Automated and Early Detection of Disease Outbreaks

Kasper Schou Telkamp

Section for Dynamical Systems

Outline

- Data
  - Focusing on STEC

- Model formulas
  - Agegroup
  - Agegroup and seasonality

- Hierarchical Poisson Normal model
  - Implementation
  - Results
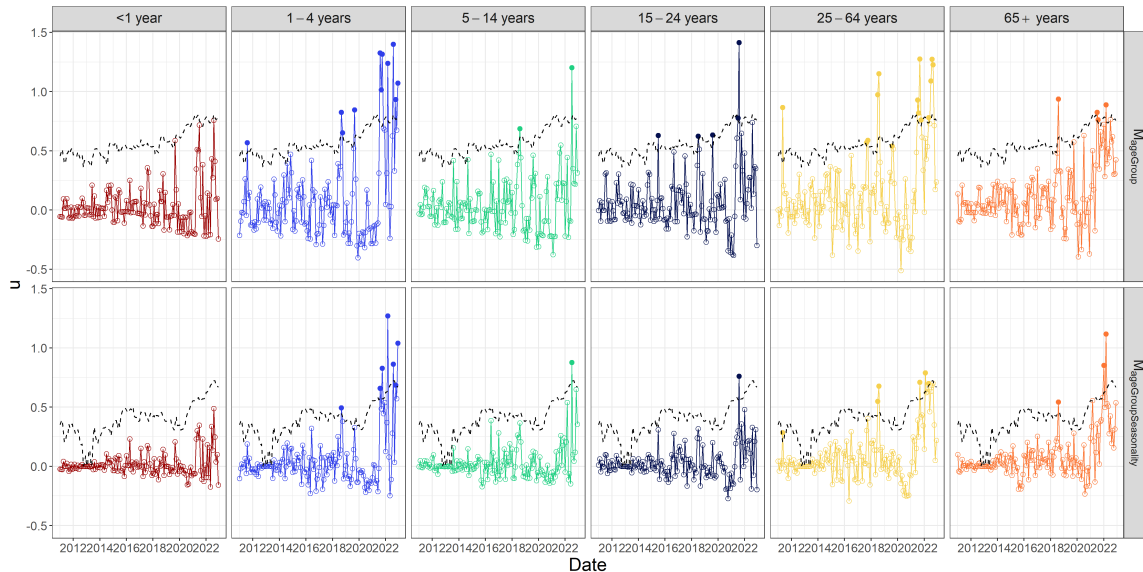
- Hierarchical Poisson Gamma model
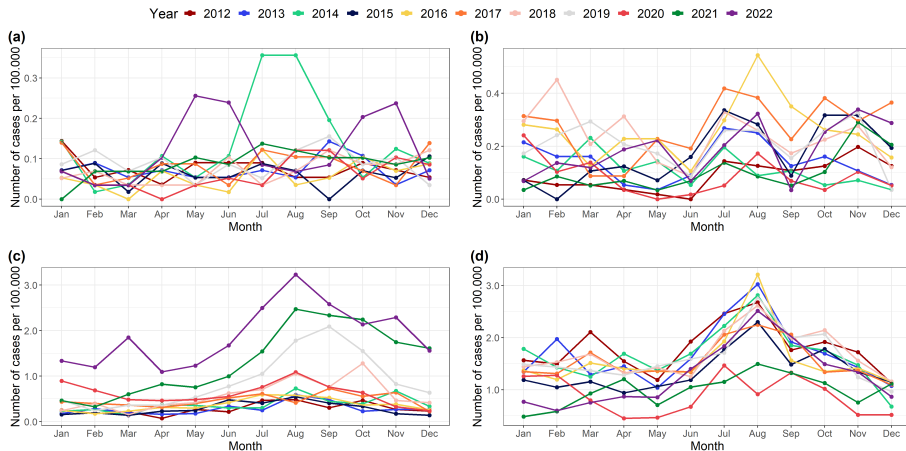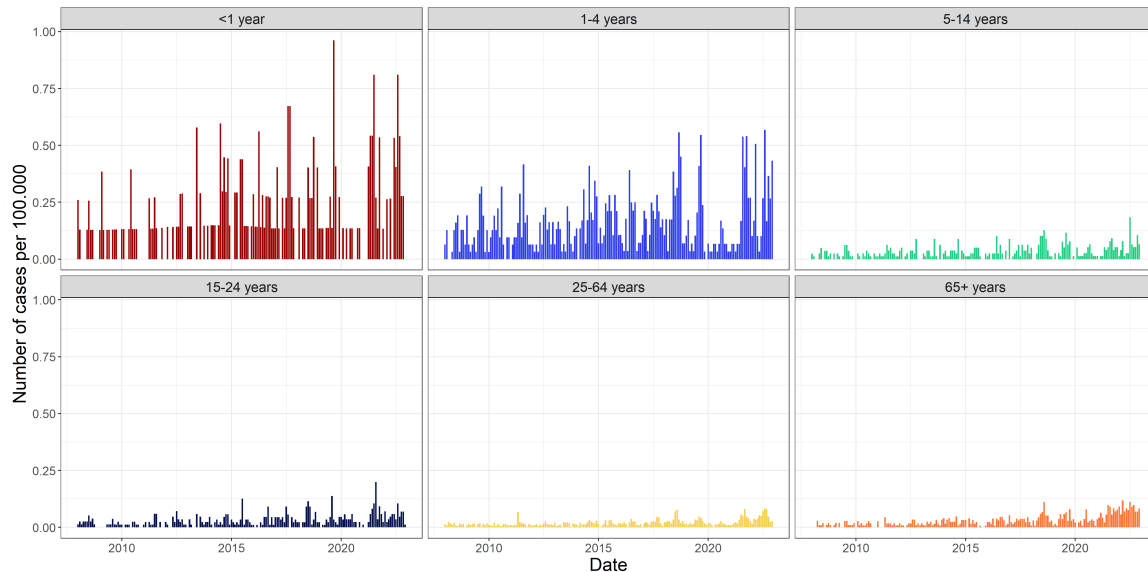 - Implementation
  - Results

