

Estimating the rebound effect for irrigation water in the Murray Darling Basin

Christopher Malone

Introduction

The management of water resources is of significant concern in Australian environmental policy. In particular, debate surrounds the socially optimal strategy for managing supplies of water for agricultural usage. Inland, water supply is subject to volatility, and the government must constrain usage to sustain the health of the natural environment. Competing strategies for curbing consumption include (a) governmental buybacks of water rights and (b) subsidies for construction of efficiency-improving technologies on farms and elsewhere. Given finite resources, the government must trade off between these strategies. Looming large in the debate about resource management strategy is the so-called *rebound effect* – the tendency for resource efficiency improvements to increase net use via the mechanism of price decreases.

In this paper, I attempt to approximate the rebound effect, by estimating the *elasticity* for irrigation water with respect to price in the Murray Darling Basin. In particular, I employ a Bayesian random effects regression analysis, with AR(1) error covariance structure. Some key findings are that (a) agricultural demand for water is relatively (but not entirely) inelastic, and (b) elasticity varies heavily with crop type. Results indicate that government should consider an industry-sensitive approach to water policy, based on relative concentrations of industries at distinct water catchments.

I begin the report by discussing some important background conceptual terrain, such as the notions of elasticity and the rebound effect, in minimal technical depth. I then move on to a discussion of the dataset and variables analysed in the current report. Then, I specify a Bayesian linear regression model. Results of the model are then discussed, along with diagnostic checks of the sampling algorithm and final model.

Background: The rebound effect and elasticity

A well known problem in environmental economics is to determine the magnitude of the “rebound effect” in a given domain. The “rebound effect” occurs when improvements to efficiency for use of a natural resource are offset by resulting increases in demand. For example, fuel-saving technology might be designed in order to curb the amount of fuel consumed by drivers. However, fuel-saving technology reduces the price of fuel, and so drivers drive more in response. Thus, the net effect of the efficiency improvement may be nullified by the resulting “rebound” in consumer demand.

The rebound effect has important implications for Australian water management policy. In particular, debate currently surrounds the management of water flows in the Murray Darling basin. The Murray Darling Basin provides water for roughly 40% of Australian agricultural produce (source: <https://www.mdba.gov.au/importance-murray-darling-basin>). A goal for the Australian government is to keep flow volume in the Murray Darling at sustainable levels, despite, on one hand, increasing demand from the growing agricultural industry, and, on the other hand, threats to supply from extreme climactic events, such as drought. Key strategies for conservation include (a) governmental buybacks (whereby water allocations are bought from the market and removed from use), and (b) efficiency improvement programs, which involve the government working with the private stakeholders to improve water efficiency on farms. The Australian government faces a trade-off between these strategies, under the constraint of finite resources for expenditure. Importantly, the larger the rebound effect is, the more we should prefer option (a) to option (b), as a program to maximise water

conservation. This is because, if the rebound effect is large, improvements to efficiency will be nullified by increased total usage (via the mechanism of decreased prices).

Closely related to the rebound effect for a given resource is the *elasticity* for that resource. In general, the elasticity for a good is defined as the change in quantity demanded of that good that is due to changes in its price per unit. If elasticity is high, for example, consumers will curb their usage greatly in response to price hikes. On the other hand, if it is low, they will change their usage habits much less. So elasticity is essentially a measure of responsiveness to price.

Elasticity is a good proxy for estimating the rebound effect. Firstly, elasticity scales with the rebound effect. This is because price sensitivity provides the most plausible proximate mechanism for the rebound effect. In other words, improvements to efficiency are liable to increase overall usage precisely because they lower the price of the relevant resource. Secondly, it is far easier to measure elasticity than to measure the rebound effect directly. This is because information about prices and market activity is more readily obtained than information about efficiency improvements. For these reasons, I am concerned primarily in this paper with the estimation of the elasticity for water.

The dataset

To estimate elasticity for water in agricultural usage, I have chosen to use the Murray Darling Basin water market dataset (source: <https://www.agriculture.gov.au/abares/research-topics/water/mdb-water-market-dataset>). This dataset is maintained for public use by the Australian Bureau of Agricultural and Resource Economics and Sciences (ABARES). Information in the dataset was primarily obtained primarily from the Australian Bureau of Statistics, before being cleaned and interpolated by ABARES.

The MDB water market dataset is comprised of two components, namely “supply” and “demand”. The “supply” dataset contains information on water prices and distributions of types of water entitlement. The “demand” dataset supplies information on levels of food production and value of total goods produced. Entries in the demand dataset each correspond to a region-crop-year triple. Regions here correspond to catchments - areas along the Murray-Darling river system where water collects naturally. Entries in the supply dataset, on the other hand, correspond to region-year pairs - they are not differentiated by crop type. Years for which measurements are included in each dataset range between roughly 2000 and 2020, with most of the data being concentrated in the 13 year interval 2006-2018.

Importantly, while “demand” information is provided for all MDB regions with relative consistency, “supply” information is provided for only roughly half of the total catchment regions. In particular, supply information is not provided for any region north of the Queensland border. As such, the analysis I present here only uses data from a subset of the total regions along the Murray Darling.

What we have, then, is panel (or longitudinal) data. Panel data is data comprised of a cross section of the population, with measurements taken at various time points. We denote the number of observations $n = N \times T$, where N is the number of individuals (panels), and T is the number of time points. We say that panels are “balanced” when we have the same number of time intervals for each of the N individuals.

So what are the panels in the data under consideration? For our data, it is convenient to think of each crop-region pair as a panel. For example, “almond farmers in NSW, South of the Murray river” is one panel; “fruit farmers in NSW, North of the Murray river” is another. Clearly, farmer-specific information would be preferable, but crop-region pairs are the finest-grain of information available, given the data.

Information available for the present study

As mentioned above, supply information is only given for roughly half of the catchment regions. Also, for many panels in the remaining data, time observations are scarce, or unevenly concentrated across time, beyond the range of 2006-2018. Given this, and for the convenience of working with balanced panels, I have restricted my analysis to the period 2006-2018. Also, for the convenience of using the AR(1) model to estimate capture serial correlation in the error terms, I have removed panels which are incomplete for this time-range.

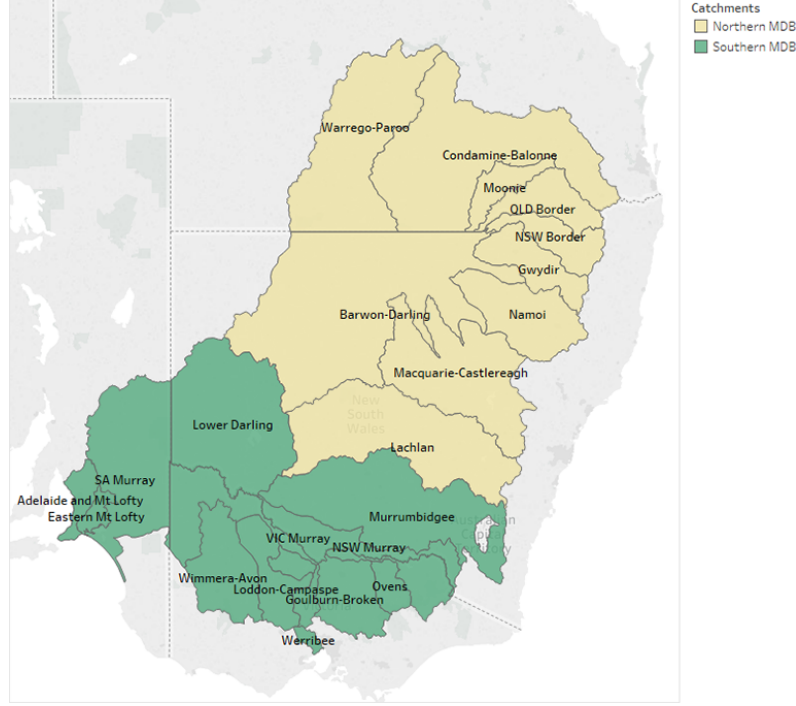


Figure 1: Map of MDB regions. Source: ABARES. See the link below.

