Block 2.2 & 2.4: Inference, uncertainty, and stochastic models

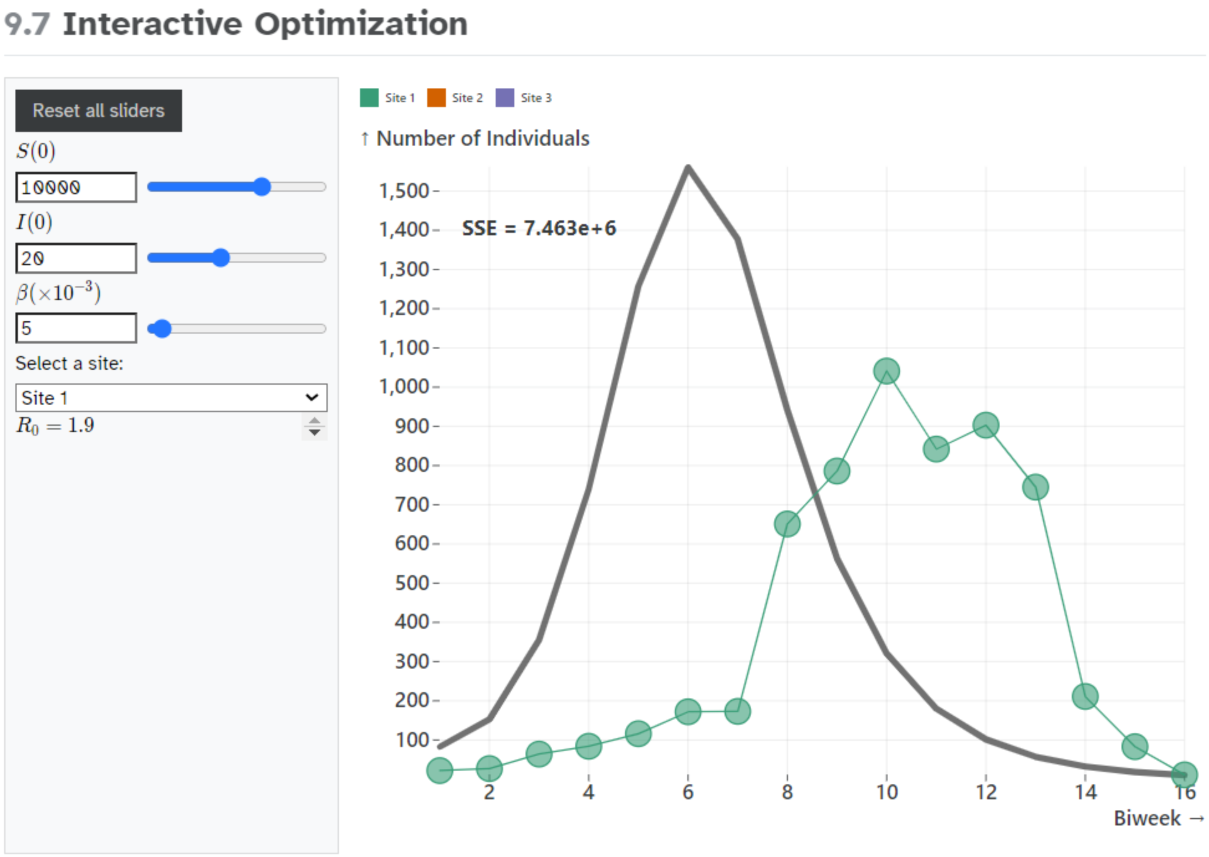
Objectives in this section

- Models can be used to quantitatively interpret data.
- Data can be used to set parameters in models.
- Probability gives us guiding principles for comparing models and data.
- Incorporating sources of uncertainty into models gives them flexibility but it also gives us a lot of new model considerations.

Let’s try it with an SIR model.

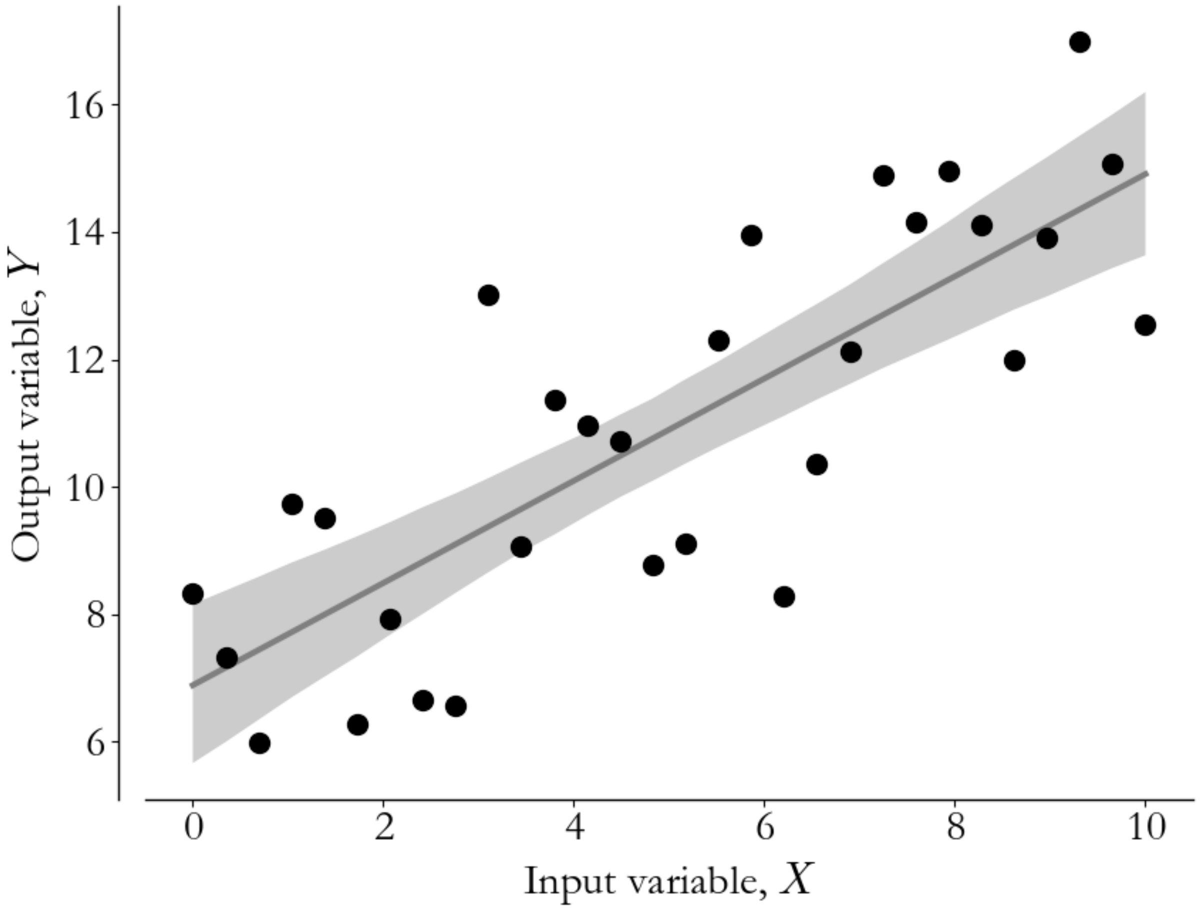
https://sismid2023.callumarnold.com/r-session-03

Start at section 9.5, play at section 9.7



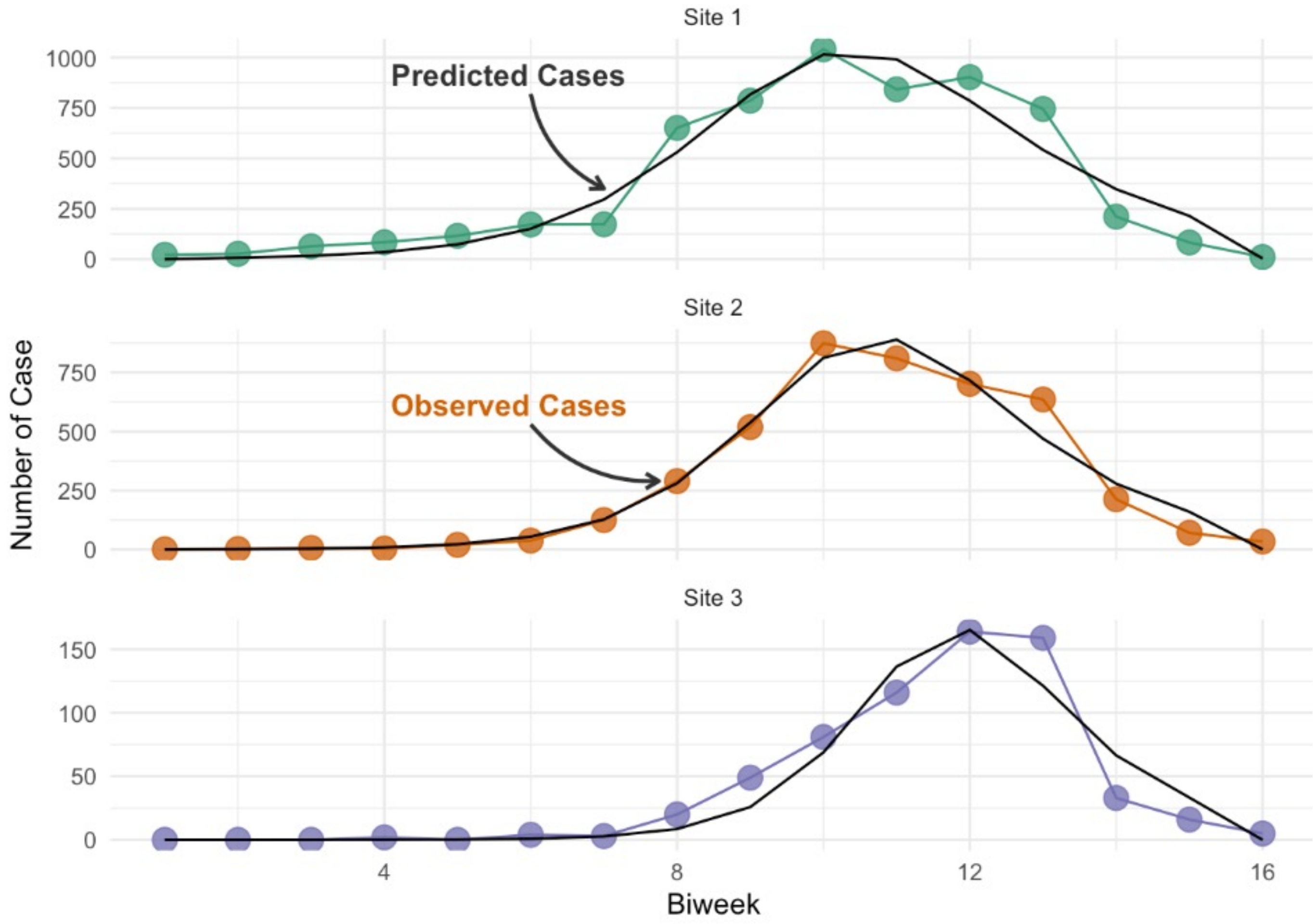
Probability, inference, and linear regression

```
# Linear regression pseudo-code
def LinearRegression(X [Nxp],y [Nx1]):

  # Get the problem dimensions
  N,p = dimension(X)

  # Construct the regression operator
  L = inverse(X.T * X)

  # Calculate the MAP estimate
  beta_hat = L * X.T * y

  # Calculate the residuals
  residual = y - X*beta_hat

  # Calculate the covariance matrix
  sigma_sq = sum(residual^2)/(N-1)
  cov = L * sigma_sq

  return beta_hat, cov, residual
```



