical report by Ryan and Wand[52] These analyses using the coal train data have motivated the remainder of this thesis. A mainstay of this work is the use of temporal aggregation, which we describe presently.

2.4 Temporal aggregation

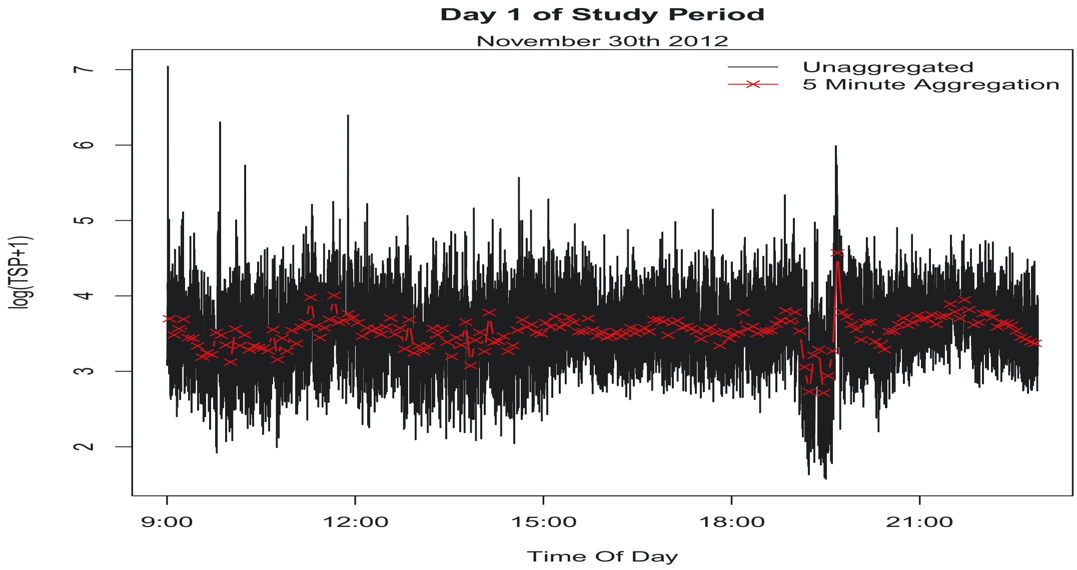
The size of our data, and the issue of the computational power required for analyses such as the gam model in equation (1), led us to consider the implementation of temporal aggregation to reduce the size of our dataset. This is covered in more detail throughout this thesis, however we explore some of the effects of this aggregation here.

Temporal aggregation is the process dividing a series of observations into non-overlapping intervals, and transforming each interval into an aggregate. This aggregation can be into any number of parameters, such as into means, or medians. Given a series $\{y_n\}_{n=1}^{N}$, we can transform through aggregation to a series of means, $\{s_m\}_{m=1}^{M}$,

$$s_m = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} y_n \tag{2}$$

where J is the size of the non-overlapping interval, and $M = N/J$ resulting observations. In the event that M is not an integer, we remove the final n observations such that N/J is an integer. The aggregated series s_m can be a mean or another form of aggregate. In our coal train dataset, in particular for the log(TSP+1) air particulate measurements, we can aggregate the data into means for every 5 minutes of observations, in this particular example. This effect is shown in figure 2.4. The unaggregated, original data is recorded at a 6 second interval, thus for 5 minute aggregation, we take each consecutive 50 observations and record the mean as in equation (2). The black line here correspond to the unaggregated data for the first day of our study period, and the blue line is the temporally aggregated dataset.

Figure 2.4: Comparison of unaggregated and aggregated log(TSP+1) data, with an aggregation period of 5 minutes.



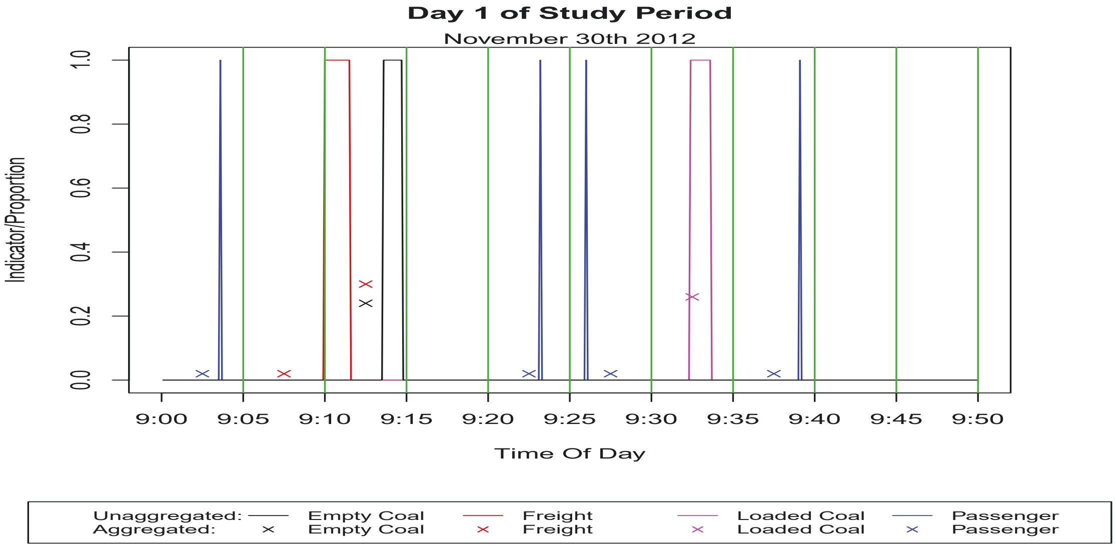
It is clear that the implementation of temporal aggregation results in a loss of information, however, in the case of 5 minute aggregation, we have reduced the number of observation for the first day of our analysis from 9255 to 185. Once we extend this aggregation to the complete dataset, we transform the data from 614,119 to 12,282 observations. This is a large reduction, however it is necessary for some of the modelling techniques that we utilise throughout this thesis.

The temporal aggregation must also occur on any independent variables, in this case we apply an aggregation on the indicator variable for each train type in the following method,

$$z_{km} = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} x_{kn}, \tag{3}$$

where $k = 1, \ldots, 5$ for each of the train types respectively. Each train type z_k is thus transformed from an indicator variable to a proportion of time it passed the monitor in each aggregation block m. Thus, for the empty coal train indicators in our coal train dataset, the aggregation of the indicator results in a proportion of time the empty coal train was passing the monitor in each 5 minute block of time, for the case of 5 minute aggregation. The same occurs for the other train types. This effect is shown in figure 2.5. We can see that trains such as the loaded coal train, denoted by the purple line, are reduced to one observation of around 0.25, which is the proportion of that 5 minute interval that it was passing the monitor. The empty coal and freight trains have a similar effect. The passenger trains however, pass the monitor with great speed, and as such are reduced to a much smaller proportion, barely greater than zero. In the case of a smaller aggregation period, of say 1 minute, this proportion is higher.

Figure 2.5: Comparison of unaggregated and aggregated data for each passing train variable. The aggregation period is 5 minutes. We show the indicator variable for passing coal(empty and loaded), freight and passenger trains, as well as their respective proportions as a result of temporal aggregation. The green vertical lines indicate each 5 minute block upon which the aggregation is performed.

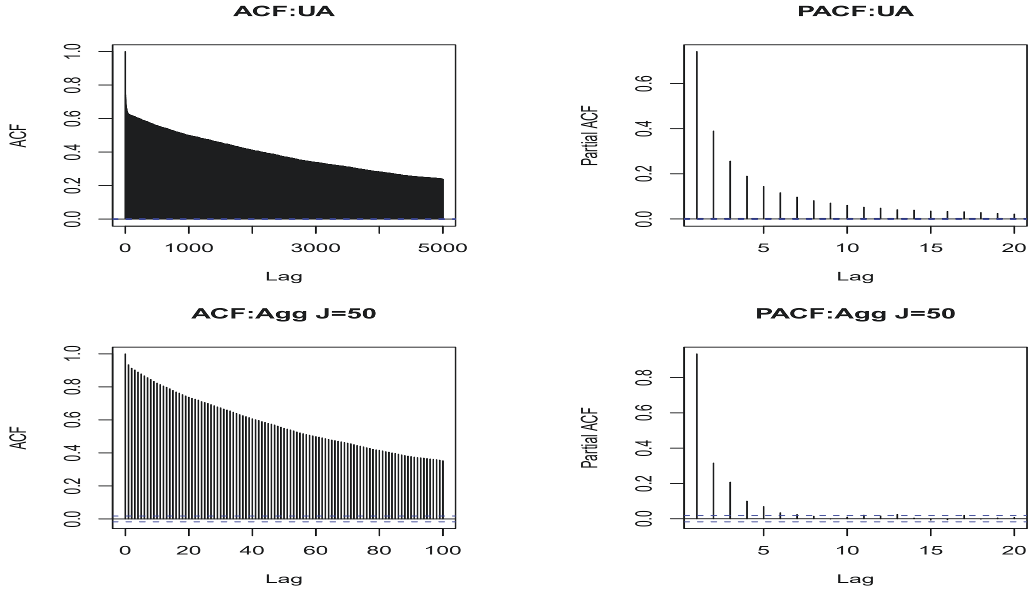


Having shown the effect of temporal aggregation on the coal train data variables, we now turn to its impact on the autocorrelation present in the dataset.

2.5 Long memory dependence

Our initial analyses indicated that there was a strong autocorrelation effect in the coal train dataset. Due to the computer memory requirements for models such as the gam model in equation (1), we considered the use of temporal aggregation to reduce the size of the dataset, thereby allowing the use of such methods. In figure 2.6 we present the Auto Correlation Function (ACF) and Partial AutoCorrelation Function (PACF) for the unaggregated and aggregated logged TSP measurements. We continue with 5 minute aggregation as in the previous section. The ACF plots, shown on the left hand side of figure 2.6 indicate that there is a strong level of autocorrelation. The PACF plots on the right, indicate that there is also a Moving Average effect in the data. These correlation effects are not strongly impacted by temporal aggregation. The Moving Average component is reduced somewhat by the aggregation, which can be seen by the lower number of significant lags.

Figure 2.6: ACF and PACF plots for the unaggregated and aggregated (5 minutes) log transformed TSP measurements.



Upon further review of the ACF plots in figure 2.6, we noticed that the lags in the ACF plot were never reaching zero. For the unaggregated data in the top right plot, the lags remain significant even after lags of 5000. Furthermore, the lags of the ACF plots are decreasing hyperbolically. These two points of information indicate the the coal train dataset contains a long memory dependence.

This is a much stronger form of correlation than the more often encountered short term dependence that is present in AutoRegressive (AR), Moving Average (MA), and ARMA time series models.

The effective analysis of a time series with long memory dependence is a key goal of this thesis. We cover strategies to overcome this complexity in chapter 3, 4 and 6 of this thesis. In the following chapter 3, we present a detailed review of long memory models.

2.6 Discussion

The coal train dataset described in this chapter is the key motivating dataset for this thesis. The difficulties that arise from its large size, of over 600,000 observations, and the complexity of the data due to the presence of long memory dependence, are the focus of this thesis. In this chapter we have presented aspects of the coal train dataset that are important for our following analyses. We described the different air particulate measurements PM1, PM2.5, PM10 and TSP. There are a number of train types that pass the monitor during our study, and they vary wildly in number of passing trains, speed of transit and length. These factors have a considerable impact on our analyses.

We have presented the goal of the original study as requested by the EPA, and covered some previous analyses of this dataset. The large size of the dataset, prompted the idea of using temporal aggregation to reduce its size and thereby permit the use of more applicable statistical methods. Upon further inspection of the data, we found clear evidence of long memory dependence in the air particulate measurements.

The implementation of temporal aggregation, and a comprehensive review of long memory time series, are the focus of the following chapter of this thesis.

3 Linear Regression of a Long Memory Time Series with ARFIMA Errors

3.1 Introduction

Analysis of long memory time series has been well researched for many years. Difficulties arise when the size of the dataset is large. Current regression with AutoRegressive Fractionally Integrated Moving Average (hereby referred to as ARFIMA) errors is constrained by computational memory limitations. The estimation of the inverse of a covariance matrix with long memory dependence computationally is infeasible on the average computer. Methods such as banding the covariance matrix to reduce the size of the covariance matrix are not possible due to the significance of large lags. To overcome this issue, we implement a temporal aggregation on a dataset to reduce the size of the dataset. The motivation for our analysis is the coal train data as described in the previous chapter.

The goal of our analysis is to capture and quantify how much a passing train impacts air particle measurements in the Newcastle coal transport corridor. This dataset has a length of over 600,000 observations as well as a number of covariates of interest. As the data is recorded at 6 second intervals, there is a strong level of dependence present. We first conduct a literature review on the topic of long memory time series, focusing in particular on temporal aggregation. Following we describe the linear regression with ARFIMA errors modelling strategy, which we apply in this chapter.

We conduct a simulation study to determine if a regression with ARFIMA errors is an adequate model for long memory data. We generate data to mimic the coal train dataset with long memory and compare linear regressions with iid and ARFIMA errors. Furthermore, we explore the effect of temporal aggregation through a comparison of unaggregated and aggregated simulated data, using the two error structures as just mentioned.

In our application, we apply the linear regression with ARFIMA errors to the coal train dataset. We use insights gained from our simulations study, namely, that the ARFIMA error model provides an adequate fit, and that temporal aggregation does not negatively affect the data analysis unless it is over an unreasonably large aggregation period. An unreasonably large aggregation period is strongly data dependent. For the coal train data, where the majority of freight and coal trains pass the monitor between 1 and 3 minutes, an aggregation period of every 2 hours is unreasonable. We further justify this aggregation period selection through an interpretation of the variance of our coefficient estimates at each aggregation period. We are restricted to a minimum aggregation period of 5 minutes, due to the difficulties in estimating the inverse of a covariance matrix for data of any larger size.

Our analysis also covers the effect of temporal aggregation on the order of our ARFIMA error structure. As shown in the literature on the topic, temporal aggregation does not affect the long memory in a dataset. It does however impact the short term memory, which is associated with the AutoRegressive (hereby referred to as AR(p)) and Moving Average MA(q) components found in many time series data. We are able to show that the use of aggregation results in a simpler error structure, which is an advantage as the estimation of this error structure is a difficult task.

An interesting advantage of temporal aggregation in this situation is that before we aggregate, our passing train variables are indicators of whether a train is passing the monitor or not. After aggregation we have set each train's covariate to be a proportion of time a train was passing in each aggregation period. We hypothesize that a passing train's contribution to air particulate levels does not end the moment it has finished passing the monitor. From previous work of ours, as outlined in chapter 2 and in [52] and [41], we conclude that a period of 5 minutes after a train has passed the monitor is significant to air particulate levels. By analysing the aggregated data, where we have a proportion instead of an indicator, we hope to capture more of a passing train's effect than just what occurs when it is passing the monitor. The idea behind this is that for an indicator variable we do not include any of the after effect for a passing train, while a proportion variable can estimate the trains effect on each aggregation period.

We conclude this chapter with a review of this analysis and also discuss further work on the topic which we undertake in chapter 4. Here we attempt to improve our model from this chapter by specifying a tail length for each passing train.

3.1.1 Literature review

Time series data is marked by levels of autocorrelation that result in data being dependent over time. Dependence can be in the form of short term memory and long term memory. Short term memory is predominantly analysed through AutoRegressive (AR), Moving Average (MA), a mixture of the two as in an ARMA(p,q) model and also in AutoRegressive Integrated Moving Average (ARIMA) models. Certain time series datasets are dependent over much longer periods. Such time series can be described as possessing long memory. The first work on this topic was by Hurst[27], who while working in Egypt, was tasked with predicting how much the Nile river flooded from year to year, with the goal of designing dam characteristics to regulate the flow of the Nile. He found significant long term correlations from the Niles outflows. From his initial proposal the study of long memory began.

For a time series $\{y_n\}_{n=1}^{N}$, Mcleod and Hippel [44] define a long memory process when the autocorrelations are non-summable, as in,

$$\sum_{j=-\infty}^{\infty} \rho_j = \infty$$

where ρ_j is the autocorrelation at lag j. To account for the long memory dependence in a time series, an ARFIMA model can be implemented. Presently they remain the most used long memory models[19]. The ARFIMA model is an extension of the AR, MA, ARMA and ARIMA models, which we describe briefly here. A more complete review can be found in An AutoRegressive AR(p) model can be expressed as,

$$y_n - \phi_1 y_{n-1} - \phi_2 y_{n-2} - \ldots - \phi_p y_{n-\rho} = \epsilon_n$$

which we can rewrite as,

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_\rho B^p) y_n = \epsilon_n$$

$$\phi(B) y_n = \epsilon_n$$

where B is called a backshift or lag operator, and ϵ_n are iid normal with $E[\epsilon_n] = 0$ and $E[\epsilon_n^2] = \sigma_\epsilon^2$.

A Moving Average MA(q) model is written as,

$$y_n = \epsilon_n - \theta_1 \epsilon_{n-1} - \theta_2 \epsilon_{n-2} - \ldots - \theta_q \epsilon_{n-q},$$

which, using the backshift operator we can rewrite as,

$$y_n = \theta(B)\epsilon_n.$$

An ARMA(p,q) model is the combination of these AR(p) and MA(q) models, and is written as,

$$\phi(B)y_n = \theta(B)\epsilon_n.$$

We can then extend the ARMA(p,q) model to the ARIMA(p,d,q) model, by differencing the y_n,

$$\phi(B)(1-B)^d y_n = \theta(B)\epsilon_n,$$

where d is an integer.
Mcleod and Hippel [44] then considered a fractionally integrated process as,

$$(1-B)^d y_n = \epsilon_n, \tag{4}$$

where y_n is integrated of order d, and d is a real number. Through this finding, Granger[16], Granger and Joyeux[17], and Hosking[26], developed the class of ARFIMA model. The ARFIMA model extends on the ARIMA model by allowing the differencing parameter, d, to be a real number.

$$\phi(B)(1-B)^d y_n = \theta(B)\epsilon_n, \tag{5}$$

where $-1/2 < d < 1/2$. Mcleod and Hippel defined the process $\{y_n\}_{n=1}^{N}$ as being long memory when $0 < d < 1/2$ and its autocorrelations are all positive with a hyperbolic decay rate. For $-0.5 < d < 0$, the process has short memory and it is 'anti-persistent'. Here it's autocorrelations are negative, except at lag 0, and hyperbolically decay to zero.

In 1992, Sowell[57] derived the exact maximum likelihood estimator of the ARFIMA process with normally distributed innovations ϵ_n. It is at this stage that estimation becomes difficult as the inverse of the autocovariance matrix must be calculated at each iteration of the likelihood. When the dataset has a large number of observations N this can be computationally demanding.

Due to these difficulties in the estimation of the maximum likelihood of long memory ARFIMA process for datasets with large N, in this thesis we are interested in the application of temporal aggregation to reduce the size of the dataset. Temporal aggregation has been previously considered for AR, ARMA, ARIMA and ARFIMA processes. Temporal aggregation is the aggregation of observations into non-overlapping intervals, such as, for a series $\{y_n\}_{n=1}^{N}$, we transform through aggregation to a series of means, $\{s_m\}_{m=1}^{M}$,

$$s_m = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} y_n \tag{6}$$

where J is the size of the non-overlapping interval, and $M = N/J$ resulting observations. In the event that M is not an integer, we remove the final n observations such that N/J is an integer. The aggregated series s_m can be a mean or another form of aggregate, as is defined by flow aggregation, or it can be a sample of the interval, also known as stock aggregation[56]. Throughout this chapter and the thesis, we consider flow aggregation with means as the aggregating procedure, a more thorough review of the temporal aggregation process is in chapter 2.

The effect of temporal aggregation on short and long term memory has been the topic of many statistics articles. Beginning with short term memory, Amemiya and Wu [2] explored the effect of temporal aggregation on an AR(p) process, with the resulting outcome being that for an AR(p) process, after aggregation, the aggregated process will be an ARMA(p,q) process, where the MA(q) have at most p lags. Thus, after aggregation from an AR(p), which we can also rewrite as, ARMA(p,0), we have an ARMA(p,p) process. This finding shows that aggregation of an AR(p) process results in an ARMA(p,p) process. For the special case ARIMA(0,d,q) of an ARIMA(p,d,q) process, Tiao [60] shows that as the aggregation block size m increases, the MA(q) lag coefficients reduce to the order of differencing d. Thus we are able to remove the some of the short term memory through temporal aggregation, and the data becomes ARIMA(0,d,d). Stram and Wei[59] debate the previous theory that aggregation of an AR(p) process results in an ARMA(p,q) process. They determine, in some cases, that the lag order of the AR(p) and ARIMA(0,d,q) process can be reduced as a result of temporal aggregation. Thereby reducing the amount of short term memory for an aggregated process.

Moving on to long memory dependence, Ding, Granger and Engle[13] showed that temporal aggregation does not affect the long memory dependence for an economic time series. This finding was then extended by Chambers[8] to the general case for data with long memory, where he stated that ”it has been shown that the temporally aggregated variable, whether it be a stock or flow variable, retains the same (possibly fractional) order of integration as the underlying series”.

Our interests in this chapter are the analysis of an environmental dataset with long memory and a large number of observations. We utilise temporal aggregation to reduce the size of the dataset, thereby allowing the use of a linear regression with ARFIMA errors, which would not be possible using current methods for the size of our dataset. There have been a number of analysis on similar datasets.

An applied analysis using an ARFIMA model on rainfall totals and lake inflows was conducted by Montari, Russo and Taqqu[45]. This analysis shows the power of an ARFIMA model to capture long and short term memory. They also consider temporal aggregation for the lake inflows and determine the reduced size of the aggregated data to be too small to detect long memory. Their analysis for rainfall totals found this specific data to not have any long term or short term memory. Nonetheless they implemented an ARFIMA model with ARFIMA(0,0,0) order and found the ARFIMA model to remain useful in the event that the data was uncorrelated. Iglesias, Jorquera and Palma[29] present an analysis of air particulate data, namely PM2.5 (one of our air particulate measurements as described in Chapter 2). Here they implement a linear regression with ARFIMA errors, as we proceed to do in this chapter. They consider the impact of missing observations and also present a Bayesian analysis.They conclude that ignoring the long memory persistence in the residuals distorts the results. Pan and Chen[46] also consider air quality data with an ARFIMA model. They compare ARFIMA and ARIMA models for long memory data and conclude the ARFIMA outperforms the ARIMA. Their air quality measure was PM10, also described in Chapter 2. Lau, Hung, Yuen and Cheung [35] consider seasonal ARIMA models for roadside air quality data. They determine that there is a long memory presence in their data and that their ARIMA approach does not adequately account for this effect. Ham et al[24] consider temporal aggregation on nanoparticle exposure levels. Due to a non-stationary dataset, they compare AR(1) and ARIMA models over aggregation periods of 1, 5 and 10 minutes. They determine that the ARIMA outperforms the AR(1), as the data is non-stationary. Furthermore, they consider the averaging time to be data dependent.

Aside from these applications on long memory data, there is a number of works on statistical properties of long memory. Yajima[70][71] worked on regression with long memory errors and explored the asymptotic efficiency loss from using Ordinary Least Squares (OLS) instead of Generalized Least Squares (GLS) estimators. Koul and Mukherjee[34] considered rank statistics in linear regression models with long range dependent errors, while Hall[23] applied the block bootstrap for regression with long memory errors. Robinson and Hidalgo[51] explored regression with long memory in both the errors and stochastic regressors. Dahlhaus[11] constructs an efficient weighted least squares estimator for regression with long memory. A thorough review of long memory can be found in Beran[5], Baille[3], and Hang and Palma[25].

Time series analysis has also been considered in machine learning and bayesian literature. Makridakis et al[42] compares machine learning and ARIMA/ETS modelling strategies and concludes that machine learning is not as effective, while Bontempi et al[6] determined that one step ahead forecasting is viable under machine learning methods, however that multi step ahead forecasting was not effective, particularly in comparison with statistical methods such as ARMA and ARIMA. In the instance of long memory time series, there is limited research conducted. We point the reader to Greaves-Tunnell and Harchaoui[20] who discuss some of the challenges faced by machine learning for long memory time series. Bayesian analysis has also been conducted on time series data. Of particular note is the literature review by Steel[58], where bayesian analysis for long memory using ARFIMA models has been used, especially by Koop et al[33]. We now present the methodology for our implementation of a regression with ARFIMA errors.

3.2 Long memory regression models

Given a linear model, we outline the procedure for estimating the regression model with ARFIMA(p,d,q) error structure. We begin with a linear regression of form,

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{7}$$

where $\mathbf{y} = (y_1, \ldots, y_N)^\top$ is a vector of dependent variables, with $n = 1, \ldots, N$. $\mathbf{X}$ is a matrix of dimension $n \times p$, with p variables, and β is the vector of coefficients and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ is a vector of regression errors.

The AutoRegressive Fractionally Integrated Moving Average ARFIMA(p,d,q) process is described by

$$\Phi(B)(1-B)^d \epsilon_n = \Theta(B)\eta_n, \tag{8}$$

where $-1/2 < d < 1/2$, $\eta_n \sim N(0, \sigma_\eta^2)$ is white noise, B is the backshift operator $B\eta_n = \eta_{n-1}$, the AR operator is $\Phi(B) = 1 + \phi_1 B^1 + \ldots + \phi_p B^p$, and the MA operator is $\Theta(B) = 1 + \theta_1 B^1 + \ldots + \theta_q B^q$. The fractional difference operator is $(1-B)^d = \sum_{k=0}^{\infty} \tau_k B^k$, where $\tau_k = \Gamma(k-d)/\Gamma(k+1)\Gamma(-d)$. $\Gamma(\cdot)$ denotes the Gamma function. For $d \in (0, 1/2)$ we have a long memory process. When $d = 0$ we have an ARMA model, and for $d \in (-1/2, 0)$ we have a short memory process.

In the instance where we have ARMA(p,q) short memory and long memory in the data, we can apply a linear regression with ARFIMA(p,d,q) errors, which we consider below.

To fit of a linear regression with ARFIMA(p,d,q) errors, we utilise the `arfima`[62] package in R. A brief overview of this process is presented here.
Firstly, we estimate the residuals from the Ordinary Least Squares (OLS) model,

$$\epsilon = y - X\beta.$$

From these residuals we can determine the ARFIMA(p,d,q) error structure. We begin by calculating the autocovariance function of a stationary ARMA process with mean $\mu = \frac{\sum_{n=1}^{N} \epsilon_n}{N}$ is,

$$\gamma_k = E[(\epsilon_n - \mu)(\epsilon_{n-k} - \mu)] \tag{9}$$

where the variance matrix of $\epsilon = (\epsilon_1, \ldots, \epsilon_N)^{'}$ is,

$$\Sigma = \begin{pmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{N-1} \\ \gamma_1 & \gamma_0 & \cdots & \gamma_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{N-1} & \gamma_{N-2} & \cdots & \gamma_0 \end{pmatrix}, \tag{10}$$

which is a symmetric Toeplitz matrix with $\epsilon \sim N_N(\mu, \Sigma)$. To calculate the autocovariances in equation (9), we use the method developed by Sowell[57]. Then the log likelihood is,

$$logL(d, \phi, \theta, \sigma_\eta^2) = -\frac{N}{2}log(2\pi) - \frac{1}{2}log|\Sigma| - \frac{1}{2}z^{'}\Sigma^{-1}z, \tag{11}$$

where $z = \epsilon - \mu$. This process gives us our estimates for the d fractional differencing parameter, the $\phi_1, \ldots, \phi_p$ AutoRegressive parameters and the $\theta_1, \ldots, \theta_q$ Moving Average parameters. From these estimates we can update our autocovariance function, which we insert into the autocovariance matrix, $V\hat{a}r(y) = \hat{\Gamma} = \hat{\gamma}_k$, for the model in equation (7). The inverse and determinant of Γ is calculated through the Trench algorithm. Then to estimate the coefficients of the regression model (7), the Generalized Least Squares (GLS) estimate is,

$$\hat{\beta} = [X\Gamma^{-1}X^{'}]^{-1}X^{'}\Gamma^{-1}y \tag{12}$$

Using the linear regression with ARFIMA errors as presented above, we are able to analyse the coal train data that is the motivation for this chapter. Firstly we conduct a simulation study to determine if the linear regression with ARFIMA errors is an adequate modelling strategy.

3.3 Simulations

The aim of this chapter is to model a long memory process correctly and to consider the impact of temporal aggregation on our data. In our simulations we show that regression with iid gaussian errors will lead to an incorrect analysis. This is a result of incorrect standard errors which will hinder the analysis. We thus consider regression with ARFIMA errors. This new strategy adjusts for the long memory in the data. We are also interested in the effect of temporal aggregation on our data. We simulate using a number of aggregation block sizes and interpret these outcomes.

We begin our simulation study by describing our data generating model. We present one linear regression with long memory errors. Once we have generated our simulation data, we compare the fit of a linear regression with iid gaussian error structure and a linear regression with ARFIMA errors.
Following this comparison, we compare output for regression with ARFIMA errors on a number of alternate aggregation block lengths, namely $J = 10, 20, 50$ and 100. We conclude with a review of our simulations before moving on to our application study.

3.3.1 Data generating model

We aim to mimic the Hunter Valley Coal Train dataset in a most simple form. Our simulated data has the form,

$$y_n = \beta_0 + \beta_1 x_n + \epsilon_n, \tag{13}$$

where $n = 1, \ldots, N$ is the length of the simulated data. The error structure is set as having $\epsilon_n \sim ARFIMA(3, 0.4, 3)$. We simulate ARFIMA errors from equation 8, as having AutoRegressive order of AR(3), where $\rho_1 = 0.5$, $\rho_2 = 0.2$ and $\rho_3 = 0.2$. The Moving Average order, MA(3) are all equal at $\theta_{1,2,3} = 0.2$. The $\eta_n \sim N(0, \sigma_\eta^2)$ are white noise where we have set $\sigma_\eta^2 = 0.2$.

We set the independent variable to be,

$$x_n = \begin{cases} 1 & \text{if train is present at time } n, \\ 0 & \text{otherwise.} \end{cases}$$

Our simulations set each passing train to have a fixed length of ten observations. We set $\beta_0 = 3$ and $\beta_1 = 5$. The covariate x_n representing the passing train variable is created by running through one observation at a time and deciding if there is a train present or not according to a Bernoulli random variable with $B(p = 0.05)$. If a train is generated, we let it run for ten observations. Once the train has passed, we set the following two observations to be zero. This is implemented to be consistent with reality, whereby there must be a gap after a train passes before another can follow. If the Bernoulli random variable decides that no train has passed we move onto the next observation. The selection of this probability of $B = (p = 0.05)$ was chosen as it resulted in the number of simulated trains matching that of the applied dataset.

In the following subsection we compare our analysis of the simulated data by fitting the model with iid gaussian errors and long memory ARFIMA errors. We also explore the effect of aggregation on the fit of the model and then we compare alternative aggregation sizes. We implement the temporal aggregation procedure as in the previous section.
For the dependent variable after aggregation, we transform the $\{y_n\}_{n=1}^N y_n$ into $\{s_m\}_{m=1}^M$, as specified in equation (6). Our simulated data has a size reduction of a factor of J, where the aggregated data has a length of $M = N/J$. This same aggregation process is applied to the independent variable $\{x_n\}_{n=1}^n$, as shown in equation (6), which is transformed into a proportion of time a train is passing in each aggregation block. Our aggregated independent variable is now $\{z_m\}_{m=1}^M$.

Having presented our data generating model, we now turn to some analyses of the simulations. We begin with a comparison of models for fitting the data with long memory.

3.3.2 Comparison of regression with iid errors and ARFIMA errors

Given data simulated from equation (13) we fit a linear regression to the unaggregated and aggregated data. Our simulated data has long memory persistence and as such we need to account for this occurrence. We thereby fit a linear regression with ARFIMA errors and with gaussian iid errors to determine which model captures the error structure correctly.

On the unaggregated data we fit the model,

$$y_n = \beta_0 + \beta_1 x_n + \epsilon_n, \tag{14}$$

with error structure either, having iid gaussian errors,

$$\epsilon_n \sim i.i.d.N(0, \sigma_\epsilon^2), \tag{15}$$

or with ARFIMA(p,d,q) errors,

$$\Phi(B)(1-B)^d \epsilon_n = \Theta(B)\eta_n, \tag{16}$$

with

$$\eta_n \sim i.i.d.N(0, \sigma_\eta^2).$$

For the aggregated data we have,

$$s_m = \beta_0 + \beta_1 z_m + \omega_m, \tag{17}$$

with error structure either,

$$\omega_m \sim i.i.d.N(0, \sigma_\omega^2), \tag{18}$$

or

$$\Phi(B)(1-B)^d \omega_m = \Theta(B)\zeta_n, \tag{19}$$

with,

$$\zeta_m \sim i.i.d.N(0, \sigma_\zeta^2).$$

We want to determine which of the above 4 models correctly fit the simulated data. To do so, we firstly examine the fit of each model by considering their residuals. We compare the autocorrelation (ACF) and partial autocorrelation (PACF) functions for each of the 4 models. We then compare the coefficient estimates for each model as well as their standard errors. We consider 20 replicates of the data generating model as given in model (13). We set $n = 1, \ldots, N$ to have length of 5000 observations. The aggregated models had aggregation block size of $J = 10$, therefore the aggregated data was of length $M = 500$.

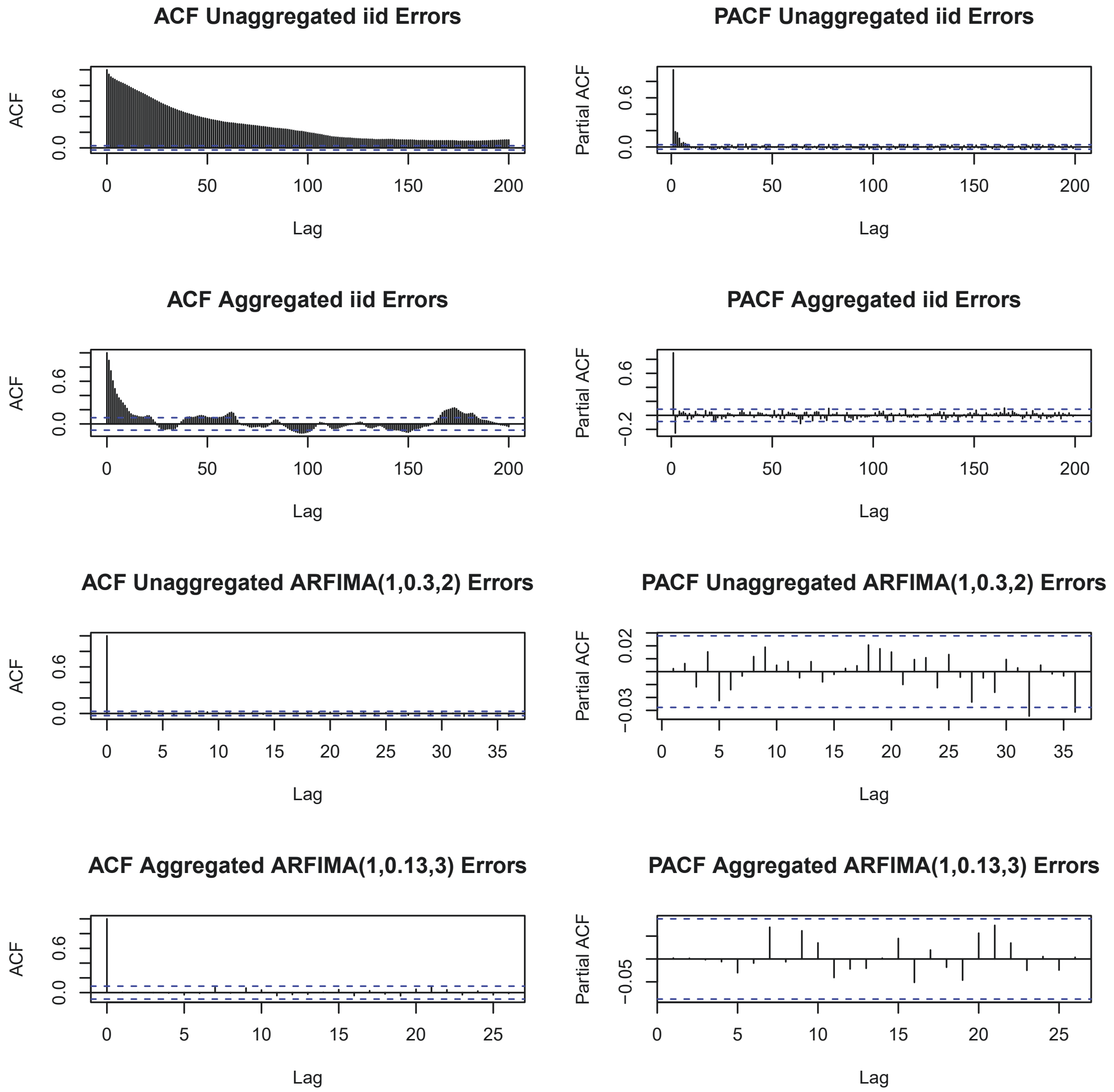
In figure 3.1 we show the residuals for one of the 20 replicates of each model (14) and (17) and both regressions with iid and ARFIMA errors as in (15),(16),(18) and (19). The first row of figure 3.1 shows the ACF and PACF for model (14) with iid errors as in (15). The ACF shows a hyperbolic decay, with over 200 lags being significant. The PACF also shows significant lags. The fact that the model (15) with iid errors has a significant error structure remaining after the model fit indicates that it is not correctly specified.

In the second row we present the residuals for one of the 20 replicates for the model (17) with iid

errors as in (18). We are interested if temporal aggregation will have an effect on the short and/or long memory. As we mentioned previously in our literature review (section 3.1.1), previous work on temporal aggregation for time series shows that as we increase the aggregation block length J, the short term memory (ARMA) component of ARFIMA, namely the AR(p) and MA(q) parameters approach zero, see [8],[28], and [60]. This effect is shown in the second row, where the ACF is still significant at large lags, showing that the long memory has been only slightly affected by temporal aggregation. However the AR(p) parameters have been reduced as the rate of decay is not as large, which is shown by the ACF being no longer in the form of hyperbolic decay. The PACF which shows the moving average, MA(q), component of the short term memory is no longer as significant as in the first row.
We can see from a comparison of the first and second rows that temporal aggregation affects the short but not long term memory. Temporal aggregation permits a simpler ARMA(p,q) parameter selection for the data, however we must still consider another technique on the data to account for the long term memory.

We turn to the regression with ARFIMA errors as show by (16) and (19) for the unaggregated and aggregated models. In the third row of figure 3.1 we show the residuals after fitting model (14) with ARFIMA errors. The simulated data had ARFIMA(3,0.4,3) error structure. Our model in this particular case estimated the ARFIMA order to be (1,0.3,2). The ACF and PACF show white noise residuals. This indicates a correct fit. The final row in figure 3.1 shows the fit of model (17) with ARFIMA errors where we have (1,0.13,3) as our ARFIMA parameters. Again as in the the third row, the residuals have white noise structure. We can see that regression with ARFIMA errors fits the data well.

Figure 3.1: Inspection of residuals for one of the 20 replicates of the linear regression with gaussian iid errors and ARFIMA long memory errors, for both unaggregated and aggregated data. ACF and PACF plots for each model. ACFs show the autocorrelation structure of the errors as well as the presence of long memory in the data. The PACF's show the moving average aspect of the ARMA structure.



ACF Unaggregated iid Errors

PACF Unaggregated iid Errors

ACF Aggregated iid Errors

PACF Aggregated iid Errors

ACF Unaggregated ARFIMA(1,0.3,2) Errors

PACF Unaggregated ARFIMA(1,0.3,2) Errors

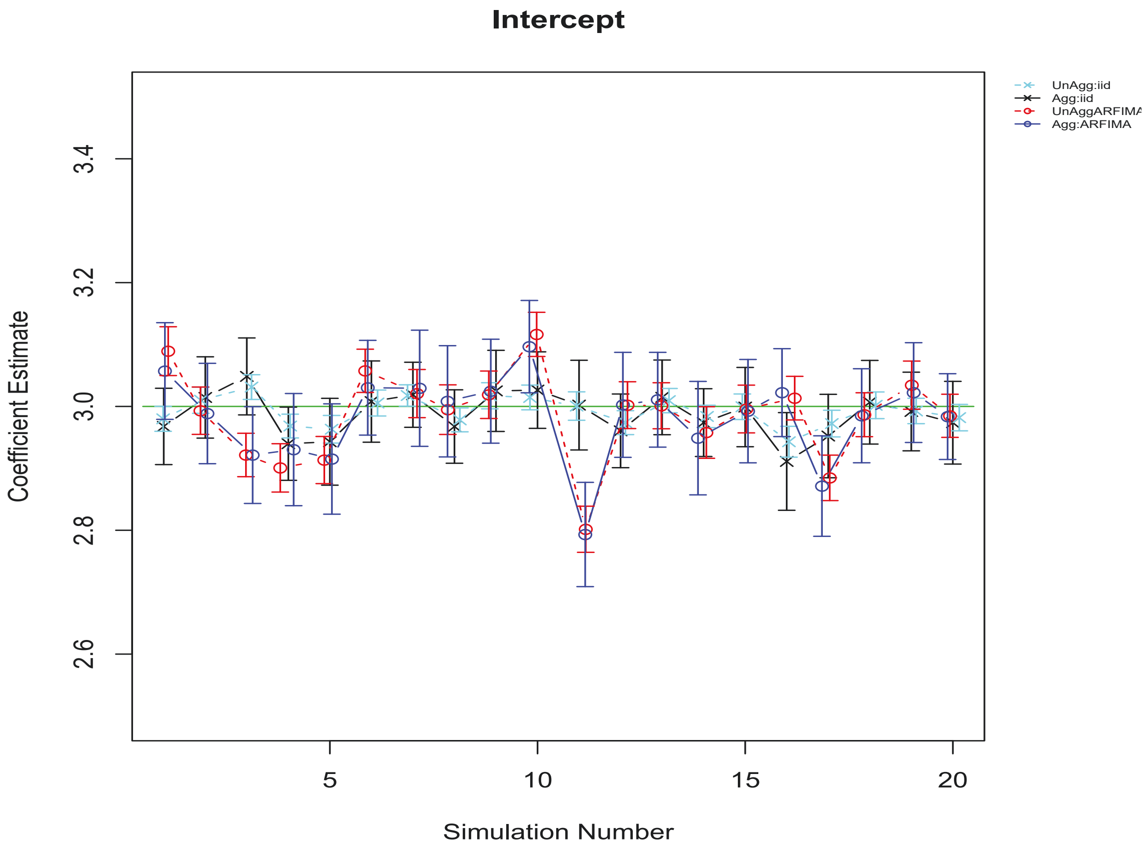
ACF Aggregated ARFIMA(1,0.13,3) Errors

PACF Aggregated ARFIMA(1,0.13,3) Errors

Confident in our choice to utilise linear regression with ARFIMA errors for an analysis of data with long-memory dependence, we now consider the models (15),(16),(18) and (19), and their coefficient estimates. We analyse the same 20 replications over the data generating model in (13) and in the residuals analysis in figure 3.1. We begin with the results for the intercept, $\hat{\beta}_0$, in figure 3.2 and then consider the passing train coefficient, $\hat{\beta}_1$, in figure 3.3.

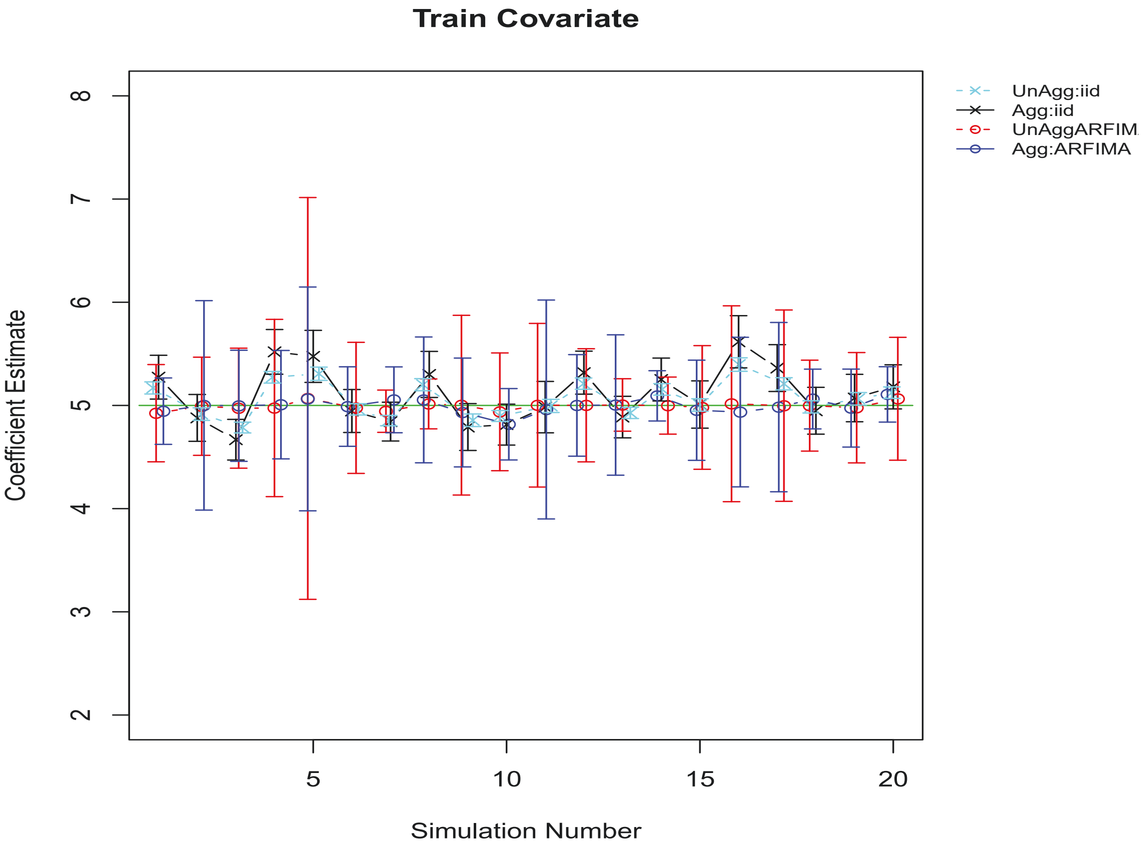
In figure 3.2 below, we compare the 4 model fits. The true simulated value is $\beta_0 = 3$ which is marked by the horizontal green line. The 2 models for the linear regression with iid gaussian errors are marked with a cross for the unaggregated and aggregated cases. The regressions with ARFIMA errors are marked with a circle.
Overall from figure 3.2 we can see that the 4 models cannot be distinguished. All 20 simulations perform as to be expected from each model. The only case that differs from the others is the simulation number 11, where the regression with ARFIMA errors diverges from the mean. We consider this to be a negligible error as it only occurs once in twenty cases.

Figure 3.2: Comparison of intercept estimate for linear regression with guassian iid errors and arfima long memory errors, for both unaggregated and aggregated data. Standard Error bars are included as well as the true $\beta_0 = 3$ as indicated by the solid green line. The solid black and blue lines are the aggregated models, for gaussian and ARFIMA models respectively. The dotted cyan and red lines are the unaggregated gaussian and ARFIMA models.



We now turn to the estimates of the passing train covariate $\hat{\beta}_1$. From figure 3.3 below, we can see two clear effects. Firstly the standard errors for the models with ARFIMA errors are much larger than those with iid gaussian errors. This is to be expected as, when we have autocorrelation in the data, the iid gaussian errors will result in consistent coefficient estimates and incorrect standard errors as the estimation is no longer BLUE(best linear unbiased estimator). The standard errors in this case are underestimated. Secondly, the estimates under the unaggregated and aggregated models with ARFIMA errors are more accurate than the iid models. In none of the 20 simulations does the linear regression with iid errors have a more accurate $\hat{\beta}_1$ than the ARFIMA models.

Figure 3.3: Comparison of passing train covariate for linear regression with gaussian iid errors and arfima long memory errors, for both unaggregated and aggregated data. Standard Error bars are included as well as the true $\beta_1 = 5$ as indicated by the solid green line. The solid black and blue lines are the aggregated models, for gaussian and ARFIMA models respectively. The dotted cyan and red lines are the unaggregated gaussian and ARFIMA models.



In our simulations thus far, we have shown that in the presence of long memory in a data set, the use of a linear regression model with ARFIMA errors outperforms the iid model. This is to be

expected as we are not accounting for the autocorrelation structure in the data.
This chapter utilises linear regression with ARFIMA errors and we present an arguement for its use above.

3.3.3 Effect of temporal aggregation

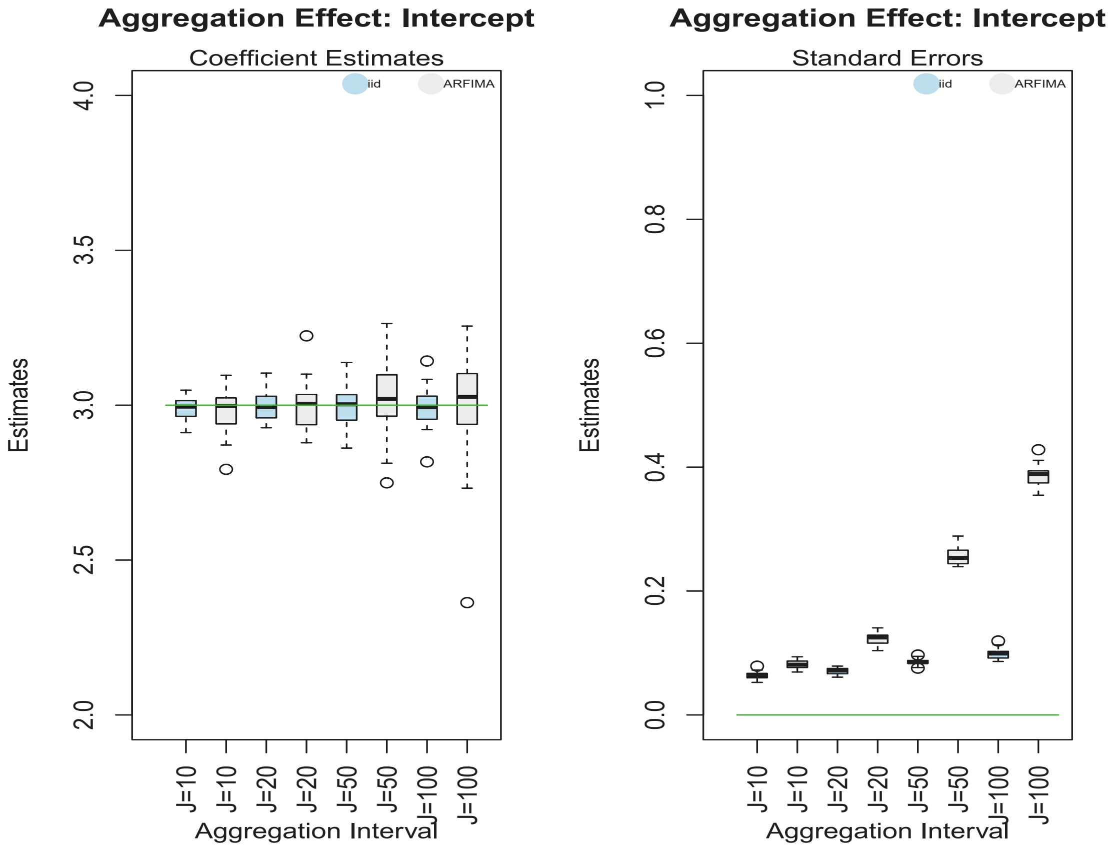
An issue with regression with ARFIMA errors that we have mentioned throughout, is that there are computational limitations to the size of a dataset that can be analysed using current methods. Therefore, a method around this limitation is to temporally aggregate the data to reduce its size. We consider the impact of temporal aggregation on a linear regression with ARFIMA errors.