After these removals, we are then left with $N = 94$ panels, each with a complete time series of length $T = 13$ (starting in 2006 and ending in 2018). So, while a significant portion of observations have been removed, the dataset remains large ($n = N \times T = 1222$). In future studies, it may be desirable to explore techniques of interpolating missing values. However, for many panels, there are simply no observations at all. Further, the exclusion of data points may be non-random in other ways. Thus missing data imputation may be unreliable.

Methodology

Variables

Here, I will outline and justify my choice of variables for the model fit. The dataset provides a large amount of variables to consider (the source, linked above, provides a total list, viewable in a web browser). I have, however, decided to fit a fairly simple model, in terms of variable choice. I use only the two independent variables that are most directly causally relevant to volume applied. A minimal number of independent variables was important here, as my modelling strategy involves modelling random slopes for each individual. My variable choices have largely been led by a priori considerations, though they have also been checked with prior modelling using frequentist OLS and GLS estimation.

For the response variable, I use the volume of water applied (measured in megalitres). I have chosen to log transform this variable as I found in prior modelling that it improved the quality of fit for all models. A priori, it is not surprising that this would be the case, as volume applied is strictly positive, and likely to be right skewed, as some industries will be particularly intensive in some regions, and data is not adjusted for volume of active farming taking place on nearby land.

The key explanatory variable is **price**. This variable is key because it is the goal of the study to better understand the relationship between price and volume of water applied. I also apply a log transform to **price**, for similar reasons as above, and to improve the ease of interpretation of the final model. Importantly, in log-log regression, the estimated coefficient can be directly interpreted as an estimate of elasticity. This makes the log transform a particularly attractive choice.

I also include the important confounding variable **rainfall**. A priori, we can expect rainfall to be negatively correlated with both water price and volume applied. When rainfall is high for a given year, a number of important changes occur. Firstly, farmers do not need to apply as much irrigated water as they would, if rainfall were low. So rainfall decreases volume applied directly. Secondly, when rainfall is high, catchment water levels rise, increasing the supply of water, which negatively affects the price. Given that rainfall can be expected to directly affect both price and volume applied, it is important that we adjust for it by including it as an auxiliary predictor variable.

Note on casual validity

Before moving on, it is important to note a crucial point about the causal validity of the present study. Though we are interested in the causal relationship between price and volume of water applied, we should be cautious about causal interpretations of the regression coefficients for the posterior model. This is because price, our key explanatory variable, is “endogenous” with respect to volume applied. In particular, there is a two-way (i.e. “simultaneous”) causal relationship between these two variables. Changes in price cause changes in volume applied, because farmers are cost-averse. Simultaneously, changes in volume of water applied cause changes in price. This is because application of water reduces the available supply. Given this complex causal relationship, the estimated coefficient of price should be treated with caution and skepticism. In future studies, it may be desirable to harness causal inference techniques, such as (e.g.) instrumental variable analysis.

The model

To explore the question discussed above, I have chosen to employ a Bayesian multilevel regression model, with AR(1) variance-covariance structure.

The first important feature of the model is that it is multilevel (or hierarchical), because regression coefficients are considered to be individual-specific, but drawn from a shared distribution. So, in estimating the model, we estimate the individual specific and shared distributions simultaneously. The main benefit of the hierarchical approach here is that it will allow us to understand some of the heterogeneity in elasticity between crop types and regions, while allowing us to pool information between all of the individuals. This is particularly desirable for the data under consideration, as we only have a small number of observations ($T = 13$) for each individual. Beyond this benefit, the assumption of random coefficients seems appropriate here, as we have access to only a subset of the whole population, with respect to regions and crop types.

The second important feature of the model is that errors are not considered independent. Preliminary modelling led me to believe that there was a significant degree of time dependence in the residuals. To control for this, I have adapted the “Bayesian GLS” technique with AR(1) errors, recommended in Hoff’s textbook.

Notably, I have adapted the technique to account for multiple simultaneous time series, with the assumption of zero correlation between them. To see what I mean, consider the following. Consider the standard AR(1) model for time series, for a time series of length $T = 3$. This can be thought of as a panel dataset, where $N = 1$. If Σ , the variance covariance matrix of the error terms, has an AR(1) structure, then it is given by

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix}$$

where ρ is the coefficient of correlation between adjacent time periods, and σ^2 is the variance of each $y_i|X\beta$.

Extending the model to panel data is simple. Take the case where we have $N = 2$ individuals and a time series of length $T = 3$ for each of them. Further, assume that errors between individuals are independent. Then, the variance covariance matrix of the errors is just given by

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & 0 & 0 & 0 \\ \rho & 1 & \rho & 0 & 0 & 0 \\ \rho^2 & \rho & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \rho & \rho^2 \\ 0 & 0 & 0 & \rho & 1 & \rho \\ 0 & 0 & 0 & \rho^2 & \rho & 1 \end{pmatrix}$$

Note that each observation from the first panel is assumed not to correlate with any observations in the second panel. On the other hand, *within* a given panel, observations are correlated, with correlation coefficient ρ describing the dependence relationship.

As the reader will recall, the present panel has $N = 94$, $T = 13$, and $n = N \times T = 1222$. So the covariance matrix is an $n \times n = 1222 \times 1222$ matrix, which has $N = 94$ matrices of size $T \times T = 13 \times 13$ “down its diagonal”, and zeroes elsewhere.

It is important to note that my model assumes that observations between panels at identical time periods are not correlated. Clearly, this assumption is not entirely plausible. In particular, we should expect panels that are geographically clustered or share a crop type to have correlated prices at shared time intervals. In future analysis, it would be desirable to explore spatial models, where parameters for spatial correlation between time series are estimated as well as temporal correlation parameters. This may be achievable simply by augmenting the structure of the variance covariance matrix Σ . I refrain from attempting this here, due to constrained time resources.

The model

Now that I have discussed the broad outline of the model, I will move on to discuss the full formal specification of the model.

Sampling model and priors

Here, I assume the prior distributions proposed by Hoff (see last example in ch. 10, first example in ch. 11). The sampling model is given as follows

$$\mathbf{Y}_j = \begin{pmatrix} Y_{1,j} \\ \vdots \\ Y_{n,j} \end{pmatrix} \sim \text{multivariate-normal}(\mathbf{X}_j \boldsymbol{\beta}_j, \sigma^2 \mathbf{C}_\rho) \quad (1)$$

where \mathbf{C}_ρ is the correlation matrix, the structure of which is described in the preceding section. In turn, the $\boldsymbol{\beta}_j$ terms have the following distribution.

$$\boldsymbol{\beta}_j \stackrel{iid}{\sim} \text{multivariate-normal}(\boldsymbol{\theta}, \Sigma)$$

In other words, I assume that `log(price)` observations, conditional on the values for `log(price)` and `rainfall` are drawn from a multivariate normal distribution, with a vector of means which depend on regression coefficients. In turn, these regression coefficients are themselves drawn from a multivariate normal distribution.