Let’s revisit the SIR model.

https://sismid2023.callumarnold.com/r-session-03
Start at section 9.8

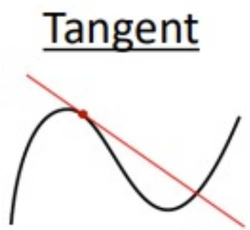


The TSIR model applied to England and Wales

The TSIR model is a natural modification of the chain-binomial.

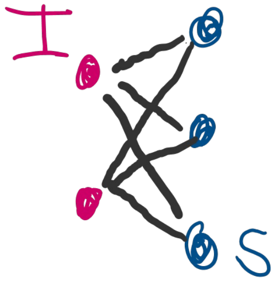
$$
\begin{aligned}
I_t &= \beta_{t-1} S_{t-1} I_{t-1}^{\alpha} \varepsilon_{t-1},\\
S_t &= S_{t-1} + B_{t-1} - I_t,\\
C_t &\sim \text{Binomial}\{I_t, r_t\}
\end{aligned}
$$

- Finkenstadt and Grenfell 2000 is a key paper introducing this model.
- The choice of time step as 2 weeks is practical based on the data but also related to **equating incidence and prevalence in the model.**

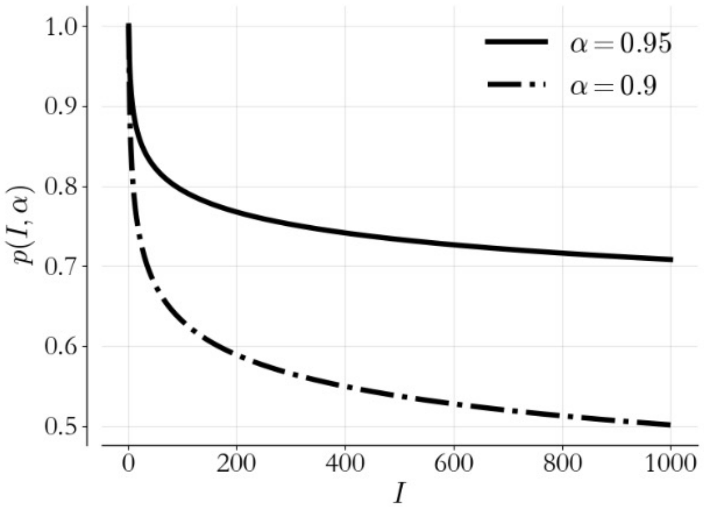


You'll read often that the α **exponent models "network effects"**.

- The exponent is pragmatic – models with it outperform models with α set to 1.

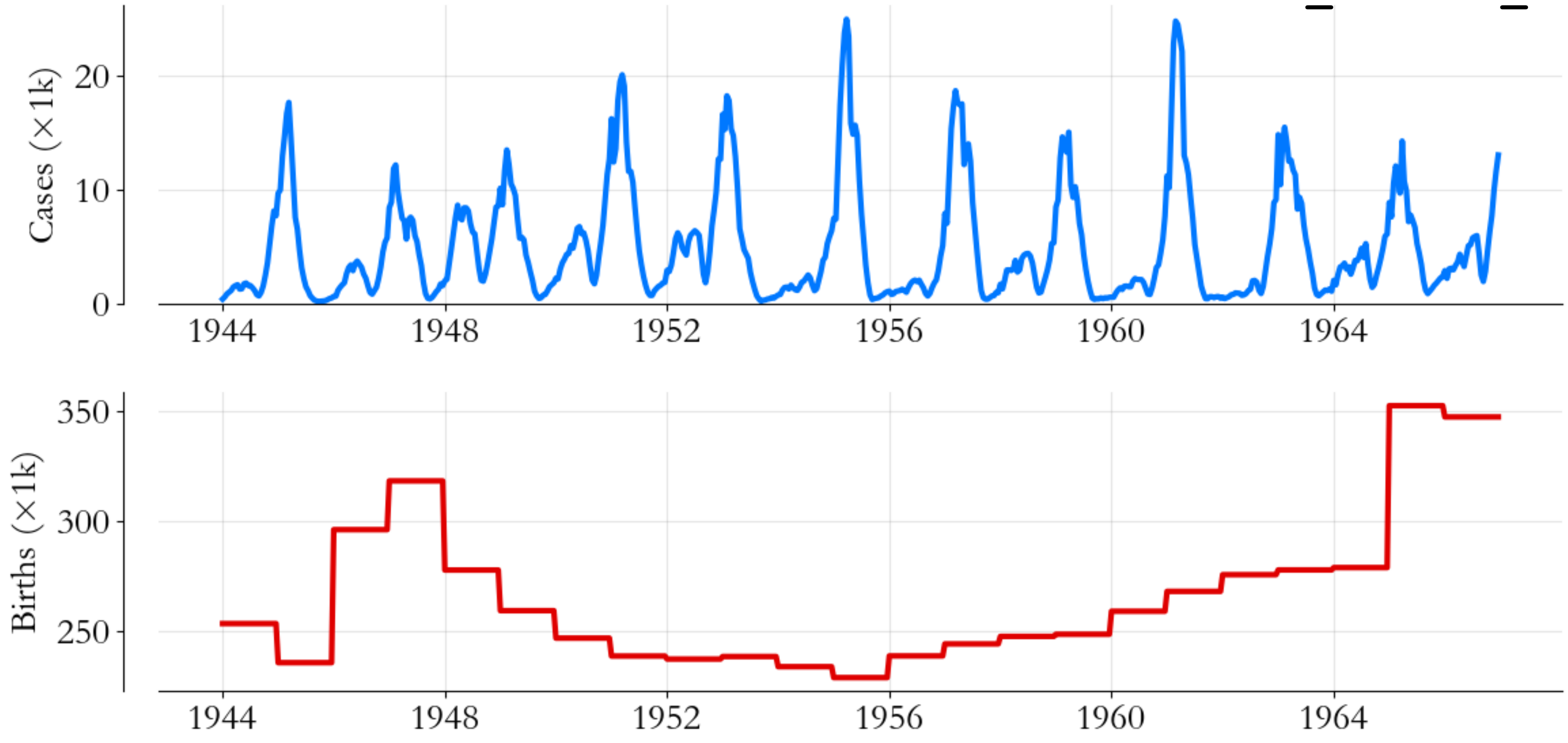


$$
\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1I} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2I} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{S1} & a_{S2} & a_{S3} & \dots & a_{SI} \end{bmatrix}
$$



- All edge weights a_{ij} = 1 implies $S \times I$ edges.
- If for each j, a_{ij} = 1 with probability p, a_{ij} = 0 otherwise, then we expect $SI \times p$ edges.
- We can choose $p(I, \alpha) = 1/I^{1-\alpha}$

UK_measles_data.csv



$$I_t = \beta_{t-1} S_{t-1} I_{t-1}^{\alpha} \varepsilon_{t-1},$$
$$S_t = S_{t-1} + B_{t-1} - I_t,$$
$$C_t \sim \text{Binomial}\{I_t, r_t\}$$

