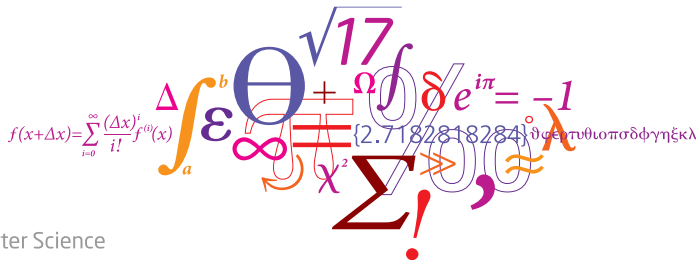


# Automated and Early Detection of Disease Outbreaks

Kasper Schou Telkamp

Section for Dynamical Systems



DTU Compute

Department of Applied Mathematics and Computer Science

# Outline

- Data
  - Focusing on STEC
- Model formulas
  - Agegroup
  - Agegroup and seasonality
- Hierarchical Poisson Normal model
  - Implementation
  - Results
- Hierarchical Poisson Gamma model
  - Implementation
  - Results

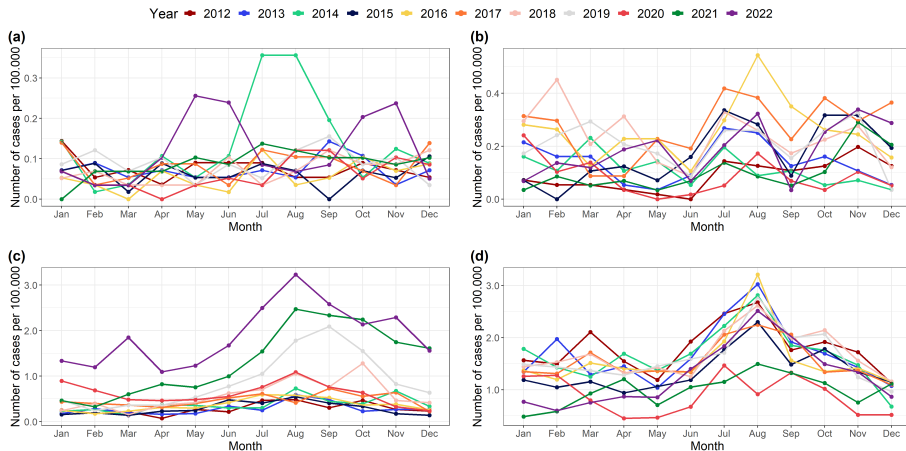
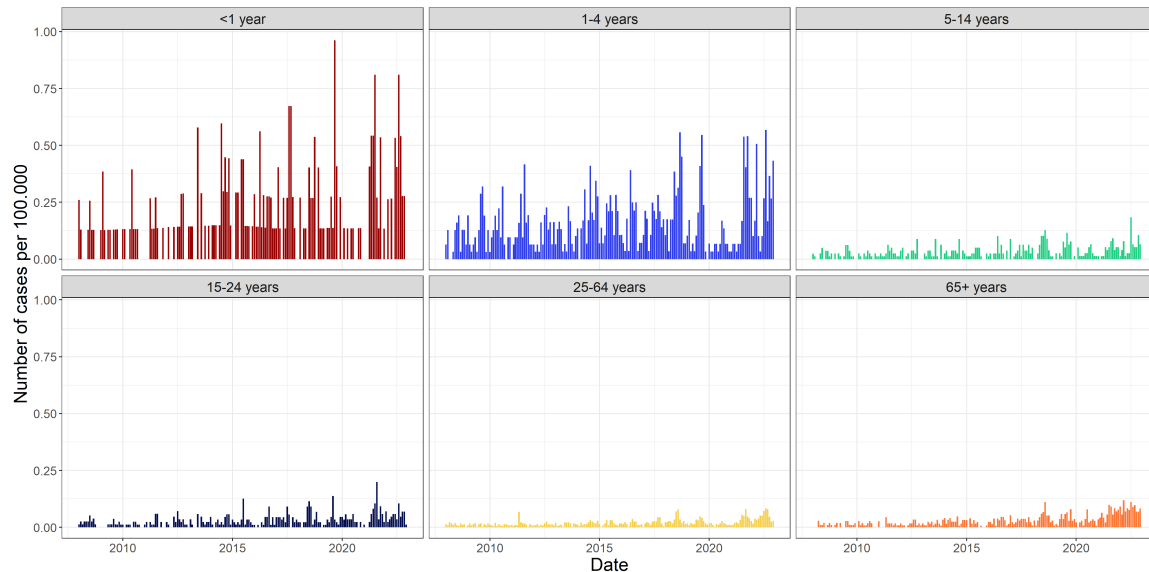


