**Queensland University of Technology**

**Western Australia**
**Department of Health**


# Bayesian modelling project

Deliverable 2: Modelling recommendations

**Prepared by**
James Hogg
Susanna Cramb

# Contents

# List of Figures



# List of Tables

# List of Code

# Abbreviations

| | |
|---|---|
| AIC | Akaike Information Criteria |
| AOD | Alcohol and Other Drugs |
| ASDALY | Age-Standardised DALY |
| ASMR | Age-Standardised Mortality Ratios |
| ASR | Age-Standardised Rates |
| ASRA | Age-Standardised Rates with Age |
| ASRAME | Age-standardised Rates with Age and Measurement Error |
| ASYLD | Age-Standardised YLD |
| ASYLL | Age-Standardised YLL |
| AUC | Area Under Curve |
| AYA | Area, Year and Age |
| BIC | Bayesian Information Criteria |
| BoD | Burden of Disease |
| DALY | Disability-Adjusted Life Years |
| DOHWA | Department of Health Western Australia |
| DPP | Difference in Posterior Probability |
| ERP | Estimated Residential Population |
| ESS | Effective Sample Size |
| FH | Fay-Herriot |
| FHELN | Fay-Herriot Empirical Logistic Normal |
| HD | Health District |
| HDI | Highest Density Interval |
| HR | Health Region |
| HWSS | Health and Wellbeing Surveillance System |
| LGA | Local Government Areas |
| MCMC | Markov Chain Monte Carlo |
| MrP | Multilevel Regression and Poststratification |
| Nimble | Numerical Inference for statistical Models using Bayesian and Likelihood Estimation |
| PP | Point Prevalence |
| PPC | Posterior Predictive Checks |
| QUT | Queensland University of Technology |
| RE | Random Effect |
| ROC | Receiver Operating Characteristic |
| SA2 | Statistical Areas Level 2 |
| SAE | Small Area Estimation |
| SAS | Statistical Analysis System |
| SIR | Standardised Incidence Ratio |
| SMR | Standardised Mortality Ratio |
| SRR | Standardised Rate Ratio |
| ST | Spatio-Temporal |
| WAIC | Widely Applicable Information Criteria |
| WMrP | Weighted Multilevel Regression and Poststratification |
| YLD | Years Lived with Disability |
| YLL | Years of Life Lost |

# Notation

| | Index | Total | Notation |
|---|---|---|---|
| Individual | $j$ | $n_{it}$ | $j = 1, \ldots, n_{it}$ |
| Unique combination | $f$ | $F$ | $f = 1, \ldots, F$ |
| Area | $i$ | $M$ | $i = 1, \ldots, M$ |
| Area (alt.) | $k$ | $M$ | $k = 1, \ldots, M$ |
| Time | $t$ | $T$ | $t = 1, \ldots, T$ |
| Age | $a$ | $A$ | $a = 1, \ldots, A$ |
| Covariate | $q$ | $Q$ | $q = 1, \ldots, Q$ |
| Posterior draws | $d$ | $D$ | $d = 1, \ldots, D$ |
| Health state | $h$ | $H$ | $h = 1, \ldots, H$ |

Table 1: Dictionary of indices used in this report.

| | Notation | Subscript | Superscript |
|---|---|---|---|
| Observed data | $y$ | any | st |
| Fitted values | $\mu$ | any | $(d)$, nz |
| Expected counts | $E$ | $it$ | $(d)$ |
| Intercept | $\alpha$ | | $(d)$, nz |
| Design matrix of fixed effects | $\mathbf{X}$ | any | |
| Design matrix of random effects (see 8.1) | $\mathbf{G}$ | any | |
| Coefficients for fixed effects | $\boldsymbol{\beta}$ | $q$ | $(d)$ |
| Coefficients for fixed effects (see 8.8) | $\boldsymbol{\beta}^{\text{qr}}$ | $q$ | $(d)$ |
| Coefficients for random effects (see 8.1) | $\boldsymbol{\lambda}$ | | $(d)$ |
| Standard normal random variables | $Z$ | any | any of $\gamma, \delta, s, v$ |
| Spatial random effects (ICAR) | $s$ | $i$ | |
| Unstructured random effects | $v$ | $i$ | |
| Combined spatial random effects | $\theta$ | $i$ | |
| Unstructured and structured mixing parameter | $\rho$ | | |
| Scaling (BYM2) | $\kappa$ | | |
| Temporal random effects (ICAR) | $\gamma$ | $t$ | |
| Space-time random effects | $\delta$ | $it$ | |
| Spatial weight matrix | $\mathbf{W}^{\text{S}}$ | $ik$ | |
| Temporal weight matrix | $\mathbf{W}^{\text{T}}$ | $ik$ | |
| Variance | $\sigma^2$ | any of $\gamma, \delta, s, v, \phi$ | |
| Survey weights | $w$ | $jit$ | |
| Population | $N$ | any | |
| Sample size | $n$ | any | |
| Disability weight | $e$ | $h$ | |
| Life expectancy | $L$ | $a$ | |
| Probability | $p$ | any | $(d)$ |
| Direct estimate (from survey) | $\hat{p}$ | $it$ | u |
| Sampling variance (of $\hat{p}$) | $\hat{\psi}$ | $it$ | u |

Table 2: Definitions of the notation used for specifying the Bayesian models in this report.

# 1 Introduction

The Department of Health Western Australia (DOHWA) is currently working on a modelling and mapping project to improve health insights in Western Australia. This project aims to leverage a large quantity of administrative and survey data along with spatio-temporal (ST) Bayesian models to generate robust small area estimates and measures of uncertainty for a variety of health metrics for a wide range of conditions and indicators across multiple years. The goal is to generate smoothed estimates of these health metrics at three geographical levels; namely health districts (HDs), local government areas (LGAs) and statistical areas level 2 (SA2s). Finally, health metrics will be provided separately for males, females and persons and broken down by Aboriginality where the data is available.

The project has access to three distinct types of data, which require different models.[1] The first and largest is administrative/registry data, which includes cancer incidence, hospitalisations and mortality. These data require spatial and temporal smoothing to meet issues relating to small counts and populations and to avoid some privacy and confidentiality requirements.

The second type of data are annual Western Australian population health surveys which will be used to provide prevalence estimates for a variety of health factors, including but not limited to smoking, diet, alcohol and obesity. These data require more complex modelling, as non-sampled data must be imputed whilst simultaneously correcting for sampling and non-response bias in the survey design and collection process.

Finally, mapping metrics related to the burden of disease is required. There is a large amount of data required to estimate the burden of disease, which include prevalence estimates, mortality data, comorbidity adjusted disability weights, and life expectancy estimates.

As the funding agencies for the project, both DOHWA and FrontierSI have contracted researchers from the Queensland University of Technology (QUT) to explore and assess the amassed data and recommend suitable Bayesian models. This report provides additional details based on the recommendations provided in Deliverable 1.

## 1.1 Structure

To begin with we'll discuss the details of Bayesian inference and computation (Section 2), which are critical to drawing valid conclusions from Bayesian models. We'll then introduce Bayesian hierarchical models and discuss the details and construction of spatio-temporal models in general (Section 3). In the following sections, we'll provide the R code and math details for the recommended models for the three data types (Sections 4, 5, 6), along with examples of the data structure and plots of results where necessary. Note that this report (and the R code contained within it) are not substitutes for the training materials. As such, the R code in this report cannot be solely used to conduct the modelling.

[1] See Figure 22 in Section 8 for an overview flowchart of the data and models.

Scattered throughout the report we've included *Tech Talk!* and *Consider!* boxes that aim to briefly highlight any technical or theoretical details that arise from our discussions. The large appendix (Section 8) includes a brief introduction to the generic mathematical notation (vectors, matrices, etc) used in this report (Section 8.1), the epidemiology metrics (Section 8.3) and details of any computational tricks used. The mathematical notation used throughout most of this report can be found in Tables 1 and 2.

The Bayesian ST models we discuss in this report have been purposely recommended for their wide applicability across different data types and conditions. As long as the format of the data and the outcome is of the correct type (e.g. count or binary), Bayesian ST models learn the best way to approximate the data, given the model structure we impose. To ensure a reasonable balance between efficiency, ease of use and appropriateness, the models we recommend impose enough structure to achieve the goals of the project, but also the flexibility for the models to learn what is required from the data itself. Thus, this technical report purposely does not focus on any *single* condition.

# 2 Bayesian Inference

The benefit of Bayesian inference and modelling is its flexibility, probabilistic interpretation and simple reporting of uncertainty.[2] Bayesian inference considers model parameters, $\mathbb{P}$, as random and data, $\mathbf{y}$, as fixed (see Table 3 for a breakdown of the differences between Bayesian and frequentist inference). Unlike in frequentist inference, where the parameter estimates are those that maximise the log likelihood, $\log p(\mathbf{y}|\mathbb{P})$, in Bayesian inference the parameter estimates are the posterior distributions, $p(\mathbb{P}|\mathbf{y})$,[3] which specifies the distribution of the parameters of the Bayesian model, given our data, $\mathbf{y}$. The posterior distribution is a combination of the likelihood, $p(\mathbf{y}|\mathbb{P})$, and the prior distribution, $p(\mathbb{P})$. The posterior distribution is derived using Bayes theorem,

$$p(\mathbb{P}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbb{P})\, p(\mathbb{P})}{p(\mathbf{y})}. \tag{2.1}$$

Given that the posterior is a distribution, the model parameters have a natural probabilistic interpretation. For example, Bayesian inference allows us to derive the probability that a parameter, $\mathbb{P}$, is greater than some value, $\Pr(\mathbb{P} > c|\mathbf{y})$.

| **Frequentist** | **Bayesian** |
|---|---|
| Probability is "long-run frequency" | Probability is "degree of certainty" |
| $p(\mathbf{y}\|\mathbb{P})$ is a sampling distribution (function of $\mathbf{y}$ with fixed $\mathbb{P}$) | $p(\mathbf{y}\|\mathbb{P})$ is a likelihood (function of $\mathbb{P}$ with fixed $\mathbf{y}$) |
| No prior | prior |
| $p$-values (null hypothesis tests) | Full probability model available for summary/decisions |
| Confidence intervals | Credible intervals |

Table 3: Some of the core differences between Bayesian and frequentist thinking and inference.

## 2.1 Computation

Since Bayesian models are often numerically intractable, estimates are generally computed through an algorithm called Markov Chain Monte Carlo (MCMC) (Gelman et al. 2014a), which approximates the posterior distribution of our Bayesian models by drawing a very large number of samples, say $D$, from $p(\mathbb{P}|\mathbf{y})$.[4] Although there are a wide range of MCMC algorithms, those proposed in this work rely on methods called random walk or Gibbs sampling.[5] These methods propose (or step) to new parameter value by comparing the likelihood of the current value to the proposed value (see the box on page 11 for details on how to select an appropriate step size). MCMC begins by specifying an initial set of

[2] Please see the rigorous introduction to Bayesian workflow by Gelman et al. (2020)

[3] $p(.)$ denotes a probability distribution. For example, $p(X)$ would denote the probability distribution for the random variable $X$

[4] Other methods (e.g. Variational Inference) can be used to obtain the posterior distribution. These methods can be considerably faster than MCMC methods, but at the cost of accuracy and simplicity.

[5] Please see Chapter 9 of McElreath (2020) for an intuitive introduction to random walk and Gibbs samplers.

parameter values, before running a "chain of steps" (technically called a Markov Chain) with the goal that the entire collection of steps (the posterior draws) will approximate the true posterior distribution. In this example, $D$ is the number of steps we ask the MCMC algorithm to take.

MCMC algorithms can provide exact inference for Bayesian models when $D$ is very large. However, for finite $D$, say $10,000$, the validity of inferences from Bayesian models depends on whether the algorithm has *converged*. Technically speaking, convergence refers to the stabilization of the Markov chain that is used to simulate the posterior distribution. Given the importance of convergence in Bayesian analysis, convergence *must* be assessed prior to drawing any model inferences. Unfortunately, in practice, it is impossible to validate whether an MCMC algorithm has converged to the *true* posterior. That said, there are two pivotal checks and corresponding metrics we recommend using to have confidence in the validity of the MCMC output and resulting inferences.

**R-hat** It is recommended to run multiple, independent MCMC algorithms for the same model, called chains. By starting each chain with a different set of initial parameter values, we can ascertain whether convergence is acceptable by comparing the behaviour of the posterior draws from different chains. Well behaved chains should converge to the same area of the parameter space regardless of the initial parameter values used. Separate chains that converge to the same density are described as "mixing well".

Note that a single chain can also be used but must be run for a long time compared to the shorter runs we can use for each of multiple chains. Furthermore, current MCMC diagnostics rely on the assessment between chains (Vehtari et al. 2021), which makes multiple-chain approaches preferable. The posterior draws from MCMC are the combined draws from multiple chains or the draws from a single chain.

The $\hat{R}$, which is always greater or equal to 1, is used for these assessments [6]. An $\hat{R} = 1$ denotes convergence and is desirable for all parameters of a model. Vehtari et al. (2021) suggest a softer and more reasonable cutoff for acceptable convergence; $\hat{R} < 1.01$. We use the recommendation by Vehtari et al. (2021).

**Effective sample size (ESS)** Given the stepping method described above, the posterior draws are *not* independent — even though we would like them to be. A good measure of the efficiency of an MCMC algorithm is the effective sample size (ESS). The ESS considers the dependence in the posteriors and estimates the number of independent posterior draws that our $D$ draws represent. Like the $\hat{R}$, the ESS is a good measure of convergence and is a standard output from Bayesian software [7].

A good check of the correlation in the posterior draws is the autocorrelation plot. An example can be found in Figure 2. In general, if a parameter has been effectively sampled, we should see an autocorrelation plot similar to the left or middle columns (`sigma2_theta`

[6] Readers interested in the formula behind $\hat{R}$ should refer to Vehtari et al. (2021).
[7] Readers interested in the formula behind ESS should refer to Vehtari et al. (2021).

and `sigma2_gamma`) in Figure 2, which suggests that even after a single iteration the posterior draws are close to independent (zero correlation). Observe that the posterior draws on the right are still reasonably strongly correlated even after 5 iterations. This suggests that `sigma2_delta` is particularly difficult to sample and may have low ESS. See Section 2.1 for some recommendations on how to improve convergence and ESS.

A highly correlated or inefficient MCMC algorithm would give very low values of ESS. In most cases, ESS can be artificially increased by taking more posterior draws (e.g. setting $D$ higher). However, these decisions must be balanced with the computation cost. An efficient MCMC algorithm should achieve an ESS as close to $D$ as possible - indicating completely independent draws. Note that the accuracy of any inference drawn from MCMC depends on the ESS. A crude rule of thumb used by `rstan` (Stan Development Team 2022) is that all model parameters should have ESS larger than the number of chains multiplied by 100. For example, if one is to run 4 independent chains, the recommended cutoff is for all parameters to have an ESS larger than 400.

***Tech talk!* Step size and adaption**

For these stepping methods, the size of the step is very important and can have a drastic effect on the efficiency and validity of the MCMC algorithm. Fortunately, in practice, we do not need to manually select the step size. The software recommended in this project uses an automatic adaption scheme that selects the most efficient step size for the model and data, before producing usable posterior draws. This adaptive period of the MCMC algorithm is called the *burn-in*. Posterior draws produced during burn-in should not be used for inference and are generally discarded.

**Assessing convergence** In Figure 1 we display trace plots of the posterior draws of a single parameter: the mean of some continuous data. Trace plots show the evolution of the posterior draws during the algorithm and are very helpful tools to ascertain the convergence of our MCMC algorithms. In plot (a) and (b) of Figure 1, we use a poorly optimised step size while plots (c) and (d) use a well-chosen step size.

In plot (a) the posterior draws move extremely slowly toward the true value, indicating very slow and poor mixing. We see extremely low ESS and a very large $\hat{R}$, both indicating that the chains have not mixed; we should not trust the posterior draws. The core problem in plot (a) is that the posterior draws are highly correlated given the poorly chosen step size. One can still obtain convergence with this step size, but must dramatically increase the number of posterior draws, $D$.

In plot (b) of Figure 1, we increase $D$ tenfold and also thin the chains by 100. *Thinning*, in this case, involves discarding every 1-99th draw and keeping only each hundredth in the hope that after 100 draws, samples will be much less correlated.[8] Although the trace plot

[8]Thinning by 100 is generally not advisable as this can indicate a problem with the model. In practice, thinning between 10-20 is usual.

and convergence diagnostics ($\hat{R}$ and ESS) in plot (b) suggest convergence, computation took 100 times longer than that for plot (a). Similar to plot (b), plot (d) in Figure 1 shows very good convergence. The well optimised MCMC algorithm has provided almost independent posterior draws for a fraction of the computational cost required to obtain the draws in plot (b). Plot (c) in Figure 1 is another example of poor convergence. However, unlike in plot (a) where our reason for claiming non-convergence was due to poor mixing, the draws presented in plot (c) should *not* be trusted because each chain has converged to a different area of the parameter space. The convergence diagnostics highlight the issue as the ESS is very low and $\hat{R}$ is very high.

The illustrations in this section are pivotal to understanding the importance of Bayesian computation in applications of Bayesian modelling. We observed that a well optimised MCMC algorithm can provide substantially faster and more accurate inference.[9] Note that in the past decade there have been significant improvements in MCMC algorithms, with the current state-of-the-art being implemented in Stan (Stan Development Team 2022).

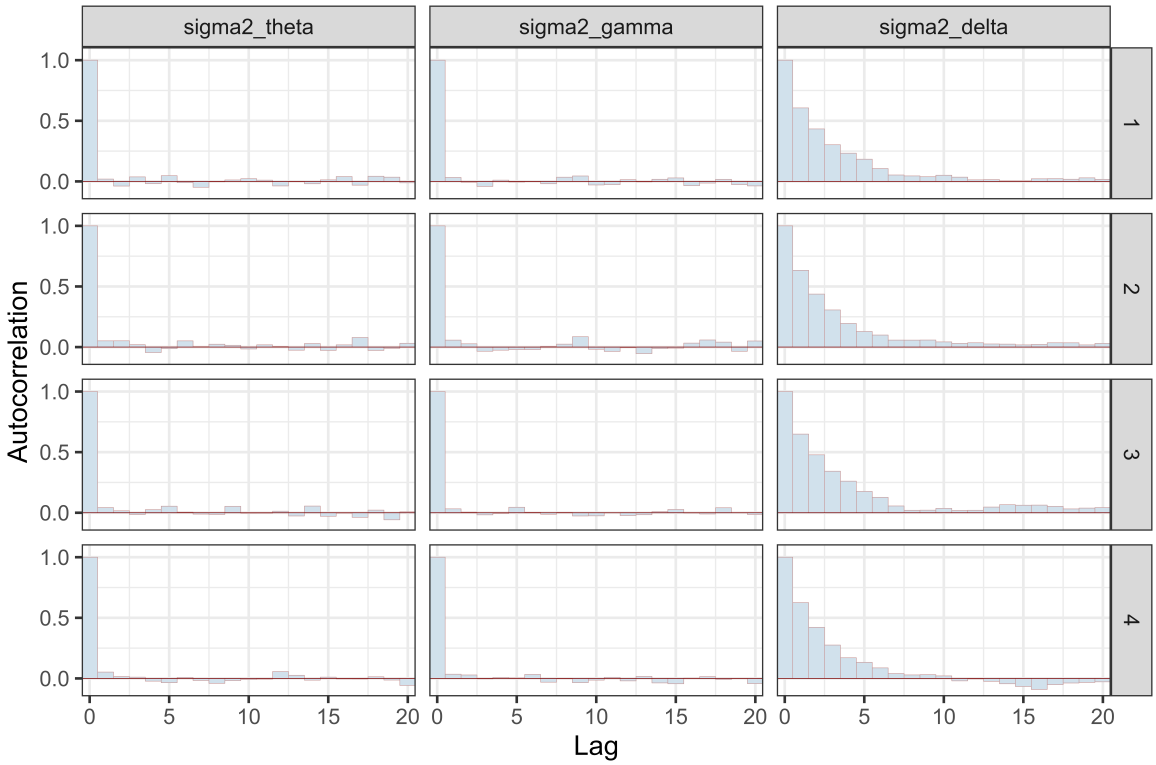



Figure 2: Example of an autocorrelation plot for three parameters (columns) across 4 chains (rows). Autocorrelation plots describe the correlation between draws in the chains. The $x$-axis describes the lag number of iterations (after thinning), while the $y$-axis gives the correlation. For example, the bar at lag 5 gives the average correlation of the posterior draws that are 5 samples apart.

[9]The term 'better' here refers to the quality of the posterior draws in terms of efficiency and accuracy.

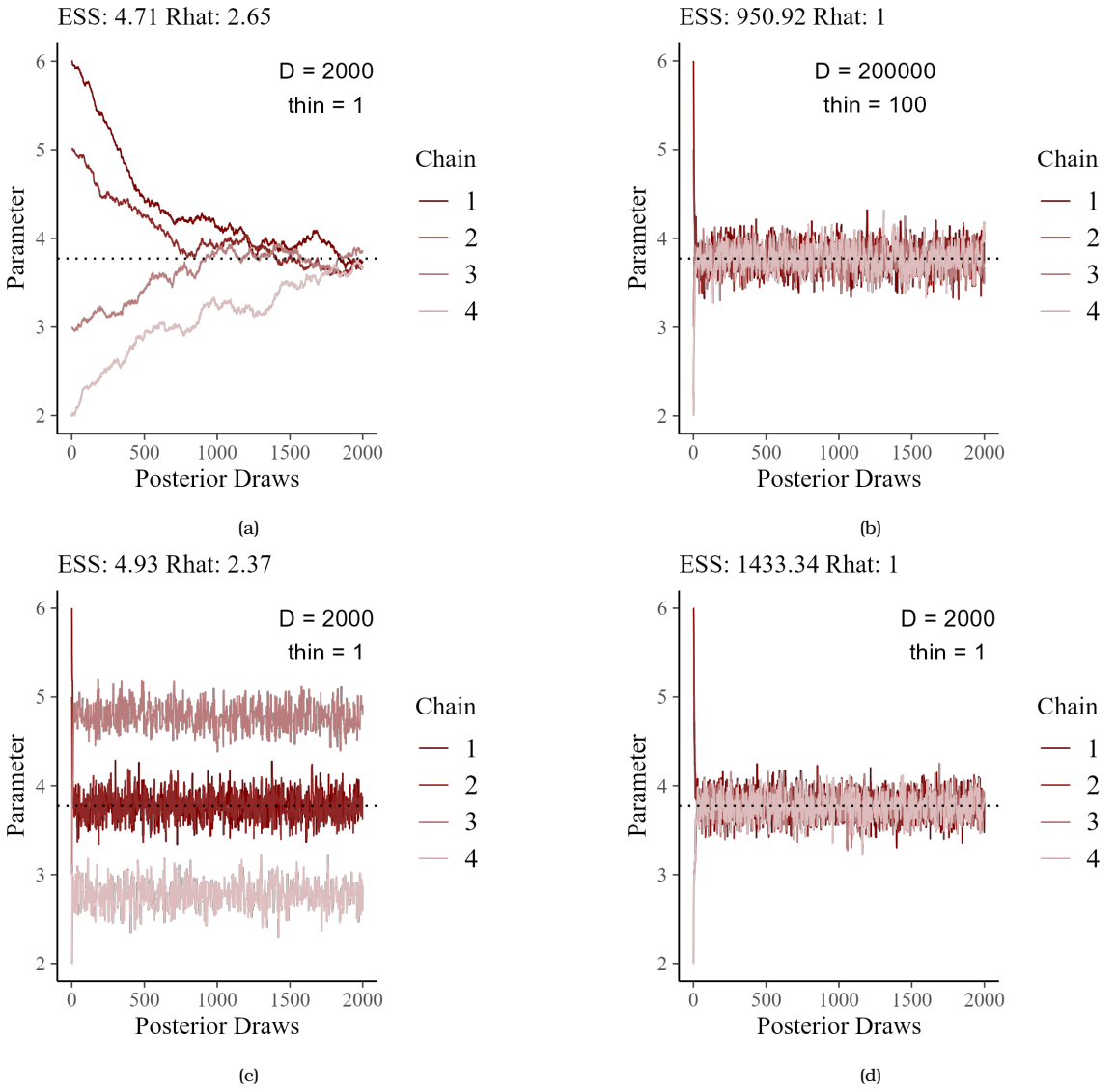



Figure 1: Trace plot of a single parameter (mean of continuous data) estimated using a simple random walk MCMC algorithm ($D = 2,000$ draws for each of 4 chains). Each plot has the number of iterations, $D$, thinning value, effective sample size (ESS) and $\hat{R}$. The initial values for the algorithm are 2,3,5,6 and the dotted line represents the true parameter value from the simulated data. Plot (a) illustrates non convergence because of poor mixing. Plot (b) illustrates convergence, but uses $D = 200,000$ posterior draws with 100 thinning rather than $D = 2000$ like the other three plots. Plot (c) illustrates non convergence because the chains have converged to different parameter values. Plot (d) illustrates good convergence of the MCMC algorithm.

**Improving convergence** As we've shown, convergence is essential to drawing valid inference from Bayesian models fitted via MCMC. Advanced users may wish to apply a range of computational tricks, but in most situations convergence can be improved through the following:

- Increase $D$ (i.e. run the algorithm for longer)
- Increase the number of iterations for burn-in
- Increase the level of thinning[10] (see Section 2.1)
- Use more informative initial values (e.g. taken from frequentist models)
- To ascertain which component/s of the model is causing convergence problems, we recommend reducing the complexity of the model (e.g. dropping random effects or fixed effects) until convergence is achieved.
- Increase the frequency of adaption (see the box on page 11)
- See the box on page 41 for more specific help

Please see Section 8.9 for example Bayesian software output with annotations.

## 2.2 Inference

Unlike frequentist model estimation, where model output is generally comprised of point estimates, standard errors and $p$-values, the output of Bayesian models estimated using MCMC are the $D$ posterior draws. With access to the posterior draws, a Bayesian[11] can calculate summary metrics (e.g. means, medians and quantiles), or apply any transformation to derive posterior distributions for other variables of interest. Below we describe the two core outputs required for this project: point estimates and measures of uncertainty.

### 2.2.1 Point estimate

In Bayesian inference a point estimate, $\hat{\theta}$, for a single parameter, say $\theta$, can be calculated using the empirical mean (or median) of the corresponding posterior draws for that parameter,

$$\hat{\theta} = \frac{1}{D}\sum_{d} \theta^{(d)}$$

where $\theta^{(d)}$ is the $d$th posterior draw of $\theta$ from our Bayesian model. Details of this notation can be found in Section 8.1.

[10] Note that increasing the thinning amount without also increasing $D$, will result in fewer usable draws.

[11] Crudely, a "Bayesian" indicates any scientist who takes a Bayesian perspective when conducting statistical analysis.

### 2.2.2 Uncertainty

There are three common methods for reporting the uncertainty of Bayesian model parameters: posterior standard deviations, credible intervals and exceedance probabilities. The uncertainty measures we recommend are derived from the posterior draws directly. Thus, given that one can derive posterior draws for *any* quantity of interest (age-standardised rates, years of life lost, prevalence, etc), one can also calculate uncertainty measures for all metrics in the same manner.

First is the standard deviation of the posterior draws, which is referred to as the *posterior standard deviation*. For relatively symmetric posterior distributions, the posterior standard deviation may be similar to a frequentist standard error. Note that for posterior distributions that are not *approximately* symmetric, the posterior standard deviation can be a poor measure of uncertainty.

The second uncertainty method is the *credible interval*. Bayesian credible intervals give an interval which has a 95% chance that the true parameter value lies within it. Credible intervals can be derived for all model parameters by calculating the empirical quantiles of the posterior draws. For some parameters, the posterior distribution may be highly skewed which means the quantile method of deriving credible intervals can be ineffective (see Figure 3) at capturing the most appropriate interval, in terms of values with the highest plausibility. An alternative interval is the highest density interval (HDI). Unlike quantile credible intervals, which are symmetric around the median, HDIs cover the parameter values corresponding to the highest density of the posterior. For approximately normally distributed posteriors, HDIs and quantile credible intervals will be very similar. Thus, we recommend using HDIs where possible. Our user-made function `jf$getResultsData()` returns HDIs as default. See Section 2.3 from Gelman et al. (2014a) for a more thorough comparison of quantile credible intervals and HDIs.

The third measure of uncertainty is *exceedance probabilities*: the probability of the posterior being above a certain value. These can be derived from the posterior draws using

$$\frac{1}{D}\sum_{d}\mathbb{I}\left(\theta^{(d)} > c\right),$$

where $\mathbb{I}\left(\theta^{(d)} > c\right) = 1$ if $\theta^{(d)}$ is larger than some specified value $c$.[12] For this project, for example, exceedance probabilities can indicate whether the age-standardised rate (ASR) in a particular area is significantly higher than the state ASR. Commonly values above 0.80 (i.e. 80% of the posterior) are considered likely to be above, while if the exceedance probability is below 0.2 (so 80% of the posterior is below the value) it is considered likely to be below.

A variant of exceedance probabilities were used to great effect in the Australian Cancer Atlas (Duncan et al. 2019). They used the difference in posterior probabilities (DPP),

[12] $\mathbb{I}(.)$ is called the indicator function.