Figure: Epidemic curve showing the incidence per 100,000 in Denmark, 2012-2022, for the subset of diseases considered in this master thesis. (a) Listeriosis, (b) Shigellosis, (c) STEC, and (d) Salmonellosis.

Focusing on STEC

$$\log(\lambda_{it}) = \boldsymbol{X}_t^T \boldsymbol{\beta} + \log(x_{it}), \quad i = 1, \ldots, k, \quad t = 1, \ldots, n \tag{1}$$

$$\log(\lambda_{it}) = ageGroup_{it}\beta_i + \log(x_{it}) \tag{2}$$

$$\log(\lambda_{it}) = ageGroup_{it}\beta_i + \sin\left(\frac{\pi \cdot monthInYear_t}{6}\right)\beta_{\sin,t} + \cos\left(\frac{\pi \cdot monthInYear_t}{6}\right)\beta_{\cos,t} + \log(x_{it})$$
$$(3)$$

$$\boldsymbol{Y}|\boldsymbol{u} \sim \text{Pois}\left(\boldsymbol{\lambda}\exp(\boldsymbol{u})\right) \tag{4a}$$

$$\boldsymbol{u} \sim \text{N}(\boldsymbol{0}, \sigma^2) \tag{4b}$$

Implementation

```cpp
#include <TMB.hpp>                    // Links in the TMB libraries