Using the data generating model from equation (13) we consider four different temporal aggregation block sizes, $J = 10$, $J = 20$, $J = 50$ and $J = 100$. For each simulation the aggregated data is kept to a size of $M = 500$, where $M = \frac{N}{J}$. The length of each simulation under different aggregation block size is kept constant to remove the effect of different data sizes on the analysis.

In figures 3.4 and 3.5, the effect of temporal aggregation on $\hat{\beta}_0$ and $\hat{\beta}_1$ from the model (17) with errors as in (18) and (19) is shown. In each figure the results of 20 replications for each aggregation block size are contrasted with iid gaussian and ARFIMA errors. As discussed in the previous subsection, the standard errors for the iid errors can be ignored as they are incorrect. This is due to to the model ignoring the autocorrelation in the data.

Figure 3.4 compares the results of the simulations for $\hat{\beta}_0$. On the left of the figure we present the coefficient estimates. For the model (17) with ARFIMA errors, as in equation (19), as marked by the grey boxplots, an increase in the temporal aggregation block size from $J = 10$ to $J = 100$ results in a more variable estimate. As the aggregation increases, the variation in the estimates increases. On the right, we have the standard errors associated with each replicate. As the temporal aggregation increases the standard errors also increase. While the aggregation periods $J = 10$ and $J = 20$ are not dissimilar, the aggregation causes the coefficient estimates and their standard errors to become much more variable as we increase the aggregation block size.

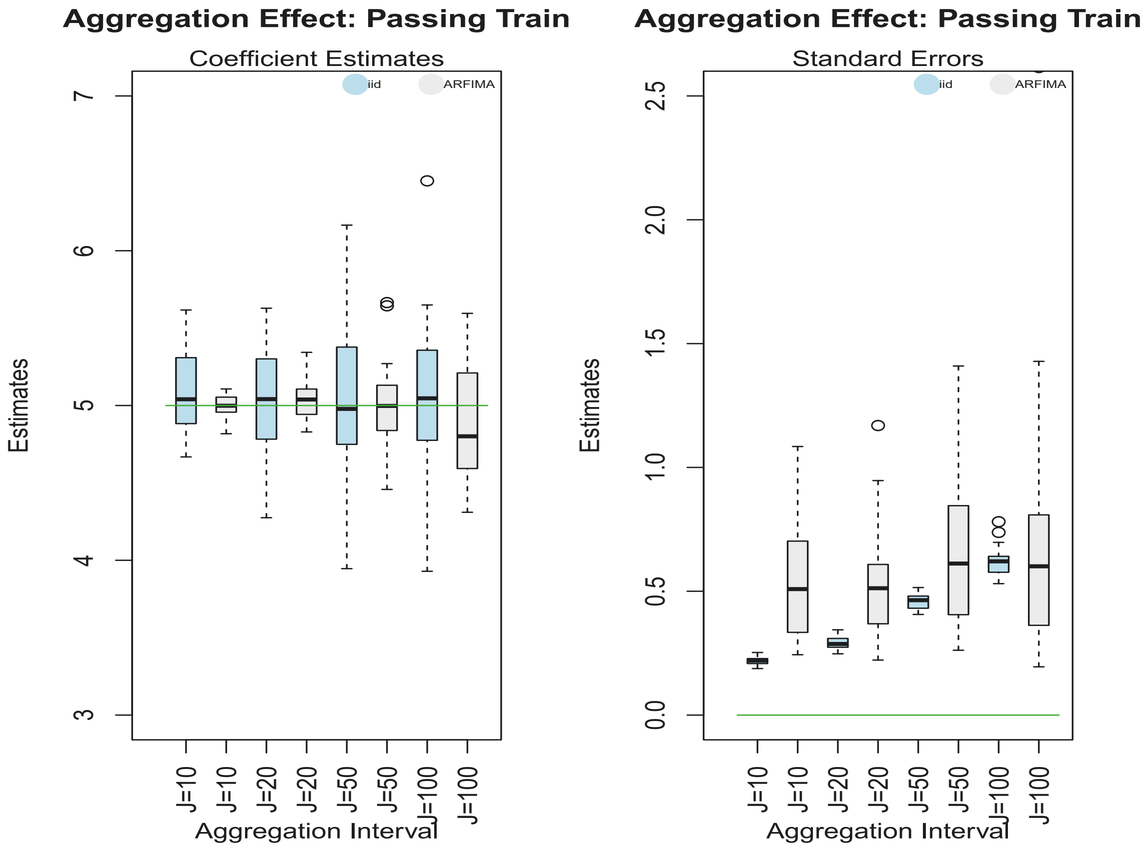
Figure 3.4: Comparison of the intercept estimates, $\hat{\beta}_0$, for aggregated guassian iid and ARFIMA(p,d,q) regression as the aggregation interval increases . The true value is denoted by green line. Guassian iid error implementation is shown by the blue shaded boxplots, while the ARFIMA errors are shown by the grey shaded boxplots.



Having observed the effect of aggregation on the intercept from model (17), we now turn to the covariate effect. In figure 3.5 we show the effect of aggregation on model (17) for $\hat{\beta}_1$. Again a comparison of the model with both the iid and ARFIMA error structures is presented. As shown in the previous subsection, the model with ARFIMA errors outperforms the iid case. This is shown by the coefficient estimated being less variable while the standard errors cannot be analysed as we know they are incorrect as mentioned throughout this chapter.

From the data generating model in equation (13), we set each passing train to have a length of ten observations. The passing train covariate in the aggregated case is a proportion of time a train is passing in each block. Thus, as we increase the aggregation block size, we find the values of z_m, the proportion of a time a train was present in each aggregation block, decreases. The plot on the right side of this figure shows that as we increase the aggregation size, the coefficient estimates become more variable in the ARFIMA case, and at $J = 100$ they are no longer centered around the true value of $\beta_1 = 5$. This leads us to conclude that the aggregation block size should not be too large in relation to this simulated data.

Figure 3.5: Comparison of passing train covariate estimates, $\hat{\beta}_1$, for aggregated guassian iid and ARFIMA(p,d,q) regressions as the aggregation interval increases. The true value is denoted by green line. Gaussian iid error implementation is shown by the blue shaded boxplots, while the ARFIMA errors are shown by the grey shaded boxplots.



In table 3.1 below we consider the impact of aggregation on the ARFIMA errors. As mentioned in our literature review at the beginning of this chapter, temporal aggregation does not affect the fractoral differencing parameter d. Short term memory is affected by temporal aggregation as the aggregation can reduce the AR(p) and/or MA(q) order. This effect is evident from table 3.1. If we compare the AR(p) and MA(q) components of the ARFIMA error structure in the unaggregated and aggregated case, we find that in only one case does the aggregated data have a higher ARMA(p,q) order than the unaggregated case (simulation 20). In every other case the aggregated data has a lower ARMA order than the unaggregated data. Furthermore the fractoral differencing parameter d is largely similar between the unaggregated and aggregated data sets.

Table 3.1: Simulation Results for ARFIMA(p,d,q) error structure for unaggregated and aggregated models ($J = 10$ as the aggregation block size).

		Unaggregated			Aggregated J=10	
Sim No.	AR(p)	d (SE)	MA(q)	AR(p)	d (SE)	MA(q)
1	1.00	0.28 (0.08)	2.00	1.00	0.20 (0.11)	1.00
2	2.00	0.29 (0.09)	3.00	1.00	0.41 (0.08)	1.00
3	3.00	0.34 (0.08)	1.00	1.00	0.33 (0.09)	1.00
4	3.00	0.40 (0.08)	4.00	1.00	0.32 (0.14)	1.00
5	2.00	0.46 (0.04)	4.00	2.00	0.40 (0.09)	2.00
6	3.00	0.33 (0.11)	3.00	1.00	0.23 (0.13)	1.00
7	2.00	0.12 (0.08)	1.00	1.00	0.21 (0.08)	0.00
8	1.00	0.12 (0.13)	2.00	1.00	0.31 (0.10)	0.00
9	1.00	0.38 (0.06)	2.00	1.00	0.29 (0.16)	1.00
10	1.00	0.32 (0.07)	2.00	1.00	0.21 (0.13)	1.00
11	1.00	0.35 (0.07)	2.00	1.00	0.38 (0.10)	0.00
12	1.00	0.32 (0.07)	2.00	1.00	0.28 (0.13)	0.00
13	3.00	0.14 (0.27)	3.00	1.00	0.34 (0.09)	0.00
14	1.00	0.20 (0.08)	2.00	2.00	0.18 (0.10)	0.00
15	1.00	0.32 (0.08)	2.00	1.00	0.26 (0.11)	0.00
16	4.00	0.37 (0.10)	3.00	2.00	0.32 (0.10)	1.00
17	4.00	0.40 (0.08)	5.00	2.00	0.38 (0.08)	2.00
18	1.00	0.23 (0.10)	2.00	1.00	0.11 (0.13)	1.00
19	1.00	0.30 (0.07)	2.00	1.00	0.21 (0.18)	2.00
20	1.00	0.30 (0.06)	2.00	1.00	0.13 (0.19)	3.00

3.3.4 Review of simulations

This simulation study has been implemented to show that a linear regression with ARFIMA errors can be used to fit a data set with long memory dependence. Due to memory constraints on the average computer, whereby the storage and inversion of a large dense matrix with a large number of significant lags is difficult to compute, has led to our use of temporal aggregation to reduce the size of a dataset.

Throughout this simulation study we contrast the results of a linear regression with iid and ARFIMA error structure and determine that the ARFIMA model results in a correct fit. By comparing the residuals of models for unaggregated and aggregated data with iid and ARFIMA errors, we show that the iid errors do not account for the high levels of autocorrelation in the data. The ARFIMA models result in white noise errors and the models fit well for both unaggregated and aggregated data.

Temporal aggregation is implemented as it reduces the size of the data allowing for the ARFIMA model to be used on large time series as is the case in the following application. Furthermore, the implementation of temporal aggregation results in in a reduction in the ARMA(p,q) order which is part of the ARFIMA(p,d,q) error structure. This is a known result in the theory as discussed in our literature review. Another aspect of temporal aggregation is that it does not affect the long memory dependence of a dataset. This is also shown to be the case in our simulations.

Another key aspect of temporal aggregation is that there is a loss of information from reducing the size of the dataset. By comparing our models under a number of aggregation block sizes, we show that as we increase the level of temporal aggregation, our results become more variable. This is to be expected, however from our simulations we show that temporal aggregation, when not done overzealously, can provide consistent outcomes. This particular outcome is of considerable interest as in our following application analysis, we must balance the level of temporal aggregation so that we can reduce the data to a size that allows our analysis to be estimated by our models, and also to provide inference about our analysis. We continue on to this application analysis where we apply the linear regression with ARFIMA errors and temporal aggregation as presented in this section.

3.4 Application

The goal of this chapter is to estimate the effect of a passing train on air quality, in the Newcastle rail corridor, based on the data presented in Chapter 2 of this thesis. As outlined in Chapter 2, the citizens of Newcastle, Australia, are concerned by the passing coal trains in the residential area of Newcastle. Given data provided to the Environmental Protection Authority (EPA) by the Australian Rail Track Corporation (ARTC), we are tasked with determining the effect of each passing train type on air quality levels.

The dataset provided has been thoroughly discussed in Chapter 2. Here we attempt to model the data correctly. There are a number of constraints we must overcome to achieve this. Firstly, the data is large and complex. Of particular concern is the presence of long memory dependence in the data. To correctly model this effect we utilise a linear regression with ARFIMA errors. This model, while directly applicable to this dataset, comes with its own limitations. The key being that the covariance matrix of a long memory model is dense and the calculation of its inverse is particularly demanding on computer memory. Current statistical methods to reduce this computational restriction include the banding of the covariance matrix, in Toeplitz form, to reduce the computational memory required to conduct this inverse operation. However, the significance of a large number of lags, regardless of their size, by nature of their long memory removes this possibility. Our dataset has a length of 614,000 observations, while the current limitations on the estimation of the inverse matrix with long memory suggest a matrix with dimension of 15,000 is the maximum capacity of the average computer. Thereby, without access to a super computer, we must reduce the dimension of the data to be able to implement the regression with ARFIMA errors.

It is at this point that we turn to temporal aggregation to reduce the size of the dataset. Although aggregation results in a loss of information, we argue that it is necessary to fit the best model for the data. Furthermore, it is our hypothesis that the unaggregated data does not adequately capture a passing trains effect. Our dataset records when a train is passing the monitor and the type of train. There is an arguement to be made that there is an after effect that occurs in the period immediately after a train has passed the monitor. This suggestion is further explored in the following chapter, yet here we believe that temporal aggregation is a simple way of estimating part of this after effect. A final benefit of temporal aggregation is that it reduces the complexity of the data. As we discussed in our literature review, there has been a number of statistical analyses on the effect of temporal aggregation on short and long term memory. The current theory is that temporal aggregation will reduce the amount of short term memory, as recorded by the ARMA(p,q) parameters, and that temporal aggregation does not affect the long memory dependence. Thus, the use of temporal aggregation allows the implementation of a simpler ARMA(p,q) structure, as part of the ARFIMA(p,d,q), in our models on aggregated data. This simpler ARMA(p,q) structure is beneficial under the principle of parsimony, whereby a a reduced number of parameters can still maintain a strong explanatory effect.

In this application section, we apply temporal aggregation to our coal train dataset. Following which, we describe the linear regression with ARFIMA errors model that we apply in our analysis. Based on our results using this model, we are able to estimate the linear regression with ARFIMA errors and draw inferences on the data. We compare a number of aggregation block sizes on the dataset and make the case that the regression on the temporally aggregated data yields enlightening results. We conclude this section with a review of the analysis.

3.4.1 Temporal aggregation of the Hunter Valley Coal Train dataset

Given the size of the unaggregated coal train dataset, we use temporal aggregation to reduce its size. The temporal aggregation process is presented in Chapter 2 of this thesis. We provide a brief repetition of this process in relation to the application here.

The data is recorded at 6 second intervals. With about 614,000 observations we are unable to estimate a linear regression with ARFIMA errors. The inversion of the covariance matrix fails on our computer for data of length greater than 15,000 observations at a maximum. We are thus constrained to aggregate to a level below this size. Therefore if $N = 614,000$, and our $M < 15,000$, the lowest temporal aggregation we can implement is of a 5 minute aggregation block size. Here J is the aggregation block length. For observations every 6 seconds, 5 minutes equates to $J = 50$. Thus, at 5 minute aggregation our aggregated data has length $M = \frac{614000}{50} = 12,280$. For ten minute aggregation our aggregated data now has length $M = 6,140$. We consider temporal aggregation blocks of 5 minutes to 10 minutes, as well as 15, 20, 30 minute aggregation and then onto 1,1.5 and 2 hour aggregation. We discuss our choices for these aggregations in the analysis.

For this analysis we are interested in the air particle measurement Total Suspended Particulate (TSP), which is our dependent variable and our dependent variables are each of the passing train types, empty coal, loaded coal, freight, passenger and unknown trains.

For the dependent variable TSP, we take the logarithm of the data to reduce the effect of outliers in the data. In unaggregated from the dependent variable is $y_n = log(TSP_n + 1)$. We add one to the data before the logarithm to overcome any simulations where the data was observed as zero. Temporal aggregation transforms the unaggregated data from y_n to s_m,

$$s_m = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} y_n.$$

The temporal aggregation must also occur on the independent variables, in this case we apply an aggregation on the indicator variable for each train type in the following method,

$$z_{km} = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} x_{kn},$$

where $k = 1, \dots, 5$ for each of the train types respectively. Each train type z_{km} is thus transformed from an indicator variable to a proportion of time it passed the monitor in each aggregation block m.

Our covariates Z are listed as, z_{1m}:= the proportion of passing Empty Coal trains in period m,
z_{2m}:= the proportion of passing Freight trains in period m,
z_{3m}:= the proportion of passing Loaded Coal trains in period m,
z_{4m}:= the proportion of passing Passenger trains in period m, and
z_{5m}:= the proportion of passing Unknown vehicles in period m.
z_{5m} can be attributed to a passing train who's type has not registered, a maintenance vehicle, or any object which has not been classified.

With our newly aggregated dataset, we can now specify a model to analyse the data.

3.4.2 Application: Linear regression with ARFIMA errors

Once we have aggregated our dataset we can apply our model to the data. In this section we present our model for the analysis and our results from the model. We discuss the choice of aggregation block size J and explore the outcomes for each train type. The question we are trying to answer is how much does each train type affect air particulate measurements. Our linear regression model attempts to capture the effect of each train on the aggregated data.

Having aggregated our data for each choice of J, we fit a linear regression model of form,

$$\begin{aligned} s_m =& \beta_0 + \beta_1 z_{1m} + \beta_2 z_{2m} + \beta_3 z_{3m} \\ &+ \beta_4 z_{4m} + \beta_5 z_{5m} + \omega_m, \end{aligned} \tag{20}$$

where the errors, ω_m have an ARFIMA(p,d,q) structure, which we denote as,

$$\Phi(B)(1-B)^d \omega_m = \Theta(B)\eta_m, \tag{21}$$

where $-1/2 < d < 1/2$, $\eta_m \sim N(0, \sigma_\eta^2)$ is white noise, B is the backshift operator $B\eta_m = \eta_{m-1}$, the AR operator is $\Phi(B) = 1 + \phi_1 B^1 + \ldots + \phi_p B^p$, and the MA operator is $\Theta(B) = 1 + \theta_1 B^1 + \ldots + \theta_q B^q$. The fractional difference operator is $(1-B)^d$.

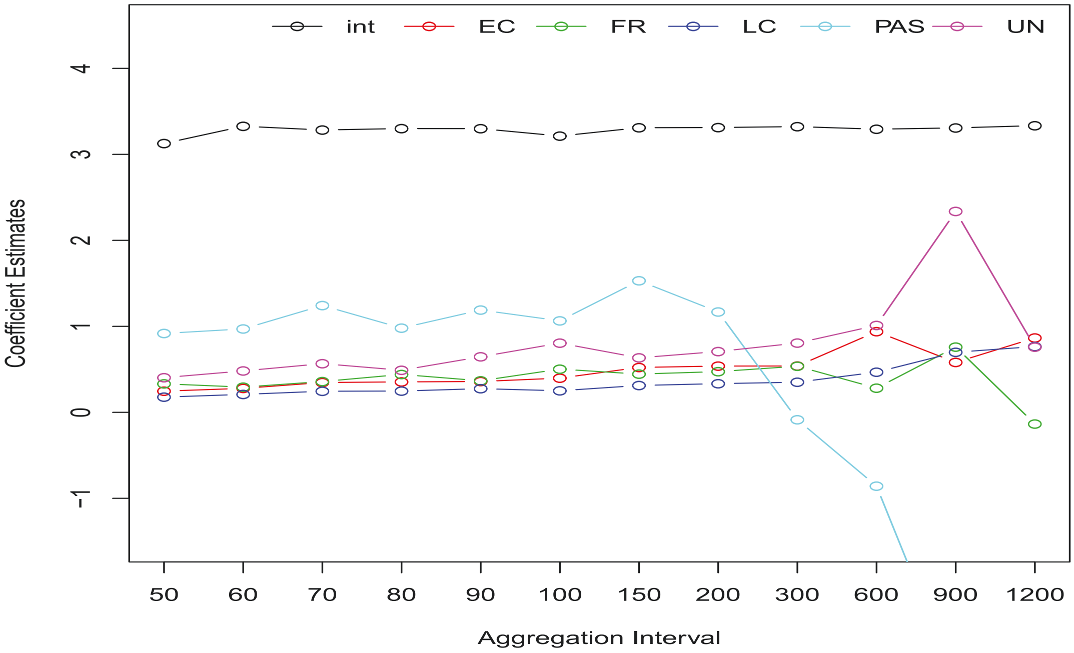
This model is analysed for each of the aggregation blocks of size J. A more detailed methodology for a linear regression with ARFIMA errors is provided in section 3.2.

In chapter 2 we provided a more detailed discussion of the dataset, a key insight on deciding the length of the aggregation block is the time it takes for each train type to pass the monitor. From figure 2.2, we can see that the majority of coal and freight trains take between 1 and 3 minutes to pass the monitor. As we showed in our simulation study, the level of temporal aggregation affects the results of models similar to the ARFIMA model in equation (20). Therefore we are in a trade off of data reduction and inference. As the majority of trains take between 1 and 3 minutes to pass the monitor, we hypothesize that an aggregation of between 1 and 10 minutes should provide reasonable outcomes. However as we have mentioned throughout, due to the size of the data, we can only aggregate to 5 minutes as a minimum. This does not appear to be an issue, however we will discuss this presently in our results. It is important to mention that the passenger trains typically come and go within one observation. This is attributed to the short length and increased speed over the other train types. Temporal aggregation greatly affects this variable. Fortunately it is not crucial to our analysis as we are concerned with the coal train effect. This allows us to use a larger aggregation size. For the unknown variable it is difficult to make any inference as we are unsure of what each passing object is. Due to the issues with passenger and unknown trains, we only include them in our initial analysis. We then focus on the coal and freight trains.

Our previous work in [53], [41] and [52] explored the hypothesis that a passing train causes an increase in air particle measurement not only when it passes the monitor but also in the period after it has past the monitor. In [52],[41] we considered a 5 minute period after a train passed as significant. In the unaggregated data, we can only attempt to capture an effect in the period a train is passing the monitor. By transforming the indicator variables into proportions, we are able to capture the effect of a passing train in that period. This allows for the chance that we will include some of this after effect for each passing train.

We now present our results of model (20) in the figure 3.6, which shows our results for the aggregation periods from 5 minutes to 2 hours. The passing train coefficient estimates for the covariates empty coal, freight, loaded coal and unknown, increase only slightly as the aggregation period increases. This indicates that the increase in aggregation does not affect the model greatly. Once we reach an interval of 30 minutes, the estimates become much more variable. We attribute this effect to the fact that the proportion of time a train is passing in a period of 30 minutes is becoming negligibly small. This is also the case for passenger trains where for a 5 minute aggregation period of 50 observations there may be only one observation where there is a passing train. This results in the vast majority of $z_{4m} < 0.02$, and by ten minute aggregation $z_{4m} < 0.01$ for any $m = 1, \ldots, M$. Furthermore the data size has reduced to around 2,000 observations at 30 minute aggregation, and at 2 hours we have only 500 observations remaining. This is clearly too large an aggregation for a passing train that takes between 1 to 3 minutes to pass the monitor on average.

Figure 3.6: Outcomes of linear regression with ARFIMA errors, as in model (20), for the aggregation periods of 5 minutes to 2 hours. The intercept is the black line, with loaded empty coal trains in red, freight trains in green, loaded coal in blue, passenger trains in light blue and unknown in purple.



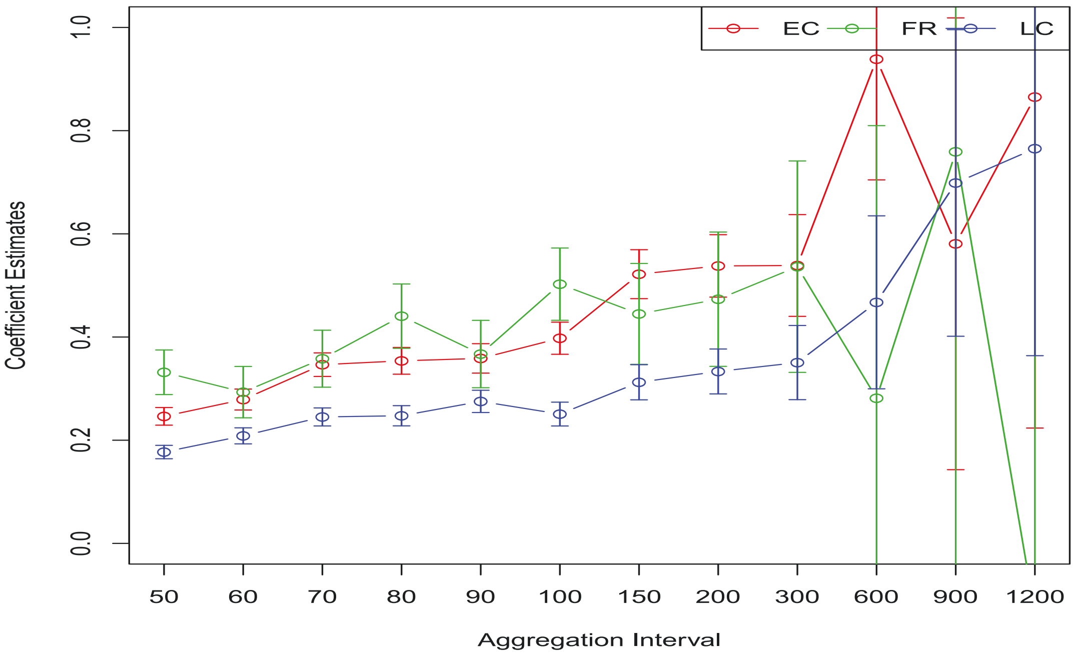
The unknown covariate has the largest effect on air particle measures. This covariate could be a mixture of any of the train types or any other vehicle, such as rail maintenance, and as such we do not consider it in any further analyses. Nonetheless, we are predominantly interested in how the empty coal, freight and loaded coal trains are impacting the air particulate measures. For these three train types, at all aggregation periods up to 30 minutes, we find that passing freight trains are causing the largest increase, followed by empty coal trains and finally the loaded coal trains. The order of largest to smallest effect on air particulate for these three trains is the same as in our previous analysis using GAM modelling in [52] and [41] which we discuss in chapter 2.

To make further inferences about each passing trains effect on air particulate measurements, we consider the standard errors for each coefficient estimate using model (20). Figure 3.7 is a zoomed in plot of the previous figure 3.6 where we have limited the results to empty coal, freight and loaded coal. The goal of this analysis has been to estimate the effect of each train type on air particulate levels. It is difficult to determine a final answer with regards to how much each train type contributes to air particulate levels as the coefficient estimates are increasing as we increase the aggregation period. However, we are able to determine the order of largest to smallest contributor amongst the train types. It is clear that freight trains have the largest contribution to increased air particulate levels, followed by empty coal and loaded coal.

The inclusion of the standard errors indicates that our confidence in the empty coal and loaded coal train estimates is high. Their standard errors are small for the periods between 5 and 10 minute aggregation. For freight trains this is not the case. The standard errors are much larger and at times such as at 6 and 9 minute aggregation they overlap with coefficient estimate of the empty coal trains. This overlap suggests that the estimates for freight trains in these cases may not be the largest effect. In chapter 2 we showed a table 2.1 that showed the number of passing trains for each train type. It shows that the number of passing freight trains is much lower than both the loaded and empty coal trains. As such, we can attribute this larger standard error to the fewer number of passing freight trains.

As we aggregate for periods larger than 10 minutes, the coefficient estimates begin to change in their order. From figure 3.7 we can also see that the standard errors for all three train types become increasingly large. We use this fact to determine that an aggregation period of over ten minutes results output that is less useful.

Figure 3.7: Outcomes of linear regression with ARFIMA errors, as in model (20), for the aggregation periods of 5 minutes to 2 hours. Empty coal trains are in red, freight trains in green and loaded coal in blue. We have included standard errors for each train type.



In our simulation study we showed that the theory that temporal aggregation does not affect the fractoral differencing parameter d is true. What this means is that after aggregation the long memory dependence does not disappear. Temporal aggregation affects short term memory, as found in the ARMA(p,q) component of ARFIMA, resulting in a reduction of the ARMA(p,q) order. We study the estimation of the ARFIMA order for model 20 over the aggregation periods of 5 minutes to 2 hours in the table 3.2 below. Due to the size of the unaggregated data, we cannot compare ARFIMA models for the unaggregated and the aggregated regressions. However we can see that long memory remains in the data for all aggregation periods, this is shown by the fractoral differencing parameter d remaining between $d \in (0, 0.45)$. We also note that the short term memory in the AR(p) and MA(q) parameters reduces as we increase the aggregation period. At 5 minute aggregation we have an ARMA(3,1) order while after 2 hours of aggregation we have an ARMA(1,1) order.

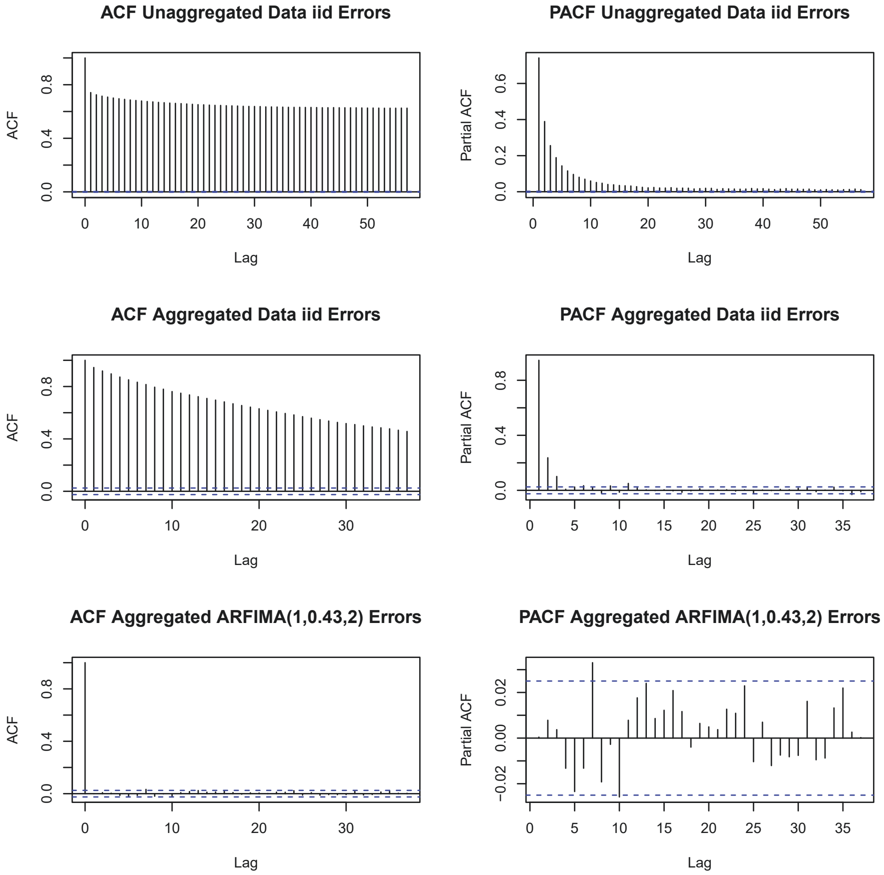
Table 3.2: Temporal aggregation effect on ARFIMA(p,d,q) errors from model (20) for aggregation periods from 5 minute to 2 hours.

J	5min	6min	7min	8min	9min	10min
AR(p)	3.00	2.00	3.00	1.00	1.00	1.00
d	0.45	0.31	0.45	0.37	0.40	0.43
MA(q)	1.00	2.00	2.00	1.00	3.00	2.00
J	15min	20min	30min	60min	90min	120min
AR(p)	3.00	2.00	1.00	1.00	1.00	1.00
d	0.30	0.30	0.29	0.32	0.31	0.32
MA(q)	2.00	2.00	3.00	0.00	0.00	1.00

To conclude our application analysis we check the residuals of the model (20) in figure 3.8. We study the ACF and PACF for the residuals of the unaggregated linear regression with iid errors, the aggregated linear regression with iid errors and the model we use throughout this section, the linear regression with ARFIMA errors as in model (20). We restrict this figure to the aggregation period of 10 minutes($J = 100$) as the other aggregation periods have the same outcomes, and we do not wish to repeat this figure.

The first plot on the right of the first row, shows the ACF for the unaggregated iid model. Here we can clearly see that there is long memory in the data as there is a hyperbolic decay in autocorrelations and all lags are significant. Furthermore, the PACF indicates that there is a significant MA(q) component to the data as well. Moving on to the second row, we present the linear regression with iid errors for the aggregated data. Here there is still long memory as shown again by the hyperbolic decay in the ACF, this shows that aggregation does not remove the long memory, however it can change it. Furthermore the number of significant lags in the PACF signifying the MA(q) has reduced. Again the short term memory has been affected by temporal aggregation. The final row shows the ACF and PACF for the model (20) with ARFIMA errors. In this case we have estimated the order of the ARFIMA(1,0.43,2). The two plots show the errors of this model to be white noise. It confirms that our model is correctly estimating the data. While this is a single simulation, this pattern repeats itself in all other simulations conducted.

Figure 3.8: Residual analysis showing ACF and PACF plots for unaggregated iid model, aggregated iid model, and aggregated ARFIMA(1,0.43,2) model at $J = 100$ (10 minute aggregation).



In this subsection we have aggregated our data, provided a model and analysed its outcomes. We are able to make meaningful inferences and show that the model fit is adequate. We now provide an overview of this application.

3.4.3 Review of application

Due to the large size and complexity of the coal train data, we have implemented temporal aggregation. This has allowed us to utilise a linear regression with ARFIMA errors. In this application we begin by applying the temporal aggregation to the data. The choice of the temporal aggregation block size is dependent on our data. The minimum amount of temporal aggregation we can use is 5 minute periods. While we would be interested in a lower level of aggregation, it would not be possible to use this model due to the computer memory limitations involved with estimating the inverse of a large covariance matrix. Thus, we consider aggregation periods of between 5 minutes and 2 hours. From our analysis we are able to determine a reasonable aggregation period.

As passing trains usually take around 1 to 3 minutes to pass the monitor, we hypothesize that a 5 to 10 minute aggregation period captures the data most accurately. by comparing the results for all aggregation periods, we are able to show that an aggregation period of over 10 minutes causes too much information loss. This is shown in figures 3.6 and 3.7, where the coefficient estimates become more variable and their standard errors also increase to levels that signify insignificance.

After providing our estimates from model (20), we also check the residual structure to make sure we have a white noise series remaining, which confirms that the regression with ARFIMA errors is an adequate model. Another aspect of temporal aggregation is its effect on short and long term memory. We show that the aggregation results in a simpler ARMA structure, and that long memory is not affected by the temporal aggregation. This results is consistent with the literature on temporal aggregation for time series data. It also allows the estimation of an ARFIMA model that is not as complicated as would be the case for unaggregated data. The benefit of this is that we are less likely to misspecify the error structure.

The aim of this chapter is to determine the effect of each passing train on air particulate levels in the Newcastle rail corridor. Our priority is to determine which of the freight and coal trains impacts the air quality the most. We are able to show that freight trains contribute to the largest increase in air particulate levels, followed by empty coal trains and that loaded coal trains are the least likely to increase air particulate levels of the three. This result is consistent with our previous work on this data, where we used GAM modelling to investigate the same question.

Having completed our application analysis, we now turn to a discussion of this chapter.

3.5 Discussion

To account for the long memory structure of the coal train dataset, we utilise a linear regression with ARFIMA errors. The goal of this chapter is to quantify the effect of each passing train type on air particulate levels in the Newcastle rail corridor. As our dataset has a length of over 600,000 observations on air particulate levels, as well as a number of independent variables, we are limited in our analysis by the computational power and memory required to estimate such a model. An option to overcome this issue is the use of temporally aggregated data. By applying temporal aggregation, we are able to reduce the size of the dataset so that the estimation of our regression model is possible.

We show the process of this regression with ARFIMA errors and then conduct a simulation study to test its suitability for data with long memory dependence. By comparing regressions with iid and ARFIMA errors, we are able to determine that the ARFIMA errors adequately account for the large level of autocorrelation. The level of our temporal aggregation is dictated by the demands of the data, however we must select a reasonable aggregation period nonetheless. Therefore, in our simulation study we also cover the effect of temporal aggregation on a regression with ARFIMA errors. We conclude that as the aggregation period increases, the independent variables, which are aggregated to means, decrease, and this results in more variation in our analyses. Furthermore, the sensitivity analysis of many simulations considered the effect of temporal aggregation on long memory dependence. Our findings are consistent with the literature in that temporal aggregation does not affect the long memory component of dependency, merely the short term memory.

From our outcomes in the simulation study, we are able to discern the effect on air particulate levels between different train types in the coal train data. We conclude that the freight trains are associated with the largest increase in air particulate levels, followed by empty coal trains and then loaded coal trains. This outcome of the order of effect on air particulate levels is consistent with our previous work in chapter 2. Our application considers the aggregation periods of 5 minutes to 2 hours. Due to the increased variation of the coefficient estimates and their standard errors as the aggregation period increases from 10 minutes, we determine this range to be a reasonable level of temporal aggregation.

A further use of temporal aggregation for our hypothesized train tail effect. Namely the belief that a passing train contributes to air quality in the period after it has passed the monitor. As our unaggregated data is in the form of an indicator of presence for each train type, we are unable to estimate this hypothesized tail effect. However, the transformation to aggregated data, in particular a proportion of time a train is passing in each aggregation period, allows us to consider the possibility that each aggregated variable is capturing some of this tail effect. At this stage of this thesis this is a matter of speculation, however the focus of our next chapter is on suggesting this tail effect in the form of an omitted variable.

4 Impact of Model Misspecification for Time Series Modelling

4.1 Introduction

In chapter 3, we analysed the coal train data, as described in chapter 2, using a linear regression with AutoRegressive Fractionally Integrated Moving Average (ARFIMA) errors on temporally aggregated data. Regression with ARFIMA errors is a method for accounting for long memory dependence in a dataset. We provide a brief overview of this method shortly, and for a further review of regression with ARFIMA errors, please see chapter 3. Our results showed that as we increased the aggregation period, the coefficient estimates for each train type were systematically increasing. Based on our simulations and also considering the underlying theory, aggregation should not result in any systematic change in estimated parameters, only a change in their variability. For this reason, the results suggest that there was some model misspecification. In chapter 2, we presented some of our previous work [53] where we hypothesized that there is a residual effect after each train passes. Probably most readers will have had the experience of standing on a railway platform as an express train passes through, and feeling a continued sensation of swirling and stirred up air in the period after the train has passed. This is the phenomenon we refer to and for ease of exposition, we simple refer to a "tail effect". The goal of this chapter is to explore further whether this hypothesis might be true. We do this through some theoretical considerations as well as simulations to explore what happens when a tail effect has been incorrectly omitted from a fitted model. Our available data only indicates if a train is passing or not at any given point in time. There is no direct information regarding the presence of a tail effect. So, to determine if there is a tail effect, we compare two models. We consider a 'simple' model, where we only include a variable indicating a trains presence or absence. This is the same model as used in chapter 3. The second model we call the 'assumed' model. Here in addition the train presence/absence variables and we also assume a tail variable for each train type. These tail variables simply correspond to indicators that a train has finished passing by within a specified period of time. We will be considering tails of varying lengths throughout the remainder of this chapter.

We start out by considering linear models with independent errors since this lends itself well to some theoretical explorations of what happens when a tail variable is omitted. Then, as in chapter 3, we implement a linear regression with ARFIMA errors on the aggregated data. We are unable to fit an ARFIMA model to the coal train data for any aggregation period of less than 5 minutes due to the computational memory limitations that arise from the storage and inversion of a covariance matrix consisting of long memory autocorrelations. We discuss this in more detail in chapter 3. The underlying question that has motivated this entire thesis, is whether we can determine the effect of each passing train type on air quality. Our data is unaggregated, and recorded every 6 seconds in a two month span. The train variables indicate only whether a train is passing the monitor at a specific time or not.

In comparing the simple and assumed models, we begin by considering some theory for an omitted variable. In the case of a linear regression, we are able to show that there is a bias on the coefficient estimates when we ignore the presence of a tail effect. Heuristically the results make sense: if a passing train continues to have an impact by stirring up dust into the air even after the train has passed, then a naive comparison of levels in the presence versus absence of a train will tend to dampen the true effect. We then consider the assumed model. Here we explore the impact on the bias when we incorrectly assume the tail length. We are able to do this by comparing the assumed model with a known true model. These outcomes show that the assumed model has a reduced bias compared with the simple model.

We then conduct a simulation study, with data generated to mimic the coal train data. We consider regression with both iid and ARFIMA errors for data generated with and with long memory dependence. We have several main goals. Firstly we wish to assess via simulations whether our theory makes sense. We do this by simulating data with iid errors. Another goal is to compare the simple and assumed models against the true model, with the aim of identifying which model has the lowest bias. A broader and very important goal is to develop some guidelines for selecting an appropriate length of a tail variable. To do so, we consider and compare a number of tail lengths. We are able to show that the inclusion of the tail length, results in consistent coefficient estimates over a range of aggregation periods. We can thereby assume the tail length to be correct if the coefficient estimates for the train and tail remain constant for a number of aggregation periods.

Using the results of the theory and simulations, we compare the outcomes on the application, for the simple, assumed and combined models. The combined model is the special case of the assumed model, where we combine the known train variable with an assumed tail variable. We can immediately see that the assumed model has more consistent coefficient estimates over a range of aggregation periods, in contrast to the simple model. Confident that there is a tail effect, we must next decide what the length of the tail is. To do this, as in the simulations, we compare a number of tail lengths on the coal data for each train type. We conclude, that although there is no certain tail length, a reasonable choice is four minute tails. We determine this as the tail covariate estimates remain constant as we increase the aggregation period. The implementation of a tail variable for each train type then permits further inference on the effect each passing train type has on air quality. The selection of the length of a trains tail is not guided by an air pollution or train experts, but rather on the data and the aggregation effect that we uncover, which allows us to determine the tail length. At the conclusion of this chapter, we discuss the analysis and some further possibilities that we were unable to include at this stage. We now turn to a review of the literature on model misspecification and temporal aggregation, as well as a short review of linear regression with ARFIMA errors, which is used throughout this chapter.

4.1.1 Literature review

In the case of a data analysis where there is a belief that there is an omitted variable, Wickens [63], McCullum[43] and Fuller[15], all showed that the use of a proxy variable is better than omitting a variable, in terms of bias. The omitted variable bias is further explored in Wooldridge[68]. Here the omitted variable can be overcome through a proxy variable or an instrumental variable. Instrumental variables are often used in economics, social sciences and epidemiolgy. A famous analysis is in Card[7], where an instrumental variable is included in an analysis on university schooling. Here they include an instrumental variable for proximity to a university and determine that this unobserved (in the initial analysis) variable improves the results.

Analyses of the impact of an unobserved variable in the context of temporal aggregation are limited. Kim [32] studies the impact of omitted variables in a multilevel model. Through a simulations study, he shows that an omitted variable can cause more bias at a lower level than a higher level. Johnston et. al [30] also come to this conclusion, where the aggregated estimates are more accurate than unaggregated data. In the event that an analysis is more interested in results at grouped level, then Li[39] shows that in certain situations, the aggregated data analysis will yield less biased results in comparison to the individual level analysis. Such possible situations arise in the event of model misspecification, i.e. when certain data is unavailable or there is an incorrect understanding of the phenomenon, or when the analysis is more concerned with aggregated rather than individual data.

The models used throughout this chapter must account for the long memory dependence in our coal train data. We implement a regression with ARFIMA errors to model this situation. We now provide a brief overview of this strategy.

4.1.2 Linear regression with ARFIMA errors

This section provides a brief recap of ARFIMA modelling, as discussed in more detail in Chapter 3. Consider the time series regression $y_t = \beta X_t + \epsilon_t$, where the errors ϵ_t for $t = 1, \ldots, T$, present with long memory. Time series that have a long memory dependence can be characterised by having an autocorrelation that decays to zero at a hyperbolic rate. We can model the error with an AutoRegressive Fractionally Integrated Moving Average (ARFIMA(p,d,q)) process, as described by,

$$\Phi(B)(1-B)^d \epsilon_t = \Theta(B)\eta_t, \tag{22}$$

where $-1/2 < d < 1/2$, $\eta_t \sim N(0, \sigma_\eta^2)$ is white noise, B is the backshift operator $B\eta_t = \eta_{t-1}$, the AutoRegressive (AR) operator is $\Phi(B) = 1 + \phi_1 B^1 + \ldots + \phi_p B^p$, and the Moving Average (MA) operator is $\Theta(B) = 1 + \theta_1 B^1 + \ldots + \theta_q B^q$. The fractional difference operator is $(1-B)^d = \sum_{k=0}^{\infty} \tau_k B^k$, where $\tau_k = \Gamma(k-d)/\Gamma(k+1)\Gamma(-d)$. $\Gamma(\cdot)$ denotes the Gamma function. In the event that $0 < d < 0.5$, the errors are a long memory process. The parameters $(d, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$ are all to be estimated, before the linear regression coefficients can then be estimated.
For a more detailed review of the ARFIMA model and its fitting, please refer back to chapter 3 where we describe it in detail.

4.2 Model misspecification

As discussed earlier, our hypothesis for this chapter is that the systematic changes in estimated parameters associated with increasing levels of aggregation can be explained by the omission of an additional covariate that reflects the presence of what we call a tail effect. Consider the following simple regression model that includes a covariate indicating the presence or absence of a train, but no tail effect. Throughout the chapter, we will refer to this as the **simple model**. The model is

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t, \tag{23}$$

with a train passing set as,

$$x_t = \begin{cases} 1 & \text{if train is present at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

We then consider, what we refer to as the **assumed model**. Here we include both a train and a tail variable,

$$y_t = \beta_0 + \beta_1 x_t + \beta_2 u_t + \epsilon_t, \tag{24}$$

with the train's tail set as,

$$u_t = \begin{cases} 1 & \text{if tail is present at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

For both the simple and assumed model we consider, for now, to have independent identically distributed errors with $\epsilon_t \sim N(0, \sigma^2)$. In our subsequent bias calculations, we will compare what happens when models are fit with various assumed tail lengths, relative to what we call the True Model, that includes a correct tail length via the variable r_t, where

$$r_t = \begin{cases} 1 & \text{if a tail is present at time } t, \\ 0 & \text{otherwise,} \end{cases}$$

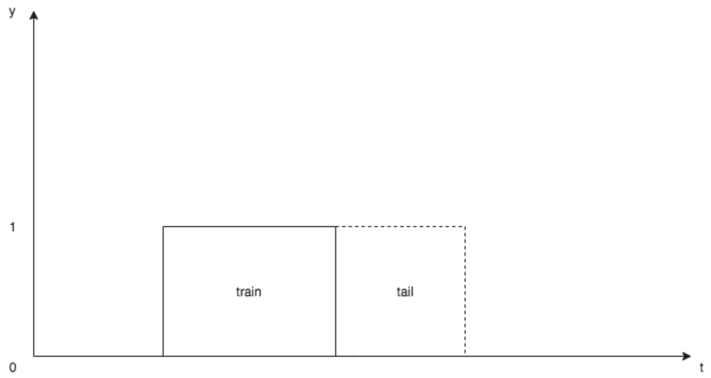
We now consider the simple model and the resulting bias due to the omission of a tail variable.

4.2.1 Simple model: No tail

In chapter 3, we considered the simple model, where we utilised the train variables which where present in our dataset. However, if there is a tail effect and we do not include it in our model, we believe there to be some model misspecification. We consider this situation here.

Firstly we consider the regression model (23), which we call the simple model. We are interested in determining the effect of the omitted variable in the form of a tail covariate. We believe the our model should be accounting for a tail effect as shown in figure 4.1 below.

Figure 4.1: Illustration of passing train and tail indicators



In this event, the model in equation (23) is incorrect. We thus consider an alternative model, where in an attempt to capture a passing trains tail, we specify both the presence of a train and a possible tail.

$$y_t = \beta_0 + \beta_1 g(x_t) + \epsilon_t \tag{25}$$

We set $g(x_t) = x_t + h(x_t)$. Here we denote a train as passing at, $x_t = 1$ and in the period after a train has passed, we set the tail as, $h(x_t) = 1$. Our interest is determining the bias that results from fitting model (23) when we believe the model (25) to be a more accurate representation of a train's total effect.

Firstly we begin by considering the design matrix X for the simple model (23):

$$X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_t \end{pmatrix}.$$

Our coefficient estimates for model (23) thus have the form,

$$\hat{\beta} = (\mathbf{X^T X})^{\mathbf{-1}} \mathbf{X^T Y}, \tag{26}$$

where the assumption is that the errors, $\epsilon_t \sim N(0, \sigma^2)$ are i.i.d gaussian, and Y is a vector of of $(t \times 1)$ observations on the dependent variable. We now consider the expected value of $\hat{\beta}$ when the model (25) is considered true.

$$E[\hat{\beta}] = (X^T X)^{-1} X^T W \beta, \tag{27}$$

where,

$$W = \begin{pmatrix} 1 & g(x_1) \\ \vdots & \vdots \\ 1 & g(x_t) \end{pmatrix}.$$

Then,

$$\begin{aligned} X^T W &= \begin{pmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_t \end{pmatrix} \begin{pmatrix} 1 & g(x_1) \\ \vdots & \vdots \\ 1 & g(x_t) \end{pmatrix} \\ &= \begin{pmatrix} \sum_t 1 & \sum_t g(x_t) \\ \sum_t x_t & \sum_t x_t g(x_t) \end{pmatrix}. \end{aligned}$$

The expression above is for a general setting. However, as both our train and tail variables are indicators, then in this special case, where $g(x_t) = x_t + h(x_t)$, it is straightforward to show that

$$\sum_t g(x_t) = T_1 + T_2,$$

where T_1 corresponds to the total amount of time when a train is present and T_2 is the total amount of time a train's tail is present. From here we can see that

$$\sum_t x_t = \sum_t x_t g(x_t) = T_1.$$

This occurs as a train and a tail cannot be present at the same time, thus the product of a train and tail indicator results in the train only. We can thus consider $X^T W$ as

$$X^T W = \begin{pmatrix} T & T_1 + T_2 \\ T_1 & T_1 \end{pmatrix}$$

which we can rewrite as, $X^T W = X^T X + V$,

$$\begin{aligned} X^T W &= \begin{pmatrix} T & T_1 + T_2 \\ T_1 & T_1 \end{pmatrix} \\ &= \begin{pmatrix} T & T_1 \\ T_1 & T_1 \end{pmatrix} + \begin{pmatrix} 0 & T_2 \\ 0 & 0 \end{pmatrix} \\ &= X^T X + V. \end{aligned}$$

It then follows that

$$\begin{aligned} E[\hat{\beta}|X] &= (X^T X)^{-1} X^T W \beta \\ &= (X^T X)^{-1} \left(X^T X + V \right) \beta \\ &= (X^T X)^{-1} X^T X \beta + (X^T X)^{-1} V \beta \\ &= \beta + (X^T X)^{-1} V \beta, \end{aligned}$$

given that $(X^T X)^{-1} X^T X = I$.