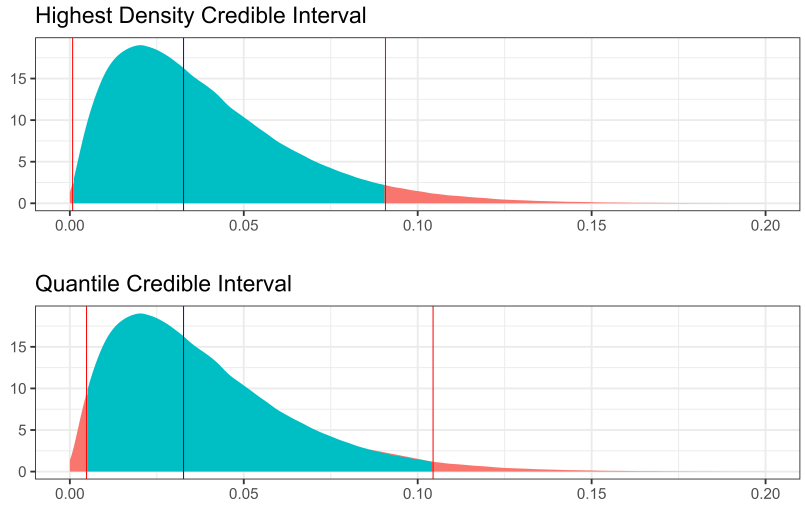



Figure 3: Comparison of highest density interval and quantile credible interval for a skewed distribution. The middle blue vertical line is the median, while the red lines on either side denote the bounds of the intervals.

$$2\left|\left(\frac{1}{D}\sum_{d}\mathbb{I}\left(\theta^{(d)}>c\right)\right)-0.5\right|.$$

By using the `jf$getDPP(.)` function (see Code 1), one can easily derive the exceedance probabilities and DPPs. The function also returns a binary vector denoting which columns of `draws` are significantly different to the `null_value` at the 60% level.

```
# `draws` is a matrix with D rows and n_obs columns
dpp_obj <- jf$getDPP(draws, null_value = 1, sig_level = 0.60)

# Exceedence probability
dpp_obj$EP

# DPP
dpp_obj$DPP

# binary vector of significance
# of the DPPs
dpp_obj$DPP_sig
```

R Code 1: Calculating DPP using our user-made function

Please see Section 8.9 for example Bayesian software output with annotations.

## 2.3 Goodness-of-fit

Once we deem that our Bayesian model has converged, we recommend some simple model checks to ensure the results are plausible. These include comparing the observed and fitted values, examining model residuals and performing posterior predictive checks and sensitivity analysis.

**Observed vs fitted** Bayesian inference via MCMC is a difficult task - particularly identifying when a coding error has occurred. It is always recommended to plot the observed data (e.g. counts or rates) versus the modelled estimates. Note that we do *not* expect (or wish) for exact concordance between the observed and modelled estimates — remember the point of this modelling project is to smooth the data and thus, provide more reliable estimates.

Observed versus fitted plots are a great way to identify any model specification or coding errors. In our explorations, we always include a diagonal line of equality in all our observed versus fitted plots, and expect the points to sit either on the line or close to it. Figures 11 and 12 provide some examples.

**Plausibility checks** Although comparing the observed and fitted data is a useful check, for some of the models (Section 5) discussed in this report, we recommend further plausibility checks which are not a purely Bayesian check, but recommended for all statistical analysis. Plausibility checks are an excellent way to ensure that the specified model is working as expected and that your code is correct even though no errors were produced by the software [13]. These checks might include, for example, comparing ASRs or prevalence estimates against remoteness or socioeconomic status or by comparing the posterior standard deviations of estimates to the corresponding area-by-time population or sample sizes.

**Residual plots** We recommend examining the relationship between the posterior of the standardised residuals and the fitted values, $\mu_i$, to ensure there are no systematic patterns in the residuals. Standardised residuals for Poisson models can be derived for the $d$th posterior draw using the following formula.

$$r_i^{(d)} = \frac{y_i - \mu_i^{(d)}}{\sqrt{\mu_i^{(d)}}} \tag{2.2}$$

To simplify these checks, we suggest taking the median of both the residual and fitted count draws. Generally, these residual plots should have little pattern across the fitted counts. Note that for severely sparse count data, residual checks are difficult to interpret, and one should rely more on posterior predictive checks. See Figure 4 for examples of residual plots for common and sparse count data.

[13] Note that incorrectly specified priors in `nimble` may not throw any errors, thus giving the false pretence of a *correct* model.

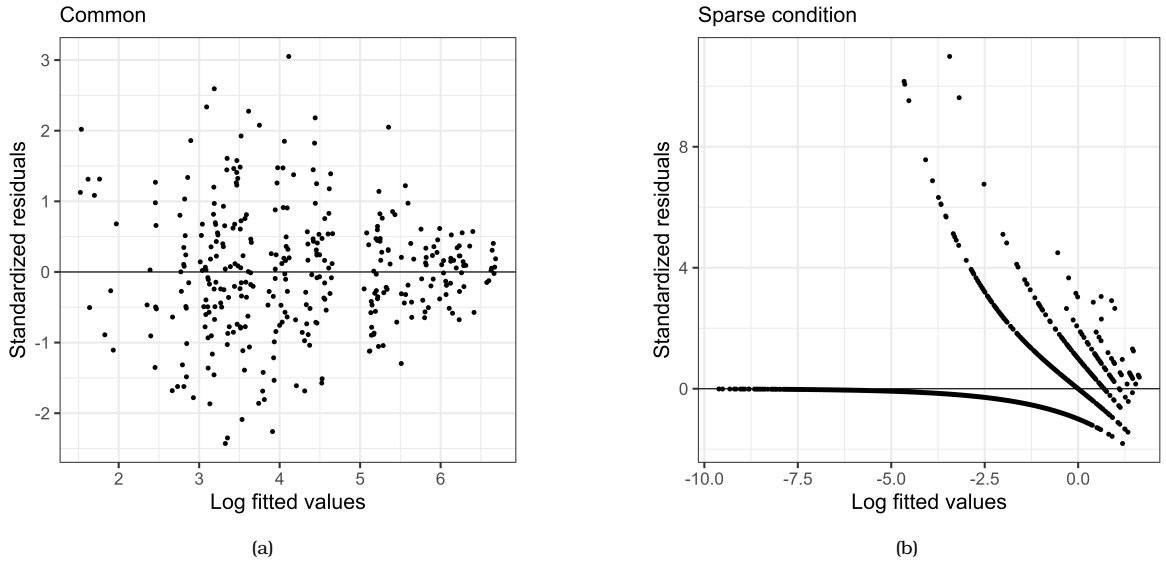



Figure 4: Some examples of residual plots for common (a) and sparse (b) count data, where the points on the plot are posterior medians. Plot (a) is relatively easy to interpret and has no horizontal patterns of concern. On the other hand, plot (b) is unintuitive and unhelpful. That said, the posterior predictive checks for the sparse condition, shown in (b), suggest a very good fit for the model.

**Sensitivity analysis** To ensure our models are robust, prior choices should be investigated using sensitivity analysis. This is particularly important for priors on model hyperparameters, i.e. hyperpriors (see Section 3.2 for an example of hyperpriors). The idea is to fit the same model with different hyperpriors[14] and compare the resulting posterior distribution (Gelman et al. 2020). If the data is sufficiently large and the model is well specified, the choices of priors and hyperpriors will often have little effect on model inference. However for some of these complex ST models, hyper/prior choices and sensitivity analyses are important to ensure that model inference is not strongly dependent on our choice of priors.

**Posterior predictive checks** Posterior predictive checking (PPC) involves simulating new data, conditional on the posterior distribution (Gelman et al. 2020). We can then derive a metric or series of metrics for each set of simulated data and compare these metrics to the actual data.

For Poisson models, we may be interested in ensuring that our Bayesian ST models approximate the correct total number of counts. This process can be achieved by using Code 2, or the following process,

$$\tilde{y}_i^{(d)} \sim \text{Poisson}\left(\mu_i^{(d)}\right)$$
$$\text{Metric}^{(d)} = \sum_i \tilde{y}_i^{(d)},$$

[14]For example, one could change the hyperprior distribution on a variance parameter from a Gamma$(2, 0.5)$ to a Gamma$(2, 0.01)$.

where the distribution of Metric (across all $D$) can be compared to the sum of the raw counts, $\sum_i y_i^{(d)}$. Figure 5 illustrates one graphical approach to posterior predictive checking using the `R` package `bayesplot`. Observe how the black vertical lines all sit at the mean of their corresponding histograms; this indicates that the model is fitting the data very well (even in this sparse case). Figure 5 can be created using `jf$PoissonPPC(df$y, yrep)` where `yrep` can be derived using Code 2. Note that while `bayesplot` uses $T(.)$ to denote Metric, we avoid this notation as $T$ is used to denote the total number of time points later in this report.

```r
# mu_draws is a matrix with D rows and n_obs columns
for(i in 1:D){
    # yrep: posterior predictive distribution
    # yrep is a matrix with D rows and n_obs columns
    yrep[i,] <- rpois(n_obs, mu_draws[i,])
}

# Alternative method to get posterior
# predictive distribution
    yrep <- jf$getPoisson_rep(mu_draws)

# Simple posterior predictive check for the sum of the counts
    # return the sum of the counts
    sum_y <- function(x) sum(x, na.rm = T)
    # apply the above function to each row of yrep
    # and then summarize the D values
    summary(apply(yrep, 1, sum_y))
```

R Code 2: Simulating the posterior predictive distribution for a Bayesian Poisson model and conducting a simple PPC.

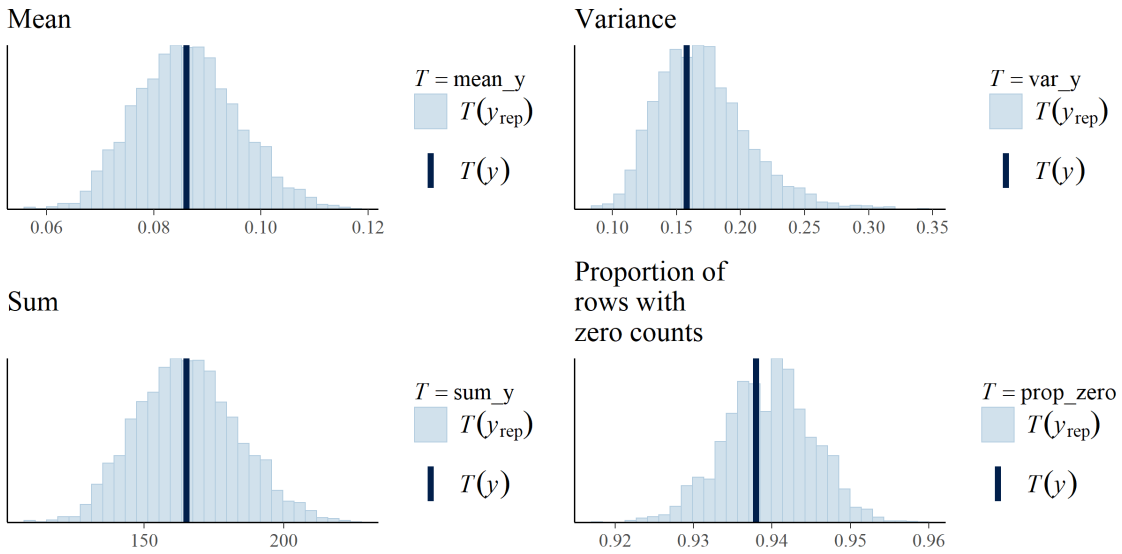



Figure 5: Illustration of how posterior predictive distributions can be used to check the fit of a Bayesian Poisson model. We choose the mean, variance, sum and proportion of zeros as the four metrics to evaluate the model fit. The four plots each display a histogram, $T(y_{\text{rep}})$, of the metrics, $T(.)$, evaluated on the $D$ posterior predictive draws, $y_{\text{rep}}$. Overlaid on the histograms are solid black lines which are the metrics evaluated on the observed data, $T(y)$.

## 2.4 Bayesian workflow

Now that we have described a variety of important components of Bayesian analysis, we recommend loosely following a generic *order of operations* for Bayesian analysis depicted in Figure 6.

1. In the first step, one prepares and formats the data. This step can be conducted in external software (e.g. `SAS`).

2. The second step involves the use of MCMC to fit the specified model (Section 2.1).

3. The third and arguably most important step is the goodness-of-fit checks (Section 2.3), where one ensures that both the model and computation are working as expected. Note that we have included a cycle between model fitting and goodness-of-fit checks as often one must repeat these steps several times to arrive at the final model.

4. Once the goodness-of-fit checks are complete, one can complete any post-processing of the posterior draws from the final model, which generally includes deriving point estimates and measures of uncertainty for the parameters or epidemiology metrics of interest (Section 2.2).

Note that the schematic in Figure 6 is an outline, designed to help guide practice.

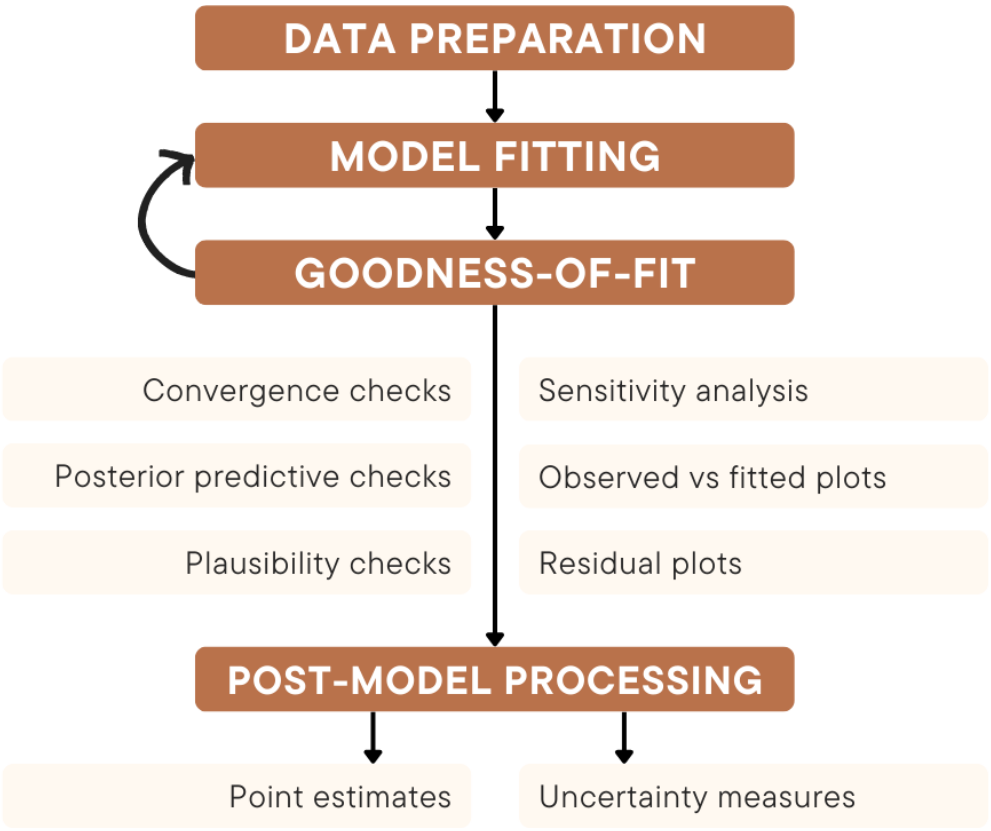



Figure 6: This schematic provides an *order of operations* for Bayesian analysis.

# 3 Bayesian spatio-temporal (ST) modelling

Bayesian spatio-temporal models are complex extensions of Bayesian hierarchical/multilevel models. These methods can reduce the variance and instability of estimates by borrowing information across both areas and time via intuitive local and global smoothing.

Most Bayesian ST models have five distinct elements. More details can be found in the following sections.

- Intercept: estimates the overall mean across all years and areas in the data,
- Spatial random effect (RE): allows the estimate for each area to deviate from the intercept,
- Temporal RE: allows the estimate for each year to deviate from the intercept,
- Space-time interaction RE: accommodates any area-specific temporal trends not captured by either the spatial or temporal RE,
- Fixed effects: adjusts the estimates according to important covariates such as age group, remoteness and socioeconomic status.

Although there is a large quantity of literature describing different specifications for the spatial, temporal and space-time REs (Haining and Li 2020; Lawson 2020; Ugarte et al. 2014), for this project we suggest using the common specifications that have useful theoretical properties, convenient interpretations, computational efficiency and significant successful applications in the field (Urdangarin et al. 2022).

***Consider!* Interpretation of Bayesian estimates**

The estimates from Bayesian ST models have the same interpretation as the raw values, however they would now be defined as "fitted", "modelled" or "smoothed" versions. For example, ASRs derived from ST models would have the same interpretation from a policy standpoint, however they would be classed as "smoothed" ASRs.

## 3.1 Regression models

Almost any statistical model can be fitted using Bayesian inference by first specifying the model using a series of probability distributions. Consider the standard linear model used for continuous outcomes, where data are assumed to be independent and identically distributed (iid). In most introductory courses of regression modelling, the linear model is written as follows,

$$\begin{aligned} y_i &= \alpha + \beta x_i + \epsilon_i && i = 1, \ldots, n \\ \epsilon_i &\overset{iid}{\sim} N(0, \sigma^2). && i = 1, \ldots, n \end{aligned} \tag{3.1}$$

The parameters of this model are $\mathbb{P} = (\alpha, \beta, \sigma)$, where $\alpha$ is the intercept, $\beta$ is the regression coefficient and $\sigma$ is the residual standard deviation. The data are denoted as $\mathbf{y} = (y_1, \ldots, y_n)$ and $\mathbf{x} = (x_1, \ldots, x_n)$. See Section 8.1 for more details of this notation.

We can easily rewrite this model using the likelihood + prior form. The likelihood for the standard linear model is the normal distribution, $N(\mu, \sigma^2)$, with a mean, $\mu$ and variance, $\sigma^2$. The prior distributions are chosen dependent on the parameter (e.g. a Gamma prior is chosen for $\sigma$ as the standard deviation must be positive).

$$
\begin{aligned}
&y_i \overset{iid}{\sim} N\left(\alpha + \beta x_i, \sigma^2\right) && i = 1, \ldots, n \\
&\alpha \sim N(0, 1000^2) \\
&\beta \sim N(0, 1000^2) \\
&\sigma \sim \text{Gamma}\left(\text{shape} = 2, \text{rate} = 0.5\right)
\end{aligned}
\tag{3.2}
$$

(3.1) and (3.2) are identical models but written in different forms. The linear model in (3.2) uses priors for $\alpha, \beta$ and $\sigma$ that are likely to be uninformative. A prior is uninformative if the distribution implies that a very large range of values for the parameter are reasonable before seeing the data. For example, the prior distribution for $\alpha$ implies that values up to 3000 and down to -3000 are plausible. For fixed effects such as $\alpha, \beta$, the uninformative prior, $N(0, 1000^2)$, is extremely common in practice. The Gamma prior used for $\sigma$ implies that before seeing data, values of $0 < \sigma < 10$ are plausible. With sufficient data, model inference from the Bayesian linear model in (3.2) would be identical to that from frequentist inference (e.g. fitted using ordinary least squares).

## 3.2 Hierarchical models

Spatio-temporal models are necessary extensions of hierarchical models. There are countless high level and detailed recounts and resources on multilevel and hierarchical models in the Bayesian framework. We recommend McElreath (2020) and Gelman et al. (2014a). For completeness we provide a very brief outline here.

Consider again, the Bayesian linear model specified in (3.2). Suppose we have data at the unit-level on people from 8 areas, where $y_{ij}$ denotes the $i$th person from area $j$. For these data, it may not be valid to assume the data are independent. However, it would be natural to assume independence of the people within each area (i.e. conditional independence). To accommodate this hierarchical structure into our linear model, we could estimate a separate intercept for all persons from the same area. In the model below, the effect of area is considered fixed and thus, we call them fixed effects.

$$
\begin{aligned}
y_{ij} &\overset{iid}{\sim} N\left(\beta_j, \sigma_e^2\right) && i = 1, \ldots, n_j; j = 1, \ldots, 8 \\
\beta_j &\overset{iid}{\sim} N(\alpha, 1000^2) && j = 1, \ldots, 8 \\
\alpha &\sim N(0, 1000^2) \\
\sigma_e &\sim \text{Gamma}\,(2, 0.05)
\end{aligned} \tag{3.3}
$$

For didactic purposes, a more familiar,[15] but *equivalent*, model to (3.3) could be written as

$$
\begin{aligned}
y_{ij} &= \alpha + \beta_j + \epsilon_{ij} \\
\epsilon_{ij} &\overset{iid}{\sim} N\left(0, \sigma_e^2\right).
\end{aligned}
$$

This Bayesian approach is generally classed as the "no-pooling" solution as the area-specific intercepts, $\beta_j$, do not share any information (i.e. they are independent). A pragmatic alternative is to let the 8 intercepts *themselves* come from a distribution (e.g. $\beta_j \overset{iid}{\sim} N\left(\alpha, \sigma_\beta^2\right)$).

$$
\begin{aligned}
y_{ij} &\overset{iid}{\sim} N\left(\beta_j, \sigma_e^2\right) && i = 1, \ldots, n_j, j = 1, \ldots, 8 \\
\beta_j &\overset{iid}{\sim} N(\alpha, \sigma_\beta^2) && j = 1, \ldots, 8 \\
\alpha &\sim N(0, 1000^2) \\
\sigma_e &\sim \text{Gamma}\,(2, 0.5) \\
\sigma_\beta &\sim \text{Gamma}\,(2, 0.5)
\end{aligned} \tag{3.4}
$$

You'll notice that instead of using a $N(0, 1000^2)$ prior for all the $\beta_j$'s (as in (3.3)), in (3.4) we now learn the parameters, $\alpha, \sigma_\beta$, of this distribution from the data. Of course, because $\sigma_\beta$ is now a parameter of our model, we must place a prior distribution on it; a *hyperprior*. In this case, the effect of area is random and thus, we call them random effects (REs).

REs are extremely powerful tools in Bayesian inference. Unlike the first example in this section, which used "no-pooling", REs provide useful partial-pooling properties. By construction, random effects for small areas will be smoothed toward the mean of the prior distribution (e.g. $N\left(\alpha, \sigma_\beta^2\right)$), whereas large areas will be able to escape the pooling effect and provide REs that may be very similar to those from the no-pooling solution.

Bayesian inference automatically determines the amount of pooling that should be applied via estimation of $\sigma_\beta$. For this example, very small values of $\sigma_\beta$ would indicate that the REs are indistinguishable from the mean area effect, $\alpha$. See the box on page 25 for more details.

[15] "Familiar" as in frequentist.

Note that (3.4) can also be written as,

$$
\begin{aligned}
y_{ij} &= \alpha + \beta_j + \epsilon_{ij} && i = 1, \ldots, n_j, j = 1, \ldots, 8 \\
\beta_j &\overset{iid}{\sim} N(0, \sigma_\beta^2) && j = 1, \ldots, 8 \\
\epsilon_{ij} &\overset{iid}{\sim} N\left(0, \sigma_e^2\right) && i = 1, \ldots, n_j, j = 1, \ldots, 8 \\
\alpha &\sim N(0, 1000^2) \\
\sigma_e &\sim \text{Gamma}\,(2, 0.5) \\
\sigma_\beta &\sim \text{Gamma}\,(2, 0.5)
\end{aligned}
$$

***Consider!* Variance terms**

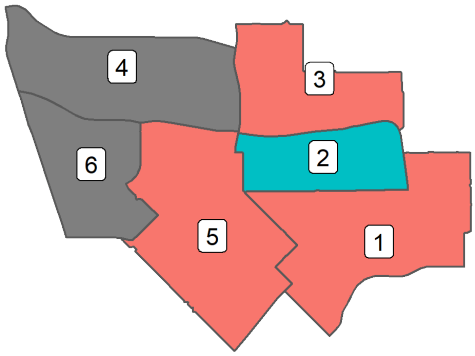
When a Bayesian hierarchical model is slow to converge, it is always good to check the estimated size of $\sigma_\beta$. If $\sigma_\beta$ is extremely small, then it may be more parsimonious to remove the random effect (RE) all together. Of course, there are formal model diagnostics one can use to help with these kinds of modelling choices.

As long as the MCMC algorithm has converged, including a RE with a very small variance will not affect the fitted values. Thus, in this work, researchers may examine the estimated values for the RE variances, but should not remove terms. This recommendation will help ensure an efficient, clear and consistent workflow for the modelling work. Note that these recommendations are aimed at the administrative and registry data. Modelling choices for the survey data are unique and are discussed in Section 5.4.

Consider the example introduced in Section 3.2. Suppose now we wish for the area-specific random effects (REs), $\beta_j$'s, to share more information if the areas are near to each and less information if the areas are far from each other geographically. The independent RE structure imposed in (3.4), borrows information globally (across the entire data) because it treats each RE, $\beta_j$, as a random draw from the distribution, $N(\alpha, \sigma_\beta^2)$, of area-specific REs. Thus, standard hierarchical models must be extended to accommodate the local smoothing/sharing of information we desire. Furthermore, if we assume that data are spatially correlated — which means we assume that data for areas near to each other will be more similar than areas far from each other — standard hierarchical models cannot create the conditional independence we require. Note that from this section onward we strictly follow the indices and notation described in Tables 1 and 2 on page 6.

## 3.3 Spatial priors

Any spatial analysis starts by defining the neighbourhood structure for the disjoint areas via a weight matrix, denoted $\mathbf{W}^{\mathrm{S}}$. Generally $\mathbf{W}^{\mathrm{S}}$ is defined via the binary specification where $W^{\mathrm{S}}_{ik} = 1$ if area $i$ and area $k$ are neighbours, and zero otherwise. Figure 7 shows a simple map of six contiguous areas, with its corresponding binary weight matrix.




$$\mathbf{W}^{\mathrm{S}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Figure 7: Example of a simple six area map and the corresponding binary contiguity weight matrix using a Queen-1 adjacency (see the box on page 26). Note that area 2 (blue) has areas 1,3,5 as neighbours (red), since it does not share a boundary with areas 4 (despite appearances) nor 6.

By specifying the neighbourhood structure in this way, we can now proceed to specify a distribution for the random effects (REs). Spatial REs are constructed to accommodate the spatial structure of the data, usually by smoothing over adjacent areas. For this reason, spatial REs are also referred to as spatially structured REs.

***Consider!* Queen vs Rook adjacency**

The binary weight matrix, $\mathbf{W}^{\mathrm{S}}$, can be defined in various ways, with the Rook or Queen adjacency being very common approaches in disease mapping applications. Rook adjacency considers an area a neighbour if at least one *side* borders the area, whilst Queen adjacency considers an area a neighbour if at least one side *or* corner borders the area. Of course, these methods can be further split according to whether only immediate neighbours will be defined as such (Queen-1) or whether neighbours of neighbours will also be considered neighbours (Queen-2) (Earnest et al. 2007). For simplicity, we recommend Queen-1 adjacency for this project.

### 3.3.1 ICAR

Let $s_i, i = 1, \ldots, M$, where $M$ is the total number of areas (e.g. HDs, LGAs or SA2s). See Table 1 for notation details. The intrinsic conditional autoregressive (ICAR) prior for a RE, $s_i$, is described by the following conditional normal distribution,

$$s_i \sim N\left(\frac{\sum_{k=1}^{M} W_{ik}^{\mathrm{S}} s_k}{m_i}, \frac{\sigma_s^2}{m_i}\right) \tag{3.5}$$

where $m_i = \sum_{k=1}^{M} W_{ik}^{\mathrm{S}}$ is the number of neighbours that area $i$ has. Under the ICAR prior, the mean of the RE, $s_i$, for area $i$ is the empirical mean of its neighbours' REs. The conditional variance of $s_i$ is the global variance, $\sigma_s^2$, divided by the number of neighbours. See Section 8.2 for an example using Figure 7. For clarity a vector of REs, $\mathbf{s} = (s_1, \ldots, s_M)$, from an ICAR prior will be denoted as

$$\mathbf{s} \sim \mathrm{ICAR}\left(\mathbf{W}^{\mathrm{S}}, \sigma_s^2\right).$$

### 3.3.2 BYM

Although the data could be highly spatially correlated, it is best practice to include both spatially structured and unstructured spatial random effects (REs). Unstructured REs do not accommodate the spatial structure and treat each area as independent of its neighbours. Without allowing for unstructured REs, areas with very high values relative to their neighbours may have a large impact on all the spatial REs. To address this issue Besag et al. (1991) proposed the well known BYM specification, where both a structured ICAR prior, $s_i$, and an unstructured standard RE, $v_i \sim N(0, \sigma_v^2)$ are used.

$$\begin{aligned} \theta_i &= s_i + v_i \\ \mathbf{s} &\sim \mathrm{ICAR}(\mathbf{W}^{\mathrm{S}}, \sigma_s^2) \\ v_i &\sim N(0, \sigma_v^2), \end{aligned} \tag{3.6}$$

### 3.3.3 BYM2

The BYM can cause significant identifiability and convergence problems, which is mostly related to the two variance parameters, $\sigma_s^2, \sigma_v^2$. Thus, more recently Riebler et al. (2016) developed the BYM2 prior, which places a single variance parameter, $\sigma_\theta^2$, on the combined components with the help of a mixing parameter, $\rho \in (0, 1)$, that represents the amount of spatially structured as opposed to unstructured residual variation. The BYM2 prior is

$$\begin{aligned} \theta_i &= \sigma_\theta \left( s_i \sqrt{\rho/\kappa} + v_i \sqrt{1-\rho} \right) \\ \mathbf{s} &\sim \text{ICAR}(\mathbf{W}^{\text{S}}, 1) \\ v_i &\sim N(0,1), \end{aligned} \tag{3.7}$$

where $\kappa$ is a scaling factor that is estimated from the weight matrix, $\mathbf{W}^{\text{S}}$, and ensures that $\sigma_\theta$ is a legitimate standard deviation. The parameter, $\rho$, is generally estimated from the data by placing a uniform prior on it. For clarity, a vector of REs, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$, from a BYM2 prior will be denoted as

$$\boldsymbol{\theta} \sim \text{BYM2}\left(\mathbf{W}^{\text{S}}, \rho, \kappa, \sigma_\theta^2\right).$$

### 3.3.4 Leroux

Another common prior used to accommodate both structured and unstructured spatial variation is that of Leroux et al. (2000). Similar to the BYM2 prior the Leroux prior uses a single RE that can model a mixture of structured and unstructured spatial variation.