Prior distributions

The prior distributions I assume for each parameter are as follows. (I use the prior distributions and hyperparameters suggested by Hoff in chapters 10 and 11.) For ρ , the prior is chosen out of convenience with working with the standard uniform density. Other priors are chosen as their full conditional posterior distributions are known. Terms subscripted by 0 are known constants, which are assigned below.

$$\begin{aligned}
\sigma^2 &\sim \text{inverse-gamma}(\nu_0/2, \nu_0\sigma_0^2/2) \\
\boldsymbol{\theta} &\sim \text{multivariate-normal}(\boldsymbol{\mu}_0, \Lambda_0) \\
\Sigma &\sim \text{inverse-Wishart}(\eta_0, \mathbf{S}_0^{-1}) \\
\rho &\sim \text{uniform}(0, 1)
\end{aligned}$$

Prior hyperparameters

First, I set $\nu_0 = \sigma_0^2 = 1$, so that the prior on σ^2 relatively diffuse. Next, I set the mean vector for the prior distribution on $\boldsymbol{\theta}$ to be $\mu_{0,k} = \hat{\beta}_k^{\text{OLS}}$, $k \in \{0, 1, 2\}$, where β_0 is the intercept; β_1 and β_2 are the regression coefficients associated with `log(price)` and `rainfall`, respectively; and $\hat{\beta}_k^{\text{OLS}}$ is the OLS estimate for β_k . Similarly, I set Λ_0 to be the sample variance-covariance matrix for the β terms from the OLS fit. Finally, I set $\mathbf{S}_0^{-1} = \Lambda_0^{-1}$, and $\eta_0 = p + 2 = 5$, so that the prior mean of Σ is the sample covariance of the OLS estimates of the β_k 's.

Posteriors

The above prior distribution assumptions result in the following set of posteriors. Note that the analytic form of the posterior distribution for ρ is not known. For this reason, I estimate moments of the posterior using the Metropolis method.

$$\{\boldsymbol{\theta} \mid \beta_1, \dots, \beta_N, \Sigma\} \sim \text{multivariate-normal}(\boldsymbol{\mu}_N, \Lambda_N), \text{ where}$$

$$\begin{aligned}
\Lambda_N &= (\Lambda_0^{-1} + N\Sigma^{-1})^{-1} \\
\boldsymbol{\mu}_N &= \Lambda_N (\Lambda_0^{-1} \boldsymbol{\mu}_0 + N\Sigma^{-1} \bar{\boldsymbol{\beta}}) \\
\bar{\boldsymbol{\beta}} &= \frac{1}{N} \sum_{j=1}^N \boldsymbol{\beta}_j
\end{aligned}$$

$$\{\Sigma \mid \boldsymbol{\theta}, \beta_1, \dots, \beta_N\} \sim \text{inverse-Wishart}(\eta_0 + N, [\mathbf{S}_0 + \mathbf{S}_\theta]^{-1}), \text{ where}$$

$$\mathbf{S}_\theta = \sum_{j=1}^N (\beta_j - \boldsymbol{\theta})(\beta_j - \boldsymbol{\theta})^T$$

$$\{\beta_j \mid \mathbf{y}_j, \mathbf{X}_j, \boldsymbol{\theta}, \Sigma, \sigma^2, \rho\} \sim \text{multivariate-normal}(\beta_n, \Sigma_n), \text{ where}$$

$$\begin{aligned}
\Sigma_n &= (\mathbf{X}_j^T \mathbf{C}_\rho^{-1} \mathbf{X}_j / \sigma^2 + \Sigma^{-1})^{-1} \\
\beta_n &= \Sigma_n (\mathbf{X}_j^T \mathbf{C}_\rho^{-1} \mathbf{y}_j / \sigma^2 + \Sigma^{-1} \beta_0), \text{ and}
\end{aligned}$$

$$\{\sigma^2 \mid \mathbf{X}, \mathbf{y}, \beta, \rho\} \sim \text{inverse-gamma}([\nu_0 + n]/2, [\nu_0\sigma_0^2 + \text{SSR}_\rho]/2), \text{ where}$$

$$\text{SSR}_\rho = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{C}_\rho^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Note that the β_j 's are sufficient for $\boldsymbol{\theta}$ and Σ . That is, their prior distributions depend only on ρ through each β_j .

The algorithm

In the previous section, I detailed the full specification for the Bayesian hierarchical regression model. I now move on to describing the steps of the MCMC sampling algorithm used to obtain posterior moments for each

parameter. For ρ , the full conditional distribution is not known, and so Metropolis-Hastings sampling must be used. (I use the uniform, i.e. “reflected random walk” jump distribution, with $\delta = 0.05$.) But for the other parameters, the full conditional distribution *is* known, so we can use Gibbs sampling.

First, we start out by setting starting values for each of the parameters. In particular, I set $\rho = 0.7$, close to the posterior mean, to hasten convergence. For the β_j terms (where β_j is a vector of 3 regression coefficients) I have used OLS estimates for the individual time series for each panel. For θ , I have used the means of the starting values for the β_j ’s, averaged over all the OLS models. For Σ , I have used the sample variance-covariance matrix of the OLS β_j ’s. Finally, for σ^2 , I have used the MSE from an OLS fit to the full dataset (with one single set of coefficients).

Steps of the algorithm are as follows (adapted from Hoff). Prior and posterior hyperparameters (subscripted by 0 and m) are defined in the preceding sections.

1. Update β : Sample $\beta^{(s+1)} \sim \text{multivariate-normal}(\beta_n, \Sigma_n)$, where β_n and Σ_n depend on $\sigma^{2(s)}$ and $\rho^{(s)}$.
2. Update θ : Sample $\theta^{(s+1)} \sim \text{multivariate-normal}(\mu_N, \Lambda_N)$, where μ_N depends on $\beta^{(s+1)}$.
3. Update Σ : Sample $\Sigma^{(s+1)} \sim \text{inverse-Wishart}(\eta_0, \mathbf{S}_0^{-1})$, where \mathbf{S}_θ depends on $\beta^{(s+1)}$ and $\theta^{(s+1)}$.
4. Update σ^2 : Sample $\sigma^{2(s+1)} \sim \text{inverse-gamma}([\nu_0 + n]/2, [\nu_0 \sigma_0^2 + \text{SSR}_\rho]/2)$. where SSR_ρ depends on $\beta^{(s+1)}$ and $\rho^{(s)}$.
5. Update ρ :
 - a) Propose $\rho^* \sim \text{uniform}(\rho^{(s)} - \delta, \rho^{(s)} + \delta)$, $\delta = 0.05$. If $\rho^* < 0$ then reassign it to be $|\rho^*|$. If $\rho^* > 1$ reassign it to be $2 - \rho^*$.
 - b) Compute the acceptance ratio

$$r = \frac{p(\mathbf{y} | \mathbf{X}, \beta^{(s+1)}, \sigma^{2(s+1)}, \rho^*) p(\rho^*)}{p(\mathbf{y} | \mathbf{X}, \beta^{(s+1)}, \sigma^{2(s+1)}, \rho^{(s)}) p(\rho^{(s)})}$$

I have run the algorithm for 100,000 iterations, and thinned the sample by taking only every 10th draw for each parameter. Such a large number of iterations may seem excessive. However, earlier experimentation led to “effective sample sizes” that were smaller than $n = 1222$ for some parameters, even with large numbers of iterations. This is likely due to a combination of (a) the number of parameters to be estimated being large, and (b) the sample size being large (setting a high bar for the effective MCMC sample sizes).

As for a burn-in period, I have decided not to employ one for any parameter estimated. Checking of individual trace plots (which I cannot show here due to space restrictions) shows that convergence is reached for all parameters almost immediately. This is the case in part because I have chosen starting values close to the mean of the posterior distribution, and in part because I have thinned the collection posterior draws, so that autocorrelation in the initial run is less of an issue than it would otherwise be.

With respect to code authorship and source, I have used a combination of (a) code given in lecture exercises (namely the exercises for the ice data in chapter 10 and the math test example in chapter 11); (b) my own code written for the second online test (namely an empty template for MCMC sampling from data manipulation through to sampling); and (c) my own original code, written with the current task in mind. My largest original contribution has been to adapt the AR(1) modelling technique to panel data, where we observe multiple time series, with observed dependence within (but not between) groups.