Then,

$$\begin{aligned} (X^T X)^{-1} V \beta &= \begin{pmatrix} T & T_1 \\ T_1 & T_1 \end{pmatrix}^{-1} \begin{pmatrix} 0 & T_2 \\ 0 & 0 \end{pmatrix} \beta \\ &= \frac{1}{(T T_1 - T_1^2)} \begin{pmatrix} T_1 & -T_1 \\ -T_1 & T \end{pmatrix} \begin{pmatrix} 0 & T_2 \\ 0 & 0 \end{pmatrix} \beta \\ &= \frac{1}{T_1 (T - T_1)} \begin{pmatrix} T_1 & -T_1 \\ -T_1 & T \end{pmatrix} \begin{pmatrix} 0 & T_2 \\ 0 & 0 \end{pmatrix} \beta. \end{aligned}$$

We set $T_0 = (T - T_1)$, so that T_0 is equal to the total amount of time that there are no trains passing. We now continue the above equation,

$$\begin{aligned} (X^T X)^{-1} V \beta &= \frac{1}{T_1 T_0} \begin{pmatrix} 0 & T_1 T_2 \\ 0 & -T_1 T_2 \end{pmatrix} \beta \\ &= \begin{pmatrix} 0 & T_2 / T_0 \\ 0 & -T_2 / T_0 \end{pmatrix} \beta. \end{aligned}$$

Thus,

$$E[\hat{\beta}|X] = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} 0 & T_2/T_0 \\ 0 & -T_2/T_0 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix},$$

results in,

$$E[\hat{\beta}_0] = \beta_0 + T_2/T_0\beta_1 \tag{28}$$

$$E[\hat{\beta}_1] = \beta_1 - T_2/T_0\beta_1 \tag{29}$$

where, T_2/T_0 is the proportion of total time where there is a train tail in relation to the amount of time that there are no trains passing. From this we can see that the intercept, β_0 is biased up and the train covariate, β_1 is biased down by T_2/T_0 . Therefore, if we have a large tail effect that is not captured by a variable, then the model will be significantly affected.

We now consider the effect of assuming an incorrect tail length.

4.2.2 Assumed model: Short tail

In this section, we consider the expected estimates obtained from a linear regression model where we include an assumed tail. In the ideal situation we can correctly specify the correct length of the tail. Unfortunately, when we do not know the exact tail length, we can consider a model where we assume the tail. Here we explore the effect of assuming the tail length to be shorter than the true tail.
We fit the assumed model from equation (24), when the true model is,

$$y_t = \beta_0 + \beta_1 x_t + \beta_2 r_t + \epsilon_t. \tag{30}$$

Here the true and known train variable is x_t, and the assumed tail variable is u_t, while the true tail variable is r_t. For these calculations, we set the assumed tail to be half of the true tails length, so that $\sum_t u_t = 1/2 \sum_t r_t$. In the same manner as in the previous subsection, we have the design matrix for the assumed model,

$$X = \begin{pmatrix} 1 & x_1 & u_1 \\ \vdots & \vdots & \\ 1 & x_t & u_t \end{pmatrix},$$

and

$$\begin{aligned} X^T X &= \begin{pmatrix} \sum_t 1 & \sum_t x_t & \sum_t u_t \\ \sum_t x_t & \sum_t x_t^2 & \sum_t x_t u_t \\ \sum_t u_t & \sum_t x_t u_t & \sum_t u_t^2 \end{pmatrix} \\ &= \begin{pmatrix} T & T_1 & U_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & U_1 \end{pmatrix}. \end{aligned}$$

The W matrix is for the true model, where we known the true tail indicator, is r_t,

$$W = \begin{pmatrix} 1 & x_1 & r_1 \\ \vdots & \vdots & \vdots \\ 1 & x_t & r_t \end{pmatrix}.$$

Thus,

$$\begin{aligned} X^T W &= \begin{pmatrix} \sum_t 1 & \sum_t x_t & \sum_t r_t \\ \sum_t x_t & \sum_t x_t^2 & \sum_t x_t r_t \\ \sum_t u_t & \sum_t x_t u_t & \sum_t u_t r_t \end{pmatrix} \\ &= \begin{pmatrix} T & T_1 & R_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & U_1 \end{pmatrix}. \end{aligned}$$

Here $\sum_t u_t r_t = \sum_t u_t$ as $u_t < r_t$. Then, as $\sum_t u_t = \sum_t r_t/2$, $R_1 = U_1 + U_1$, and we have,

$$X^T W = X^T X + V$$

$$\begin{pmatrix} T & T_1 & R_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & U_1 \end{pmatrix} = \begin{pmatrix} T & T_1 & U_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & U_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & U_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Then the inverse of $(X^T X)$ is,

$$(X^T X)^{-1} = \frac{1}{T_1 U_1 (T - T_1 - U_1)} \begin{pmatrix} T_1 U_1 & -T_1 U_1 & -T_1 U_1 \\ -T_1 U_1 & T U_1 - U_1^2 & T_1 U_1 \\ -T_1 U_1 & T_1 U_1 & T T_1 - T_1^2 \end{pmatrix},$$

and thus we have,

$$\begin{aligned} (X^T X)^{-1} V \beta &= \frac{1}{T_1 U_1 (T - T_1 - U_1)} \begin{pmatrix} T_1 U_1 & -T_1 U_1 & -T_1 U_1 \\ -T_1 U_1 & T U_1 - U_1^2 & T_1 U_1 \\ -T_1 U_1 & T_1 U_1 & T T_1 - T_1^2 \end{pmatrix} \begin{pmatrix} 0 & 0 & U_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &= \frac{1}{T_1 U_1 (T - T_1 - U_1)} \begin{pmatrix} 0 & 0 & T_1 U_1^2 \\ 0 & 0 & -T_1 U_1^2 \\ 0 & 0 & -T_1 U_1^2 \end{pmatrix} \beta \\ &= \begin{pmatrix} 0 & 0 & U_1/(T - T_1 - U_1) \\ 0 & 0 & -U_1/(T - T_1 - U_1) \\ 0 & 0 & -U_1/(T - T_1 - U_1) \end{pmatrix} \beta, \end{aligned}$$

which results in,

$$E[\hat{\beta}_0] = \beta_0 + \beta_2 \frac{U_1}{(T - T_1 - U_1)}$$

$$E[\hat{\beta}_1] = \beta_1 - \beta_2 \frac{U_1}{(T - T_1 - U_1)}$$

$$E[\hat{\beta}_2] = \beta_2 - \beta_2 \frac{U_1}{(T - T_1 - U_1)}$$

The above expectations for each of the covariates from the assumed model, show that the $E[\hat{\beta}_0]$ is biased up, while the train and tail covariates are biased down. Here the bias is proportional to the difference between the total train and the tail length. This is a similar result as in the omitted variable model. However, the bias is reduced in comparison to the simple model in the previous subsection.
We now consider the effect of assuming the tail length to be greater than the true tail.

4.2.3 Assumed model: Long tail

In practice, we will not know the true tail length, but we can guess at an appropriate tail length which will be either shorter than the true tail, equal to the true tail, or greater than the true tail. We have covered the first of these options above. The second option results in a correct model fit and will not incur any additional bias. The third option, where we assume the tail to have a greater length than the true tail is considered here.
In the previous subsection we considered the effect of estimating a tail length that is half of the true tail. We now consider a tail that is one and a half greater than the true tail. This allows us to compare our expectations with the previous model. The same assumed model (23) and true model (30) are considered here. In this case, we have$\sum_t u_t = \frac{3}{2} \sum_t r_t$, which corresponds to the assumed tail being greater than the true tail.

Our design matrix X, is the same as in the previous subsection,

$$X = \begin{pmatrix} 1 & x_1 & u_1 \\ \vdots & \vdots & \\ 1 & x_t & u_t \end{pmatrix},$$

and

$$\begin{aligned} X^T X &= \begin{pmatrix} \sum_t 1 & \sum_t x_t & \sum_t u_t \\ \sum_t x_t & \sum_t x_t^2 & \sum_t x_t u_t \\ \sum_t u_t & \sum_t x_t u_t & \sum_t u_t^2 \end{pmatrix} \\ &= \begin{pmatrix} T & T_1 & U_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & U_1 \end{pmatrix}, \end{aligned}$$

and our W matrix has the same form as previously,

$$W = \begin{pmatrix} 1 & x_1 & r_1 \\ \vdots & \vdots & \vdots \\ 1 & x_t & r_t \end{pmatrix}.$$

However when we consider the expected value of the assumed model when the true model is correct,

$$
\begin{aligned}
X^TW &= \begin{pmatrix} \sum_t 1 & \sum_t x_t & \sum_t r_t \\ \sum_t x_t & \sum_t x_t^2 & \sum_t x_t r_t \\ \sum_t u_t & \sum_t x_t u_t & \sum_t u_t r_t \end{pmatrix} \\
&= \begin{pmatrix} T & T_1 & R_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & R_1 \end{pmatrix},
\end{aligned}
$$

we find that the bottom right result in the matrix above is now R_1. This is due to $\sum_t u_t r_t = \sum_t r_t$ as $r_t < u_t$. As we know that the true tail is shorter than the assumed tail, our product of the two tails results in the true tail remaining.

Continuing with our matrix calculations, when, in this special case that $\sum_t u_t = 3/2 \sum_t r_t$, and thereby $R_1 = U_1 - U_1/3$, we have,

$$
X^TW = X^TX + V
$$

$$
\begin{pmatrix} T & T_1 & R_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & R_1 \end{pmatrix} = \begin{pmatrix} T & T_1 & U_1 \\ T_1 & T_1 & 0 \\ U_1 & 0 & U_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & -U_1/3 \\ 0 & 0 & 0 \\ 0 & 0 & -U_1/3 \end{pmatrix},
$$

$$
(X^TX)^{-1} = \frac{1}{T_1U_1(T - T_1 - U_1)} \begin{pmatrix} T_1U_1 & -T_1U_1 & -T_1U_1 \\ -T_1U_1 & TU_1 - U_1^2 & T_1U_1 \\ -T_1U_1 & T_1U_1 & TT_1 - T_1^2 \end{pmatrix},
$$

then,

$$
\begin{aligned}
(X^TX)^{-1}V\beta &= \frac{1}{T_1U_1(T - T_1 - U_1)} \begin{pmatrix} 0 & 0 & -T_1U_1^2/3 + T_1U_1^2/3 \\ 0 & 0 & 0 \\ 0 & 0 & T_1U_1^2/3 - U_1(TT_1 - T_1^2)/3 \end{pmatrix} \beta \\
&= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{U_1 - T + T_1}{3(T - T_1 - U_1)} \end{pmatrix} \beta,
\end{aligned}
$$

which gives us,

$$
\begin{aligned}
E[\hat{\beta}_0] &= \beta_0 \\
E[\hat{\beta}_1] &= \beta_1 \\
E[\hat{\beta}_2] &= \beta_2 + \beta_2 \frac{U_1 - T + T_1}{3(T - T_1 - U_1)}
\end{aligned}
$$

Where, as long as $U_1 + T_1 < T$ this will be negative. That is, as long as the total assumed tail and the total train is less than the total observations, the bias will be negative for $E[\hat{\beta}_2]$.

Therefore, we again have the $E[\hat{\beta}_2]$ being biased down as in the previous two considerations. However the $E[\hat{\beta}_0]$ and $E[\hat{\beta}_1]$ no longer produce biased estimates. This is an interesting outcome, and it suggests, that not only are we better off by including a proxy variable rather than omitting the variable, but also, that by assuming the tail to be longer than the true tail, we are able to remove the bias in our estimators of $\hat{\beta}_0$ and $\hat{\beta}_1$. In the following simulation study we assess whether these theoretical expectations hold.

4.3 Simulations

In chapter 3, we showed in our analysis of the coal train data, that as we increased the aggregation period, the coefficient estimates for each passing train type increased. This effect was not expected, and it lead us to believe that we may have omitted a tail effect for each passing train type. It is the goal of this chapter to examine the reasons for this systematic increase in the coefficient estimates. We believe that there is an omitted tail effect due to the increase in coefficient estimates, and that our simple model, as in equation (23), where we have no train tail covariates, induces bias. In our previous section we showed that if there is a tail effect, than a simple model will be biased. Furthermore, we showed that under the assumption of a tail, as in the assumed model from equation (24), we can reduce the bias by assuming a tail variable.

We conduct a simulation study to explore the effects of an incorrectly specified model. Firstly, we describe our data generating model in section 4.3.1. Here we create a 'true' model, where we have both a passing train and its passing tail. We assume two error structures for this data generating model. In first case we have iid errors, as this allows us to test our theory from the previous section. We then include long memory errors, which allows us to more closely mimic the coal train data in our application.

In the previous section 4.2, we showed theoretically, that the omission of a tail variable results in a bias in the estimators of $\hat{\beta}_0$ and $\hat{\beta}_1$ in the linear regression for the simple model from equation (23). We also considered the effect of under or over specifying a tail variable, and the resulting bias for these coefficient estimates. Using our simulations we will see that these expected values match the estimated values.

We then move on to comparing the simple and assumed models against the known true model. Here we consider these three models for a number of tail lengths. This comparison shows that by assuming a tail, of reasonable length, we can improve the analysis against the case where we omit a tail variable.

Throughout this thesis, we are utilising temporal aggregation due to the presence of long memory in our applied dataset, and the difficulties we face in the estimation of models suitable for such time series data. We continue our simulation study from here on, by generating our simulated data with long memory. In subsection 4.3.7, we explore the effect of temporal aggregation on our previous three models along with a new fourth model. This fourth model we call the combined model,

$$y_t = \beta_0 + \beta_1 v_t + \epsilon_t, \tag{31}$$

where $v_t = x_t + u_t$. This model is an extension of the assumed model, whereby we combine the train and assumed tail variables into one. We are interested to see how this model impacts on the coefficient estimates. The comparison of the four models, and specifically their coefficient estimates, over a range of aggregation periods, is conducted for a number of reasons. Firstly, we are interested if the coefficient estimates will systematically increase as we increase the aggregation period, as in the application analysis in chapter 3. Secondly, if we specify the correct tail, how will the coefficient estimates react. Thirdly, we are interested in how the temporally aggregated data compares to the unaggregated data. And finally, are we able to capture the same effects for both datasets. These question are explored in this subsection.

Through our simulation study, we are able to show that the true and assumed model (when the tail is correctly specified) estimates for both passing train and tail covariates, are not affected by

temporal aggregation. This finding allows us to suggest a tail length for our assumed model by comparing a number of tail lengths.
We conclude our simulation with a review of our methods and findings.

4.3.1 Data generating model

We generate our data from the following model:

$$y_t = \beta_0 + \beta_1 x_t + \beta_2 r_t + \epsilon_t, \tag{32}$$

where ϵ_t is an error term discussed below. Each simulation has a length of $t = 1, \ldots, 5000$. We are limiting this length for computational reasons, as regression with ARFIMA errors takes a number of hours for data of length greater than 5000, and furthermore, is not possible for data of length greater than 12,000 observations. We consider two cases for the errors. To simulate a long memory process we set the errors, $\epsilon_t \sim ARFIMA(3, 0.4, 3)$, with $\eta_t \sim N(0, \sigma_\eta^2)$ and $\sigma_\eta^2 = 0.4$. We also consider the case where the error $\epsilon_t \sim N(0, \sigma_\epsilon^2)$ are independent and identically distributed. In this case $\sigma_\epsilon^2 = 1$. The selection of an ARFIMA(3,0.4,3) error structure to instigate a long memory process was chosen as the short term component of ARMA(3,3) allows for the effect of aggregation to affect the short term memory, and the long memory component given by the fractoral differencing parameter $d = 0.4$ results in a long memory process that can be, although is more often than not, affected by temporal aggregation. This selection was made by trial and error such that the error structure cold be investigated by a number of models.

The passing train indicator variable x_t, illustrated in figure 4.1, is assumed to have length 10. The true tail variable, r_t has length of 10 observations, and begins immediately after a passing train. It is also set to be an indicator variable. Having created an error structure, and the passing trains and tails, we set our dependent variable y_t as in our data generating equation as noted in (32) with $\beta_0 = 3$ and $\beta_1 = \beta_2 = 5$.

The aggregation process is the same as outlined in chapter 3 and is continued throughout this thesis. We provide a brief overview here. For more detail please refer back to Chapter 3, Section 2.

For a time series of length $t = 1, \ldots, T$, we split the data into non-overlapping subsets of size J. We then take the mean for each subset. The aggregated data now has length $M = T/J$. In the event that M is not an integer, we remove a sufficient number of the final observations such that T/J is an integer. For the covariates, we apply the same method and take a mean. For an indicator variable as in these simulations and the data at large, the mean can be interpreted as a proportion of time a train (or tail) has spent passing in each block. More precisely, the variables, $\{y_t\}_{t=1}^T$ are transformed into $\{s_m\}_{m=1}^M$ as shown below,

$$s_m = \frac{1}{J} \sum_{t=(m-1)J+1}^{mJ} y_t \tag{33}$$

The independent variables are transformed, for the train variable $\{x_t\}_{t=1}^T$, for the true tail variable $\{r_t\}_{t=1}^T$, and for the assumed tail variable $\{u_t\}_{t=1}^T$,in the same manner as below,

$$z_m = \frac{1}{J} \sum_{t=(m-1)J+1}^{mJ} x_t. \tag{34}$$

We now present our simulation study.

4.3.2 Evaluating our theoretical results: Unaggregated data

In section 4.2 we considered the impact of an omitted variable on a linear regression. We then explored the effect of assuming a tail. Here we continue these calculations by simulating the simple and assumed models and comparing the expected results from the theory, and the estimated outcomes from the simulations.

4.3.3 Simple model: No tail

In section 4.2.1 we showed that the expected values for the estimated coefficients $\hat{\beta}$ from the simple model were,

$$E[\hat{\beta}_0] = \beta_0 + T_2/T_0\beta_1,$$

and

$$E[\hat{\beta}_1] = \beta_1 - T_2/T_0\beta_1,$$

where, T_2/T_0 is the ratio of total time where there is a train tail in relation to the amount of time that there are no trains passing.
Table 4.1 shows the results from 5 different simulations based on generating data from model (32) assuming independent errors. For each simulation, we compute T_0 and T_1, so that the expected value of each estimated regression coefficient can be computed. The table shows a comparison between these expected values and the estimated values of the intercept and slope parameters.

Table 4.1: Outcomes for the simple model in (23). The set value is the simulated value from the data generating model. The expected value is what we expect the estimate to be based off of the algebra above and the estimated value is the simulated estimate from our analysis. All simulations have an iid error structure. We include the means of all the simulation outcomes in the final row.

Sim No.	Set	Expected	Estimated	Set	Expected	Estimated
	β_0	$\hat{\beta}_0$	$\hat{\beta}_0$	β_1	$\hat{\beta}_1$	$\hat{\beta}_1$
1	3.00	3.59	3.67	5.00	4.41	4.28
2	3.00	3.63	3.71	5.00	4.37	4.28
3	3.00	3.62	3.69	5.00	4.38	4.29
4	3.00	3.54	3.61	5.00	4.46	4.44
5	3.00	3.52	3.58	5.00	4.48	4.42
Mean	3.00	3.58	3.65	5.00	4.22	4.34

Table 4.1 suggests that our theoretical outcomes are generally consistent with our estimated results. It is interesting to note that the expected values of the two coefficients vary from simulation to

simulation because of the variable values of T_0 (total amount of time that there are no trains passing) and T_2 (total amount of train tails). Because of the way we generate the covariates corresponding to the passing train and associated tail, it is not straightforward to obtain an analytic expression for the expected values of these quantities.

4.3.4 Assumed model: Short tail

We have also shown that we can estimate the bias on the assumed model (23) when we assume the tail length to be shorter than the true tail. Our calculations in section 4.2.1 determined these to be,

$$E[\hat{\beta}_0] = \beta_0 + \beta_2 \frac{U_1}{(T - T_1 - U_1)}$$
$$E[\hat{\beta}_1] = \beta_1 - \beta_2 \frac{U_1}{(T - T_1 - U_1)}$$
$$E[\hat{\beta}_2] = \beta_2 - \beta_2 \frac{U_1}{(T - T_1 - U_1)}$$

Here U_1 is the total of the assumed tails. T is the total number of observations and T_1 is the total amount of passing trains. We present the outcomes of this simulation in the following table 4.2.

Table 4.2: Assumed model with assumed tail length half of the true tail for the $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$ estimates. The set value is the simulated value from the data generating model. The expected value is what we expect the estimate to be based off of the algebra above and the estimated value is the simulated estimate from our analysis. All simulations have an iid error structure. We include the means of all the simulation outcomes in the final row.

Sim No.	Set β_0	Expected $\hat{\beta}_0$	Estimated $\hat{\beta}_0$	Set β_1	Expected $\hat{\beta}_1$	Estimated $\hat{\beta}_1$	Set β_2	Expected $\hat{\beta}_2$	Estimated $\hat{\beta}_2$
1	3.00	3.40	3.40	5.00	4.60	4.51	5.00	4.60	4.60
2	3.00	3.43	3.42	5.00	4.57	4.62	5.00	4.57	4.60
3	3.00	3.37	3.37	5.00	4.63	4.67	5.00	4.63	4.66
4	3.00	3.30	3.31	5.00	4.70	4.70	5.00	4.70	4.72
5	3.00	3.39	3.40	5.00	4.61	4.57	5.00	4.61	4.59
Mean	3.00	3.38	3.38	5.00	4.62	4.61	5.00	4.62	4.63

Table 4.2 shows that the bias is induced by incorrectly assuming the tail length. Comparing tables 4.1 and 4.2, for the simple and assumed models, we can see that the bias is reduced for both $\hat{\beta}_0$ and $\hat{\beta}_1$ in the assumed model. This is to be expected as we are now only omitting half of the trains tail. In the event of a present tail effect, assuming a tail that is shorter than the true tail results in a reduction in the bias as compared to the simple model with no tail. However, if we do not know the true tail, then we can also over assume the tail length. We now consider this.

4.3.5 Assumed model: Long tail

To conclude section 4.2.1 we presented some theory on the effect of assuming a tail length to be greater than the true tail for a linear regression with iid errors. We set the assumed tail to be 1.5 times that of the true tail. Our calculations showed that the expectation of $\hat{\beta}_2$ resulted in an introduction of bias, as shown below.

$$E[\hat{\beta}_0] = \beta_0$$

$$E[\hat{\beta}_1] = \beta_1$$

$$E[\hat{\beta}_2] = \beta_2 + \beta_2 \frac{U_1 - T + T_1}{3(T - T_1 - U_1)}$$

with U_1 := Total amount of assumed tail, T := Total observations, and T_1 := total amount of passing trains.
In table 4.3 we show that the estimated outcomes from our simulations are consistent with our theory.

Table 4.3: Assumed model with assumed tail length greater than the true tail for the $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$ estimates. The set value is the simulated value from the data generating model. The expected value is what we expect the estimate to be based off of the algebra above and the estimated value is the simulated estimate from our analysis. All simulations have an iid error structure. We include the means of all the simulation outcomes in the final row.

Sim No.	Set	Expected	Estimated	Set	Expected	Estimated	Set	Expected	Estimated
	β_0	$\hat{\beta}_0$	$\hat{\beta}_0$	β_1	$\hat{\beta}_1$	$\hat{\beta}_1$	β_2	$\hat{\beta}_2$	$\hat{\beta}_2$
1	3.00	3.00	3.02	5.00	5.00	5.02	5.00	3.33	3.30
2	3.00	3.00	3.05	5.00	5.00	4.86	5.00	3.33	3.31
3	3.00	3.00	3.00	5.00	5.00	4.93	5.00	3.33	3.37
4	3.00	3.00	3.01	5.00	5.00	4.97	5.00	3.33	3.32
5	3.00	3.00	3.03	5.00	5.00	5.00	5.00	3.33	3.28
Mean	3.00	3.00	3.02	5.00	5.00	4.96	5.00	3.33	3.32

Theoretically, we showed that the $\hat{\beta}_0$ and $\hat{\beta}_1$ were not biased for the assumed model when the assumed tail is greater than the true tail. The simulations shown in table 4.3 suggest that this effect is true, as only our $\hat{\beta}_2$ estimates for the tail are biased. We will come back to this point later, but the results have important implications in practice.

4.3.6 Comparison of simple and assumed models against the true model: Independent error structure

We begin our simulations having generated our data from the model in equation (32). For these present simulations we consider the ϵ_t to have a gaussian iid error structure and the train and tail have an equal effect, given by $\beta_1 = \beta_2$.

In table 4.4 we compare the $\hat{\beta}_0$ outcomes for the three models as described in equations (23), (24) and (25) above. The middle row shows the true known model. Here we have set $\beta_0 = 3$. It is evident that the coefficient estimates for the true model are accurate and consistent for the unaggregated. The middle row shows the results for all three models when we assume the tail to have a length of 10 observations, which is the same as the true model. Thus the assumed model has accurate estimates for $\hat{\beta}_0$. However we can see that the simple model, which does not include a tail variable is biased high. This result is consistent with the outcomes of our theory in the previous section. We can also see that the assumed model has lower bias than that of the simple model. All estimates for the assumed model are closer to the true value of $\beta_0 = 3$ than those of the simple model. Thus, we can see that the inclusion of a reasonable tail variable improves our model fit.

Table 4.4: Estimated ($\hat{\beta}_0$) from the simple, true and assumed mode from data generated from equation (32) with iid gaussian error structure. For the assumed model, we consider 3 possible tail covariates. The first column has estimates for the assumed tail of length 5 observations, the 2nd column has an assumed tail of 10 observations, which is equal to the true tail length. The final column shows the estimates for the assumed model with a tail length set to 15 observations. We compare all three assumed tail lengths with the simple model (no tail covariate), and the true model where the tail length is known.

Model	Assumed Tail Length is 5	Assumed Tail Length is 10	Assumed Tail Length is 15
Simple	3.63	3.62	3.62
True	3.00	2.99	2.99
Assumed	3.34	2.99	3.00

In table 4.5, we consider the coefficient estimates for the train effect, $\hat{\beta}_1$, and the tail effect, $\hat{\beta}_2$. Again, we compare all three models. The true model is accurate again for the unaggregated data. Comparing the first and the final two rows, namely the simple and assumed models, shows that the simple model, while not capturing the tail effect at all, is biased for the $\hat{\beta}_1$ coefficient estimate. The assumed model is also biased, however it is more accurate than the simple model. This shows that the assumed model again outperforms the simple model.

For the assumed model, we can see that the estimates are greatly affected by the assumption of the tail variables length. When we assume the tail to be 5 observations long, the estimate of the tail covariate, $\hat{\beta}_2 = 4.69$ has a smaller bias than when the tail is assumed to be 15 observations long, $\hat{\beta}_2 = 3.33$. The train coefficient estimates for $\hat{\beta}_1$, are less biased in the case when we assume the tail to be longer than the true tail. These results are consistent with our theoretical results in the previous section. From table 4.5 we can see that the assumed model has a reduced bias in comparison to the simple model.

Table 4.5: Estimated $\hat{\beta}_1$ and $\hat{\beta}_2$ from models 1-3 from data generated from equation (32) with iid gaussian error structure. For the assumed model, we consider 3 possible tail covariates. The first column has estimates for the assumed tail of length 5 observations, the 2nd column has assumed tail of 10 observations, which is equal to the true tail length. The final column shows the estimates for the assumed model with a tail length set to 15 observations. We compare all three assumed tail lengths with the simple model (no tail covariate), and the true model where the tail length is known.

Model	Assumed Tail Length is 5	Assumed Tail Length is 10	Assumed Tail Length is 15
Simple $\hat{\beta}_1$	4.38	4.38	4.35
True $\hat{\beta}_1$	5.01	5.02	4.98
True $\hat{\beta}_2$	5.00	5.03	5.00
Assumed $\hat{\beta}_1$	4.68	5.02	4.93
Assumed $\hat{\beta}_2$	4.69	5.03	3.33

4.3.7 Effect of temporal aggregation on simple, assumed, combined and true models: Long memory error structure

In our applied analysis in chapter 3, we found that the coefficient estimates systematically increased as we increased the aggregation period. In the following two tables 4.6 and 4.7, we explore the simple, assumed, true and combined models for a number of aggregation periods. We consider both unaggregated and aggregated data. The goal of this simulation study is to explore the impact of aggregation on a correct and incorrect tail specification. As our coal train data has a long memory dependence structure, for these simulations we implement ARFIMA errors to account for the long memory process that we generate.

In table 4.6 we explore the intercept estimates for our four models. The true set value of our simulations is $\beta_0 = 3$. The simple model is as in equation (23), the assumed model from equation (24) and the combined model from equation (31). From our previous simulations, we showed that the bias resulting from an omitted variable under the simple model, results in a positive increase in the bias for $\hat{\beta}_0$. This bias is consistent for all the aggregation periods. The assumed model has a lower bias than that of the simple model. This is to be expected in each of the 4 assumed tail length options we consider here. If we were to assume the tail to be of length 100, when the true tail has length 10, then the assumed model would be less accurate than the simple model. Thus, the assumed tail must be of a reasonable length, in relation to the true, or in the application case, the possible tail length. The choice of a reasonable length can be selected based off of knowledge of a true trains length, or by trial and error as we implement in this section. The combined model appears to follow the assumed model, which is to be expected as it is an extension of it. We can see from the table 4.6, that the intercept estimate is slightly more accurate for the combined model than the simple model. The main message from table 4.6, is that as we increase the aggregation period from $J = 5$ to $J = 40$, the coefficient estimates remain constant for the assumed model if we have assumed the correct tail length. This is a critical observation for this chapter, as it allows us to suggest a tail length in our coal train application.

Table 4.6: Coefficient estimates for ($\hat{\beta}_0$) from the true, simple, assumed and combined models, with data generated from equation (32) with long memory error structure. We consider 4 possible tail covariates. The first two columns have estimates for the assumed tail of length 5 observations, the 3rd and 4th columns have assumed tail of 10 observations, which is equal to the true tail length. The 5th and 6th columns show the estimates for the assumed model with a tail length set to 15 observations. The final two columns show the estimates for an assumed tail of length 20 observations. We compare all four assumed tail lengths for the simple model (no tail covariate), the assumed model, the combined model and the true model where the tail length is known. UA is short for the unaggregated data and AGG is the aggregated data, at aggregation period J.

J	Model	Assumed Tail		Assumed Tail		Assumed Tail		Assumed Tail	
		Length is 5		Length is 10		Length is 15		Length is 20	
		UA	AGG	UA	AGG	UA	AGG	UA	AGG
5	Simple	3.91	3.95	3.92	3.90	3.80	3.83	3.70	3.72
	True	3.06	3.06	3.04	3.04	3.01	3.03	2.92	2.93
	Assumed	3.69	3.65	3.04	3.04	3.39	3.36	3.20	3.08
	Combined	3.69	3.71	3.04	3.01	3.31	3.42	3.11	3.08
10	Simple	3.80	3.79	3.81	3.76	3.88	3.82	3.48	3.54
	True	2.99	2.99	2.96	2.96	3.03	3.06	2.62	2.70
	Assumed	3.61	3.34	2.96	2.96	3.48	2.95	2.99	2.85
	Combined	3.68	3.36	2.96	2.94	3.40	2.96	2.89	2.92
20	Simple	3.95	3.47	3.87	3.36	3.84	3.29	3.85	3.47
	True	3.08	3.08	3.02	3.07	2.97	2.87	3.09	3.11
	Assumed	3.80	3.26	3.02	3.07	3.39	2.68	3.39	2.88
	Combined	3.76	3.28	3.09	3.07	3.33	2.73	3.31	3.06
30	Simple	4.02	3.35	3.82	3.20	3.94	3.33	4.00	3.39
	True	3.14	3.08	3.02	2.96	3.05	3.09	3.16	3.08
	Assumed	3.85	3.18	3.02	2.96	3.59	3.02	3.51	2.96
	Combined	3.86	3.19	3.00	2.94	3.45	3.02	3.40	3.05
40	Simple	3.80	3.05	3.85	3.18	3.87	3.17	3.91	3.28
	True	2.91	2.88	3.04	2.99	3.34	3.02	3.05	3.06
	Assumed	3.57	2.91	3.04	2.99	3.46	3.00	3.38	3.03
	Combined	3.59	2.95	3.09	2.99	3.35	2.93	3.33	3.10

Turning to the coefficient estimates for the train and tail, $\hat{\beta}_1$ and $\hat{\beta}_2$, under the simple, assumed, combined and true models, we present the table 4.7. As in the intercept estimates, the simple model has the largest bias in comparison to the assumed and combined models. Furthermore, as we increase the aggregation period the estimates for the simple model increase. At $J = 5$, the estimate for $\hat{\beta}_1 = 2.31$, and this increases to $\hat{\beta}_1 = 6.45$ at $J = 10$, and further increases to $\hat{\beta}_1 = 8.44$ at $J = 40$.

We are particularly interested in the impact of aggregation on the assumed model. We can see from the fourth and fifth columns, that the assumed model estimates remain constant as we increase the aggregation period. This is the same effect as for the intercepts in our previous table. In the event that we do not correctly specify the tail length for the assumed model, the estimates for both $\hat{\beta}_1$ and $\hat{\beta}_2$ vary. This is temporal aggregation effect is a useful feature in our tail length selection in the coal train application. A particularly interesting piece of information from table 4.7 is that the combined model estimates for $\hat{\beta}_1$ appear to be almost an average of the $\hat{\beta}_1$ and $\hat{\beta}_2$ estimates from the assumed model. Nonetheless, if we correctly assume the tail length, then the combined model also results in a consistent and accurate estimate for each aggregation period.

Table 4.7: Coefficient estimates for $\hat{\beta}_1$ and $\hat{\beta}_2$ from models 1-4 from data generated from equation (32) with long memory error structure. We consider 4 possible tail covariates. The first two columns have estimates for the assumed tail of length 5 observations, the 3rd and 4th columns have assumed tail of 10 observations, which is equal to the true tail length. The 5th and 6th columns show the estimates for the assumed model with a tail length set to 15 observations. The final two columns show the estimates for an assumed tail of length 20 observations. We compare all four assumed tail lengths for the simple model (no tail covariate), the assumed model, the combined model and the true model where the tail length is known. UA is short for the unaggregated data and AGG is the aggregated data, at aggregation period J.

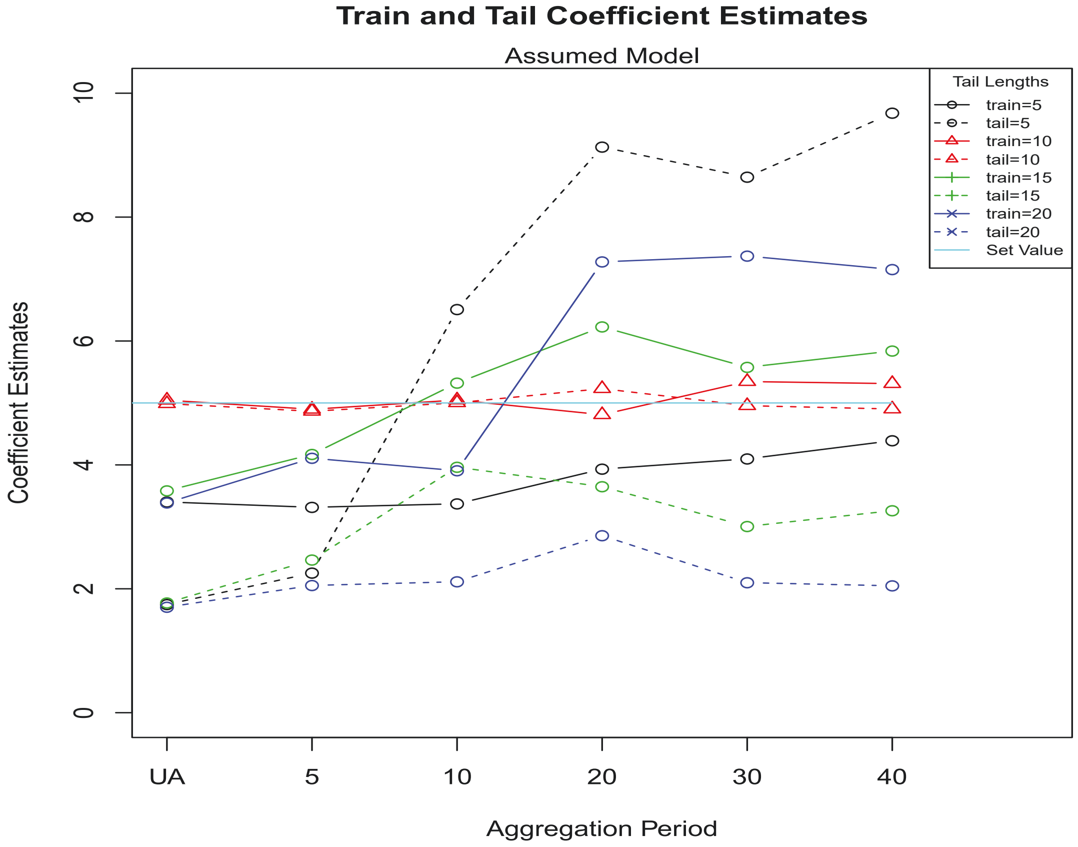
J	Model	Assumed Tail Length is 5		Assumed Tail Length is 10		Assumed Tail Length is 15		Assumed Tail Length is 20	
		UA	AGG	UA	AGG	UA	AGG	UA	AGG
5	Simple $\hat{\beta}_1$	2.55	2.31	2.56	2.59	2.55	2.44	2.57	2.53
	True $\hat{\beta}_1$	5.04	5.01	5.02	5.08	5.02	5.00	5.02	5.01
	True $\hat{\beta}_2$	5.03	5.05	4.96	4.97	4.99	4.96	4.95	4.93
	Assumed $\hat{\beta}_1$	3.40	3.33	5.02	5.08	3.55	4.24	3.49	4.11
	Assumed $\hat{\beta}_2$	1.77	2.21	4.96	4.97	1.91	2.72	1.74	2.12
	Combined $\hat{\beta}_1$	2.71	2.93	4.99	5.03	2.70	3.18	2.60	2.79
10	Simple $\hat{\beta}_1$	2.45	2.80	2.50	2.65	2.53	2.89	2.58	2.93
	True $\hat{\beta}_1$	4.94	4.98	5.01	4.93	4.99	5.01	5.03	5.07
	True $\hat{\beta}_2$	5.05	5.16	5.04	4.94	4.94	4.96	4.95	4.97
	Assumed $\hat{\beta}_1$	3.32	3.56	5.01	4.93	3.50	5.25	3.48	4.35
	Assumed $\hat{\beta}_2$	1.79	6.45	5.04	4.94	1.88	3.80	1.72	2.04
	Combined $\hat{\beta}_1$	2.69	4.30	5.02	5.01	2.74	4.27	2.58	2.39
20	Simple $\hat{\beta}_1$	2.49	6.21	2.59	7.01	2.55	6.96	2.56	6.77
	True $\hat{\beta}_1$	4.90	5.13	5.10	5.23	5.03	5.41	5.04	5.15
	True $\hat{\beta}_2$	4.91	4.84	5.05	4.95	5.02	5.11	4.99	4.71
	Assumed $\hat{\beta}_1$	3.31	3.87	5.10	5.23	3.53	6.74	3.46	7.24
	Assumed $\hat{\beta}_2$	1.71	9.00	5.05	4.95	1.91	3.59	1.72	2.62
	Combined $\hat{\beta}_1$	2.64	5.39	5.08	5.10	2.76	4.72	2.58	3.63
30	Simple $\hat{\beta}_1$	2.50	8.00	2.53	8.23	2.51	7.56	2.52	7.84
	True $\hat{\beta}_1$	4.96	5.65	4.94	5.11	4.98	4.60	4.97	5.16
	True $\hat{\beta}_2$	4.97	4.61	4.97	5.18	4.98	5.06	4.97	5.18
	Assumed $\hat{\beta}_1$	3.34	5.04	4.94	5.11	3.50	5.60	3.43	6.85
	Assumed $\hat{\beta}_2$	1.74	8.77	4.97	5.18	1.88	3.06	1.75	2.28
	Combined $\hat{\beta}_1$	2.68	6.25	4.95	5.18	2.78	4.08	2.59	3.56
40	Simple $\hat{\beta}_1$	2.49	8.11	2.52	8.16	2.49	8.35	2.55	8.24
	True $\hat{\beta}_1$	4.94	5.01	4.96	5.22	5.00	4.96	5.02	4.43
	True $\hat{\beta}_2$	4.95	4.61	4.97	4.86	4.74	4.74	5.01	5.57
	Assumed $\hat{\beta}_1$	3.35	4.96	4.96	5.22	3.44	5.33	3.53	6.17
	Assumed $\hat{\beta}_2$	1.80	8.44	4.97	4.86	1.82	3.14	1.84	1.99
	Combined $\hat{\beta}_1$	2.65	5.99	4.96	5.03	2.68	3.96	2.63	3.27

Tables 4.6 and 4.7 provide a valuable insight that we utilise extensively in the following simulation section and in the application of the coal train data. As we increase the aggregation period, the simple model estimates will increase, and the assumed model estimates will also be variable. If we are able to correctly assume the tail length, then the coefficient estimates will remain accurate for each aggregation period.

4.3.8 Selection of tail length for the assumed model

As expected, our simulations suggest that if we know the correct tail length then we will have unbiased results for both β_1 and β_2 for the unaggregated and aggregated cases. In practice, of course, we do not know the true tail length and will need to assume a value instead. Our previous simulations have shown that assuming a tail length will produce results that have a lower bias than ignoring the tail. Furthermore, if we correctly assume the tail length, then temporal aggregation will not affect the outcomes. This presents an interesting point, namely, that if we assume the correct tail length then the coefficient estimates will be consistent over any reasonable aggregation period. We thus consider a simulation study to compare different assumed tail lengths for a range of aggregation periods. With the true tail length set to be ten observations in length, we consider tails with a length of 5, 10, 15, and 20 observations. These simulations are generated with long memory dependence and as such we consider a regression with ARFIMA errors for the assumed model.

Figure 4.2: Comparison of alternate tail lengths under the assumed model with ARFIMA errors on simulated data with long memory and $\beta_1 = \beta_2 = 5$. We analyse the unaggregated data and then we consider the aggregated data for the aggregation periods from J=5 to J=40. The solid lines show the $\hat{\beta}_1$ estimates which correspond to the train effect, and the dotted lines show the $\hat{\beta}_2$ estimates which correspond to the assumed train tails. The true set value is shown by the cyan line.



In figure 4.2 we fit the assumed model as in equation (24) to the data generated from model equation (32). We consider tail lengths for 5 observations to 20 observations. By comparing the results for these simulations over the unaggregated data and aggregation periods of 5 to 40 observations, we can make an inference about the length of tail. The true tail length is 10 observations. The red lines, show the results for $\hat{\beta}_1$ and $\hat{\beta}_2$ for the train and tail covariates when we assume the tail to have length of 10 observations. In this event, both $\hat{\beta}_1$ and $\hat{\beta}_2$ are consistent over all aggregation periods.

The assumption of a tail that is incorrect, such as of length 5, 15 or 20, results in a systematic increase in the coefficient estimates as we increase the aggregation period.

These results suggest that, by fitting a series of models with different tail lengths over a number of levels of aggregation, we can estimate the correct tail length. When the coefficient estimates change systematically as we increase the aggregation level, this is an indication that an incorrect model is being fitted. However, when the model is consistent over different aggregation periods, there is evidence that the correct model has been fitted.

4.3.9 Review of simulations

The goal of our simulation study has been to explore the impact of a misspecified model. We have compared four different classes of model to consider this. The comparison of the simple, assumed and combined model classes against the true model has yielded a number of interesting effects. The simple model, where we do not have a tail effect included in our regression, results in biased estimates. As we increase our aggregation period, these estimates systematically increase. This finding is the same as our analysis in chapter 3, and suggests that the omitted variable is the cause of this effect. We therefore consider the assumed model.

In the assumed model, we suggest a tail length and include this variable in the regression. Comparing the coefficient estimates against a true known tail, we are able to show that if we assume the tail to be close to the true tail length, then we will have consistent estimates for both unaggregated and aggregated data. This effect, that the temporal aggregation does not impact the accuracy of the estimates as we increase the temporal aggregation, suggests that we can use the temporally aggregated data at different J aggregation periods to assume a tail length. When we correctly assume the tail length the coefficient estimates do not vary, unlike the other assumed tail lengths. We are able to use the findings in this simulation study to improve our modelling strategy in our following application analysis.

4.4 Application: Hunter Valley Coal Train dataset

Our previous analysis of the coal train data in chapter 3, revealed that as we increased the aggregation period, the coefficient estimates for each passing train type tended to increase systematically. It is our hypothesis that this increase is due to an omitted variable, which we consider to be a tail effect occurring in the period after a train has passed the monitor. To understand this effect, the focus of this chapter has been to understand the misspecification of a linear regression for this data. In our simulations we have shown that in the event of a tail effect, we can improve our model by assuming a tail variable instead of ignoring it. We now implement this theory on the coal train data.

We begin our analysis by comparing the simple model as shown below in equation (35), that of no train tail variables, with the assumed model for the coal train data in equation (36) below,

$$\begin{aligned} s_m = \beta_0 &+ \beta_1 EmptyCoalRate_m + \beta_2 FreightRate_m + \beta_3 LoadedCoalRate_m \\ &+ \beta_4 PassengerRate_m + \beta_5 UnknownRate_m + \epsilon_m, \end{aligned} \tag{35}$$

and the new assumed model implementing tails is,

$$\begin{aligned} s_m = \beta_0 &+ \beta_1 EmptyCoal_m + \beta_2 EmptyCoalTail_m + \beta_3 Freight_m \\ &+ \beta_4 FreightTail_m + \beta_5 LoadedCoal_m + \beta_6 LoadedCoalTail_m \\ &+ \beta_7 Passenger_m + \beta_8 PassengerTail_m + \beta_9 Unknown_m \\ &+ \beta_{10} UnknownTail_m + \epsilon_m. \end{aligned} \tag{36}$$

In both equations (35) and (36) we apply a linear regression with both iid and ARFIMA(p,d,q) errors. The dependent variable, s_m, is the aggregated mean of the log(TSP+1) air quality observations, where TSP is the Total Suspended Particulate matter as observed in our Hunter Valley Coal Train dataset. A more detailed review of this data can be found in Chapter 2 of this thesis. Each of the covariates, both for the passing trains and the suggested tails are also temporally aggregated to means, so m refers to a value computed for interval m. We continue with temporal aggregation as utilised throughout this chapter and also more thoroughly covered in chapter 2. These covariates are thereby transformed from an indicator to a proportion of time a train and/or tail is present in each aggregation interval. Our data has length M, where $M = N/J$. Here N is the total length of the unaggregated data, and J is the length of the aggregation block, i.e. for 5 minute aggregation, $J = 50$. In the case that N is not divisible by J, we simply remove the needed number of observations from the end of the series so that N becomes divisible by J and hence that M is an integer. We are limited to the lower aggregation period of 5 minutes, as at this point our unaggregated dataset, with over $600,000$ observation is reduced to $12,000$ observations. However, the method of a linear regression with ARFIMA errors is currently unable to be estimated for data with over $12,000$ to $15,000$ observations due to memory constraints, that occur as a result of storing and inverting the covariance matrix with a long memory dependence. We cover this reasoning in chapter 3 of this thesis.

From this initial analysis, by comparing the results for the models (35) and (36), we will see that indeed the data support the idea that a tail effect is present. We then shift our focus to determining the tail length for each train type. We consider a tail length from 1 minute to 5 minutes in length. In our simulations in section 4.3.8, we showed that if we assume a tail length that is correct, the coefficient estimates for the tail variable remain consistent as we change our aggregation period. This constant effect across aggregation periods continues in our application. We are thereby able to suggest a tail length for each train type, by analysing the tail estimates for each aggregation period. We complete our analysis with the implementation of what we call the combined model, whereby we combine the train and assumed tail variables into one variable and explore the outcomes.

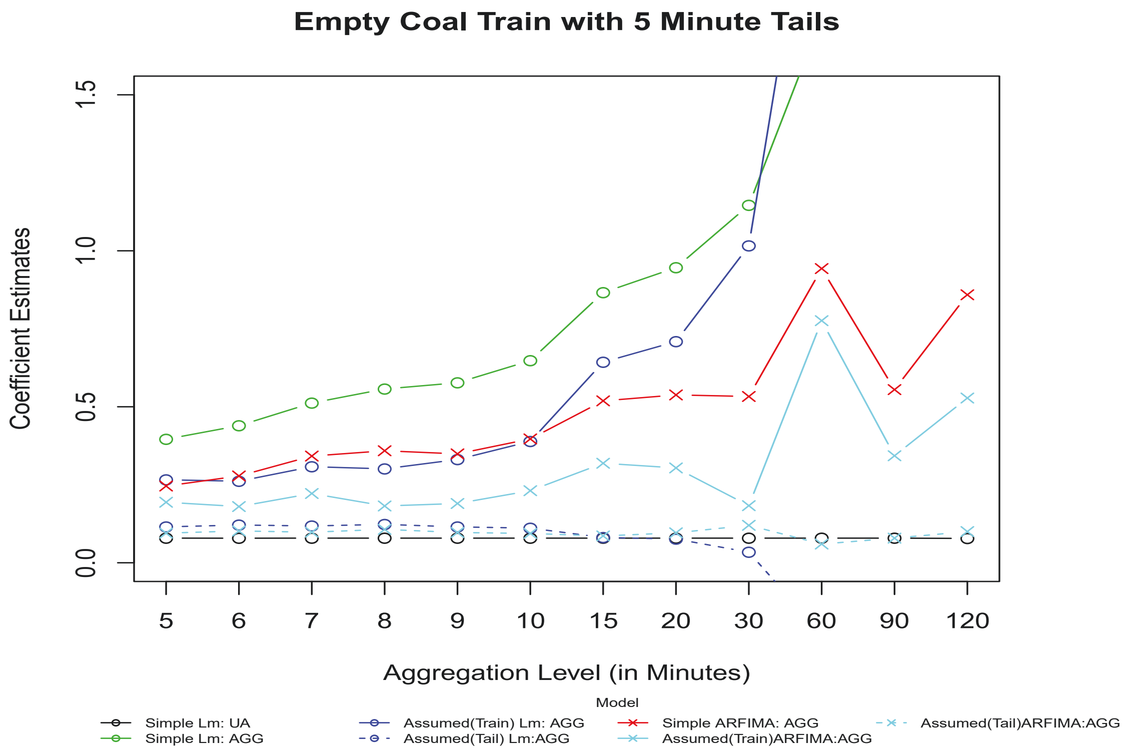
4.4.1 Comparison of the simple and assumed models

We begin our analysis of the coal train data by comparing the simple and assumed models from equations (35) and (36). Through this comparison we are able to show that there is indeed a tail effect present in the data. Figure 4.3 shows the coefficient estimates for the empty coal train and tail covariates, under the models (35) and (36). For the assumed model (36), we have assumed the tail length to be 5 minutes. We consider the aggregation periods from 5 minutes to 2 hours. For both the simple and assumed models we implement a regression with iid errors and ARFIMA errors.