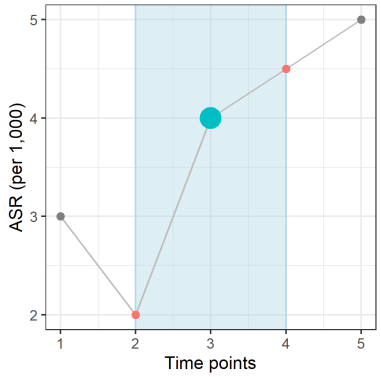
$$\theta_i \sim N\left( \frac{\rho \sum_k W_{ik}^{\text{S}} \theta_k}{\rho \sum_k W_{ik}^{\text{S}} + 1 - \rho}, \frac{\sigma_\theta^2}{\rho \sum_k W_{ik}^{\text{S}} + 1 - \rho} \right) \tag{3.8}$$

This mixture representation comprises of uncorrelated smoothing to a global mean of zero (weighted by $1-\rho$) as well as correlated smoothing of the nearby REs (weighted by $\rho$). Note that when $\rho = 0$, the Leroux prior collapses to an independent standard normal prior, while $\rho = 1$ gives the ICAR prior. For conciseness, a vector of REs, $\boldsymbol{\theta}$, that come from a Leroux prior will be denoted as,

$$\boldsymbol{\theta} \sim \text{Leroux}\left(\mathbf{W}^{\text{S}}, \rho, \sigma_\theta^2\right)$$

## 3.4 Temporal Priors

Spatial and temporal priors generally follow similar construction. The key difference is that spatial priors must accommodate two-dimensions (longitude and latitude), while temporal priors need only one-dimension. By altering the weight matrix accordingly, the spatial priors introduced above can also be used for temporal settings. Figure 8 illustrates a simple five time point example with the corresponding temporal weight matrix, $\mathbf{W}^{\text{T}}$.




$$\mathbf{W}^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 8: Example of how temporal random effects share information locally. The corresponding temporal weight matrix shows how each time point's neighbours are a single time point before and after. Consider time point 3 (highlighted using a large blue dot). It borrows information from time points 2 and 4 (red dots) which are a single point before and after (shaded in blue) time point 3.

### 3.4.1 RW1

Let $\gamma_t, t = 1, \ldots, T$ be the temporal random effect (RE) for time point $t$ and $T$ be the total number of time points in the data (see Table 1). In this project, the time points are years. The temporal REs can be modelled using the ICAR prior,

$$\boldsymbol{\gamma} \sim \text{ICAR}\left(\mathbf{W}^{\mathrm{T}}, \sigma_\gamma^2\right). \tag{3.9}$$

A temporal prior of this kind is commonly referred to as a random walk of order 1 (RW1) (Haining and Li 2020). The intuition is identical to before.

For example, the conditional mean and variance of $\gamma_3$ is the mean of $\gamma_2, \gamma_4$ and $\sigma_\gamma^2/2$, respectively (see Figure 8). Note that the BYM2 and Leroux spatial priors introduced above can also be used for temporal REs by simply using the appropriate temporal weight matrix, $\mathbf{W}^{\mathrm{T}}$, but this is rare, and we use the RW1 throughout this report.

## 3.5 Space-time interaction priors

The temporal and spatial random effects (REs) cannot capture any variation specific to one area at one-time point. For example, consider a particular area that generally has a low rate of a given disease, but for some reason has an extremely high rate for one of the years in the data. Without including some form of space-time interaction, this outlier could alter the temporal and spatial REs and smooth neighbouring time points and areas in undesirable ways. There are a variety of possible REs that can be used to allow for space-time variation (Knorr-Held 2000). For parsimony, we recommend using a standard normal distribution for the space-time interaction RE, which assumes independent variation.

## 3.6 Spatio-temporal models

The most generic spatio-temporal model can be written in the following likelihood + prior form. Given that a large quantity of DOHWA data are given as raw counts, we present a generic Poisson model in (3.10) below.

Let $y_{ita}$ and $N_{ita}$ be the raw counts and population size, respectively, for age group $a$ ($a = 1, \ldots, A$), area $i$ and time $t$. In addition, let $\mathbf{X}_{ita} \in \mathbb{R}^{1\times(A-1)}$ be the design matrix of indicators for the $A$ age groups and $\boldsymbol{\beta} \in \mathbb{R}^{(A-1)\times 1}$ their respective regression coefficients. A design matrix is a condensed matrix formulation that represents multiple fixed effects (e.g. covariates). The design matrix, $\mathbf{X}_{ita}$, can also include any adjustment factors such as SEIFA and remoteness (see the box on page 43). The fitted or smoothed counts are given by $\mu_{ita}$. Please refer to Section 8.1 and Tables 1 and 2 for notation help.

We use a BYM2 prior for the spatial RE, an ICAR prior for the temporal RE and a standard normal distribution for the space-time interaction RE. We place reasonably weakly informative gamma priors on the variance terms, uninformative normal distributions on the regression coefficients and a uniform prior on $\rho$. As highlighted in Section 3.1 the priors on the regression coefficients are extensively used.

The priors used for variance terms vary widely in the literature, for example Urdangarin et al. (2022) use a Gamma$(1, 0.01)$ prior on the precision,[16] $\sigma_\delta^{-2}$, which implies a highly informative prior on $\sigma_\delta$, whilst Lawson (2020) uses a uniform prior with an arbitrary cutoff of 10, Uniform$(0, 10)$. Following recommendations by the `stan` community, found here, the Gamma prior used below balances pushing density away from zero, whilst providing a long tail to make the distribution relatively uninformative.

$$
\begin{aligned}
y_{ita} &\sim \text{Poisson}\left(\mu_{ita}\right) \\
\log\left(\mu_{ita}\right) &= \log\left(N_{ita}\right) + \alpha + \mathbf{X}_{ita}\boldsymbol{\beta} + \theta_i + \gamma_t + \delta_{it} \\
\delta_{it} &\sim N(0, \sigma_\delta^2) \\
\boldsymbol{\theta} &\sim \text{BYM2}\left(\mathbf{W}^{\text{S}}, \rho, \kappa, \sigma_\theta^2\right) \\
\boldsymbol{\gamma} &\sim \text{ICAR}\left(\mathbf{W}^{\text{T}}, \sigma_\gamma^2\right) \\
\rho &\sim \text{Uniform}(0, 1) \\
\sigma_\theta, \sigma_\gamma, \sigma_\delta &\sim \text{Gamma}(2, 0.5) \\
\alpha, \boldsymbol{\beta} &\sim N(0, 1000^2)
\end{aligned}
\tag{3.10}
$$

By including the offset term, $\log\left(N_{ita}\right)$, we are implicitly modeling the fitted rate for age $a$, area $i$ and time $t$. See Section 8.4 for more details.

[16] The precision is the inverse of the variance, $\tau_\delta = \frac{1}{\sigma_\delta^2}$. Although some Bayesian software (Lunn et al. 2000; Plummer 2003) defaults to putting priors on the precision terms, it is preferable to place priors instead on standard deviations as these have a more convenient interpretation.

# 4 Administrative data

The administrative data include mortality (overall and avoidable deaths and alcohol and other drugs (AOD) related deaths), emergency department (ED) attendances (overall and GP-type ED attendances), hospitalisations (overall and potentially preventable hospitalisations, AOD related hospitalisations, injury and poisoning related hospitalisations), notifiable communicable diseases and cancer incidence. Regardless of the condition, these administrative data are reported as raw counts and are thus modelled as such.

We recognise two key metrics that should be reported from these data (formula for the epidemiology metrics discussed in this report can be found in Section 8.3). The first metric is the area-by-year standardised incidence ratios (SIRs) (Section 4.1), which are calculated by dividing the observed counts by the expected counts in each area and year. An SIR of 1 indicates that the incidence in a particular area is similar to that of the state. The SIRs derived from the models are equivalent to standardised rate ratios (SRRs) (for the hospitalisation, ED and notifiable communicable disease data) and standardised mortality ratios (SMRs) (for the mortality data).

The second metric is ASRs (Section 4.2), which involves direct standardisation of the area, year and age (AYA) counts to the 2001 Australian Standard Population.

Modelling or smoothing administrative data across areas and years requires both population estimates and raw counts by area and year. Poisson models are used extensively in the field of disease mapping to model raw counts with a necessary offset term (see (3.10)); making them a great choice for the ASR and SIR-type models required in this project. The difference between the ASR and SIR-type models is the definition of the offset term (see Table 4). Two versions of ASR models are provided — the ASR_ST and ASRA_ST models [17] — where the ASRA_ST model includes age groups within the model. Figure 11 compares the posterior ASRs from the ASR_ST and ASRA_ST models to the raw ASRs.

We acknowledge that metrics are required by sex. Although age-period models include sex and age in the same model (Riebler and Held 2017), unless otherwise stated, for this project we recommend fitting separate models for males, females and persons.

| | **Input data by** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Model** | Area | Year | Age | **Input data** | **Offset term** | **Key model output calculation** ‡ | **Software** | **Code** | **Eq.** |
| **SIR_ST** | ✓ | ✓ | | Counts | Expected counts | Fitted counts ÷ offset | `CARBayesST` | 3 | (4.1) |
| **ASR_ST** | ✓ | ✓ | | Counts | Counts ÷ ASRs | Fitted counts ÷ offset | `CARBayesST` | 4 | (4.3) |
| **ASRA_ST** | ✓ | ✓ | ✓ | Counts | Population | Fitted counts (then calculate ASR) | `nimble` | 5 | (4.5) |

Table 4: Summary of models for administrative data. ‡ Grey-coloured text denotes the calculations carried out on the fitted counts to derive the core metrics. For more details see Sections 4.1.1, 4.2.1, 4.2.2. Approximate run time for these models is of the order of days to weeks (ASRA_ST) or minutes (SIR_ST, ASR_ST).

[17] ASRA (Age-Standardised Rate with Age) is *not* a epidemiology metric but a model identifier we have *created* to help differentiate the three recommended administrative ST models.

## 4.1 Standardised incidence ratio (SIR)

The most common metric reported in disease mapping applications is the SIR (Cramb et al. 2020). Since an SIR is the observed counts divided by the expected counts, when modelling SIRs, the offset term is the expected counts (Lee 2011). See the box on page 32. This generic style of ST model will be denoted as the SIR_ST model hereafter.

***Consider!* Expected counts**

The expected counts for the SIR_ST model are derived by applying the overall age-specific rate (across all years and areas) to the known age-specific populations in each area and year. This means that the SIRs describe the ratio of the current area-by-time counts to the overall expected counts, which allows examination of temporal trends.

Alternative approaches derive overall age-rates for each year. In this case, the SIRs describe the ratio of the current area-by-time counts to the time-specific expected counts. Note that this approach prevents us from obtaining temporal trends.

### 4.1.1 Model: SIR_ST

The SIR_ST model requires data by area and year as illustrated in Table 5. The model can be fitted using the R package `CARBayesST`, which uses efficient MCMC to fit Bayesian ST models in R (Lee et al. 2022). Unfortunately this package does not naturally allow the user to run multiple chains. We have written a wrapper function that automatically runs four chains[18] behind the scenes, returning useful output, including model convergence metrics (see Code 3).

| y | E | M_id | T_id | LGA | year | N |
|---|---|---|---|---|---|---|
| $y_{it}$ | $E_{it}$ | $i$ | $t$ | | | |
| 3 | 6.72 | 1 | 1 | 50080 | 2011 | 17807 |
| 9 | 8.88 | 2 | 1 | 50210 | 2011 | 32611 |
| 2 | 0.43 | 3 | 1 | 50250 | 2011 | 3628 |
| 0 | 1.86 | 4 | 1 | 50280 | 2011 | 6133 |
| 1 | 2.57 | 5 | 1 | 50350 | 2011 | 7558 |
| 10 | 11.15 | 6 | 1 | 50420 | 2011 | 33210 |
| 13 | 6.13 | 7 | 1 | 50490 | 2011 | 18318 |
| 0 | 0.37 | 8 | 1 | 50560 | 2011 | 782 |
| 0 | 0.20 | 9 | 1 | 50630 | 2011 | 866 |
| 0 | 0.34 | 10 | 1 | 50770 | 2011 | 818 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Table 5: Example data structure for the SIR_ST model. `M_id` is a sequential identifier for the areas, while `T_id` is a sequential identifier for the time periods. In the table, $y_{it}$ and $E_{it}$ are the raw and expected counts for area $i$ and year $t$ (see (4.1) for details).

[18] Please review Section 2.1 for details on why multiple chains are used.

```r
# Use the wrapper function to fit 4 chains using CARBayesST
SIR_model <- jf$SampleCBST(y ~ offset(log(E)),
                           # Number of MCMC samples to draw for each chain
                           n.sample = 2500,
                           # burn-in
                           nburnin = 1250,
                           # amount to thin by
                           thin = 1,
                           # define the dataset
                           data = df,
                           # binary contiguity weight matrix
                           W = W,
                           # area and year variables in df
                           area = "M_id",
                           year = "T_id",
                           # offset term as a numeric vector
                           ofs = df$E,
                           # observed count as a numeric vector
                           y = df$y)
```

R Code 3: Example code to fit the SIR_ST model.

The model fitted using the `jf$SampleCBST()` function in R (see Code 3) is specified below. Let $y_{it}$ and $E_{it}$ be the raw and expected counts for area $i$ and year $t$. See Section 8.3 for calculation of $E_{it}$. The SIR_ST model uses a Leroux prior for the vector of spatial random effects (REs), $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M) \in \mathbb{R}^M$, an ICAR prior for the temporal REs, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_T) \in \mathbb{R}^T$, and a normal distribution for the space-time interaction REs, $\boldsymbol{\delta} \in \mathbb{R}^{MT}$. Please review Section 8.1 for help with this notation.

$$
\begin{aligned}
y_{it} &\sim \text{Poisson}(\mu_{it}) \\
\log(\mu_{it}) &= \log(E_{it}) + \alpha + \theta_i + \gamma_t + \delta_{it} \\
\boldsymbol{\theta} &\sim \text{Leroux}\left(\mathbf{W}^{\text{S}}, \rho, \sigma_\theta^2\right) \\
\boldsymbol{\gamma} &\sim \text{ICAR}\left(\mathbf{W}^{\text{T}}, \sigma_\gamma^2\right) \\
\delta_{it} &\sim N(0, \sigma_\delta^2) \\
\rho &\sim \text{Uniform}(0, 1) \\
\alpha &\sim N(0, 1000^2) \\
\sigma_\theta^2, \sigma_\gamma^2, \sigma_\delta^2 &\sim \text{InvGamma}\left(\text{shape} = 1, \text{scale} = 0.01\right)
\end{aligned}
\tag{4.1}
$$

The wrapper function, `jf$SampleCBST()`, returns the posterior draws for the smoothed SIRs, which are calculated by performing the following computation for each posterior draw $d$,

$$SIR_{it}^{(d)} = \frac{\mu_{it}^{(d)}}{E_{it}}. \tag{4.2}$$

The matrix of posterior draws for the fitted counts and SIRs can be accessed by `SIR_model$fitted_draws` and `SIR_model$rate_draws`, respectively.

***Consider!* Population and spatial smoothing**

The output of disease mapping models adapt to the population size in each area and year (i.e. the offset term). That is they provide more smoothing to those areas and years with small populations and less to those with large populations. Figure 9 illustrates this behaviour at the health district (HD) level. Observe that the SIRs for HDs with very large populations are not smoothed by the Bayesian ST model (e.g. the raw and modelled SIRs agree almost perfectly). In general, the smoothed and raw SIRs become increasingly different as the population size decreases.

## 4.2 Age-standardised rates (ASR)

### 4.2.1 Model: ASR_ST

The age-standardised model by year and area, denoted as the ASR_ST model, requires the raw counts for the condition of interest and an offset term given by the raw counts divided by the ASRs, denoted as `N_tilde` in Table 6. Use of this offset term ensures that the Poisson model is implicitly modelling the ASRs rather than the crude rates (see Section 8.6 for proof of this statement). The ASR_ST model can also be fitted using the R package `CARBayesST`. The data structure (Table 6) is identical to that required for the SIR_ST model. Note that ASRs are age-standardised to the Australian standard population 2001.

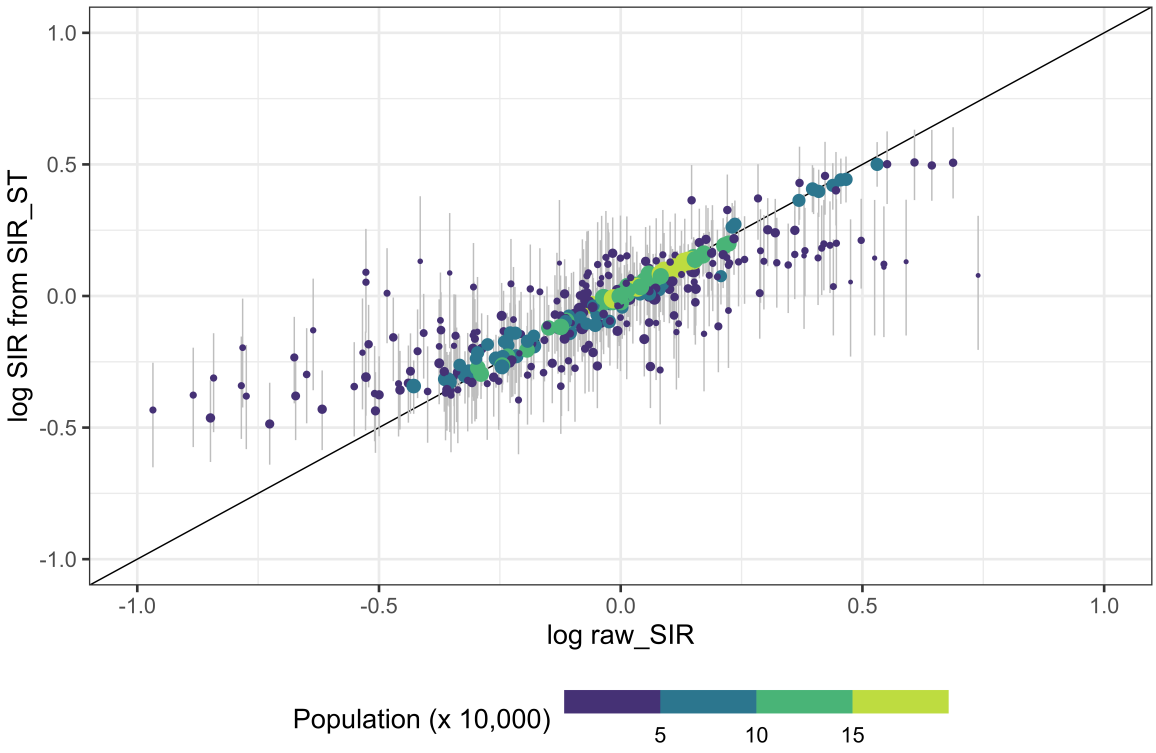



Figure 9: Plots comparing the raw and modelled log SIRs at the health district level. We use the SIR_ST model to obtain the modelled results. Each point is the posterior median of the SIRs from the SIR_ST model with corresponding 95% quantile credible intervals. The points are coloured according to the HDs population size. Larger points also denote larger population sizes. The diagonal line denotes perfect agreement between the log raw SIR and the modelled log SIR.

| y | N_tilde | M_id | LGA | year | T_id | N | ASR |
|---|---|---|---|---|---|---|---|
| $y_{it}$ | $\tilde{N}_{it}$ | $i$ | | $t$ | | | |
| 3 | 20078 | 1 | 50080 | 2011 | 1 | 17807 | 0.000149 |
| 9 | 31431 | 2 | 50210 | 2011 | 1 | 32611 | 0.000286 |
| 2 | 5662 | 3 | 50250 | 2011 | 1 | 3628 | 0.000353 |
| 0 | 6133 | 4 | 50280 | 2011 | 1 | 6133 | 0 |
| 1 | 8643 | 5 | 50350 | 2011 | 1 | 7558 | 0.000116 |
| 10 | 38984 | 6 | 50420 | 2011 | 1 | 33210 | 0.000257 |
| 13 | 22034 | 7 | 50490 | 2011 | 1 | 18318 | 0.000590 |
| 0 | 782 | 8 | 50560 | 2011 | 1 | 782 | 0 |
| 0 | 866 | 9 | 50630 | 2011 | 1 | 866 | 0 |
| 0 | 818 | 10 | 50770 | 2011 | 1 | 818 | 0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Table 6: Example data structure for the ASR_ST model. In the table, $y_{it}$ and $\tilde{N}_{it}$ are the raw counts and adjusted populations for area $i$ and year $t$ (see (4.3) for details). Observe that in some cases $\tilde{N}_{it}$ can be *very* different to N, which is related to the offset adjustment described in Section 8.6.

```r
# Use the wrapper function to fit 4 chains using CARBayesST
ASR_model <- jf$SampleCBST(y ~ offset(log(N_tilde)),
                           # Number of MCMC samples to draw for each chain
                           n.sample = 2500,
                           # burn-in
                           nburnin = 1250,
                           # amount to thin by
                           thin = 1,
                           data = df,
                           # binary contiguity weight matrix
                           W = W,
                           # area and year variables in df
                           area = "M_id",
                           year = "T_id",
                           # offset term as a numeric vector
                           ofs = df$N_tilde,
                           # observed count as a numeric vector
                           y = df$y)
```

R Code 4: Example code to fit the ASR_ST model.

The model fitted using the `jf$SampleCBST()` function in R (see Code 3) is specified below. Let $y_{it}$ and $\tilde{N}_{it}$ be the raw counts and adjusted populations for area $i$ and year $t$. Please see Section 8.6 for a description of the adjusted population. The ASR_ST model is specified identically to the SIR_ST model, except for the different offset term.

$$
\begin{aligned}
y_{it} &\sim \text{Poisson}(\mu_{it}) \\
\log(\mu_{it}) &= \log\left(\tilde{N}_{it}\right) + \alpha + \theta_i + \gamma_t + \delta_{it} \\
\boldsymbol{\theta} &\sim \text{Leroux}\left(\mathbf{W}^{\text{S}}, \rho, \sigma_\theta^2\right) \\
\boldsymbol{\gamma} &\sim \text{ICAR}\left(\mathbf{W}^{\text{T}}, \sigma_\gamma^2\right) \\
\delta_{it} &\sim N(0, \sigma_\delta^2) \\
\rho &\sim \text{Uniform}(0, 1) \\
\alpha &\sim N(0, 1000^2) \\
\sigma_\theta^2, \sigma_\gamma^2, \sigma_\delta^2 &\sim \text{InvGamma}\,(1, 0.01)
\end{aligned}
\tag{4.3}
$$

As we saw for the SIR_ST model, we can derive the posterior draws for the rates by performing the following computation for each posterior draw $d$,

$$ASR_{it}^{(d)} = \frac{\mu_{it}^{(d)}}{\tilde{N}_{it}}. \tag{4.4}$$

The posterior draws of the ASRs provided by `ASR_model$rate_draws` are rates per individual. Multiplying *all* draws by $10,000$, for example, would provide posterior draws for the ASRs per $10,000$ people.

### 4.2.2 Model: ASRA_ST

Since estimates by age are required for reporting, we can also run a variant of the above model that aggregates the raw counts by area, year and the desired age groups. This model is denoted as the ASRA_ST model (see Table 4). An example of the data structure can be found in Table 7, which has 6 age groups. Observe that the input data has six times the number of rows than those data used for the SIR_ST and ASR_ST models. In the ASRA_ST model the input data are raw counts, the offset is the population (Jay et al. 2021), and age group is included as a covariate.

| y | N | age | M_id | T_id | MT_id | year | LGA |
|---|---|---|---|---|---|---|---|
| $y_{ita}$ | $N_{ita}$ | $a$ | $i$ | $t$ | | | |
| 0 | 1065 | 0-4 years | 1 | 1 | 1 | 2011 | 50080 |
| 0 | 2348 | 5-14 years | 1 | 1 | 1 | 2011 | 50080 |
| 0 | 2092 | 15-24 years | 1 | 1 | 1 | 2011 | 50080 |
| 1 | 4164 | 25-44 years | 1 | 1 | 1 | 2011 | 50080 |
| 0 | 4884 | 45-64 years | 1 | 1 | 1 | 2011 | 50080 |
| 2 | 3254 | 65+ years | 1 | 1 | 1 | 2011 | 50080 |
| 1 | 2428 | 0-4 years | 2 | 1 | 11 | 2011 | 50210 |
| 0 | 4246 | 5-14 years | 2 | 1 | 11 | 2011 | 50210 |
| 0 | 4855 | 15-24 years | 2 | 1 | 11 | 2011 | 50210 |
| 0 | 9217 | 25-44 years | 2 | 1 | 11 | 2011 | 50210 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Table 7: Example dataset for the ASRA_ST model. In the table, $y_{ita}$ and $N_{ita}$ are the raw counts and populations for age $a$, area $i$ and year $t$, respectively (see (4.5) for details). Notice how the dataset required for the ASRA_ST model will have $A \times M \times T$ rows.

The age standardisation is calculated afterwards from the area, year and age (AYA) fitted counts to produce a smoothed ASR by area and year. Of course, this process is applied to all posterior draws of the fitted counts. The ASRA_ST model can be fitted using the R package `nimble` (Numerical Inference for statistical Models using Bayesian and Likelihood Estimation) [3], where we first declare the model using BUGS syntax.

The BUGS syntax (which is read by `nimble`) required to fit the ASRA_ST model can be found in Code 5 below. Unlike `CARBayesST` (used for fitting the ASR_ST and SIR_ST models), `nimble` requires a very specific structure of input data; these details will be described in the training. Like above, we have written an R wrapper function that simplifies several

steps of modelling with `nimble`, including reporting the MCMC diagnostics and warnings (see Section 2.1). An example of the wrapper function can be found in Code 6.

```
code <- nimbleCode({
    # iterate across all the rows of the data (n_obs)
    for(i in 1:n_obs){
        # likelihood
        y[i] ~ dpois(mu[i])
        # mean - linear predictor
        log(mu[i]) <- log(N[i]) + alpha
        # Fixed effects using the inner product
        + inprod(B_qr[1:q], Q_ast[i,])
        # BYM2 spatial term
        + theta[M_id[i]]
        # ICAR temporal term
        + gamma[T_id[i]]
        # Space time term
        + delta[MT_id[i]]
    }

    # Spatial: BYM2 #
    # iterate across all areas (M)
    for(i in 1:M){
        theta[i] <- sigma_theta * (s[i] + v[i])
        # structured component
        s[i] <- Z_s[i] * sqrt(rho/kappa)
        # unstructured component
        v[i] <- Z_v[i] * sqrt((1 - rho))
        # standard normal
        Z_v[i] ~ dnorm(0, 1)
    }

    # ICAR prior #
        Z_s[1:M] ~ dcar_normal(adj[1:L_s],
                               weights[1:L_s],
                               num[1:M],
                               1, # unit-variance ICAR
                               # enforce sum-to-zero
                               zero_mean = 1)

    # Space time interaction #
```

```
        # iterate across all time points and areas (MT)
        for(i in 1:MT){
            Z_delta[i] ~ dnorm(0,1)
            # multiply by sigma_delta
            delta[i] <- sigma_delta * Z_delta[i]
        }

    # RW1 prior #
        Z_gamma[1:T] ~ dcar_normal(T_adj[1:L_t],
                                   T_weights[1:L_t],
                                   T_num[1:T],
                                   1, # unit-variance ICAR
                                   # enforce sum-to-zero
                                   zero_mean = 1)
        for(i in 1:T){
            # multiply by sigma_gamma
            gamma[i] <- sigma_gamma * Z_gamma[i]
        }

    # Other priors #
        # iterate over all elements of B_qr
        for(i in 1:q){
            B_qr[i] ~ dnorm(0, sd = 1000)
        }
        alpha ~ dnorm(0, sd = 1000)
        sigma_theta ~ dgamma(2, 0.5)
        sigma_gamma ~ dgamma(2, 0.5)
        rho ~ dunif(0,1)

    # recreate true coefficient values
        B[1:q] <- R_ast_inverse[1:q,1:q] %*% B_qr[1:q]

})
```

R Code 5: Example BUGS syntax to fit the ASRA_ST model.

The model fitted using Code 5 is given in (4.5) on page 40. Assume access to the raw count, $y_{ita}$, and population size, $N_{ita}$, for age category $a$, in area $i$ and time $t$. Let $\mathbf{X}_{ita} \in \mathbb{R}^{1\times(A-1)}$ be the design matrix of indicators for the $A$ age groups and $\boldsymbol{\beta} \in \mathbb{R}^{(A-1)\times 1}$ their respective regression coefficients. Please see the box on page 41 for details in setting an

appropriate reference age group. Although not explicitly defined in (4.5) below, we use the QR decomposition (Section 8.8) and non-mean centered parameterisation (Section 8.5) in Code 5.

To make the connection between Code 5 and the model definition explicit, we index the likelihood + prior form given below with the corresponding code lines.