Results

I now turn to a discussion of the results. The most important parameter estimated is the mean of the θ term associated with $\log(\text{price})$. Thanks to the fact that the model has a logged response and logged explanatory variable, the elasticity is just this estimated coefficient.

The plot of the density for this θ term is given in figure 2. As we can see, the curve is concentrated at around -0.175. More precisely, the estimated mean is -0.1671058. Taking this number as our supposed overall mean elasticity, we can infer that, on average, a 1% increase in price should lead to a $\sim 0.17\%$ decrease in quantity of volume applied for irrigation purposes. (Though, please keep in mind the note on causal validity given above.) Further, as figure 2 shows, under our model, it is highly unlikely that the overall mean elasticity is equal to zero. Note, however, that this does not imply that any given β_j sampled from $MVN(\theta, \Sigma)$ (i.e. the elasticity for any given crop-region pair j) is highly unlikely to be 0. The β_j terms have their own densities, which may or may not be close to 0.

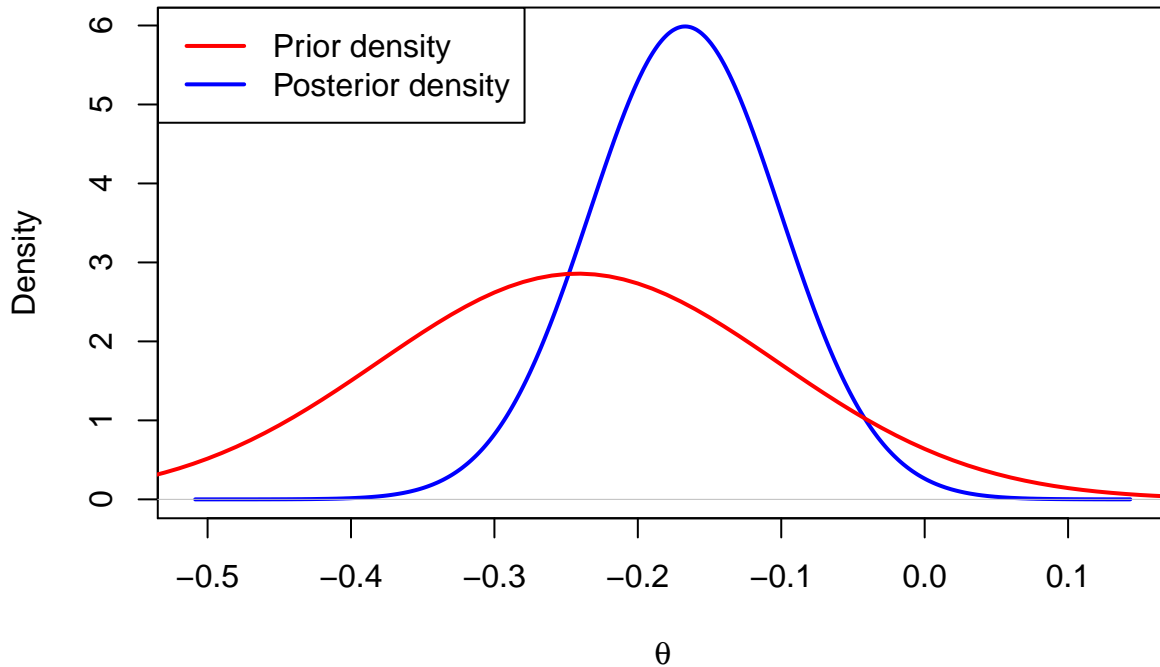


Figure 2: Posterior density of mean of coefficients associated with price

Presenting the results of the model are difficult, given the large number of elasticities estimated. However, such plots may be misleading in this case. This is because not all crops are represented for each of the regions in our sample, and vice versa. For example, consider the case where, say, almond farmers have unusually high elasticity for irrigated water, and cotton farmers have unusually low elasticity. Further, suppose that, in the original data, regions A and B each had information for both almond and cotton farming. However, almond data was removed from region A, and cotton data from region B. Then, it will likely appear as if overall elasticity is significantly larger for B than for A, when this may not be the case in reality.

Also of primary interest is the heterogeneity in elasticities, which multilevel modelling has allowed us to estimate. Table 1 displays the means of the posterior distributions of elasticities – i.e. coefficients for $\log(\text{price})$. Rows represent catchment regions in the sample (a subset of the full population of catchment regions), and columns represent crop types.

There are some interesting differences in elasticities at the level of crop type. Notably, under the model, elasticity appears to be particularly high for cotton, rice, grazing pastures, and miscellaneous broadacre crops. (Broadacre crops include grains and oilseeds, such as wheat and sunflower.) It is significantly lower for fruits, vegetables, and grapevines. The former set (particularly rice and cotton) are generally recognised as being water-intensive crops. So we might infer that industries for more water-intensive produce are more responsive to changes in water price.

In contrast to the case for crop type, elasticities do not appear to be differentiated as much by region. However, there are a few interesting exceptions. The NSW portion of the Lower Darling (row #3) has an unusually high number of positive coefficients. Under our model, these indicate negative elasticities. When

elasticity is negative, *more* water is applied when price increases. In particular, the model appears to suggest that, for a 1% increase in price, water applied for hay production in the Lower Darling will *increase* by 1.37%.

High negative elastic is fairly unusual behaviour. Inspection of the data reveals that the presence of this is likely due to unusually low values of water volume applied for this region-crop pair. In particular, the first three observations for this time series are extremely close to zero, while the mean for the whole dataset is very far from zero, at 45546.12. These three small values are extremely unusual in the dataset, and are most likely due to a data entry error. Still, it is worrying that we are given such a strange result due to only three pathological data point. This may indicate that my chosen model overfits the data, perhaps by fitting too many parameters.

Table 1: Table of mean estimated elasticities by crop type and region.

	Cotton	Grapevines	Almonds	Fruit	Grazing	Hay
NSW Gwydir	-0.17					
NSW Lachlan	-0.305	-0.004				
NSW Lower Darling		-0.152	0.027	-0.127	0.161	1.37
NSW Macquarie-Castlereagh	-0.437	-0.114				
NSW Murray Above	-0.113	-0.083		-0.06	-0.305	-0.208
NSW Murray Below		-0.127	-0.206	-0.078	-0.297	-0.198
NSW Murrumbidgee	-0.256	-0.138	0.046	-0.104	-0.289	-0.058
NSW Namoi	-0.247					
SA Murray		-0.162	-0.128	-0.123	-0.187	-0.081
VIC Goulburn-Broken		-0.051		-0.113	-0.374	-0.15
VIC Loddon-Campaspe		0.071		-0.191	-0.242	-0.234
VIC Murray Above		-0.176		-0.113	-0.286	-0.086
VIC Murray Below		-0.131	-0.183	-0.108	-0.183	-0.162

	Rice	Dairy	Vegetables	Misc broadacre	Misc cereal	Other
NSW Gwydir				-0.563		
NSW Lachlan					-0.004	
NSW Lower Darling			-0.07		-0.454	-0.263
NSW Macquarie-Castlereagh						
NSW Murray Above	-0.797	-0.16	-0.116	-0.312	-0.188	0.129
NSW Murray Below	-0.592	-0.249	0.106	-0.281	-0.138	
NSW Murrumbidgee	-0.634	-0.195	-0.036	-0.236	-0.179	-0.202
NSW Namoi				-0.246		
SA Murray		-0.06	-0.101	-0.485		0.02
VIC Goulburn-Broken		-0.311	-0.1		-0.266	0.029
VIC Loddon-Campaspe		-0.28	-0.158		-0.206	-0.516
VIC Murray Above		-0.142	-0.023	-0.317	-0.094	-0.627
VIC Murray Below		-0.273	-0.038	-0.111	-0.018	-0.083

Finally, the reader will recall that we also fit a lag coefficient ρ to capture any potential dependence in the error terms of the regression model. A density plot for the posterior distribution of this coefficient is displayed below. The density is peaked strongly at the mean of ~ 0.681 , and becomes extremely small below 0.55. This indicates fairly strong dependence in the error terms.