The green line shows the simple model in (35) with iid error structure. We can see that as we increase the aggregation period J, the coefficient estimates systematically increase. The red line plots the same simple model, however here we are incorporating ARFIMA errors to account for the long memory error structure that is present in the data. Here we also see a systematic increase as the aggregation period increases, suggesting that this systematic change in the coefficient estimates is not a result of the autocorrelation in the data. These two models are presented in chapter 3. The dark and light blue lines, solid and dotted, show the outcomes for the assumed model in (36). In the dark blue lines we do not account for the long memory in the data and implement a linear regression with iid errors. The light blue lines are the results of the assumed model with ARFIMA errors. As we showed in chapter 3, the ARFIMA model correctly models the long memory. The solid lines are the train coefficient estimates, and the dotted lines are the assumed tail estimates.

In our simulation study, we showed that if we have omitted a significant variable in a similar setting, then the coefficient estimates are biased. This bias results in a varying coefficient estimate as we aggregate the data. This effect can be seen in both the green and red lines. They both increase systematically as we increase our aggregation period. In contrast the coefficient estimates for the assumed model, as shown by the dark and light blue solid and dotted lines, remain constant. This suggests that the simple model is omitting a tail effect, and that the assumed model with 5 minute tails may adequately capture a tail effect. Empty coal trains take, on average, between 1 and 3 minutes to pass the monitor. From figure 4.3, as we increase the aggregation over a period of 10 minutes, the estimates become more variable. For this reason, we limit the remainder of this analysis to the aggregation periods between 5 and 10 minutes

Figure 4.3: Comparison of empty coal train and tail covariates from the simple and assumed models (35) and (36) with both iid and ARFIMA error structures over the aggregation periods of 5 minutes to 2 hours.



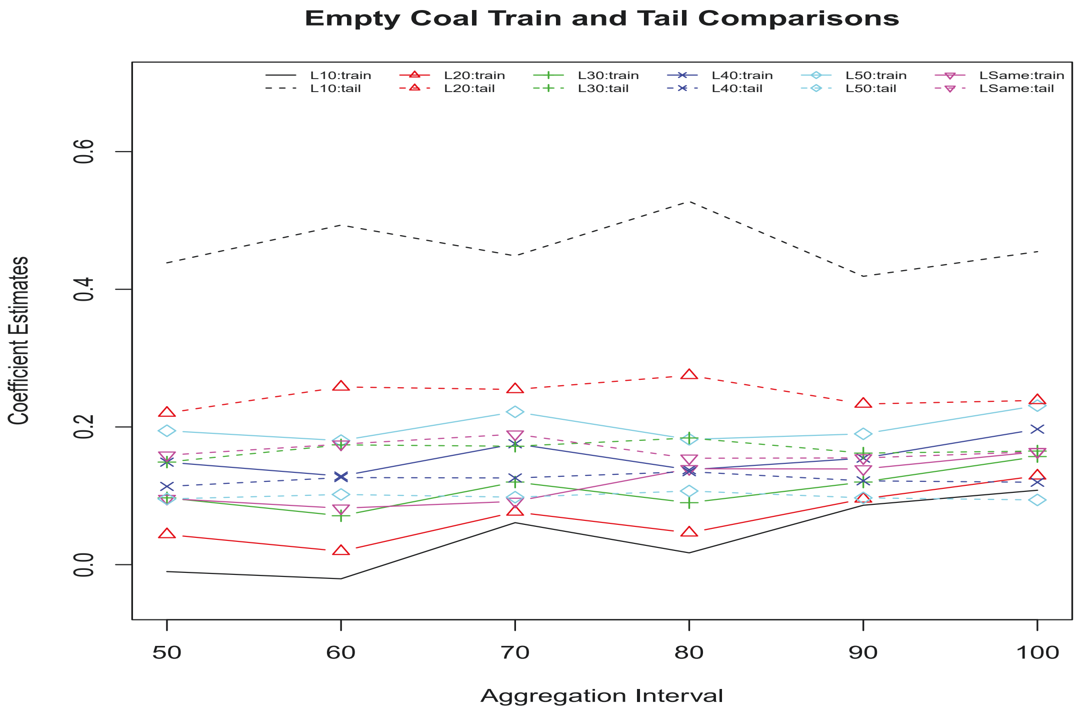
The systematic increase in coefficient estimates under the simple model, and the consistent nature of the estimates under the assumed model, confirms our theory that there is train tail effect, and that our simple model results in a misspecified model. However, the figure 4.3 does not explore the possible tail lengths for the assumed model. We thus now consider a number of tail lengths for the assumed model for the coal and freight trains.

4.4.2 Selection of tail length for the assumed model

It has been shown in the simulations and in the application thus far, that the assumed model reduces the amount of bias, in comparison to the simple model. We now attempt to determine the tail length with our assumed model in (36) with 1 minute to 5 minute tails. For each tail selection we analyse the data separately. In our simulation study, we determined that the analysis of the correct tail results in constant coefficient estimates as we change the aggregation period. Thus in the following analyses, for each of the coal and freight trains, we suggest that the tail length is correct when the train and tail estimates remain constant over aggregation.

For empty coal trains, in figure 4.4, we present the estimated coefficients for $\hat{\beta}_1$ and $\hat{\beta}_2$. These are the train and tail covariates from the assumed model (36). The solid lines correspond to the train covariate and the dotted lines are each suggested tail. When we analyse the assumed model with 1 minute tails, our results are presented by the black lines. Firstly, we can see that the train and tail estimates are vastly different. Furthermore, of key interest, is that the train estimates increase as we increase the aggregation period. For the model with 2 minute tails, shown by the red lines, this systematic increase is also occurring. However, once we suggest 3 minute tails, marked with the green lines, the model outcomes for the train and tail begin to stabilise. The assumed tails of 4 (dark blue) and 5 (light blue) minutes, result in the most constant coefficient estimates for both the train and tail covariates.

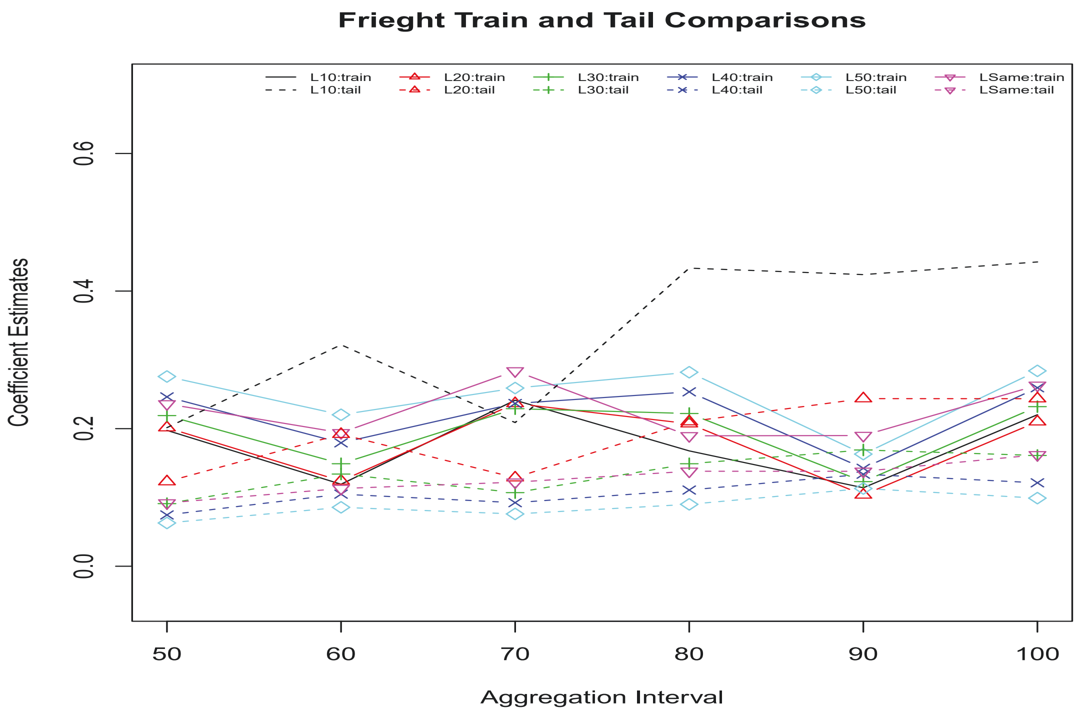
Figure 4.4: Comparison of Empty Coal train and tail covariates ($\hat{\beta}_1$ and $\hat{\beta}_2$) for 1, 2, 3, 4 and 5 minute tails. We analyse the assumed model for 5 to 10 minute aggregations.



It is difficult to differentiate between the 4 and 5 minute tail lengths. Therefore, we will consider the other train types before we make our final selection.

In figure 4.5 we present the outcomes of the assumed model (36) for the freight train and tail covariates. The most striking feature of this figure is the variation in each assumed model outcome. At first glance, the decrease then increase effect seems to be suggesting an issue. However, upon closer inspection, we can see that a large part of this variation is due to the assumed model with 1, 2 and 3 minute tail lengths. The estimates for the tail lengths of 4 and 5 minutes are much more consistent. Other than the sharp decrease at 9 minute aggregation, these tail lengths of 4 and 5 minutes, can be considered relatively constant. Although the tail estimates for 4 and 5 minutes show an increase as we increase the aggregation period, the increase is the least of the tail selections. This leads us to select either the 4 or 5 minute tail options for the assumed model.

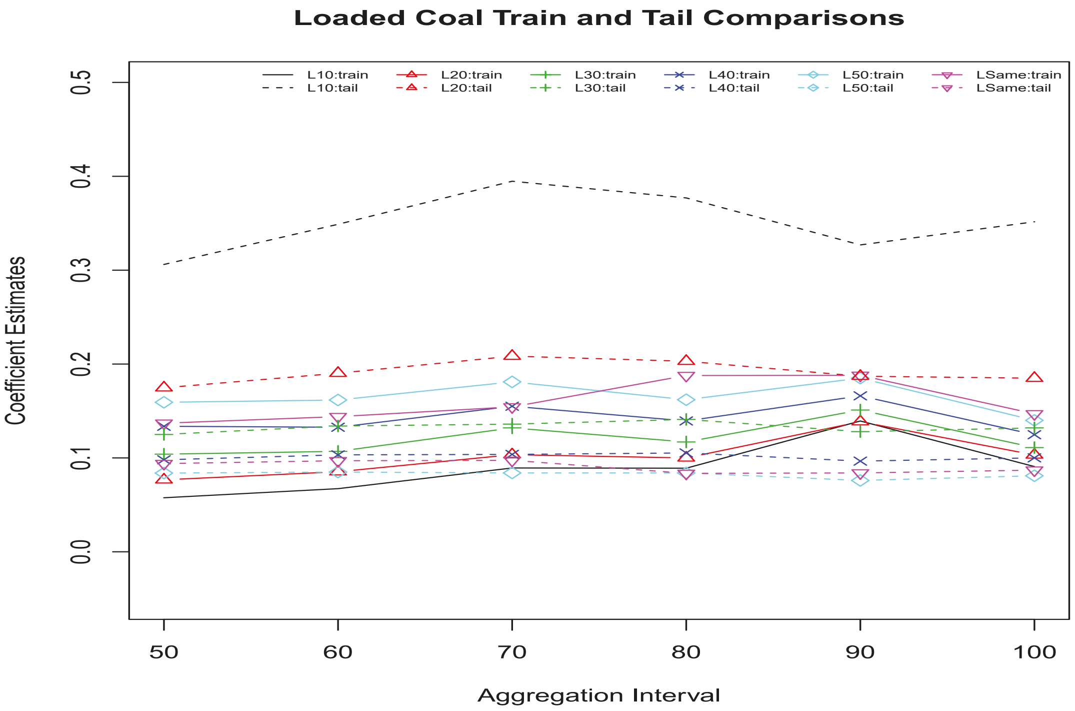
Figure 4.5: Comparison of Freight Coal Train and Tail Covariates ($\hat{\beta}_3$ and $\hat{\beta}_4$) for Alternate Tail Choices



For the loaded coal train and tails, we present figure 4.6. The results are similar to the empty coal train outcomes. We can immediately discount the assumed model with 1 minute tails, as the coefficient estimates show clear signs of variation.

Inspecting the tails, we can see that the 3, 4 and 5 minute tails are constant over the aggregation periods. These tail estimates remain at around $\hat{\beta}_2 = 0.1$. As the empty coal and freight trains had optimal results under tails of 4 and 5 minutes, we consider these same tails for the loaded coal trains.

Figure 4.6: Comparison of Loaded Coal Train and Tail Covariates ($\hat{\beta}_5$ and $\hat{\beta}_6$) for Alternate Tail Choices



Having considered a number of tail length options for the empty coal, freight and loaded coal trains, we now re-analyse the model (36) for the selected tail length of 4 minutes. This tail length has been shown to be the most constant for each of the train types. In table 4.8 we have the assumed model results for aggregation blocks of 5 to 10 minutes. The freight train outcomes are the most variable, especially as we increase the aggregation block length.This is attributed to the much lower number of passing trains in the data, in comparison to the empty and loaded coal trains.

Table 4.8: Coefficient estimates ($\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$, $\hat{\beta}_4$, $\hat{\beta}_5$ and $\hat{\beta}_6$) for the assumed model with 4 minute tails for the coal train data. We consider outcomes for the aggregation periods from 5 to 10 minutes.

Aggregation	Intercept	Empty	Empty	Freight	Freight	Loaded	Loaded
(Minutes)		Coal	Coal	Train	Tail	Coal	Coal
		Train	Tail			Train	Tail
	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$
5	3.2889	0.1489	0.1137	0.2462	0.0746	0.1337	0.0980
6	3.2734	0.1287	0.1265	0.1793	0.1048	0.1327	0.1033
7	3.2626	0.1755	0.1259	0.2366	0.0924	0.1551	0.1037
8	3.2767	0.1378	0.1350	0.2535	0.1107	0.1394	0.1053
9	3.2738	0.1547	0.1217	0.1424	0.1338	0.1661	0.0966
10	3.2559	0.1968	0.1197	0.2594	0.1213	0.1247	0.1001

In table 4.9 we expand on table 4.8 by including the standard errors for the empty coal trains, the loaded coal trains and the freight trains. As we increase the size of the aggregation, the standard errors increase. This is to be expected due to the reduction in the size of the dataset. The standard errors for the tail covariates, $\hat{\beta}_2$, $\hat{\beta}_4$ and $\hat{\beta}_6$, are much lower than their respective train covariates, indicating that a tail of 4 minutes in length is an accurate proxy variable. All of the coefficient estimates are significant, however the freight train standard errors increase significantly as we increase the aggregation period, and the coefficient estimates become quite variable for this covariate as we increase the aggregation. We attribute this to the much lower amount of passing freight trains in the dataset.

Table 4.9: Coefficient estimates ($\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$, $\hat{\beta}_4$, $\hat{\beta}_5$ and $\hat{\beta}_6$) and standard errors for the assumed model with 4 minute tails for the coal train data. We consider outcomes for the aggregation periods from 5 to 10 minutes.

Aggregation	Empty Coal				Loaded Coal				Freight			
(Minutes)	Train		Tail		Train		Tail		Train		Tail	
	$\hat{\beta}_1$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_2$	$\hat{\beta}_5$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_6$	$\hat{\beta}_3$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_4$
	Est	SE	Est	SE	Est	SE	Est	SE	Est	SE	Est	SE
5	0.149	0.046	0.114	0.017	0.134	0.174	0.098	0.006	0.246	0.013	0.075	0.007
6	0.129	0.054	0.127	0.020	0.133	0.227	0.103	0.008	0.179	0.016	0.105	0.008
7	0.176	0.063	0.126	0.023	0.155	0.267	0.104	0.009	0.237	0.019	0.092	0.010
8	0.138	0.077	0.135	0.026	0.139	0.305	0.105	0.010	0.253	0.022	0.111	0.011
9	0.155	0.079	0.122	0.027	0.166	0.349	0.097	0.011	0.142	0.025	0.134	0.013
10	0.197	0.091	0.120	0.031	0.125	0.399	0.100	0.013	0.259	0.028	0.121	0.014

In our simulation study, we extended the assumed model to what we call the combined model. Here we combine the train and assumed tail into one variable. We now consider this model on the application.

4.4.3 Fitting of the combined model

We are interested to see how our analyses reacts if we model the train and tail variables as one variable. Under this hypothesis we suggest that each passing train type has a tail of 4 minutes, as in our assumed model previously. We then create a new variable that adds the 4 minute tail onto the end of each passing train type and we consider the model in equation (37) below.
This process is the same as in our simulation section. We apply the same aggregation transformation as throughout this chapter and thesis. In interest of simplicity, we set the aggregation of the new variable for empty coal, to be "$EmptyCoalAndTail_m$". We conduct this process for every train type. The resulting model is below,

$$\begin{aligned} s_m = \beta_0 &+ \beta_1 EmptyCoalAndTail_m + \beta_2 FreightAndTail_m + \beta_3 LoadedCoalAndTail_m \\ &+ \beta_4 PassengerAndTail_m + \beta_5 UnknownAndTail_m + \epsilon_m \end{aligned} \tag{37}$$

Again we implement regression with ARFIMA(p,d,q) errors to account for the long memory structure of the data.

In Table 4.10 we present the outcomes of this modelling strategy for the aggregations from 5 to 10 minutes. The first first thing we notice is that the estimates for each train type are much more consistent than our previous analysis. If we compare our results here with that in table 4.8, where we set the trains and tails as seperate variables, we observe that the current analysis could almost be interpreted as the average of the train and tail coefficient estimates in table 4.8. This effect was also seen in our simulation study. However, as our simulations showed, this averaging effect can result in an increased bias in comparison to the assumed model.

Table 4.10: Comparison of train coefficient estimates under the combined model with 4 minute tails.

Aggregation (Minutes)	Intercept	Empty Coal	Freight	Loaded Coal	Passenger	Unknown
5	3.2909	0.1210	0.1076	0.1072	0.0422	0.0898
6	3.2324	0.1275	0.1202	0.1118	0.0435	0.0885
7	3.2566	0.1360	0.1231	0.1184	0.0561	0.1147
8	3.2770	0.1357	0.1405	0.1152	0.0544	0.1050
9	3.2741	0.1288	0.1349	0.1165	0.0602	0.0988
10	3.3006	0.1369	0.1499	0.1068	0.0522	0.1469

In table 4.11, we present the standard errors for the coefficient estimates in table 4.10. We can see that the intercept and the empty coal, freight and loaded coal trains, and passenger trains, and unknown are all statistically significant, due to their markedly lower standard errors.

Table 4.11: Comparison of standard errors for the train coefficient estimates, from table 4.10 above, under the combined model with 4 minute tails.

Aggregation (Minutes)	Intercept	Empty Coal	Freight	Loaded Coal	Passenger	Unknown
5	0.3005	0.0053	0.0129	0.0055	0.0051	0.0231
6	0.3959	0.0061	0.0146	0.0061	0.0058	0.0247
7	0.6259	0.0067	0.0157	0.0066	0.0063	0.0281
8	0.3438	0.0072	0.0165	0.0072	0.0070	0.0304
9	0.4159	0.0080	0.0180	0.0078	0.0075	0.0316
10	0.5407	0.0087	0.0187	0.0084	0.0082	0.0350

Of particular interest is the empty coal estimates under the model (37) in the third column of table 4.10. Here the coefficient estimates increase from 0.12 to 0.136, while under the assumed model this increase is from 0.14 to 0.196, as we increase the aggregation from 5 to 10 minutes. Continuing with our simulations and previous application under the simple and assumed models, this suggests that we are not correctly specifying the model to the data.

We continue our application with a review of the models and their results.

4.4.4 Review of application

Our analysis of the coal train data gives further evidence of our hypothesis that there is a train tail effect in the data. A comparison of the simple and assumed models, under different levels of aggregation, suggest that the simple model is suffering from a bias due to the omitted variable of the tail. In practice, of course, the true tail effect is unknown. However, we consider a number of tail options and make the case for selecting a tail length. Through our simulations and application, we are able to show that if we have a correctly specified tail length, then its estimates will not vary as we change the aggregation period.

We conclude that a train tail length of 4 minutes is the most reasonable outcome for this assumed tail variable. As shown in Table 4.8, the freight train's have the largest effect on air quality measurements. There is no significant difference between empty and loaded coal trains, however the tail for a passing empty coal train is the largest of any tail effect. As these trains are as long as the loaded coal trains, but are empty, they travel much faster. This is consistent with our hypothesis that the dust that is on the tracks is stirred up by a passing train, and the faster a train passes the monitor the more it will be stirred up.

4.5 Discussion

In chapter 3, we found that our estimated coefficients for each train type increased as we increased the aggregation period. This systematic increase led us to believe that we were working with misspecified models. In our previous work, as in chapter 2, we hypothesized that there was a tail effect for each passing train, and we analysed the data under the assumption of a 5 minute tail. In our applied analysis in this chapter, we have found that this systematic increase in coefficient estimates continues. Thus by applying an assumed tail to our data, we have been able to improve our analysis. To the best of our knowledge this is a novel finding. To understand the influence of a tail effect, we conduct a more thorough review of the possible omitted tail variable.

We explore the effect of a simple model, with no tails, and an assumed model, where we suggest a number of possible tail lengths. Our goal is to determine if we have incorrectly specified the model with no tails, and if so, how to select a tail length for the assumed model.
Beginning with some theoretical calculations on a linear regression, we consider the induced bias from the simple model if there is a tail effect. We contrast this bias for a simple model, with that of an assumed tail model. We show that if there is a tail effect in a true model, then the assumed model will have a lower coefficient bias than if we ignore the tail.

We follow this theory with a simulation study. Here we explore the impact of a misspecified model. We generate a simulated dataset that, although simplistic, mimics the coal train data. This allows us to confirm the expected bias from our theory. For our assumed model, in the case that we assume a tail length that is similar to the true tail length, we can reduce the bias in comparison to the simple model.
Our simulations show that if we omit a tail variable, then the coefficient estimates increase as we increase the aggregation period. Therefore, we consider a number of tail lengths for our assumed model. In the event that we correctly specify the tail, we are able to achieve consistent coefficient estimates for each aggregation period. This improves our confidence in selecting the correct tail for the application.

Following the simulations, we reanalyse the coal train data. By comparing the simple and assumed models for a range of aggregation periods from 5 minutes to 2 hours, we are able to clearly find evidence of a misspecified model. As we increased the aggregation period, the simple model resulted in coefficient estimates that increased systematically. This suggested that there was a tail effect. By suggesting a tail length, we were able to determine that the assumed model is capturing the train and tail effect. Furthermore, the increase in variation for the coefficient estimates as we increased the aggregation period above ten minutes, coupled with the fact that the freight and coal trains took, on average, 1 to 3 minutes to pass the monitor, suggested that an aggregation period of up to 10 minutes was reasonable.

Having determined that there is a tail effect present in the coal train data, we set out out estimate the possible length of the tail. Analysing the assumed model with tail lengths of 1, 2, 3, 4 and 5 minutes, we compare the results for the empty coal, freight and loaded coal trains. We conclude that a tail of 4 minutes results in the most consistent results over the three train types. We are thus able to make inferences on the coal train data. We determine that the freight trains cause the largest increase in air quality measurements. However, we also find that their tail has the lowest tail effect. We attribute this to the fact that the freight trains are the fastest of the three trains, and thus pass the monitor the quickest. This leads to the coal dust that is on the tracks being stirred up with the most vigour, resulting in the increased passing train effect, and then the dust settles the quickest of the three tails, as the train is not present for as long as the coal trains.
The empty coal and loaded coal trains have a similar, lower, effect on air quality. We note that the empty coal train is the same train as the loaded coal train, except that it is returning to the mine after having unloaded its cargo in the port. As such the only difference between the two is the speed. The empty coal trains travel faster than the loaded coal trains, and we attribute this speed to the slightly larger impact on air quality.

We conclude our application, with an extension of the assumed model, which we also discussed in our simulations. This new model, called the combined model, combines the train and assumed tail variables into one variable. We then reanalyse the coal train data for the aggregation periods of 5 to 10 minutes. Our results here are consistent with our simulations, whereby the combined model can reduce the accuracy in comparison to the assumed model. We find that the results of the combined model, appear to be an average of the train and tail coefficient estimates from the assumed model.

This chapter improves our knowledge of the impact of passing trains on air quality in the Newcastle region of Australia. We are able to show that there is a tail effect for each passing train, and that including this tail in our modelling results in a more thorough analysis.

Further research into this application could consider the passing train tail as a distribution rather than an indicator as we have specified in this case. Improvements in the capabilities of the linear regression with ARFIMA errors, that allow for the analysis of a larger dataset are also possibilities. Another possibility is to approach the unaggregated data from another perspective such as divide and recombine, whereby we can implement the regression with ARFIMA errors on the complete dataset.

5 Bivariate Time Series Modelling

5.1 Introduction

Having explored the effect of temporal aggregation to means and then the issues with our data in the form of model misspecification in the previous two chapters, where we focused solely on the TSP data, here we introduce temporal aggregation of a bivariate series. The applied dataset also includes other particulate matter measurements in the form of PM1, PM2.5 and PM10. The differences in these three measures are the size of the particulate matter diameter.

Due to the similarities in these observed data, we consider a bivariate analysis to exploit the structure between these series. We are interested in questions such as: how similar are these particulate measures and do trains have the same effect on different air measures? Further, we believe that a more thorough model can extract some new information from the data.

We approach this bivariate time series analysis by utilising a random effects model. This permits the inclusion of covariate effects as well as a myriad of error structures to be applied. We are particularly interested in the modelling of error structures through the random effects model as we can explore the relationships between the two series and across each series.

Lee and Nelder[36][37], introduce the hierarchical log-likelihood. We extend their use of the h-likelihood to random effects models with the implementation of AutoRegressive AR(1) structure on the errors, and also on the random effects in a number of models. The use of h-likelihood permits the inference of both fixed and random unknowns, and it is calculated by treating unobserved random effects as parameters to be estimated. Reversible jump MCMC as discussed by Graves et al [18] could also be considered for treating random effects as unknown parameters in the model, however we have not considered this approach due to it demanding computational nature.

We present a number of models utilising the h-likelihood. Our first model is a random effects model with an i.i.d error structure. This is the basic model and is simple to analyse and compute in a number of software programmes such as with r:nlme:lme[49] and also in SAS:Proc:Mixed. Due to the time series nature of the datasets, we extend our second model to include an AR(1) error structure on our error terms. Again this has already been covered in theory and is programmed into certain statistics languages such as SAS:PROC:MIXED. However we show how we analyse and code this using a h-likelihood.
In our third model, we assume an AR(1) structure on both the error terms and the random effects. This model was inspired by the complexity of our data, where we can see there is a lot of structure to exploit. We can interpret such a model as having a dependency not only across time, across series, but also across time and series concurrently.
Our fourth and fifth models are extensions of models 2 and 3 where we include a constant for each series that we multiply over the random effects. The thought behind this is to cover more of the variation between each series.

Through a number of simulation studies as well as the coal train analysis, we present our findings. Initial results are positive but are constrained by a number of limitations. A few of these limitations are consistent with earlier chapters such as increasing coefficient estimates as the aggregation block length increases, which we attribute to model misspecification.

Other limitations are new such as the memory constraints imposed by the storage and inversion of the dense covariance matrix in our models, the selection options for our initial values as well as the effects of long memory dependence on our models.

Nonetheless we show that the models all fit accurately in the simulations, and the effect of aggregation is positive by reducing the data size and model complexity while producing correct estimates. The applied analysis of the coal train data is consistent with models in previous chapters while introducing further information which is helpful and of interest.

Sections 2 and 3 explore the temporal aggregation process to bivariate time series, the models implemented in this chapter, and an algorithm for their estimation. Two simulation studies are presented in sections 4 and 5 that explore the models fit and the effect of aggregation of the models. We follow this with our analysis of the PM1 and PM2.5 series using the bivariate time series with random effects models, and conclude this chapter with a discussion of our results.

5.2 Aggregation and mixed effect models for bivariate time series

The series PM1 and PM2.5, as shown in chapter 2, have a complex structure and large data size. As we have done throughout this thesis, we apply temporal aggregation to each series. Here we outline this process of temporal aggregation of the dependent variable y and also the covariates x. We then organise the resulting aggregated bivariate series so that we can analyse them using the 5 models that we present in this section.

5.2.1 Temporal aggregation for a bivariate time series

Assume a data set with observed outcome variable, y_n, where $n = 1, \dots, N$, and P predictors for x_{np}, where $p = 1, \dots, P$. We select a suitable block length J, over which we divide the data set into a series of blocks. This choice of J is data dependent. The division of the data of length N by J, creates M blocks, $M = N/J$. M must be an integer. Thus, in the case N/J is not an integer, then we must remove the final $n_N - n_{(M\times J)}$ observations such that M is an integer. The resulting data set is reduced from $n = 1, \dots, N$, to $n = 1, \dots, n_{(M\times J)}$.

Having selected our block length J, we can proceed to the aggregation of our outcome variable y_n into the new series s_m. In this chapter, we again focus on temporal aggregation to the mean. Thus, the temporally aggregated series,

$$s_{m1} = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} y_n. \tag{38}$$

Our focus is on the relationships between two series, PM1 and PM2.5. After we have aggregated each series, noted as $k = 1, 2$, the resulting bivariate series are,

$$S_1 = (s_{11}, s_{21}, s_{31}, \dots, s_{M1})^T$$

and

$$S_2 = (s_{12}, s_{22}, s_{32}, \dots, s_{M2})^T .$$

We must also apply the aggregation to the covariates of interest. Given our matrix X of covariates P, we transform through aggregation into the matrix Z with Q aggregated covariates, where $q = 1, \ldots, Q$ and Q is a combination of the P original covariates. There is no restriction on the number of Q covariates as there may be less, equal or more than the original P, depending on the aggregations utilised. In this chapter we focus on proportions, but in the next chapter we explore other possibilities such as indicators or sums.

In the case that we take proportions as the aggregation method for a particular covariate,

$$z_{mq} = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} x_{np}, \tag{39}$$

our original covariates x_{np} transform to z_{mq}. In our following linear mixed effects models, we include a column of ones in our aggregated Z covariate matrix. This allows the intercept to be included in our model, thereby simplifying our notation.

$$Z = \begin{pmatrix} 1 & z_{11} & z_{12} & z_{13} & \cdots & z_{1Q} \\ 1 & z_{21} & z_{22} & z_{23} & \cdots & z_{2Q} \\ 1 & z_{31} & z_{32} & z_{33} & \cdots & z_{3Q} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & z_{M1} & z_{M2} & z_{M3} & \cdots & z_{MQ} \end{pmatrix}.$$

Now that we have transformed our data through aggregation, we present our models below.

5.2.2 Mixed effect models for bivariate time series

Given a bivariate series s_k, where there is a dependence structure in the error terms, we consider a number of models. We adopt a linear mixed models framework, which has been widely used in the longitudinal data setting. The key idea is to introduce a shared random effect that induces cross-correlation between the two time series that we are considering. The model then also incorporates series-specific error terms that may have serial correlation as well. Specifically, we consider the following linear mixed effects model:

$$S_k = Z\beta_k + \boldsymbol{\delta} + \epsilon_k, \ k = 1, 2 \tag{40}$$

where S_k is the $(M \times 1)$ vector of responses for the kth series, defined above, Z is the $(M \times Q)$ matrix of covariates of interest and β_k is the corresponding (Q x 1) vector of regression coefficients. $\boldsymbol{\delta}$ is a $(M \times 1)$ vector of shared random effects and ϵ_k is a $(M \times 1)$ vector of errors, specific to the kth symbol. δ and ϵ_k are independent and we will discuss various possible distributional assumptions on these error terms presently. Note that while the models for the two series have the same set of covariates, the corresponding coefficients are allowed to differ. It would be straightforward to allow different sets of covariates for each series, but we don't discuss this option here in order to simply our notation.

5.2.3 Model 1

We are interested in a number of possible models of varying levels of complexity. In some scenarios, aggregation may render the residuals independent over time and therefore allow us to proceed with a simple correlation structure, namely i.id. gaussian. We to this throughout the paper as **Model 1**. More precisely, Model 1 corresponds to the specific case of equation (40) where

$$\delta \sim N(0, \tau^2 I) \quad \text{and} \quad \epsilon_k \sim N(0, \sigma_k^2 I), \tag{41}$$

where I is the $(M \times M)$ identity matrix.

5.2.4 Model 2

In the event that either S_1 or S_2 or both have some degree of serial correlation present, we must accomodate this. We do so in Model 2. We extend Model 1 by incorporating a correlation structure on the error terms. We select an Autoregressive AR(1) structure on the residuals as this is most defining of our motivating coal data set. Model 2 is defined as,

$$\begin{aligned} &S_k = Z\beta_k + \delta + \epsilon_k, \qquad k = 1, 2 \\ &\epsilon_{m1} = \rho_1 \epsilon_{(m-1)1} + \eta_{m1} \quad \text{and} \quad \epsilon_{m2} = \rho_2 \epsilon_{(m-1)2} + \eta_{m2} \end{aligned} \tag{42}$$

where, ρ_k are the autoregressive AR(1) correlation coefficients for each error series k, with $|\rho_k| < 1$, for stationarity and

$$\delta \sim N(0, \tau^2 I) \quad \text{and} \quad \eta_k \sim N(0, \sigma_k^2 I), \tag{43}$$

where I is the $(M \times M)$ identity matrix, η_k is a $(M \times 1)$ vector of errors, specific to the k^{th} symbol, and δ and η_k are independent.

Using Model 2 with autoregressive error structure AR(1) as shown in equation (42), we calculate a series of variances and covariances for the model.

We begin with the variance of the series k. The random effects δ and residuals ϵ are independent, thus,

$$\begin{aligned} Var[s_{mk}] &= E[(s_{mk} - E[s_{mk}])(s_{mk} - E[s_{mk}])] \\ &= E[(\delta_m + \epsilon_{mk})(\delta_m + \epsilon_{mk})] \\ &= E[(\delta_m^2 + 2\delta_m \epsilon_{mk} + \epsilon_{mk}^2] \\ &= E[\delta_m^2 + \epsilon_{mk}^2] \end{aligned}$$

where the expectation of δ^2, $E[\delta^2]$ is the variance of δ which is given in equation (43),

$$\begin{aligned} E[\delta_m^2] &= E[\delta_m.\delta_m] \\ &= \tau^2 \end{aligned}$$

and the expectation of the residuals, $E[\epsilon_{mk}^2]$, containing an AR(1) structure is

$$\begin{aligned} E[\epsilon_{mk}^2] &= Var[\epsilon_{mk}] \\ &= Var[\rho_k \epsilon_{(m-1)k} + \eta_{mk}] \\ &= \rho_k^2 Var[\epsilon_{(m-1)k}] + Cov[\rho_k \epsilon_{(m-1)k}, \eta_{mk}] + Var[\eta_{mk}] \\ &= \rho_k^2 Var[\epsilon_{mk}] + 0 + \sigma_k^2 \\ Var[\epsilon_{mk}](1 - \rho_k^2) &= \sigma_k^2 \\ Var[\epsilon_{mk}] &= \frac{\sigma_k^2}{(1 - \rho_k^2)} \end{aligned}$$

thus, combining the above two expectations we determine the variance of a series k to be,

$$\begin{aligned} Var[s_{mk}] &= E[\delta_m^2 + \epsilon_{mk}^2] \\ &= \tau^2 + \frac{\sigma_k^2}{(1 - \rho_k^2)}. \end{aligned}$$

For the covariance between $Cov[S_{mk}, S_{(m-1)k}]$, the covariance between two time points of the same series,

$$\begin{aligned} Cov[s_{mk}, s_{(m-1)k}] &= E[(s_{mk} - E[s_{mk}])(s_{(m-1)k} - E[s_{(m-1)k}])] \\ &= E[(\delta_m + \epsilon_{mk})(\delta_{(m-1)} + \epsilon_{(m-1)k})] \\ &= E[(\delta_m \delta_{(m-1)} + \delta_m \epsilon_{(m-1)k} + \delta_{(m-1)} \epsilon_{mk} + \epsilon_{mk} \epsilon_{(m-1)k}] \\ &= E[\delta_m \delta_{(m-1)} + \epsilon_{mk} \epsilon_{(m-1)k}] \end{aligned}$$

where, the expectation of δ is i.i.d, and as such,

$$E[\delta_m \delta_{(m-1)}] = 0$$

and, extending the calculation of $E[\epsilon_{mk}^2]$ above, the expectation of ϵ_{mk} with a lag is,

$$\begin{aligned}
E[\epsilon_{mk}\epsilon_{(m-1)k}] &= E[(\rho_k\epsilon_{(m-1)k} + \eta_{mk})\epsilon_{(m-1)k}] \\
&= E[\rho_k\epsilon^2_{(m-1)k}] \\
&= \rho_k E[\epsilon^2_{(m-1)k}] \\
&= \rho_k E[\epsilon^2_{mk}] \\
&= \rho_k Var[\epsilon_{mk}] \\
&= \rho_k \frac{\sigma^2_k}{(1-\rho^2_k)} \\
&= \frac{\rho_k\sigma^2_k}{(1-\rho^2_k)}
\end{aligned}$$

thus, the covariance at lag 1 of the series k is given by,

$$\begin{aligned}
Cov[s_{mk}, s_{(m-1)k}] &= E[\delta_m\delta_{(m-1)} + \epsilon_{mk}\epsilon_{(m-1)k}] \\
&= 0 + \frac{\rho_k\sigma^2_k}{(1-\rho^2_k)} \\
&= \frac{\rho_k\sigma^2_k}{(1-\rho^2_k)}
\end{aligned}$$

We now focus on the covariance across series $k = 1$ and $k = 2$ at the same time point m,

$$\begin{aligned}
Cov[s_{m1}, s_{m2}] &= E[(s_{m1} - E[s_{m1}])(s_{m2} - E[s_{m2}])] \\
&= E[(\delta_m + \epsilon_{m1})(\delta_m + \epsilon_{m2})] \\
&= E[\delta^2_m + \delta_m\epsilon_{m2} + \delta_m\epsilon_{m1} + \epsilon_{m1}\epsilon_{m2}] \\
&= E[\delta^2_m + \epsilon_{m1}\epsilon_{m2}] \\
&= \tau^2
\end{aligned}$$

where, we know from previous equations above,

$$E[\delta^2_m] = \tau^2 \quad \text{and} \quad E[\epsilon_{m1}\epsilon_{m2}] = 0$$

as the two series are assumed to be independent under model 2.

We also consider the cross-covariance for the two time series $k = 1$ and $k = 2$ and for two time points $m = 1$ and $m = 2$,

$$\begin{aligned}
Cov[s_{m1}, s_{(m-1)2}] &= E[(s_{m1} - E[s_{m1}])(s_{(m-1)2} - E[s_{(m-1)2}])] \\
&= E[(\delta_m + \epsilon_{m1})(\delta_{(m-1)} + \epsilon_{(m-1)2})] \\
&= E[\delta_m\delta_{(m-1)} + \delta_m\epsilon_{(m-1)2} + \delta_{(m-1)}\epsilon_{m1} + \epsilon_{m1}\epsilon_{(m-1)2}] \\
&= E[\delta_m\delta_{(m-1)} + \epsilon_{m1}\epsilon_{(m-1)2}] \\
&= 0
\end{aligned}$$

as the δ are independent over time in this model and the ϵ are independent across time series.

We present the covariance matrix below (which is an $2M \times 2M$ matrix), for the first $m = 1, \ldots, 3$ observations and $k = 1, 2$. The implementation of seperate correlation coefficients ρ_1 and ρ_2 creates a situation where the correlation in the model over time for each series is accounted for, as well as allowing for the cross-correlation to be included by the random effects variance τ^2.

$$Cov\begin{pmatrix} s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_{31} \\ s_{32} \end{pmatrix} = \begin{pmatrix} \tau^2 + \frac{\sigma_1^2}{(1-\rho_1^2)} & \tau^2 & \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 & \frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & 0 \\ \tau^2 & \tau^2 + \frac{\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} \\ \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 & \tau^2 + \frac{\sigma_1^2}{(1-\rho_1^2)} & \tau^2 & \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 \\ 0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \tau^2 & \tau^2 + \frac{\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} \\ \frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & 0 & \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 & \tau^2 + \frac{\sigma_1^2}{(1-\rho_1^2)} & \tau^2 \\ 0 & \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \tau^2 & \tau^2 + \frac{\sigma_2^2}{(1-\rho_2^2)} \end{pmatrix}$$

It is also possible to calculate the covariance terms for each series k at each observation m. The covariance over time of the same series k is,
for $t = 0$,

$$Cov[s_{mk}, s_{(m-t)k}] = \tau^2 + \frac{\sigma_k^2}{(1 - \rho_k^2)}, \tag{44}$$

and for $t \neq 0$,

$$Cov[s_{mk}, s_{(m-t)k}] = \frac{\rho_k^t \sigma_k^2}{(1 - \rho_k^2)}. \tag{45}$$

We also consider the covariance across the two series $k = 1$ and $k = 2$ at the same time point, $t = 0$,

$$Cov[s_{m1}, s_{(m-t)2}] = \tau^2. \tag{46}$$

And the covariance across the two series $k = 1$ and $k = 2$ over time at any time point other than the same ($t \neq 0$), as,

$$Cov[s_{m1}, s_{(m-t)2}] = 0. \tag{47}$$

5.2.5 Model 3

Model 2 and it's covariance matrix above account for much of the possible correlation in and between the two series k. However, we believe that there is also some cross-correlation present between the two series **and** across time. We consider this proposal in model 3. In Model 3 we utilise the correlation structure in Model 2 and extend it by including a autoregressive AR(1) structure on the random effects τ. We thus have the case of expression (40) with

$$\begin{aligned} S_k &= Z\beta_k + \delta + \epsilon_k, \qquad k = 1, 2 \\ \epsilon_{m1} &= \rho_1 \epsilon_{(m-1)1} + \eta_{m1} \\ \epsilon_{m2} &= \rho_2 \epsilon_{(m-1)2} + \eta_{m2} \\ \delta_m &= \omega \delta_{m-1} + \gamma_m \end{aligned} \tag{48}$$

with $\gamma \sim N(0, \tau^2 I)$ and δ is a $(M \times 1)$ vector of random effects. The AR(1) correlation coefficient for the random effects is given by ω, with $|\rho_k, \omega| < 1$ for stationarity.

The introduction of an autoregressive AR(1) correlation structure on the random effects as shown in equation (48) results in a more complex covariance matrix structure to that presented in Model 2. Below we present the algebra necessary to construct the covariance matrix for Model 3.