**Likelihood**: line 5

$$y_{ita} \sim \text{Poisson}\left(\mu_{ita}\right) \tag{4.5}$$

**Linear predictor**: lines 7–15

$$\log\left(\mu_{ita}\right) = \log\left(N_{ita}\right) + \alpha + X_{ita}\boldsymbol{\beta} + \theta_i + \gamma_t + \delta_{ti}$$

**Spatial RE**: lines 20–36

$$\boldsymbol{\theta} \sim \text{BYM2}\left(\mathbf{W}^{\text{S}}, \rho, \kappa, \sigma_\theta^2\right)$$

**Temporal RE**: lines 47–56

$$\boldsymbol{\gamma} \sim \text{ICAR}(\mathbf{W}^{\text{T}}, \sigma_\gamma^2)$$

**Space-time RE**: lines 40–44

$$\delta_{ti} \sim N\left(0, \sigma_\delta^2\right)$$

**Priors**: lines 60–66

$$\begin{aligned} \rho &\sim \text{Uniform}(0,1) \\ \sigma_\theta, \sigma_\gamma, \sigma_\delta &\sim \text{Gamma}(2, 0.5) \\ \alpha, \boldsymbol{\beta} &\sim N(0, 1000^2) \end{aligned}$$

After fitting the ASRA_ST model, the $d$th posterior draw for the 'smoothed' or modelled ASRs is given by

$$\widehat{ASR}_{it}^{(d)} = \sum_a \frac{\mu_{ita}^{(d)} N_a^{2001}}{N_{ita}}. \tag{4.6}$$

***Tech Talk!* Reference age level**

If convergence is very poor (particularly for the fixed effects), we recommend explicitly specifying the reference age group. In most cases using an age category with many counts can help.

When poor convergence is related to the fixed effects, this is most likely due to sampling of $\alpha$, which can be empirically estimated as

$$\widehat{\alpha} = \log\left(\frac{\sum_i \sum_t y_{it\tilde{a}}}{\sum_i \sum_t N_{it\tilde{a}}}\right),$$

where $\tilde{a}$ is the reference age group. If the reference age group has no or very few counts across all years and areas then $\widehat{\alpha} \to -\infty$. Given that all the other fixed effects are constructed as comparisons to the reference group, models such as these will fail to converge to reasonable values (i.e. will roam around the parameter space and never find any area of density). For example, with a poorly chosen reference category, we can recover coefficients as high as $17$, which are interpreted as a rate increase of around $56$ million! In cases where MCMC is struggling to converge, we recommend trying a frequentist run with just the fixed effects (i.e. dropping the spatio-temporal terms), to determine if the Bayesian fixed effect estimates are plausible. This can be achieved by running, `glm(y ~ offset(log(N)) + as.factor(age), data = df, family = "poisson")`. For other tricks to help with convergence see Section 2.1.

```r
ASRA_model <- jf$SampleNimble(  # BUGS code
                                code = code,
                                # data list
                                nD = nD,
                                # initial value function
                                nI = nI(),
                                # constant list
                                nC = nC,
                                # parameters to monitor
                                monitors = monitors,
                                # total iterations per chain
                                niter = 4000,
                                # burn-in per chain
                                nburnin = 2000,
                                thin = 20,
                                nchains = 4,
                                # check samplers are correct
                                print_samplers = T,
                                # optimize sampling
                                # of the fixed effects
                                optimBeta = T,
                                beta_name = "B_qr",
                                # use an RW_block sampler
                                sampler_name = "RW_block",
                                # decrease adaption during burn-in
                                adaptInterval = 10 # defaults to 200)
```

R Code 6: Example code to use the wrapper function, `jf$SampleNimble(.)`, to fit models in `nimble`.

***Consider!* Adjusted analysis**

There is interest in this project to present adjusted and unadjusted estimates, where adjusted estimates are derived from models which include covariates for remoteness and area-level socioeconomic status. These covariates are only available every 5 years with the census, and in this project 2016 indices will be used in the adjusted models (i.e. kept constant across time). Since these are area-level covariates, they can readily be incorporated into any of the models discussed above. Unadjusted estimates are not adjusted for both remoteness and SEIFA. The calculation of raw SIRs are described in Section 8.3.

One of the benefits of smoothing administrative data using unadjusted models is that estimates from each area and year can only learn from their neighbouring areas and years, resulting in smooth maps. Alternatively, adjusted ST models allow particular areas and years to differ significantly from the overall ST trends according to socioeconomic status and remoteness.

Interpreting the differences in model estimates from the adjusted and unadjusted models is difficult as both models attempt to approximate the data (albeit in different ways). There are a plethora of metrics that can describe the level of ST variation that is explained by remoteness and socioeconomic status. Please see Duncan and Mengersen (2020) for more details.

To fit an adjusted ASRA_ST model, the desired variables (e.g. remoteness and/or socioeconomic indices) must be included in the design matrix, $\mathbf{X}$, for the fixed effects along with the age group variable.

Figure 10 compares the SIR from a spatial only model. Observe that both the posterior median SIRs and their standard deviations are similar, but not identical. The Perth specific map displays almost identical SIRs across LGAs, while the WA plot displays some key differences (particularly in the Pilbara and Midwest). This is most likely related to the smaller populations in these areas resulting in more model smoothing occurring (see the box on page 34).

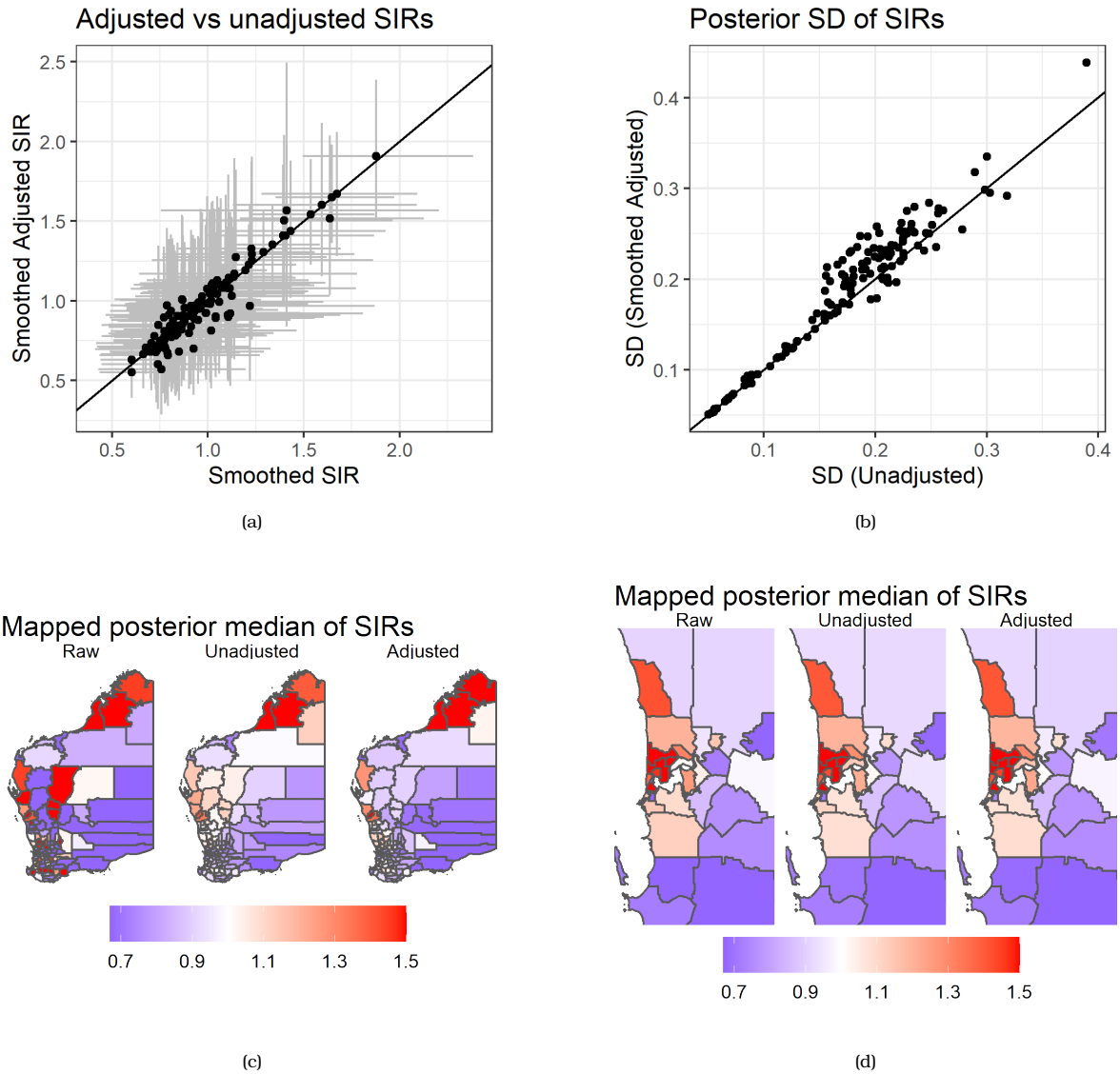



Figure 10: This plot compares SIRs at the LGA level which are derived from adjusted and unadjusted SIR models (spatial only). The data in this plot are the counts of ear, nose and throat infections for females in 2015. Plot (a) compares the posterior medians of the smoothed SIRs along with their 95% credible intervals. Plot (b) compares the posterior standard deviations of the smoothed SIRs. The diagonal black lines in (a) and (b) represents equivalence between the $x$ and $y$ axes. Plots (c) and (d) compare the raw, unadjusted and adjusted SIRs for all of Western Australia and greater Perth, respectively.

## 4.3 Data sparsity

In some cases, both the SIR and ASR models may be impacted by sparsity (i.e. when many rows of the input data have very small or zero raw counts). Of course, the ASRA_ST model is more prone to these issues given that its input data is of higher resolution. We are hesistant to suggest *any* rules of thumb for when data are considered as sparse as cutoffs should be dependent on the model, data and objective of the analysis. That said, the impact of sparsity on model fit can be identified by conducting the model checks recommended in Section 2.3.

Our empirical explorations suggest that ST models can still provide similar performance in these severely sparse settings. An example is shown in Figure 5 which illustrates posterior predictive checks for an ASRA_ST model fitted to data where approximately 95% of the rows have zero counts. Note that for the ASRA_ST model, our recommendation assumes that an appropriate reference age group has been chosen. See the box on page 41 for more details.

It is possible to run more complex models that can handle large amounts of zeroes (Corpas-Burgos et al. 2018; Neelon et al. 2014), but a general recommendation is to reduce the resolution of the data in some dimension, such as by aggregating time periods or, for the ASRA_ST model, further aggregating age groups, and then fitting the ST models discussed above. The amount of information included when smoothing can also be modified by tweaking the priors so that instead of smoothing over adjacent time periods, it smooths over the two previous and subsequent time periods (i.e. a random walk of order 2) (Haining and Li 2020).

Alternatively, we suggest a simple reparameterisation where the design matrix in (4.5) is replaced by indicators for the age groups directly.[19]

$$
\begin{aligned}
y_{ita} &\sim \text{Poisson}\left(\mu_{ita}\right) \\
\log\left(\mu_{ita}\right) &= \log\left(N_{ita}\right) + \alpha + \beta_a + \theta_i + \gamma_t + \delta_{it} \\
\beta_a &\sim N(0, 1000^2) \\
\delta_{it} &\sim N(0, \sigma_\delta^2) \\
\boldsymbol{\theta} &\sim \text{BYM2}\left(\mathbf{W}^{\text{S}}, \rho, \kappa, \sigma_\theta^2\right) \\
\gamma &\sim \text{ICAR}\left(\mathbf{W}^{\text{T}}, \sigma_\gamma^2\right) \\
\rho &\sim \text{Uniform}(0, 1) \\
\sigma_\theta, \sigma_\gamma, \sigma_\delta &\sim \text{Gamma}(2, 0.5) \\
\alpha &\sim N(0, 1000^2)
\end{aligned}
\tag{4.7}
$$

We found that for severely sparse data, this parameterisation can offer computational advantages. However, note that this parameterisation is weakly unidentifiable[20] and thus,

[19] See Section 8.1.2 of McElreath (2020) for more details.

[20] *Unidentifiability* is a modelling term that refers to situations where the data alone cannot distinguish between

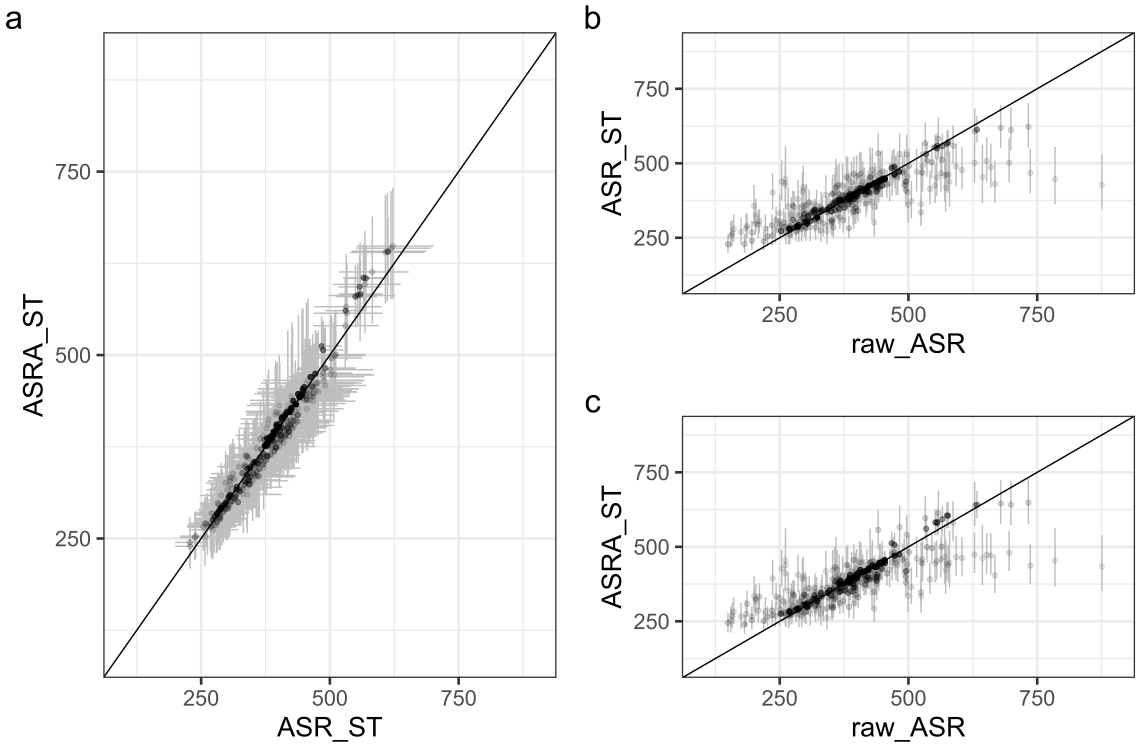



Figure 11: Plots comparing raw and modelled age-standardised rates from the ASR_ST and ASRA_ST models. The transparency of the points denotes the area-by-year population. The diagonal line denotes equality between the $x$ and $y$ axes. The error bars denote the 95% credible intervals of the posterior ASRs. The ASRs in these plots are for the potentially preventable hospitalisations of ear, nose and throat infections for females. The data is at the health district level from 2011-2020. Plot (a) compares the posterior ASRs from the ASRA_ST and ASR_ST. Observe the models' very strong agreement. Plots (b) and (c) compare the posterior ASRs from the ASR_ST and ASRA_ST models to the raw ASRs. Note that both models smooth the ASRs relative to the population size of the area and time point. In this example, the ASRA_ST model took around 6 times longer to fit than the ASR_ST model.

should *only* be used in severely sparse settings, or when the standard ASRA_ST model fails to converge (even after applying the convergence tricks in Section 2.1).

model parameters, which can make inference and computation more difficult.

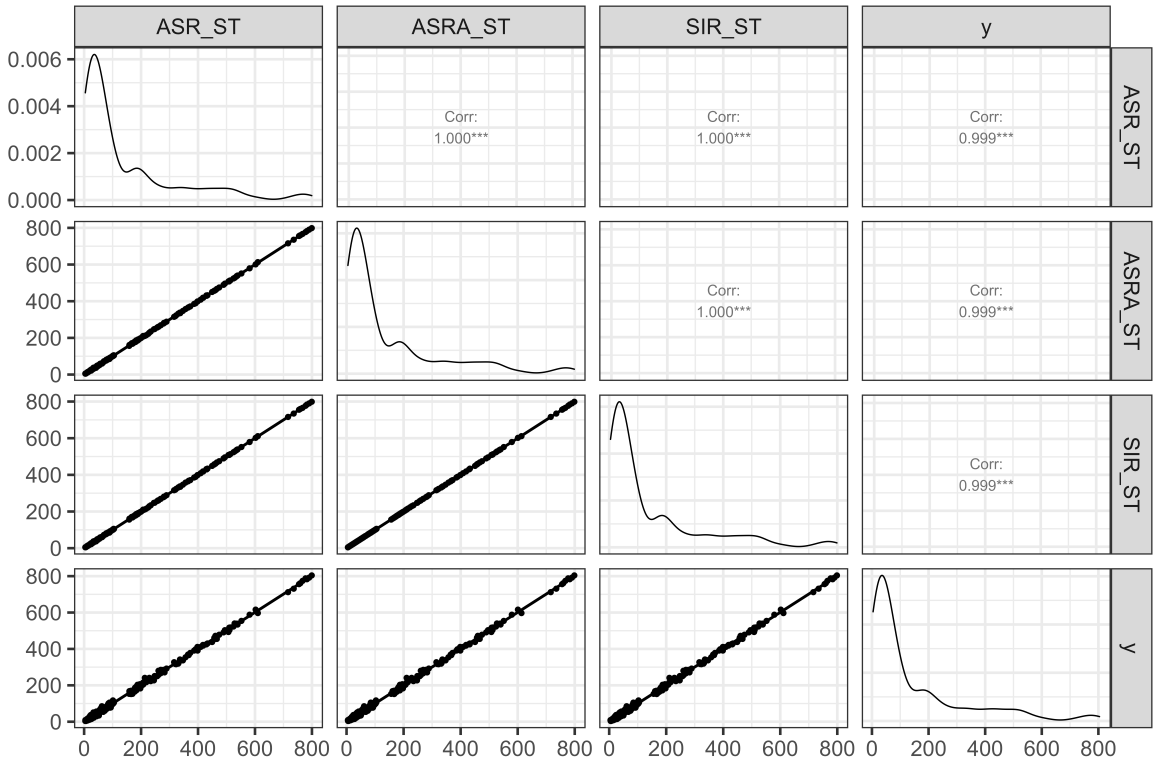



Figure 12: Plots comparing the fitted counts (posterior medians) from the three recommended models for administrative data. The $x$ and $y$ axis represent fitted counts. Observe the almost perfect agreement between the raw counts (denoted as y) and the modelled/fitted posterior median counts: all the points sit on the line of equality and all the pairwise correlations are above 0.99.

## 4.4 Counts

There is also interest in spatially smoothed counts by area, year and age (AYA). From our empirical investigations, the SIR_ST, ASR_ST and ASRA_ST models should produce relatively similar fitted counts by area and year (see Figure 12). However, in general, we would recommend using either the SIR_ST or, if wanting to separate by age, the ASRA_ST model to derive smoothed counts as these seem to agree better in practice.

# 5 Survey data

Data from the Health and Wellbeing Surveillance System (HWSS) survey will be used to produce prevalence estimates for a variety of health factors ranging from smoking to diabetes to self-reported health, among many others (Health Survey Unit, Epidemiology Branch 2011). In the survey data, these health factors are recorded as binary variables, and thus the overall area-by-year estimates will be proportions. Although we focus on methods for proportions in this report, please review the box on page 71 for details on non-binary variables such as BMI.

## 5.1 Small area estimation

Unlike administrative data, which is generally complete, the HWSS survey data are a small cross-section of the WA population each year. Modelling in these circumstances is far more onerous than that for administrative data. Fortunately, methods of small area estimation (SAE) have become commonplace for these kinds of applications (Rao and Molina 2015). SAE targets the small sample size problem by leveraging auxiliary data (often complete census data) along with the observed survey data to generate robust estimates for small areas. In many applications of SAE, researchers use Bayesian hierarchical models along with survey weights (Chen et al. 2014; Gomez-Rubio et al. 2008; You and Rao 2000).

There are two approaches possible: individual-level and area-level models, and these differ in the choice of models and structure of the input data (Table 8).

| Model | Sample weights | Population counts | Input data | Covariates | Model output | Software | Code | Eq. |
|---|---|---|---|---|---|---|---|---|
| **MrP_ST**$^{\pm}$ | | ✓ | Individual-level survey data | Individual- and area-level covariates | Fitted probabilities$^{\S}$ | `nimble` `mcmcsae` | 7 | (5.1) |
| **WMrP_ST**$^{\pm}$ | ✓ | ✓ | Individual-level survey data | Individual- and area-level covariates | Fitted probabilities | `nimble` | 7 | (5.1) |
| **FHELN_ST**$^{\ddagger}$ | ✓ | ✓ | Area-by-year proportion estimates and sampling variances | Area-level census covariates | Fitted probabilities | `mcmcsae` | 10 | (5.8) |

Table 8: Summary of models for survey data. $^{\pm}$ Individual level models, $^{\ddagger}$ Area level models. $^{\S}$Prevalence estimates are derived from the fitted probabilities via a variety of calculations and aggregations (see Section 5.2.2). Approximate run time for these models is of the order of days to weeks (MrP_ST, WMrP_ST) or hours (FHELN_ST).

## 5.2 Individual-level modelling: Multilevel regression and poststratification (MrP)

For individual-level survey data, we recommend using a very common Bayesian method called multilevel regression and poststratification (MrP) (Park et al. 2004). Although extremely common in election modelling (Ghitza and Gelman 2013), MrP has been applied in a variety of health fields (Barker et al. 2013; Berkowitz et al. 2016), and has recently been extended to ST applications (Gao et al. 2021).

MrP requires individual-level survey data (Table 9) and poststrata data (Table 10). The poststrata dataset should have a row for all unique combinations of the factors included in the model (e.g. area, time, age, sex). Each unique combination is called a cell. For each cell in the poststrata data one also requires the corresponding census populations. Note that MrP models should include all persons in the surveys, rather than stratify by sex before modelling.

MrP has several benefits over many SAE methods, which include: modelling at the individual-level as opposed to the area-level; simple prediction for any cell combination of interest; automatic weighting to accommodate nonresponse and sample bias; access to probabilities for cells not observed in the data (Ghitza and Gelman 2013); and relatively simple implementation. MrP is best described in two steps:

- ❖ Model fitting: Fit an ST logistic model to the individual-level survey data
- ❖ Poststratification: Derive fitted probabilities from the logistic model for each cell of the poststrata data and multiply these by the known census population counts.

By summing across all cells apart from area and year, MrP can provide estimates for the proportion or number of persons with the health factor in each area and year. The uncertainty of the prevalence estimates from MrP can be estimated by applying the post-stratification step to all posterior draws of the fitted probabilities.

MrP is best suited to adjust for sampling bias when survey weights are unavailable. If all variables used to derive the survey weights are known, available and included in the MrP model, then the sample design becomes ignorable and the inclusion of survey weights unnecessary. This standard MrP model will be denoted as MrP_ST hereafter and can be fitted using the R packages `mcmcsae` (Boonstra and Baltissen 2021) and `nimble` (de Valpine et al. 2017).

***Consider!* Survey weights**

In practice the MrP_ST model should be avoided as it does not accommodate the survey design (i.e. does not use the individual-level survey weights). Further, we recommend using the WMrP_ST model (see below), as only a single line of `nimble` code is needed to convert the WMrP_ST to the MrP_ST model.

The previous procedure for deriving weights for the HWSS survey data was relatively simple, involving stratification by only several variables that could be easily included in the MrP_ST model. However, the weighting procedure for the HWSS survey is amidst a significant change, that will result in more precise survey weights derived from many more census covariates. It will not be possible to adjust for all these variables in the MrP model and thus a weighted version of MrP_ST is recommended moving forward.

| smokes $y_{jit}$ | sex | age | M_id $i$ | T_id $t$ | MT_id | RA_Name | IRSD_5 | TERTIARYQ_P |
|---|---|---|---|---|---|---|---|---|
| 0 | Female | 1 | 1 | 1 | 1 | Outer Regional | 3 | 57.66 |
| 1 | Female | 1 | 1 | 1 | 1 | Outer Regional | 3 | 57.66 |
| 0 | Male | 1 | 2 | 1 | 2 | Major Cities | 3 | 58.30 |
| 0 | Male | 1 | 2 | 1 | 2 | Major Cities | 3 | 58.30 |
| 0 | Female | 1 | 3 | 1 | 3 | Very Remote | 5 | 72.29 |
| 0 | Female | 1 | 3 | 1 | 3 | Very Remote | 5 | 72.29 |
| 0 | Female | 1 | 4 | 1 | 4 | Inner Regional | 4 | 65.72 |
| 0 | Female | 1 | 4 | 1 | 4 | Inner Regional | 4 | 65.72 |
| 0 | Female | 1 | 5 | 1 | 5 | Major Cities | 4 | 60.95 |
| 0 | Male | 1 | 5 | 1 | 5 | Major Cities | 4 | 60.95 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Table 9: Example of individual-level survey data for input to the MrP_ST and WMrP_ST models, where $y_{jit}$ denotes the binary smoking variable for sampled individual $j$ in area $i$ and time $t$ (see (5.1) for more details of the notation). These data have $n$ observations. To illustrate how the area-level covariates are constant for individuals in the same area and time point, we have included area-level remoteness (`RA_Name`), SEIFA (`IRSD_5`) and the proportion of persons with a tertiary education (`TERTIARYQ_P`). As before, `M_id` and `T_id` are sequential identifiers for the areas and years, respectively.

### 5.2.1 Model: WMrP_ST — model fitting

Until recently it has not been clear how to incorporate survey weights into the Bayesian MrP framework (Kolczynska et al. 2022). Fortunately, one can leverage Bayesian pseudo-likelihood which adjusts the parameter estimates according to the sample design (Savitsky and Toth 2016).

The weighted MrP model (denoted as WMrP_ST) is estimated using the following `nimble` code.

```
code <- nimbleCode({
    # vectorised pseudo-likelihood for Bernoulli
    y[1:n] ~ dwbern_v(# sample scaled weights
                      w = w[1:n],
                      # vector of probabilities
                      p = p[1:n])

    # iterate across all the rows of the data (n)
    for(i in 1:n){
        # linear predictor
        logit(p[i]) <- alpha +
        # Fixed effects using the inner product
        + inprod(B_qr[1:q], Q_ast[i,])
        # BYM2 spatial term
        + theta[M_id[i]]
        # ICAR temporal term
```

```
        + gamma[T_id[i]]
    }

    # Spatial: BYM2 #
    # iterate across all areas (M)
    for(i in 1:M){
        theta[i] <- sigma_theta * (s[i] + v[i])
        # structured component
        s[i] <- Z_s[i] * sqrt(rho/kappa)
        # unstructured component
        v[i] <- Z_v[i] * sqrt((1 - rho))
        # standard normal
        Z_v[i] ~ dnorm(0, 1)
    }

    # ICAR prior #
        Z_s[1:M] ~ dcar_normal(adj[1:L_s],
                               weights[1:L_s],
                               num[1:M],
                               1, # unit-variance ICAR
                               # enforce sum-to-zero
                               zero_mean = 1)

    # RW1 prior #
        Z_gamma[1:T] ~ dcar_normal(T_adj[1:L_t],
                                   T_weights[1:L_t],
                                   T_num[1:T],
                                   1, # unit-variance ICAR
                                   # enforce sum-to-zero
                                   zero_mean = 1)
        # multiply by sigma - scalar by vector multiplication
        gamma[1:T] <- sigma_gamma * Z_gamma[1:T]

    # Other priors #
        # iterate over all elements of B_qr
        for(i in 1:q){
            B_qr[i] ~ dnorm(0, sd = 1000)
        }
        alpha ~ dnorm(0, sd = 1000)
        sigma_theta ~ dgamma(2, 0.5)
        sigma_gamma ~ dgamma(2, 0.5)
```

```
58          rho ~ dunif(0,1)
59
60      # recreate true coefficient values
61          B[1:q] <- R_ast_inverse[1:q,1:q] %*% B_qr[1:q]
62
63  })
```

R Code 7: Example BUGS syntax to fit the WMrP_ST model.

The model shown in Code 7 is written as follows. Note that `n` is the total number of sampled persons across all areas and time points. Let $y_{jit}$ be a binary variable (taking values of 0 or 1 to denote presence of the condition) for sampled individual $j$ in area $i$ and time $t$.

$$\begin{aligned} y_{jit} &\sim \text{Bernoulli}\left(p_{jit}\right)^{w_{jit}} \\ \text{logit}\left(p_{jit}\right) &= \alpha + \mathbf{X}_{jit}\boldsymbol{\beta} + \theta_i + \gamma_t \\ \boldsymbol{\theta} &\sim \text{BYM2}\left(\mathbf{W}^{\text{S}}, \rho, \kappa, \sigma_\theta^2\right) \\ \boldsymbol{\gamma} &\sim \text{ICAR}(\mathbf{W}^{\text{T}}, \sigma_\gamma^2) \\ \rho &\sim \text{Uniform}(0,1) \\ \sigma_\theta, \sigma_\gamma &\sim \text{Gamma}(2, 0.5) \\ \alpha, \boldsymbol{\beta} &\sim N(0, 1000^2) \end{aligned} \tag{5.1}$$

Note that we do not include a space-time interaction in this model as we cannot assume that all areas across all time points have been sampled.

The notation of Bernoulli $\left(p_{jit}\right)^{w_{jit}}$ given in (5.1), denotes the pseudo-likelihood component of this model, where $w_{jit} = w^r_{jit}\frac{n}{\sum w^r_{jit}}$ are the sample scaled weights and $w^r_{jit}$ are the raw weights available from the HWSS data. Pseudo-likelihood involves weighting each likelihood contribution by its corresponding $w_{jit}$. Following the notation of (2.1) on page 9, the pseudo log likelihood is of the form

$$\sum_i w_i \log\, p\left(y_i|\mathbb{P}\right). \tag{5.2}$$

Pseudo-likelihood approaches can be crudely implemented in `nimble`, however we found it more efficient to use a vectorised version of the pseudo-likelihood function. This user-created function, `dwbern_v()`, is implemented on line 3 of Code 7.

