

[illegible]

# Outline

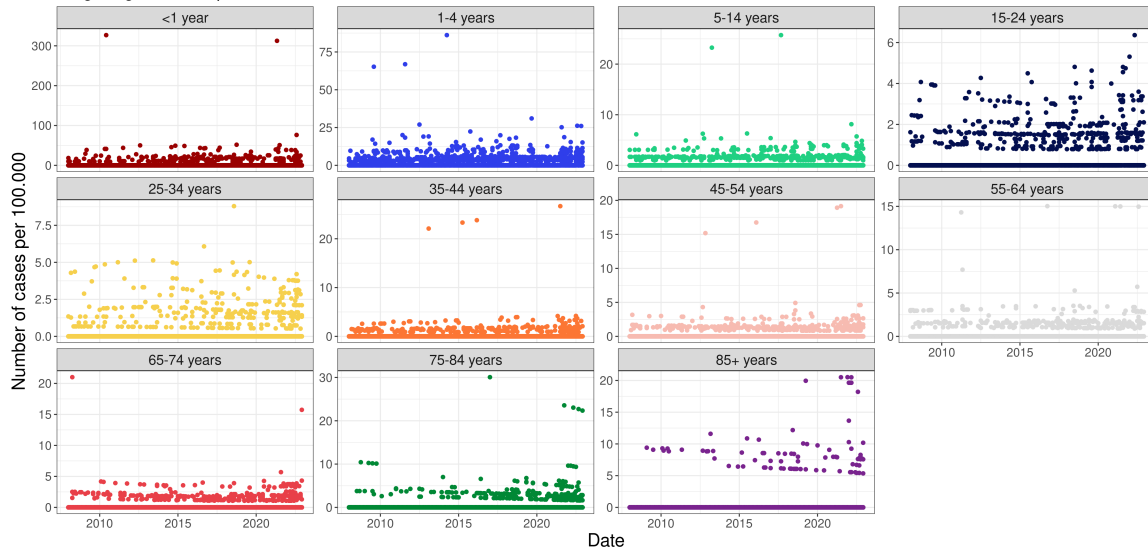
- Data exploration
  - VTEC / STEC
- Hierarchical Poisson Normal model
  - Formulation
  - Implementation
- Hierarchical Poisson Gamma model
  - Formulation

Date	ageGroup	y
2008-01-01	<1 year	2
2008-01-01	1-4 years	2
2008-01-01	5-14 years	2
2008-01-01	15-24 years	1
...	...	...
2022-12-01	55-64 years	2
2022-12-01	65-74 years	5
2022-12-01	75-84 years	4
2022-12-01	85+ years	3

# Data exploration

## VTEC / STEC

Shiga- og verotoxin producerende E. coli.



## Hierarchical Poisson Normal model

### Formulation



$$Y_i | u_i \sim \text{Pois}(\lambda_i \exp(u_i)) \quad (1a)$$

$$u_i \sim N(0, \sigma^2) \quad (1b)$$

## Implementation - Objective function in C++

```
#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_FACTOR(ageGroup);    // Data factor transmitted from R

    PARAMETER_VECTOR(u);       // Random effects

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

    Type sigma_u = exp(log_sigma_u);

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

    int j;

    Type f = 0;                // Declare the "objective function" (neg. log. likelihood)
    for(int i=0; i < nobs; i++){
        f -= dnorm(u[i],mean_ran,sigma_u,true);
        j = ageGroup[i];
        f -= dpois(y[i],lambda[j]*exp(u[i]),true);
    }

    return f;
}
```

## Hierarchical Poisson Normal model

# Implementation - Call from R

```
# Import libraries
library(readr)
library(dplyr)
library(TMB)

# Import the data
dat <- read_rds(file = "../data/processed/dat.rds")

# Only consider some of the data
y <- dat %>%
  filter(caseDef == "Shiga- og veratoxin producerende E. coli.") %>%
  group_by(Date, ageGroup) %>%
  summarize(y = sum(cases))

compile(file = "PoissonLognormal.cpp") # Compile the C++ file
dyn.load(dynlib("PoissonLognormal")) # Dynamically link the C++ code

# Function and derivative
PoisLN <- MakeADFun(
  data = list(y = y$y, ageGroup = y$ageGroup),
  parameters = list(u = rep(1, length(y$y)),
                    lambda = rep(1, nlevels(y$ageGroup)),
                    log_sigma_u = log(1)),
  random = "u",
  DLL = "PoissonLognormal"
)

opt <- nlminb(start = PoisLN$par, PoisLN$fn, PoisLN$gr, lower = c(0.01, 0.01))
```