Again we begin with the variance of the series k,

$$\begin{aligned}
Var[s_{mk}] &= Cov[s_{mk}, s_{mk}] \\
&= E[(s_{mk} - E[s_{mk}])(s_{mk} - E[s_{mk}])] \\
&= E[(\delta_m + \epsilon_{mk})(\delta_m + \epsilon_{mk})] \\
&= E[(\delta_m^2 + 2\delta_m \epsilon_{mk} + \epsilon_{mk}^2] \\
&= E[\delta_m^2 + \epsilon_{mk}^2]
\end{aligned}$$

where, the variance of δ is given by the expectation of $E[\delta_m^2]$, and the expectation of γ^2 is τ^2 as expressed in equation 48,

$$\begin{aligned}
Var[\delta_m] &= E[(\omega\delta_{(m-1)} + \gamma_m)^2] \\
&= E[\omega^2 \delta_{(m-1)}^2] + Cov[\omega\delta_{(m-1)}, \gamma_m] + E[\gamma_m^2] \\
&= \omega^2 E[\delta_{(m-1)}^2] + 0 + \tau^2 \\
&= \omega^2 E[\delta_m^2] + \tau^2 \\
&= \omega^2 Var[\delta_m] + \tau^2 \\
Var[\delta_m](1 - \omega^2) &= \tau^2 \\
Var[\delta_m] &= \frac{\tau^2}{(1 - \omega^2)}
\end{aligned}$$

thus, the variance for S_{mk} under Model 3 is,

$$\begin{aligned}
Var[s_{mk}] &= E[\delta_m^2 + \epsilon_{mk}^2] \\
&= \frac{\tau^2}{(1 - \omega^2)} + \frac{\sigma_k^2}{(1 - \rho_k^2)}.
\end{aligned}$$

For the covariance across series $k = 1$ and $k = 2$ at the same time point m,

$$\begin{aligned}
Cov[s_{mk}, s_{(m-1)k}] &= E[(s_{mk} - E[s_{mk}])(s_{(m-1)k} - E[s_{(m-1)k}])] \\
&= E[(\delta_m + \epsilon_{mk})(\delta_{(m-1)} + \epsilon_{(m-1)k})] \\
&= E[\delta_m \delta_{(m-1)} + \delta_m \epsilon_{(m-1)k} + \delta_{(m-1)} \epsilon_{mk} + \epsilon_{mk} \epsilon_{(m-1)k}] \\
&= E[\delta_m \delta_{(m-1)} + \epsilon_{mk} \epsilon_{(m-1)k}]
\end{aligned}$$

where,

$$\begin{aligned}
E[\delta_m\delta_{(m-1)}] &= E[(\omega\delta_{(m-1)} + \gamma_m)\delta_{(m-1)}] \\
&= E[\omega\delta_{(m-1)}\delta_{(m-1)}] \\
&= \omega E[\delta^2_{(m-1)}] \\
&= \omega E[\delta^2_m] \\
&= \omega Var[\delta_m] \\
&= \frac{\omega\tau^2}{(1-\omega^2)}
\end{aligned}$$

Thus,

$$\begin{aligned}
Cov[s_{mk}, s_{(m-1)k}] &= E[\delta_m\delta_{(m-1)} + \epsilon_{mk}\epsilon_{(m-1)k}] \\
&= \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_k\sigma_k^2}{(1-\rho_k^2)}
\end{aligned}$$

We also consider the instance of cross covariance between the two series $k = 1$ and $k = 2$,

$$\begin{aligned}
Cov[s_{m1}, s_{m2}] &= E[\delta^2_m + \epsilon_{m1}\epsilon_{m2}] \\
&= \frac{\tau^2}{(1-\omega^2)}
\end{aligned}$$

and we conclude with the covariance across both time and series,

$$\begin{aligned}
Cov[s_{m1}, s_{(m-1)2}] &= E[\delta_m\delta_{(m-1)} + \epsilon_{m1}\epsilon_{(m-1)2}] \\
&= E[\delta_m\delta_{(m-1)}] \\
&= \frac{\omega\tau^2}{(1-\omega^2)}
\end{aligned}$$

We present the covariance matrix for Model 3 in the same vein as above in Model 2. The introduction of an Autoregressive AR(1) term for the random effects δ allows us to estimate the correlation at all time points and cross-correlations. This is shown through, τ^2, the variance of the random effects δ.

$$Cov\begin{pmatrix} s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_{31} \\ s_{32} \end{pmatrix} = \tag{49}$$

$$\begin{pmatrix}
\frac{\tau^2}{(1-\omega^2)} + \frac{\sigma_1^2}{(1-\rho_1^2)} & \frac{\tau^2}{(1-\omega^2)} & \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau^2}{(1-\omega^2)} & \frac{\omega^2\tau^2}{(1-\omega^2)} + \frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega^2\tau^2}{(1-\omega^2)} \\
\frac{\tau^2}{(1-\omega^2)} & \frac{\tau^2}{(1-\omega^2)} + \frac{\sigma_2^2}{(1-\rho_2^2)} & \frac{\omega\tau^2}{(1-\omega^2)} & \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \frac{\omega^2\tau^2}{(1-\omega^2)} & \frac{\omega^2\tau^2}{(1-\omega^2)} + \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} \\
\frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau^2}{(1-\omega^2)} & \frac{\tau^2}{(1-\omega^2)} + \frac{\sigma_1^2}{(1-\rho_1^2)} & \frac{\tau^2}{(1-\omega^2)} & \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau^2}{(1-\omega^2)} \\
\frac{\omega\tau^2}{(1-\omega^2)} & \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \frac{\tau^2}{(1-\omega^2)} & \frac{\tau^2}{(1-\omega^2)} + \frac{\sigma_2^2}{(1-\rho_2^2)} & \frac{\omega\tau^2}{(1-\omega^2)} & \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} \\
\frac{\omega^2\tau^2}{(1-\omega^2)} + \frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega^2\tau^2}{(1-\omega^2)} & \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau^2}{(1-\omega^2)} & \frac{\tau^2}{(1-\omega^2)} + \frac{\sigma_1^2}{(1-\rho_1^2)} & \frac{\tau^2}{(1-\omega^2)} \\
\frac{\omega^2\tau^2}{(1-\omega^2)} & \frac{\omega^2\tau^2}{(1-\omega^2)} + \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} & \frac{\omega\tau^2}{(1-\omega^2)} & \frac{\omega\tau^2}{(1-\omega^2)} + \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \frac{\tau^2}{(1-\omega^2)} & \frac{\tau^2}{(1-\omega^2)} + \frac{\sigma_2^2}{(1-\rho_2^2)}
\end{pmatrix}$$

It is also possible to calculate the covariance terms for each series k at each observation m. The covariance over time of the same series k is,

$$Cov[s_{mk}, s_{(m-t)k}] = \frac{\omega^t\tau^2}{(1-\omega^2)} + \frac{\rho_k^t\sigma_k^2}{(1-\rho_k^2)}. \tag{50}$$

We also consider the covariance across the two series $k = 1$ and $k = 2$ over time,

$$Cov[s_{m1}, s_{(m-t)2}] = \frac{\omega^t\tau^2}{(1-\omega^2)}. \tag{51}$$

5.2.6 Model 4

Model 4 is an extension of Model 2, whereby each series s_k has it's own constant τ_k that we multiply over the random effects. The thought behind this is that each series is correlated with the other but there can be a few ways that they are modelled whereby they have different effects. An example of this is the situation where $\tau_1 = 2$ and $\tau_2 = 1$; here the two series are highly correlated but S_1 has more variation. Thus, we consider Model 4 as,

$$\begin{aligned}
&S_k = Z\beta_k + \tau_k\delta + \epsilon_k, \qquad k = 1,2 \\
&\epsilon_{m1} = \rho_1\epsilon_{(m-1)1} + \eta_{m1} \quad \text{ and } \quad \epsilon_{m2} = \rho_2\epsilon_{(m-1)2} + \eta_{m2}
\end{aligned} \tag{52}$$

where, ρ_k are the autoregressive AR(1) correlation coefficients for each error series k, and $|\rho_k| < 1$ for stationarity, and

$$\delta \sim N(0,1)$$
$$\eta_k \sim N(0, \sigma_k^2 I)$$

where I is the $(M \times M)$ identity matrix, η_k is a $(M \times 1)$ vector of errors, specific to the k^{th} symbol, and δ and η_k are independent.

$$\begin{aligned}
Var[s_{mk}] &= E[(s_{mk} - E[s_{mk}])(s_{mk} - E[s_{mk}])] \\
&= E[(\tau_k\delta_m + \epsilon_{mk})(\tau_k\delta_m + \epsilon_{mk})] \\
&= E[(\tau_k^2\delta_m^2 + 2\tau_k\delta_m\epsilon_{mk} + \epsilon_{mk}^2] \\
&= E[\tau_k^2\delta_m^2 + \epsilon_{mk}^2] \\
&= \tau_k^2 + \frac{\sigma_k^2}{(1-\rho_k^2)}
\end{aligned}$$

$$\begin{aligned}
Cov[s_{m1}, s_{m2}] &= E[(s_{m1} - E[s_{m1}])(s_{m2} - E[s_{m2}])] \\
&= E[(\tau_1\delta_m + \epsilon_{m1})(\tau_2\delta_m + \epsilon_{m2})] \\
&= E[\tau_1\tau_2\delta_m^2 + \tau_1\delta_m\epsilon_{m2} + \tau_2\delta_m\epsilon_{m1} + \epsilon_{m1}\epsilon_{m2}] \\
&= E[\tau_1\tau_2\delta_m^2 + \epsilon_{m1}\epsilon_{m2}] \\
&= \tau_1\tau_2
\end{aligned}$$

For the covariance between $Cov[S_{mk}, S_{(m-1)k}]$, the covariance between two time points of the same series,

$$\begin{aligned}
Cov[s_{mk}, s_{(m-1)k}] &= E[(s_{mk} - E[s_{mk}])(s_{(m-1)k} - E[s_{(m-1)k}])] \\
&= E[(\tau_k\delta_m + \epsilon_{mk})(\tau_k\delta_{(m-1)} + \epsilon_{(m-1)k})] \\
&= E[(\tau_k\delta_m\tau_k\delta_{(m-1)} + \tau_k\delta_m\epsilon_{(m-1)k} + \tau_k\delta_{(m-1)}\epsilon_{mk} + \epsilon_{mk}\epsilon_{(m-1)k}] \\
&= E[\tau_k\delta_m\tau_k\delta_{(m-1)} + \epsilon_{mk}\epsilon_{(m-1)k}]
\end{aligned}$$

where, the expectation of δ is i.i.d, and as such,

$$E[\delta_m\delta_{(m-1)}] = 0$$

and, extending the calculation of $E[\epsilon_{mk}^2]$ above, the expectation of ϵ_{mk} with a lag is,

$$\begin{aligned}
E[\epsilon_{mk}\epsilon_{(m-1)k}] &= E[(\rho_k\epsilon_{(m-1)k} + \eta_{mk})\epsilon_{(m-1)k}] \\
&= E[\rho_k\epsilon_{(m-1)k}^2] \\
&= \rho_k E[\epsilon_{(m-1)k}^2] \\
&= \rho_k E[\epsilon_{mk}^2] \\
&= \rho_k Var[\epsilon_{mk}] \\
&= \rho_k \frac{\sigma_k^2}{(1-\rho_k^2)} \\
&= \frac{\rho_k\sigma_k^2}{(1-\rho_k^2)}
\end{aligned}$$

thus, the covariance at lag 1 of the series k is given by,

$$\begin{aligned}
Cov[s_{mk}, s_{(m-1)k}] &= E[\tau_k^2 \delta_m \delta_{(m-1)} + \epsilon_{mk}\epsilon_{(m-1)k}] \\
&= \tau_k^2 \times 0 + \frac{\rho_k \sigma_k^2}{(1-\rho_k^2)} \\
&= \frac{\rho_k \sigma_k^2}{(1-\rho_k^2)}
\end{aligned}$$

Resulting in the covariance matrix for model 4 of,

$$Cov\begin{pmatrix} s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_{31} \\ s_{32} \end{pmatrix} = \begin{pmatrix}
\tau_1^2 + \frac{\sigma_1^2}{(1-\rho_1^2)} & \tau_1\tau_2 & \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 & \frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & 0 \\
\tau_1\tau_2 & \tau_2^2 + \frac{\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} \\
\frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 & \tau_1^2 + \frac{\sigma_1^2}{(1-\rho_1^2)} & \tau_1\tau_2 & \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 \\
0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \tau_1\tau_2 & \tau_2^2 + \frac{\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} \\
\frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & 0 & \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & 0 & \tau_1^2 + \frac{\sigma_1^2}{(1-\rho_1^2)} & \tau_1\tau_2 \\
0 & \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} & 0 & \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \tau_1\tau_2 & \tau_2^2 + \frac{\sigma_2^2}{(1-\rho_2^2)}
\end{pmatrix}$$

5.2.7 Model 5

As in the previous subsection, Model 5 is the extension of Model 3 except that we now have τ_1 and τ_2 instead of just τ. Thus,

$$\begin{aligned}
&S_k = Z\beta_k + \tau_k\delta + \epsilon_k, \qquad k = 1, 2 \\
&\epsilon_{m1} = \rho_1\epsilon_{(m-1)1} + \eta_{m1} \quad \text{and} \quad \epsilon_{m2} = \rho_2\epsilon_{(m-1)2} + \eta_{m2} \\
&\delta_m = \omega\delta_{(m-1)} + \gamma_m
\end{aligned} \tag{53}$$

where, ρ_k are the autoregressive AR(1) correlation coefficients for each error series k, with $|\rho_k, \omega| < 1$ for stationarity, and

$$\begin{aligned}
\gamma &\sim N(0, 1) \\
\eta_k &\sim N(0, \sigma_k^2 I)
\end{aligned}$$

where I is the $(M \times M)$ identity matrix, η_k is a $(M \times 1)$ vector of errors, specific to the k^{th} symbol, and δ and η_k are independent.

$$\begin{aligned}
Var[s_{mk}] &= E[(s_{mk} - E[s_{mk}])(s_{mk} - E[s_{mk}])] \\
&= E[(\tau_k \delta_m + \epsilon_{mk})(\tau_k \delta_m + \epsilon_{mk})] \\
&= E[(\tau_k^2 \delta_m^2 + 2\tau_k \delta_m \epsilon_{mk} + \epsilon_{mk}^2] \\
&= E[\tau_k^2 \delta_m^2 + \epsilon_{mk}^2] \\
&= \frac{\tau_k^2}{(1-\omega^2)} + \frac{\sigma_k^2}{(1-\rho_k^2)}
\end{aligned}$$

$$\begin{aligned}
Cov[s_{m1}, s_{m2}] &= E[(s_{m1} - E[s_{m1}])(s_{m2} - E[s_{m2}])] \\
&= E[(\tau_1 \delta_m + \epsilon_{m1})(\tau_2 \delta_m + \epsilon_{m2})] \\
&= E[\tau_1 \tau_2 \delta_m^2 + \tau_1 \delta_m \epsilon_{m2} + \tau_2 \delta_m \epsilon_{m1} + \epsilon_{m1} \epsilon_{m2}] \\
&= E[\tau_1 \tau_2 \delta_m^2 + \epsilon_{m1} \epsilon_{m2}] \\
&= \frac{\tau_1 \tau_2}{(1-\omega^2)}
\end{aligned}$$

For the covariance between $Cov[S_{mk}, S_{(m-1)k}]$, the covariance between two time points of the same series,

$$\begin{aligned}
Cov[s_{mk}, s_{(m-1)k}] &= E[(s_{mk} - E[s_{mk}])(s_{(m-1)k} - E[s_{(m-1)k}])] \\
&= E[(\tau_k \delta_m + \epsilon_{mk})(\tau_k \delta_{(m-1)} + \epsilon_{(m-1)k})] \\
&= E[(\tau_k \delta_m \tau_k \delta_{(m-1)} + \tau_k \delta_m \epsilon_{(m-1)k} + \tau_k \delta_{(m-1)} \epsilon_{mk} + \epsilon_{mk} \epsilon_{(m-1)k}] \\
&= E[\tau_k \delta_m \tau_k \delta_{(m-1)} + \epsilon_{mk} \epsilon_{(m-1)k}]
\end{aligned}$$

where,

$$\begin{aligned}
E[\delta_m \delta_{(m-1)}] &= E[\omega \delta_{(m-1)} \delta_{(m-1)} + \lambda_m \delta_{(m-1)}] \\
&= \omega E[\delta_{(m-1)}^2] \\
&= \frac{\omega}{(1-\omega^2)}
\end{aligned}$$

and, extending the calculation of $E[\epsilon_{mk}^2]$ above, the expectation of ϵ_{mk} with a lag is,

$$
\begin{aligned}
E[\epsilon_{mk}\epsilon_{(m-1)k}] &= E[(\rho_k\epsilon_{(m-1)k} + \eta_{mk})\epsilon_{(m-1)k}] \\
&= E[\rho_k\epsilon^2_{(m-1)k}] \\
&= \rho_k E[\epsilon^2_{(m-1)k}] \\
&= \rho_k E[\epsilon^2_{mk}] \\
&= \rho_k Var[\epsilon_{mk}] \\
&= \rho_k \frac{\sigma_k^2}{(1-\rho_k^2)} \\
&= \frac{\rho_k\sigma_k^2}{(1-\rho_k^2)}
\end{aligned}
$$

thus, the covariance at lag 1 of the series k is given by,

$$
\begin{aligned}
Cov[s_{mk}, s_{(m-1)k}] &= E[\tau_k^2\delta_m\delta_{(m-1)} + \epsilon_{mk}\epsilon_{(m-1)k}] \\
&= \frac{\omega\tau_k^2}{(1-\omega^2)} + \frac{\rho_k\sigma_k^2}{(1-\rho_k^2)}
\end{aligned}
$$

$$
Cov\begin{pmatrix} s_{11} \\ s_{12} \\ s_{21} \\ s_{22} \\ s_{31} \\ s_{32} \end{pmatrix} =
$$

$$
\begin{pmatrix}
\frac{\tau_1^2}{(1-\omega^2)} + \frac{\sigma_1^2}{(1-\rho_1^2)} & \frac{\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega\tau_1^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega^2\tau_1^2}{(1-\omega^2)} + \frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega^2\tau_1\tau_2}{(1-\omega^2)} \\
\frac{\tau_1\tau_2}{(1-\omega^2)} & \frac{\tau_2^2}{(1-\omega^2)} + \frac{\sigma_2^2}{(1-\rho_2^2)} & \frac{\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega\tau_2^2}{(1-\omega^2)} + \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \frac{\omega\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega^2\tau_2^2}{(1-\omega^2)} + \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} \\
\frac{\omega\tau_1^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau_1\tau_2}{(1-\omega^2)} & \frac{\tau_1^2}{(1-\omega^2)} + \frac{\sigma_1^2}{(1-\rho_1^2)} & \frac{\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega\tau_1^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau_1\tau_2}{(1-\omega^2)} \\
\frac{\omega\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega\tau_2^2}{(1-\omega^2)} + \frac{\sigma_2^2}{(1-\rho_2^2)} & \frac{\tau_1\tau_2}{(1-\omega^2)} & \frac{\tau_2^2}{(1-\omega^2)} + \frac{\sigma_2^2}{(1-\rho_2^2)} & \frac{\omega\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega\tau_2^2}{(1-\omega^2)} + \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} \\
\frac{\omega^2\tau_1^2}{(1-\omega^2)} + \frac{\rho_1^2\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega^2\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega\tau_1^2}{(1-\omega^2)} + \frac{\rho_1\sigma_1^2}{(1-\rho_1^2)} & \frac{\omega\tau_1\tau_2}{(1-\omega^2)} & \frac{\tau_1^2}{(1-\omega^2)} + \frac{\sigma_1^2}{(1-\rho_1^2)} & \frac{\tau_1\tau_2}{(1-\omega^2)} \\
\frac{\omega^2\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega^2\tau_2^2}{(1-\omega^2)} + \frac{\rho_2^2\sigma_2^2}{(1-\rho_2^2)} & \frac{\omega\tau_1\tau_2}{(1-\omega^2)} & \frac{\omega\tau_2^2}{(1-\omega^2)} + \frac{\rho_2\sigma_2^2}{(1-\rho_2^2)} & \frac{\tau_1\tau_2}{(1-\omega^2)} & \frac{\tau_2^2}{(1-\omega^2)} + \frac{\sigma_2^2}{(1-\rho_2^2)}
\end{pmatrix}
$$

Given the above 5 models, we are yet to explain their estimation. We do so in the upcoming section.

5.3 H-Likelihood for bivariate time series

While Models 1 and 2 in the previous section are known and readily available over many software packages, models 3 to 5 are new. The process of fitting models 3, 4 and 5 was difficult. Attempts at extending previous r packages such as "nlme"[49]to accomodate the new covariance matrices were unsuccessful. We believe these are possible, but due to time imitations we did not pursue these further. Thus, we turned to hierarchical log-likelihood to overcome the issue of fit.

H-likelihood was first introduced by [36] and it permits the inference of both fixed and random unknowns. It is a method that allows us to fit generalized linear models with a dependence structure. A simplified view is that it treats unobserved random effects as parameters to be estimated.

We now present the h-likelihood and the process of its calculation. To limit the repetition of 4 models as in the previous section, we consider only Model 3. Model 2 is a simpler version of this where $\omega = 0$, while Models 4 and 5 can be easily extended from the calculations, and the necessary step will be noted where it occurs.

The h-likelihood for Model 3, as seen in equation (48), is expressed as,

$$\ell^H(\delta, \beta_k, \sigma_k, \rho_k | S_k) = \prod_{k=1,2} f(S_k|\delta; \beta_k, \rho_k, \sigma_k) f(\delta|\tau, \omega) \tag{54}$$

The first term $f(S_k|\delta; \beta, \rho_k, \sigma_k)$ is the conditional distribution of the S_k series given the δ. The second term is the density of the δ, where for Model 3, we have, $\delta \sim N(0, \tau^2\Omega)$. We now outline the steps necessary for the estimation of the h-likelihood, for the case of Model 3,

Step 0.1: Initialise by setting the vector of random effects, δ, to zero.

Step 0.2: Given the current values for δ, we calculate the β_k, by fitting an generalized least squares regression (GLS) model to the adjusted outcome, $S_k - \delta$.

Step 0.3: With our new estimates of β_k from Step 0.2, we calculate our residuals, r_k, where $r_k = S_k - Z_k^T \beta_k$.

Using these residuals, we can calculate initial values for ω, τ^2, ρ_k and σ_k^2, in that order specifically. The calculation of these parameters is performed by using theoretical method of moments from equations in Section 3, resulting in the following outcomes:

$$\omega = \widehat{cov}(r_{m1}, r_{(m-1)2}) / \widehat{cov}(r_{m1}, r_{m2})$$

$$\tau^2 = \widehat{cov}(r_{m1}, r_{m2}) / (1 - \omega^2),$$

$$\rho_k = \frac{\widehat{cov}(r_{mk}, r_{(m-1)k}) - \omega\tau^2/(1-\omega^2)}{\widehat{cov}(r_{mk}, r_{mk}) - \tau^2/(1-\omega^2)},$$

$$\sigma_k^2 = (1 - \rho_k^2)\left(\widehat{cov}(r_{mk}, r_{mk}) - \frac{\tau^2}{(1-\omega^2)}\right).$$

We set these initial values as $\theta = (\sigma_1, \rho_1, \sigma_2, \rho_2, \tau, \omega)$. With our initial values estimated we move on to the estimation of the parameters.

Step 1: Given the current values for δ, we calculate the β_k, by fitting an generalized least squares regression (GLS) model to the adjusted outcome, $S_k - \delta$.

Step 2: With our new estimates of β_k from Step 1, we calculate our residuals, r_k, where $r_k = S_k - Z_k^T \beta_k$.

We use the values of $\theta = (\sigma_1, \rho_1, \sigma_2, \rho_2, \tau, \omega)$, as our initial values for the minimisation of equation (55).

We minimise equation (55) by utilising the optim function in R[50] with the quasi-Newton method "L-BFGS-B" [72] which allows us to implement constraints such as $|\rho_k, \omega| < 1$ and $\tau, \sigma_k \geq 0$.

$$min \sum_1^7 f_i^2 \tag{55}$$

where f_i are,

$$\begin{aligned}
f_1 &= \tau^2/(1-\omega^2) + \sigma_1^2/(1-\rho_1^2) - var(r_{m1}) \\
f_2 &= \tau^2/(1-\omega^2) + \sigma_2^2/(1-\rho_2^2) - var(r_{m2}) \\
f_3 &= \tau^2/(1-\omega^2) - cov(r_{m1}, r_{m2}) \\
f_4 &= \big(\omega\tau^2/(1-\omega^2) + \rho_1\sigma_1^2/(1-\rho_1^2)\big) - cov(r_{m1}, r_{(m-1)1}) \\
f_5 &= \big(\omega\tau^2/(1-\omega^2) + \rho_2\sigma_2^2/(1-\rho_2^2)\big) - cov(r_{m2}, r_{(m-1)2}) \\
f_6 &= \big(\omega^2\tau^2/(1-\omega^2) + \rho_1^2\sigma_1^2/(1-\rho_1^2)\big) - cov(r_{m1}, r_{(m-2)1}) \\
f_7 &= \big(\omega^2\tau^2/(1-\omega^2) + \rho_2^2\sigma_2^2/(1-\rho_2^2)\big) - cov(r_{m2}, r_{(m-2)2})
\end{aligned}$$

Minimising equation (55) results in updated θ. Each of the f_i are derived from the covariance matrix for Model 3 as in equation (49).

Step 3: Given our new updated θ from Step 2, we find the values of δ that maximise expression (54). Steps required to do this are discussed below. We update our values for δ.

Step 4: Iterate Steps 1, 2 and 3 until convergence. Convergence is measured once $|\theta_i - \theta_{i-1}| < 0.001$ for each iteration I.

Step 3 requires more discussion, we present this here.

To find the values of δ that maximise expression (54), a complex algebraic solution is possible. However, we consider another option. By keeping the parameters estimated in Step 3 constant, we note that the expression (54) is proportional to the conditional distribution of δ given S_1 and S_2. We note this link as S_1, S_2 and δ all have normal distributions, as such we can utilise the properties of multivariate normal distributions to determine this conditional distribution.

We calculate the maximum h-likelihood estimator of δ by finding the mean of the conditional distribution $f(\delta|S_1, S_2)$. The first term $f(S_k|\delta; \beta, \rho_k, \sigma_k)$ is the conditional distribution of the S_k series given the δ. The second term is the density of the δ, where for each model,

- Model 2: $\delta \sim N(0, \tau^2 I)$ and $\omega = 0$
- Model 3: $\delta \sim N(0, \tau^2 \Omega)$

- Model 4: $\delta \sim N(0, \tau_k^2 I)$ and $\omega = 0$
- Model 5: $\delta \sim N(0, \tau_k^2 \Omega)$

We note that I and Ω are defined at the end of this section in the interest of formatting.

To find this, note from equation (48), that the joint distribution of δ, S_1 and S_2 is

$$\begin{pmatrix} \delta \\ S_1 \\ S_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ Z_1\beta_1 \\ Z_2\beta_2 \end{pmatrix}, \begin{pmatrix} \tau^2\Omega & \tau^2\Omega & \tau^2\Omega \\ \tau^2\Omega & \tau^2\Omega + \Sigma_1 & \tau^2\Omega \\ \tau^2\Omega & \tau^2\Omega & \tau^2\Omega + \Sigma_2 \end{pmatrix}\right), \tag{56}$$

we determine the conditional mean of δ given S_1 and S_2, and thereby the estimated δ's as

$$\hat{\delta} = \begin{pmatrix} \tau^2\Omega & \tau^2\Omega \end{pmatrix} \begin{pmatrix} \tau^2\Omega + \Sigma_1 & \tau^2\Omega \\ \tau^2\Omega & \tau^2\Omega + \Sigma_2 \end{pmatrix}^{-1} \begin{pmatrix} S_1 - Z_1\beta_1 \\ S_2 - Z_2\beta_2 \end{pmatrix}. \tag{57}$$

Here, I is the identity matrix,

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and Ω is the matrix that considers the AR(1) random effect,

$$\Omega = \begin{pmatrix} \frac{1}{(1-\omega^2)} & \frac{\omega}{(1-\omega^2)} & \frac{\omega^2}{(1-\omega^2)} & \cdots & \frac{\omega^{m-1}}{(1-\omega^2)} \\ \frac{\omega}{(1-\omega^2)} & \frac{1}{(1-\omega^2)} & \frac{\omega}{(1-\omega^2)} & \cdots & \frac{\omega^{m-2}}{(1-\omega^2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\omega^{m-1}}{(1-\omega^2)} & \frac{\omega^{m-2}}{(1-\omega^2)} & \frac{\omega^{m-3}}{(1-\omega^2)} & \cdots & \frac{1}{(1-\omega^2)} \end{pmatrix},$$

Here, Σ_k is the matrix of AR(1) errors,

$$\Sigma_k = \begin{pmatrix} \frac{1}{(1-\rho_k^2)} & \frac{\rho_k}{(1-\rho_k^2)} & \frac{\rho_k^2}{(1-\rho_k^2)} & \cdots & \frac{\rho_k^{m-1}}{(1-\rho_k^2)} \\ \frac{\rho_k}{(1-\rho_k^2)} & \frac{1}{(1-\rho_k^2)} & \frac{\rho_k}{(1-\rho_k^2)} & \cdots & \frac{\rho_k^{m-2}}{(1-\rho_k^2)} \\ \frac{\rho_k^2}{(1-\rho_k^2)} & \frac{\rho_k}{(1-\rho_k^2)} & \frac{1}{(1-\rho_k^2)} & \cdots & \frac{\rho_k^{m-3}}{(1-\rho_k^2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\rho_k^{m-1}}{(1-\rho_k^2)} & \frac{\rho_k^{m-2}}{(1-\rho_k^2)} & \frac{\rho_k^{m-3}}{(1-\rho_k^2)} & \cdots & \frac{1}{(1-\rho_k^2)} \end{pmatrix}.$$

Having presented our models and algorithms in this section, we are interested in how well they perform. In the following section we present a simulation study to determine their effectiveness.

5.4 Simulations: Fits of models 2 through 5

In the previous section, we presented the H-Likelihood and introduced our extension of the the H-Likelihood to include an AR(1) structure on the Random Effects. We now evaluate the fit of these models on some simulated data. In section 3 we presented five models:

1. Model 1: Independent errors.
2. Model 2: Random Effects with AR(1) errors.
3. Model 3: AR(1) Random Effects and AR(1) errors.
4. Model 4: Random Effects with AR(1) errors and seperate τ_k constant.
5. Model 5: AR(1) Random Effects and AR(1) errors and seperate τ_k constant.

Model 1 is fit by the lme function in R package nlme, and as we are using a linear mixed effects model that does not introduce anything new, we do not analyse the fit of model 1 in this section. The simulations for fitting models 2 to 5 are presented here.

We are interested in determining how well models 2 to 5 can be fit to simulated data, as we will be utilising them on the applied coal train data set in section 6. We explore the fit of these two models in 2 stages:

1. We run an analysis using the complete model on a simulated set of parameters. The results are shown for 50 replicates of the analysis on a data set of size $n = 1,000$.
2. To account for current limitations in data size, we rerun these simulations above with 3 n sizes: $n = 1,000$, $n = 10,000$ and $n = 100,000$. Due to the covariance matrix's dense structure and it's inversion difficulties, we set our β estimates to zero and run the simulations. The β estimates are set to zero for all Method of Moments simulations in this section. Models 2 and 3 have a closed form solution to estimate the variance components and correlation coefficients. However Models 4 and 5 are not so fortunate, and we thus utilise the r:optim function and it's iterative numerical procedure to estimate the parameters.

First we present our data-generating model for these simulations, followed by both stages of fitting the models.

5.4.1 Data generating model

To accurately assess the fit of models 2 to 5, we utilise a data generating model of from,

$$y_k = \beta_{0k} + \beta_{1k}x + \delta + \epsilon_k,$$

where $k = 1, 2$. The errors have an AR(1) autoregressive structure with $\epsilon_{mk} = \rho_k \epsilon_{(m-1)k} + \eta_{mk}$, as do the random effects, $\delta_m = \omega\delta_{m-1} + \gamma_m$. The covariate x is created with a normal distribution $x \sim N(0, 1)$. The random effects follow the distribution $\gamma \sim N(0, \tau^2 I)$, or $\gamma \sim N(0, \tau_k^2 I)$ for Models 4 and 5, and the errors, $\eta \sim N(0, 1)$.

Table 5.1: Fixed effects parameter selection for data generation all models

Parameter	Value
β_{01}	3
β_{02}	5
β_{11}	5
β_{12}	7

Tables 5.1 and 5.2 presents the parameter values selected for the data generating model. Models 2 and 4 are the special case of models 3 and 5 where $\omega = 0$ and is calculated as such when fitting for model 2 and 4.

For the complete analysis in each of model 2 through 5, we use a data size of $n = 1,000$. For the method of moments, we use 3 data sizes for each replicate. Here $n = 1,000$; $n = 10,000$ and $n = 100,000$. This is possible as we are only calculating the expected values and thereby bypass the storage and inversion of the covariance matrix, removing the limitations on computer memory and processing power.

5.4.2 Fitting model 2

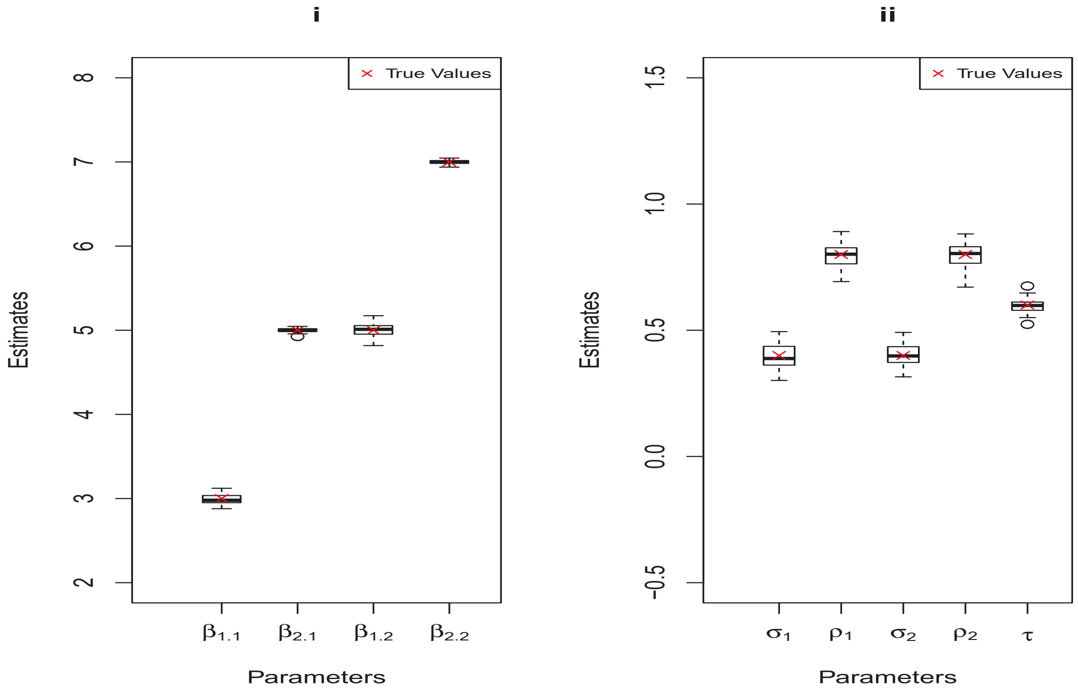
Model 2, as introduced in equation (42), is a mixed effects model which considers an autocorrelated error structure and a random effect. The random effect is utilised to account for correlation between two series, while the autocorrelation of lag 1 is implemented on the errors of each series y_k.

In figure 5.1 we present our results for 50 replicates of the analysis. Here we have simulated the data and run the analyses, with the boxplots showing the variation of fit for the replicates. The plot on the left (i) shows the fixed effects, and the plot on the right shows the variance components and correlation coefficients. From the minimal variation in the boxplots about the true value (marked with a red cross), it is evident that the model is fitting accurately. For this figure, we only run the parameters from option 1 as outlined in Table 5.2.

Table 5.2: Parameter selection for data generation Model 2

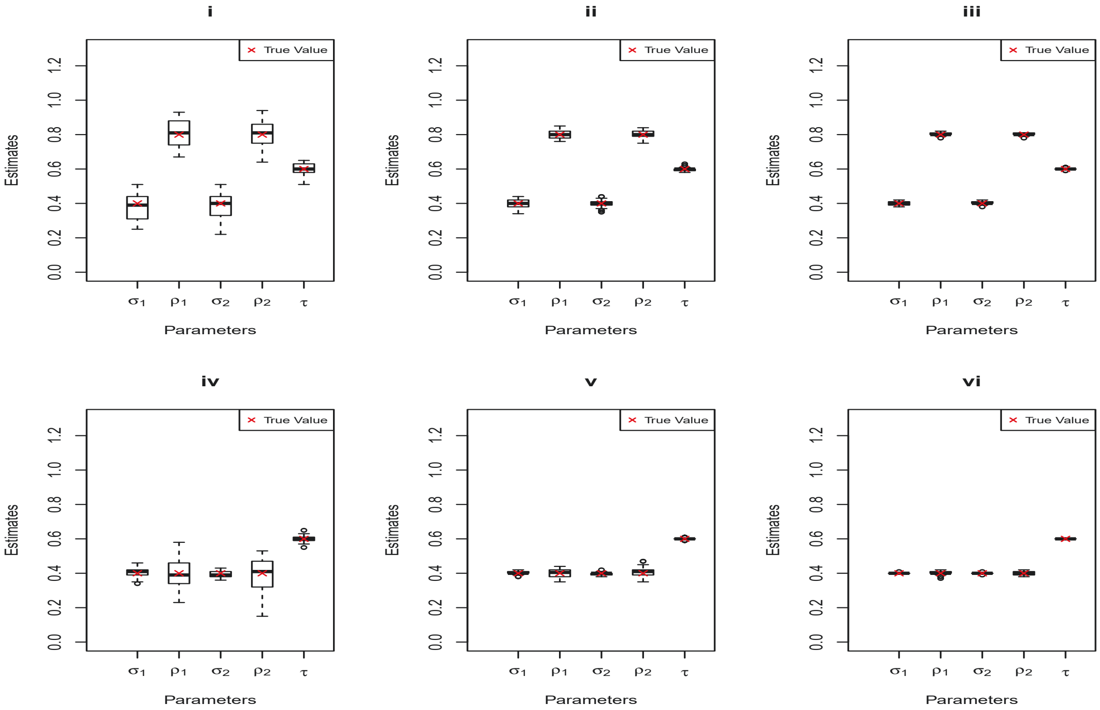
Parameter	Option 1	Option 2
σ_1	0.4	0.4
σ_2	0.4	0.4
ρ_1	0.8	0.4
ρ_2	0.8	0.4
τ	0.6	0.6
ω	0	0

Figure 5.1: Model 2 Simulations, 50 Replicates; plot (i) shows the beta coefficient estimates, and plot (ii) shows the variance components



Considering the fact that we are limited to a small data set size for the analysis due to the memory constraints of inverting the covariance matrix, we run some more analyses without estimating the fixed effects. This allows us to bypass the covariance matrix inversion, and we can see how our model fits for a larger sample size. In figure 5.2 below, we can see that as we increase the size of the dataset, the models fit improves. The top row of the figure containing plots (i), (ii), and (iii) shows the simulations using Option 1 in table 5.2, and the bottom row containing plots (iv), (v), and (vi) shows the simulations using Option 2 in table 5.2. We consider 2 options for parameter values to see if different variance components effect our model fit. With a smaller sample size there is more variation, but as we increase the data size, both options fit accurately.

Figure 5.2: Method of Moments Model 2: (i) & (iv) have $n = 1,000$; (ii) & (v) have $n = 10,000$; and (iii) & (vi) have $n = 100,000$.



From figures 5.1 and 5.2 above, we determine that model 2 has an effective fit. We now consider the fit for model 3, where $\omega \neq 0$.

5.4.3 Fitting model 3

Model 3, as introduced in equation (48), extends on model 2 by allowing for an AR(1) correlation structure on both the errors and the random effects.

Following the format of section 5.2, we first present our results for 50 replicates of the analysis for model 3.

In figure 5.3 we present our results for 50 replicates of the analysis. Here we have simulated the data and run the analyses, with the boxplots showing the variation of fit for the replicates. The plot on the left (i) shows the fixed effects, and the plot on the right shows the variance components and correlation coefficients. From the minimal variation in the boxplots about the true value (marked with a red cross), it is evident that the model is fitting accurately. For this figure, we only run the parameters from option 1 as outlined in table 5.3.

Figure 5.3: Model 3 Simulations, 50 Replicates; plot (i) shows the beta coefficient estimates, and plot (ii) shows the variance components

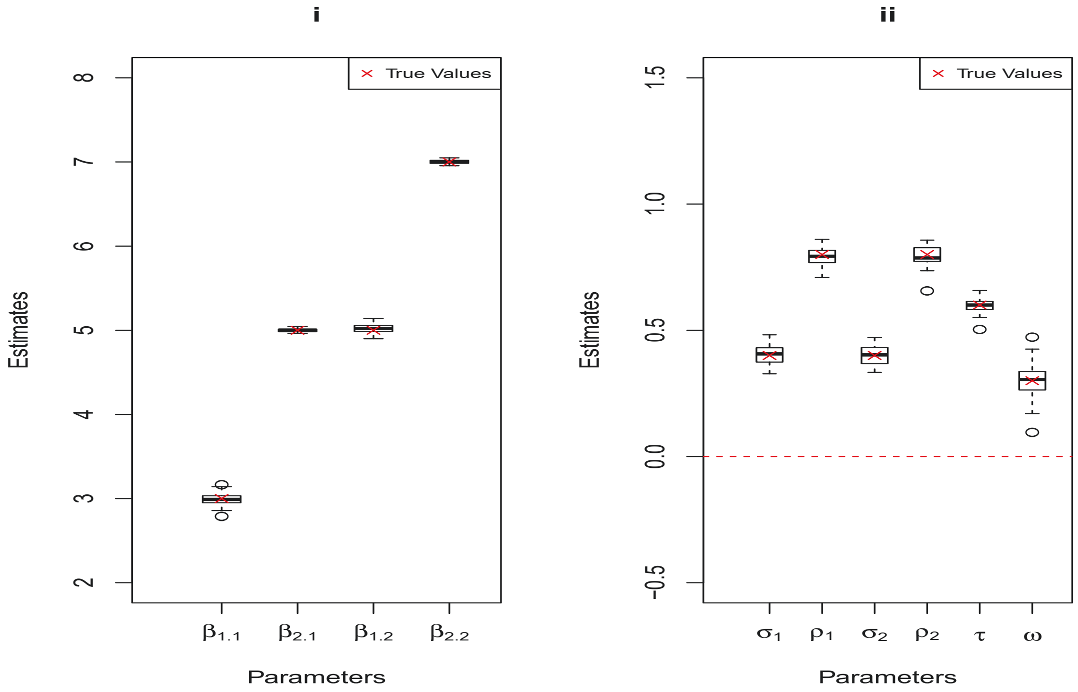
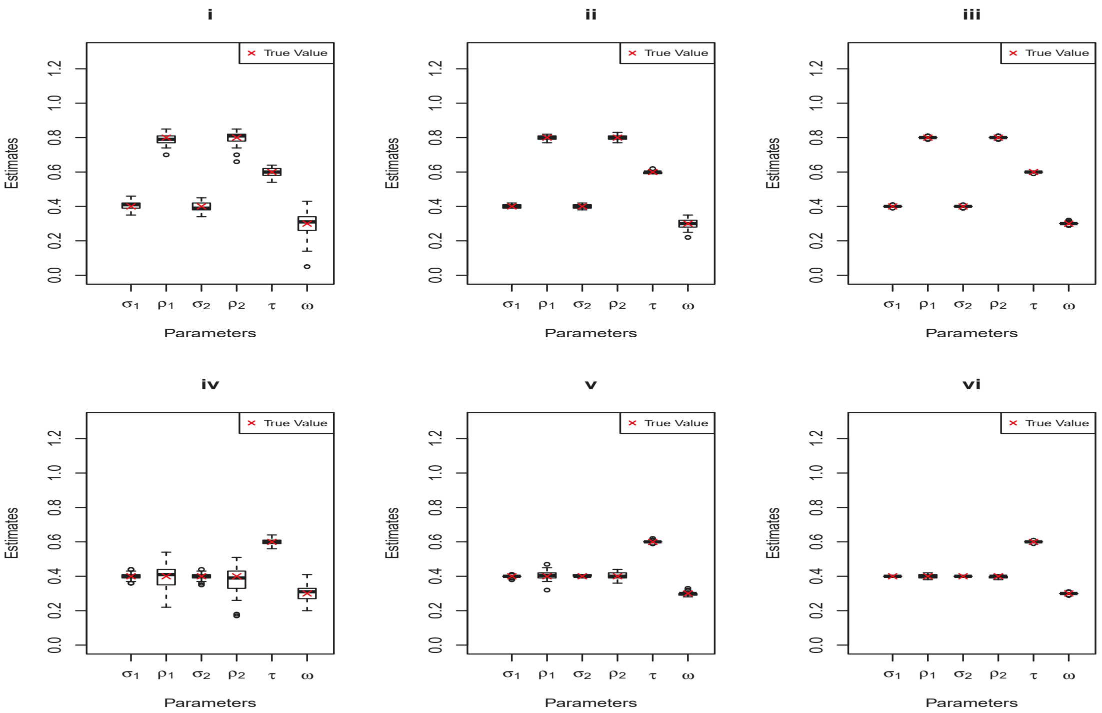


Table 5.3: Parameter selection for data generation Model 3

Parameter	Option 1	Option 2
σ_1	0.4	0.4
σ_2	0.4	0.4
ρ_1	0.8	0.4
ρ_2	0.8	0.4
τ	0.6	0.6
ω	0.3	0.3

Again we consider a larger data set size and run some more simulations. Figure 5.4, as in the previous section, outlines that the variation in the model fit is due to the small sample size. Model 3 has a precise fit as can be seen in plots (iii) and (vi) where we have a simulated dataset of length $n = 100,000$.

Figure 5.4: Method of Moments Model 3: (i) & (iv) have $n = 1,000$; (ii) & (v) have $n = 10,000$; and (iii) & (vi) have $n = 100,000$.



In a repeat of the fit of model 2, model 3 is again largely affected by sample size. This is evident from figure 5.4, where as the sample size increases, the variability in our estimates reduces.

We now consider the fit of Model 4, where we are introducing a seperate value for τ_k.

5.4.4 Fitting model 4

Model 4, as introduced in equation (52), is an extension of Model 2, whereby we allow for a seperate random effect multiplier for each series. Therefore we have extended τ to τ_k.

In figure 5.5 we present our results for 50 replicates of the analysis. Here we have simulated the data and run the analyses, with the boxplots showing the variation of fit for the replicates. The plot on the left (i) shows the fixed effects, and the plot on the right shows the variance components and correlation coefficients. From the minimal variation in the boxplots about the true value (marked with a red cross), it is evident that the model is fitting accurately. For this figure, we only run the parameters from option 1 as outlined in table 5.4.

Figure 5.5: Model 4 Simulations, 50 Replicates; plot (i) shows the beta coefficient estimates, and plot (ii) shows the variance components

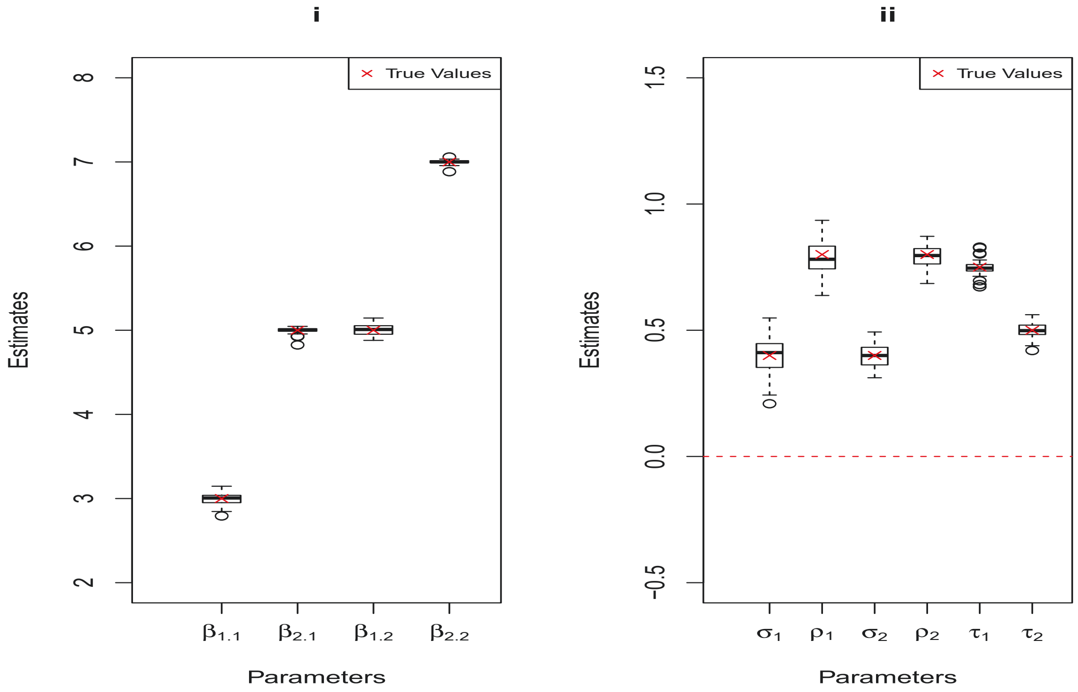
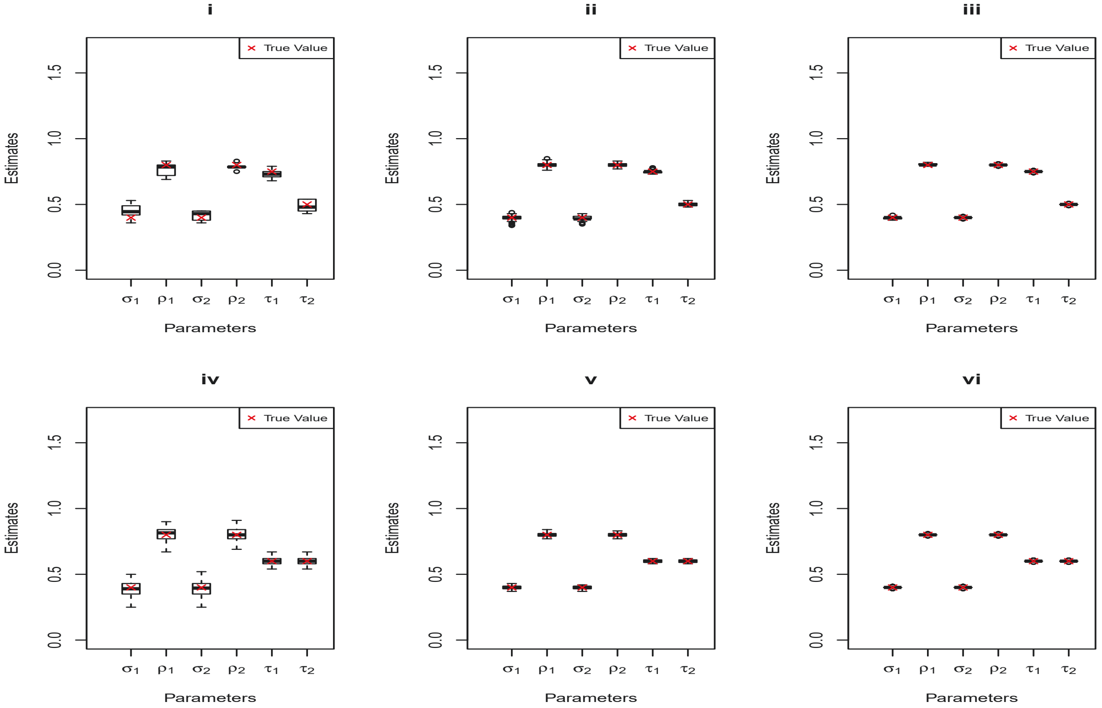


Table 5.4: Parameter selection for data generation Model 4

Parameter	Option 1	Option 2
σ_1	0.4	0.4
σ_2	0.4	0.4
ρ_1	0.8	0.8
ρ_2	0.8	0.8
τ_1	0.75	0.6
τ_2	0.5	0.6

We consider Option 2 in table 5.4 that differs in the values for τ_k. We are interested in the initial estimate of the τ_k value and its impact on the model fit. This is of particular interest, as we cannot find a closed form solution to each τ_k and as such we set $\tau_1 = \tau_2$ as our initial value. The only difference in the top and bottom rows of figure 5.6 is this τ_k initial value. There fit of Model 4 in figure 5.6 is more accurate when we have $\tau_1 = \tau_2$. We can see that the variation in plots (iv) is less than that in plot (i). However, once we increase the data size, the fit is accurate for both options.

Figure 5.6: Method of Moments Model 4: (i) & (iv) have $n = 1,000$; (ii) & (v) have $n = 10,000$; and (iii) & (vi) have $n = 100,000$.



From figures 5.5 and 5.6, we are again confident of our model fit. We finish our simulations, concerning the fit of the models, with Model 5 below.

5.4.5 Fitting model 5

As in the previous section where Model 2 and 4 were closely related, we again consider the same extension however in this case it's for Model 5, as introduced in equation (53). Again, we allow for a seperate random effect multiplier for each series. Therefore we have extended τ to τ_k.

In figure 5.7 we present our results for 50 replicates of the analysis. Here we have simulated the data and run the analyses, with the boxplots showing the variation of fit for the replicates. The plot on the left (i) shows the fixed effects, and the plot on the right shows the variance components and correlation coefficients. For this figure, we only run the parameters from option 1 as outlined in table 5.5.

This is the first of our simulated models that does not have an accurate fit. In particular from figure 5.7, we can see that the variance components are struggling for accuracy. Of particular interest is the estimates for ρ_2 and σ_2. These are much lower than the true value. Comparing these analyses with those in figure 5.8, we see that increasing the sample size does not improve this fit. What does dramatically improve the estimation is having $\tau_1 = \tau_2$. This is the equivalent of fitting Model 3. Thus in our simulations Model 5 does not differ significantly from Model 3, nonetheless it is can be an effective tool, the struggles in the fit of the model can be attributed to the highly parameterized model where the starting values can significantly impact the outcomes.