template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_VECTOR(y);                                // Data vector transmitted from R
  DATA_VECTOR(x);                                // Data vector transmitted from R
  DATA_MATRIX(X);                                // Design matrix transmitted from R

  PARAMETER_VECTOR(u);                           // Random effects

  // Parameters
  PARAMETER_VECTOR(beta);         // Parameter value transmitted from R
  PARAMETER(log_sigma_u);                 // Parameter value transmitted from R

  vector<Type> lambda  = exp(X*beta-log(x)+u);
  Type sigma_u = exp(log_sigma_u);

  int nobs = y.size();
  Type mean_ran = Type(0);

  Type f = 0;                           // Declare the "objective function"
  for(int t=0; t < nobs; t++){
    f -= dnorm(u[t],mean_ran,sigma_u,true);
    f -= dpois(y[t],lambda[t],true);
  }

  return f;
}
```
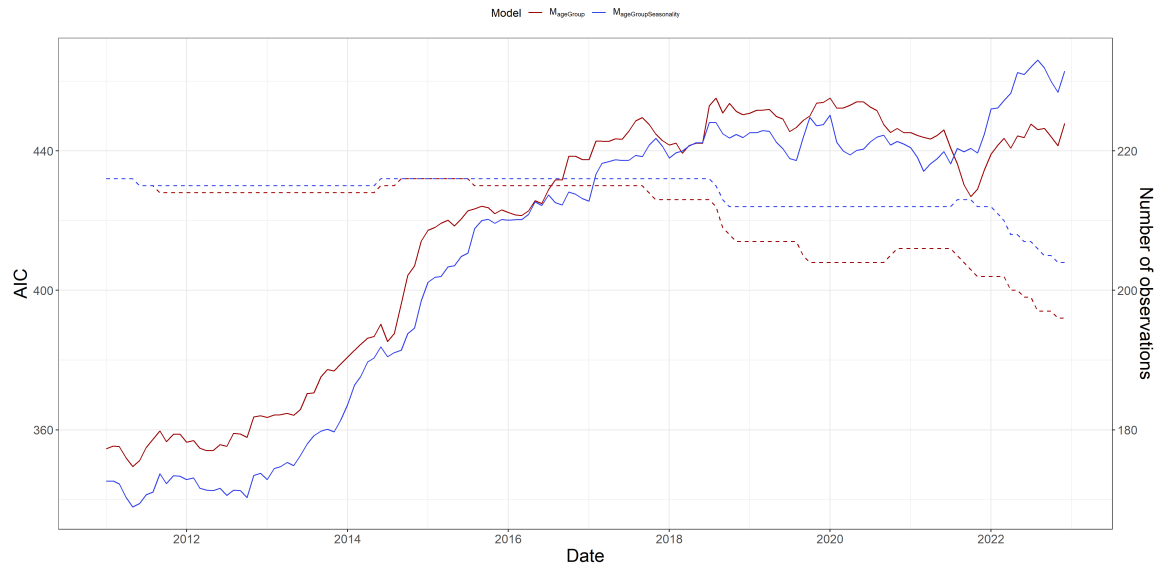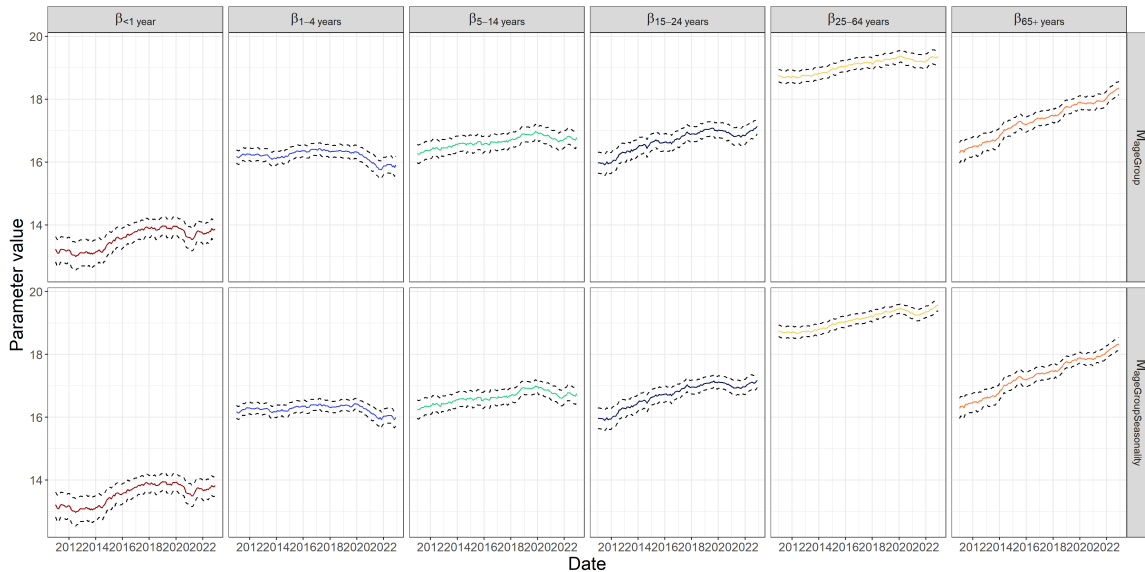
Model performance

