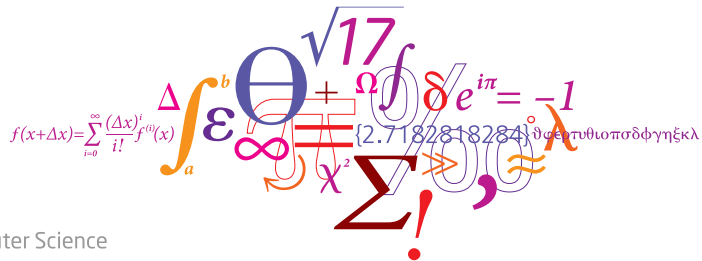
Models

Kasper Schou Telkamp

Section for Dynamical Systems



DTU Compute
Department of Applied Mathematics and Computer Science

Outline

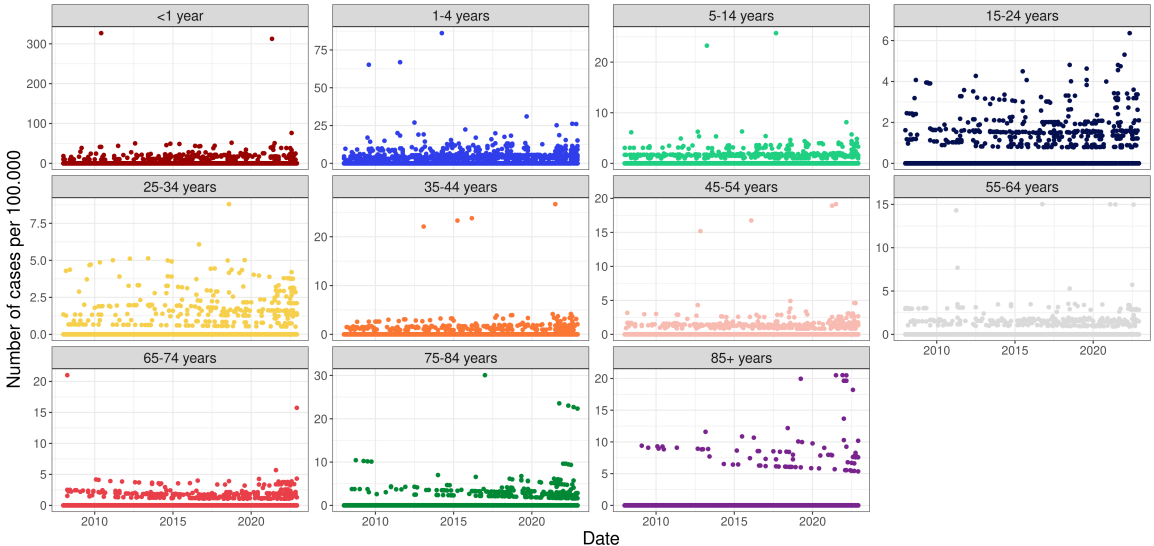
- Data exploration
  - VTEC / STEC
- Model formulation
  - Poisson-Lognormal
  - Poisson-Gamma

VTEC / STEC

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

VTEC / STEC

Shiga- og veratoxin producerende E. coli.



Poisson-Lognormal

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

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

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;
}
```

Implementation - Call from R

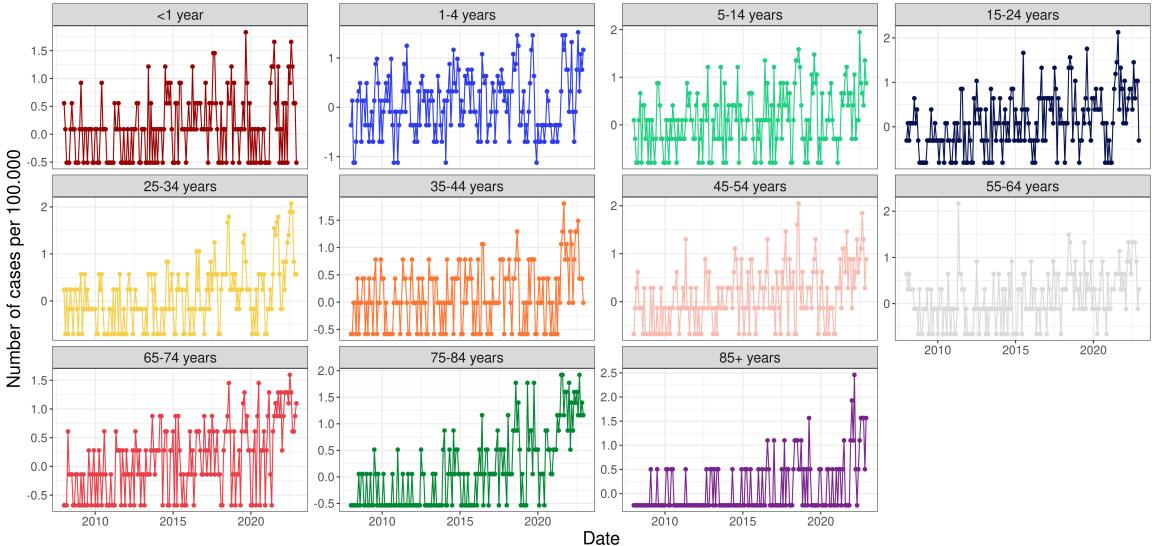
```
# Import libraries
library(readr)
library(dplyr)
library(TMB)

# 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.



Poisson-Gamma

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