If you want to install Python:
https://www.anaconda.com/download

Let’s write some (pseudo)code

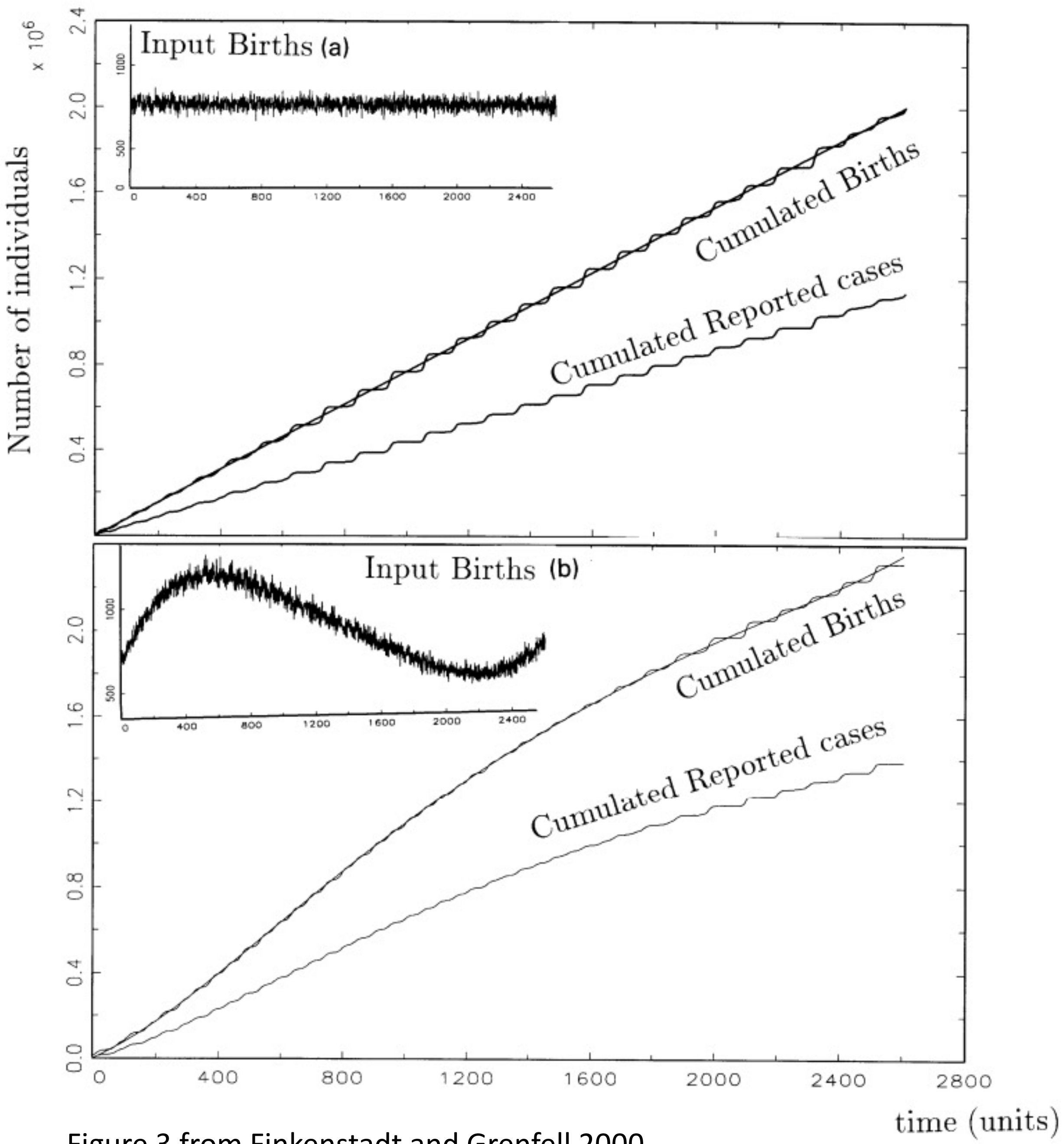


Figure 3 from Finkenstadt and Grenfell 2000

```
# Reporting rate estimate
def ReportingRateRegression(Ct,Bt):

  # Calculate the cumulative sums
  cumul_C = cumsum(Ct[1:])
  cumul_B = cumsum(Bt[:-1])

  # Solve the regression problem
  rho, var_rho, rr_resid =
       LinearRegression(cumul_C,cumul_B)

  # Calculate the reporting rate
  # estimate and variance
  r = 1/rho
  var_r = var_rho/(rho^4)

  return rho, var_rho, rr_resid, r, var_r
```

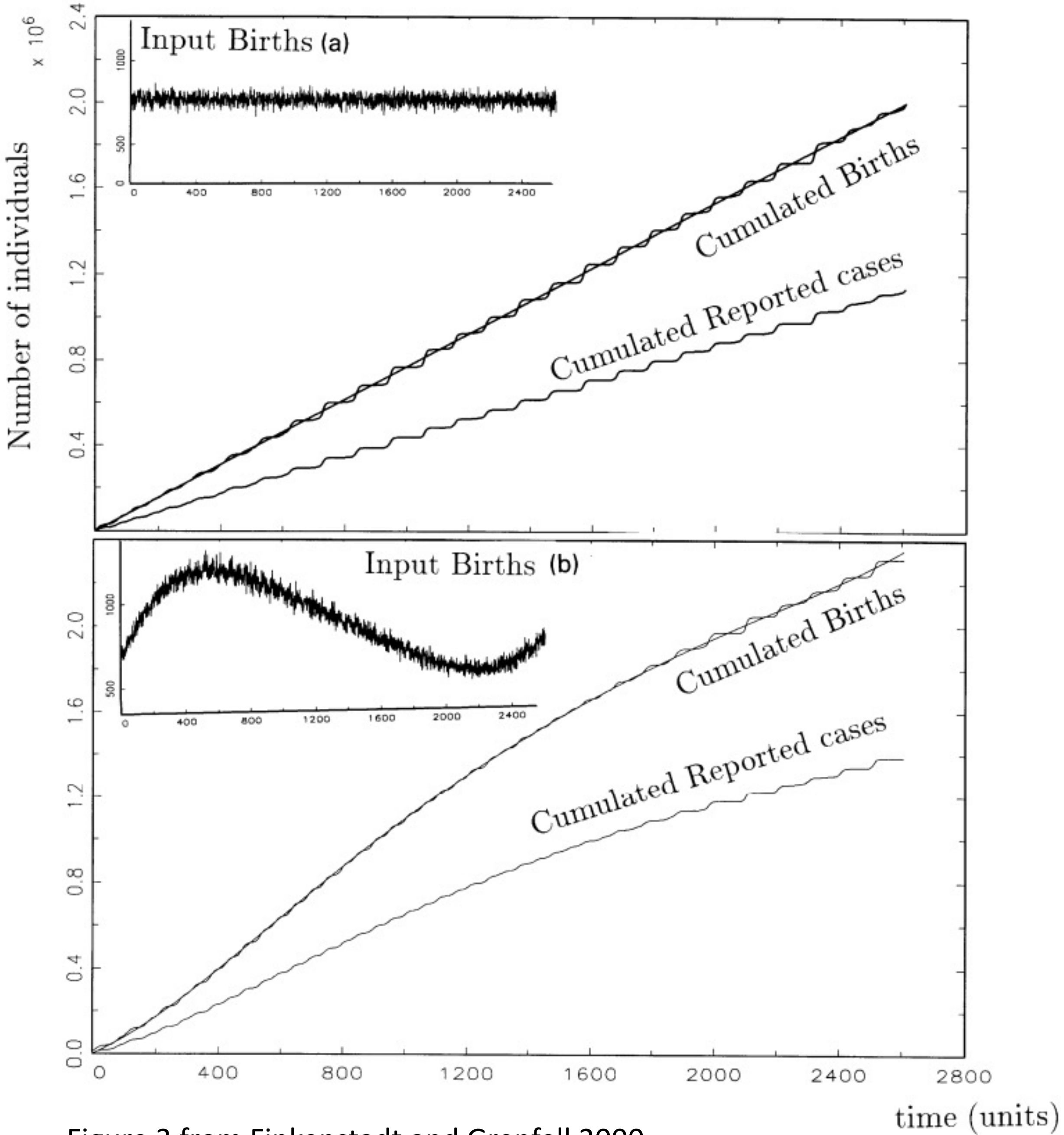


Figure 3 from Finkenstadt and Grenfell 2000

Let’s write some (pseudo)code

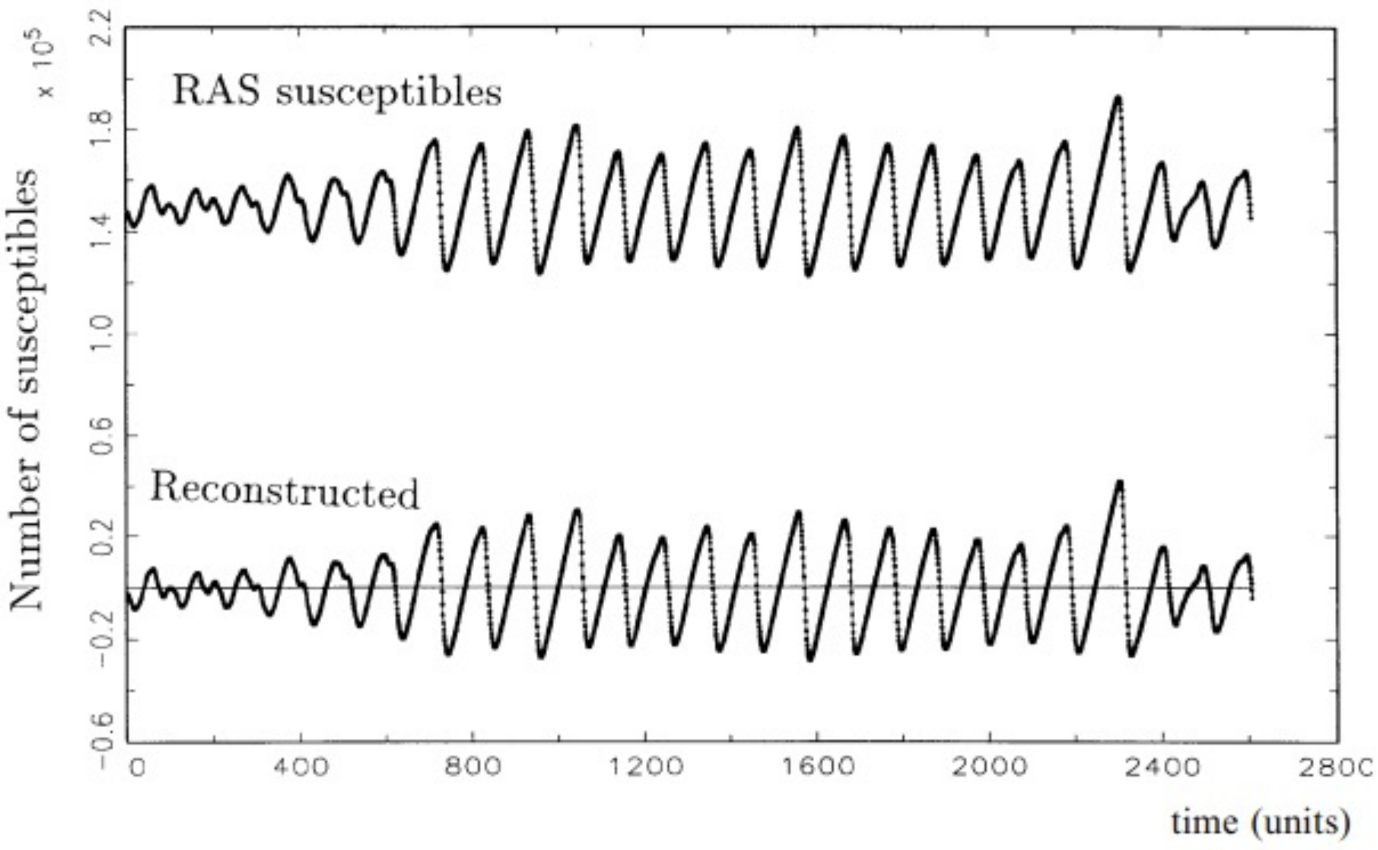


Figure 4 from Finkenstadt and Grenfell 2000

```
# Hidden state reconstruction
def HiddenStateRecon(Ct,Bt,rho):

  # Calculate the It estimate
  It = rho*(Ct+1)-1

  # Compute the residual
  Zt = zeros(dimension(It))
  Zt[1:] = cumsum(Bt[:-1] - It[1:])

  return It, Zt
```