Figure 5.7: Model 5 Simulations, 50 Replicates; plot (i) shows the beta coefficient estimates, and plot (ii) shows the variance components

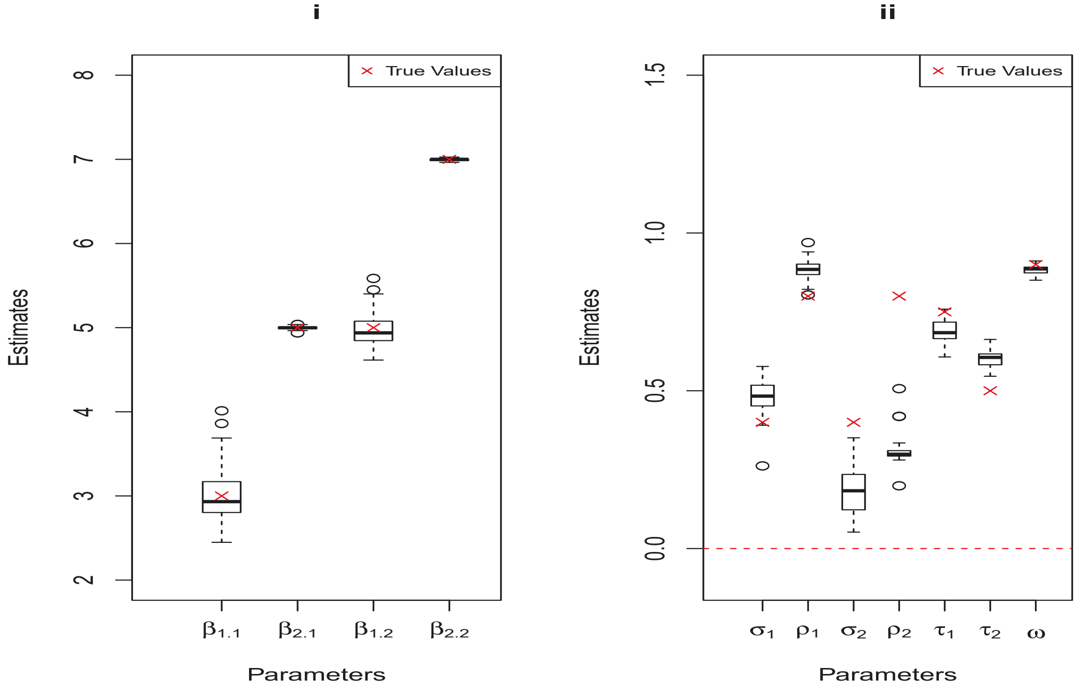
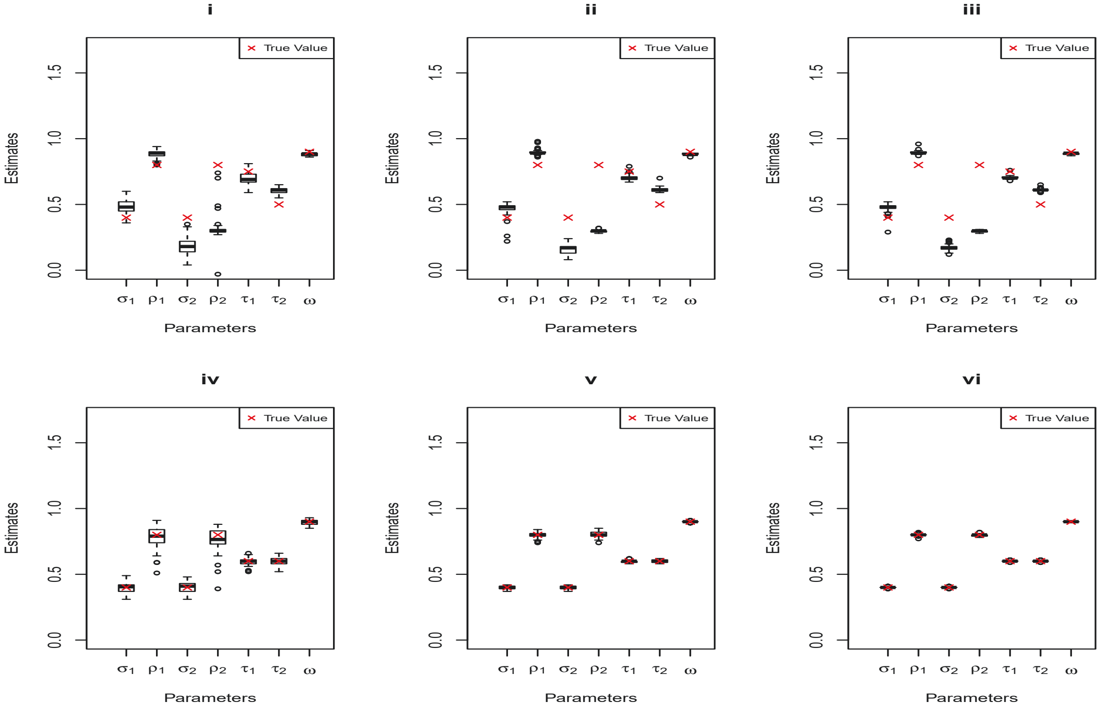


Table 5.5: Parameter selection for data generation Model 5

Parameter	Option 1	Option 2
σ_1	0.4	0.4
σ_2	0.4	0.4
ρ_1	0.8	0.4
ρ_2	0.8	0.4
τ_1	0.75	0.6
τ_2	0.5	0.6
ω	0.9	0.9

Figure 5.8: Method of Moments Model 5: (i) & (iv) have $n = 1,000$; (ii) & (v) have $n = 10,000$; and (iii) & (vi) have $n = 100,000$.



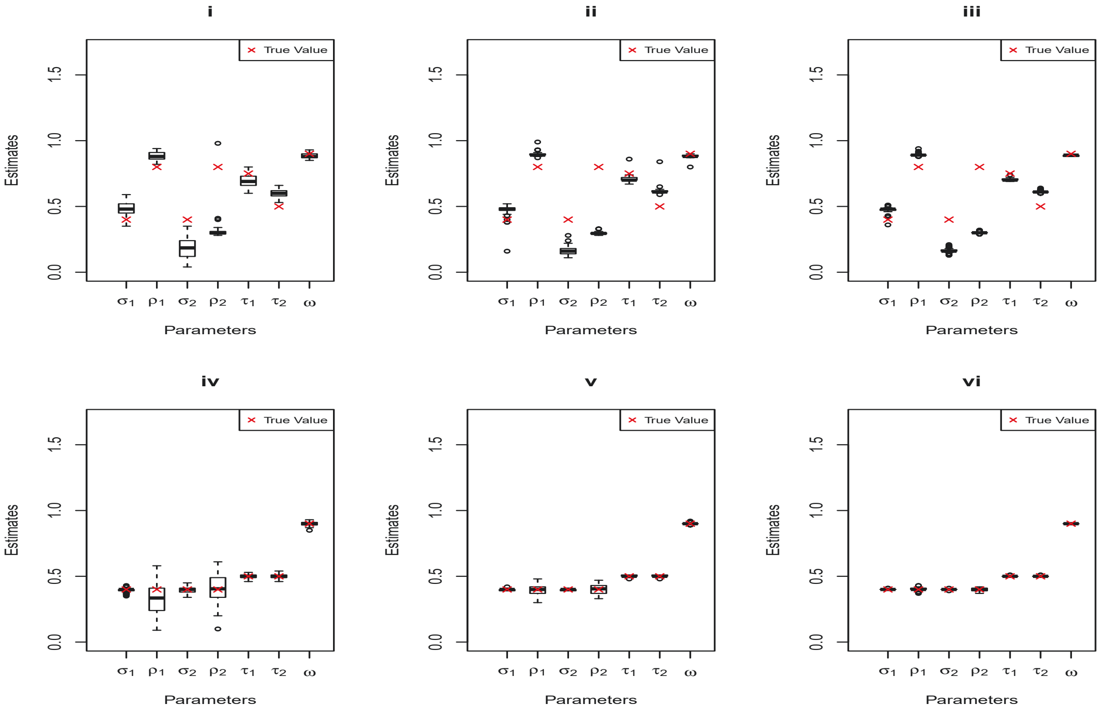
5.4.6 Review of model fits

We first presented models 2 through 5 in section In this section we were concerned with their fit and their accuracy. By running a number of simulations for each of the models, we were able to determine the strengths and weaknesses of the model estimation.

Our main consideration is sample size. As for most of statistics, the larger the data, the more accurate the results, as is clearly the case here. At lower sample sizes, we can see that the models can have difficulty accurately estimating particular parameters.

Model 5 is the only model that has issues with its estimation. Further simulations, as shown in figure 5.9, show that the model is having issues with larger correlation coefficients (ρ_1 and ρ_2). By reducing the correlation coefficients, we have an accurate fit.

Figure 5.9: Method of Moments Model 5b: (i) & (iv) have $n = 1,000$; (ii) & (v) have $n = 10,000$; and (iii) & (vi) have $n = 100,000$.



With confidence in our models from their fitting in this section. we move on to another simulation study where we explore the impact of aggregation on the simulated data.

5.5 Simulations: Effect of aggregation

How does temporal aggregation effect our analysis? In this section we aim to explore this question by running analyses on two different data generating models. We show that temporal aggregation, especially in a time series setting, can simplify the models used for analysis. We present results for four aggregation interval lengths, $J = 10$, $J = 20$, $J = 50$ and $J = 100$.

How does aggregation impact our simulated data set? How does aggregation affect our choice of model? How do we interpret the outcomes after aggregation? There are a number of questions that result from aggregation. Above are a few of the key questions we have. To explore these, and others that may arise from aggregation, we have conducted another simulation study. This simulation differs from that of the previous section as we are considering the effect of aggregation on each model rather than the fit of each model. It is important to explore this topic as we conduct our analysis on the Hunter Valley Coal Train data set in the next section using the models on aggregated data.

After describing the data generating model for our simulation, we undertake the fitting of models 1 and 2. Firstly, on the unaggregated data, and then on the aggregation over a number of interval lengths.

The original Hunter Valley Coal Train data set is extremely large and complex. We've mentioned that this is a simplified case for the simulation and as such we consider 2 options:

1. Independent gaussian error structure.
2. AR(1) error structure on the errors with high τ.

5.5.1 Data generating model

We construct a set of simulations that aim to mimic the coal train data described in Section 2, while being more well behaved, allowing us to consider the effects of aggregation.

Our simulated data set has form

$$y_k = \beta_{0k} + \beta_{1k} x + \delta + \epsilon_k, \tag{58}$$

where $k = 1, 2$. x_k is an indicator covariate for a passing train, with random effects $\delta \sim N(0, \tau^2)$ and ϵ_k is the error term that we discuss presently. Our simulations contain only one passing train at a time with a fixed length of ten observations.

The covariate x_k representing the passing train variable is created by running through one observation at a time and deciding if there is a train present or not according to a Bernoulli random variable with $B(p = 0.05)$. If there is a train present, we let it run for ten observations. Once the train has passed, we set the following two observations to be zero. This is implemented to be consistent with reality, whereby there must be a gap after a train passes before another can follow. If the Bernoulli random variable decides that no train has passed we move onto the next observation. The resulting time series that is created is the covariate x for a passing train.

As outlined above, we simulate two different options for the error structures for the simulations. Each of the parameters are presented as we simulate in their subsections. We aggregate equation (58) in the manner as outlined in equation (38), namely we aggregate the y_{nk} into s_{mk} over means,

$$s_{mk} = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} y_n$$

Here J is the aggregation block length choice. We perform the aggregation over a range of choices for J, specifically $J = 10, 20, 50, 100$. After we have aggregated each series, noted as $k = 1, 2$, the resulting bivariate series are,

$$S_1 = (s_{11}, s_{21}, s_{31}, \ldots, s_{M1})^T$$

and

$$S_2 = (s_{12}, s_{22}, s_{32}, \ldots, s_{M2})^T .$$

We also apply the aggregation to the covariates of interest. In this case we take proportions as the aggregation method for a particular covariate,

$$z_{mq} = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} x_{np}. \tag{59}$$

Thus our original train covariate x_1 transform to z_1. For these simulations we only have one covariate therefore x_{n1} transforms to z_{m1}.

We consider 2 options for the error terms $\epsilon = 1, \ldots, n$. In the first option, we set the errors as independent with a $N \sim (0, 1)$ distribution. For option 2 we consider the implementation of serial correlation in the errors. We set the errors as having an autoregressive AR(1) structure.The correlation coefficients are $\rho_1 = 0.8$.

Table 5.6 below indicates the fixed effects values for all the simulated datasets utilised in this section. These are the true values for equation (58).

Table 5.6: Parameter selection for data generation Options 1 and 2

Parameter	Value
β_{01}	3
β_{02}	5
β_{11}	5
β_{12}	7

5.5.2 Fitting for data from option 1

Before we begin fitting model 1 to the data we present the simulated data and the effect of temporal aggregation on the data. The data was simulated using the data generating model in equation (58), with the error structure, namely independent gaussian errors, being simulated using the parameters shown in table 5.7.

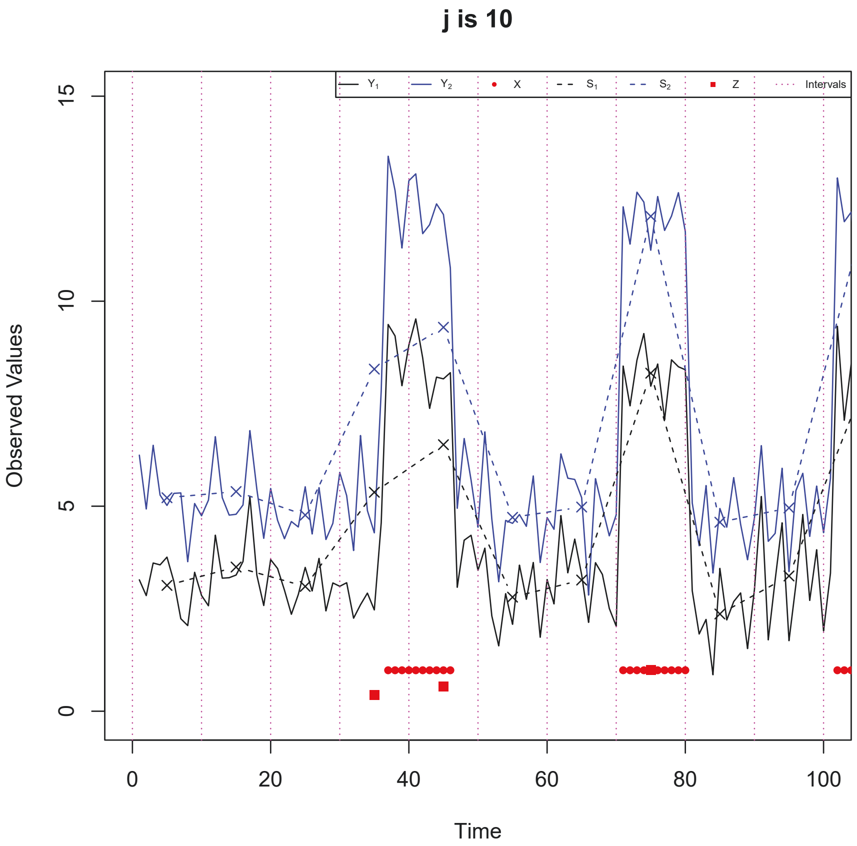
The aggregation of the response variable is conducted as shown in equation (38). All intervals were analysed and the results are displayed below. We also aggregate the covariate. We are interested in the percentage of time a train was passing in each interval. This calculation is conducted as shown in equation (39).

Table 5.7: Parameter selection for data generation Option 1

Parameter	Case 1
σ_1	0.4
σ_2	0.4
ρ_1	0
ρ_2	0
τ	0.6

Figure 5.10 shows the two unaggregated series S_1 and S_2 in the solid black and blue lines. Their resulting aggregated series, for the aggregation interval length of $J = 10$, are in the same colours but are the dashed lines. The passing train indicator values, X, are shown by the red dots, and their aggregated Z value is indicated by the red squares. The impact of a passing train is visible in the marked jump in each series. The jumpy nature of both series is attributed to the lack of an autoregressive error structure. Although we are simulating datasets of size $n = 10,000$, we can only manage to show the first 100 observations. Any more and there would be no discernible features in this figure.

Figure 5.10: First 100 observations of Option 1 simulated dataset including aggregated data



Figures 5.11 and 5.12 show the effect of aggregation on this simulated data. We have run 50 replicates of the dataset and analysed the results. Figure 5.11 shows the effect aggregation has on the fixed effects. We can see that aggregation does not have any negative effect here. All the fixed effects are constant after aggregation.

The variance components are greatly affected by the aggregation. As we increase the aggregation interval length J, from $J = 10$ to $J = 100$, we can see that the cross-covariance, as determined through the random effects, τ reduces towards zero. The variances of each series, σ_k also reduces towards zero. This is as we would expect, as by aggregating both series, we are reducing the variability in each.

Figure 5.11: Aggregation Effect for Model 1:i. J=10, ii. J=20, iii. J=50, iv. J=100 ; n=10,000, with red crosses indicating the initial true parameter as set in the unaggregated data simulation; 5 replicates; Beta Coefficients

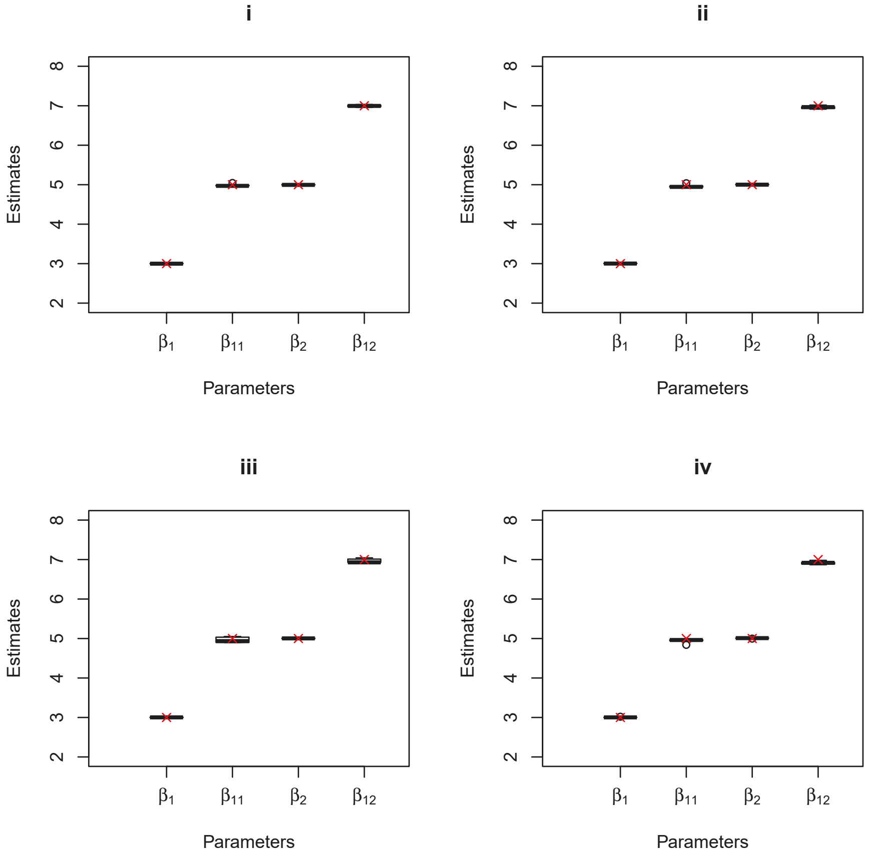
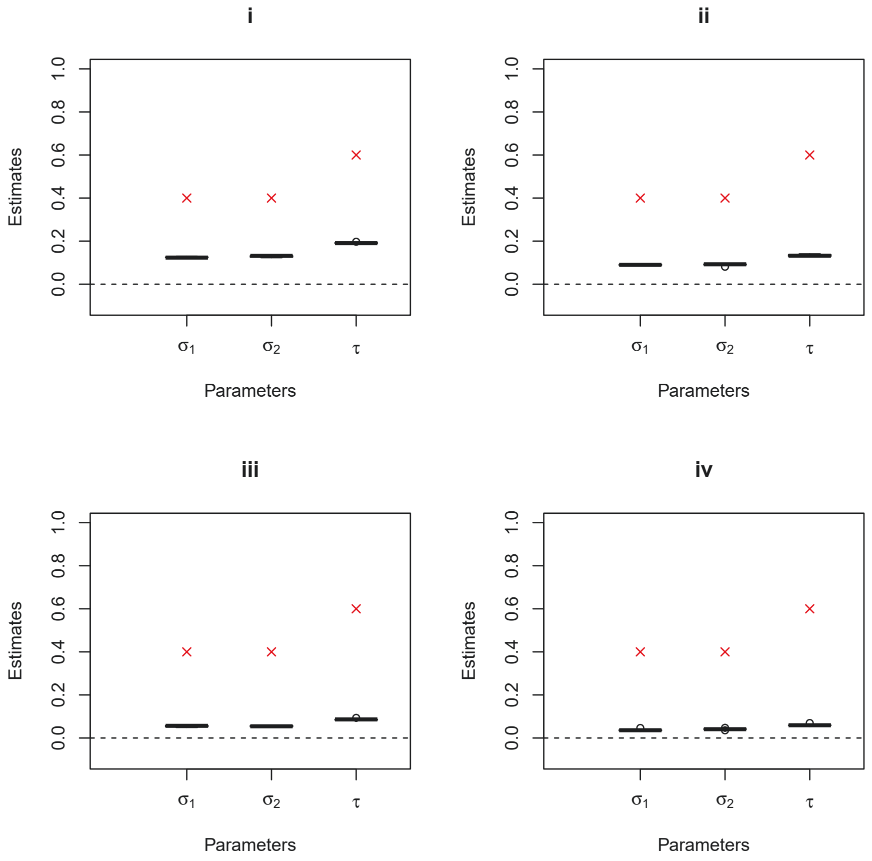


Figure 5.12: Aggregation Effect for Model 1:i. J=10, ii. J=20, iii. J=50, iv. J=100 ; n=10,000, with red crosses indicating the initial true parameter as set in the unaggregated data simulation; 5 replicates; Variance Components



From the table and figures above we note that there is no cross-correlation between the series as evidenced by the τ coefficient approaching to zero from aggregation. Further there is no evidence of autocorrelation in either of the error series ϵ_k and as such Model 1 is the ideal fit for this dataset.

5.5.3 Fitting for data from option 2

The nature of the applied dataset for this paper is that of a time series model. The coal train data has a strong autocorrelation present. As such we cannot hope to adequately model the coal data with Model 1. Thus, we move on to the next model; Model 2. This model allows for an autoregressive AR(1) series on each series y_k.

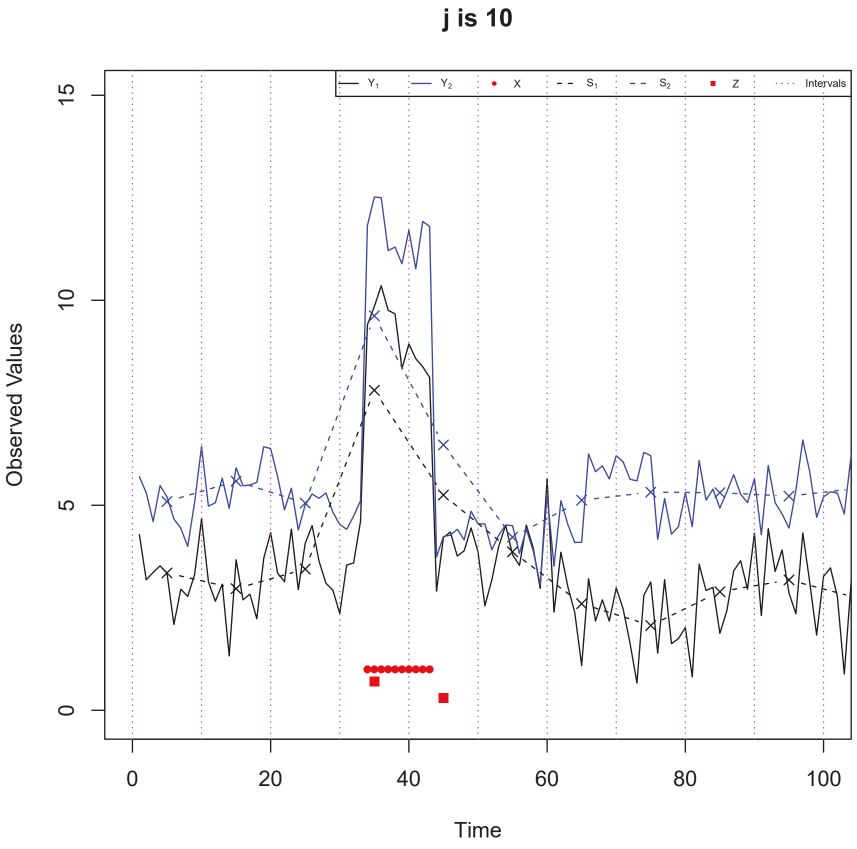
As for the fitting of the data for option 1 as presented in the previous subsection, we first generate our simulated data. Again we use equation (58), however this time with an AR(1) error structure. The correlation coefficients for each series, ρ_1 and ρ_2 are shown in table 5.8. Following the generation of the data we aggregate the series into a bivariate series of means. The covariate X is again aggregated into Z.

Table 5.8: Parameter selection for data generation Option 2

Parameter	Value
σ_1	0.4
σ_2	0.4
ρ_1	0.8
ρ_2	0.8
τ	0.6

Figure 5.13 shows one of our simulated data sets for Option 2.

Figure 5.13: First 100 observations of Option 2 simulated dataset including aggregated data



Comparing the figure 5.13 with that of the i.i.d gaussian errors in figure 5.10, we can see that the induced correlation structure leads to a smoother effect on the observations of each series. This is the effect of autocorrelation on the data. Comparing figures 5.15 with 5.12 shows that the variances of the error series σ_1^2 and σ_2^2 are differently affected by autocorrelation and aggregation. In figure 5.15 the σ_1^2 and σ_2^2 do not reduce as much when the aggregation block length is increased. However, as we increase the length of aggregation, we can see that the correlation coefficients for ρ_1 and ρ_2 reduce to zero. This means that at the lower levels of aggregation we should be using Model 2 to fit the data, while at a higher level such as $J = 100$, temporal aggregation has resulted in a removal of autocorrelation and we can fit the data using Model 1.

Model 2 fits the fixed effects accurately as can be seen in figure 5.14, the increasing variation seen in plot (iv) can be attributed to the reduced data size from aggregation.

Figure 5.14: Aggregation Effect for Model 2:i. J=10, ii. J=20, iii. J=50, iv. J=100 ; n=10,000, with red crosses indicating the initial true parameter as set in the unaggregated data simulation; 5 replicates; Beta Coefficients

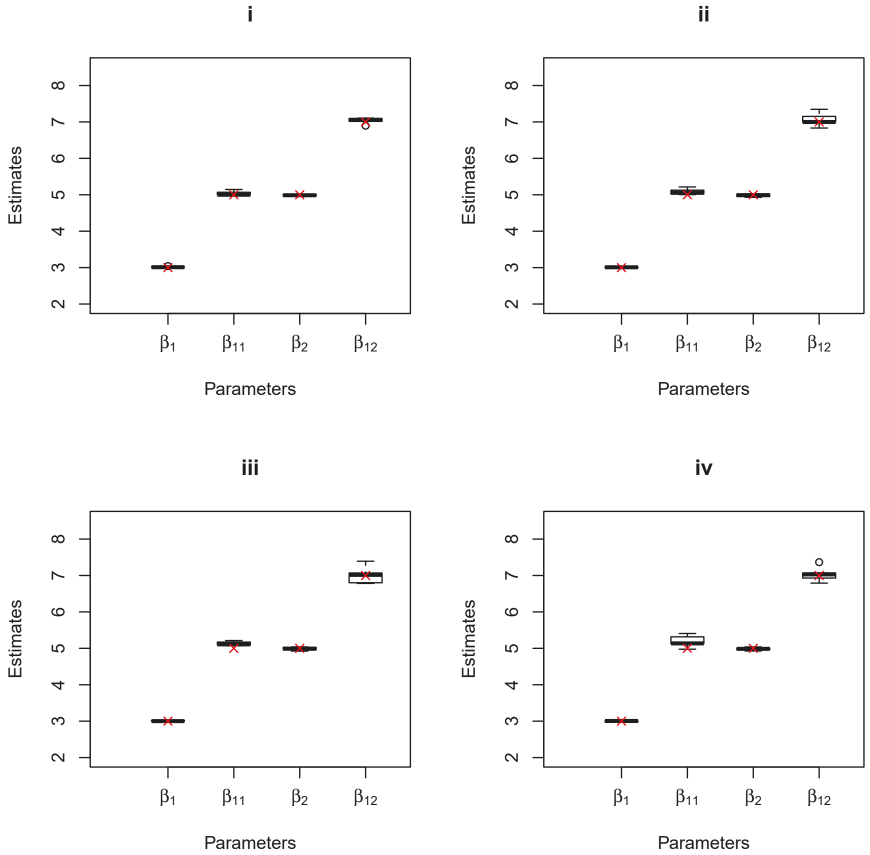
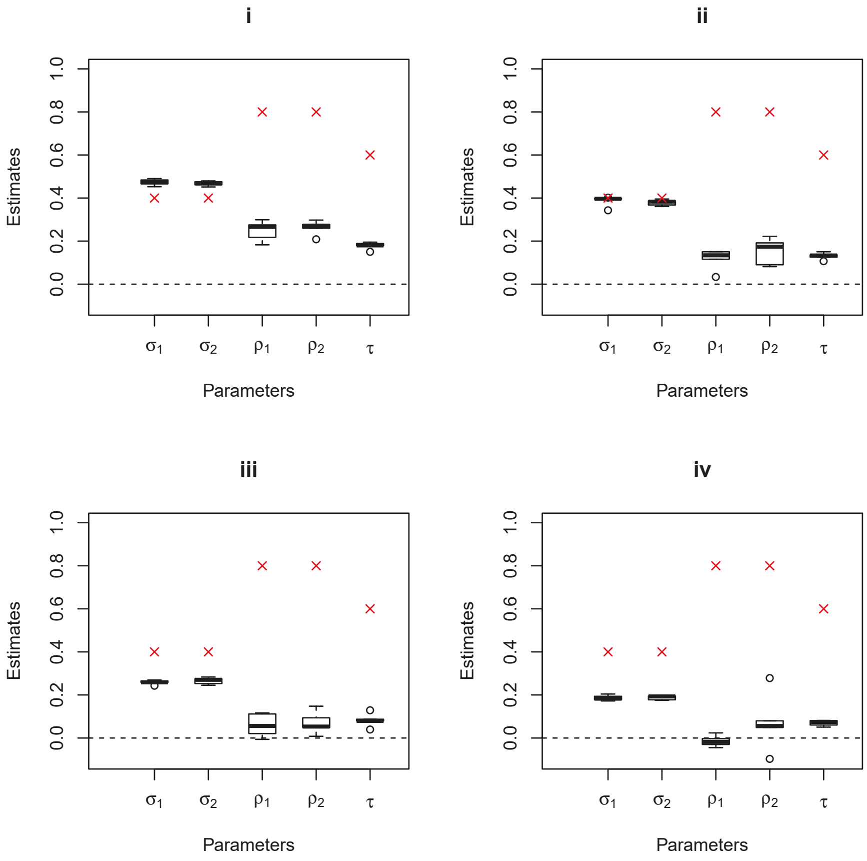


Figure 5.15: Aggregation Effect for Model 2:i. J=10, ii. J=20, iii. J=50, iv. J=100 ; n=10,000, with red crosses indicating the initial true parameter as set in the unaggregated data simulation; 5 replicates; Variance Components



5.5.4 Review of effect of aggregation

In this section we explored how aggregation can have a simplifying effect on the data. When analysing data from Option 1, we saw that Model 1 is an accurate fit. Extending the simulations to that with correlation present in the errors. We saw that aggregation can remove a simplistic AR(1) autoregressive error structure. The larger the aggregation block length, the more error structure reduces. Temporal aggregation does not affect the fixed effects coefficient estimates. From figures 5.11 and 5.14 we can see that all aggregation periods have consistent estimates. We now move on to the analysis of the applied data set on the ARTC coal train data.

5.6 Application

The primary focus of this chapter of the thesis is on the use of h-likelihood for mixed effects models. Although the data in the Hunter Valley Coal Train dataset has a long-memory dependence, as shown in chapter 2, our model is currently unable to consider more complex residual structure than AR(1) errors. Long-memory analysis predominantly is covered with ARFIMA errors. As such, we know that our analysis is not ideal, however we show that there is an improvement in fit from fitting an AR(1) to i.i.d error structure. Moreover, in line with Schabenberger and Pierce[54], who consider that it is more important to model the correlation structure in a reasonable way rather than attempting to model it perfectly. The temporal aggregation reduces the high order ARFIMA(p,d,q) to a more reasonable ARMA(p,q) correlation structure. It is possible to nitpick through a range of orders for the AR(p) and MA(q) components, however an AR(1) error structure is more than adequate.

In the following section we present results for Models 1-5 on the Hunter Valley Coal Train dataset for the two series of PM1 and PM2.5. All models are fit over a range of aggregation block lengths in the same manner as throughout this thesis, namely 5 minute aggregation and above. In Chapter 2, we presented a detailed overview of the data. Of particular note, as shown in figure 2.2, was the fact that the coal (loaded and unloaded) and freight trains took a number of minutes to pass the monitor, while the passenger trains, due to their much shorter length, passed the air monitor almost instantaneously.

Our previous analysis in chapters 2, 3 and 4 uncovered a tail effect for each passing train, as well as the possibility of an effect preceding a trains passing of the monitor. Thus we hypothesized that an interval block length from 1 to 10 minutes could capture the effect of a passing train. The literature of temporal aggregation has primarily covered the economics domain, where monthly, quarterly and annual aggregation periods are considered reasonable. However, there is no widely accepted aggregation periods for environmental applications, and furthermore, no temporal aggregation has been previously applied to train applications.

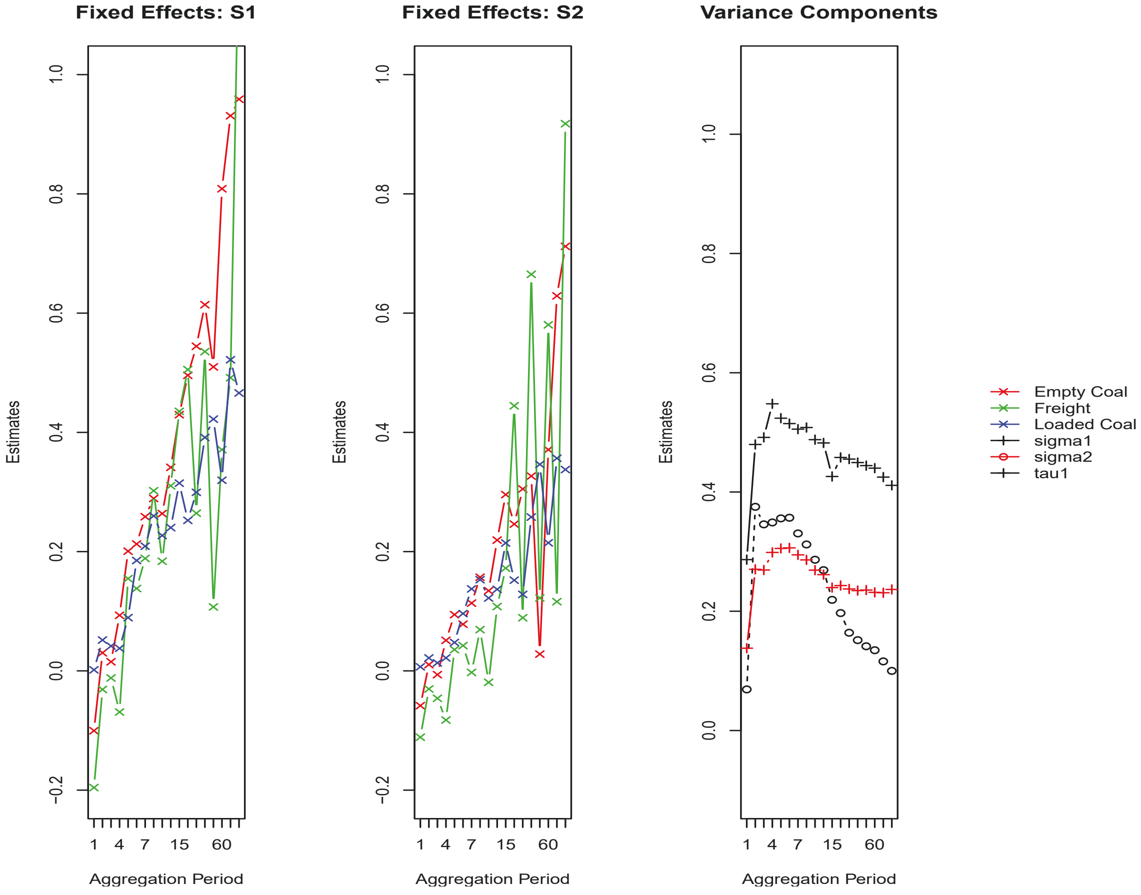
As such we consider a range of aggregation block lengths from 1 minute to 2 hours. We are further constrained by the limitation of memory storage of our covariance matrix and therefore only apply our models to aggregated data of size $m = 5,000$. For 1 minute aggregation this results in using the first nine days of the unaggregated data, however by 15 minute aggregation we are using the complete dataset.

5.6.1 Fit of model 1

We begin our applied analysis with the simplest model, Model 1, as expressed in equation (41). We do not consider autocorrelation in this model. Clearly this is not going to be an adequate modelling strategy, however we consider it to see how much better our other models do in comparison. Model 1 can be easily analysed through the nlme package in R with the lme function [49]. Nonetheless we have coded it up ourselves using the h-likelihood. Of note from Figure (5.16) is the third plot for the Variance Components. We can see that the estimates for σ_1 and σ_2 are much larger than in the figures for the other 4 models considered in this application. We conclude that this is due to the high levels of serial correlation present in the data. Our h-likelihood models are affected by serial correlation that has an order more complex than AR(1). As such, the high estimate for ρ_1 and ρ_2 pushes the estimates for σ_1 and σ_2 towards zero.

Further from figure 5.16 we can see that the variance components of σ_1 and σ_2 reduce as we increase the aggregation block period J. The first two plots show the results for our estimates of the fixed effects for each series S_1 and S_2. In this figure, and all others in the application, we note the fact that the fixed effect coefficient estimates for each passing train increase as we increase the aggregation period. This is consistent with our analyses in Chapter 3 and 4. As mentioned previously, this is occurring due to model misspecification.

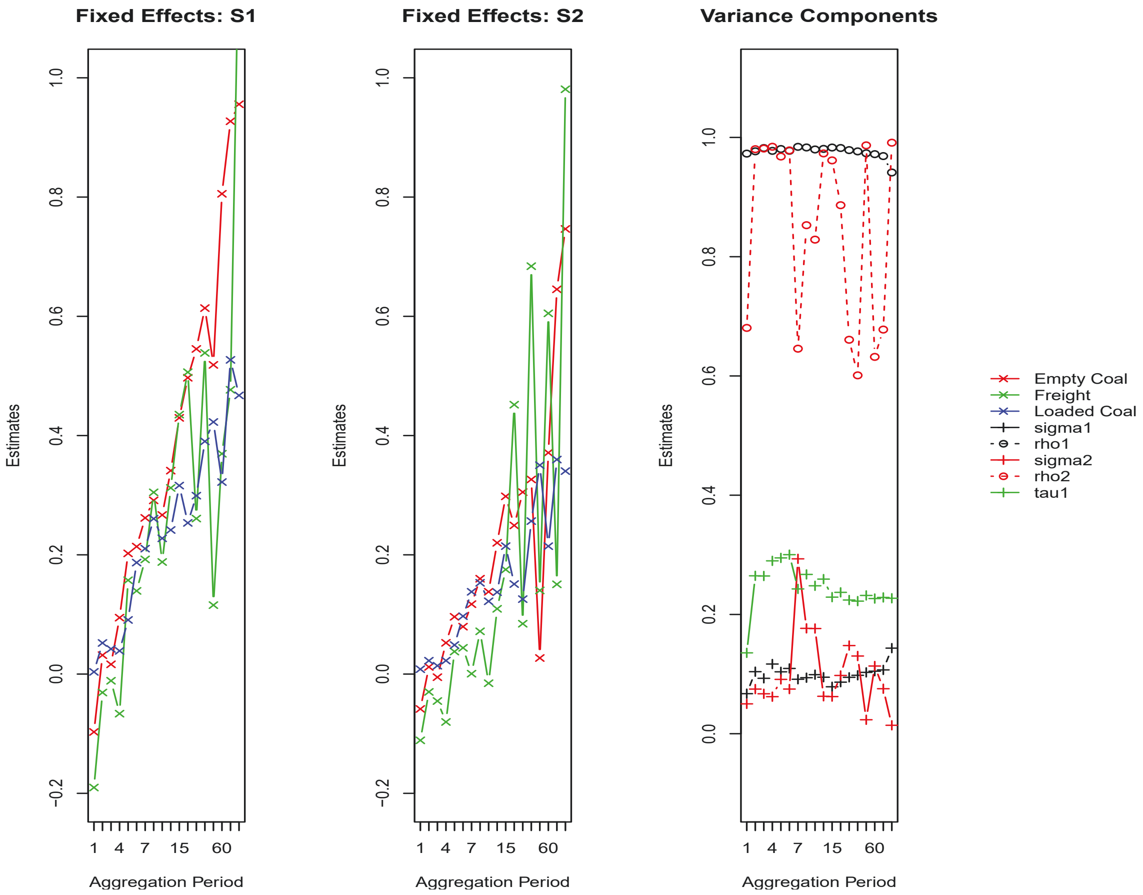
Figure 5.16: Application: Fit of Model 1



5.6.2 Fit of model 2

In Model 2 we have introduced an error structure to account for the serial correlation present in the data. We focus on an Autoregressive AR(1) correlation structure. From figure (5.17), and focusing on the third plot for the variance components, we can immediately see that the estimates for both ρ_1 and ρ_2 are extremely high, almost at 1. This indicates that we have not accounted for the correct level of autocorrelation in the data. The estimates for ρ_2 are lower than those for ρ_1 which we attribute to the data. The levels of serial correlation in the air particle observations for PM1 are lower than those in PM2.5. Our model has difficulty settling on an estimate for the fixed effects at each aggregation period as the aggregation period increases, which is consistent with our results in our previous chapters.

Figure 5.17: Application: Fit of Model 2

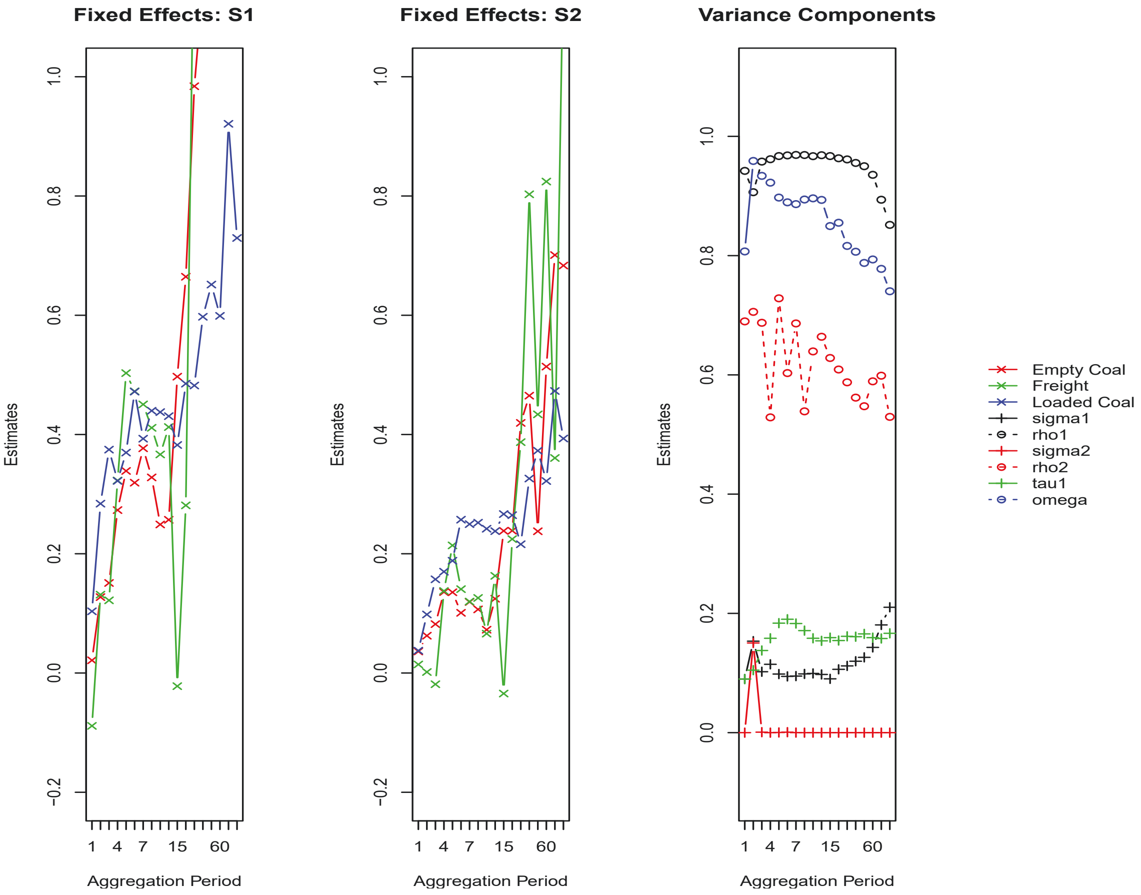


The third plot in figure 5.17 shows the variance components of model 2. The τ_1 estimate is shown by the green line. We can see that aggregation does not affect the level of cross-correlation. The only value of τ_1 that is below 0.2 is for 1 minute aggregation, where we have a very small subsample of the data.

5.6.3 Fit of model 3

In figure 5.18 we have the results for model three for the coal train data. Here we are considering an autoregressive AR(1) order on the cross correlation τ_1, which is denoted by ω. We find that there is such an effect present in the coal train data. This is shown by the blue line in the third plot of figure 5.18, with values of $\omega > 0.8$ consistently seen throughout the different aggregation periods. The fixed effects coefficient estimates for the passing trains are consistent with previous models 1 and 2.

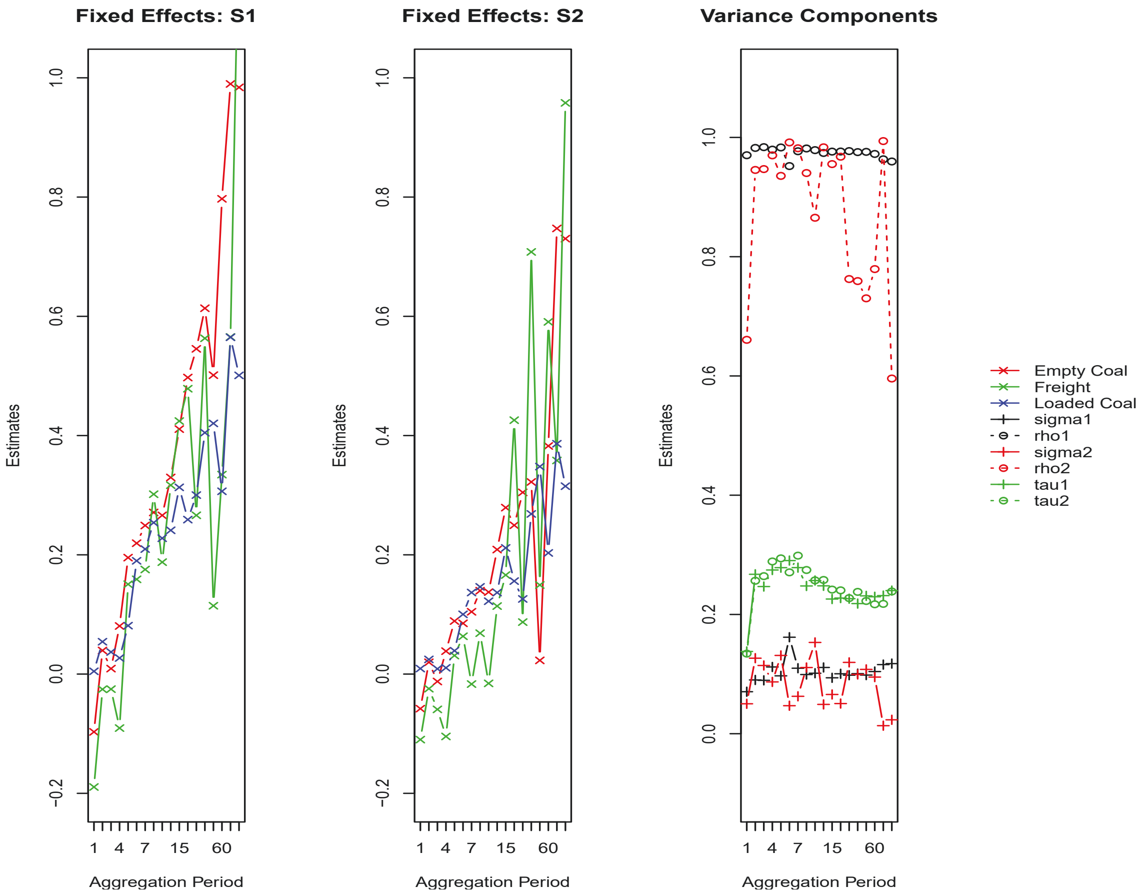
Figure 5.18: Application: Fit of Model 3



5.6.4 Fit of model 4

In model 4 we extend on model 2, where we have a seperate AR(1) structure on each series, by allowing for a different constant effect (τ_1, τ_2) for each random effect. This allows for a more of the variation in each series to be modelled. In figure 5.19, we present the results for the coal train dataset. In the third plot, we can see the estimated values for τ_1 and τ_2. They appear to be quite similar, suggesting that there is no advantage in implementing model 4 for this dataset. However, model 5 shows that this may not be the case. The fixed effects coefficient estimates for the passing trains for both PM1 and PM2.5 series, S_1 and S_2 show signs of deviating from the previous models. This suggests that the extra number of variance parameters are capturing some of the model misspecification from the simpler models.

Figure 5.19: Application: Fit of Model 4



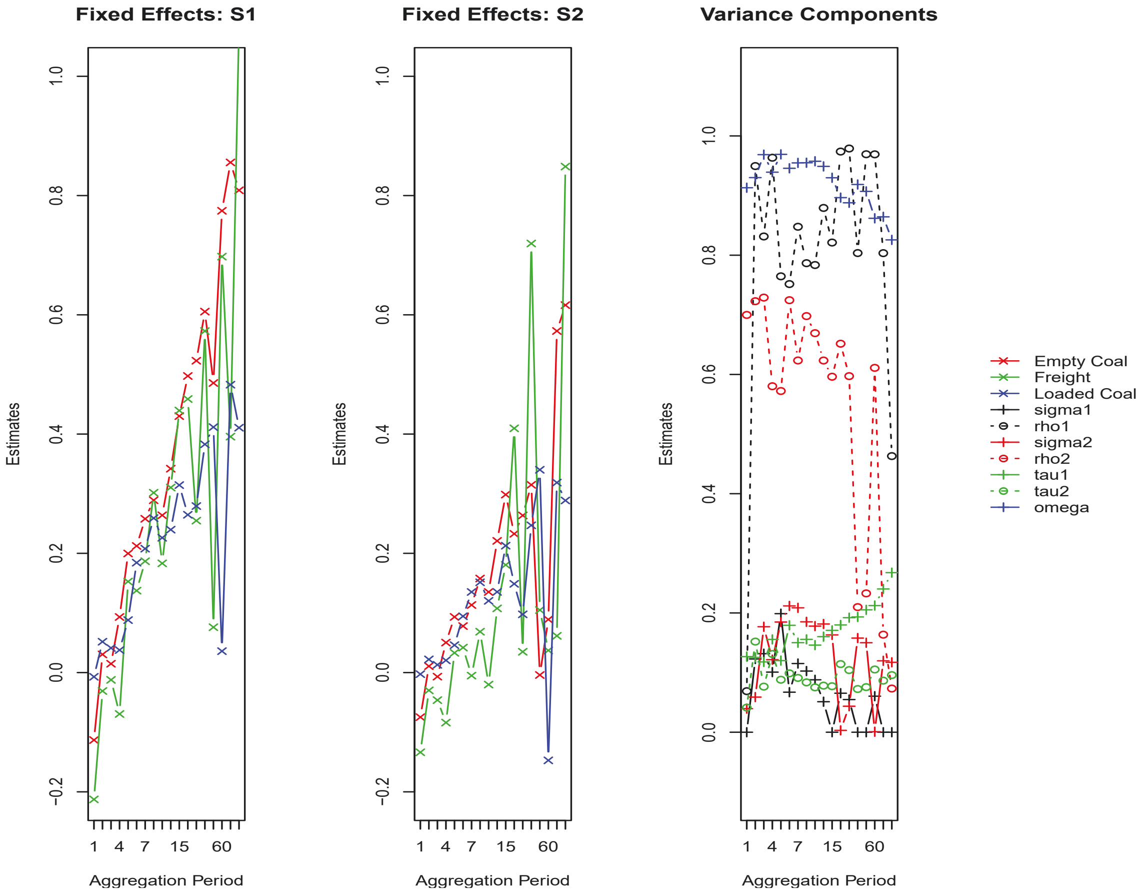
5.6.5 Fit of model 5

For model 5, we present the results for the coal train dataset in figure 5.20. This is the final model in this chapter. We consider a number of variance components. We have an AR(1) correlation and seperate variance for each series S_k. We then incorporate an AR(1) structure with ω on the cross correlation τ_1 and τ_2, where we are able to determine how much each series contributes to the cross-correlation due to the seperate τ parameters. In model 4 from figure 5.19, our model estimated that the $\tau_1 = \tau_2$, however for model 5 we are able to see that there actually is some deviation between these two series. From the third plot in figure 5.20, we can see that the estimate $\tau_1 > \tau_2$, we suggests that the series S_1 contributes more to the cross-correlation than the series S_2 for PM2.5.