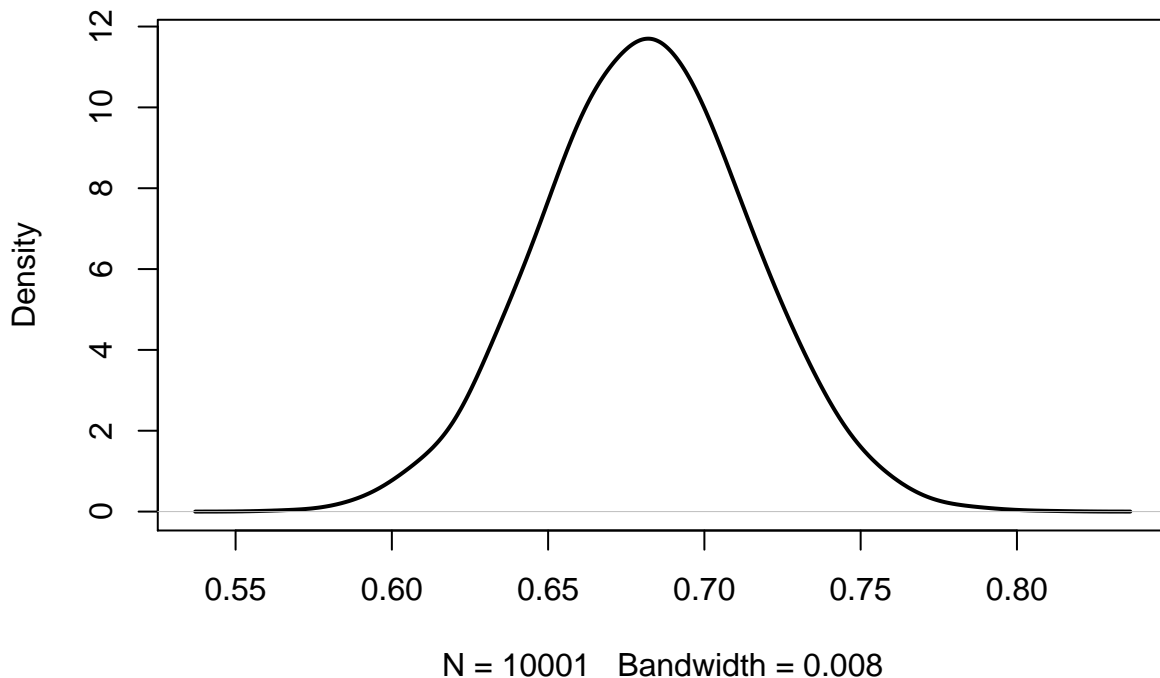


Figure 3: Posterior density of the correlation coefficient

Model checking

Here, I present MCMC convergence diagnostics, as well as regular regression diagnostics, and, finally, a posterior predictive check of the model. First, MCMC convergence diagnostics are comfortably satisfied for all parameters in the model. This is thanks to a large number of iterations and a relatively wide thinning interval. Second, regression diagnostics show that important model assumptions (notably constant variance and normality of errors) are unfortunately not reasonably satisfied. Third, a model check shows that the observed sample mean of the y values is fairly probably under our model, vindicating the model somewhat. I turn to a more detailed discussion of these issues now.

Convergence diagnostics

It is difficult to comprehensively show convergence diagnostics here, as the number of parameters estimated is very large. For my analysis, I have estimated 287 parameters. This was mainly caused by my choice to model all regression coefficients hierarchically (see motivation above in the section on model specification). So, while I have checked plots individually for convergence, I cannot show them here. Instead, I will present (1) figures displaying effective sizes, (2) acf plots for the variables with the parameters with the smallest effective sizes, and (3) the Metropolis-Hastings acceptance rate for the parameter ρ .

First, we can look at effective sizes to get an impression of the relative stationarity of the sampling algorithm for each parameter. Figure 4 shows a barplot of effective sizes for each of the 287 parameters. Effective sample sizes are all quite large, with most of them being at or near 10000 (the size of the thinned MCMC sample). The minimum effective sample size is just below 4000, less than half of the full set of draws – though it is comfortably larger than the sample size $n = 1222$. Looking closely at this graphic, we can see that the rightmost parameters have the lowest effective sizes. These two bars correspond to the parameters ρ and σ^2 , respectively.

To investigate ρ and σ^2 , we can observe the ACF plot shown in figure 5. As we can see, values damp towards zero fairly quickly. This gives us peace of mind, that the distribution is relatively stationary, and that we are justified in treating our MCMC samples of these parameters as collections of independent draws.