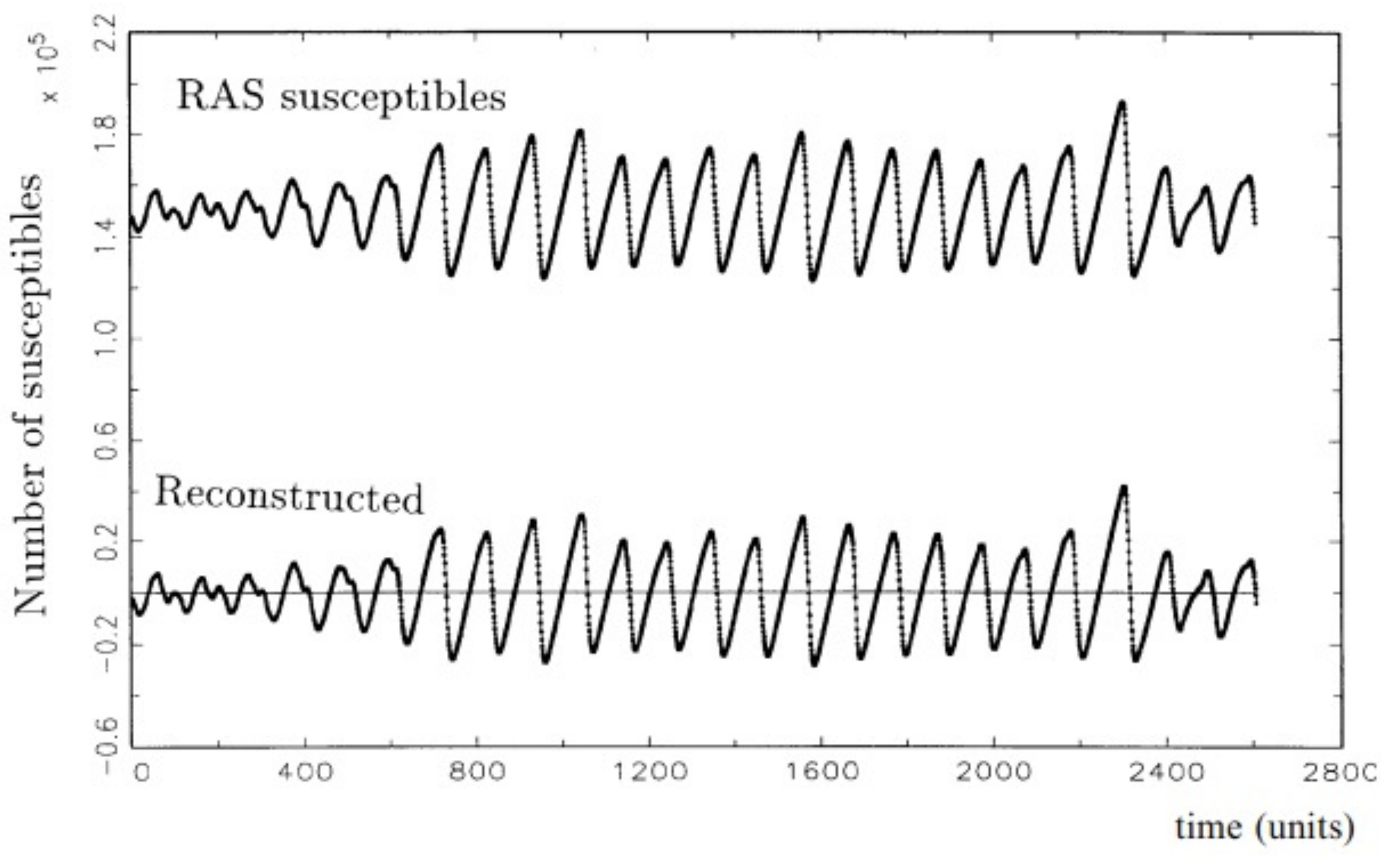
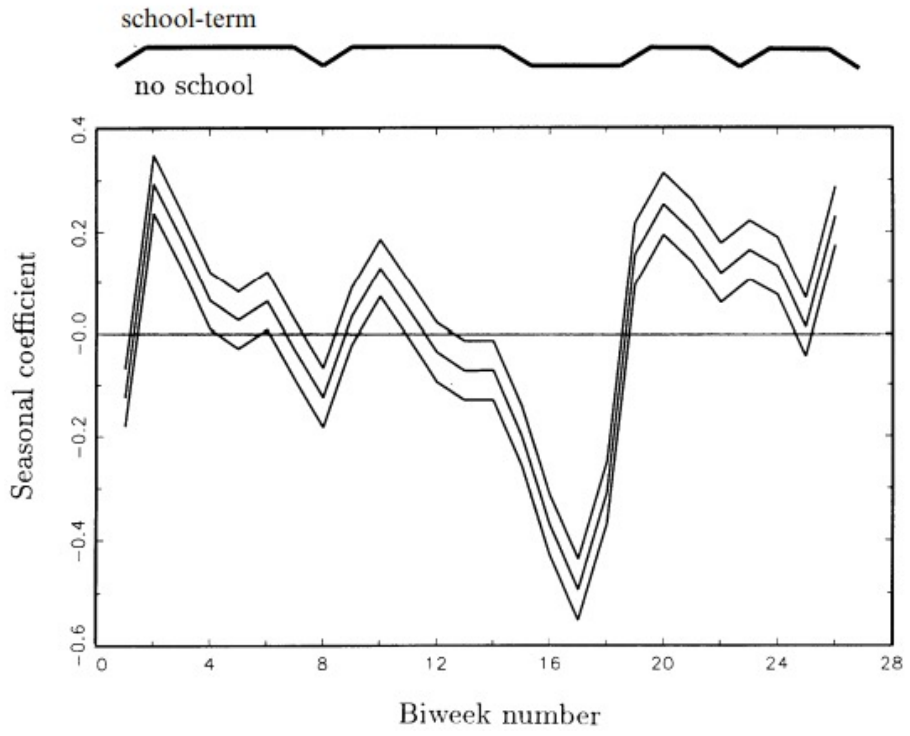
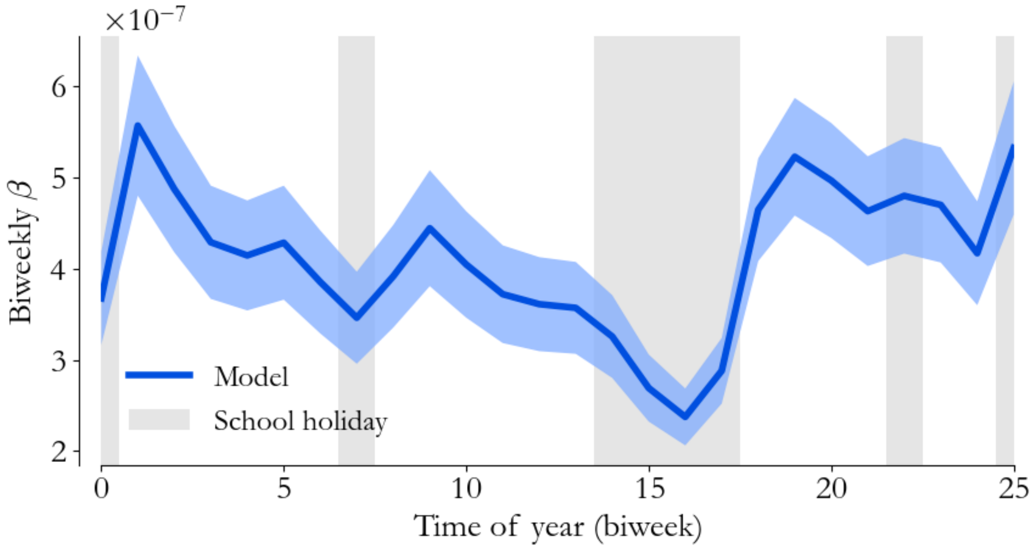
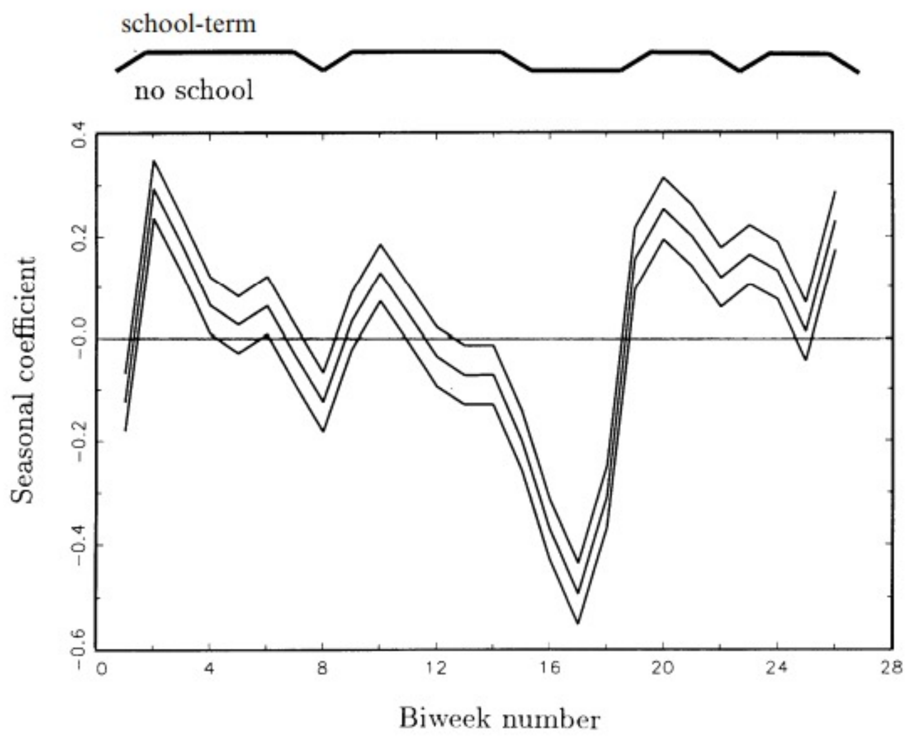