An interesting outcome is that the increased number of variance parameters in model 5, has reduced the impact of using an AR(1) correlation structure on each series, which we know is not an adequate process for the data. This finding is shown by the ρ_1 and ρ_2 values being lower for this model than the previous options. Nonetheless, the σ_1 and σ_2 values are still being severely limited by the use of the simplistic AR(1) model for this dataset.

Our fixed effects coefficient estimates for the passing trains in both series S_1 and S_2 are not consistent with our previous models. By comparing the first two plots of figures 5.19 and 5.20, we can see that there is some variation in our estimates.

Figure 5.20: Application: Fit of Model 5



5.6.6 Review of model fits

Having fit models 1 through 5 on the Hunter Valley Coal Train dataset to the variables PM1 and PM2.5, there are a number of key observations. Firstly, our analysis is constrained by the limitations of our h-likelihood process for autoregressive AR(1) series. We have shown in chapters 2, 3 and 4 that our Hunter Valley Coal Train dataset has a long memory dependence. Therefore these results are limited in their accuracy. However, we have shown that the models 3 and 5 have been able to capture some of the dependence in the series, thereby reducing the need for a complex correlation structure. The extension of these models to ARMA or ARFIMA errors would be ideal. Nonetheless, our findings suggest that these models would be of interest for datasets such as the Hunter Valley Coal Train dataset, where there is a complex time series nature.

The fixed effects coefficients are affected by each of the models. As we capture more of the variance components, we are able to reduce the impact of a misspecified error structure that is currently used. As we increase the aggregation period beyond ten minutes, the coefficient estimates become extremely variable. This is consistent with our work in previous chapters, and suggests that these are not viable aggregation periods due to the loss in information.

5.7 Discussion

We extend the current h-likelihood theory to include correlation in the data. Namely, we consider an Autoregressive AR(1) structure on the errors, in Models 2, 3, 4 and 5, and an Autoregressive AR(1) structure on the random effects, in Models 3 and 5. Further we consider that the random effects has a normal distribution, but that each series has it's own constant multiplier of this covariance. This allows for more of the variation between each series to be considered in the models.

The use of the hierarchical log likelihood allows for inference of both the fixed and random effects. Our simulation study shows that these models fit accurately and consistently. Through our simulation studies we can see that for simple correlation structures, temporal aggregation simplifies the necessary models, while also accurately fitting the data.

Our applied data analysis is severely limited by the complexity of the dataset. The presence of long-memory correlation in the data results in temporal aggregation not removing all error structure. It does however simplify the short term memory which is helpful in simplifying the data to the workable AR(1) correlation structure. Furthermore, the knowledge that the simple model of one regressor for each train type is not capturing the total effect of each passing train, constrains our models to model misspecification. Chapter 4 of this thesis, addresses this concern, and thereby allows us to concentrate on the hierarchical likelihood and its development and fit here.

In situations where a bivariate data analysis is affected by serial correlation, especially of a simpler short term memory such as AR(1) or a reasonable ARMA(p,q), we consider the h-likelihood with AR(1) structure a productive model to fit to data, particularly in the event that aggregation has been utilised to reduce the data volume. Hereby we are able to reduce some of the information loss as a result of this data transformation, and the analysis of a bivariate series can recover some details in the data that have been obscured by the use of temporal aggregation.

A primary reason that we consider temporal aggregation in this thesis is to not only reduce the model complexity but also to reduce the size of extremely large datasets. As is shown in section

5.6, the application of the coal train data, we are limited by the size of the data in fitting our models. This is due to the computational needs for storing and inverting the large and dense, yet symmetric, covariance matrix in each model. At present on our personal laptop computer, we are limited to a dataset of size $n = 5,000$, for estimation of each model. Although beyond the scope of this thesis, we believe that further investigation into dense matrix estimation via topics such as regularization, especially banding and tapering, would have a drastic impact on the size of possible datasets to fit using these models.

6 Divide And Recombine in a Time Series Setting

6.1 Introduction

The presence of long memory dependence in a dataset can severely limit the modelling strategies available for a statistical analysis. Couple this complex data structure with a large number of observations, and an analysis can quickly become impossible without, or even with, access to a supercomputer. One of the goals of this chapter is to reduce the need for a supercomputer by extending current statistical methods. Divide and Recombine, herein referred to as D&R, is a process that can overcome such difficulties. D&R is a process whereby a dataset is divided into a number of subsets, each subset is analysed individually, and then these results are recombined into a final output.

Throughout this thesis, we have been analysing a number of models on the coal train dataset as described in chapter 2. So far we have been limited to analysing aggregated data, as our model strategy that accounts for the long memory dependence present in this dataset, is limited to datasets of a maximum size of 12,000 observations. In unaggregated form, the dataset has well over 600,000 observations. The implementation of D&R, allows us to analyse the unaggregated dataset for the first time in this thesis. The use of a linear regression with ARFIMA errors, overcomes the issues of a long memory time series. However, the requirement to calculate the inverse of the dense covariance matrix, requires a large amount of computer memory. Not only does this limit the size of the dataset that is possible for analysis, but it can also take a considerable amount of time. Applying the D&R process to our dataset results in a significant speed up in computation. Some analyses times are reduced from 40 minutes to 5 minutes.

The research of the D&R process is in it's infancy. We aim to further this knowledge with particular reference to time series analysis. We explore aspects such as whether the division choice impacts the time series structure of a dataset, as well as whether the subsets retain the same structure as the full, undivided dataset. An inherent issue with the D&R process is that as a result of the division, one must estimate a model for each subset, rather than just one for the full dataset. This presents complications in certain situations, such as in time series, where the estimation of a number of parameters may be difficult. In this situation, the difficulty lies in the correct estimation of the time series parameters. Nonetheless, this is a trade-off one must endure, particularly when the analysis of the full dataset is not possible, or time wise infeasible.

We begin this chapter with a review of the D&R literature, which we then follow with an outline of the D&R process. Our analyses begin with a simulation study. Here, we consider both an independent and identically distributed and a long memory process. Through a comparison of a linear regression with iid and ARFIMA errors respectively, we show that the D&R results in consistent results for both unaggregated and aggregated datasets. The number of subsets used for the D&R models does not impact the analysis, nor does it change the time series structure of the simulated datasets.

Having shown through a simulation study (although quite a simplistic one), that the D&R process can be effectively used for both iid and long memory data, we turn to our coal train application. The analysis of the coal train dataset is split into two sections. As in chapter 4, we consider a 'seperate' model, where we have seperate train and tail covariates for each train type in our dataset, and a 'combined' model, where each train and tail variable is combined into one covariate. In our 'seperate' model analysis, we conduct a number of analyses on the unaggregated and aggregated data. The use of D&R requires the selection of a method for the division. This can be a conditioning-variable division, which is data dependent, or replicate division, where the data is divided into consecutive non-overlapping subsets. For both of these options, we compare the model results and the estimates of our time series parameters. As in our simulations, we show that the division method does not impact our analysis. However, the analysis of our unaggregated and aggregated datasets yields conflicting results. As we increase the aggregation period, the coefficient estimates for each passing train covariate increases. This effect is also seen in the tails, however it is not as substantive. We suggest that this is a result of the dataset, rather than the D&R process. For our analysis of the 'combined' model, we limit the D&R process to conditioning-variable division. The remainder of this analysis is as for the 'seperate' model. Here the results are less variable, however the increasing effect still remains.

We complete this chapter with a comparison of the D&R results with those of the full and undivided dataset that we produced in chapter 4. This comparison constitutes the coefficient estimates for the models, and the time taken to analyse each. Here we show that the D&R process can be immensely beneficial in a time series analysis with large data. The significant improvements in computational speed can allow the statistician to focus on different effects in the data, such as the aggregation effect we find in the coal data, that would otherwise be impossible or at the very least, too time consuming.

6.1.1 Literature review

Divide and Recombine is an extension of the MapReduce process as developed at Google by Dean and Ghemawat [12]. The introduction of MapReduce saw the distribution and analysis of data on a cluster of computers. This is a method to circumvent situations where data cannot be stored on one local machine. Wickham [64] extended upon this work in a statistical setting and termed it as the split-apply-combine process. This was further considered by Guha[21], Chen[10], Hafen[22] and Tung[61], under the terminology of Divide and Recombine.

This same process was implemented by Fan [14], although not termed as Divide and Recombine or Split-Apply-Combine. Here a large dataset with 2 million observations was divided into 1000 blocks, with 2000 observations in each block. A linear regression was fit to each subset and then recombined using a weighted average. Her results were consistent with the full model. Li [40] applied the same process to an application for internet traffic data. Both these papers where able to reduce computational times while achieving efficient estimates in comparison with the full model. Xu[69] applied this approach to a quantile regression, achieving similar results, while implementing a simple average for the recombination of each subsets results. Li[38] also implemented a D&R approach to a large dataset, with an application to forecasting drug users in Hong Kong.

In a time series setting, Pierrot [48] split a dataset into 48 subsets and applied a GAM model to each. Wood [67] commented on this method, stating that this approach suffers from three disadvantages, namely that,

1. the correlation between subsets is not exploited,
2. there are interpretation difficulties, as information between subsets is lost, and

3. a reduced sample size can increase the chance of overfitting.

Wood then analysed the same data, by introducing a method to account for the large data size. This is however not applicable to a long memory setting, as they used a banding process on the covariance matrix, which would eliminate significant correlations in a long memory dataset. These issues mentioned by Wood, are still to be accounted for in a D&R process for time series data. Some work has been developed in this field, notably with Battey[4], who outlines that the split subsamples should be independent. Unfortunately this is not always possible, in such cases Jordan [31] argues that for dependent data, resampling should be done in ways that respects the dependence.

A related area to D&R in a time series setting, is time series segmentation. Although the main focus is on determining structural breaks in a time series dataset, certain qualities are transferrable to the D&R domain. In particular, this area of research has considered factors such as the impact of long memory on a dataset. By creating structural breaks or segments in a long memory dataset, Perron [47] and Chatzikonstantia[9] showed that a long memory dataset can actually be a collection of short memory subsets. They also found that for a truly long memory process, the subdivision of the data will not impact the time series structure. These findings are directly applicable to our D&R process for long memory time series. We are able to check the structure of our time series through its subdivision, and more importantly, we can retain the long memory structure after the division into subsets.

6.2 Divide and Recombine process

We present the method of Divide and Recombine for a linear regression with ARFIMA errors which can be used for time series data that presents long memory dependence. This process is also applicable to other models, as shown in, amongst others, [14], [21] and [69]. For a further review of this long memory model please refer back to chapter 3. For the model,

$$Y = X\beta + \epsilon, \tag{60}$$

Y is a vector $(n \times 1)$ of dependent observations, X is a design matrix of size $((p+1) \times n)$, with p covariates, and ϵ is a vector of $(n \times 1)$ errors with long memory dependence, as defined by an ARFIMA(p,d,q) model with errors,

$$\Phi(B)(1-B)^d \epsilon_n = \Theta(B)\eta_n, \tag{61}$$

where $-1/2 < d < 1/2$ and $\eta_N \sim (0, \sigma_\eta^2)$. B is the backshift operator $B\eta_n = \eta_{n-1}$, the AutoRegressive (AR) operator is $\Phi(B) = 1 + \phi_1 B^1 + \ldots + \phi_p B^p$, and the Moving Average (MA) operator is $\Theta(B) = 1 + \theta_1 B^1 + \ldots + \theta_q B^q$. The fractional difference operator is $(1-B)^d = \sum_{k=0}^{\infty} \tau_k B^k$, where $\tau_k = \Gamma(k-d)/\Gamma(k+1)\Gamma(-d)$. $\Gamma(\cdot)$ denotes the Gamma function. In the event that $0 < d < 0.5$, the errors are a long memory process.

In certain situations such as large or complex data, we may be limited by the size of N, to conduct an analysis of the data. We can thereby use D&R to split the data into a number of smaller datasets. In the event of time series data, particularly with the presence of long memory dependence, we are constrained by memory limitations for the modelling of the complex data. We now cover the methodology for division of the data using D&R. We also consider the impact of D&R on time series data.

6.2.1 Division step

The divide and recombine process begins with the division of the dataset. There are two methods for this division. Conditioning-variable division and replicate division [22]. Under conditioning-variable division, we divide our data based on the subject matter. In our application, we have data recorded for a number of days. Thus a possible conditioning-variable division is to divide the dataset into each day. In certain cases, this division will result in subsets that remain too large for an analysis. In this case, one must consider another division. In our application, we encounter this issue, and divide each day further into two subsets for each day. The other option for the divide part of divide and recombine, is to partition the dataset into sequential replicates without replacement. An example of this, is to divide the dataset into a number of subsets with the same length.

For our linear regression model in equation (60), we illustrate both methods. The division is applied to the data, and as such the time series nature of the model is not encountered until we analyse each subset.

6.2.2 Replicate division

We set the vector of observations from equation (60), as the original undivided vector $\{Y_n^O\}_{n=1}^N$. We divide the N observations into K equal and non-overlapping subsets of length L. Thus $K = N/L$. Here L is an integer. The divided vector $\{Y^O\}_{n=1}^N$ becomes,

$$\{Y_k\}_{k=1}^K = \begin{cases} Y_1 = (y_1, \dots, y_L) \\ Y_2 = (y_{L+1}, \dots, y_{2L}) \\ \vdots \\ Y_K = (y_{(K-1)L+1}, \dots, y_{KL}) \end{cases}$$

Under replicate division we may encounter the possibility that N/L is not an integer. In this case we must remove a sufficient number of the final observations such that N/L is an integer.

6.2.3 Conditioning-variable division

We set the vector of observations from equation (60), as the original undivided vector $\{Y_n^O\}_{n=1}^N$. We then divide the N observations into K non-overlapping subsets. Instead of L being an integer, we now set it to be a vector, $L_k = (L_1, \dots, L_K)$, where each L_k is the length of each subset k. The selection of each L_k is dependent on the conditioning-variable of the data. The divided vector $\{Y_n^O\}_{n=1}^N$ becomes,

$$\{Y_k\}_{k=1}^K = \begin{cases} Y_1 = (y_1, \dots, y_{L_1}) \\ Y_2 = (y_{L_1+1}, \dots, y_{L_2}) \\ \vdots \\ Y_K = (y_{(K-1)L+1}, \dots, y_{L_K}) \end{cases}$$

Under both conditioning-variable and replicate division, we must apply the same process to any other variables in the analysis as we have to the dependent observations in vector $\{Y\}_{n=1}^{N}$. Thus in the model (60) we must also divide the X design matrix, resulting in $\{X_k\}_{k=1}^{K}$.

6.2.4 Analysis step

Having divided our data, we can now apply our statistical analysis to each subset. Thus the linear regression with ARFIMA errors in model (60) is transformed into,

$$Y_k = X_k \beta + \epsilon_k \tag{62}$$

for K subsets, with a long memory error structure that is defined as,

$$\Phi(B)(1-B)^d \epsilon_k = \Theta(B)\eta_k, \tag{63}$$

The resulting analysis for each subset leaves us with K estimates for the coefficient $\hat{\beta}_k$. The ARFIMA(p,d,q) order is estimated for each linear regression. This increased number of parameter estimates to be made is a point of concern for D&R, however we return to this at the end of this section.

6.2.5 Recombine step

Having divided the data and then performed our linear regression on each subset, we now must recombine the estimates for each $\hat{\beta}_k$. To account for differences in each subset, we consider weighting the results of each subset, through an inverse weighting scheme.

$$\hat{\beta}_{D\&R} = \frac{\sum_{k=1}^{K} \hat{\beta}_k W_k}{\sum_{k=1}^{K} W_k}. \tag{64}$$

Here the weights are $W_k = 1/\sqrt{Var(\hat{\beta}_k)}$. In the event that we wish to recombine using a simple averaging process, we set $W = 1$. To determine the variability of each estimate, we also transform the standard errors for each coefficient estimate. Thus the recombined standard error for the coefficient estimate $\hat{\beta}_k$ is,

$$SE(\hat{\beta}_{D\&R}) = \sqrt{\frac{\sum_{k=1}^{K} W_k^2 Var(\hat{\beta}_k)}{(\sum_{k=1}^{K} W_k)^2}} \tag{65}$$

Using this D&R process, we are able to reduce the size of the dataset for each analysis. This is of particular usefulness for complex and/or large datasets. We now consider the impact of D&R on complex data such as a long memory time series.

6.2.6 Divide and Recombine for time series data

For time series data, whether it has an AutoRegressive, Moving Average, ARMA, ARIMA, ARFIMA or any other structure, we can apply Divide and Recombine to the data. D&R can affect the data and its subsequent analysis in a number of ways. In particular, dividing the data may result in a loss of information between subsets. This is not a concern for iid data, however in a time series, especially for long memory dependence, we can lose the correlation between observations, across subsets. One method we employ to explore this effect is to consider differing subset sizes.

After applying our modelling strategy to each subset, we can encounter some further issues with D&R. One of these is, that by estimating an error structure (as in equation 63) for K subsets instead of just once to the full data (as in equation 61), we are increasing the chance that we have incorrectly estimated the error structure. This is not the easiest process and can thus lead to inaccuracies in the analysis. However, this is a trade-off that one encounters in the event that the full data is either too large or complex to analyse at all, or too cumbersome for the computer. We consider this situation in our application section.

Another concern is that the division of a dataset, can alter its structure. An example of this is in [9] and [47], where it was shown that a long memory financial series can have a number of structural breaks which render each subsection as a short memory process, whereby the full dataset would be incorrectly analysed under long memory dependence. To counter this possibility, it is wise to compare the structures of both the full and divided datasets. We also consider this in our application.

We now turn to a simulation study to explore the effect of Divide and Recombine on a time series dataset.

6.3 Simulations

Given the difficulty in analysing large and complex data, we implement a Divide and Recombine process that enables an analysis of the coal data in our application. We now conduct a simulation study that compares regressions for unaggregated and aggregated data with the full dataset and the D&R dataset. The aim of these simulations is to determine if the D&R process can successfully replicate the outcomes of the full dataset. We also consider the impact of D&R on the estimation of an ARFIMA error structure.

We begin our simulation with a description of the data generating model. Then we simulate and analyse a simple linear regression with gaussian iid error structure. Here the results are expected to be constant across the dataset options, as is shown in previous literature such as in [14] and [21]. Following this we turn to a simulated dataset with long memory dependence. Again we compare the results of the full and D&R datasets. For both of these simulations we consider two alternative division options for the D&R. Namely, we can divide the dataset into 2 or 5 subsets. The outcomes show that for a simple simulation, the D&R process retains constant coefficient estimates as does the full dataset.

We conclude our simulation study with a comparison of the estimated ARFIMA(p,d,q) parameters for each analysis. This shows that the D&R does not impact the long term memory. As we now have 5 more subsets of data, there is a larger chance that we incorrectly estimate the correct

ARFIMA(p,d,q) error structure. Fortunately, in this simulation, this does not impact the coefficient estimates, however it may be a cause of concern for more complicated applied analyses.

6.3.1 Data generating model

We generate our data from the following model:

$$y_n = \beta_0 + \beta_1 x_n + \beta_2 r_n + \epsilon_n, \tag{66}$$

where ϵ_n is an error term discussed below. Each simulation has a length of $n = 1, \ldots, 5000$. We are limiting this length for computational reasons, as regression with ARFIMA errors takes a considerable amount of time for data of length greater than 5000. We consider two cases for the errors. To simulate a long memory process we set the errors, $\epsilon_n \sim ARFIMA(3, 0.4, 3)$, with $\eta_n \sim N(0, \sigma_\eta^2)$ and $\sigma_\eta^2 = 0.4$. We also consider the case where the error $\epsilon_n \sim N(0, \sigma_\epsilon^2)$ are independent and identically distributed. In this case $\sigma_\epsilon^2 = 1$.

The passing train indicator variable x_n,has a length of 10 observations. The tail variable, r_n has a length of 10 observations, and begins immediately after a passing train. It is also set to be an indicator variable. Having created an error structure, and the passing trains and tails, we set our dependent variable y_n as in our data generating equation as noted in (66) with $\beta_0 = 3$ and $\beta_1 = \beta_2 = 0.10$.

The aggregation process is the same as outlined in chapter 3 and is continued throughout this thesis. We provide a brief overview here. For more detail please refer back to Chapter 3, Section 2.

For a time series of length $n = 1, \ldots, N$, we split the data into non-overlapping subsets of size J. We then take the mean for each subset. The aggregated data now has length $M = N/J$. In the event that M is not an integer, we remove a sufficient number of the final observations such that N/J is an integer. For the covariates, we apply the same method and take a mean. For an indicator variable as in these simulations and the data at large, the mean can be interpreted as a proportion of time a train (or tail) has spent passing in each block. More precisely, the variables, $\{y_n\}_{n=1}^N$ are transformed into $\{s_m\}_{m=1}^M$ as shown below,

$$s_m = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} y_n \tag{67}$$

The independent variables are transformed, for the train variable $\{x_n\}_{n=1}^N$, for the tail variable $\{r_n\}_{n=1}^N$,in the same manner as below,

$$z_m = \frac{1}{J} \sum_{n=(m-1)J+1}^{mJ} x_n. \tag{68}$$

For each simulation, we have analysed 20 replicates of the analyses.

6.3.2 Models in simulations

Having presented our data generating model, we now introduce the two models we use to analyse the simulated data. We consider the two models on both unaggregated and aggregated datasets. We are interested in the aggregated datasets, as in our application we are unable to analyse the unaggregated data without D&R. The goal of this simulation study is to compare the accuracy of the coefficient estimates of the D&R model against the full data model.

Thus our first model is a linear regression on the full data,

$$y_n = \beta_0 + \beta_1 x_n + \beta_2 r_n + \epsilon_n, \tag{69}$$

where y_n is a vector of $(N \times 1)$ dependent observations, x_n and r_n are both vectors of $(N \times 1)$ independent observations, and ϵ_n is a vector of $(N \times 1)$ errors. In section 6.3.3 we analyse the simulated data with iid errors, so we consider the model (69) with with $\epsilon_n \sim (0, \sigma_\epsilon^2)$. Then in section 6.3.4, we simulate a long memory error structure, as in our application, and thus we use a linear regression with ARFIMA(p,d,q) errors to analyse the dataset. Here, the errors ϵ_n are as presented in equation (61).

We then apply the D&R process to the simulated data. Here we have the model,

$$y_k = \beta_0 + \beta_1 x_k + \beta_2 r_k + \gamma_k, \tag{70}$$

where, as outlined in section 6.2, $k = 1, \ldots, K$ subsets and L is the length of each subset. y_k is a matrix of $(L_k \times K)$ observations, and x_k and r_k are the matrices of the independent observations with size $(L_k \times K)$. γ_k is a matrix of $(L_k \times 1)$ errors with the same error structure as in the equations 69 or 70.

These two models are for our analyses on unaggregated data. We are also interested in how D&R performs for aggregated data. Using the aggregation procedure as outlined in the equations (67) and (68), we transform the unaggregated data to aggregated data. We can then implement the full model on the aggregated data as shown here,

$$s_m = \beta_0 + \beta_1 z_m + \beta_2 u_m + \omega_m. \tag{71}$$

We also apply the D&R process to the aggregated data and model this with,

$$s_k = \beta_0 + \beta_1 z_k + \beta_2 u_k + \xi_k. \tag{72}$$

These aggregated dataset are reduced in size by an order of J, where J is the size of the aggregating period. We now present our simulation results for the iid and ARFIMA error structures.

6.3.3 Linear regression with iid errors

In our first simulation, we generate our simulated data with iid errors. This simple case allows us to compare the four models, as outlined in section 6.3.2, without concern for complex data features. We are therefore able to determine that D&R is an effective method, as we achieve consistent estimates for all 4 models. We present the results of these simulations in table 6.1.

Table 6.1: Simulation results for iid error structure. Values shown are the means of 20 replicates. The simulated data has length $N = 5000$, and for aggregated data, we aggregated every $J = 10$ observations. The number of subsets, K, for the D&R models is $K = 2$ and $K = 5$.

	Full Model		Div And Rec K=2		Div And Rec K=5	
	Estimate	Std.Error	Estimate	Std.Error	Estimate	Std.Error
	Unaggregated Data					
	$y_n = \beta_0 + \beta_1 x_n + \beta_2 r_n + \epsilon_n$		$y_k = \beta_0 + \beta_1 x_k + \beta_2 r_k + \epsilon_k$		$y_k = \beta_0 + \beta_1 x_k + \beta_2 r_k + \epsilon_k$	
$\hat{\beta}_0$	2.998	0.016	2.998	0.016	2.998	0.016
$\hat{\beta}_1$	0.106	0.045	0.106	0.045	0.108	0.045
$\hat{\beta}_2$	0.109	0.045	0.109	0.045	0.111	0.045
	Aggregated Data					
	$s_m = \beta_0 + \beta_1 z_m + \beta_2 u_m + \omega_m$		$s_k = \beta_0 + \beta_1 z_k + \beta_2 u_k + \omega_k$		$s_k = \beta_0 + \beta_1 z_k + \beta_2 u_k + \omega_k$	
$\hat{\beta}_0$	2.997	0.017	2.997	0.017	2.997	0.016
$\hat{\beta}_1$	0.110	0.057	0.109	0.057	0.113	0.057
$\hat{\beta}_2$	0.111	0.057	0.112	0.057	0.112	0.057

The results in table 6.1 clearly show that all four models can be used interchangeably for our simulated data, as all the parameters are essentially equal. We now turn to a rather more complicated simulation, where we mimic the coal train data of our application, by inducing long memory dependence upon our error structure.

6.3.4 Linear regression with long memory errors

In our second simulation study we consider the effect of D&R on a long memory time series. Using our data generating model in (69) we implement an ARFIMA(3,0.4,3) error structure. We consider the same four models as in the previous section 6.3.3. Through our comparison of these model results in table 6.2, we notice that the full model and the D&R models are the same as in the iid case. There is some variation in the coefficient estimates, but it is minimal. By dividing the data into a number of subsets using the D&R process, we then have to estimate the same data a number of times. This can lead to some variation in the results, due to the difficulties in estimation, particularly in the time series domain. We consider the estimation of the ARFIMA(p,d,q) parameters in figure 6.1.

Table 6.2: Simulation results for long memory error structure. Values shown are the means of 20 replicates. The simulated data has length $N = 5000$, and for aggregated data, we aggregated every $J = 10$ observations. The number of subsets, K, for the D&R models is $K = 2$ and $K = 5$.

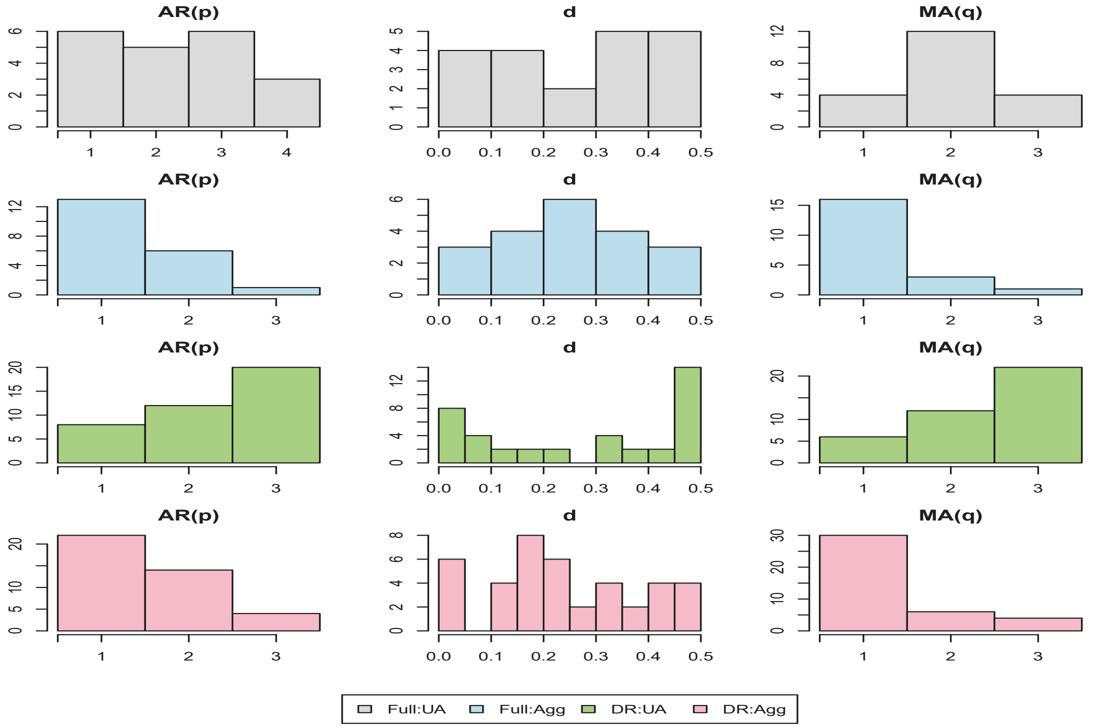
	Full Model		Div And Rec K=2		Div And Rec K=5	
	Unaggregated					
	$y_n = \beta_0 + \beta_1 x_n + \beta_2 r_n + \epsilon_n$		$y_k = \beta_0 + \beta_1 x_k + \beta_2 r_k + \epsilon_k$		$y_k = \beta_0 + \beta_1 x_k + \beta_2 r_k + \epsilon_k$	
	Estimate	Std.Error	Estimate	Std.Error	Estimate	Std.Error
$\hat{\beta}_0$	2.952	0.336	2.986	0.402	3.000	0.115
$\hat{\beta}_1$	0.096	0.021	0.093	0.020	0.092	0.021
$\hat{\beta}_2$	0.094	0.021	0.095	0.020	0.093	0.021
	Aggregated					
	$s_m = \beta_0 + \beta_1 z_m + \beta_2 u_m + \omega_m$		$s_k = \beta_0 + \beta_1 z_k + \beta_2 u_k + \omega_k$		$s_k = \beta_0 + \beta_1 z_k + \beta_2 u_k + \omega_k$	
	Estimate	Std.Error	Estimate	Std.Error	Estimate	Std.Error
$\hat{\beta}_0$	2.981	0.282	2.991	0.176	2.961	0.131
$\hat{\beta}_1$	0.095	0.041	0.094	0.041	0.098	0.041
$\hat{\beta}_2$	0.082	0.041	0.082	0.041	0.086	0.041

In table 6.2 we can see that both the unaggregated and aggregated datasets provide consistent coefficient estimates. A comparison of the full and D&R model results shows some minimal variation in the estimates. However this is too small to make any meaningful inferences.

For each of the four models above, we have also show the estimates for the ARFIMA(p,d,q) parameters in the following figure 6.1. Of particular interest are the first and third rows (in light grey and light green respectively), which show that there is an amount of variation in the estimates for the parameters due to D&R. Upon inspection of the fractoral differencing parameter d, we can see that the spread of its estimates is similar, however there is a tendency to the extremes for the D&R case. This is clearly due to the data generating model not simulating the data to be $d = 0.4$ as constructed, but it is possible that the estimation using the ARFIMA model can result in some variation. This is a difficulty faced not ony by D&R but also in the time series domain. The estimation of a time series model is complicated by the parameter selection, whether it is an AR(p), MA(q), ARMA(p,q) or ARFIMA(p,d,q) model.

The second and fourth rows compare the full and D&R models for aggregated data. An interesting note here is that the aggregation does not affect the long memory structure, as shown by the consistency of the fractoral differencing parameter d across all the models. However the short term memory, indicated by the AR(p) and MA(q) parameters is clearly reduced by aggregation. Both of these results are consistent with time series theory.

Figure 6.1: Comparison of estimated ARFIMA(p,d,q) parameters for simulations with K=2. 20 replicates for each model. The third and fourth rows of this figure have 40 estimates as the Divide and Recombine has split each replicate into two periods.



6.3.5 Review of simulations

Our simulation study has considered the effects of Divide and Recombine on a couple of simplistic datasets. They have shown that D&R should not impact the estimation of linear regressions, whether they have a iid or long memory error structure. Our long memory simulation study has presented some of the issues with D&R, namely that the increased number of estimations required can result in some variation in estimates. In our simulations this difficulty is not a concern, however in a more complex dataset, where the time series structure is more complex, such as in a higher order ARMA(p,q) order, it may be an issue. Furthermore D&R does not affect the structure of a time series in our simulations. Again this is a finding of our simulations but may be different in our application. This simulation study thus prepares for our upcoming application.

6.4 Application: Seperate train and tails model

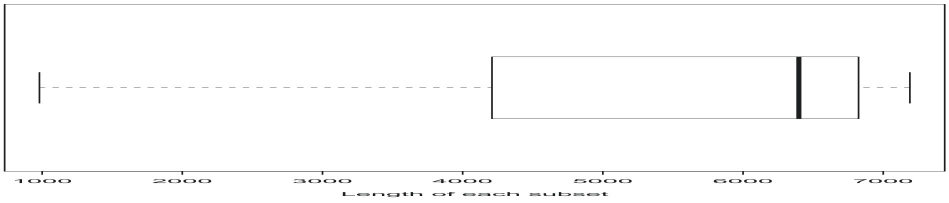
The restriction to temporal aggregation for an analysis of the coal train dataset has been the motivation behind our use of the D&R process. In this application we apply the D&R process to the unaggregated data as well as one and five minute aggregated data. For each analysis we compare a number of division choices. As part of this comparison we include the coefficient estimates as well as the effect of the division on the estimated error structure of each subset. The larger the subset size, the more likely that each subset retains the long memory structure. The vast majority of subsets for each dataset present long memory dependence, however there is a strong variation in the estimated ARFIMA(p,d,q) parameters. This is a key issue with D&R, as we are more likely to overfit the model due to the increased number of subsets versus one analysis on the complete data.

We complete this analysis by comparing the outcomes of the three datasets. We find that as we increase the aggregation period, the coefficient estimates for each passing train also increase. Based on our simulation results, and the fact that the estimates are not affected by the division method, we attribute this effect to the temporal aggregation rather than the D&R process. Furthermore, the tail estimates are less affected by the temporal aggregation. This suggests that the increased coefficient estimates are a result of the temporal aggregation.

6.4.1 Unaggregated Divide and Recombine model

To analyse the coal train dataset, and to account for the long memory dependence in the dataset, we utilise a linear regression with ARFIMA errors. For a more detailed description of this modelling strategy please refer back to chapter 3. The key issue with this model is that computer memory requirements restrict our analysis to datasets of a maximum size of around 12,000 observations. The particular challenge is the calculation of an inverse of a dense covariance matrix. It is for this reason that we have used aggregated datasets in this thesis so far. D&R allows us to analyse the unaggregated dataset for the first time without aggregation. We divide our dataset using replicate division and conditioning-variable division. For conditioning-variable division, we divide the dataset into a subset for each day of the analysis. As some subsets have a length exceeding our limit of 12,000 observations, we further divide each day again. Therefore for the 55 days of observations in our dataset, we now have 110 subsets. The distribution of the lengths of each subset is show in figure 6.2. Certain days do not have the maximum possible number of observations, namely 14,400, as there is some missing data in the dataset. In the case of replicate division, we divide the data by observations of length L, where we set $L = 500, 1000, 2000$ and 5000.

Figure 6.2: Lengths of L_k for each subset under conditioning-variable division.



Once we have completed the divide part of the D&R process, we fit the linear regression with ARFIMA errors to each subset k. Here each passing train is an indicator variable, indicating if it is passing the monitor at each observation. The tail variable we have created, and is also an indicator variable, where it is set to be 1 in the four minutes after each train has passed the monitor and 0 otherwise. This model is the subject of chapter 4 of our thesis. Here y_k is the log transform of the TSP air pollution variable.

$$\begin{aligned} y_k = \beta_{0k} &+ \beta_{1k} EmptyCoal + \beta_{2k} EmptyCoalTail + \beta_{3k} Freight \\ &+ \beta_{4k} FreightTail + \beta_{5k} LoadedCoal + \beta_{6k} LoadedCoalTail \\ &+ \beta_{7k} Passenger + \beta_{8k} PassengerTail + \beta_{9k} Unknown \\ &+ \beta_{10k} UnknownTail + \epsilon_k. \end{aligned} \tag{73}$$

For each subset k, we fit a long memory structure with ARFIMA(p,d,q) errors. Some of the subsets do not present evidence of long memory(although this is a small sample), in these cases an AR(p), MA(q) or ARMA(p,q) model is estimated for the subset.

$$\Phi(B)(1-B)^d \epsilon_k = \Theta(B)\gamma_k, \tag{74}$$

where $\gamma_k \sim NID(0, \sigma^2_{\gamma_k})$, and NID is normally and identically distributed. In figure 6.3 we show the distribution of the ARFIMA parameter estimates for two choices of subset length L. Once we have fit the model (73) for each subset, we recombine the estimates using the weighted means in equations (64) and (65).

In table 6.3, we can see that there is no difference in coefficient estimates and standard errors for each of the replicate division selections. The final 2 columns show the results when we utilise conditioning-variable division. Here we can see that the coefficient estimates are slightly larger than in replicate division. This difference in estimation suggests that the size of each subset has an effect on the estimation. This is further shown by the increase in the intercept estimate as we increase the size of each subset. It is our belief that this increase is due to difficulty in estimating the error structure for a larger amount of subsets as occurs when we have a smaller subset size.

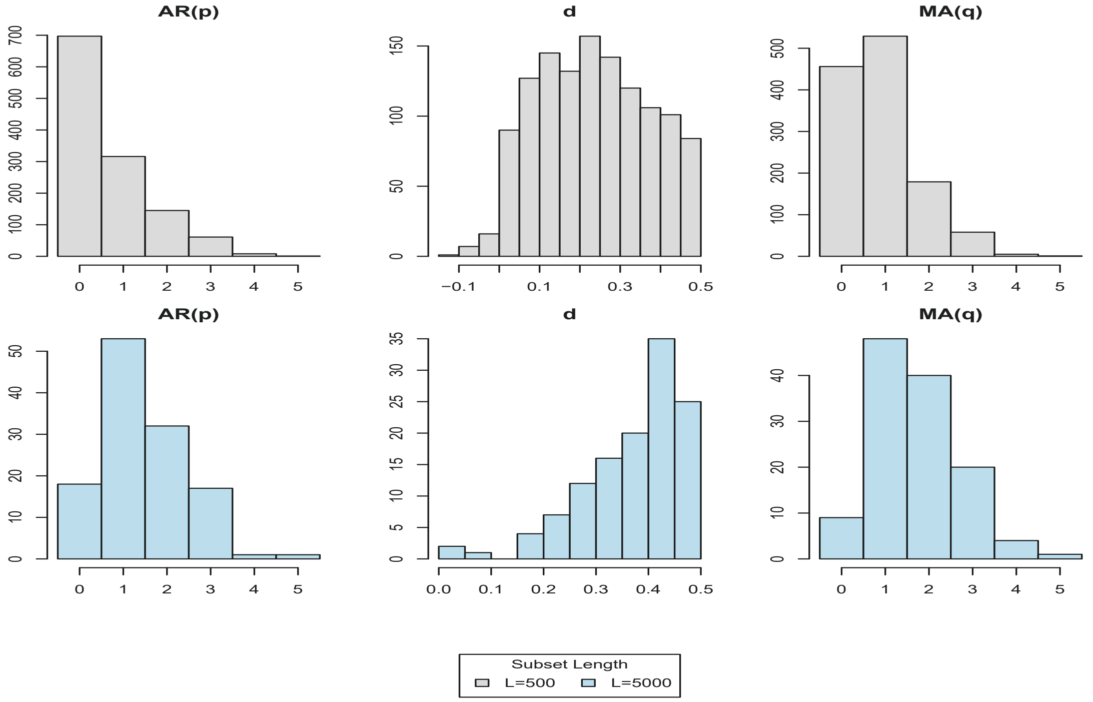
Table 6.3: D&R results. Unaggregated Data.

Subset	500	500	1000	1000	2000	2000	5000	5000	Days/2	Days/2
Size	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)
Int	3.452	0.002	3.457	0.003	3.578	0.005	3.822	0.011	3.715	0.014
Ec	0.025	0.004	0.024	0.004	0.023	0.004	0.022	0.004	0.032	0.004
Ec Tail	0.051	0.003	0.051	0.003	0.054	0.003	0.057	0.003	0.066	0.004
Fr	0.041	0.010	0.037	0.010	0.033	0.011	0.030	0.012	0.048	0.012
Fr Tail	0.060	0.007	0.062	0.008	0.059	0.008	0.055	0.009	0.061	0.009
Lc	0.028	0.004	0.025	0.004	0.024	0.004	0.022	0.004	0.031	0.004
Lc Tail	0.051	0.003	0.050	0.003	0.049	0.003	0.052	0.004	0.068	0.004

This is the first time in our analysis of the coal train dataset that is the motivation for this thesis, that we are able to analyse the unaggregated data while accounting for its long memory structure. There is a strong tail effect for each train, and the order of influence on air quality levels for the train types, in that the freight train has the largest effect, followed by the coal trains, is consistent with our previous analyses. We further compare our results using D&R and the previous modelling strategies in section 6.6.

From figure 6.3, we show the distribution of the estimate ARFIMA(p,d,q) parameters for each subset k, as in the equation (74), for the replicate division of size $L = 500$ and 5000. The ARMA(p,q) order is largely unaffected by the size of the dataset, indicating that the short term memory is constant. The fractional differencing parameter d, as shown by the middle column, however changes due to the size of each subset. For the longer subset length, the data shows a higher level of long term memory. In addition, we are able to show that the choice of a shorter subset length does not remove the long memory dependence. This indicates that the unaggregated dataset is unaffected by structural breaks, and is a true long memory dataset.

Figure 6.3: Comparision of ARFIMA(p,d,q) order for the residuals of each Divide and Recombine subset for the division of 500 and 5000 observations. Unaggregated data.



We now continue our analysis of the coal data with temporal aggregation.

6.4.2 Aggregated Divide and Recombine model

We now consider the effect of temporal aggregation on the coal train dataset. We consider both one and five minute aggregation. Under one minute aggregation, we take the mean of every ten observations for the air quality measure y_n, and for each train and tail type, we take a proportion of time each was passing the monitor for those ten observations. For five minute aggregation, we conduct the same process except for every 50 observations. Of particular interest is the 5 minute aggregation, as this is the lowest level of aggregation we have been able to implement thus far in our thesis. For both aggregation levels we utilise the same linear regression with ARFIMA errors model as in the unaggregated data. After the temporal aggregation has been applied, the s_k is log transformed. Here we have,

$$\begin{aligned} s_k = {} & \beta_{0k} + \beta_{1k} EmptyCoalRate + \beta_{2k} EmptyCoalTailRate + \beta_{3k} FreightRate \\ & + \beta_{4k} FreightTailRate + \beta_{5k} LoadedCoalRate + \beta_{6k} LoadedCoalTailRate \\ & + \beta_{7k} PassengerRate + \beta_{8k} PassengerTailRate + \beta_{9k} UnknownRate \\ & + \beta_{10k} UnknownTailRate + \omega_k. \end{aligned} \tag{75}$$

For each subset k, we again have an ARFIMA(p,d,q) error structure,

$$\Phi(B)(1-B)^d \omega_k = \Theta(B)\eta_k, \tag{76}$$

where $\eta_k \sim NID(0, \sigma^2_{\eta_k})$.

For consistency, we analyse the data with both division options for D&R. For conditioning-variable division, we divide the aggregated data into days. We do not need a further division, as the maximum length of each day under 1 minute aggregation is 1,500 observations, and for 5 minutes it is 300 observations. For aggregation at 5 minutes, an aspect of D&R is that we can have subsets that are short in length. For this reason, we also include the replicate division. In the case of $L = 100$, this subset corresponds to $L = 5000$ for the unaggregated data. The choice of $L = 2000$ equates to 100,000 observations of the unaggregated data, which is almost 1/6 of the dataset. The use of these alternate division selections is helpful in two ways. Firstly it helps us determine if there is any change in the time series structure of the data for different periods, and secondly we can determine if the choice of D&R subset length impacts the analysis.

6.4.3 1 minute aggregation

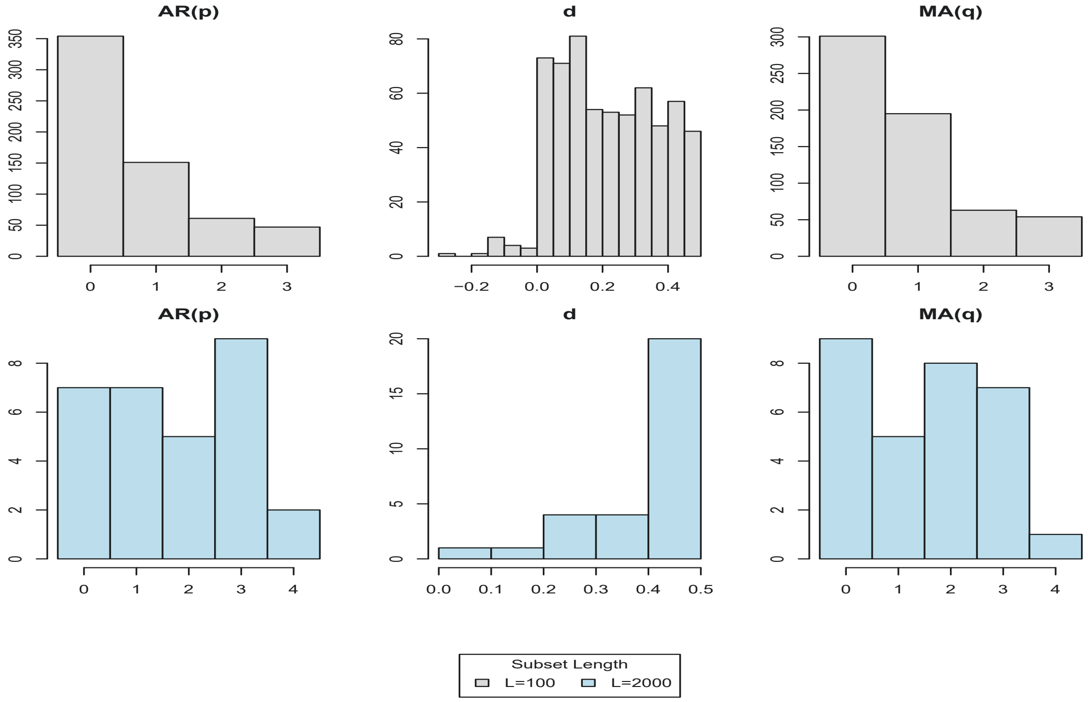
We begin out aggregated D&R analysis with one minute aggregation. Table 6.4 shows the outcomes for the model in equation (75) under D&R. The coefficient estimates are increasing in variation, depending on the subset size, in comparison to the unaggregated data. The coefficient estimates have also increased. We consider this effect in further detail in section 6.4.5.

Table 6.4: D&R results. Aggregation period of $J = 1$ minute.

Subset	100	100	200	200	500	500	1000	1000	2000	2000	Days	Days
Size	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)
Int	3.534	0.002	3.650	0.004	3.798	0.012	3.551	0.032	3.494	0.075	3.669	0.027
Ec	0.033	0.005	0.033	0.005	0.036	0.005	0.034	0.005	0.042	0.006	0.033	0.005
Ec Tail	0.081	0.003	0.082	0.004	0.089	0.004	0.093	0.004	0.104	0.004	0.095	0.004
Fr	0.046	0.014	0.032	0.015	0.036	0.016	0.053	0.017	0.030	0.017	0.037	0.016
Fr Tail	0.084	0.008	0.076	0.009	0.077	0.010	0.085	0.011	0.083	0.011	0.084	0.010
Lc	0.032	0.004	0.035	0.005	0.036	0.005	0.040	0.005	0.036	0.006	0.031	0.005
Lc Tail	0.064	0.003	0.067	0.004	0.074	0.004	0.082	0.004	0.082	0.004	0.081	0.004

In figure 6.4 we compare the estimated ARFIMA(p,d,q) parameters for the replicate lengths of $L = 100, 2000$. As for the unaggregated data, we find that the larger the length of each subset the more likely it is that the series has long memory. For the shorter subsets, they again mainly have a long memory dependence, however in some cases, we have short term memory only (when $d = 0$). We also have a few subsets where the fractoral differencing parameter $d < 0$. This is an indication of over-differencing. This underlines one of the issues with D&R for complex data, where the extra amount of datasets to analyse can increase the possibility of an incorrect estimation of the model. Nonetheless, we would be unable to analyse the data at this level of aggregation without D&R so it is a trade-off we must accept.

Figure 6.4: Comparision of ARFIMA(p,d,q) order for the residuals of each Divide and Recombine subset for the division of 100 and 2000 observations. 1 minute aggregation.



We now consider the same analysis for five minute aggregation.

6.4.4 5 minute aggregation

In all of our previous analyses, see chapter 3, 4 and 5, we have been limited to data that has been aggregated to a five minute period. Due to D&R we can now analyse the data in its unaggregated format using the ARFIMA model. Nonetheless, we implement the D&R process for the 5 minute aggregation. This primarily results in a significant speed up of our analysis. We are able to now analyse the aggregated data in under five minutes. This is extremely welcome, particularly when the full model (with no D&R) takes upwards of 40 minutes for the aggregated data (at 5 minute aggregation). We discuss this calculation time further in section 6.6.3.