Figure: Epidemic curve showing the incidence per 100,000 in Denmark, 2012-2022, for the subset of diseases considered in this master thesis. (a) Listeriosis, (b) Shigellosis, (c) STEC, and (d) Salmonellosis.

## Focusing on STEC



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



$$\log(\lambda_{it}) = \text{ageGroup}_{it}\beta_i + \log(x_{it}) \quad (2)$$



$$\log(\lambda_{it}) = \text{ageGroup}_{it}\beta_i + \sin\left(\frac{\pi \cdot \text{monthInYear}_t}{6}\right)\beta_{\sin,t} + \cos\left(\frac{\pi \cdot \text{monthInYear}_t}{6}\right)\beta_{\cos,t} + \log(x_{it}) \quad (3)$$



## Hierarchical Poisson Normal model

$$\mathbf{Y}|\mathbf{u} \sim \text{Pois}(\boldsymbol{\lambda} \exp(\mathbf{u})) \quad (4a)$$

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





# Hierarchical Poisson Normal model

## Implementation

```
#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);

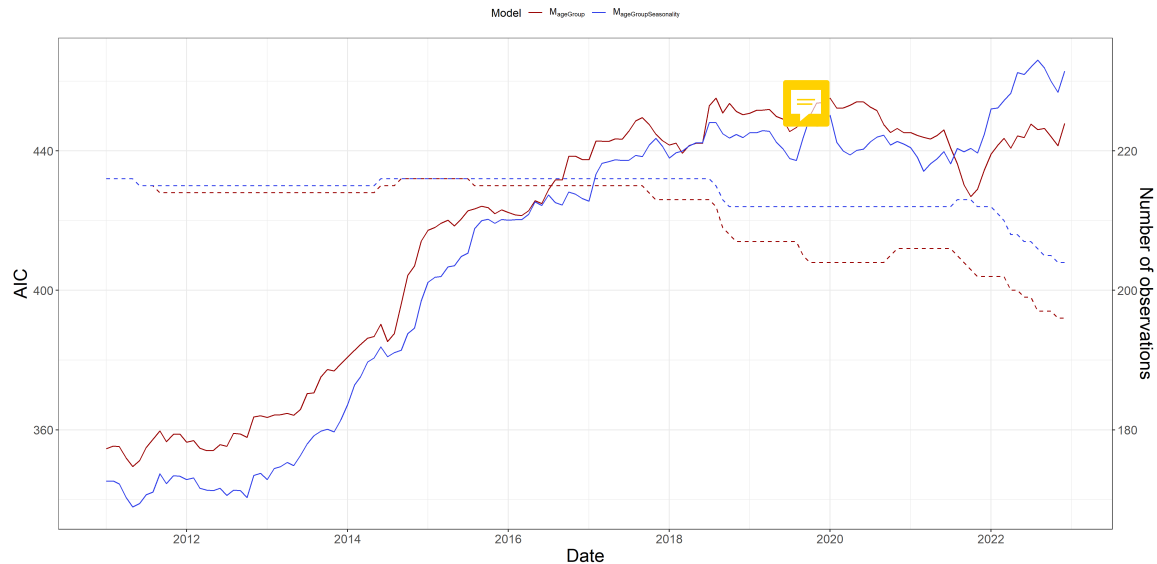
    int nobs = y.size();
    Type mean_ran = Type(0);

    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);
    }

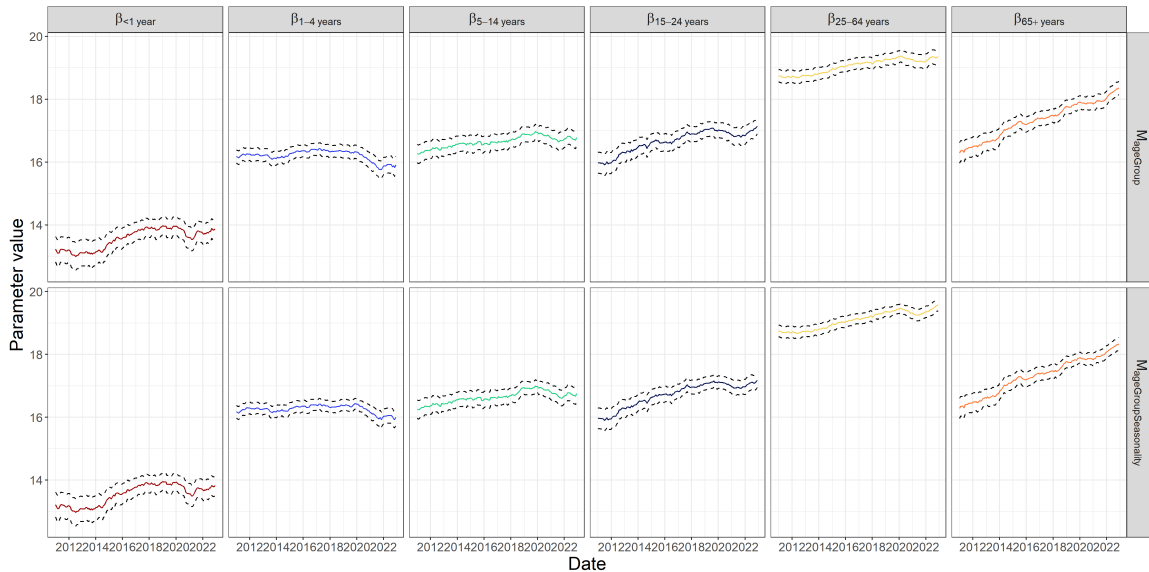
    return f;
}
```