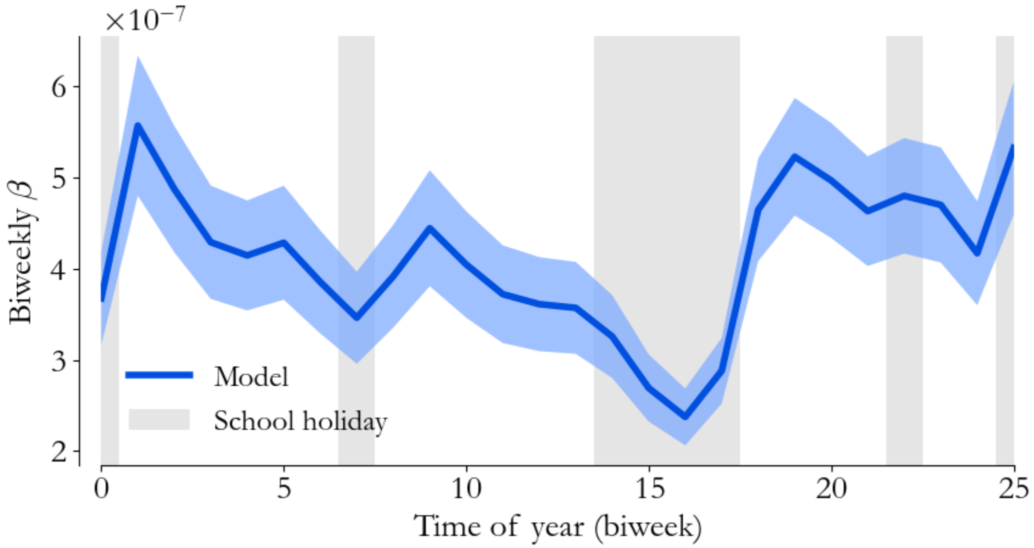
Figure 4 from Finkenstadt and Grenfell 2000



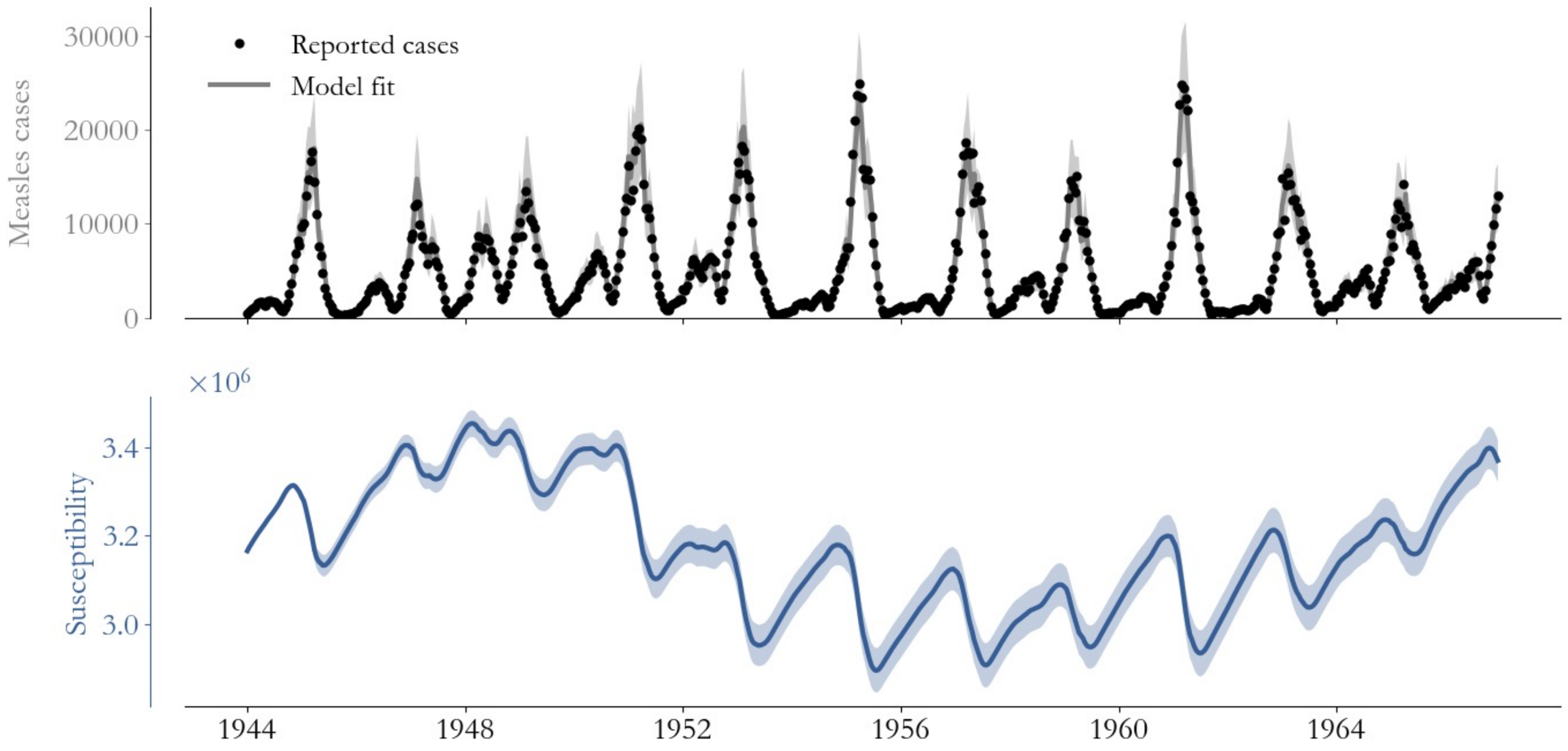


Let’s write some (pseudo)code






```
# Transmission regression function
def TransmissionRegression(It,Zt,tau):

  # Construct the design matrix
  # which has tau+2 columns, and the
  # response variable
  X = zeros((len(It)-1,tau+2))
  X[:,-1] = Zt[:-1]
  X[:,-2] = log(It[:-1])
  X[:,t mod tau] = 1
  y = log(It[1:])

  # Solve the regression problem
  theta, cov, residual = LinearRegression(X,y)
  sigma2 = sum(residual^2)/(len(X)-1)

  # Unpack theta and compute estimates
  # beta shown here for example
  xt, alpha, 1/S0 =
         theta[:tau],theta[tau], theta[tau+1]
  beta = exp(xt)/S0

  # And the std error (again, beta shown)
  sig2s = diagonal(cov)
  sig2 = sig2s[:tau] + sig2s[tau+1]/(S0**2)
  beta_sig = beta*sqrt(sig2)

  # Compile parameters in a table
  model = {"beta":beta, "beta_sig":beta_sig,
          "sig_eps":sqrt(sigma2), ...}
  return model
```

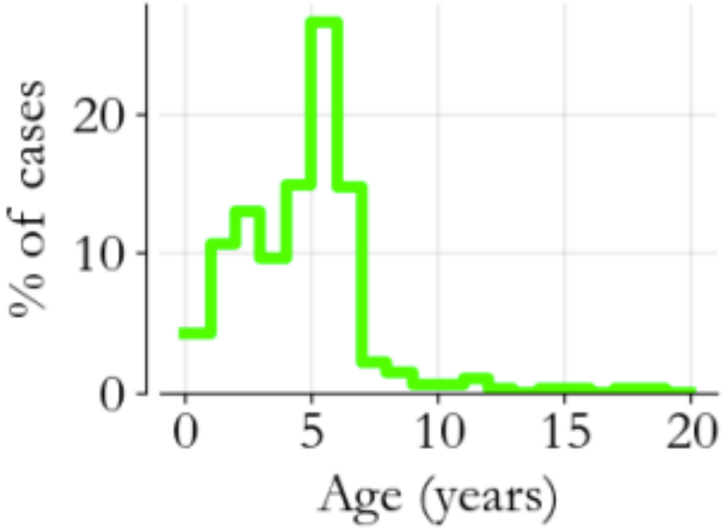


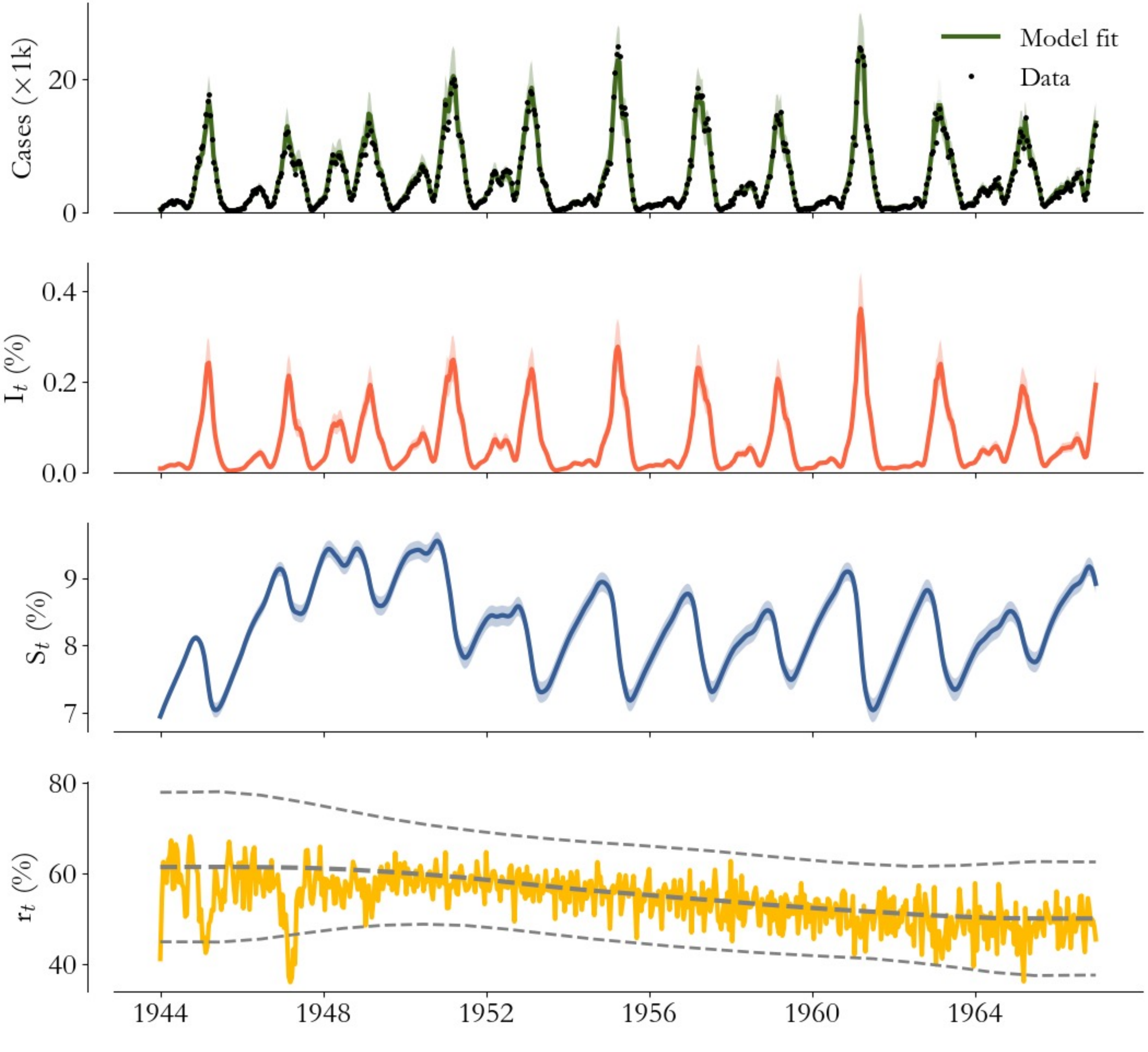
$$
\begin{aligned}
I_t &= \beta_{t-1} S_{t-1} I_{t-1}^{\alpha} \varepsilon_{t-1}, \\
S_t &= S_{t-1} + B_{t-1} - I_t, \\
C_t &\sim \mathrm{Binomial}\left\{I_t, r_t\right\}
\end{aligned}
$$

How do we validate a model like this?