The motivation for using pseudo-likelihood is to ensure that the posterior distribution is similar to that from the same specified MrP model fit to the entire population. The sample

scaling of the survey weights ensures that the posterior distribution has the correct uncertainty given the sample size and sample design. Note that by fixing all $w_{jit} = 1$, we can easily fit the MrP_ST model using Code 7. See the box on page 49.

The fixed effect design matrix, $\mathbf{X}_{jit}$ in (5.1), will generally include a variety of unit-level and area-level covariates. Code 8 shows how to construct this matrix. In the example we use the interaction of age group and sex, along with many area level covariates including area-level remoteness and socioeconomic status.

```
X <- model.matrix(~agegroup2*sex + RA_Name + IRSD_5
                  + Tot_P_FP + sqrt(Tot_Indigenous_PP)
                  + lowincome_Totp + TERTIARYQ_P
                  + OLOMW_P + Age_yr_35_39_MP
                  + Age_yr_15_19_FP,
                  # what dataset
                  data = df)
```

R Code 8: Constructing the design matrix for the WMrP_ST model. In practice, we also take the QR decomposition of `X` (Section 8.8).

### 5.2.2 Model: WMrP_ST — poststratification

After we fit the WMrP_ST model we must derive the probabilities for all combinations of the individual-level covariates for each area and year. For the example in Code 8, we would require probabilities for all combinations of sex, age, area and year, even if these combinations did not appear in the survey data explicitly. To generalise our notation, we'll let $p_{fit}$ denote the probability for the $f$th ($f = 1, \ldots, F$) combination of age and sex in area $i$ and time $t$. Thus, the poststrata dataset should have $F \times M \times T$ rows (see Table 10).

***Consider!* Population counts**

MrP models require population counts, $N_{fit}$, for all combinations of $f, i$ and $t$. This can become restrictive in situations where more individual-level covariates must or should be included. For example, if we believe that smoking should be modelled by the individual covariates of income, occupation, education, sex, and age, then $F$ becomes all these combinations. Without access to microdata, publicly available census counts for all $F$ in all areas and years will not be accessible. In most applications, $F$ is conveniently chosen to align with the data that is available, however these practical decisions do not always correspond with the best MrP model.

Once we derive the unique probabilities, $\mathbf{p} \in \mathbb{R}^{(F\times M\times T)\times 1}$, we multiply them by their corresponding populations to get estimated counts. Finally, to derive the area by year

| Probability | sex | age | M_id | T_id | MT_id | RA_Name | IRSD_5 | TERTIARYQ_P |
|---|---|---|---|---|---|---|---|---|
| $p_{111}$ | Male | 1 | 1 | 1 | 1 | Outer Regional | 3 | 57.66 |
| $p_{211}$ | Female | 1 | 1 | 1 | 1 | Outer Regional | 3 | 57.66 |
| $p_{121}$ | Male | 1 | 2 | 1 | 2 | Major Cities | 3 | 58.30 |
| $p_{221}$ | Female | 1 | 2 | 1 | 2 | Major Cities | 3 | 58.30 |
| $p_{131}$ | Male | 1 | 3 | 1 | 3 | Very Remote | 5 | 72.29 |
| $p_{231}$ | Female | 1 | 3 | 1 | 3 | Very Remote | 5 | 72.29 |
| $p_{141}$ | Male | 1 | 4 | 1 | 4 | Inner Regional | 4 | 65.72 |
| $p_{241}$ | Female | 1 | 4 | 1 | 4 | Inner Regional | 4 | 65.72 |
| $p_{151}$ | Male | 1 | 5 | 1 | 5 | Major Cities | 4 | 60.95 |
| $p_{251}$ | Female | 1 | 5 | 1 | 5 | Major Cities | 4 | 60.95 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Table 10: Example structure of the poststrata dataset for the MrP_ST and WMrP_ST models. Note that the poststrata dataset can become very large as $F$ increases. For this example, we have 3 age groups and 2 sex groups, and thus $F = 6$. This means that the poststrata dataset will have 7344 rows ($F = 6, M = 136, T = 9$).

prevalence estimates, $\hat{p}_{it}$, we sum across all the $F$ categories and divide by the area and year populations. Of course, these calculations are complete for each posterior draw of the model parameters from `nimble`.

To simplify the computation we use a series of matrix multiplications within `for` loops, which are shown in Code 9 and described below.

$$\begin{aligned} \mathbf{p}^{(d)} &= \text{logit}^{-1}\left(\alpha^{(d)} + \mathbf{Q}\boldsymbol{\beta}^{(d),\text{qr}} + \mathbf{G}\boldsymbol{\lambda}^{(d)}\right) \\ \hat{p}_{it}^{(d)} &= \frac{\sum_f \left(N_{fit} p_{fit}^{(d)}\right)}{\sum_f N_{fit}} \end{aligned} \tag{5.3}$$

Let $\boldsymbol{\lambda}^{(d)} = \left(\boldsymbol{\theta}^{(d)}, \boldsymbol{\gamma}^{(d)}\right) \in \mathbb{R}^{(M+T)\times 1}$ and $\mathbf{G} \in \mathbb{R}^{N\times(M+T)}$. The matrix $\mathbf{G}$ is a sparse matrix that specifies the area and year for each row of the poststrata dataset. Please see Section 8.1 for details on the $\mathbf{G}$ and $\boldsymbol{\lambda}$ matrices and Section 8.8 for details on the $\mathbf{Q}$ and $\boldsymbol{\beta}^{\text{qr}}$ matrices.

Although we've discussed prevalence estimates by area and year only, the vector $\mathbf{p}^{(d)} = \left(p_{111}^{(d)}, \ldots, p_{FMT}^{(d)}\right)$ in (5.3) gives prevalence estimates by age, sex, area and year. Thus, we can collapse/aggregate across any of these variables to derive prevalence estimates.

```
# Extract posterior draws for ALL parameters
    # get samples from wrapper function
    fit_draws <- as.matrix(WMrPST_fit$fit$samples)

# get draws for specific parameters - iterations by MT
    # use user-made function to select the correct matrix of draws
    alpha <- jf$getSubsetDrawsNimble(fit_draws, "alpha")
    beta <- jf$getSubsetDrawsNimble(fit_draws, "B_qr\\[")
    # both spatial and temporal random effects - iterations by MT
```

```
    lambda <- cbind(jf$getSubsetDrawsNimble(fit_draws, "theta\\["),
                    jf$getSubsetDrawsNimble(fit_draws, "gamma\\["))

# construct the sparse G matrix for poststrata
    G <- Matrix(cbind(
      # ensure no intercept
      # spatial, temporal
      model.matrix(~as.factor(M_id) - 1, data = poststrata),
      model.matrix(~as.factor(T_id) - 1, data = poststrata)
      # sparse makes computations much faster
    ), sparse = T)

# Posterior draws for counts - all rows of poststrata matrix
    # empty matrix of iterations by `nrow(poststrata)`
    counts <- matrix(NA,
                     nrow = length(alpha),
                     ncol = nrow(poststrata))
    # add a progress bar
    pb <- txtProgressBar(min = 0, max = length(alpha), style = 3)
    for(i in 1:length(alpha)){
    # returns a vector of probabilities of length `nrow(poststrata)`
    linear_predictor <- as.numeric(jf$jinvlogit(alpha[i]
                                                # fixed effects
                                                + QR_ps$QR$Q_ast %*% beta[i,]
                                                # random effects
                                                + G %*% lambda[i,]))
    # population by probabilities gives counts
    counts[i,] <- poststrata$N * linear_predictor
    setTxtProgressBar(pb, i)
    }
    close(pb) # remove the progress bar

# get posterior draws by area and year
    # we need the population by area and year
    temp <- poststrata %>%
        group_by(MT_id) %>%
        summarise(N = sum(N))
    # create temporary function to
    # apply to each row of `counts`
        # sum counts across sex and age and then
        # divide by corresponding population
```

```
51        foo <- function(x){aggregate(x,
52        list(poststrata$MT_id), sum)[,2]/temp$N}
53
54    # apply foo to each row of `counts`
55    mrp_prev_draws <- t(apply(counts, 1,
56                              FUN = foo))
57    # ordered according to MT_id
58    # iterations times MT
59
60    # keep environment clean
61    rm(temp, foo) # remove `temp` and `foo`
```

R Code 9: Example code for the poststratification step of the MrP_ST or WMrP_ST models.

The R object `mrp_prev_draws` in Code 9 gives a matrix of posterior draws (as rows) and areas and years (as columns) which can then be summarised as needed.

***Tech Talk!* Model checking for logistic regression**

Model checking for logistic regression requires slightly different techniques to those recommended in Section 2.3. Useful options include:

- Posterior predictive checks (particularly the mean of the data)
- Examine the predictive performance of the model (sensitivity, specificity, etc)
- Examine the receiver operating characteristic (ROC) curve and the area under this curve (AUC) (see Figure 13).

Note that most of these options are related to model selection (Section 5.4).

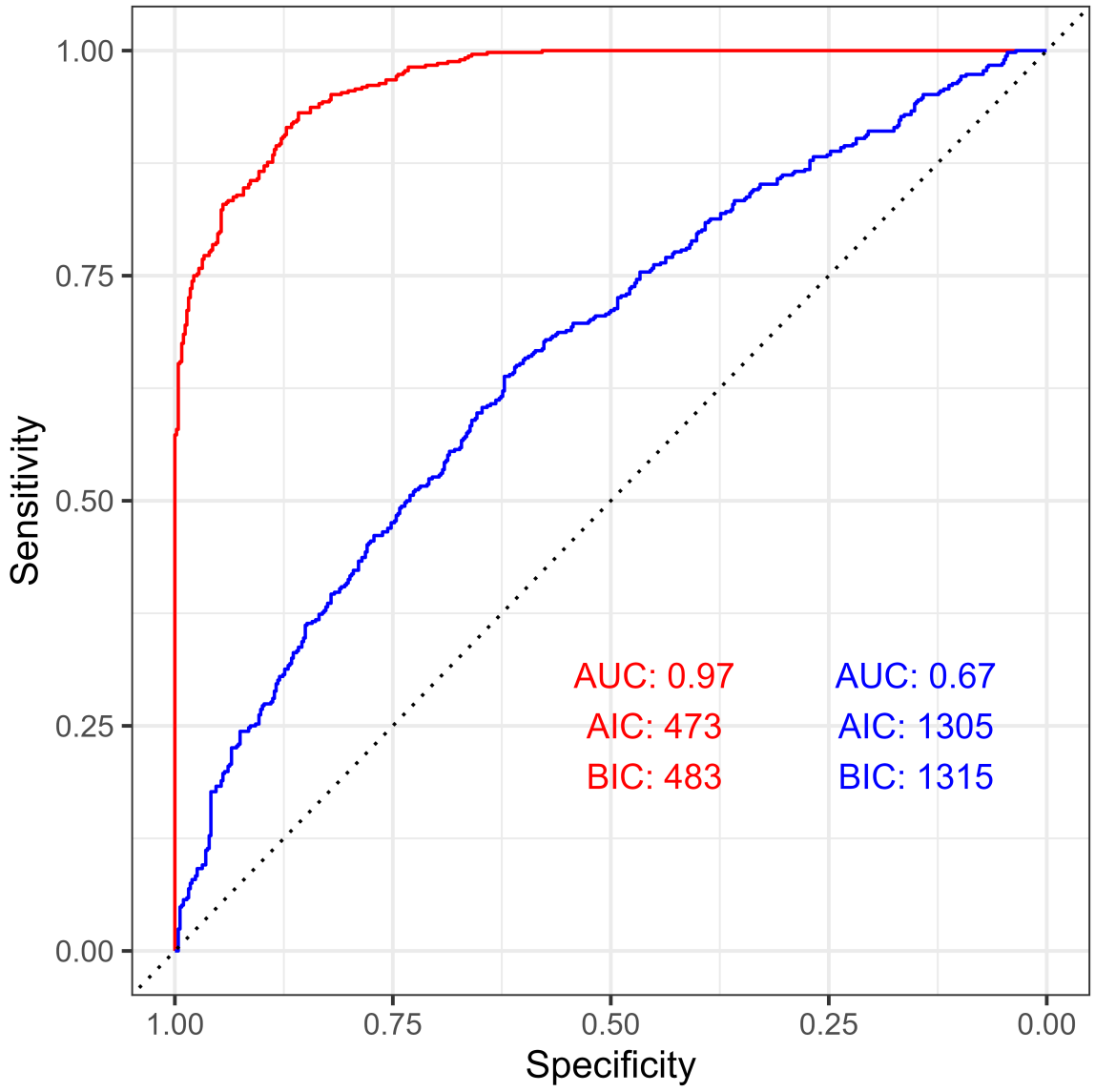



Figure 13: Comparison of the receiver operating characteristics (ROC) curves for two logistic models. The ROC curves compare the specificity and sensitivity of the predictions from the two models. On the bottom right of the plot we provide the area under the ROC (AUC), AIC and BIC (see Section 5.4). Higher values of AUC are preferred and thus, we would select the model represented by the red ROC.

## 5.3 Area-level modeling: Fay-Herriot

In some cases, MrP is not feasible, especially with temporal data, where the input datasets can become increasingly large. Standard MCMC algorithms are generally not optimised to be scalable to large datasets. In the case of the temporal HWSS survey data, where there are over 50,000 data points across LGAs and years, run time can exceed 5 days.

Area-level SAE models are a useful alternative to individual-level SAE models when computational feasibility is of great concern and/or where access to the individual-level survey data is unobtainable. The very common Fay-Herriot (FH) area-level model (Fay and Herriot 1979) only requires survey data summaries by area and year; a significantly smaller dataset than that required for individual-level models. These area by year summaries correspond to both a weighted proportion estimate, $\hat{p}_{it}$, and sampling variance, $\hat{\psi}_{it}$, for area $i$ and year $t$ which are estimated from the individual-level survey data. These area by year summaries are often called *direct* estimates as they rely only on the sampled individuals within that area and year. See the box on page 61 for details on how to derive the direct estimates.

By using the survey weights in the proportion estimates and sampling variances, the FH model ensures that the resulting model estimates are unbiased under the sample design. Given that the FH model is a special case of standard Bayesian hierarchical models, spatial and temporal terms can be easily included (Gomez-Rubio et al. 2008). Although initially proposed for continuous outcomes, the FH model can also be used for proportions by applying an empirical logistic transformation prior to modelling (Mercer et al. 2014) (see Section 5.3.1). We'll denote the area-level ST FH empirical logistic model as the FHELN_ST model (see Table 8). The FHELN_ST model can be fitted in the R package `mcmcsae`, which provides an efficient means of estimating complex SAE models using Bayesian inference (Boonstra and Baltissen 2021). R Code for fitting the FHELN_ST model are provided in Code 10, with detailed explanation in the following pages.

```
# use GVF to impute unstable sampling variances
    # fit the GVF using OLS
    gvf <- lm(log(phat_u_SE) ~ log(n) + log(N) + jf$jlogit(HT), data = df)
    # generate phat_u_SE values for all areas and years
    imputed <- exp(predict(gvf, newdata = df))
    # phat_u_SE_smoothed will be equal to the GVF estimate if
    # `unstable` is true
    df$phat_u_SE_smoothed <- with(df, ifelse(unstable, imputed, phat_u_SE))
    # sort by area and time
    df <- arrange(df, MT_id)

# Set linear predictor details
    # the reg(.) gives the fixed effects
    lp <- phat_u ~  reg(~ 1 + RA_Name + IRSD_5 + Tot_P_FP +
                        sqrt(Tot_Indigenous_PP) +
```

```
                    lowincome_Totp + TERTIARYQ_P + OLOMW_P +
                    Age_yr_35_39_MP +
                    Age_yr_15_19_FP, name = "beta") +
  # Random walk for the temporal term
      gen(factor = ~ RW1(T_id), prior = pr_invchisq(df = 1))+
  # BYM for spatial terms
      # ICAR component - spatially structured
      gen(factor=~spatial(M_id, poly.df = map_sp,
                          snap = T, queen = T),
                          prior = pr_invchisq(df = 1))+
      # unstructured component
      gen(factor=~iid(M_id), prior = pr_invchisq(df = 1))

  # create sampler
  sampler <- create_sampler(lp, # linear predictor
                            sigma.fixed = TRUE,
                            # Q0 is precision (1/SE^2)
                            Q0=1/(df$phat_u_SE_smoothed)^2,
                            linpred = "fitted",
                            data = df)

  # fit the FH ELN model using MCMC
  # usable draws: (n.iter - burnin) * n.chain
  FHELNST_fit <- MCMCsim(sampler,
                         store.all = T,
                         n.chain = 4,
                         n.iter = 8000,
                         burnin = 4000,
                         thin = 2,
                         verbose = T)

  # get Nimble-like summary measures
  summary <- jf$mcmcsaeGetSummaries(FHELNST_fit,
                                    # time (mins) to fit
                                    # the FHELNST model
                                    time = diff)
```

R Code 10: Example code to fit the FHELN_ST model.

### 5.3.1 Fay-Herriot model

The FH model (Fay and Herriot 1979) was originally developed for continuous data, however we are working with proportions which are bounded between $0$ and $1$. Furthermore, MCMC can be far more efficient when utilising a normal likelihood rather than alternatives such as the Beta distribution. Thus, prior to modelling we convert the direct proportion estimates and sampling variances to the unconstrained space (denoted by the superscript u) using an empirical logit transformation (Mercer et al. 2014).

$$\begin{aligned} \hat{p}_{it}^{\mathrm{u}} &= \text{logit}\left(\hat{p}_{it}\right) \\ \hat{\psi}_{it}^{\mathrm{u}} &= \hat{\psi}_{it}\left[\hat{p}_{it}\left(1-\hat{p}_{it}\right)\right]^{-2} \end{aligned} \tag{5.4}$$

It is the above quantities that are used as input into the FHELN_ST model. See example data in Table 11.

| phat | phat_u | phat_VAR | phat_u_VAR | n | N | M_id | T_id | TERTIARYQ_P | IRSD_5 | Tot_P_FP | lowincome_Totp |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\hat{p}_{it}$ | $\hat{p}_{it}^{\mathrm{u}}$ | $\hat{\psi}_{it}$ | $\hat{\psi}_{it}^{\mathrm{u}}$ | $n_{it}$ | $N_{it}$ | $i$ | $t$ | | | | |
| 0.16 | -1.68 | 0.00097 | 0.06 | 298 | 27434 | 1 | 1 | 57.66 | 3 | 50.65 | 7.34 |
| 0.11 | -2.13 | 0.0013 | 0.14 | 87 | 50788 | 2 | 1 | 58.30 | 3 | 50.25 | 6.01 |
| 0.17 | -1.60 | 0.0042 | 0.22 | 35 | 8334 | 3 | 1 | 72.29 | 5 | 27.64 | 2.88 |
| 0.05 | -3.02 | 0.0011 | 0.55 | 73 | 9526 | 4 | 1 | 65.72 | 4 | 50.03 | 6.87 |
| 0.001 | -6.91 | | | 4 | 1832 | 9 | 1 | 55.10 | 3 | 47.07 | 8.21 |
| 0.001 | -6.91 | | | 10 | 824 | 14 | 1 | 52.81 | 2 | 49.30 | 4.14 |
| 0.001 | -6.91 | | | 3 | 428 | 21 | 1 | 48.68 | 3 | 47.45 | 7.56 |
| 0.001 | -6.91 | | | 8 | 929 | 23 | 1 | 57.71 | 5 | 49.79 | 6.94 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | |

Table 11: Example dataset for the FHELN_ST model. In the table, $\hat{p}_{it}$ and $\hat{p}_{it}^{\mathrm{u}}$ represents the direct proportion estimate and unconstrained estimate for area $i$ and time point $t$, respectively. Furthermore, $\hat{\psi}_{it}$ and $\hat{\psi}_{it}^{\mathrm{u}}$ represent the direct proportion sampling variance and unconstrained sampling variance for area $i$ and time point $t$, respectively. Finally, $n_{it}$ and $N_{it}$ give the sample size and population size for area $i$ and time $t$, respectively. See the following pages for notation details. Note that for unstable direct estimates (the final 4 rows above), there are no variance estimates.

***Tech Talk!* Formula for direct estimates and sampling variance**

In our exploratory work, we used the following formula to derive the direct proportion estimates and sampling variances. Let $n_{it}$ and $N_{it}$ be the sample size and population size in area $i$ at time $t$. We also rescale the survey weights to sum to the area-by-year sample sizes (e.g. $n_{it} = \sum_j w_{jit}$).

$$\hat{p}_{it} = \frac{\sum_j w_{jit} y_{jit}}{\sum_j w_{jit}} \tag{5.5}$$

$$\hat{\psi}_{it} = \frac{1}{n_{it}} \left(1 - \frac{n_{it}}{N_{it}}\right) \left(\frac{1}{n_{it} - 1}\right) \sum_j \left(w_{jit}^2 \left(y_{jit} - \hat{p}_{it}\right)^2\right) \tag{5.6}$$

Weighted means such as these are referred to as Hajek or Horvitz–Thompson estimators (Rao and Molina 2015).

Use of area-level models is not conditional on these specific formulae. These quantities have already been derived for the DOHWA Public Health Atlas project and thus there is little need to adopt these formulae moving forward. The critical element is keeping the formula consistent throughout analysis.

### 5.3.2 Generalised variance functions

Although area-level models offer significant computational advantages over their individual-level competitors, in the proportion setting area-level models can suffer from instability. Instability occurs when a weighted proportion estimate is exactly zero or 1, which makes the sampling variance infinite or zero.

Consider an area $i$ and time $t$ with $n_{it} = 3$ and binary observations $\mathbf{y}_{it} = (0, 0, 0)$. In this example, regardless of the survey weights, $\hat{p}_{it}, \hat{\psi}_{it} = 0$ — the estimate is unstable. Applying the empirical logistic transformation in (5.4) to this unstable estimate results in $\hat{p}_{it}, \hat{\psi}_{it}$ being zero or undefined. Unstable area-by-year direct estimates such as these cannot be included in the FHELN_ST model without necessary adjustment.

Although ad-hoc, we recommend perturbing any unstable direct proportion estimates prior to modelling. For example, the $\hat{p}_{it}$ in the 5th row of Table 11 was originally 0 but has been perturbed to 0.001. A similar perturbation would be performed for direct proportion estimates of exactly 1 (e.g. setting them to 0.999). The user-supplied function `jf$jDirect()` automatically applies these perturbations prior to performing the transformations in (5.4). The function also allows the user to define the size of the perturbation via the `eps` argument. Note that some consideration should be given to the size of the perturbation because extremely small values (e.g. `eps = 0.00001`) can create extreme outliers on the logit scale.

Unlike the simple solution we recommend for unstable direct estimates, a more complex solution is required to correct the sampling variances. Generalised variance functions (GVF) (Wolter 2007) can be used to approximate the undefined (or zero-valued) sampling variances

by modelling the relationship between the stable sampling variances and other area-by-year covariates. Common choices are the stable direct estimates, sample sizes and population sizes (Das et al. 2021). The GVF we recommend takes the following form,

$$\begin{aligned} \log \sqrt{\hat{\psi}^{\mathrm{u}}_{it}} &\sim N\left(\mu_{it}, \sigma^2_\epsilon\right) \\ \mu_{it} &= \alpha + \beta_1 \log n_{it} + \beta_2 \log N_{it} + \beta_3 \hat{p}^{\mathrm{u}}_{it} \end{aligned} \tag{5.7}$$

where the model parameters are estimated using frequentist ordinary least squares (see Code 10). By leveraging the GVF fitted to the stable sampling variances, we replace the undefined (or unstable) sampling variances using

$$\hat{\psi}^{\mathrm{u}}_{it} = \left(e^{\mu_{it}}\right)^2.$$

In practice, one should investigate the validity of the GVF and choose appropriate covariates to maximise its predictive capability (via $R^2$). An example of a well specified GVF is shown in Figure 14 on page 63.

***Consider!* Avoiding GVFs in the FHELN_ST model**

GVFs are used extensively in SAE to smooth *all* sampling variances (i.e. replace even the stable values) (Das et al. 2021). We discourage this practice and recommend only replacing undefined sampling variances using GVFs.

An alternative to GVFs is to remove all unstable estimates during modelling, in the hope to impute these after using the fitted model. When instability is extremely low, then this approach is feasible. However, for extremely sparse survey data or extremely common or rare conditions, where many area-by-year estimates are unstable, removing unstable estimates may result in dropping a large portion of the data: a practice that should be avoided.

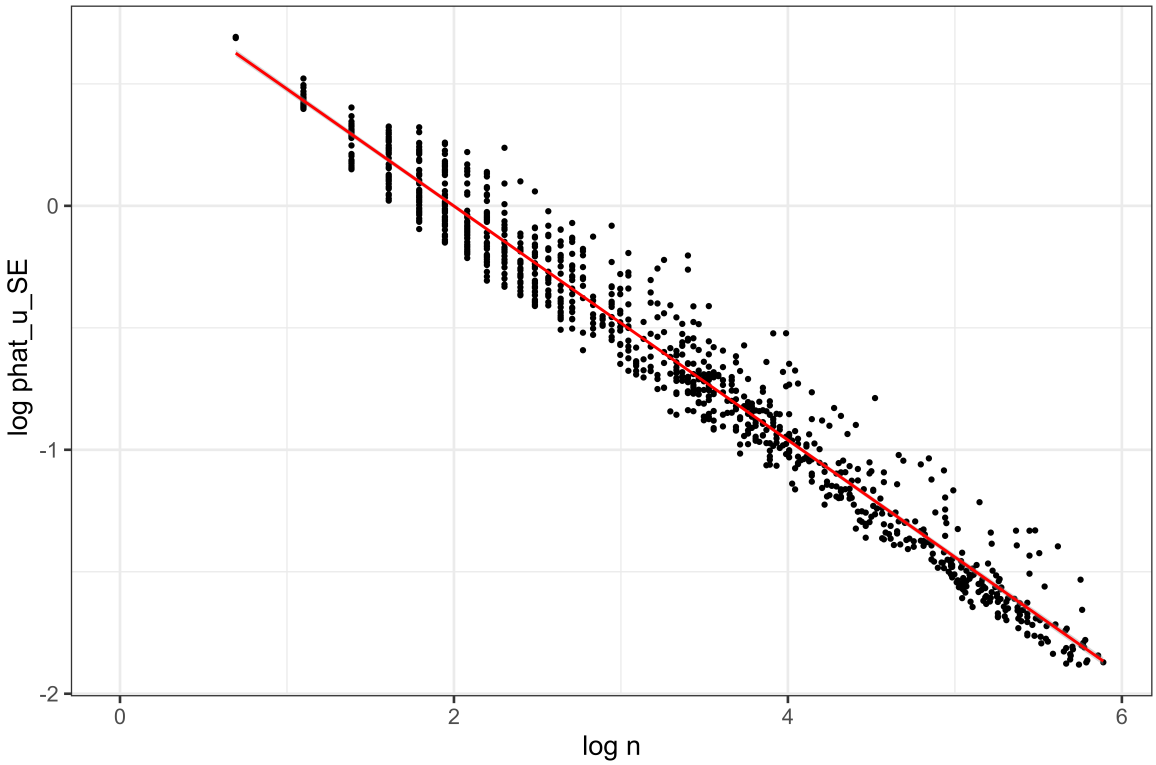



Figure 14: Example of a GVF which can model the relationship between the sampling variance and sample size. In this plot, $\log\sqrt{\hat{\psi}_{it}^{\mathrm{u}}}$ is on the $y$-axis, while the corresponding sample size, $\log n_{it}$, on the $x$-axis. The red line denotes the OLS fit which gives an $R^2 \approx 0.85$.

### 5.3.3 Model: FHELN_ST

By using the GVFs correctly, all $\hat{p}_{it}^{\mathrm{u}}, \hat{\psi}_{it}^{\mathrm{u}}$ are approximately stable. Thus, we can specify the FHELN_ST model as follows, where $p_{it}^{\mathrm{u}}$ is the proportion estimate of interest.

$$
\begin{aligned}
\hat{p}_{it}^{\mathrm{u}} &\sim N\left(p_{it}^{\mathrm{u}}, \hat{\psi}_{it}^{\mathrm{u}}\right) \\
p_{it}^{\mathrm{u}} &= \alpha + \mathbf{X}_{it}\boldsymbol{\beta} + s_i + v_i + \gamma_t \\
\mathbf{s} &\sim \mathrm{ICAR}\left(\mathbf{W}^{\mathrm{S}}, \sigma_s^2\right) \\
v_i &\sim N\left(0, \sigma_v^2\right) \\
\boldsymbol{\gamma} &\sim \mathrm{ICAR}(\mathbf{W}^{\mathrm{T}}, \sigma_\gamma^2) \\
\sigma_s^2, \sigma_v^2, \sigma_v^2 &\sim \mathrm{Inv}\chi^2\left(1\right) \\
\alpha, \boldsymbol{\beta} &\sim 1
\end{aligned}
\tag{5.8}
$$

Note that $\mathrm{Inv}\chi^2\left(1\right)$ is the inverse $\chi^2$ distribution with 1 degree of freedom and $\sim 1$ denotes a flat prior. `mcmcsae` is yet to implement the BYM2 prior and thus, we use the BYM specification for the spatial random effects (Besag et al. 1991) (see Section 3.3.2).