# Hierarchical Poisson Normal model

## Model performance

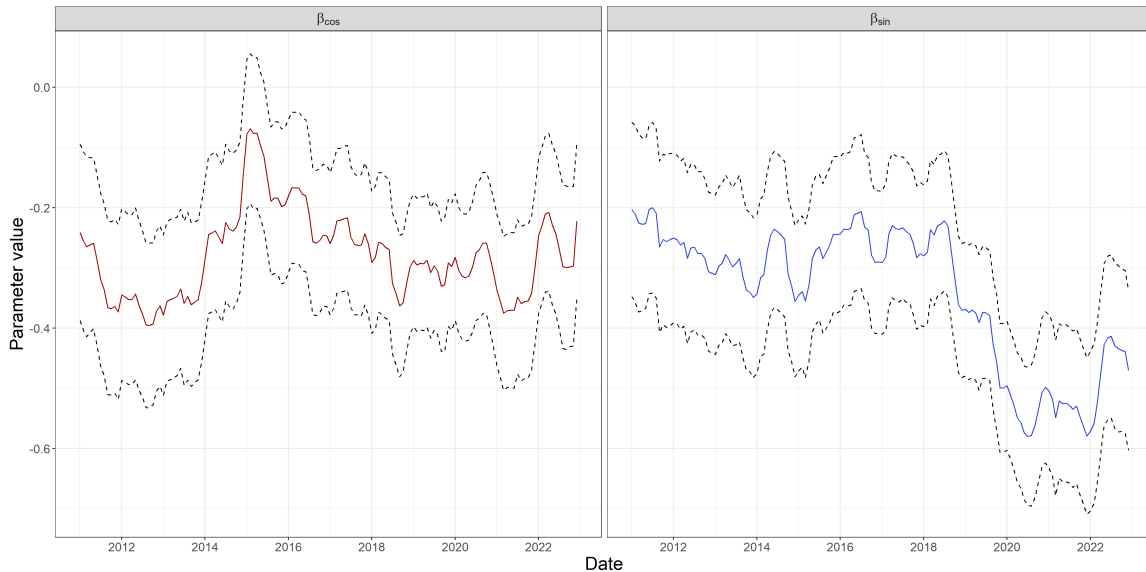


## Agegroup parameters

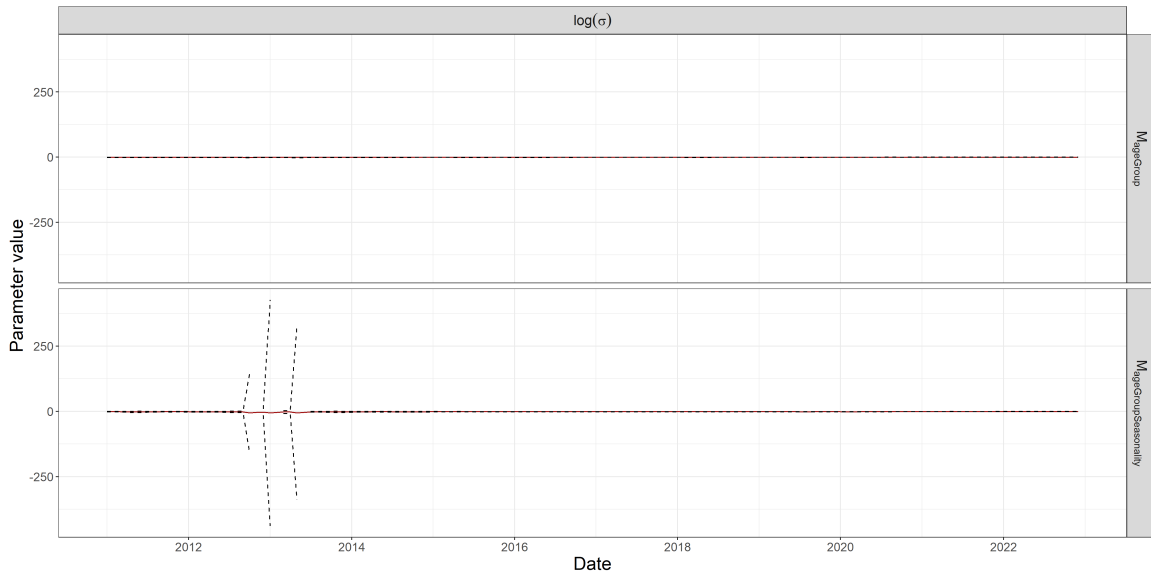


# Hierarchical Poisson Normal model

## Seasonality parameters

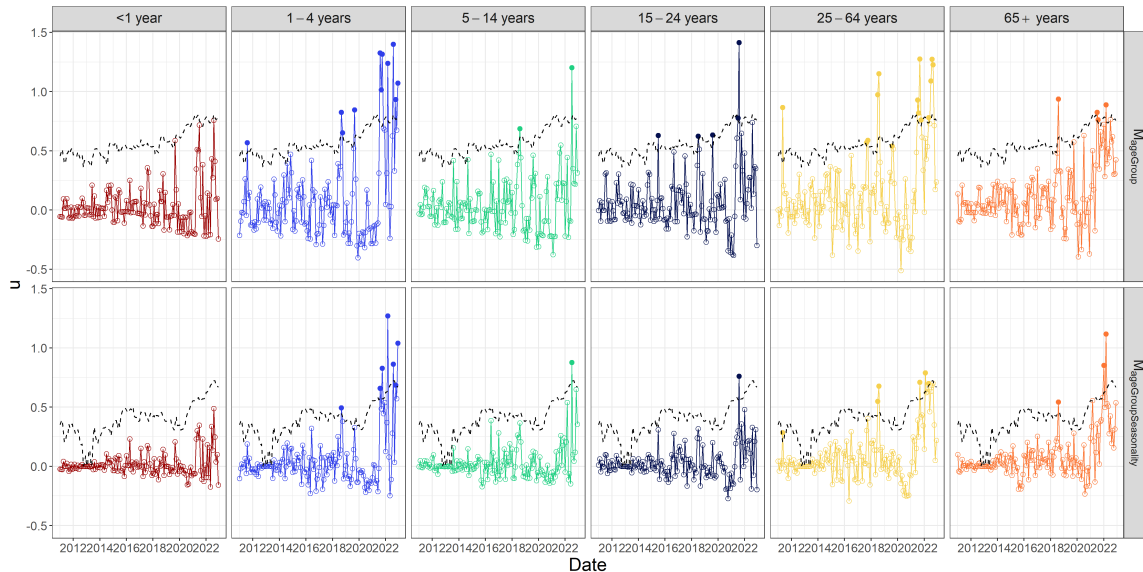


## Variance parameter



# Hierarchical Poisson Normal model

## Outbreak detection



## Hierarchical Poisson Gamma