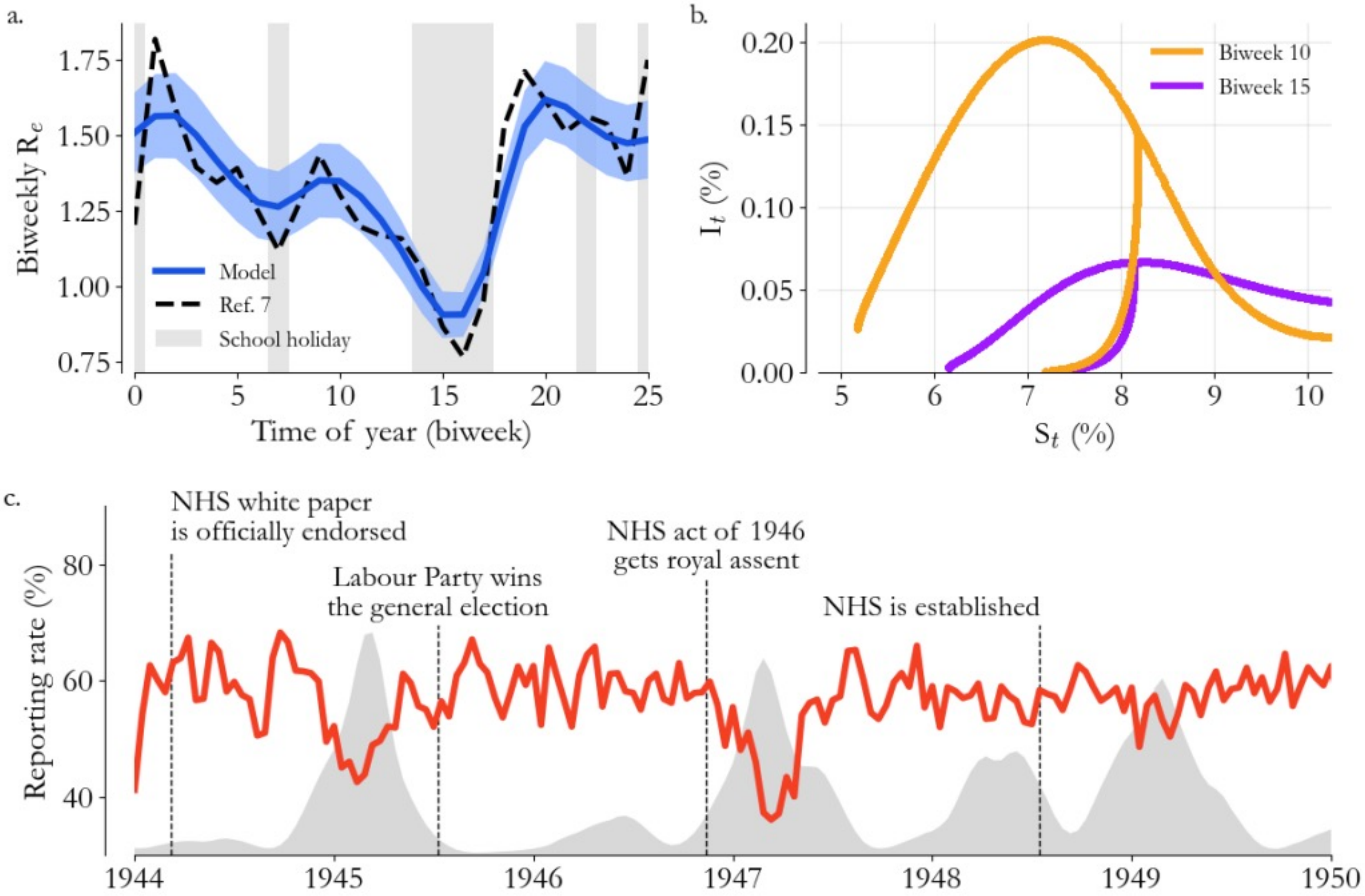
Using a similar approach and some more data we create a more sophisticated model

$$\mathcal{L}(\beta_t, \varepsilon_t, \alpha, r_t, S_0) = \frac{26T-1}{2}\ln \hat{\sigma}_\varepsilon^2 + \frac{(S_0 - \mathrm{E}[S_0])^2}{2\mathrm{V}[S_0]} + \sum_t \frac{(r_t - \mathrm{E}[\tilde{r}_t])^2}{2\mathrm{V}[\tilde{r}_t]}$$





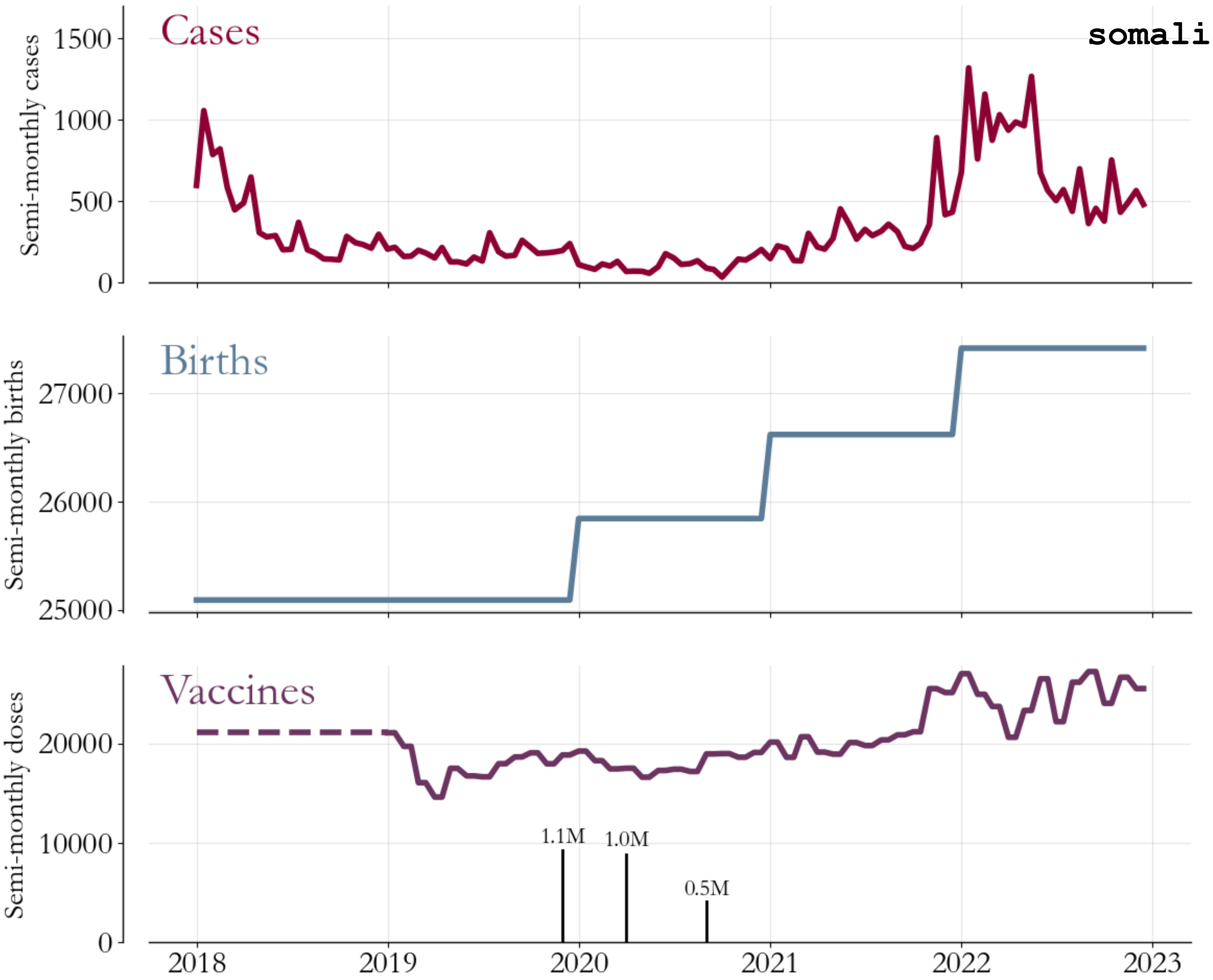
https://arxiv.org/pdf/2202.11222.pdf



Quantities that are difficult to estimate are then difficult to validate!