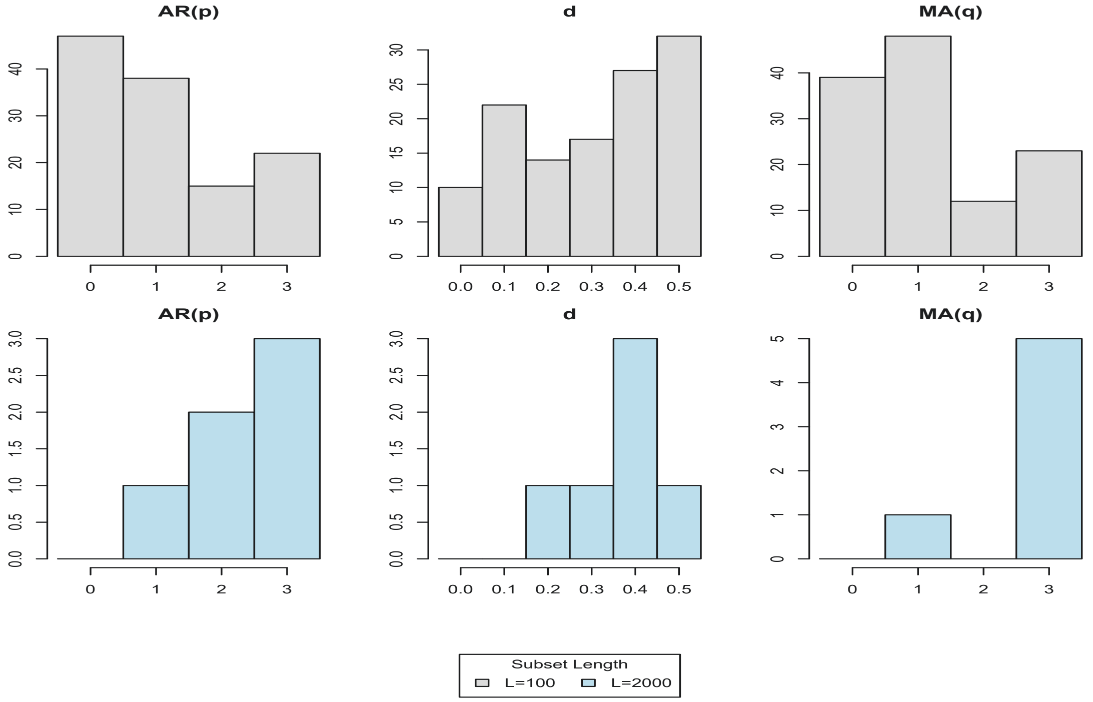
The results of our analyses are shown in table 6.5 are similar to those for 1 minute aggregation. Namely, we are seeing a continued trend in an increasing coefficient estimate as we increase the aggregation period. The choice of subset length has an impact on the coefficient estimates of our model. This is to be expected as each subset length selection will result in a slightly different error structure, which can influence the estimates.

Table 6.5: D&R results. Aggregation period of $J = 5$ minutes.

Subset	100	100	200	200	500	500	1000	1000	2000	2000	Days	Days
Size	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)
Int	3.623	0.012	3.479	0.028	3.613	0.055	3.433	0.082	3.488	0.118	3.648	0.023
Ec	0.075	0.015	0.061	0.017	0.067	0.018	0.062	0.018	0.068	0.018	0.062	0.016
Ec Tail	0.122	0.007	0.146	0.008	0.152	0.008	0.149	0.008	0.152	0.009	0.137	0.008
Fr	0.236	0.045	0.260	0.048	0.244	0.048	0.239	0.048	0.253	0.048	0.195	0.045
Fr Tail	0.066	0.016	0.081	0.017	0.076	0.017	0.082	0.017	0.075	0.017	0.075	0.016
Lc	0.091	0.010	0.098	0.012	0.108	0.012	0.101	0.013	0.114	0.013	0.096	0.011
Lc Tail	0.087	0.006	0.100	0.006	0.106	0.006	0.101	0.007	0.102	0.007	0.094	0.006

Figure 6.5 indicates that the length of each subsets division in D&R is not affecting the time series structure of the dataset. The ARMA(p,q) parameters are slightly lower for the smaller subset size of $L = 100$, however the fractoral differencing parameter d estimates are between $0 < d < 0.5$, indicating that each subset has a long memory dependence. Thus the 5 minute aggregated data retains the same error structure as the unaggregated data. This is particularly insightful, as the D&R process is not impacting the analysis.

Figure 6.5: Comparision of ARFIMA(p,d,q) order for the residuals of each Divide and Recombine subset for the division of 100 and 2000 observations. 5 minute aggregation.

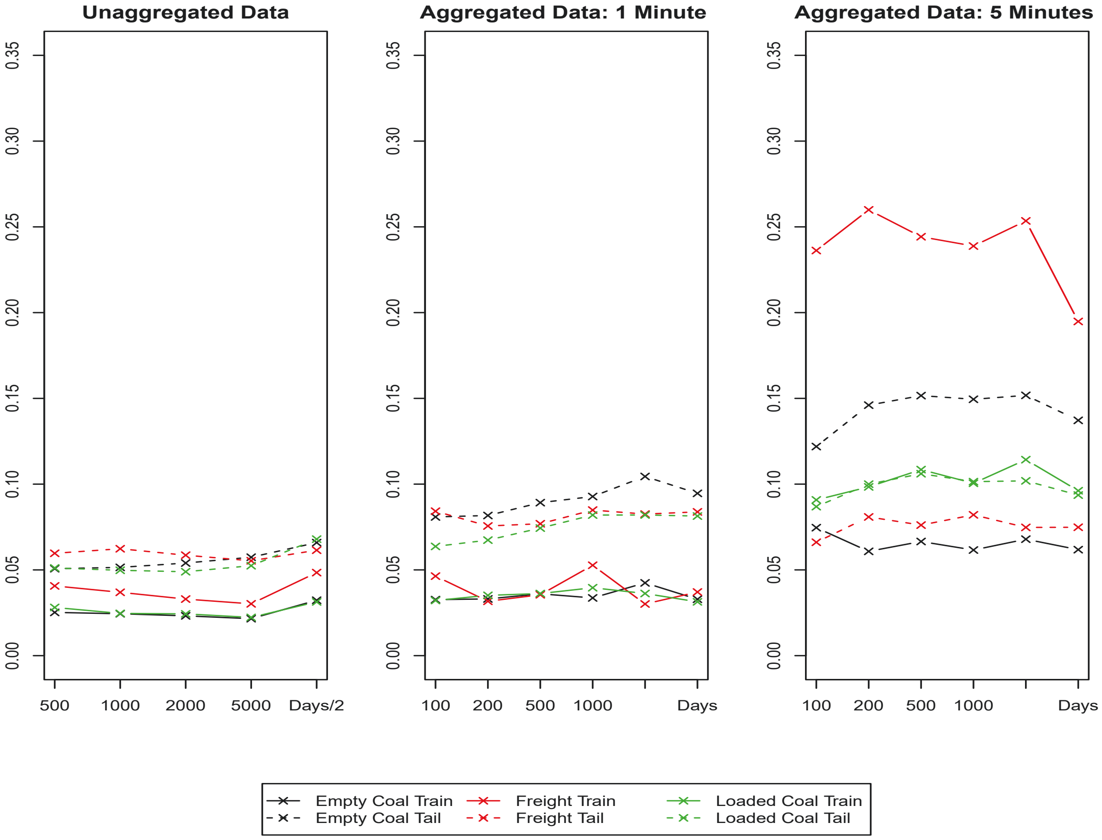


Each of our D&R analyses for the coal data have resulted in slightly differing coefficient estimates for the train types. We thus continue our analysis of this coal dataset with a review of the 3 datasets(unaggregated, 1 minute aggregated, and 5 minute aggregated).

6.4.5 Review of application

In our application we have analysed the D&R outcomes for unaggregated data and data for aggregation periods of one and five minutes. A comparison of these results is shown in figure 6.6. In this figure we can see that as we aggregate the data, the coefficient estimates for each train increase. The increasing coefficient effect is not as strong for the train tails. We can only conclude that there is an effect that is not being captured by our model. Given the data we have, we are unable to determine the further cause of this misspecification with any certainty. Nonetheless, we inspect the distribution of coefficient estimates for both the trains and tails in figures 6.7 and 6.8.

Figure 6.6: Divide and Recombine coefficient estimates for the Assumed model. Data divided into replicates and by conditioning-variables for each day.



From figures 6.7 and 6.8, we can see that the increasing coefficient effect that occurs in the aggregated data, can be attributed to a larger variation in the estimates. In particular, the distribution of the estimates is moving to the right, signifying an increase on the average. The tail estimates in figure 6.8 do not present as much of an change in the distribution as the train estimates. Our results in figure 6.6 also indicate that the tails are not as impacted by aggregation. Therefore the effect of aggregation is predominantly centered on the train covariates.

Figure 6.7: Divide and Recombine coefficient estimates for each train type by aggregation period in each subset. Data divided by conditioning-variables for each day. (In the unaggregated case, we have a further subdivision within each day.)

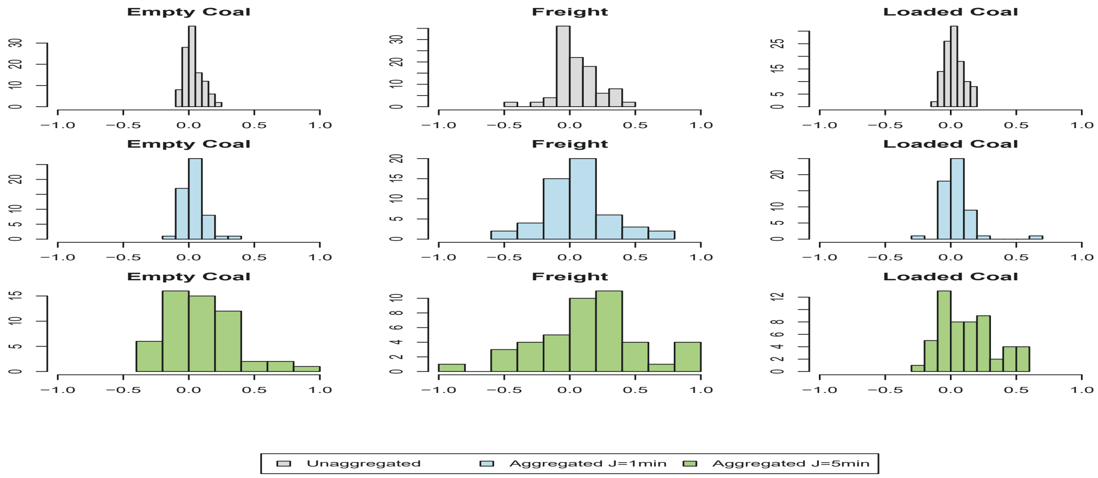
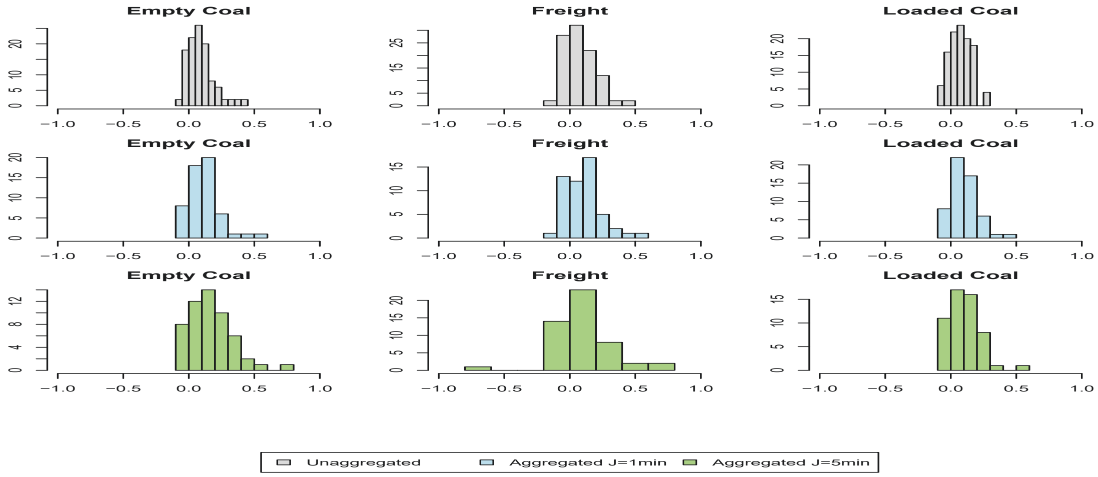


Figure 6.8: Divide and Recombine coefficient estimates for each train TAIL type by aggregation period in each subset. Data divided by conditioning-variables for each day. (In the unaggregated case, we have a further subdivision within each day.)



A key aspect of the aggregated data is the decrease in time to analyse the data. For the unaggregated data, each analysis with Divide and Recombine can take between 6 and 12 hours to complete, while we cannot even analyse the data for the undivided dataset. As we increase the aggregation period, the time to analyse the data reduces greatly. For 5 minute aggregation, we reduce the time to analyse the data without Divide and Recombine (40 minutes) to under 10 minutes using Divide and Recombine. We discuss this further in section 6.6.3.

Given the variation in our results for both the train and tail estimates, we now turn to another model, which we call the 'combined' model. We incorporated this modelling strategy in chapter 4. Here we consider its performance using D&R.

6.5 Application: Combined train and tails model

The key question of our coal dataset is to determine the effect of each passing train type on air particulate levels. Throughout our work on this dataset, we have suggested, and shown (see chapter 4), that there is a significant tail effect for the period after a train has passed the monitor. As in section 6.4, we have considered the trains and tails as seperate covariates. Another option is to combine these two variables into one. This 'combined' model answers the question of which train, where a tail effect is attributed to a particular train type, contributes to the air particulate levels the most succinctly.

In this section of the chapter we present the results of the D&R analysis for the combined model. We consider both the unaggregated dataset, and the aggregation periods of one to five minutes. We limit our analysis to the conditioning-variable division of each day as we have shown in section 6.4 that the choice of division does not impact the analysis.

Or combined model for the unaggregated dataset is,

$$\begin{aligned} y_k = {} & \beta_{0k} + \beta_{1k} EmptyCoalTrainAndTail + \beta_{2k} FreightTrainAndTail \\ & + \beta_{3k} LoadedCoalTrainAndTail + \beta_{4k} PassengerTrainAndTail \\ & + \beta_{5k} UnknownTrainAndTail + \epsilon_k. \end{aligned} \tag{77}$$

For each subset k, we have an ARFIMA error structure,

$$\Phi(B)(1-B)^d \epsilon_k = \Theta(B)\gamma_k, \tag{78}$$

where $\gamma_k \sim NID(0, \sigma^2_{\gamma_k})$.

Whereas the combined model for the aggregated dataset is,

$$\begin{aligned} s_k = {} & \beta_{0k} + \beta_{1k} EmptyCoalTrainAndTailRate + \beta_{2k} FreightTrainAndTailRate \\ & + \beta_{3k} LoadedCoalTrainAndTailRate + \beta_{4k} PassengerTrainAndTailRate \\ & + \beta_{5k} UnknownTrainAndTailRate + \omega_k. \end{aligned} \tag{79}$$

For each subset k, we have an ARFIMA error structure,

$$\Phi(B)(1-B)^d \omega_k = \Theta(B)\eta_k, \tag{80}$$

where $\eta_k \sim NID(0, \sigma^2_{\eta_k})$.

For both y_k and s_k in the equations 77 and 79, the TSP air pollution variable is log transformed. We present our results for these two models in figure 6.9 and table 6.6. In figure 6.9, we can see that as we increase the aggregation period, the coefficient estimates for each train type increase. This is the same effect that we saw in section 6.4, however the increasing effect is not as large as in the seperate model. The coefficient estimates and their respective standard errors are shown in table 6.6.

Figure 6.9: Divide and Recombine coefficient estimates for the Combined model. Data divided by conditioning-variables for each day, and in the unaggregated case, it is further divided within each day. 4 minute tails.

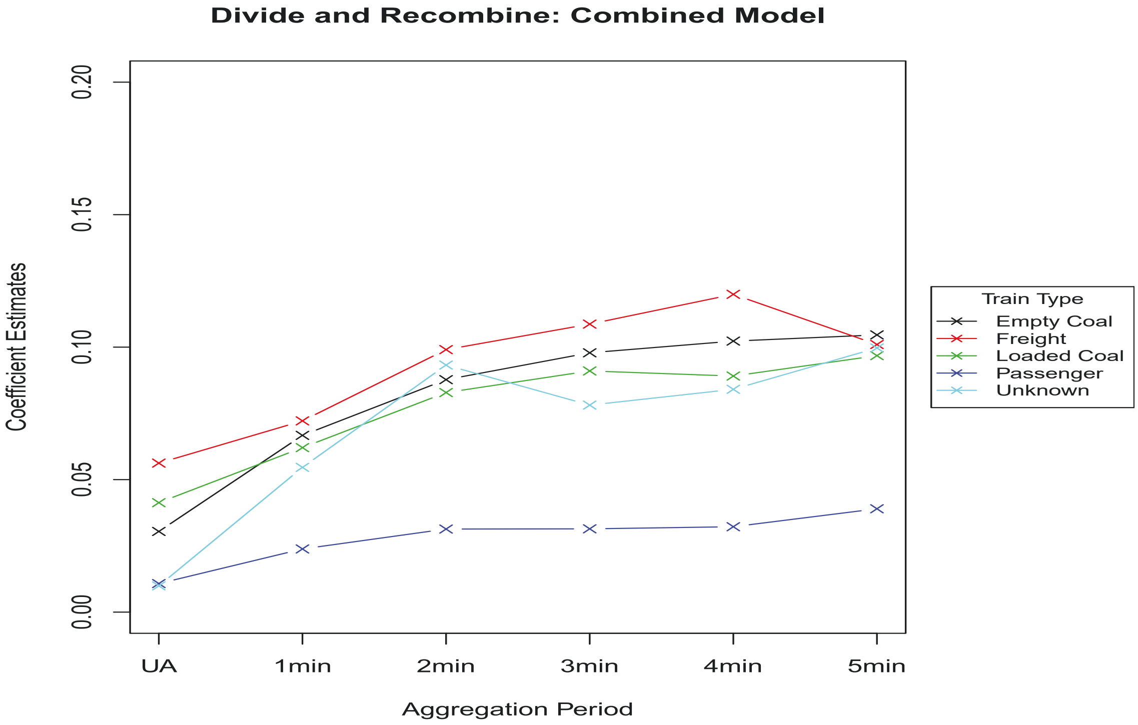
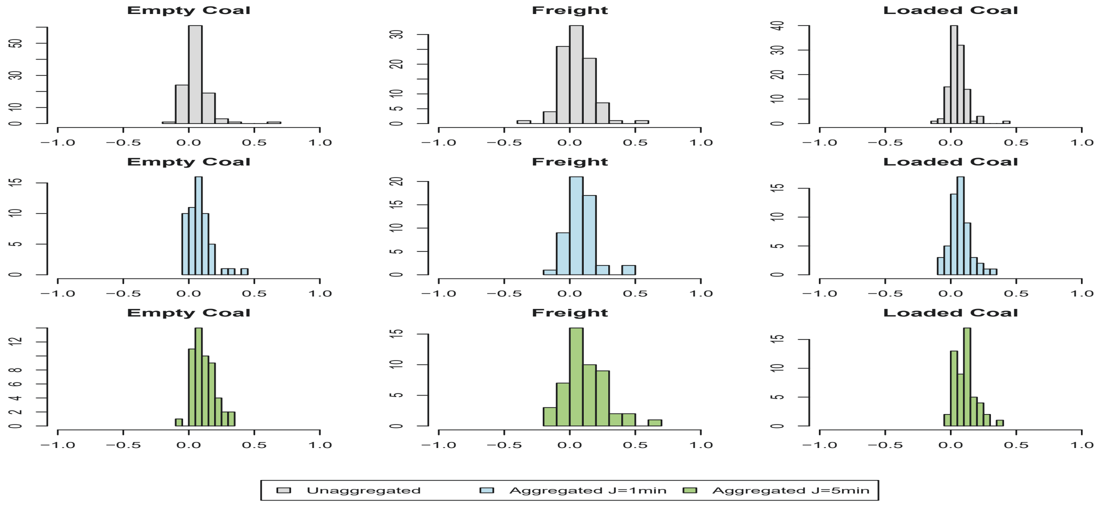


Table 6.6: Combined Model results for unaggregated to aggregated data of 1 to 5 minutes. Tail length is 4 minutes.

	UA		1 Min		2 Min		3 Min		4 Min		5 Min	
	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)
Int	3.538	0.003	3.675	0.027	3.191	0.028	3.603	0.029	3.634	0.032	3.779	0.028
Ec	0.030	0.006	0.067	0.004	0.088	0.004	0.098	0.004	0.102	0.004	0.105	0.005
Fr	0.056	0.004	0.072	0.009	0.099	0.010	0.109	0.011	0.120	0.011	0.101	0.012
Lc	0.041	0.003	0.062	0.004	0.083	0.004	0.091	0.004	0.089	0.005	0.097	0.005
Pas	0.011	0.011	0.024	0.003	0.031	0.003	0.031	0.004	0.032	0.004	0.039	0.004
Un	0.010	0.023	0.055	0.016	0.093	0.018	0.078	0.019	0.084	0.020	0.100	0.023

In figure 6.10 we explore the effect of aggregation on the coefficient estimates for each subset of the D&R model for the unaggregated and aggregated datasets. Here we can see that as we increase the aggregation, there is a movement of the distribution to the right, signifying a mean increase of the effect. However, if we compare these results to those of the seperate model in figures 6.7 and 6.8, the effect is reduced.

Figure 6.10: Divide and Recombine coefficient estimates for each train type by aggregation period in each subset. Data divided by conditioning-variables for each day. (In the unaggregated case, we have a further subdivision within each day.) Combined Model.



Our analysis of the combined model has reduced the variation in results compared to the seperate model in section 6.4. Our analyses indicate that the increased coefficient effect is due to temporal aggregation rather than the D&R process. The implementation of the combined model has stabilised our coefficient effects. To continue our analysis of the coal train dataset with the D&R process, we now turn to a comparison of the D&R process and the analyses using the full data model from

Chapter 4.

6.6 Comparison of Divide and Recombine with models using full dataset (as in chapter 4)

This chapter of the thesis has been concerned with the implementation of the D&R process to the same 'seperate' and 'combined' train and tail models as in chapter 4. The use of D&R has allowed us to analyse the unaggregated dataset, and the lower levels of aggregation below five minutes. This was not possible in our analyses in chapter 4 due to computer memory limitations, particularly the estimation of the inverse of a large, dense covariance matrix. We now compare our results for the D&R models against the 'full' model, using the complete and undivided data, from chapter 4. Firstly we compare the results and then we consider the time to analyse each analysis.

6.6.1 Seperate train and tails model

Under D&R we have the model as presented in section 6.4. For the full model, there is no D&R process and we analyse the complete dataset. This is equivalent to setting $K = 1$ in equations (81) and (82).

$$\begin{aligned} s_k = \beta_{0k} &+ \beta_{1k} EmptyCoalRate + \beta_{2k} EmptyCoalTailRate + \beta_{3k} FreightRate \\ &+ \beta_{4k} FreightTailRate + \beta_{5k} LoadedCoalRate + \beta_{6k} LoadedCoalTailRate \\ &+ \beta_{7k} PassengerRate + \beta_{8k} PassengerTailRate + \beta_{9k} UnknownRate \\ &+ \beta_{10k} UnknownTailRate + \omega_k. \end{aligned} \tag{81}$$

For each subset k, we have an ARFIMA(p,d,q) error structure,

$$\Phi(B)(1-B)^d \omega_k = \Theta(B)\eta_k, \tag{82}$$

where $\eta_k \sim NID(0, \sigma^2_{\eta_k})$.

Table 6.7: Comparison of full data (Chapter 4) and D&R (Chapter 6) results: 5 minute aggregation

	Intercept	Empty Coal		Freight		Loaded Coal	
		Train	Tail	Train	Tail	Train	Tail
	Full Model: Chapter 4						
Coef	3.289	0.149	0.114	0.246	0.075	0.1334	0.098
Std.err	0.017	0.046	0.017	0.013	0.007	0.173	0.006
	D&R Model: Chapter 6: Days Division						
Coef	3.648	0.062	0.137	0.195	0.075	0.096	0.094
Std.err	0.023	0.016	0.008	0.045	0.016	0.011	0.006

In table 6.7 we can see that the tail coefficient estimates are similar across both models. The train effect however, is quite variable between the two models. In section 6.5 we showed that the implementation of the combined model can reduce this variability. We consider this model now.

6.6.2 Combined train and tails model

In section 6.5 we applied the D&R process for the combined model. Here we 'combine' each passing train and tail into one covariate for each train type. We now compare the results of this combined model, as shown in equation (83), for both the D&R and full models. The full model is equivalent to setting $K = 1$.

$$\begin{aligned} s_k = \beta_{0k} &+ \beta_{1k} EmptyCoalTrainAndTailRate + \beta_{2k} FreightTrainAndTailRate \\ &+ \beta_{3k} LoadedCoalTrainAndTailRate + \beta_{4k} PassengerTrainAndTailRate \\ &+ \beta_{5k} UnknownTrainAndTailRate + \omega_k. \end{aligned} \tag{83}$$

For each subset k, we have an ARFIMA error structure,

$$\Phi(B)(1-B)^d \omega_k = \Theta(B)\eta_k, \tag{84}$$

where $\eta_k \sim NID(0, \sigma^2_{\eta_k})$.

In table 6.8 we can see that the estimates for each train type are similar for both processes. The main difference is in the intercept estimate $\hat{\beta}_0$.

Table 6.8: Comparison of full data (Chapter 4) and D&R (Chapter 6) results: 5 minute aggregation. Combined model.

	int	ec	fr	lc	pas	un
	Full Model: Chapter 4					
Coef	3.291	0.121	0.108	0.107	0.042	0.090
Std.err	0.300	0.005	0.013	0.005	0.005	0.023
	D&R Model: Chapter 6:Days Division					
Coef	3.779	0.105	0.101	0.097	0.039	0.100
Std.err	0.028	0.005	0.012	0.005	0.004	0.023

The results as shown in table 6.8 indicate that the D&R process can have consistent results as the model on the full dataset. Through our analyses of the seperate model, we can see that the majority of the variability in our output is due to the train rather than tail effect. This suggests that the model is not capturing the complete effect of the passing trains.

We conclude our comparison of the D&R and full models with an inspection into the time to analyse each model.

6.6.3 Time comparison of chapter 4 and chapter 6 analyses

A key reasoning for the implementation of the D&R process on our coal dataset is due to its assistance in the analysis of the unaggregated data. However, a key resulting outcome of this process is that we are able to significantly reduce the time to analyse each model. We provide the time of analysis for each model used in this chapter and chapter 4 here in table 6.9.

Table 6.9: Comparison of full data (Chapter 4) and D&R (Chapter 6) analyses timings. Calculation time is in minutes. (*) here we have divided each day into two subsets due to the memory constraints.

Aggregation Period	Full Model Chapter 4	D&R Model Chapter 6
Unaggregated	NA	449*
1 Minute	NA	23
2 Minute	NA	7
3 Minute	NA	8
4 Minute	NA	6
5 Minute	40	5

In table 6.9, we compare the computing time to analyse the models in chapter 4 and 6. Using the full dataset, as in chapter 4, we are limited to a dataset of under 15,000 observations in length, due to the requirements of the ARFIMA model needing to estimate the inverse of a dense, covariance matrix. The unaggregated dataset has over 600,000 observations. At 5 minute aggregation, we have reduced the dataset to around 12,000 observations. This is possible under the ARFIMA model, however as we can see in the 1st column of the table, this takes 40 minutes to run. On the other hand, the implementation of the D&R process allows this same dataset to be analysed in around 5 minutes.

The key impact of the D&R process, is that we are able to fit the ARFIMA model to the unaggregated dataset. This analysis is lengthy, as it take around 7.5 hours to run. Nonetheless, as we can see from the NA values in the first row, we are unable to otherwise analyse the unaggregated data, or even the lower level temporal aggregations. We are still limited by the size of the data for the unaggregated dataset, and as we can see from the third column, we must divide the data into days, and then again divide into two for the unaggregated data. This is because some of the days have over 15,000 observations, which is not possible for our ARFIMA model.

It is clear that the D&R process can be extremely useful in big data settings. This is particularly the case in situations, such as the coal train data, where the data is not only large, but also requires complex models. Not only are we able to analyse data that we were previously not able to, but in other circumstances we have significantly reduced the computational time, allowing for more meaningful insight into the data, rather than waiting for the analysis to complete.

6.7 Discussion

Our analysis of the coal train dataset has been severely constrained by it's size and complexity. The computer memory requirements that are necessary for our analysis, have limited the size of the dataset that we can utilise. This has lead to our implementation of temporal aggregation to reduce the size of our dataset.

The use of the Divide and Recombine process on our dataset has removed our need to transform our data by aggregation. We are now able to analyse the original, unaggregated dataset using the same linear regression model with ARFIMA errors as we have throughout this thesis for aggregated data. By dividing the data into subsets, performing an analysis on each subset, and then recombining the results into a final estimate, we have not only been able to analyse the unaggregated dataset for the first time, but we have also achieved a significant reduction in computational time for the aggregated datasets. In the case of the 5 minute temporally aggregated data, we have reduced the computational time from 40 minutes to 5 minutes using Divide and Recombine.

Our final output for the unaggregated and aggregated models differs in the coefficient estimates. We suggest that this is due to the temporal aggregation rather than the D&R process. We attribute this suggestion due to the fact that our simulations show no difference between the full and D&R models. We are unable to make this comparison for the unaggregated coal train data, as we cannot analyse the unaggregated dataset without D&R.

D&R has been applied to a number of datasets and modelling techniques. The majority of the literature has been on linear regressions with iid error structure. We have extended this work by considering the performance of D&R on a time series dataset. In particular, the use of D&R has been necessary for the analysis of our coal train dataset, which exhibits long memory dependence. The main difficulties presented by a D&R process on time series data are that:

1. we lose information between subsets,
2. the division into subsets results in a number of analyses, instead of just one on the full dataset. This increases the risk of overfitting the model.

To account for the loss in information between subsets, we consider two division methods for D&R. Conditioning-variable division, where we divide the data by a data dependent variable, in our case by days of the data, and replicate division, where we divide the data into non-overlapping subsets based on time. Here we select consecutive observations, such as every 500, 1000 or 5000 observations. We are thus able to compare our analyses on different subset sizes, and our results show that there is no significant difference between results on different subset sizes.

A more pressing issue, is that due to the D&R process, we must estimate a number of model parameters for each subset, rather than for just one model under the full dataset. Under a time series dataset, where there are a number of extra parameters to estimate, unlike in the iid case, there is a higher likelihood of overfitting the model. For a dataset with long memory dependence, as in our coal dataset, an ARFIMA(p,d,q) error structure can be difficult to estimate accurately. Nonetheless, without D&R, this model would not be possible. Thus this is a trade-off one must consider in application.

Research into long memory datasets, has shown that there can be structural breaks in the data, where the full dataset is not in fact long memory, but rather a mixture of short and long memory subsets. The implementation of D&R allows us to analyse the structure of the dataset that may not be considered otherwise. Furthermore, we can then fit differing error structure as required for each subset. Throughout our analysis, we have shown that the vast majority of subsets retain the long memory structure of the full dataset.

We suggest future work on D&R for a simpler time series structure, such as an AR(p) or ARMA(p,q) process, where the chance of overfitting is reduced due to a smaller number of parameters to estimate. To counter the loss of information between subsets, we also propose an extension of the D&R process whereby the data is divided into overlapping intervals. These are two possible areas we wish to explore in the future.

7 Discussion and Future Research

Big data has become a cornerstone of statistical analysis. There has been much recent work in statistics to extend current methods for the big data era. The advances in the collection and storage of such big datasets has also created issues for analysis of complex datasets such those in the time series domain.

Our focus has been on a particular area of time series, namely long memory dependence. Current methods for long memory time series are extremely computationally demanding, thus limiting analyses to relatively small datasets (in the big data era), unless the statistician has access to greater computing power. Yet this approach of a more powerful computer is not usually available, and thus, work must be done to allow for the standard computer to analyse big and complex data.

This thesis has been motivated by the Hunter Valley Coal Train dataset. It's size and complexity, with over 600,000 observations and the presence of long memory dependence, have limited the use of current statistical methods. The aim of this thesis has been to either transform big and complex datasets such that current statistical methods can be utilised on a standard personal computer, and/or to extend current statistical techniques for big and complex data. To transform the data we consider temporal aggregation. This reduces the data volume and can sometimes reduce the complexity of certain time series datasets, particularly those exhibiting long memory dependence. Temporal aggregation can often be negatively viewed as it can result in a loss of information. This is a trade-off that we must undertake, as otherwise these big and complex datasets would not be possible to analyse. To reduce this loss of information, we implement a temporal aggregation transformation of the data into a bivariate series. The idea behind this transformation, is that by aggregating to two symbols, such as means and standard deviations, or median and ranges for each aggregation interval, we are able to reduce the loss of information from the temporal aggregation transformation.

The transformation of the dataset enables their analysis. Unfortunately, under long memory dependence, we remain restricted by the size of the datasets, even after temporal aggregation. Therefore, we extend the Divide and Recombine (D&R) process for time series data. This allows us to analyse the unaggregated data, without any data transform such as temporal aggregation. However, to the best of our knowledge, D&R has not been considered for time series datasets in the statistical literature. We explore some of the issues faced by this process, and implement some approaches that allow for D&R of time series datasets, particularly in the long memory case.

In chapter 2 we discuss the motivating Hunter Valley Coal Train dataset in detail. We outline some of the challenges faced by this dataset as well as some of our previous analyses. The use of a Generalized Additive Model is restricted by the time series nature of the data. We thus implement a block bootstrap to account for the autocorrelations. This process does not adequately account for the long memory dependence and is computationally demanding. To correctly model the long memory dependence, in chapter 3, we implement a linear regression with ARFIMA errors. This modelling strategy is constrained by the need for the calculation of the inverse of a large and dense covariance matrix. Temporal aggregation reduces the data volume and some of the complexity of the time series data, enabling the analysis of the Hunter Valley Coal Train dataset. Our findings in chapter 3, suggested that our model was not accounting for all of the effects associated with each passing train on air particulate levels. Thus, in chapter 4, we considered the model misspecification under a linear regression with ARFIMA errors. Our findings suggest that there is a tail effect for each passing train. We are able to show through some theory and an extensive simulation study that temporal aggregation can be used to select a tail length for each passing train. The outcomes of this chapter are consistent with our GAM analysis in chapter 2. The use of temporal aggregation to reduce the data volume, thereby enabling the use of a linear regression with ARFIMA errors to model the long memory dependence, has the trade-off that we lose information by the aggregation transformation. In chapter 5, we extend temporal aggregation to the bivariate case. We consider two air particulate levels, PM1 and PM2.5, and implement a H-Likelihood approach that enables us to reduce the information loss from aggregation, by concurrently analysing the two time seres. We extend current H-Likelihood models to account for autocorrelation between and across the bivariate series. This strategy is the most computationally demanding of our thesis, and we are limited to datasets of 5 thousand observations. While this is not ideal for the analysis of our Hunter Valley Coal Train dataset, it can be helpful for modelling complex data on a smaller scale. This approach reduces the information loss due to temporal aggregation. We also consider the case of creating a bivariate series by aggregating to means and standard deviations, as well as to medians and ranges.

Throughout this thesis, our motivating dataset has been the Hunter Valley Coal Train data. With over 600,000 observations, our modelling strategy of a linear regression with ARFIMA errors to account for the long memory dependence, has been limited to datasets of 10 to 15 thousand observations. D&R is a process that has been created for the analyses of big data, with applications in the Hadoop and MapReduce. However this approach has not been considered for time series data. In chapter 6, we extend the D&R process to time series datasets. We show that the selection of the division method has a crucial impact on the efficient analysis. By dividing the data into subsets, such that the long memory dependence of the complete dataset is retained in each subset, we are able to achieve consistent coefficient estimates for this D&R approach as we have throughout the other modelling strategies in this thesis. We explore the impact of different subset size on our motivating dataset. The trade-off we encounter here is that as time series models are quite complex and difficult to fit, we are at risk of model overfitting as there are a number of subsets to estimate. This is a downside to the D&R process for time series, however without its use, we are unable to analyse the entire dataset without temporal aggregation. This approach not only allows us to analyse the complete dataset, it also results in a significant reduction in computer timings.

In each of the chapters we have attempted to find a way to analyse the Hunter Valley Coal Train dataset correctly. While achieving consistent results across the methods, and a vast speed up in analysis time, there are still positives and issues with all of our approaches. This work has lead us to consider a number of other methods for future study. We have shown in chapter 2 that a block bootstrap permitted the analysis of the data using the GAM model, however once we uncovered the long memory dependence, it became clear that this is not adequately accounting for all of the autocorrelation in the data. Thus, further work into bootstrap[ing for long memory dependence would be advantageous. We can also see a parallel between the current block bootstrap method and banding of the covariance matrix to reduce its complexity, thereby reducing the computational memory demands for its inversion. This can further be associated with the D&R process, where in all three methods we are cutting off information after a certain lag. It is known that for long memory data, lags after these cutoffs can remain significant, and we have shown this to be the case in our analyses. Our D&R approach for time series has been conducted for long memory data. It would be of particular interest to extend this theory to simpler ts data structures, such as AR(1) or ARMA processes. These datasets, while not as computationally demanding, would benefit from a significant increase in computational speed of analysis with such research. With regards to transforming data for big and complex datasets, the area of symbolic data analysis is a domain of interest. We have considered temporal aggregation for univariate and bivariate series. These are both symbolic data analysis methods. Further research into other symbols such as distributions and histograms is of particular interest to reduce the information loss that is a result of temporal aggregation.

While many areas of statistics are attacking the difficulties of big data analysis, we have focused on the area of long memory dependence. Many big data concern themselves with datasets of gigabytes, terrabytes, pegabytes or even data so large that it overflows data warehouses. Yet we have shown that big and complex data can limit some analyses to 10 or 15 thousand observations. If we are to draw inference from such interesting datasets, which are becoming the norm in the big data era, then we must significantly advance the current statistical literature. We have shown that transforming the data to reduce its volume and complexity is one such approach, and that extending current methods such as D&R for time series data is another.

Motivating this thesis has been the Hunter Valley Coal Train dataset. The data was collected as part of a study by the EPA, with the goal of determining if passing trains impacted air particulate levels, and if so, how much did each passing train type contribute. We have considered a number of models, from GAM to ARFIMA, to answer these questions. The results of these modelling strategies have been consistent. We have been able to show that the passing trains result in increased air particulate levels. Our analyses of this dataset have revealed many other details, such as, that the period after a train has passed has a significant impact on air particulate levels. The second question that was asked of this dataset, was if we could determine the effect of each passing train type. We are able to infer that although there is some variation in the results, the empty coal trains contribute the largest effect to air particulate levels, followed by freight and then loaded coal trains. These effect sizes are small, but that make sense in that our findings suggest that it is the coal dust that is on the rail tracks, that is being stirred up by passing trains. This suggests that train speed has the largest impact on the air particulate levels, which is supported by our results. The empty coal trains have the highest average speed, followed by freight then loaded coal trains. The analysis of this big and complex dataset has presented many difficulties, however by transforming the data and extending D&R for time series data, we have been able to determine the effect of each passing train on air particulate levels, as was not possible before.

References

[1] Hirotugu Akaike. Statistical predictor identification. Annals of the institute of Statistical Mathematics, 22(1):203–217, 1970.

[2] Takeshi Amemiya and Roland Y Wu. The effect of aggregation on prediction in the autoregressive model. Journal of the American Statistical Association, 67(339):628–632, 1972.

[3] Richard T Baillie. Long memory processes and fractional integration in econometrics. Journal of econometrics, 73(1):5–59, 1996.

[4] Heather Battey, Jianqing Fan, Han Liu, Junwei Lu, and Ziwei Zhu. Distributed estimation and inference with statistical guarantees. arXiv preprint arXiv:1509.05457, 2015.

[5] Jan Beran. Statistics for long-memory processes. Routledge, 2017.

[6] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In European business intelligence summer school, pages 62–77. Springer, 2012.

[7] David Card. Using geographic variation in college proximity to estimate the return to schooling. Technical report, National Bureau of Economic Research, 1993.

[8] Marcus J Chambers. Long memory and aggregation in macroeconomic time series. International Economic Review, pages 1053–1072, 1998.

[9] Vasiliki Chatzikonstanti and Ioannis A Venetis. Long memory in log-range series: Do structural breaks matter? Journal of Empirical Finance, 33:104–113, 2015.

[10] Xueying Chen and Min-ge Xie. A split-and-conquer approach for analysis of extraordinarily large data. Statistica Sinica, pages 1655–1684, 2014.

[11] Rainer Dahlhaus. Efficient location and regression estimation for long range dependent regression models. The Annals of Statistics, pages 1029–1047, 1995.

[12] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1):107–113, 2008.

[13] Zhuanxin Ding, Clive WJ Granger, and Robert F Engle. A long memory property of stock market returns and a new model. Journal of empirical finance, 1(1):83–106, 1993.

[14] Tsai-Hung Fan, Dennis KJ Lin, and Kuang-Fu Cheng. Regression analysis for massive datasets. Data & Knowledge Engineering, 61(3):554–562, 2007.

[15] Wayne A Fuller. Measurement error models, volume 305. John Wiley & Sons, 2009.

[16] Clive WJ Granger. Long memory relationships and the aggregation of dynamic models. Journal of econometrics, 14(2):227–238, 1980.

[17] Clive WJ Granger and Roselyne Joyeux. An introduction to long-memory time series models and fractional differencing. Journal of time series analysis, 1(1):15–29, 1980.

[18] T Graves, RB Gramacy, Christian Franzke, and NW Watkins. Efficient bayesian inference for ARFIMA processes. Nonlinear Processes in Geophysics, 22:679–200, 2015.

[19] Timothy Graves, Robert Gramacy, Nicholas Watkins, and Christian Franzke. A brief history of long memory: Hurst, Mandelbrot and the road to ARFIMA, 1951–1980. Entropy, 19(9):437, 2017.

[20] Alexander Greaves-Tunnell and Zaid Harchaoui. A statistical investigation of long memory in language and music. arXiv preprint arXiv:1904.03834, 2019.

[21] Saptarshi Guha, Ryan Hafen, Jeremiah Rounds, Jin Xia, Jianfu Li, Bowei Xi, and William S Cleveland. Large complex data: divide and recombine (d&r) with rhipe. Stat, 1(1):53–67, 2012.

[22] Ryan Hafen. Divide and recombine: Approach for detailed analysis and visualization of large complex data. In Handbook of Big Data, pages 51–62. Chapman and Hall/CRC, 2016.

[23] Peter Hall, Soumendra Nath Lahiri, Jörg Polzehl, et al. On bandwidth choice in nonparametric regression with both short-and long-range dependent errors. The Annals of Statistics, 23(6):1921–1936, 1995.

[24] Seunghon Ham, Sunju Kim, Naroo Lee, Pilje Kim, Igchun Eom, Byoungcheun Lee, Perng-Jy Tsai, Kiyoung Lee, and Chungsik Yoon. Comparison of data analysis procedures for real-time nanoparticle sampling data using classical regression and ARIMA models. Journal of Applied Statistics, 44(4):685–699, 2017.

[25] Ngai Hang Chan and Wilfredo Palma. Estimation of long-memory time series models: A survey of different likelihood-based methods. In Econometric Analysis of Financial and Economic Time Series, pages 89–121. Emerald Group Publishing Limited, 2006.

[26] Jonathan RM Hosking. Fractional differencing. Biometrika, 68(1):165–176, 1981.

[27] Harold Edwin Hurst. Long-term storage capacity of reservoirs. Trans. Amer. Soc. Civil Eng., 116:770–799, 1951.

[28] Soosung Hwang. The effects of systematic sampling and temporal aggregation on discrete time long memory processes and their finite sample properties. Econometric Theory, 16(3):347–372, 2000.

[29] Pilar Iglesias, Hector Jorquera, and Wilfredo Palma. Data analysis using regression models with missing observations and long-memory: an application study. Computational statistics & data analysis, 50(8):2028–2043, 2006.

[30] S Claiborne Johnston, Tanya Henneman, Charles E McCulloch, and Mark Van der Laan. Modeling treatment effects on binary outcomes with grouped-treatment variables and individual covariates. American Journal of Epidemiology, 156(8):753–760, 2002.

[31] Michael I Jordan et al. On statistics, computation and scalability. Bernoulli, 19(4):1378–1390, 2013.

[32] Jee-Seon Kim and Edward W Frees. Omitted variables in multilevel models. Psychometrika, 71(4):659, 2006.

[33] Gary Koop, Eduardo Ley, Jacek Osiewalski, Mark FJ Steel, et al. Bayesian analysis of long memory and persistence using ARFIMA models. Journal of Econometrics, 76(1):149–170, 1997.

[34] Hira L Koul and Kanchan Mukherjee. Asymptotics of R-, MD-and LAD-estimators in linear regression models with long range dependent errors. Probability Theory and Related Fields, 95(4):535–553, 1993.

[35] Jason C Lau, WT Hung, David D Yuen, and CS Cheung. Long-memory characteristics of urban roadside air quality. Transportation research part D: Transport and Environment, 14(5):353–359, 2009.

[36] John Lee, Youngjo A. Nelder. Hierarchical generalized linear models. Journal of the Royal Statistical Society. Series B (Methodological), 58:619–678, 1996.

[37] Youngjo Lee, John A Nelder, and Yudi Pawitan. Generalized linear models with random effects: unified analysis via H-likelihood. Chapman and Hall/CRC, 2018.

[38] Haoqi Li, Huazhen Lin, Paul SF Yip, and Yuan Li. Estimating population size of heterogeneous populations with large data sets and a large number of parameters. Computational Statistics & Data Analysis, 139:34–44, 2019.

[39] L Li, M Hudson, J Ma, et al. Aggregation gain or loss? modelling the effects of group variables with binary responses.

[40] Runze Li, Dennis KJ Lin, and Bing Li. Statistical inference in massive data sets. Applied Stochastic Models in Business and Industry, 29(5):399–409, 2013.

[41] L.Ryan and A.Malecki. Additional analysis of ARTC data on particulate emissions in the rail corridor. Technical report, University of Technology Sydney, 2015.

[42] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. PloS one, 13(3), 2018.

[43] Bennett T McCallum et al. Relative asymptotic bias from errors of omission and measurement. Econometrica, 40(4):757–758, 1972.

[44] Angus Ian McLeod and Keith William Hipel. Preservation of the rescaled adjusted range: 1. a reassessment of the Hurst Phenomenon. Water Resources Research, 14(3):491–508, 1978.

[45] Alberto Montanari, Renzo Rosso, and Murad S Taqqu. Fractionally differenced ARIMA models applied to hydrologic time series: Identification, estimation, and simulation. Water resources research, 33(5):1035–1044, 1997.

[46] Jeh-Nan Pan and Su-Tsu Chen. Monitoring long-memory air quality data using ARFIMA model. Environmetrics: The official journal of the International Environmetrics Society, 19(2):209–219, 2008.

[47] Pierre Perron and Zhongjun Qu. Long-memory and level shifts in the volatility of stock market return indices. Journal of Business & Economic Statistics, 28(2):275–290, 2010.

[48] Amandine Pierrot and Yannig Goude. Short-term electricity load forecasting with generalized additive models. Proceedings of ISAP power, 2011, 2011.

[49] José Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, Siem Heisterkamp, Bert Van Willigen, and R Maintainer. Package 'nlme'. Linear and Nonlinear Mixed Effects Models, version, pages 3–1, 2017.

[50] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[51] Peter M Robinson, FJ Hidalgo, et al. Time series regression with long-range dependence. The Annals of Statistics, 25(1):77–104, 1997.

[52] L Ryan and M Wand. Re-analysis of ARTC data on particulate emissions from coal trains. On behalf of accessUTS Pty Ltd: Prepared for the NSW Environment Protection Authority, 2014.

[53] Louise M Ryan, Matt P Wand, and Alan A Malecki. Bringing coals to Newcastle. Significance, 13(6):32–37, 2016.

[54] Oliver Schabenberger and Francis J Pierce. Contemporary statistical models for the plant and soil sciences. CRC press, 2001.

[55] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, et al. The hadoop distributed file system. In MSST, volume 10, pages 1–10, 2010.

[56] Andrea Silvestrini and David Veredas. Temporal aggregation of univariate and multivariate time series models: a survey. Journal of Economic Surveys, 22(3):458–497, 2008.

[57] Fallaw Sowell. Maximum likelihood estimation of stationary univariate fractionally integrated time series models. Journal of econometrics, 53(1-3):165–188, 1992.

[58] Mark FJ Steel. Bayesian time series analysis. In Macroeconometrics and Time Series Analysis, pages 35–45. Springer, 2010.

[59] Daniel O Stram and William WS Wei. Temporal aggregation in the ARIMA process. Journal of Time Series Analysis, 7(4):279–292, 1986.

[60] George C Tiao. Asymptotic behaviour of temporal aggregates of time series. Biometrika, 59(3):525–531, 1972.

[61] Wen-wen Tung, Ashrith Barthur, Matthew C Bowers, Yuying Song, John Gerth, and William S Cleveland. Divide and recombine (d&r) data science projects for deep analysis of big data and high computational complexity. Japanese Journal of Statistics and Data Science, 1(1):139–156, 2018.

[62] Justin Q. Veenstra. Persistence and Anti-persistence: Theory and Software. PhD thesis, Western University, 2012.

[63] Michael R Wickens. A note on the use of proxy variables. Econometrica (pre-1986), 40(4):759, 1972.

[64] Hadley Wickham et al. The split-apply-combine strategy for data analysis. Journal of Statistical Software, 40(1):1–29, 2011.

[65] Simon Wood and Maintainer Simon Wood. Package ‘mgcv’. R package version, 1:29, 2015.

[66] Simon N Wood. Generalized additive models: an introduction with R. Chapman and Hall/CRC, 2017.

[67] Simon N Wood, Yannig Goude, and Simon Shaw. Generalized additive models for large data sets. Journal of the Royal Statistical Society: Series C (Applied Statistics), 64(1):139–155, 2015.

[68] Jeffrey M Wooldridge. Introductory econometrics: A modern approach. Nelson Education, 2015.

[69] Qifa Xu, Chao Cai, Cuixia Jiang, Fang Sun, and Xue Huang. Block average quantile regression for massive dataset. Statistical Papers, pages 1–25, 2017.

[70] Yoshihiro Yajima. On estimation of a regression model with long-memory stationary errors. The annals of Statistics, pages 791–807, 1988.

[71] Yoshihiro Yajima et al. Asymptotic properties of the LSE in a regression model with long-memory stationary errors. The Annals of Statistics, 19(1):158–177, 1991.

[72] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS), 23(4):550–560, 1997.