Agegroup parameters
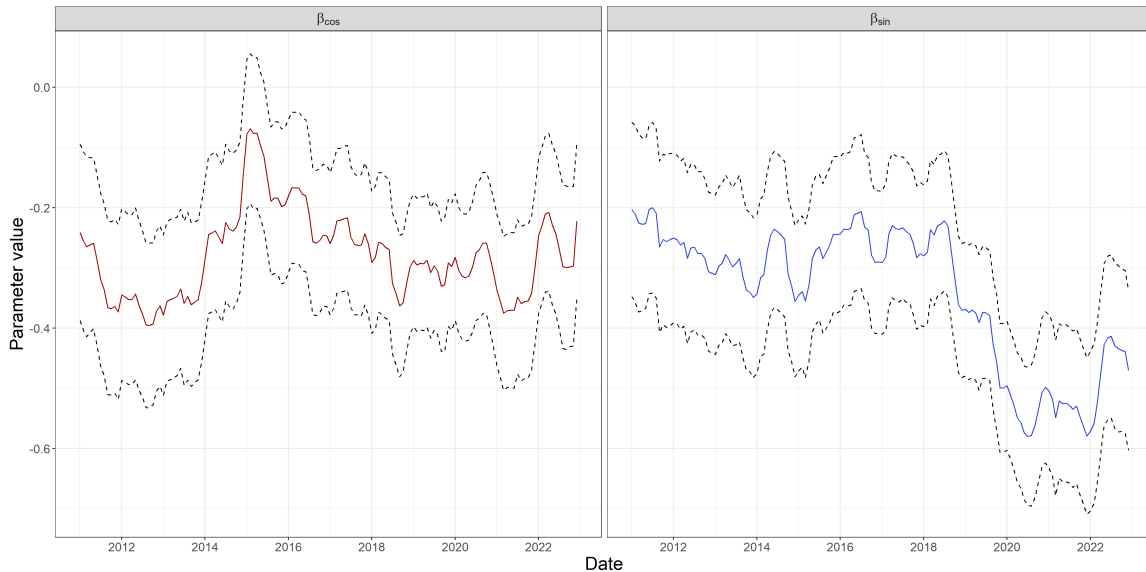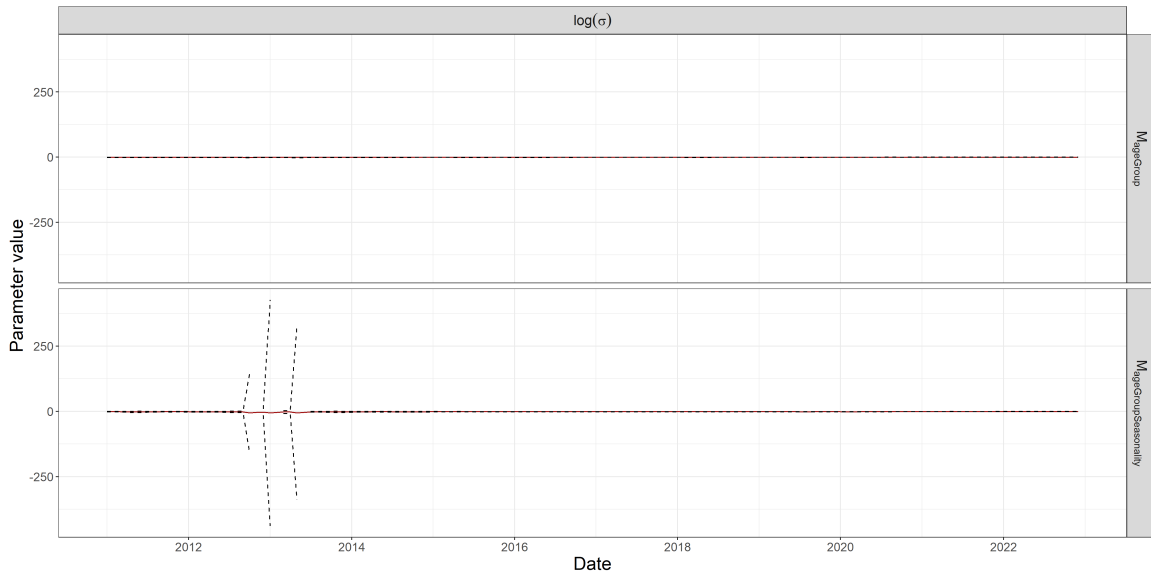
Seasonality parameters

Variance parameter

Outbreak detection

$$\boldsymbol{Y}|\boldsymbol{u} \sim \text{Pois}(\boldsymbol{\lambda}\boldsymbol{u}) \tag{5a}$$