***Tech talk!* Adaptive smoothing with the FH model**

The FH model has adaptive smoothing properties, which makes it ideal in settings where the sample sizes vary considerably across space and time. We would like the model to trust $\hat{p}^{\text{u}}_{it}$ when the sample size is large, while trust $p^{\text{u}}_{it}$ when the sample size is small. That is, we would like our SAE model to automatically decide how much of the direct, $\hat{p}^{\text{u}}_{it}$, and modelled estimate, $p^{\text{u}}_{it}$, to use. By rewriting the FH model as a weighted mean,

$$p^{\text{u}}_{it} = r_{it}\hat{p}^{\text{u}}_{it} + (1 - r_{it})\left(\alpha + \mathbf{X}_{it}\boldsymbol{\beta}\right),$$

where $r_{it} = \frac{\sigma_s^2+\sigma_v^2+\sigma_\gamma^2}{\sigma_s^2+\sigma_v^2+\sigma_\gamma^2+\hat{\psi}^{\text{u}}_{it}}$ and represents the ratio of spatio-temporal variation relative to the total variation. Notice that as the unconstrained sampling variance, $\hat{\psi}^{\text{u}}_{it}$, approaches zero (i.e. the direct estimate becomes very certain), $r_{it} \to 1$, which means $p^{\text{u}}_{it} \approx \hat{p}^{\text{u}}_{it}$. This is exactly the behaviour we would like the model to exhibit.

By assuming that not all combinations of the areas and years will have been sampled, we cannot use the inverse logit transformation of $p^{\text{u}}_{it}$ alone. Similar to the poststratification process used previously (see Code 9), we follow similar steps to derive the posterior draws for the prevalence for all areas and years, $\mathbf{p}^{(d)} \in \mathbb{R}^{MT}$.

$$\mathbf{p}^{(d)} = \text{logit}^{-1}\left(\alpha^{(d)} + \mathbf{X}\boldsymbol{\beta}^{(d)} + \mathbf{G}\boldsymbol{\lambda}^{(d)}\right) \tag{5.9}$$

Let $\boldsymbol{\lambda}^{(d)} = \left(\mathbf{s}^{(d)}, \mathbf{v}^{(d)}, \boldsymbol{\gamma}^{(d)}\right) \in \mathbb{R}^{(2M+T)\times 1}$ and $\mathbf{G} \in \mathbb{R}^{N\times(2M+T)}$. The matrix $\mathbf{G}$ is a sparse matrix that specifies the area and year for each row of the poststrata dataset.

***Consider!* Individual-level or area-level**

Figures 15, 16 and 17 compare the smoothed prevalence estimates from the FHELN_ST and WMrP_ST models for a very sparse condition (stroke) and a common condition (sufficient fruit consumption). The data are derived from the 2011-2019 individual-level HWSS survey data. Both models use the same area-level covariates (see Code 10) while the WMrP_ST model also includes sex and age at the individual level.

The state prevalence of stroke is around 2%, which results in about 50% of area-by-year $\hat{p}_{it}$ values being unstable. This level of instability makes the area-level model a poor choice; the plots support this. Figure 16 compares the prevalence estimates of stroke stratified by the sample size and coloured according to whether the relative standard error (RSE) of $\hat{p}_{it}$ is less than the cutoff of 50% (see Section 8.3 for details of RSEs). For large sample sizes (bottom left) the estimates from the FHELN_ST and WMrP_ST models agree very well. Conversely, for small sample sizes the prevalence estimates correspond very poorly. In particular, note how the FHELN_ST model predicts some values as high as 0.8 while the WMrP_ST model gives a (more appropriate) estimate around 0.02. This confirms that for these severely sparse data, area-level models can provide inadequate and implausible estimates. Furthermore, observe in plot (b) of Figure 17 that almost all the WMrP_ST estimates have RSEs below 50% while a majority of the estimates from the FHELN_ST model are above the cutoff. For sparse conditions such as stroke, it is clear that the WMrP_ST is far superior.

As expected, the conclusions above do not hold for fruit consumption (Figure 15). The state prevalence of fruit consumption is around 50% where only 10% of the area-by-year $\hat{p}_{it}$ values are unstable. The small level of instability makes the FHELN_ST a more efficient choice given that inference is very similar to that from the far more complicated WMrP_ST model (see the RSEs in Figure 17). Note that the FHELN_ST took only 1 minute to fit, while the WMrP_ST took over 4.5 days. Given the huge time gains in using the FHELN_ST model, we recommend this model for common conditions.

Although we are hesitant to give strict cutoffs that class a condition as common or sparse, the WMrP_ST model should perform well in all scenarios, so if one is unsure then the WMrP_ST model should be used.

Variables were selected using some of the methods discussed in Section 5.4. However this process was not extensive. Given that results can differ dramatically when different covariates are included, the results here are mostly for illustration purposes. One can always strive to improve the predictive accuracy of SAE models with more (or less) complicated models.

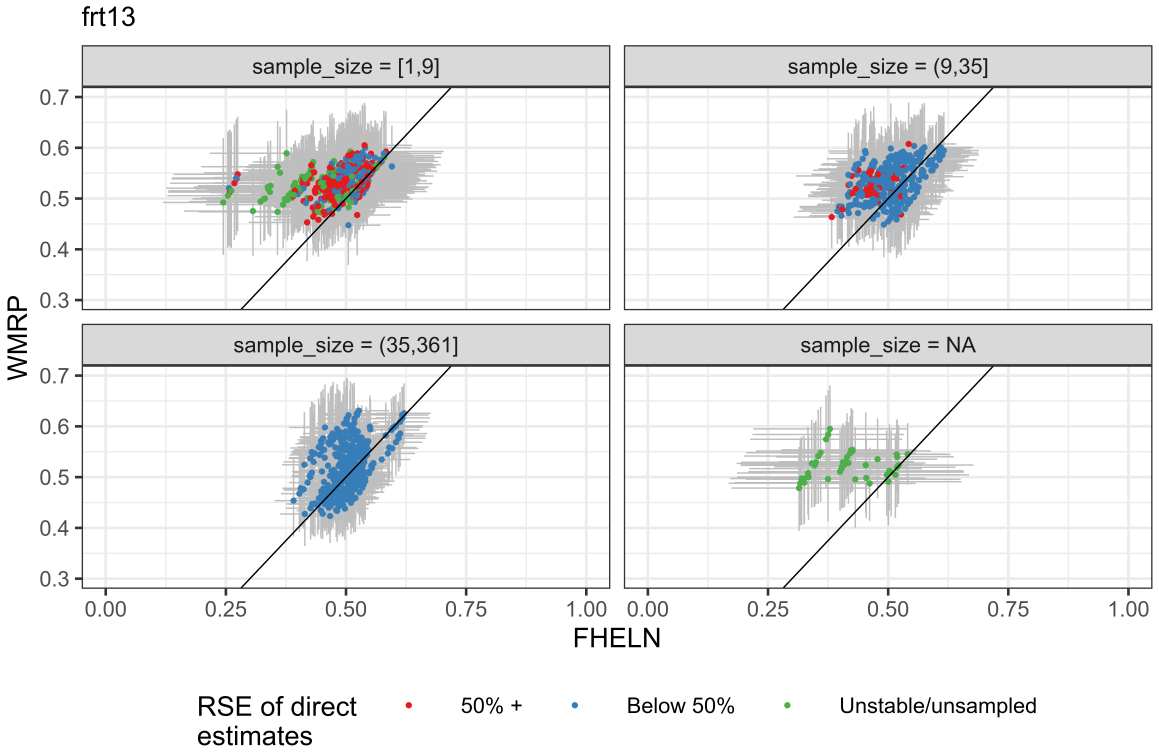



Figure 15: Comparison of modelled prevalence estimates of fruit consumption (`frt13`) at the LGA level from 2010–2019. The plots compare the area-by-year estimates from the WMrP_ST and FHELN_ST models by displaying the posterior medians and credible intervals on both axes (e.g. a single point gives the posterior medians and 95% credible intervals from both the WMrP_ST and FHELN_ST models). The four quadrants indicate different area-by-year samples sizes and the colours describe the RSEs of the direct estimates. The bottom left quadrant compares the prevalence estimates for those area-by-year combinations with no sample data. Note that $[1, 9]$ denotes all sample sizes from 1 to 9 inclusive, whilst $(9, 35]$ denotes sample sizes *above* 9 but less than 35.

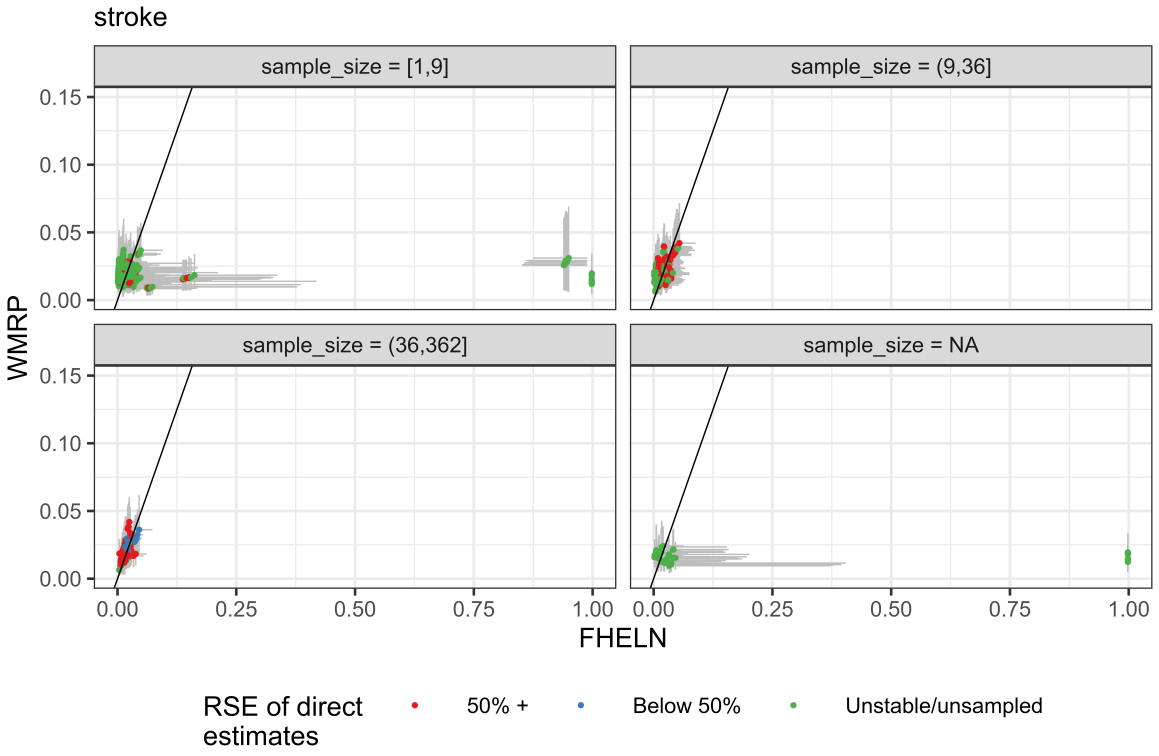
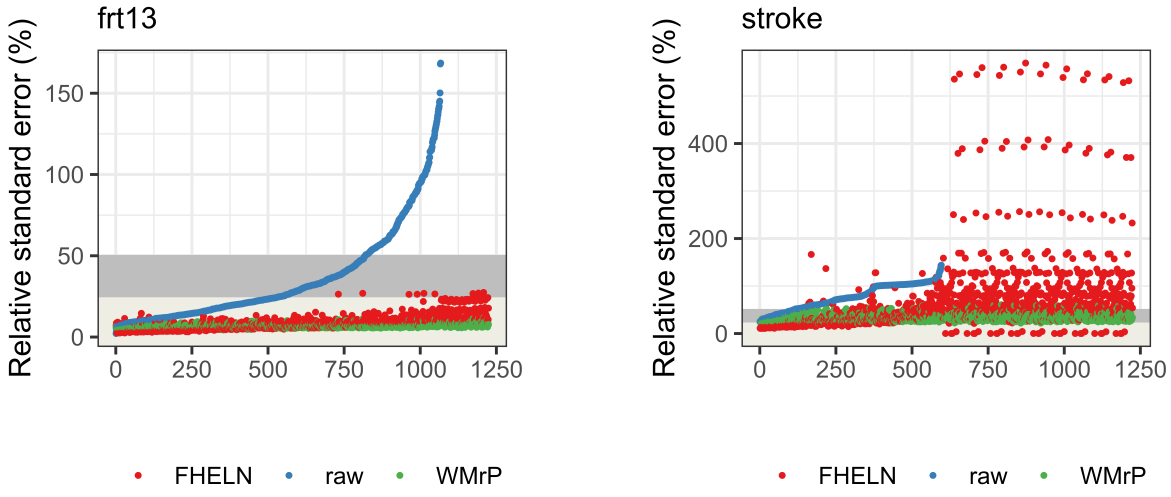



Figure 16: Comparison of modelled prevalence estimates of `stroke` at the LGA level from 2010–2019. The plots compare the area-by-year estimates from the WMrP_ST and FHELN_ST models by displaying the posterior medians and credible intervals on both axes (e.g. a single point gives the posterior medians and 95% credible intervals from both the WMrP_ST and FHELN_ST models). The four quadrants indicate different area-by-year samples sizes and the colours describe the RSEs of the direct estimates. The bottom left quadrant compares the prevalence estimates for those area-by-year combinations with no sample data. Note that $[1, 9]$ denotes all sample sizes from 1 to 9 inclusive, whilst $(9, 36]$ denotes sample sizes *above* 9 but less than 36.




(a) (b)

Figure 17: Plot (a) compares the relative standard errors (RSEs) of the direct prevalence estimates of fruit consumption to prevalence estimates from the FHELN_ST and WMrP_ST models. Estimates are ordered according to the RSE of the direct estimates. The light grey section indicates prevalence estimates with RSEs below 25%, while the dark grey section indicates prevalence estimates with RSEs between 25% and 50%. Plot (b) compares the RSEs for stroke.

***Tech talk!* Benchmarking in SAE**

Assume a reliable Western Australia state prevalence estimate is $\hat{p}_t^{\text{State}}$ for time point $t$. Ideally we would like the following to hold, where $W_{it}^{\text{B}} = \frac{N_{it}}{N_t}$.

$$\hat{p}_t^{\text{State}} = \sum_i W_{it}^{\text{B}} p_{it}$$

That is the state prevalence, $\hat{p}_t^{\text{State}}$, is equal to the weighted sum of the prevalence estimates, $p_{it}$. However, in practice, this never holds.

Therefore, using ratio benchmarking (Rao and Molina 2015), we multiply each area-by-year prevalence estimate by the following common adjustment factor,

$$c_t^{(d)} = \frac{\hat{p}_t^{\text{State}}}{\sum_i W_{it}^{\text{B}} p_{it}^{(d)}},$$

for each posterior draw.

## 5.4 Model/variable selection

Although convergence and model checking are necessary steps for all Bayesian models, model selection will be most applicable when modelling the survey data. The performance of small area estimation models rely on a set of well-chosen covariates and random effects. Thus, model/variable selection is an important topic for small area estimation and requires considerable time and effort.

There are a large amount of resources that describe methods of variable selection in the Bayesian framework. The current state of the art being the Widely Applicable Information criteria (WAIC) (Vehtari et al. 2017), which is estimated using all the posterior draws from MCMC. For more information on Bayesian variable selection, we recommend the accessible introduction in Chapter 10 of McElreath (2020).

### 5.4.1 MrP_ST

Given the high computational burden required to fit even a single Bayesian MrP model, there has been significant work in automating variable selection techniques for MrP-style models (Ornstein 2020; Si et al. 2020). However, a detailed investigation of these approaches are beyond the scope of this project.

Instead of comparing multiple MrP models via WAIC, which would be extremely computationally costly, we recommend an efficient alternative: use a frequentist estimation procedure for variable selection. This can be carried out in the R package `lme4` (see Code 11), and is a recommended approach in Goldstein (2011). Likewise, Ghitza and Gelman (2013) used frequentist methods (e.g. `lme4`) in their MrP work. Although the models fit in `lme4` are *not* the same as those fitted in Code 7, computation is around 320 times faster: the key benefit to using this approach.

Rao and Molina (2015), like others (Tzavidis et al. 2018), recommend the Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) [21] as frequentist model selection tools for SAE hierarchical models. Note that the conditional AIC (cAIC) is generally preferred over the AIC, however, for Bernoulli likelihoods, the cAIC requires bootstrapping (Säfken et al. 2018) which can be extremely computationally expensive for big data such as the temporal HWSS survey data.

The validity of our suggested approach stems from the following assumptions:

- Fixed effect coefficient estimates are generally unaffected by random effects. Thus, by using the standard normal random effects in `lme4` as opposed to the spatial and temporal random effects given in (5.1), we should recover similar regression coefficients.
- By using sufficiently uninformative priors, fixed effect coefficient estimates are generally similar in Bayesian or frequentist settings.
- By using AIC and BIC our choice of covariates should be relatively similar to those we would make using `nimble` (Goldstein 2011).

[21] Please see Gelman et al. (2014b) for an introduction to the AIC, BIC and WAIC.

In addition to the above suggestion, see Figure 13 for an example of how ROC curves can be used in conjunction with AIC and BIC to select the preferred logistic model.

```
# With area-level covariates
    fit1 <- glmer(y ~ ageg*sex + RA_Name + IRSD_5 +
            Tot_P_FP + sqrt(Tot_Indigenous_PP) +
            # standard random effects for area's
            (1|M_id) +
            # standard random effects for time point's
            (1|T_id),
            # specify the Bernoulli family
            family = binomial,
            data = df,
            # sample scaled weights
            weights = w_ss)

# no area-level covariates
    fit2 <- glmer(y ~ ageg*sex +
            (1|M_id) + (1|T_id),
            family = binomial,
            data = df,
            weights = w_ss)

    # Compare AIC and BIC
    AIC(fit1); AIC(fit2)
    BIC(fit1); BIC(fit2)
```

R Code 11: Variable selection for MrP models using the frequentist R package `lme4`. Note that these frequentist models still take around 20 minutes to fit, given the size of the individual-level survey data.

In addition to selecting covariates that provide smaller AIC and BIC, one should continue to investigate how the modelled prevalence estimates behave and ensure these align with known associations. The estimates should be plausible; if they are not then the variables are inadequate or the model has been incorrectly specified.

### 5.4.2 FHELN_ST

Since the area-level models can be fitted far faster than the individual-level models, we recommend using Bayesian variable selection methods. Fortunately `mcmcsae` automatically calculates WAIC and this can be accessed using `compute_WAIC(fit)` (Das et al. 2021). Models with smaller WAIC should be preferred.

***Consider!* Small area estimation for non-binary variables**

The methods described in Section 5 can be easily applied to continuous outcomes with some small adjustments.

At the individual level, the WMrP_ST model can be applied to continuous variables, $y_{jit} \in \mathbb{R}$, by replacing the first two lines of (5.1) with,

$$\begin{aligned} y_{jit} &\sim N\left(\mu_{jit}, \sigma^2\right)^{w_{jit}} \\ \mu_{jit} &= \alpha + \mathbf{X}_{jit}\boldsymbol{\beta} + \theta_i + \gamma_t, \end{aligned}$$

and placing a prior on $\sigma$.

At the area-level, assume that $\hat{\mu}_{it} \in \mathbb{R}$ and $\hat{\psi}_{it} \in \mathbb{R}^+$ are the direct estimates and sampling variances for area $i$ and time $t$, respectively. The FH model is specified by replacing the first two lines of (5.8) with

$$\begin{aligned} \hat{\mu}_{it} &\sim N\left(\mu_{it}, \hat{\psi}_{it}\right) \\ \mu_{it} &= \alpha + \mathbf{X}_{it}\boldsymbol{\beta} + s_i + v_i + \gamma_t, \end{aligned}$$

where $\mu_{it}$ is the estimate for area $i$ and time $t$.

# 6 Burden of Disease data

Burden of disease is generally reported via two key metrics; years of life lost (YLL) and years lived with disability (YLD). Disability-adjusted life years (DALY), is the summation of YLL and YLD and thus we restrict our discussion to models for YLL and YLD only.

There is a significant dearth of literature relating directly to Bayesian small-area modelling of the burden of disease, but MacNab (2007) shows that the class of models used for modelling YLLs and YLDs are identical to those used in standard disease mapping applications (see Section 3). We extend some of these approaches to accommodate any non-registry-based data (see Tables 12 and 13). Similar to the administrative data, models should be fit to females, males and persons separately. The exception is the WMrP_ST model, which should be fitted to all the survey data.

| | | Input data by | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **BoD Metric** | Area | Year | Age | **Input data** | **Offset term** | **Key model output calculation** | **Software** | **Code** | **Eq.** |
| **ASRA_ST** | YLL, YLD | ✓ | ✓ | ✓ | Counts/Point prevalence | Population/adjusted population | Fitted counts (then calculate YLL or YLD) | `nimble` | 5 | (4.5) |
| **ASRAME_ST** | YLD | ✓ | ✓ | ✓ | Prevalence estimates and sampling variances | Adjusted population | Fitted counts (then calculate YLL or YLD) | `nimble` | 15 | (6.1) |
| **WMrP_ST** | YLD | Individual-level survey data | | | Binary outcome | NA | Fitted probabilities[§] | `nimble` | 7 | (5.1) |

Table 12: Summary of models for burden of disease data. [§]Required point prevalence estimates are derived from the fitted probabilities given by the WMrP_ST model and then used to derive YLDs. All models can be used to derive ASYLDs and ASYLLs (see Section 8.3). Approximate run time for these models is of the order of days to weeks.

## 6.1 YLL

To model YLL, one requires raw mortality counts for each area, year and age (AYA). Smoothing of the raw mortality counts proceeds identically to the ASRA_ST model used for administrative data (see Table 4), where one also requires the population for each AYA (see Section 4.2.2).

To derive YLLs by area and year, we multiply the fitted counts, $\mu_{ita}$, from the ASRA_ST model by a standard life expectancy table (which is generally given by single ages).

$$YLL_{it} = \sum_a \mu_{ita} L_a$$

As shown in (8.3), $L_a$ denotes the life expectancy at age $a$.[22]

Correct calculation of the smoothed YLLs requires raw counts and populations by single ages. This requirement can make the input mortality data very large for the ASRA_ST model. From our empirical investigations, we found that the estimated YLLs were extremely similar

[22]Life expectancy is defined as the expected number of years until death at age $a$.

when collapsing the single age data to data with 18 age groups instead. Collapsing the data involves summing the corresponding $y_{ita}$'s and $N_{ita}$'s and calculating the median of the life expectancies.[23]

Regardless of the age groupings used, the ASRA_ST model can also be used to derive age-standardised YLLs (ASYLL), smoothed mortality counts and age-standardised mortality ratios (ASMR) (see Section 8.3), through using the posterior draws of the fitted counts.

For details of fitting the ASRA_ST model in `nimble`, see Section 4.2.2. When fitting single ages, we recommend using age as a continuous (as opposed to a categorical) covariate and including a quadratic term to accommodate the faster decline of overall health in older age. Code 12 shows how to create the QR decomposition for use in Code 5. See Section 8.8 for details on the QR decomposition.

```r
# prepare design matrix for ASRA_ST model with single years
xdm <- model.matrix(~age_single_year +
                    I(age_single_year^2),
                    data = df)
QR <- jf$getQRDecomp(xdm)
```

R Code 12: Constructing the design matrix (and QR decomposition) where age is added as a continuous quadratic effect. Note that $q = 2$ here.

## 6.2 YLD

Unlike YLLs, which are derived from registry mortality data, YLDs are generally derived from prevalence data, which themselves are estimated from a range of data sources including registries and surveys. We denote the point prevalence (PP) for age $a$, area $i$ and time $t$ as $y_{ita}$. To generate smoothed YLDs by area and year, one requires the posterior distribution of $\mu_{ita}$, which can be derived from one of three recommended models: ASRA_ST, ASRAME_ST or WMrP_ST (see Table 12).

Following the flowchart in Figure 18, the recommended model (shown in yellow) and any pre-processing (shown in red) depends on the type (individual-level or aggregated) and availability (with or without error) of prevalence data. In this project, we recognise four distinct types of PP data which are classified in Table 13.

[23] We acknowledge the crudeness of collapsing life expectancy this way. In practice, we would recommend experimenting with how results differ when using the mean or median of life expectancy.

| | | | Prevalence derived from | | Input data by | | |
|---|---|---|---|---|---|---|---|
| **Data type** | **Point prevalence estimate** | **Error**‡ | Survey data | Registry data | Area, Year, Age | Individual-level | **Model** |
| 1 | ✓ | | | ✓ | ✓ | | ASRA_ST |
| 2 | ✓ | | ✓ | | ✓ | | ASRA_ST |
| 3 | ✓ | ✓ | ✓ | | ✓ | | ASRAME_ST |
| 4 | | | ✓ | | | ✓ | WMrP_ST |

Table 13: Overview of the four kinds of prevalence data. ‡ Sampling variance for the point estimate.

As touched on above, modelling prevalence data requires several pre-processing and post-processing steps depending on the data and model. In the pre-processing stage, we must ensure that any error is accommodated (e.g. simulated) and all PP values are valid integers by applying the non-integer count adjustment (see Section 8.7). We'll discuss simulating the point prevalence distributions in Section 6.2.2. Be aware that pre-processing is not required when we have individual-level data.

YLDs are calculated by weighting the fitted PP according to the severity of the condition. These disability adjustments are applied by first splitting the AYA fitted PPs into health states. After applying these adjustments, the final area-by-year YLDs and age-standardised YLDs (ASYLD) can be easily derived by summing over the health states (see Section 8.3 and more specifically, (8.5)). Observe that the same post-processing steps are applied regardless of the model used to derive the AYA fitted PPs (see Figure 18). Code 13 illustrates these post-processing steps.

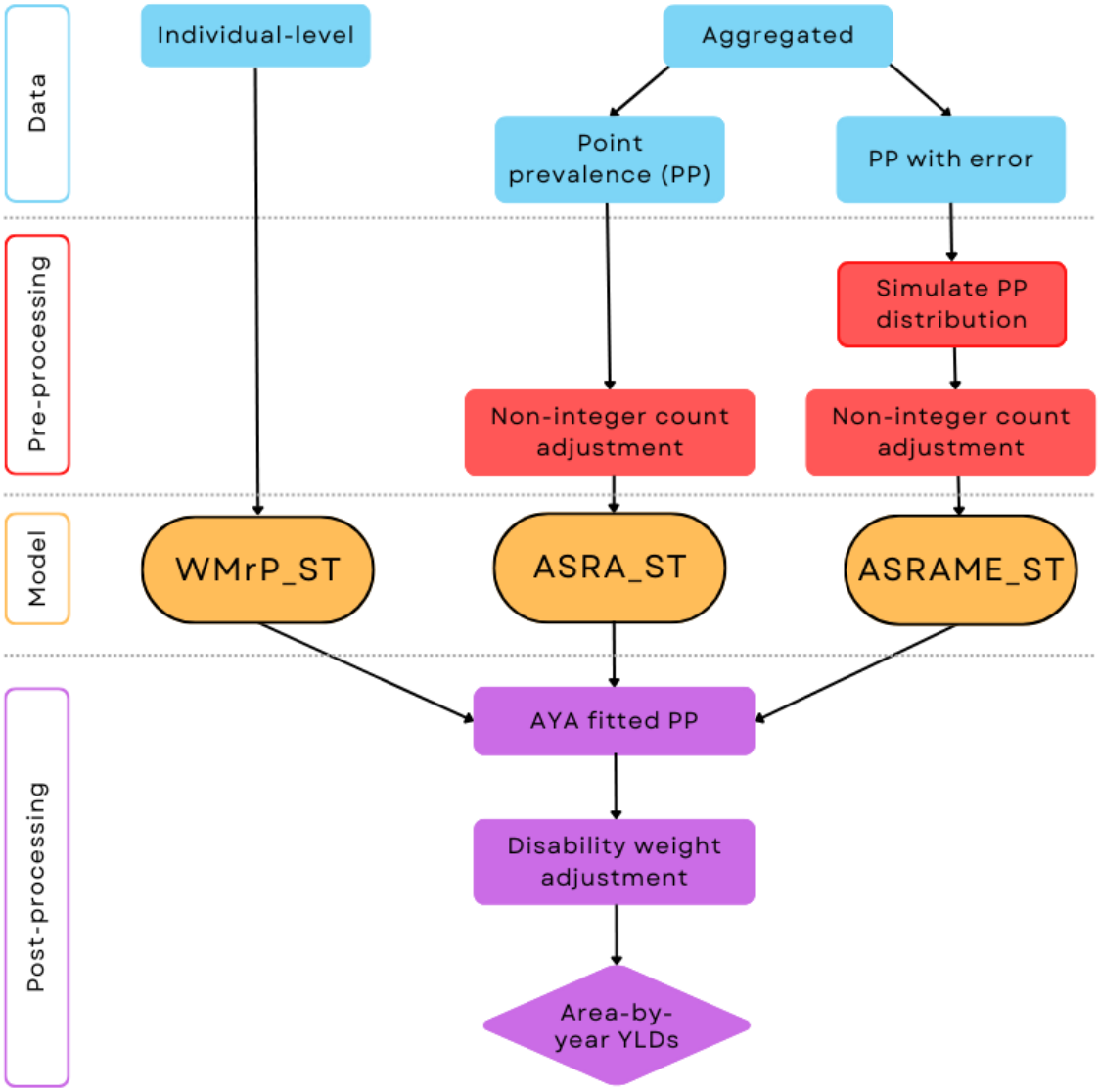



Figure 18: Flowchart illustrating how models for YLD should be selected based on the data. Observe how *all* models provide age, year and area (AYA) fitted point prevalence (PP) estimates (given as the posterior draws). Regardless of the model used to derive these, we apply the disability adjustment to the posterior draws of the AYA fitted PP, to obtain the area-by-year YLDs. Details of simulating the PP distributions can be found in Section 6.2.2, the non-integer count adjustment in Section 8.7 and disability adjustment in Section 8.3 and Code 13. Details of the likely data sources (blue boxes in the flowchart) can be found in Table 13.

```r
# create Z matrix of disability and prevalence weights
    # Z has H rows and 1 column
    Z <- matrix(with(HS_prev, HS_prev*disability_weight), ncol = 1)
        # HS_prev is a dataframe of H rows

# complete the for-loop
    # matrix is number of iterations (D) by n_obs (MTA)
    YLD_MTA_draws <- matrix(NA, nrow = D, ncol = n_obs)
    for(i in 1:D){
        # repeat each observation for each health state
        cur_it <- matrix(rep(mu_draws[i,], H), ncol = H, byrow = F)
            # cur_it is n_obs (MTA) by H
        YLD_MTA_draws[i,] <- as.numeric(cur_it %*% Z)
            # (n_obs times H) (H times 1) =
            # (n_obs times 1) -> aka a vector
    }

# Collapse across age
    # function for each iteration of fitted draws
    foo <- function(x){aggregate(x, # repeat for each hlth_st
                                 list(df$MT_id), # same for area
                                      sum)$x}
    # collapse to area and time level
    yld_MT_draws <- t(apply(YLD_MTA_draws, 1, foo))
```

R Code 13: Deriving the posterior YLDs from fitted posterior draws, (`mu_draws`).