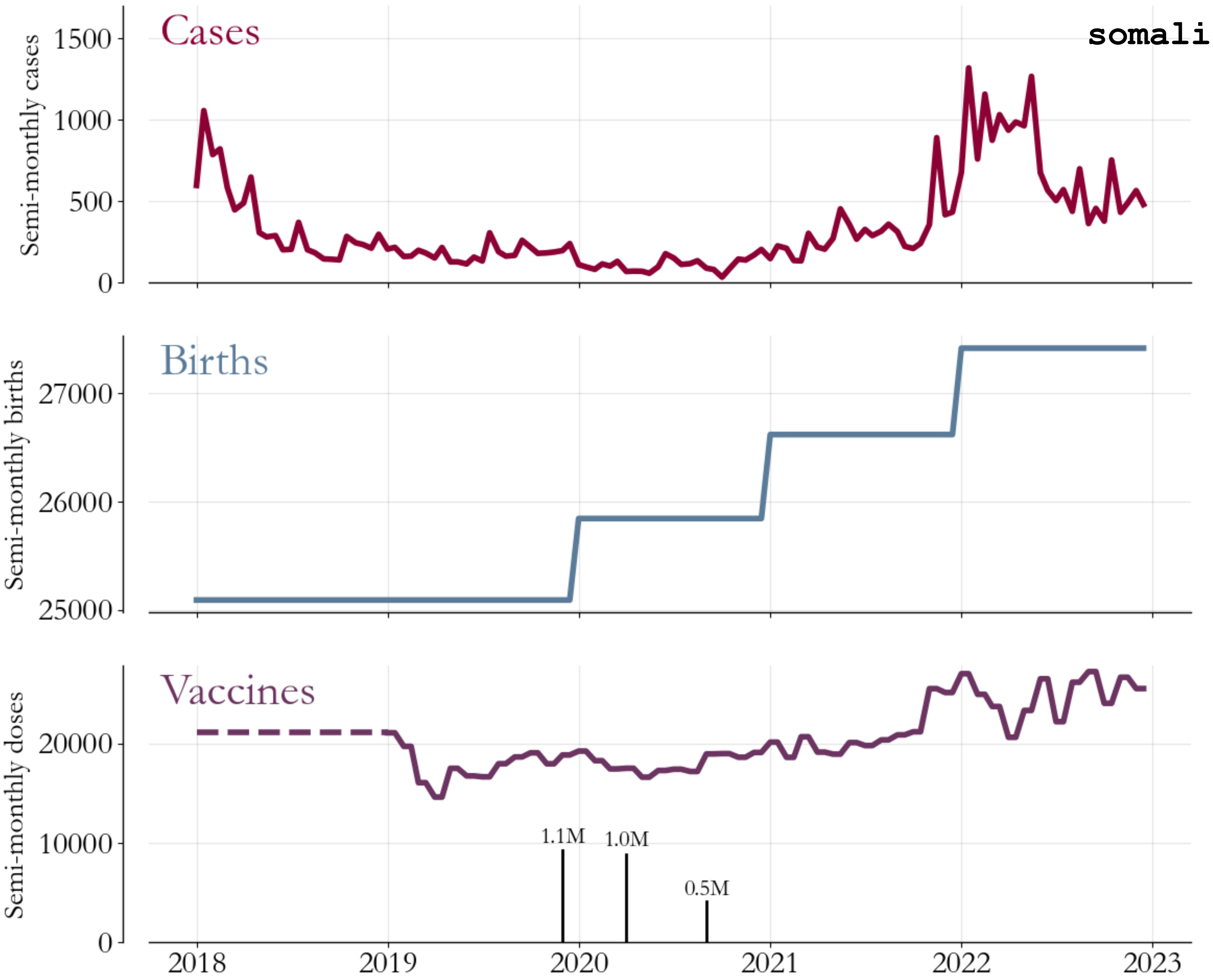
The TSIR model applied to SIA impact estimation in Somalia

somalia_synthetic_dataset.csv



$$I_t = \beta_{t-1} S_{t-1} I_{t-1}^{\alpha} \varepsilon_{t-1},$$
$$S_t = S_{t-1} + B_{t-1} - I_t,$$
$$C_t \sim \text{Binomial}\left\{I_t, r_t\right\}$$

somalia_synthetic_dataset.csv

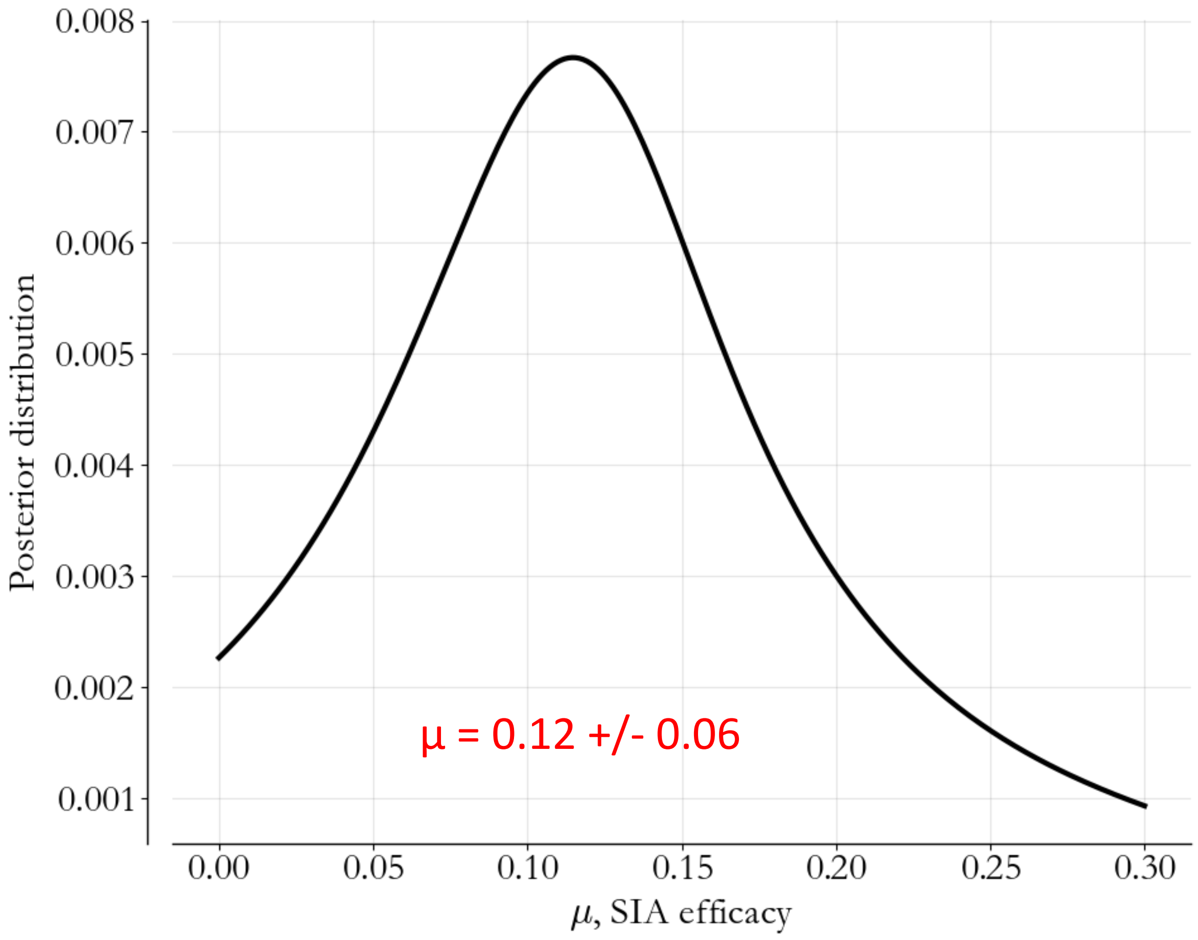


$$I_t = \beta_{t-1} S_{t-1} I_{t-1}^{\alpha} \varepsilon_{t-1},$$
$$S_t = S_{t-1} + B_{t-1} - I_t,$$
$$C_t \sim \text{Binomial}\{I_t, r_t\}$$