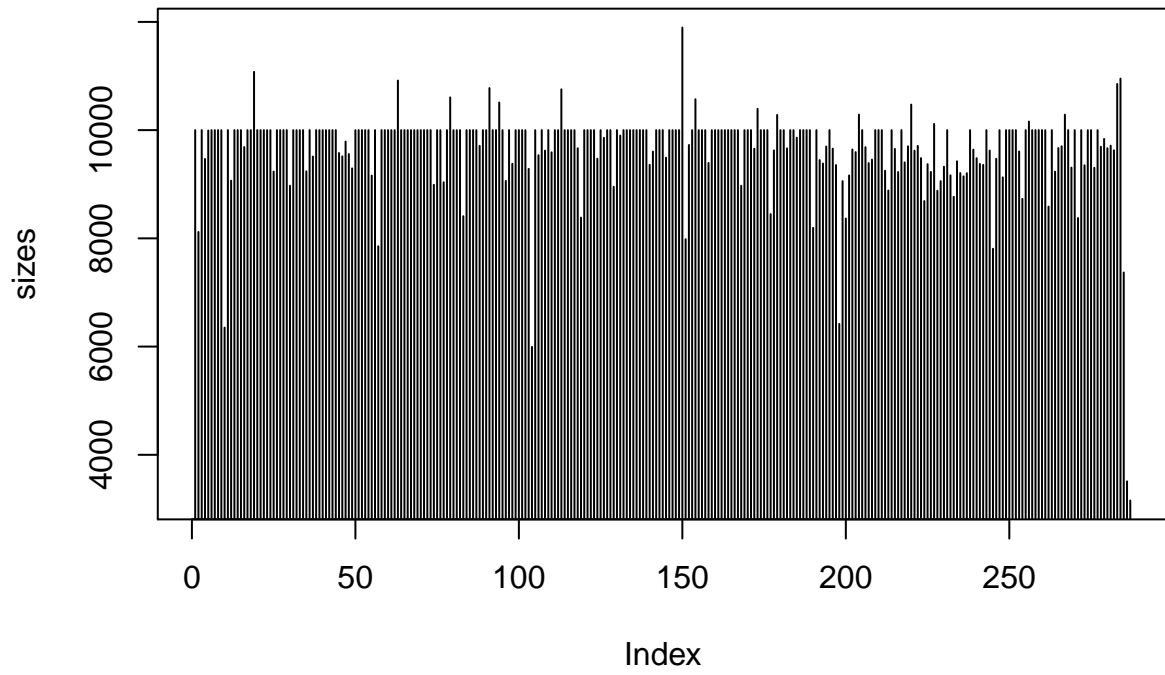


Figure 4: Effective sizes for each posterior distribution estimated

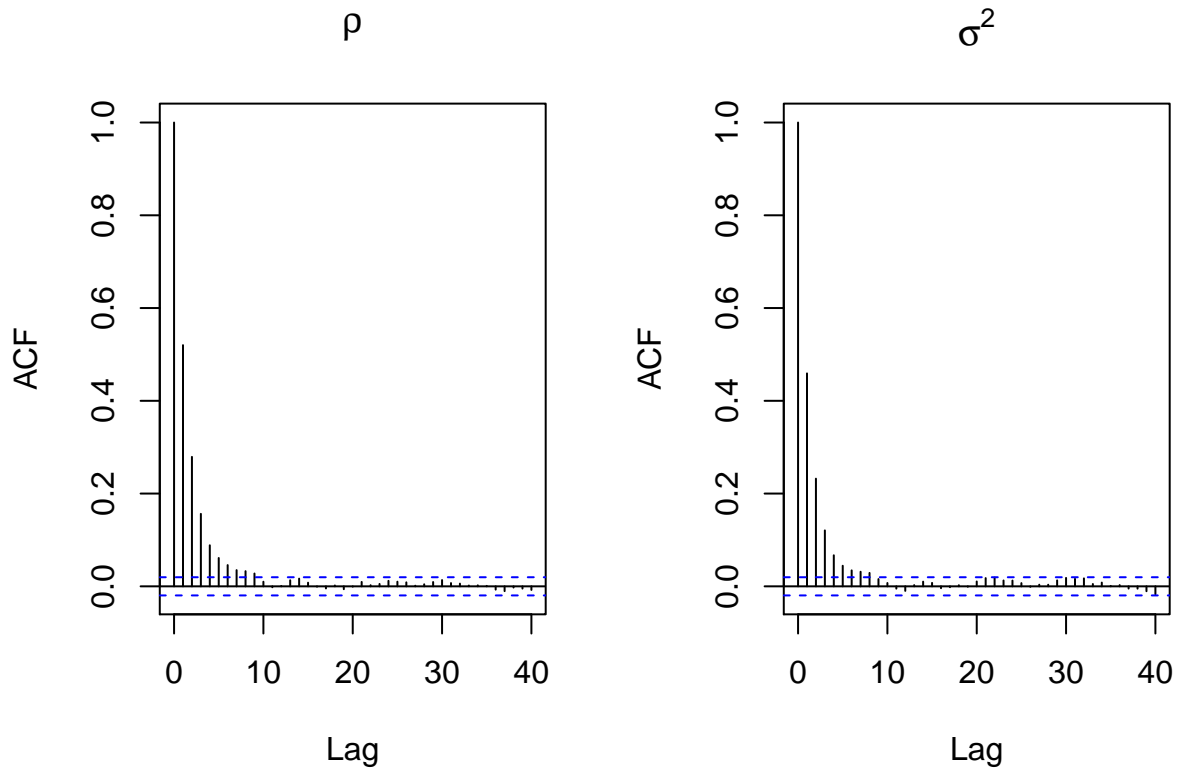


Figure 5: Autocorrelation for the temporal correlation coefficient and variance of y conditional on X .

Finally, I report the acceptance rate for ρ . With jump parameter $\delta = 0.05$, an acceptance rate of 0.40579 (over 100000 draws) was observed. This is comfortably within the desirable range of 0.2-0.5. Thus, we can be satisfied that the jump parameter δ is appropriate, with respect to efficiency and maximal reduction of dependence between draws.

Regression diagnostics

I now turn to a discussion of regression diagnostics. Figure 6 shows a collection regression diagnostic graphics. Unfortunately, the data do not appear to satisfy core assumptions of the model. Looking at the plot of residuals vs. fitted values, we see that variance appears to decrease with the $\log(\text{volume})$. So the assumption of constant variance is violated. This may be due to the log transformation being too strong for the given model.

Further, there is a slight, but distinct, linear trend in the residuals. This may be due to the “endogeneity” of the price variable, discussed above). As I noted above, this may suggest a model with greater causal validity, such as a model that uses regressors instrumental variables on `price` or `log(price)`.

Observing the normal quantile quantile plot, we see that the residuals also have a very heavy tailed distribution, indicating a near-certain violation of the normality assumption. This indicates that we may need to be cautious in making inferences based on our posterior distribution on y , which assumes that it is normally distributed, given X .

Finally, there is at least some good news, as the AR(1) structure for the covariance matrix appears to have captured most (though certainly not all) dependence in the residuals.

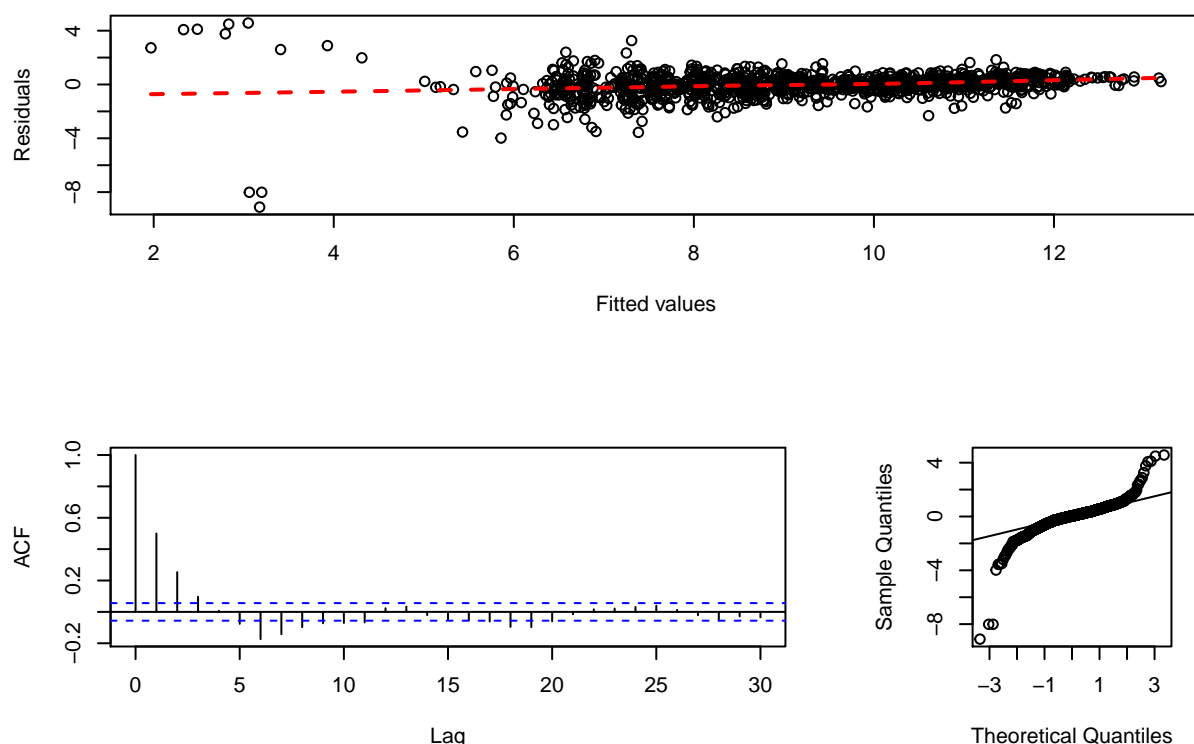


Figure 6: Regression diagnostics. Top: Residuals vs. fitted values. Bottom left: residual autocorrelation. Bottom right: Normal quantile-quantile plot of residuals.

Overall, then, regression diagnostics indicate that the model is misspecified. To improve on this, it may be worth experimenting with different distributions for y (such as a t-distribution), and using different transformations of dependent and independent variables, and/or using causal inference techniques, such as instrumental variable analysis.

To round off this section, we can perform a posterior predictive check, to see how likely the observed data are under our model. This is shown in figure 7. This plot is a histogram of 10,000 posterior means of $\log(y)|X$. The red line gives the observed sample mean of $\log(y)$. The line is fairly close to the centre of the mound. So the plot gives little further reason to doubt the model.