### 6.2.1 Applying the ASRA_ST model to prevalence data

Although point prevalence data for some conditions are estimated from survey data, similar data for other conditions can be derived from hospital or registry data. For registry point prevalence data (data type 1), we simply use the ASRA_ST model to estimate smoothed YLDs. We also recommend the ASRA_ST model for point prevalence data (data type 2) where the uncertainty is not available or cannot be quantified. Data types 1 and 2 take the middle route in Figure 18.

### 6.2.2 Model: ASRAME_ST

For some of the burden of disease data (data type 3 in Table 13), year by area by age point prevalence data is derived by applying age-specific prevalence rates for the entire state to

each year and area population. See Table 14 for an example of the data structure.

| age_p_hat | age_psi_hat | age | M_id | T_id | N | y_hat |
|---|---|---|---|---|---|---|
| $\hat{p}_a$ | $\hat{\psi}_a$ | $a$ | $i$ | $t$ | $N_{ita}$ | $\hat{y}_{ita}$ |
| 0.0527 | 0.0018 | 3 | 1 | 1 | 1331 | 70.0775 |
| 0.0603 | 0.0008 | 4 | 1 | 1 | 1293 | 77.9305 |
| 0.1382 | 0.0030 | 5 | 1 | 1 | 1019 | 140.8622 |
| 0.0858 | 0.0010 | 6 | 1 | 1 | 1128 | 96.8247 |
| 0.1303 | 0.0012 | 7 | 1 | 1 | 1007 | 131.1961 |
| 0.0527 | 0.0018 | 3 | 2 | 1 | 2355 | 123.9914 |
| 0.0603 | 0.0008 | 4 | 2 | 1 | 2514 | 151.5215 |
| 0.1382 | 0.0030 | 5 | 2 | 1 | 2810 | 388.4423 |
| 0.0858 | 0.0010 | 6 | 2 | 1 | 3455 | 296.5687 |
| 0.1303 | 0.0012 | 7 | 2 | 1 | 3560 | 463.8116 |
| 0.0527 | 0.0018 | 3 | 3 | 1 | 213 | 11.2145 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Table 14: Example data as input to the ASRAME_ST model. In the table, $\hat{p}_a$ and $\hat{\psi}_a$ are the prevalence (proportion) estimates and sampling variances for age $a$, respectively. Moreover, $\hat{y}_{ita}$ and $N_{ita}$ are the point prevalence (count) estimates and populations in age $a$, area $i$ and time $t$, respectively. Observe that the prevalence estimates and variances are constant regardless of the area and year, while the point prevalence, $\hat{y}_{ita}$, is unique for each row.

To accommodate the uncertainty of the age-specific prevalence rates used in these calculations, we assume access to prevalence estimates, $\hat{p}_a \in (0, 1)$, and sampling variances, $\hat{\psi}_a$, or point prevalence estimates, $\hat{y}_a$, and sampling variances, var $(\hat{y}_a)$, for age group $a$ . We can easily convert between point prevalence data (count e.g., estimated number of people with asthma in an area) and prevalence data (proportions e.g., proportion of people with asthma in an area) using

$$\hat{p}_a = \frac{\hat{y}_a}{N_a}, \hat{\psi}_a = \frac{\text{var}\,(\hat{y}_a)}{N_a^2}.$$

To derive AYA point prevalence data (PP) we multiple the prevalence by the population,

$$\hat{y}_{ita} = \hat{p}_a N_{ita},$$

a method which can be easily verifed from Table 14.

With access to a point estimate and measure of uncertainty (i.e. error) we can consider a *distribution* of the PP as the observed data when modelling with the ASRA_ST model. Although one could assume normality for the distribution of the PP data directly (i.e. $\hat{y}_{ita} \sim N\,(y_{ita}, \text{var}\,(\hat{y}_{ita}))$, we found that using the prevalence (e.g. proportion) data initially and *then* transforming these to counts provides a more flexible approximation to the PP distribution.

Figure 19 illustrates that a Beta distribution (being naturally bounded between $0$ and $1$) is a great candidate distribution for the underlying prevalence (e.g. proportion) data, by comparing the distribution of the PP for two extremes. For very small PP values, using

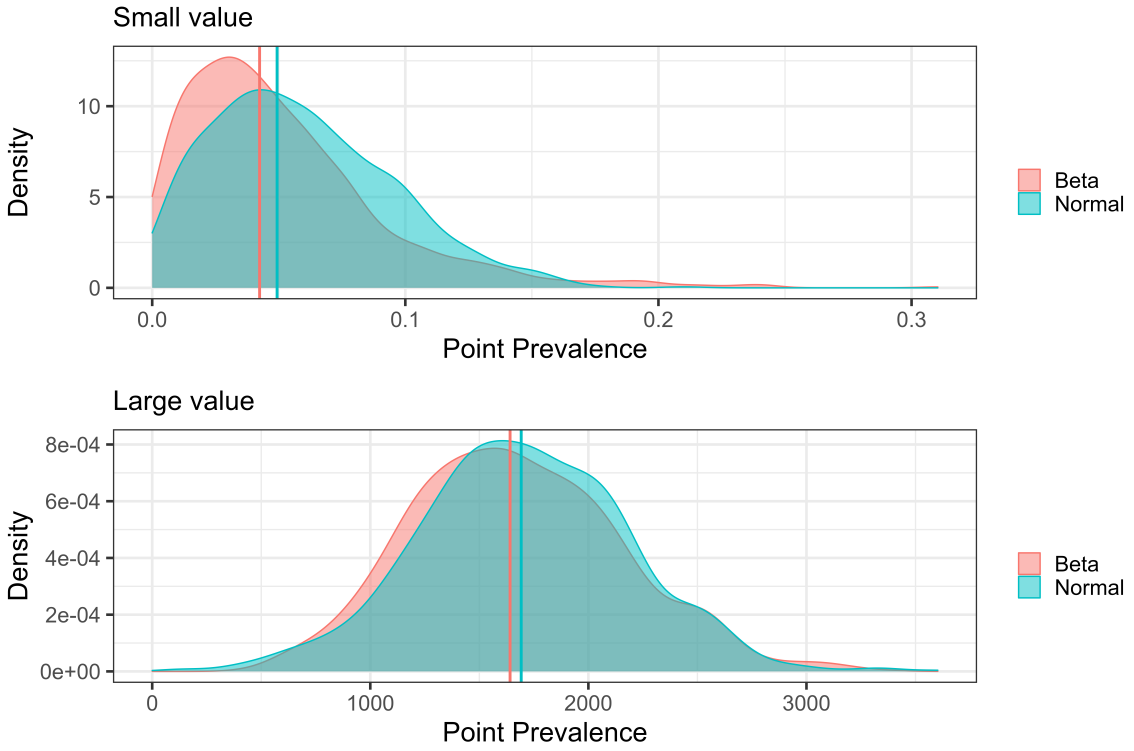



Figure 19: Example of the approximate distribution of two point prevalence (counts) — one small and one large — using $U = 10,000$. The plots compare approximating the distributions using a Beta distribution as opposed to a normal distribution. The vertical lines represent the medians of the densities. The top plot displays the approximate density when the point prevalence is very low, while the bottom plot compares the Beta and normal simulations when the point prevalence is very high.

a normal distribution as opposed to a Beta gives a different distribution: the tails of the normal density are far greater than that of the Beta (top plot). Furthermore, we expect the distribution of point prevalence (counts) to be very skewed as the values approach zero: a bell curve (aka a normal) does not exhibit this behaviour. For large PP estimates (bottom plot in Figure 19), the Beta and normal simulate very similar distributions.

Here we describe the pre-processing step of simulating the PP distribution (see Figure 18). In this project we use a Beta distribution to approximate the distribution of the prevalence for each age group. To achieve this, first simulate $U$ random draws from the correct Beta distribution, $p_a^{(u)} \sim \text{Beta}\left(\hat{p}_a, \hat{\psi}_a\right)$ and multiply each draw by its corresponding area by age by year population, $N_{ita}$. The result is a matrix with $M \times T \times A$ rows and $U$ columns where row $ita$ gives an approximation to the distribution of the PP (counts) for age $a$, area $i$, and time $t$. We will eventually use a model very similar to the ASRA_ST model (see Section 4.2.2), and so we use the non-integer count adjustment trick (see Section 8.7) on the matrix of PP draws to return $\tilde{\mathbf{y}} \in \mathbb{R}^{(M\times T\times A)\times U}$ and $\tilde{\mathbf{N}} \in \mathbb{R}^{(M\times T\times A)\times U}$. For simplicity we take the median of the $U$ $\tilde{\mathbf{N}}_{ita}$'s for use in the models, and denote them $\widehat{\tilde{N}}_{ita}$ (see line 14 in Code 14).

Fortunately, the above steps are all performed within the user-defined `jf$rbetaMP(.)`

function, which returns a list of the required matrices.

```
dist_counts <- jf$rbetaMP(# Number of simulations per observations
                          U = 100,
                          # vector of p_hat
                          mu = df$age_p_hat,
                          # vector of psi_hat
                          var = df$age_psi_hat,
                          # vector of population sizes
                          pop = df$N)

# Matrix of y_tilde (n_obs x U) (rounded point prevalence)
y_tilde = dist_counts$y_tilde

# take the median of the draws of N_tilde (vector of length n_obs)
N_hat_tilde = apply(dist_counts$N_tilde, 1, median)
```

R Code 14: Example code to simulate $U$ draws from the approximate distribution of point prevalence distribution prior to using the ASRAME_ST model.

Given that the input data for a single observation is now a *vector*, we must be wary that the certainty of the posterior distribution will now be dependent on $U$. To fix this, we weight our likelihood term with $1/U$; an approach similar to that described in Section 5.2.1. We provide a user-written `nimble` function, `dpois_me_v(.)`, which calculates the correct density.[24] The only difference between Code 5 and the ASRAMA_ST model is the definition of the likelihood and the use of $\widehat{\widetilde{N}}_{ita}$ as the offset. Simply swap lines 3-16 in Code 5 for those in Code 15.

[24]Note that setting `U = 1` in `dpois_me_v(.)` would give the standard Poisson density, but vectorised.

```
for(i in 1:n_obs){
    # likelihood
    y_tilde[i,1:U] ~ dpois_me_v(mu[i], U = U)
    # mean - linear predictor
    log(mu[i]) <- log(N_hat_tilde[i]) + alpha
    # Fixed effects using the inner product
    + inprod(B_qr[1:q], Q_ast[i,])
    # BYM2 spatial term
    + theta[M_id[i]]
    # ICAR temporal term
    + gamma[T_id[i]]
    # Space time term
    + delta[MT_id[i]]
}
```

R Code 15: Difference in BUGS syntax between the ASRA_ST and ASRAME_ST model.

By letting $\tilde{\mathbf{y}}_{ita} = \left(\tilde{y}_{ita}^{(1)}, \ldots, \tilde{y}_{ita}^{(U)}\right)$ be the vector of $U$ simulations of the point prevalence in age group $a$, area $i$ and time $t$, the full measurement error Poisson model is constructed as follows.

$$
\begin{aligned}
\tilde{\mathbf{y}}_{ita} &\sim \text{Poisson}\left(\mu_{ita}\right)^{1/U} \\
\log\left(\mu_{ita}\right) &= \log\left(\tilde{\tilde{N}}_{ita}\right) + \alpha + \mathbf{X}_{ita}\boldsymbol{\beta} + \theta_i + \gamma_t + \delta_{it} \\
\boldsymbol{\theta} &\sim \text{BYM2}\left(\mathbf{W}^{\text{S}}, \rho, \kappa, \sigma_\theta^2\right) \\
\boldsymbol{\gamma} &\sim \text{ICAR}(\mathbf{W}^{\text{T}}, \sigma_\gamma^2) \\
\delta_{ti} &\sim N\left(0, \sigma_\delta^2\right) \\
\rho &\sim \text{Uniform}(0, 1) \\
\sigma_\theta, \sigma_\gamma, \sigma_\delta &\sim \text{Gamma}(2, 0.5) \\
\alpha, \boldsymbol{\beta} &\sim N\left(0, 1000^2\right)
\end{aligned}
\tag{6.1}
$$

Figure 20 illustrates how the measurement error model affects the posterior standard deviation (uncertainty) of smoothed YLD estimates. Note that the measurement error model took over 21 times longer to fit than the standard ASRA-style model.

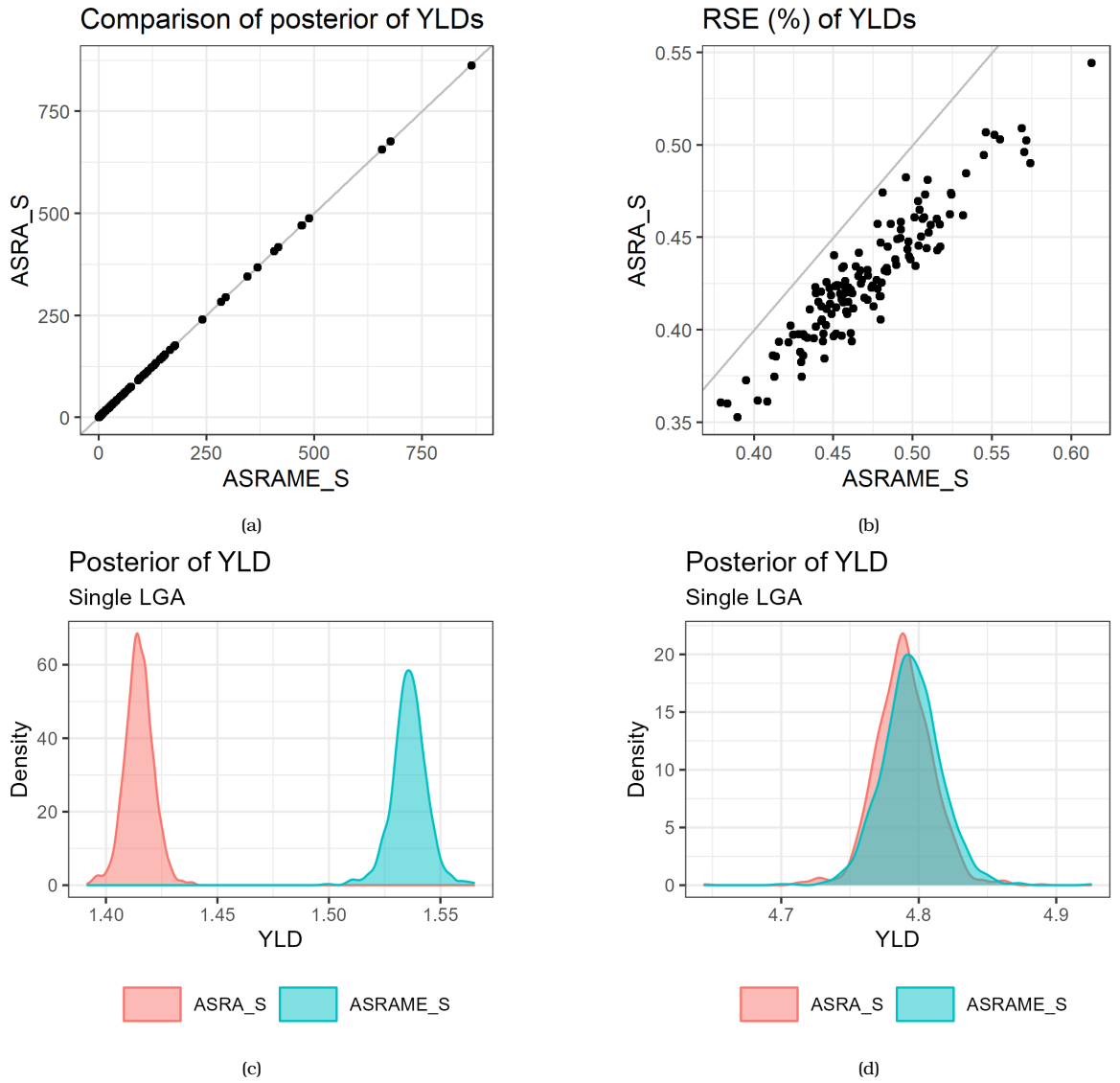



Figure 20: Comparison of the smoothed YLDs for the 137 LGAs in Western Australia from a spatial only version of the ASRA_S and ASRAME_S models. The data is the prevalence of backpain in males for 2015. Plot (a) compares the posterior medians of the smoothed YLDs from the two models. Observe the perfect agreement between both approaches as *all* points sit on the grey line of equivalence. Plot (b) compares the posterior RSE for the two models. As expected, the RSEs for the ASRAME_S model are higher than those for the ASRA_S model. To help compare the two models, plots (c) and (d) each display the posterior YLDs of two different and randomly selected LGAs. Plot (d) shows strong consistency between the two models (i.e. overlapping posterior distributions), whilst plot (c) illustrates when the two models produce *very* different posterior YLDs. However, take note of the scale of the YLDs in plots (c) and (d); we may not change our policy decision if the modelled YLD is 1.4 as opposed to 1.53.

### 6.2.3 Applying the WMrP_ST model to prevalence data

The final scenario (see Table 13) is when one has access to the individual-level HWSS survey data from which one can derive prevalence point estimates directly (data type 4). In this case, YLD modelling becomes SAE. Note that we cannot use the FHELN_ST model as we require estimates by age.

Assuming access to the posterior draws of $\hat{p}_{ita}$ — the proportion of people in age group $a$, area $i$ and time $t$ with the condition of interest — we can derive the area-by-year YLDs for the $d$th posterior draw using the following,

$$YLD_{it}^{(d)} = \sum_a \sum_h \hat{p}_{ita}^{(d)} N_{ita} p_h e_h.$$

Deriving the modelling estimates, $\hat{p}_{ita}$, follows that described in Section 5.2.1, so no further details are given here.

# 7 Conclusion

We have recommended different Bayesian models and processes for three types of data, namely, administrative data (Section 4), survey data Section (5) and burden of disease data (Section 6), based on:

- our previous similar projects such as the Australian Cancer Atlas (Duncan et al. 2019),
- extensive research in the application of Bayesian modelling methods in disease mapping and epidemiological studies, and
- feedback from epidemiological and spatial analysis experts from the DOHWA.

Some models are computationally demanding, and all models require careful checking of convergence and model fit (Section 2), but the benefits are huge. Bayesian ST modelling can:

- provide robust estimates
  - for all areas (e.g. SA2s and LGAs)
  - for areas where conventional statistical modelling and epidemiological analysis would fail
- provide measures of uncertainty to give users the confidence when using such estimates in health policy development and health program evaluation
- protect data confidentiality

Bayesian disease mapping for small area analysis is feasible, useful and reliable across all desired measures and datasets for this project. A summary of our model recommendations can be found in Figure 22.

# 8 Appendix

Please refer to Tables 1 and 2 for a summary of the indices and notation used in this report. Further notation details can be found in the next section.

## 8.1 Introduction to mathematical notation

**Vectors and Matrices** As per convention, vectors and matrices will be bold while scalars will not be. For example, $\mathbf{y}$ could denote the following vector (collection) of numbers, $(4, 5, 6)$, while $x$ (unbolded) could represent a single number (or scalar), for example, 2. Generally we define the size of vectors and matrices using the $\in$ notation, which means *in*. For example, we could define $\mathbf{y}$ one of two ways

$$\begin{aligned} \mathbf{y} &= (4, 5, 6) \\ \mathbf{y} &\in \mathbb{R}^3, \end{aligned}$$

where $\mathbb{R}$ denotes the real numbers. In terms of understanding statistical models, the second line above is more succinct and would be interpreted as follows: "The object $\mathbf{y}$ is a vector with 3 elements, all of which are real numbers."[25]

Note that in practice, the *actual* numbers will never be given. For example, you'll see notation such as $\mathbf{y} = (y_1, \ldots, y_n)$, which is interpreted as: "The object $\mathbf{y}$ is a vector with $n$ real-valued elements". Remember we could also write this as $\mathbf{y} \in \mathbb{R}^n$, where $n$ may denote our sample size.

Matrices can be defined in a similar way. Consider the following matrix.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

An efficient way to describe this matrix is by using the $\in$ notation as before. We would write, $\mathbf{A} \in \mathbb{R}^{3\times 4}$, which would be interpreted as: "The object $\mathbf{A}$ is a matrix with 3 rows and 4 columns, all of which are real numbers."

It is easy to select specific elements of vectors and matrices using indexes. Consider again the vector $\mathbf{y} = (4, 5, 6)$. By writing $y_2$, we recover the second element, 5, of the vector $\mathbf{y}$. Now that we have indexed (or chosen) a specific value from the vector, the object is a scalar and thus is no longer bold. To recover a scalar from a matrix (say $\mathbf{A}$ from above), we must provide two indices; the first for the row and the second for the column. For example, $A_{12} = 1$. If only a single index is provided, assume that we are selecting a row. As a row of a matrix is a vector, the notation remains bold. For example, $\mathbf{A}_1 = (0, 1, 0, 0)$.

[25] In this report we consider all vectors as column matrices. For example, $\mathbf{y} \in \mathbb{R}^3$, implies $\mathbf{y} \in \mathbb{R}^{3\times 1}$

**Summation** The notation for addition is $\sum$. Suppose we wish to sum across each row of the matrix $\mathbf{A}$, where $i$ indexes the rows and $k$ indexes the columns. The notation to communicate this is:

$$A_i = \sum_k A_{ik}$$

which is a succinct way to write $A_i = (A_{i1} + A_{i2} + A_{i3} + A_{i4})$. We can also be more specific and write the summation as:

$$A_i = \sum_{k=1}^{4} A_{ik}$$

Finally, we can sum all the elements in $\mathbf{A}$ by writing

$$\sum_{i,k} A_{ik}.$$

**Matrix multiplication** It will be useful to have at least a rudimentary understanding of matrix multiplication to understand the notation in this report. Consider the following two matrices,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 4}, \mathbf{B} = \begin{bmatrix} 2 \\ 1 \\ -1 \\ 8 \end{bmatrix} \in \mathbb{R}^{4\times 1},$$

and their multiplication,

$$\mathbf{C} = \mathbf{AB} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{3\times 4} \times \underbrace{\begin{bmatrix} 2 \\ 1 \\ -1 \\ 8 \end{bmatrix}}_{4\times 1} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 9 \end{bmatrix}}_{3\times 1}.$$

which is equivalent to

$$C_{ik} = \sum_{h=1}^{4} A_{ih} B_{hk}$$

for the $i$th row and $k$th column of $\mathbf{C}$.

It is pivotal to understand matrix multiplication to be able to write linear models in a very succinct manner. Consider the following vectorized linear model (i.e. where all the elements of the formula are either vectors or matrices).

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

In this case, the objects are of the following dimension, $\mathbf{y} \in \mathbb{R}^N, \mathbf{X} \in \mathbb{R}^{N \times q}, \boldsymbol{\beta} \in \mathbb{R}^q, \epsilon \in \mathbb{R}^N$, where $q$ is the number of covariates in the model and $N$ is the sample size. Note that by multiplying $\mathbf{X}$ and $\boldsymbol{\beta}$, we recover a column matrix of size $N$. That said, in most cases it is clearer to explicitly index the observations. Observe that now $y_i$ and $\epsilon_i$ are scalars and thus no longer bold.

$$y_i = \mathbf{X}_i\boldsymbol{\beta} + \epsilon_i$$

However, note that both $\mathbf{X}_i$ and $\boldsymbol{\beta}$ remain bold as both are still vectors. $\mathbf{X}_i$ is a row vector as we have selected the $i$th row of $\mathbf{X}$ while $\boldsymbol{\beta}$ remains the same. In this case, the matrix multiplication below (often called the dot product or inner product),

$$\underbrace{\mathbf{X}_i}_{1\times q}\underbrace{\boldsymbol{\beta}}_{q\times 1}$$

gives a matrix of dimension $1 \times 1$; a scalar. The `nimble` equivalent code to achieve the dot product is `inprod(beta[1:q], X[i,])`.

Matrix multiplication can also be used to speed up computation in R and provide cleaner code. Consider the following linear predictor, $\eta_{it}$, for area $i$ and time $t$. We use a spatial, temporal and space-time random effect, see Section 3 for more details. For this simple example, let $M = 3, T = 2$.

$$\eta_{it} = \alpha + \theta_i + \gamma_t + \delta_{it}$$

Now we construct a random effect design matrix as follows,

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(MT)\times(M+T+MT)},$$

along with the following vector,

$$\boldsymbol{\lambda} = (\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\delta}) \in \mathbb{R}^{(M+T+MT)\times 1}.$$

Note that $\mathbf{G}$ has the same number of rows as observations (here $3 \times 2 = 6$) and a column for each of the unique values of each of the random effect (here $3 + 2 + (3 \times 2) = 11$). The column vector, $\boldsymbol{\lambda}$, holds all the random effects in the specified order. Using this matrix and vector, we can rewrite the linear predictor from above in a vectorized form where $\boldsymbol{\eta} \in \mathbb{R}^{MT}$.

$$\boldsymbol{\eta} = \alpha + \mathbf{G}\boldsymbol{\lambda}$$

To illustrate this explicitly, consider the linear predictor value for area $1$ and time $1$. Note that unlike previously, now $\mathbf{G}_{11}$ selects a specific row, rather than a row and column.

$$\begin{aligned} \eta_{11} &= \alpha + \mathbf{G}_{11}\boldsymbol{\lambda} \\ \eta_{11} &= \alpha + [1 \times \theta_1 + \cancel{0 \times \theta_2} + \cancel{0 \times \theta_3} + 1 \times \gamma_1 + \cancel{0 \times \gamma_2} + \\ &\quad 1 \times \delta_{11} + \cancel{0 \times \delta_{12}} + \cancel{0 \times \delta_{21}} + \cancel{0 \times \delta_{22}} + \cancel{0 \times \delta_{31}} + \cancel{0 \times \delta_{32}}] \\ \eta_{11} &= \alpha + \theta_1 + \gamma_1 + \delta_{11} \end{aligned}$$

Observe that we recover the linear predictor we would expect under the original form.

## 8.2 ICAR prior

As described in Section 3.3.1, the ICAR prior for a RE, $s_i$, is described by the following conditional normal distribution,

$$s_i \sim N\left(\frac{\sum_{k=1}^{M} W_{ik}^{\mathrm{S}} s_k}{m_i}, \frac{\sigma_s^2}{m_i}\right)$$

Returning to the example in Figure 7, by using the ICAR prior, the distribution for $s_1$ can be derived as follows. First we observe that area 1 is neighbours with areas 2 and 5. Thus, $m_1 = 2$. Now consider the numerator of the mean, $\sum_{k=1}^{M} W_{ik}^{\mathrm{S}} s_k$. By fixing the row index to 1, we recover,

$$\sum_{k=1}^{M} W_{1k}^{\mathrm{S}} s_k = (0 \times s_1) + (1 \times s_2) + (0 \times s_3) + (0 \times s_4) + (1 \times s_5) + (0 \times s_6)$$
$$\sum_{k=1}^{M} W_{1k}^{\mathrm{S}} s_k = s_2 + s_5$$

Then the mean of the conditional normal distribution for $s_1$ becomes $\frac{s_2+s_5}{2}$: the mean of the random effects of the neighbours. With that we can write the ICAR distribution for $s_i$ as,

$$s_1 \sim N\left(\frac{s_2 + s_3}{2}, \frac{\sigma_s^2}{2}\right).$$

## 8.3 Epidemiology metrics

In this section, we describe the key epidemiology metrics used in this project and present the maths to derive them.

For clarity we present some common notation here. Note that while $y_{ita}$ denotes the raw counts for age $a$ in area $i$ and time $t$, $y_a$ denotes the total raw counts for age $a$ (i.e. summed across all areas and time points).

- $N_a^{2001}$ be the 2001 Australian Standard Population in age group $a$
- $N^{2001}$ be the total 2001 Australian Standard Population
- $y_{ita}$ be the raw counts for age $a$ in area $i$ and time $t$
- $N_{ita}$ be the current population size for age $a$ in area $i$ and time $t$
- $r_{ita} = \frac{y_{ita}}{N_{ita}}$ be the crude rate for age $a$ in area $i$ and time $t$
- $r_a = \frac{y_a}{N_a}$ be the crude rate for age $a$

**Standardised incidence ratio (SIR)** As noted in Section 4, SIRs are identical to SMRs and SRRs.

$$\begin{aligned} E_{it} &= \sum_a r_a N_{ita} \\ SIR_{it} &= \frac{y_{it}}{E_{it}} \end{aligned} \tag{8.1}$$

By construction $\sum_{i,t} E_{it} = \sum_{i,t} y_{it}$.