$$\boldsymbol{u} \sim \text{G}(\boldsymbol{1}/\phi, \phi) \tag{5b}$$

Implementation

```cpp
#include <TMB.hpp>                  // Links in the TMB libraries

template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_VECTOR(y);                              // Data vector transmitted from R
  DATA_VECTOR(x);                    // Data vector transmitted from R
  DATA_MATRIX(X);                    // Design matrix transmitted from R

  // Parameters
  PARAMETER_VECTOR(beta);          // Parameter value transmitted from R
  PARAMETER(log_phi_u);              // Parameter value transmitted from R

  vector<Type> lambda  = exp(X*beta-log(x)); // Construct the model parameters
  Type phi_u = exp(log_phi_u); // ... and the model parameters

  Type r = 1/phi_u; // Construct the size
  vector<Type> p = 1/(lambda*phi_u+1); // ... and the probability parameter

  Type f = -sum(dnbinom(y, r, p,true)); // Calculate the "objective function"

  return f;
}
```
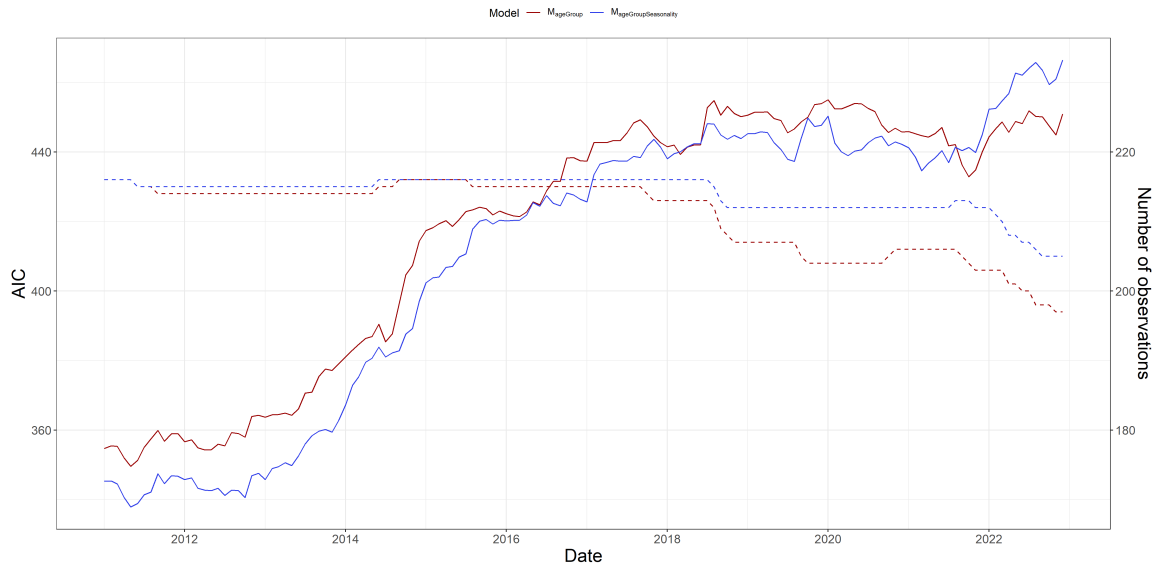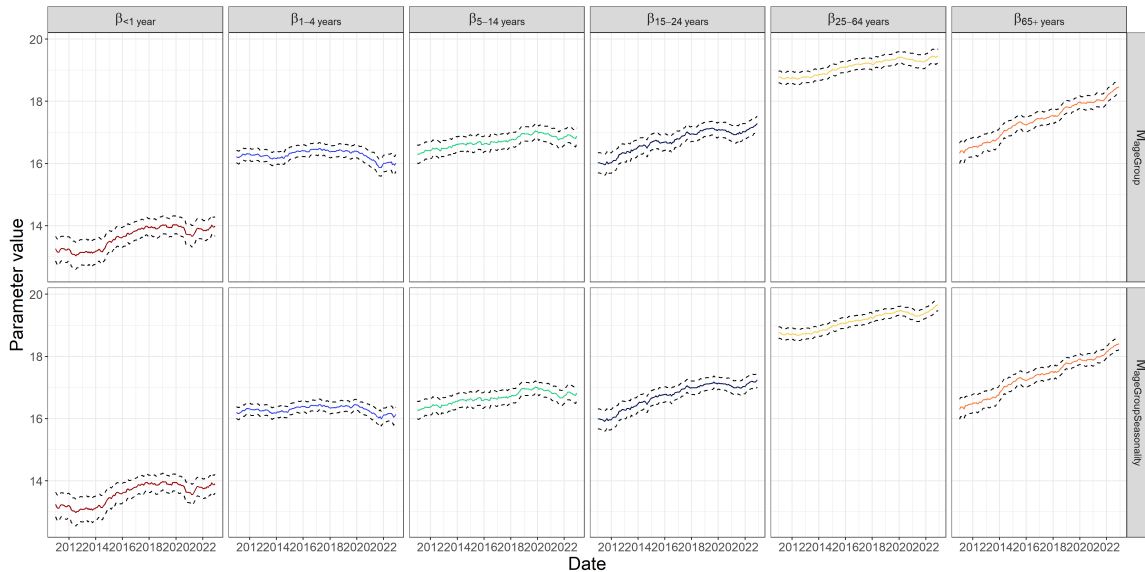
Model performance

Agegroup parameters

Seasonality parameters

Variance parameter

Outbreak detection