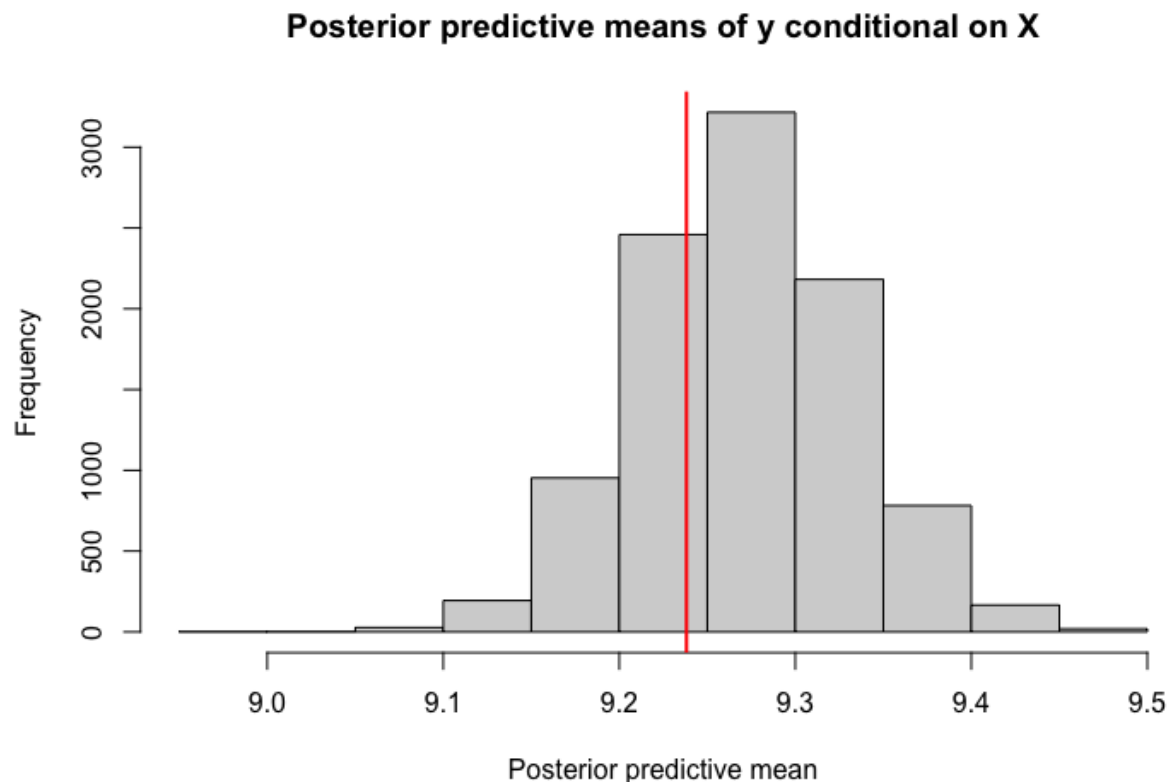


Figure 7: Posterior predictive check.

Conclusions

The core results of this analysis are that (a) demand for water is relatively inelastic, but significantly different from zero, under the model detailed; and (b) elasticity for water appears to vary significantly between crop-region pairs, with patterns emerging in particular at the level of crop type.

I am most proud of the fact that I extended the AR(1) model to deal with panel data, and modelled slopes hierarchically. The strongest point of the analysis is that it uses hierarchical modelling to understand variability in elasticities.

The major weakness of the analysis are that the results are not causally valid, given endogeneity of the explanatory variable (see the note on causal validity above). Other weaknesses include (a) the lack of a spatial component to control for spatial correlation between panels, and (b) the fact that elasticities are relative to crop-region interactions, rather than crops and regions separately. The latter makes it difficult to generalise about the effects of certain crops or regions on elasticities or the response variable.

Given more time, I would like to have addressed the issues just discussed. First, I would like to explore causal inference techniques, like instrumental variable analysis. In particular, the dataset contains information on government buybacks, which would be very useful in this regard (they change volume applied only through the mechanism of price changes). Second, I would like to have added a spatial component to the error

covariance matrix, to deal with the potential for spatial dependence in the response variable. This would be fairly simple once spatial distances had been somehow encoded. However, it may involve computationally expensive computation of large matrices, when drawing from σ^2 .

Code appendix

The code appendix is in two parts. The first part contains the sampling algorithm. The second contains all code used to construct the report (mainly graphics and some data processing).

1. Sampling algorithm (from a separate R file)

```
# Clear the environment
rm(list = ls())

# Get useful libraries
require(mvtnorm)
require(timeR)

##### LOAD IN THE DATA
murray = read.csv('murray_dummies.csv')

# Design matrix
intercept = rep(1, nrow(murray))
X = as.matrix(cbind(intercept, murray[, 3:4]))
X[, 2] = log(murray[, 3])
y = log(murray$vol)

# Number of time periods
t = 2018 - 2005

# Number of panels
N = dim(murray)[1] / t

# Number of observations
n = t * N

# Number of betas to estimate
m = ncol(X)
p = 3

##### SET STARTING VALUES
OLS = lm(y ~ - 1 + as.matrix(X))
sigma2 = summary(OLS)$sigma^2

# rho = acf(OLS$residuals, plot = F, )$acf[2]

# The ACF of the OLS residuals is ~0.84, but my model
# estimates the correlation coefficient to be closer to 0.7.
# So I set rho at 0.7, to hasten convergence.
rho = 0.7

Beta = matrix(0, nrow = N, ncol = p)
for (j in 1:N) {
  index = ((j-1) * t + 1) : (t * j)
  Beta[j, ] = lm(log(vol) ~ log(price) + rainfall, murray[index,])$coefficients
}

Sigma = cov(Beta)
```

```

inv.Sigma = solve(Sigma)
Theta = apply(Beta, 2, mean)

# Initialise the correlation matrix (and take inverse)
DY = abs(outer((1:t), (1:t), "-"))
corrs = rho^DY
inv.corrs = solve(corrs)

##### SET PRIOR HYPERPARAMETERS
mu.0 = Theta
Lambda.0 = cov(Beta)
inv.Lambda.0 = solve(Lambda.0)
nu.0 = 1
sigma2.0 = 1
S.0 = cov(Beta)
eta.0 = p + 2

# Get some useful numbers here to minimise redundant
# computation inside the algorithm.
a = (nu.0 + n) / 2
f.of.b = nu.0 * sigma2.0
zero.matrix = matrix(0, nrow = N * t, ncol = N * t)

##### SET S
S = 100000
thin = 10

#### SET TUNING/JUMP PARAMETER FOR RHO
delta.rho = 0.05

### CREATE STORAGE OBJECTS
store.intercept = matrix(nrow = S / thin, ncol = N)
store.price = matrix(nrow = S / thin, ncol = N)
store.rainfall = matrix(nrow = S / thin, ncol = N)
store.theta = matrix(nrow = S / thin, ncol = p)
store.rho = NULL
store.sigma2 = NULL
acc.rate.rho = 0

# Set a timer to measure performance
timer = createTimer()

{
timer$start("event1")

### START ALGORITHM
for (s in 1:S) {

  # Gibbs for betas
  for (j in 1:N) {
    index = ((j-1) * t + 1) : (t * j)
    X.j = X[index, ]
    y.j = y[index]