model

$$\mathbf{Y}|\mathbf{u} \sim \text{Pois}(\lambda\mathbf{u}) \quad (5a)$$

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

# Hierarchical Poisson Gamma model

## Implementation

```
#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

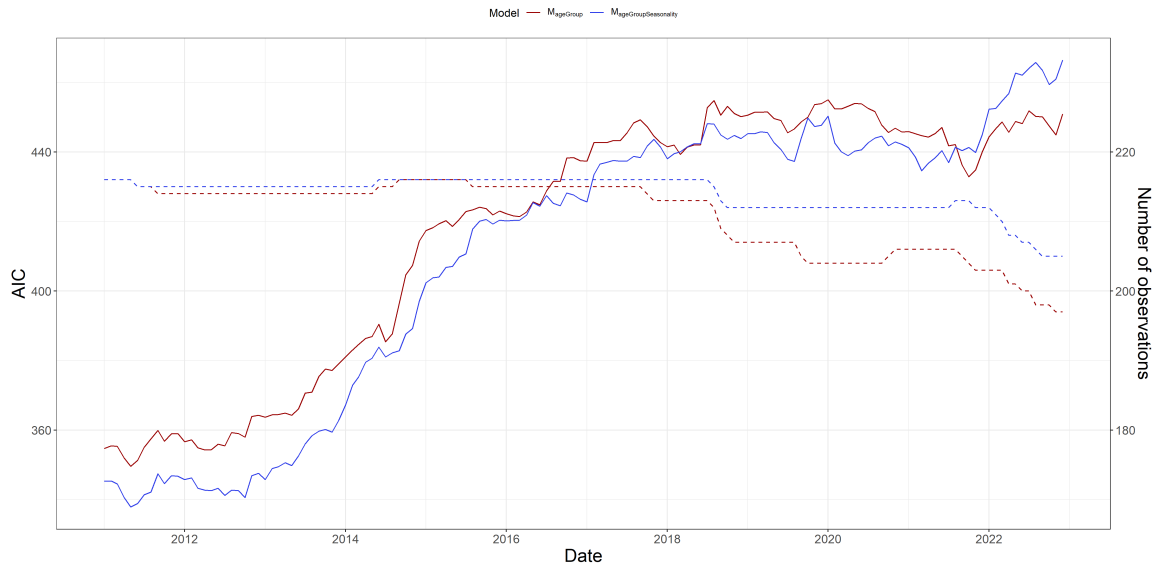
    Type f = -sum(dnbinom(y, r, p,true)); // Calculate the "objective function"

    return f;
}
```