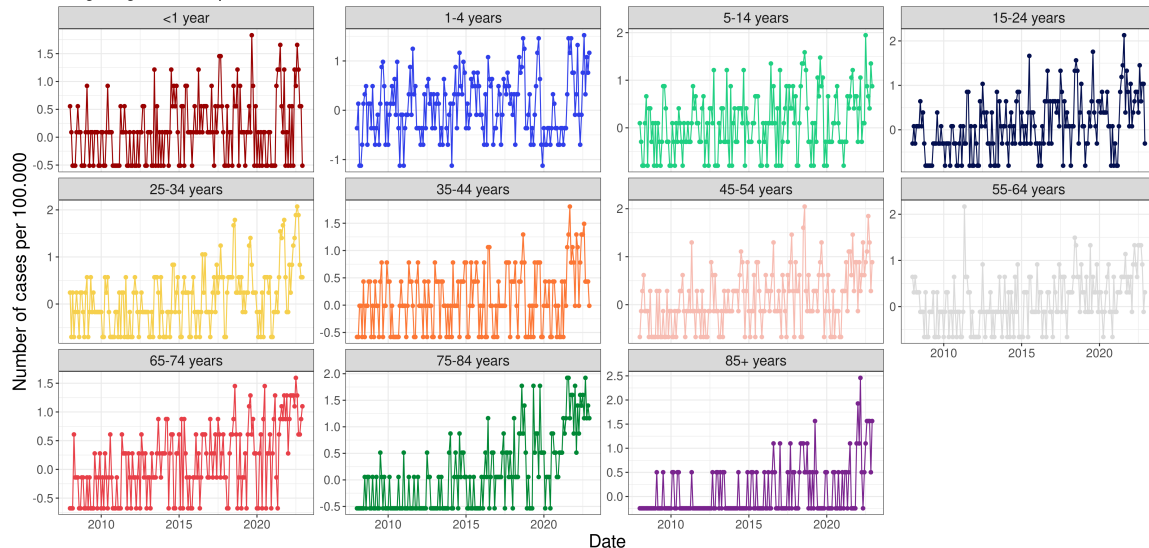
## Results

Parameter	Estimate	Std. Error
$\lambda_{<1year}$	0.83	0.09
$\lambda_{1-4years}$	3.39	0.29
$\lambda_{5-14years}$	1.73	0.16
$\lambda_{15-24years}$	1.78	0.17
$\lambda_{25-34years}$	1.38	0.14
$\lambda_{35-44years}$	1.02	0.11
$\lambda_{45-54years}$	1.29	0.13
$\lambda_{55-64years}$	1.25	0.12
$\lambda_{65-74years}$	1.31	0.13
$\lambda_{75-84years}$	0.90	0.10
$\lambda_{85+years}$	0.31	0.04
$\log(\sigma_u)$	0.01	0.03



## Results

Shiga- og veratoxin producerende *E. coli*.



## Hierarchical Poisson Gamma model

### Formulation



$$Y_i | u_i \sim \text{Pois}(\lambda_i u_i) \quad (2a)$$

$$u_i \sim G(1, \phi) \quad (2b)$$

## Hierarchical Poisson Gamma model

### Probability function for $Y$



$$\begin{aligned} P[Y = y] &= g_Y(y; \lambda, \phi) \\ &= 1234 \end{aligned} \tag{3}$$

The probability function for the conditional distribution of  $Y$  for given  $u$

$$f_{Y|u}(y; \lambda, u) = \frac{(\lambda u)^y}{y!} \exp(-\lambda u) \quad (4)$$

and the probability density function for the distribution of  $u$  is

$$f_u(u; \phi) = \frac{1}{\phi} \exp(-u/\phi) \quad (5)$$

Given (4) and (5), the probability function for the marginal distribution of  $Y$  is determined from

$$\begin{aligned} g_Y(y; \lambda, \phi) &= \int_{u=0}^{\infty} f_{Y|u}(y; \lambda, u) f_u(u; \phi) du \\ &= \int_{u=0}^{\infty} \frac{(\lambda u)^y}{y!} \exp(-\lambda u) \frac{1}{\phi} \exp(-u/\phi) du \\ &= \frac{1}{y! \phi} \int_{u=0}^{\infty} (\lambda u)^y \exp(-u(\lambda + 1/\phi)) du \end{aligned} \tag{6}$$