```



```

Sigma.n.j = solve((t(X.j) %*% inv.corr %*% X.j) / sigma2 + inv.Sigma)

mu.n.j = Sigma.n.j %*% (((t(X.j) %*% inv.corr %*% y.j) / sigma2) + nv.Sigma %*% (Theta)))

# Update Beta
Beta[j, ] = (rmvnorm(1, mu.n.j, Sigma.n.j))
}

# Gibbs for Theta (the means of the betas)
Lambda.N = solve(inv.Lambda.0 + N * inv.Sigma)
beta.mean = apply(Beta, 2, mean)
mu.N = Lambda.N %*% (inv.Lambda.0 %*% mu.0 + N * inv.Sigma %*% beta.mean)

# Update Theta
Theta = t(rmvnorm(1, mu.N, Lambda.N))

# Gibbs for Sigma (the variance-covariance matrix of the
# betas)
Theta.matrix = matrix(Theta, N, p, byrow = T)
S.theta = t(Beta - Theta.matrix) %*% (Beta - Theta.matrix)

# Update Sigma (inverted)
inv.Sigma = matrix(rWishart(1, eta.0 + N, solve(S.0 + S.theta)), 3, 3)

# Gibbs for sigma2 (the variance of the y_i's)
SSR.rho = 0

# Calculate rho by iterating over the groups/panels
for (j in 1:N) {
  index = ((j-1) * t + 1) : (t * j)

  y_j.minus.XB_j = y[index] - (X[index, ] %*% (Beta[j, ]))
  SSR.rho = SSR.rho + t(y_j.minus.XB_j) %*% inv.corr %*% y_j.minus.XB_j
}

# Update sigma2
sigma2 = 1 / rgamma(1, a, (f.of.b + SSR.rho) / 2)

# Metropolis for rho

# Get the vector X %*% beta, by iterating over
# groups/panels
XB = NULL
for (j in 1:N) {
  index = ((j-1) * t + 1) : (t * j)
  X.j.B.j = X[index, ] %*% Beta[j, ]
  XB = c(XB, X.j.B.j)
}

rho.prop = abs(runif(1, rho - delta.rho, rho + delta.rho))
rho.prop = min(rho.prop, 2 - rho.prop)

```

```

C_p.prop = sigma2 * (rho.prop^DY)
C_p.old = sigma2 * (corrs)

numerator <- denominator <- 0
for (j in 1:N) {
  index = ((j-1) * t + 1) : (t * j)

  numerator = numerator + dmvnorm(y[index], XB[index], C_p.prop, log = T)
  denominator = denominator + dmvnorm(y[index], XB[index], C_p.old, log = T)
}

r.rho.log = numerator - denominator

if (runif(1) < exp(r.rho.log)) {
  rho = rho.prop
  acc.rate.rho = acc.rate.rho + 1

  # Perform matrix computation in here so that we only
  # compute new matrices (expensive) when we accept a new rho
  corrs = rho^DY
  inv.corrs = solve(corrs)
}

# Print the current iteration number to the console, for
# monitoring purposes
if (s %% thin == 1) {
  print(s)

  # Store all drawn parameters from this iteration
  # Each column corresponds to a group
  store.intercept[s / thin, ] = Beta[, 1]
  store.price[s / thin, ] = Beta[, 2]
  store.rainfall[s / thin, ] = Beta[, 3]
  store.theta[s / thin, ] = t(Theta)
  store.sigma2 = c(store.sigma2, sigma2)
  store.rho = c(store.rho, rho)
}
}

# Stop the timer
timer$stop("event1")

# Hit the alarm so I know it's done
beepr::beep()
}

```

2. Graphics (from the Rmarkdown document)

```

# prepare code appendix
labels = knitr::all_labels()[-(length(knitr::all_labels())-1)]

```

```

knitr::opts_chunk$set(echo = F, fig.align = 'center')

price.betas = read.csv('mcmc_simulations/FINAL_price_coefficients.csv')[, -1]
# Display the map of the MDB
knitr::include_graphics('img/mdb-map-low-res.png')

# Display the posterior density of theta for price

# Get prior HPs
mu.0 = read.csv('mcmc_simulations/mu_0.csv')[2, -1]
Lambda.0 = read.csv('mcmc_simulations/lambda_0.csv')[, -1][2, 2]

xvals = seq(-1, 1, 0.01)
densities = dnorm(xvals, mu.0, Lambda.0)

theta = read.csv('mcmc_simulations/FINAL_price_thetas.csv')[, 2]
plot(density(theta, bw = 0.05), main = '',
      xlab = bquote(theta), lwd = 2, col = 'blue')
lines(densities ~ xvals, lwd = 2, col = 'red')
legend('topleft', legend = c('Prior density', 'Posterior density'), lwd = 2, col = c('red', 'blue'))
# Display a table of price coefficients
elasticity.means = apply(price.betas, 2, mean)

t = 13; n = 1222; N = 94

# Load the data
murray = read.csv('data/murray.csv')
crops.in.order = murray$crop[seq(1, n, by = t)]
regions.in.order = murray$region[seq(1, n, by = t)]
murray.sequence = data.frame(region = regions.in.order, crop = crops.in.order)

# Construct the raw table by iterating over regions and then
# crops
elasticity.table = matrix('', nrow = length(unique(murray$region)),
                          ncol = length(unique(murray$crop)),
                          dimnames = list(rownames = unique(murray$region),
                                           colnames = unique(murray$crop)))

crops = unique(murray$crop)
regions = unique(murray$region)
for (i in 1:length(regions)) {
  for (j in 1:length(crops)) {
    index = (which(murray.sequence$region == regions[i] & murray.sequence$crop == crops[j]))

    if (length(index) != 0) {
      elasticity.table[i, j] = round(elasticity.means[index], 3)
    }
  }
}

# Rename some columns to save space
colnames(elasticity.table)[7] = "Grazing"
colnames(elasticity.table)[2] = "Misc broadacre"

```

```

colnames(elasticity.table)[4] = "Misc cereal"
colnames(elasticity.table)[8] = "Hay"
colnames(elasticity.table)[9] = "Other"

# Split the table in two
elast.tab.1 = elasticity.table[, c(1, 3, 5, 6, 7, 8)]
elast.tab.2 = elasticity.table[, c(12, 11, 10, 2, 4, 9)]

# Print the table
knitr::kable(elast.tab.1, longtable = T)
knitr::kable(elast.tab.2, longtable = T, caption = "Table of mean estimated elasticities by crop type and year")

# show the posterior density of the correlation coefficient
rho = read.csv('mcmc_simulations/rho.csv')[, -1]
plot(density(rho, bw = 0.008), main = '', lwd = 2)

# Show the plot of effective sizes
effective = read.csv('mcmc_simulations/effective_sizes.csv')

sizes = effective[1:287, 2]
names = effective[1:287, 1]

plot(sizes, type = 'h', ylab = )
abline(h = 1222, lty = 2, col = 'red', lwd = 2)

# Show autocorrelation for rho and sigma2
rho = read.csv('mcmc_simulations/rho.csv')[, 2]
sigma2 = read.csv('mcmc_simulations/sigma2.csv')[, 2]

par(mfrow = c(1, 2))
  acf(rho, main = bquote(rho))
  acf(sigma2, main = bquote(sigma^2))
par(mfrow = c(1, 1))

# Regression diagnostics
layout(matrix(c(1, 1, 1, 2, 2, 3), nrow = 2, byrow = T))
fitted.vals = read.csv('mcmc_simulations/fitted_vals.csv')[, 2]
resids = as.numeric(read.csv('mcmc_simulations/residuals.csv')[, 2])

# Resids vs fitted values
plot(resids ~ fitted.vals, main = '', ylab = 'Residuals', xlab = 'Fitted values')
lines(lowess(resids ~ fitted.vals), col = 'red', lty = 2, lwd = 2)

# ACF of resids
acf(resids, main = '')

# QQ plot
par(pty = 's')
  qqnorm(resids, main = '')
  qqline(resids)
par(pty = 'm')

```

```
# Posterior predictive check (I had to make this in a  
# separate file because the data used to construct the  
# histogram was very large).  
knitr::include_graphics('figures/posterior_predictive_check.png')  
  
# prepare code appendix  
labels = knitr::all_labels()[-(length(knitr::all_labels())-1)]
```