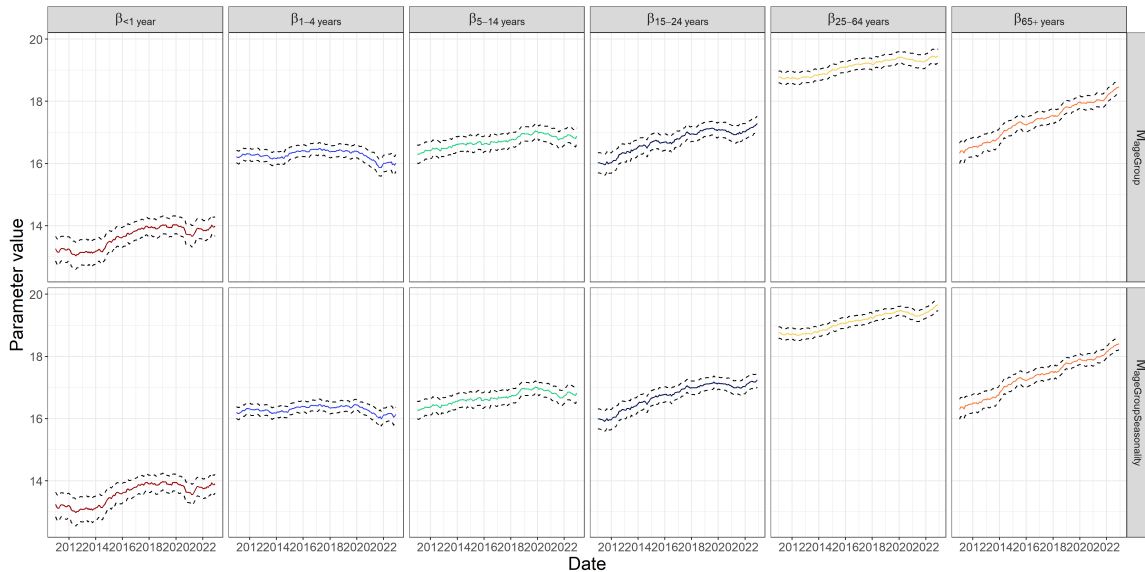
# Hierarchical Poisson Gamma model

## Model performance



# Hierarchical Poisson Gamma model

## Agegroup parameters

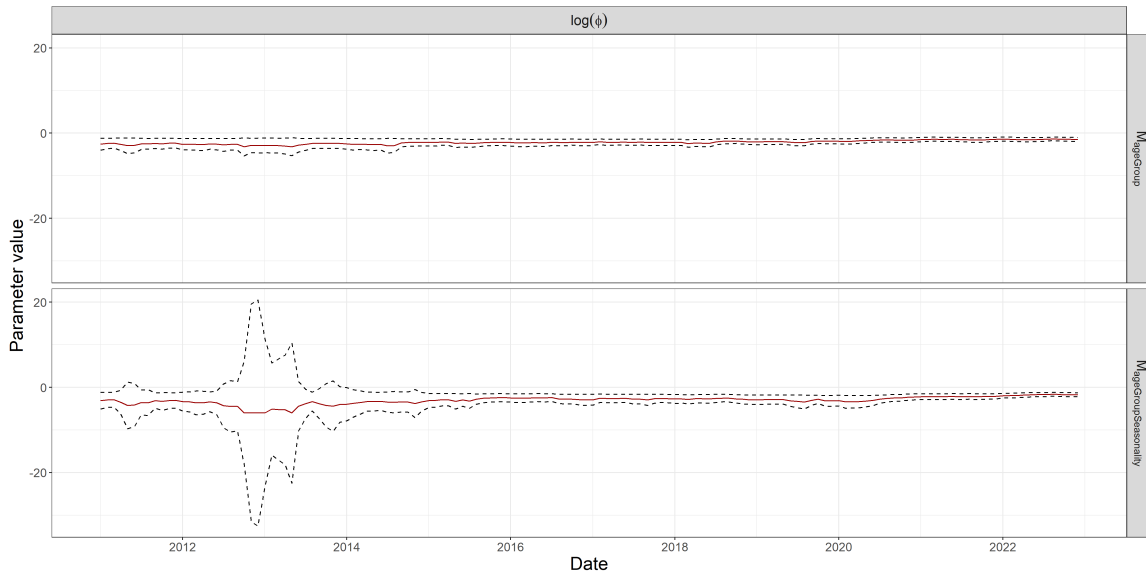


# Hierarchical Poisson Gamma model

## Seasonality parameters



# Variance parameter



# Hierarchical Poisson Gamma model

## Outbreak detection

