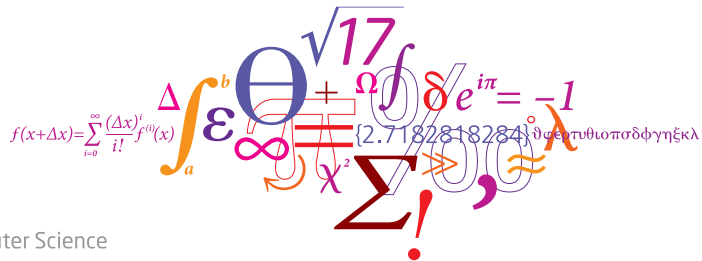
Models

Kasper Schou Telkamp

Section for Dynamical Systems

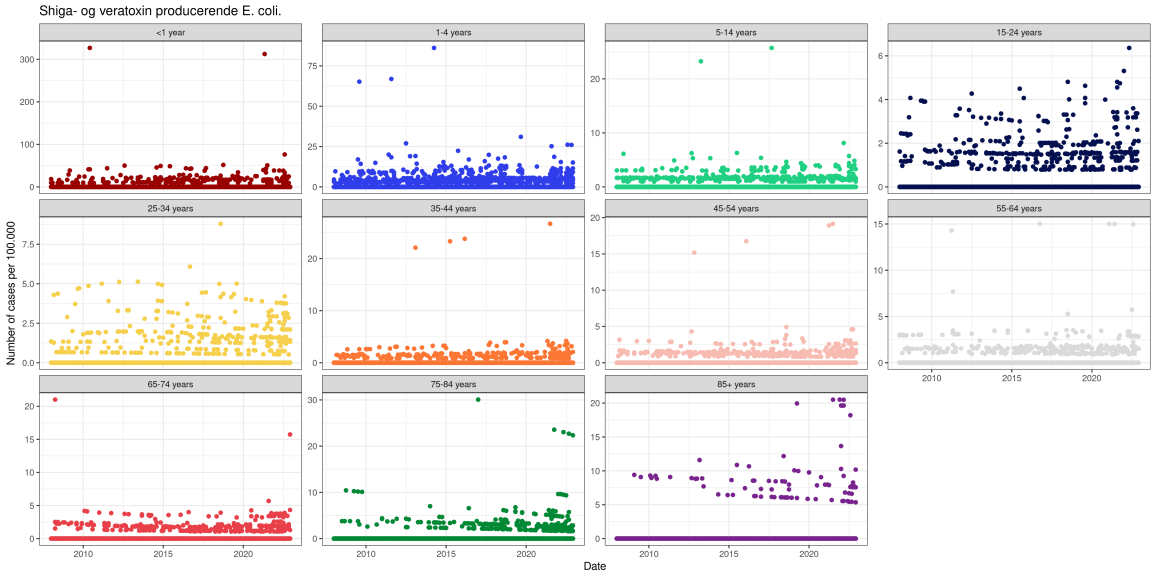


DTU Compute
Department of Applied Mathematics and Computer Science

Outline

- Data exploration
  - VTEC / STEC
- Model formulation
  - Poisson-Lognormal
  - Poisson-Gamma

VTEC / STEC


Shiga- og veratoxin producerende E. coli.
<1 year
1-4 years
5-14 years
15-24 years
25-34 years
35-44 years
45-54 years
55-64 years
65-74 years
75-84 years
85+ years
Number of cases per 100,000
Date
2010
2015
2020

Poisson-Lognormal

$$Y_i \sim \text{Pois}(\lambda \exp(u_i)) \tag{1a}$$

$$u_i \sim \text{N}(0, \sigma^2) \tag{1b}$$

Implementation - Test

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;
}
```

Implementation - Call from R

```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
f <- 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"
)
```

Poisson-Gamma

$$Y_i \sim \text{Pois}(\lambda u_i) \tag{2a}$$
$$u_i \sim \text{G}(1, \phi) \tag{2b}$$