**Age-standardised rates (ASR)** We use direct standardisation to calculate the ASR for area $i$ and time $t$.

$$\begin{aligned} E_{it}^{2001} &= \sum_a r_{ita} N_a^{2001} \\ ASR_{it} &= \frac{E_{it}^{2001}}{N^{2001}}, \end{aligned} \tag{8.2}$$

**Years of life lost (YLL)** Let $y_{ita}$ be the number of deaths from a condition in age $a$, area $i$ and time $t$. In addition, let $L_a$ be the life expectancy at age $a$. Life expectancy is defined as the expected number of years until death at age $a$.

$$\begin{aligned} YLL_{ita} &= y_{ita} L_a \\ YLL_{it} &= \sum_a YLL_{ita} \end{aligned} \tag{8.3}$$

Age-standardised YLLs (ASYLLs) can be derived as follows.

$$\begin{aligned} YLL^r_{ita} &= \frac{YLL_{ita}}{N_{ita}} \\ E^{2001}_{it} &= \sum_a YLL^r_{ita} N^{2001}_a \\ ASYLL_{it} &= \frac{E^{2001}_{it}}{N^{2001}} \end{aligned} \tag{8.4}$$

Note that $YLL^r_{ita}$ is the years of life lost per person in age $a$, area $i$ and time $t$.

**Years lived with disability (YLD)** Let $y_{itah}$ be the number of persons (with a particular condition) in health state $h$, age group $a$, area $i$ and time $t$. To derive YLDs for this project, we apply a health state specific disability weight, denoted $e_h$.

$$\begin{aligned} YLD_{itah} &= y_{itah} e_h \\ YLD_{ita} &= \sum_h YLD_{itah} \\ YLD_{it} &= \sum_a YLD_{ita} \end{aligned} \tag{8.5}$$

The age-standardised YLD (ASYLD) can be computed as well.

$$\begin{aligned} YLD^r_{ita} &= \frac{YLD_{ita}}{N_{ita}} \\ E^{2001}_{it} &= \sum_a YLD^r_{ita} N^{2001}_a \\ ASYLD_{it} &= \frac{E^{2001}_{it}}{N^{2001}} \end{aligned} \tag{8.6}$$

In practice, we do not have access to counts by health state. Instead, $y_{ita}$ is split into the $H$ health states using $p_h$, which gives the proportion of persons with the disease that are in health state $h$.[26]

$$y_{itah} = y_{ita} p_h$$

We use $y_{ita}$ as input to our models and then apply both adjustments to the fitted values via,

[26] $p_h$ is **not** the proportion of the *total* population in each health state, but the proportion of all persons *with* the condition that are in health state $h$.

$$YLD_{it} = \sum_{a,h} \mu_{ita} p_h e_h.$$

**Disability-adjusted life years (DALY)** DALYs are the summation of YLDs and YLLs.

$$\begin{aligned} DALY_{ita} &= YLL_{ita} + YLD_{ita} \\ DALY_{it} &= YLL_{it} + YLD_{it} \end{aligned} \tag{8.7}$$

We can also derive the age-standardised DALYs (ASDALY).

$$\begin{aligned} DALY^r_{ita} &= \frac{DALY_{ita}}{N_{ita}} \\ E^{2001}_{it} &= \sum_a DALY^r_{ita} N^{2001}_a \\ ASDALY_{it} &= \frac{E^{2001}_{it}}{N^{2001}} \end{aligned} \tag{8.8}$$

**Relative Standard Error (RSE)** Assume access to a point estimate, $\hat{p}$, and its variance, $\mathrm{v}(\hat{p})$, which can both be derived from posterior draws.

$$RSE(\hat{p}) = 100 \times \left( \frac{\sqrt{\mathrm{v}(\hat{p})}}{\hat{p}} \right) \tag{8.9}$$

Note that our user-made function `jf$getResultsData()` returns RSEs automatically. RSEs rely on the assumption that the posterior variance is a valid measure of uncertainty. For skewed posterior distributions this may not hold.

## 8.4 Offset term in Poisson models

Let $y_i$ and $N_i$ be the raw count and population, respectively, in area $i$ and $\eta_i$ be the linear predictor. The mean count, $\mu_i$, is modelled using the log link.

$$\begin{aligned} \log(\mu_i) &= \log(N_i) + \eta_i \\ \log(\mu_i) - \log(N_i) &= \eta_i \\ \log\left(\frac{\mu_i}{N_i}\right) &= \eta_i \\ \frac{\mu_i}{N_i} &= \exp(\eta_i) \end{aligned}$$

Observe that $\frac{\mu_i}{N_i}$ is the fitted rate in area $i$. Note that this proof can, of course, be extended to ST settings.

## 8.5 Non-mean centred parameterisation

There are a variety of methods to improve MCMC sampling efficiency. One such method is the non-mean centred parameterisation,[27] where instead of telling `nimble` to sample the vector $\phi$ as

$$\phi_i \sim N(\mu, \sigma),$$

we can instead use

$$\begin{aligned} Z_i^{\phi} &\sim N(0,1) \\ \phi_i &= \mu + Z_i^{\phi}\sigma, \end{aligned}$$

which is equivalent and can be far easier to sample. In many of the ST models we recommend, the distribution for $\phi_i$ would be $N(0, \sigma)$, which means the above form can be simplified to

$$\begin{aligned} Z_i^{\phi} &\sim N(0,1) \\ \phi_i &= Z_i^{\phi}\sigma. \end{aligned}$$

Throughout the BUGS code given in this report you will see this trick used over and over again (for example, lines 21, 43 and 55 in Code 5 alone). In some instances, the non-mean centred parameterisation can completely solve any sampling issues you are having. In practice it is best to check whether the mean-centred parameterisation is more efficient. However for these complex models, we recommend using the non-mean centred parameterisation wherever possible.

## 8.6 ASR Adjustment

Instead of modelling the area by year by age group counts, one can instead model the area by year counts and adjust the population accordingly to ensure that the derived quantities are ASRs. With access to the crude, $R = y/N$, and age-standardised rates, $ASR$, we can model the raw counts aggregated over age groups by observing the following identity, where $c$ is a adjustment factor.

$$c = \frac{R}{ASR} = \frac{\frac{y}{N}}{ASR} \tag{8.10}$$

$$\therefore ASR = \frac{\frac{y}{N}}{c} = \frac{y}{cN} \tag{8.11}$$

[27] Please see Section 13.4 from McElreath (2020) for a deeper introduction.

This result allows us to implicitly model the ASRs for each area and year. To achieve this, we use the raw counts, $y$, as the observed data and $\tilde{N} = cN = y/ASR$, as the offset. When $y = 0$ then both the $ASR$ and $R$ are also zero. In these cases, set the offset to $N$.

## 8.7 Non-integer count adjustment

For particular data in this project, we must round any non-integer counts, $y$, to integers before modelling (see Section 6.2). To ensure that inference is identical after performing the rounding we use the following adjustment.

First, we round the non-integer counts using the ceiling operator,[28] $\lceil y \rceil = \tilde{y}$. Then we introduce a corrective factor, $c = \frac{\tilde{y}}{y}$, which is used to derive the adjusted offset term in our Poisson models.

$$\tilde{N} = cN$$

In most cases, $c$ will be close to 1, unless the raw count is very small. Note that when $y = 0$, $c = 0$ and thus $\tilde{N} = 0$. In these cases, let $\tilde{N} = N$.

The adjustment described above ensures that the rate (which is implicitly modelled using Poisson models - Section 8.4) is the same regardless of the extent of rounding applied to $y$.

$$\begin{aligned} \frac{\tilde{y}}{\tilde{N}} &= \frac{\tilde{y}}{cN} \\ \frac{\tilde{y}}{\tilde{N}} &= \frac{\tilde{y}}{\frac{\tilde{y}}{y}N} \\ \therefore \frac{\tilde{y}}{\tilde{N}} &= \frac{y}{N} \end{aligned}$$

The user-made function `jf$sIntRound(.)` returns a data frame with the raw $y, N$ and the derived $\tilde{y}, \tilde{N}$. See Code 16 for an example of this code.

```
# int_ver is a dataframe with number of rows equal to `nrow(df)
int_ver <- jf$sIntRound(df$point_prevalence, df$N)

# Add N_tilde to data
df$N_tilde <- int_ver$N_tilde

# Add y_tilde to data
df$y_tilde <- int_ver$y_tilde
```

[28] The ceiling operation, $\lceil . \rceil$, rounds non-integers to the closest upper integer which ensures that any $y < 0.5$ are not rounded to zero, but instead to 1.

R Code 16: Example code to perform the non-integer count adjustment.

## 8.8 QR Decomposition

In regression problems, the linear predictor, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)$, can be calculated using $\mathbf{X}\boldsymbol{\beta}$, where the fixed effects, $\boldsymbol{\beta} \in \mathbb{R}^{q\times 1}$, are estimated using MCMC. Efficient MCMC estimation of fixed effects is difficult when the design matrix, $\mathbf{X} \in \mathbb{R}^{n\times q}$, has strongly correlated columns.

To improve convergence of the fixed effects, the design matrix can be factorized using the QR decomposition,[29] which factors the design matrix, $\mathbf{X}$, into an orthogonal matrix, $\mathbf{Q}^{\text{ast}} \in \mathbb{R}^{n\times q}$ (i.e a matrix where all columns are independent), and an upper-triangular matrix, $\mathbf{R}^{\text{ast}} \in \mathbb{R}^{q\times q}$.

$$\mathbf{X} = \mathbf{Q}^{\text{ast}}\mathbf{R}^{\text{ast}}$$

Thus, one can rewrite the linear predictor as follows,

$$\begin{aligned}\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} = \mathbf{Q}^{\text{ast}}\mathbf{R}^{\text{ast}}\boldsymbol{\beta} = \mathbf{Q}^{\text{ast}}\boldsymbol{\beta}^{\text{qr}}\\ \boldsymbol{\beta}^{\text{qr}} = \mathbf{R}^{\text{ast}}\boldsymbol{\beta},\end{aligned}$$

and sample the vector $\boldsymbol{\beta}^{\text{qr}}$ as opposed to $\boldsymbol{\beta}$, which can be considerably more efficient. During sampling, one can easily calculate the actual regression coefficients using the identity, $\boldsymbol{\beta} = (\mathbf{R}^{\text{ast}})^{-1}\boldsymbol{\beta}^{\text{qr}}$ (see line 69 in Code 5). Note that it usually advisable to remove the intercept column and then centre the design matrix prior to taking the QR decomposition. These operations are automatically performed by the `jf$getQRDecomp()` function. See Code 17.

[29] A great introduction is given in the Stan (Stan Development Team 2022) user guide here.

```r
# Construct the design matrix (N times A)
xdm <- model.matrix(~as.factor(age), data = df)

# take the QR decomposition
QR <- jf$getQRDecomp(xdm)

# get the key matrices
Q_ast = QR$QR$Q_ast,
R_ast_inverse = QR$QR$R_ast_inverse

# no intercept
# mean centered (NOT SCALED)
X = QR$X_c
```

R Code 17: QR decomposition for fixed effect design matrices

Using the QR decomposition to derive estimates for out of sample data (i.e. the poststrata dataset for MrP models), requires careful consideration. The centered design matrix ***must*** use the same column means as the original design matrix passed to the model. Fortunately, the `jf$getQRDecomp()` function allows the user to pass a vector of column means to be used in the centring process. More details will be given in the training.

## 8.9 Output from Bayesian software

Figure 21 shows an example output from Bayesian software. The top command, `SIRST_fit$summary` prints the posterior summaries from the `SIRST_fit` object. The `summary` component provides details of the point estimates (blue boxes), uncertainty measures (green boxes) and convergence diagnostics (yellow boxes) for each model parameter (column `variable`). For details please review Section 2.

Note that the `summary` component provides summaries for *all* parameters from the model (729 in this example) which include the variance terms (`sigma2_theta`, `sigma2_delta`, etc), random effects (`theta[1]`, `theta[2]`, etc), and fitted values (`mu[1]`, `mu[2]`, etc) — not shown.

The bottom command, `message(SIRST_fit$messages)`, gives a MCMC convergence report(grey box). This report describes the convergence diagnostics (yellow boxes) across all parameters. The yellow arrows illustrate which columns of the `summary` component are reported in the `message(SIRST_fit$messages)` function. The first few lines provide details on the $\hat{R}$, followed by similar summaries for two measures of effective sample size (ESS). In the example, we would be confident that our MCMC algorithm has converged; there are no parameters with a $\hat{R} > 1.01$ or too low ESS. Please review Section 2.1 for details of these convergence diagnostics.

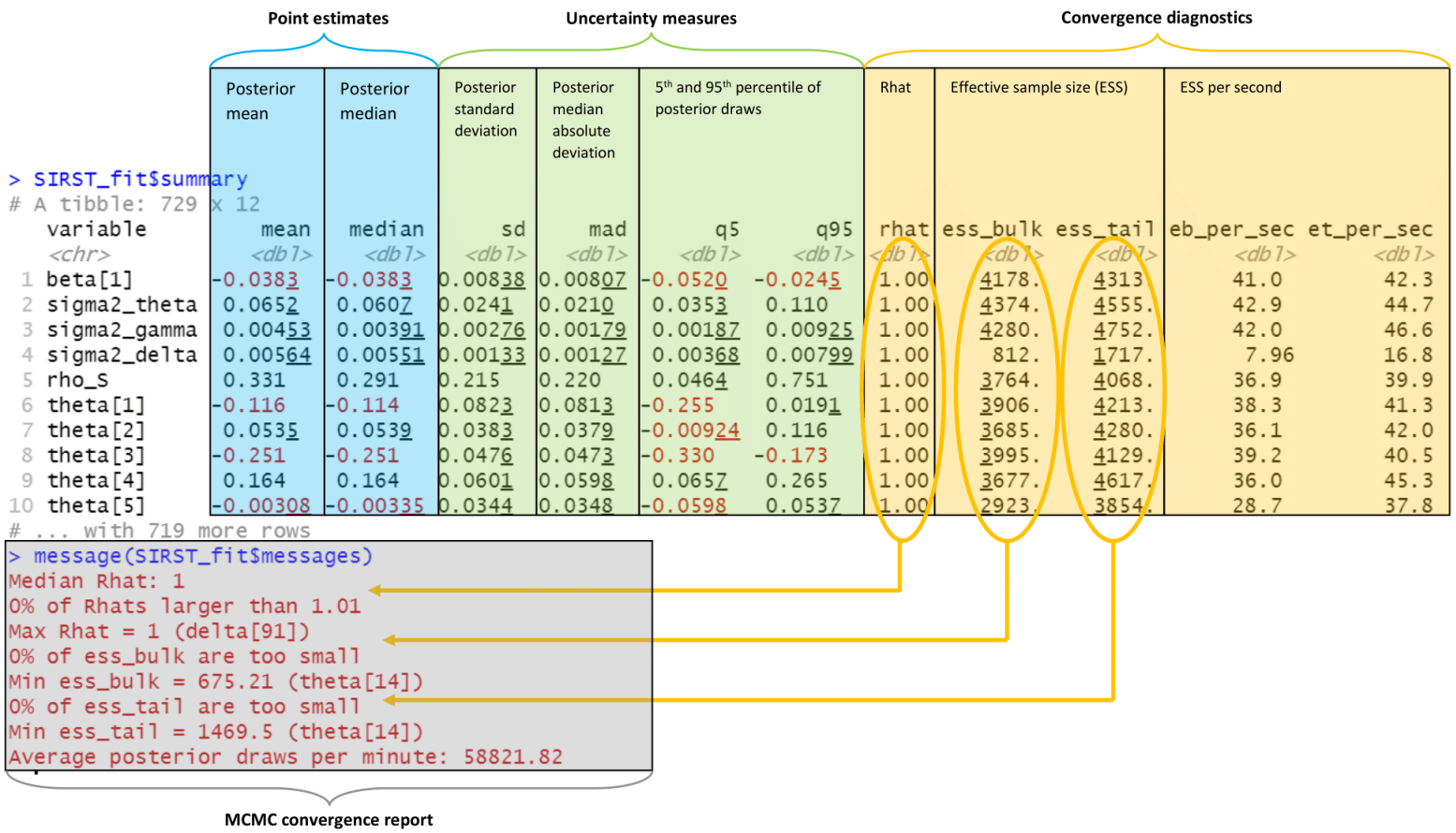



Figure 21: Example output after successfully fitting the SIR_ST model using the wrapper function shown in Code 3. Note that digits coloured red are below zero — a default option when using `tibble`.

## 8.10 Recommended Bayesian models

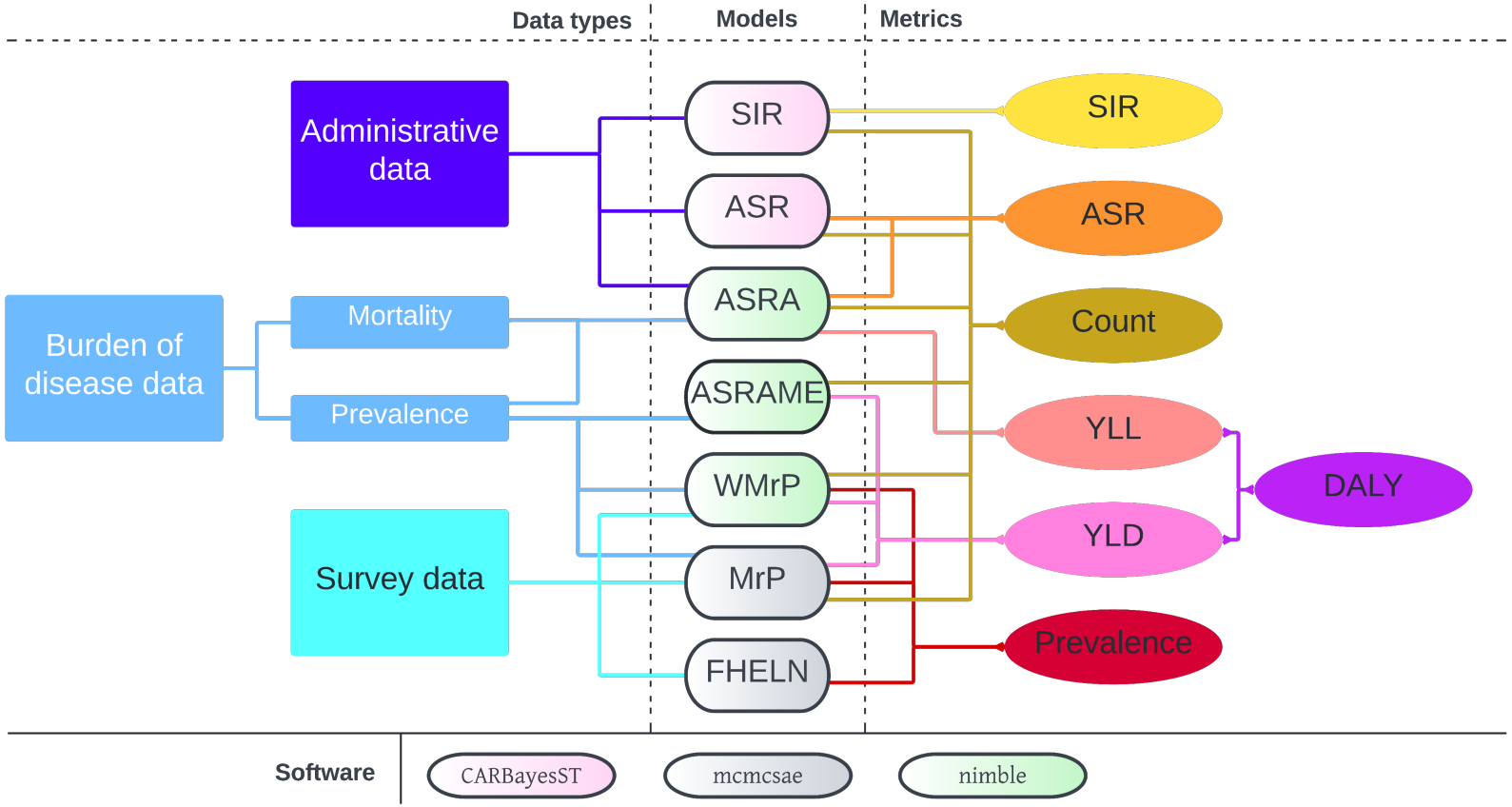



Figure 22: Schematic displaying the three data types and how these relate to the seven recommended models and core metrics.

# 9 References

Barker, L. E., Thompson, T. J., Kirtland, K. A., Boyle, J. P., Geiss, L. S., McCauley, M. M., & Albright, A. L. (2013). Bayesian Small Area Estimates of Diabetes Incidence by United States County, 2009. *Journal of Data Science*, *11*(1), 269–280.

Berkowitz, Z., Zhang, X. Y., Richards, T. B., Peipins, L., Henley, J., & Holt, J. (2016). Multilevel Small-Area Estimation of Multiple Cigarette Smoking Status Categories Using the 2012 Behavioral Risk Factor Surveillance System. *Cancer Epidemiology Biomarkers and Prevention*, *25*(10), 1402–1410.

Besag, J., York, J., & Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, *43*(1), 1–20.

Boonstra, H. J., & Baltissen, G. (2021). *mcmcsae: Markov Chain Monte Carlo Small Area Estimation*.

Chen, C., Wakefield, J., & Lumely, T. (2014). The use of sampling weights in Bayesian hierarchical models for small area estimation. *Spatial and Spatio-temporal Epidemiology*, *11*, 33–43.

Corpas-Burgos, F., García-Donato, G., & Martinez-Beneito, M. A. (2018). Some findings on zero-inflated and hurdle Poisson models for disease mapping. *Statistics in Medicine*, *37*(23), 3325–3337.

Cramb, S., Duncan, E., Baade, P., & Mengersen, K. L. (2020). A Comparison of Bayesian Spatial Models for Cancer Incidence at a Small Area Level: Theory and Performance. In K. L. Mengersen, P. Pudlo, & C. P. Robert (Eds.), *Case Studies in Applied Bayesian Data Science* (pp. 245–274). Springer International Publishing.

Das, S., van den Brakel, J. A., Boonstra, H. J., & Haslett, S. (2021). *Multilevel Time Series Modelling of Antenatal Care Coverage in Bangladesh at Disaggregated Administrative Levels*. Statistics Netherlands.

de Valpine, P., Turek, D., Paciorek, C. J., Anderson-Bergman, C., Lang, D. T., & Bodik, R. (2017). Programming With Models: Writing Statistical Algorithms for General Model Structures With NIMBLE. *Journal of Computational and Graphical Statistics*, *26*(2), 403–413.

Duncan, E. W., Cramb, S. M., Aitken, J. F., Mengersen, K. L., & Baade, P. D. (2019). Development of the Australian Cancer Atlas: Spatial modelling, visualisation, and reporting of estimates. *International Journal of Health Geographics*, *18*(1), 1–12.

Duncan, E. W., & Mengersen, K. L. (2020). Comparing Bayesian spatial models: Goodness-of-smoothing criteria for assessing under- and over-smoothing. *PLOS ONE*, *15*(5), e0233019.

Earnest, A., Morgan, G., Mengersen, K., Ryan, L., Summerhayes, R., & Beard, J. (2007). Evaluating the effect of neighbourhood weight matrices on smoothing properties of Conditional Autoregressive (CAR) models. *International Journal of Health Geographics*, *6*(1), 54.

Fay, R. E., & Herriot, R. A. (1979). Estimates of Income for Small Places: An Application of James-Stein Procedures to Census Data. *Journal of the American Statistical Association*, *74*(366), 269–277.

Gao, Y., Kennedy, L., Simpson, D., & Gelman, A. (2021). Improving multilevel regression and poststratification with structured priors. *Bayesian Analysis*, *16*(3), 719–744.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014a). *Bayesian Data Analysis: Third Edition*. CRC Press.

Gelman, A., Hwang, J., & Vehtari, A. (2014b). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, *24*(6), 997–1016.

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., & Modrák, M. (2020). Bayesian Workflow. *arXiv preprint arXiv:2011.01808*.

Ghitza, Y., & Gelman, A. (2013). Deep Interactions with MRP: Election Turnout and Voting Patterns Among Small Electoral Subgroups. *American Journal of Political Science*, *57*(3), 762–776.

Goldstein, H. (2011). *Multilevel statistical models*. John Wiley & Sons.

Gomez-Rubio, V., Best, N., Richardson, S., Li, G., & Clarke, P. (2008). *Bayesian Statistics Small Area Estimation* (Report). Office for National Statistics.

Haining, R., & Li, G. (2020). *Modelling Spatial and Spatial-Temporal Data - A Bayesian Approach*. Chapman; Hall/CRC.

Health Survey Unit, Epidemiology Branch. (2011). *The WA Health and Wellbeing Surveillance System (WAHWSS) Design and Methodology Technical Paper Series No 1* (tech. rep.). Health Survey Unit, Epidemiology Branch, Public and Aboriginal Health Division, Department of Health. Western Australia.

Jay, M., Oleson, J., Charlton, M., & Arab, A. (2021). A Bayesian approach for estimating age-adjusted rates for low-prevalence diseases over space and time. *Statistics in Medicine*, *40*, 2922–2938.

Knorr-Held, L. (2000). Bayesian modelling of inseparable space-time variation in disease risk. *Statistics in medicine*, *19*(17-18), 2555–2567.

Kolczynska, M., Bürkner, P.-C., Kennedy, L., & Vehtari, A. (2022). Modeling public opinion over time and space: Trust in state institutions in Europe, 1989-2019. *SocArXiv Papers*.

Lawson, A. B. (2020). NIMBLE for Bayesian Disease Mapping. *Spatial and Spatio-temporal Epidemiology*, *33*, 100323.

Lee, D. (2011). A comparison of conditional autoregressive models used in Bayesian disease mapping. *Spatial and spatio-temporal epidemiology*, *2*(2), 79–89.

Lee, D., Rushworth, A., Napier, G., & Pettersson, W. (2022). CARBayesST version 3.3: Spatio-Temporal Areal Unit Modelling in R with Conditional Autoregressive Priors.

Leroux, B. G., Lei, X., & Breslow, N. (2000). Estimation of Disease Rates in Small Areas: A new Mixed Model for Spatial Dependence. In M. E. Halloran & D. Berry (Eds.),

*Statistical Models in Epidemiology, the Environment, and Clinical Trials* (pp. 179–191). Springer New York.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS — a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325–337.

MacNab, Y. (2007). Mapping disability-adjusted life years: A Bayesian hierarchical model framework for burden of disease and injury assessment. *Statistics in Medicine*, *26*, 4746–4769.

McElreath, R. (2020). *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman; Hall/CRC.

Mercer, L., Wakefield, J., Chen, C., & Lumley, T. (2014). A comparison of spatial smoothing methods for small area estimation with sampling weights. *Spatial Statistics*, *8*(1), 69–85.

Neelon, B., Chang, H. H., Ling, Q., & Hastings, N. S. (2014). Spatiotemporal hurdle models for zero-inflated count data: Exploring trends in emergency department visits. *Statistical Methods in Medical Research*, *25*(6), 2558–2576.

Ornstein, J. T. (2020). Stacked regression and poststratification. *Political Analysis*, *28*(2), 293–301.

Park, D. K., Gelman, A., & Bafumi, J. (2004). Bayesian multilevel estimation with poststratification: State-level estimates from national polls. *Political Analysis*, *12*(4), 375–385.

Plummer, M. (2003). JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling. *3rd International Workshop on Distributed Statistical Computing*.

Rao, J. N. K., & Molina, I. (2015). *Small Area Estimation* (2nd). Wiley Series in Survey Methodology.

Riebler, A., & Held, L. (2017). Projecting the future burden of cancer: Bayesian age–period cohort analysis with integrated nested Laplace approximations. *Biometrical*, *59*(3), 531–549.

Riebler, A., Sørbye, S. H., Simpson, D., & Rue, H. (2016). An intuitive Bayesian spatial model for disease mapping that accounts for scaling.

Säfken, B., Rügamer, D., Kneib, T., & Greven, S. (2018). Conditional model selection in mixed-effects models with caic4. *arXiv preprint arXiv:1803.05664*.

Savitsky, T. D., & Toth, D. (2016). Bayesian estimation under informative sampling. *Electronic Journal of Statistics*, *10*(1), 1677–1708.

Si, Y., Trangucci, R., Gabry, J., & Gelman, A. (2020). Bayesian hierarchical weighting adjustment and survey inference. *Survey Methodology, Statistics Canada*, *46*(2), 181–214.

Stan Development Team. (2022). Stan, https://mc–stan.org.

Tzavidis, N., Zhang, L. C., Luna, A., Schmid, T., & Rojas-Perilla, N. (2018). From start to finish: A framework for the production of small area official statistics. *Journal of the Royal Statistical Society*, *181*(4), 927–979.

Ugarte, M. D., Adin, A., Goicoa, T., & Militino, A. F. (2014). On fitting spatio-temporal disease mapping models using approximate Bayesian inference. *Statistical Methods in Medical Research*, *23*(6), 507–530.

Urdangarin, A., Goicoa, T., & Dolores Ugarte, M. (2022). Space-time interactions in Bayesian disease mapping with recent tools: Making things easier for practitioners. *Statistical Methods in Medical Research*, *31*(6), 1085–1103.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*(5), 1413–1432.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-Normalization, Folding, and Localization: An Improved R^ for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, *16*(2).

Wolter, K. M. (2007). *Introduction to Variance Estimation*. Springer.

You, Y., & Rao, J. N. K. (2000). Hierarchical Bayes estimation of small area means using multi-level models. *Survey Methodology*, *26*(2), 173–181.