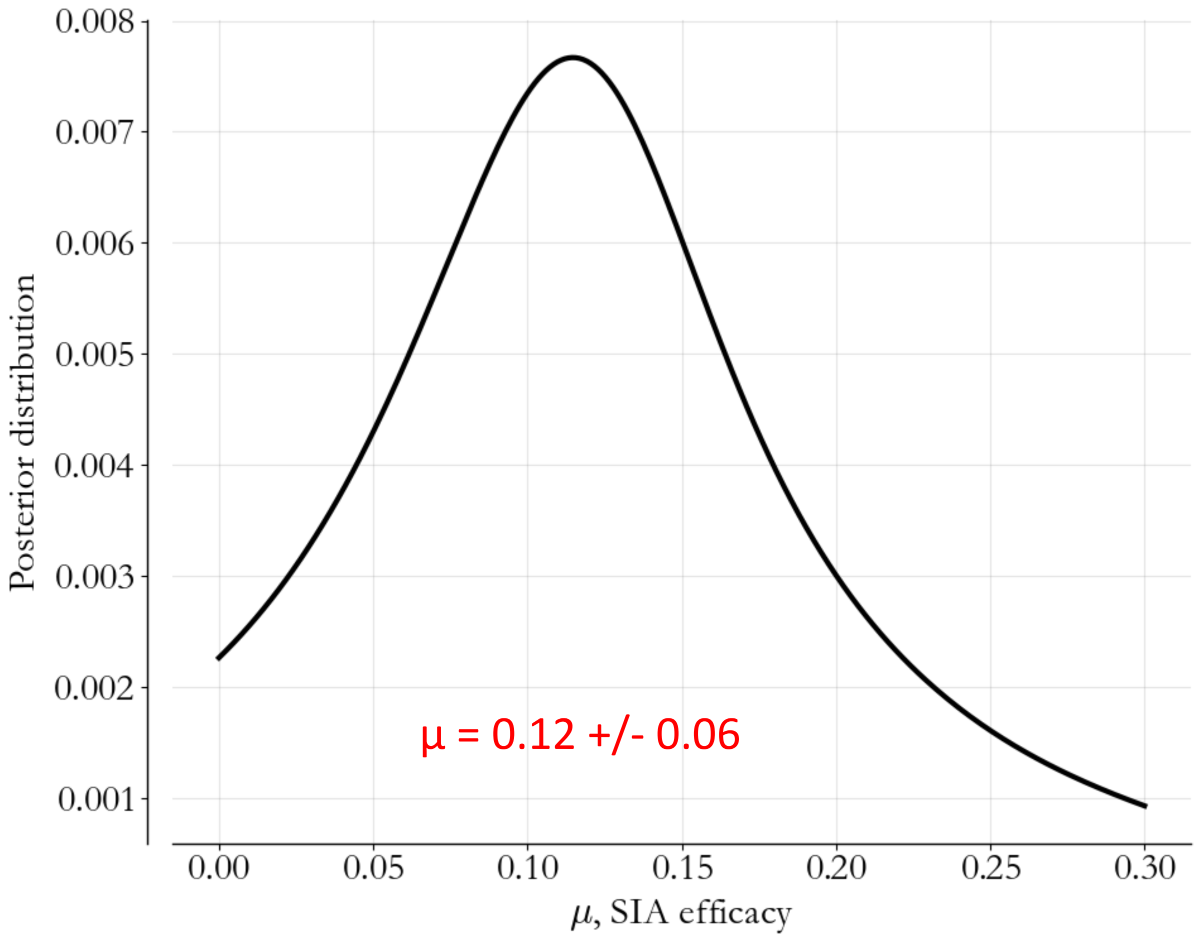
$$B_t \longrightarrow \tilde{B}_t$$

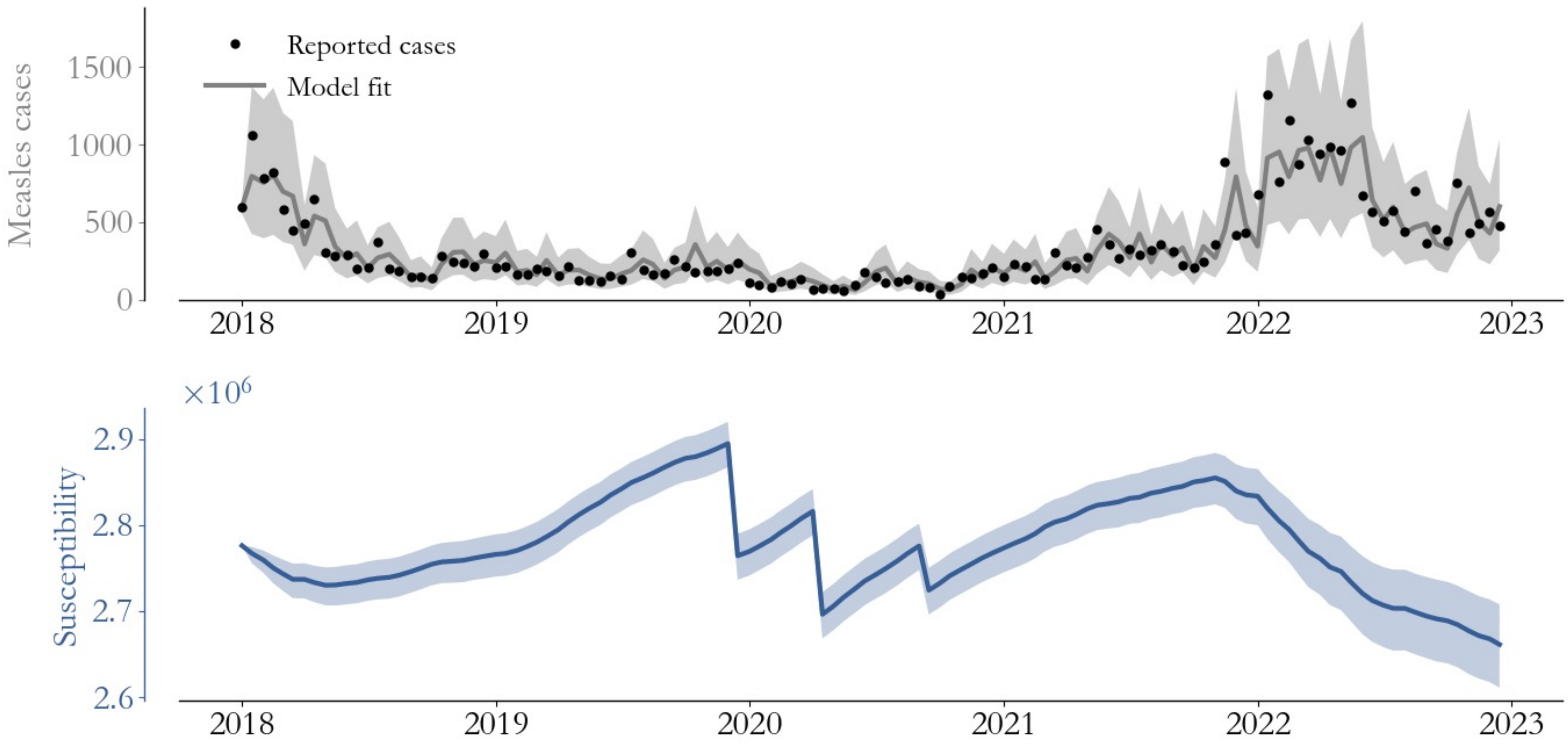
$$\tilde{B}_t = B_t - \varepsilon V_t^{(MCV1)} - \mu V_t^{(SIA)}$$

Let’s write some (pseudo)code

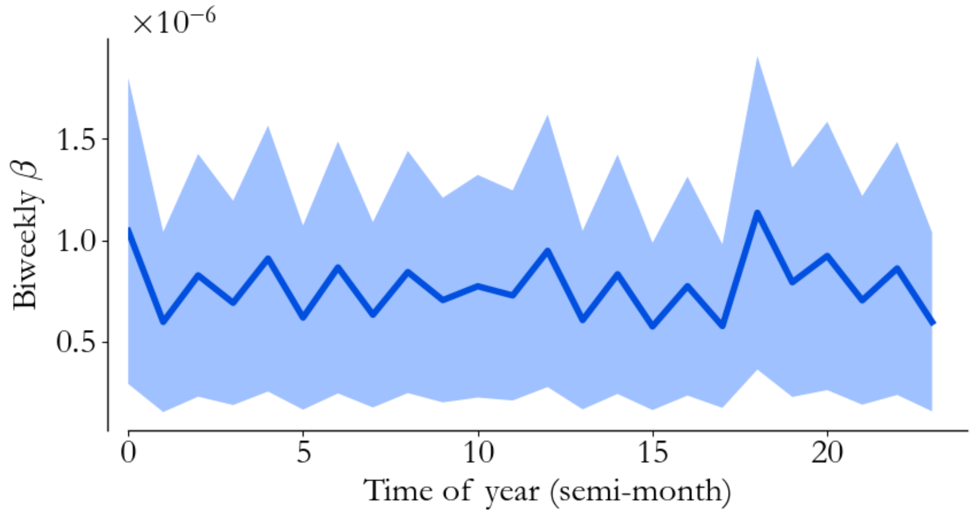


```
# SIA efficacy posterior
def NegLogPosterior(mu,data):

  # Create vaccine adjusted births
  adj_births = data[“Bt”]
        -0.85*data["mcv1"]
        - mu*data["sia"]

  # Fit the model with this SIA efficacy
  rho, _, rr_resid, _, _ =
        ReportingRateRegression(data[“cases”],
                                  adj_births)
  Zt, It = HiddenStateRecon(data[“cases”],
                         adj_births,rho)
  model = TransmissionRegr(It,Zt,24)

  # Compute the combined variances
  return log(sum(rr_resid^2)
        +(length(data)-1)*log(model[“sig_eps”])
```



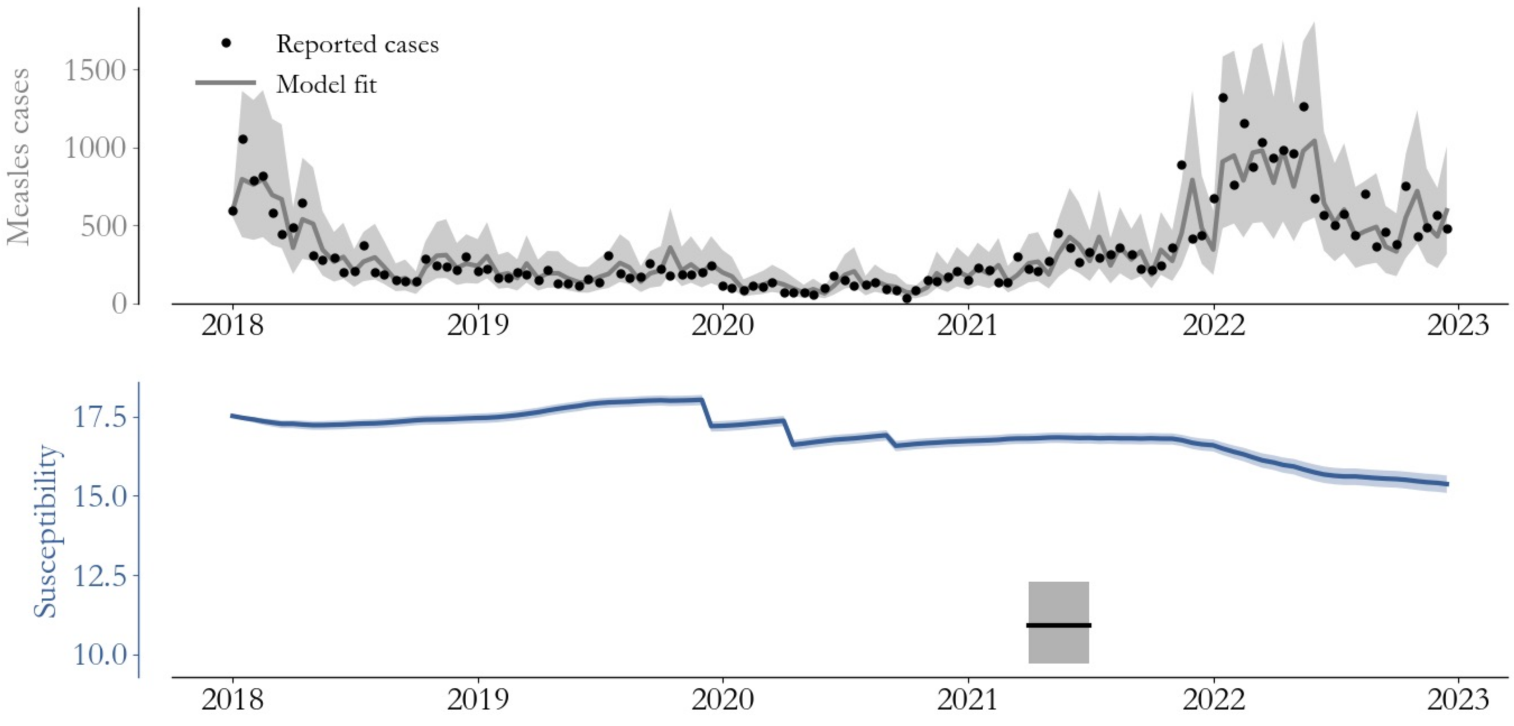


How do we validate a model like this?

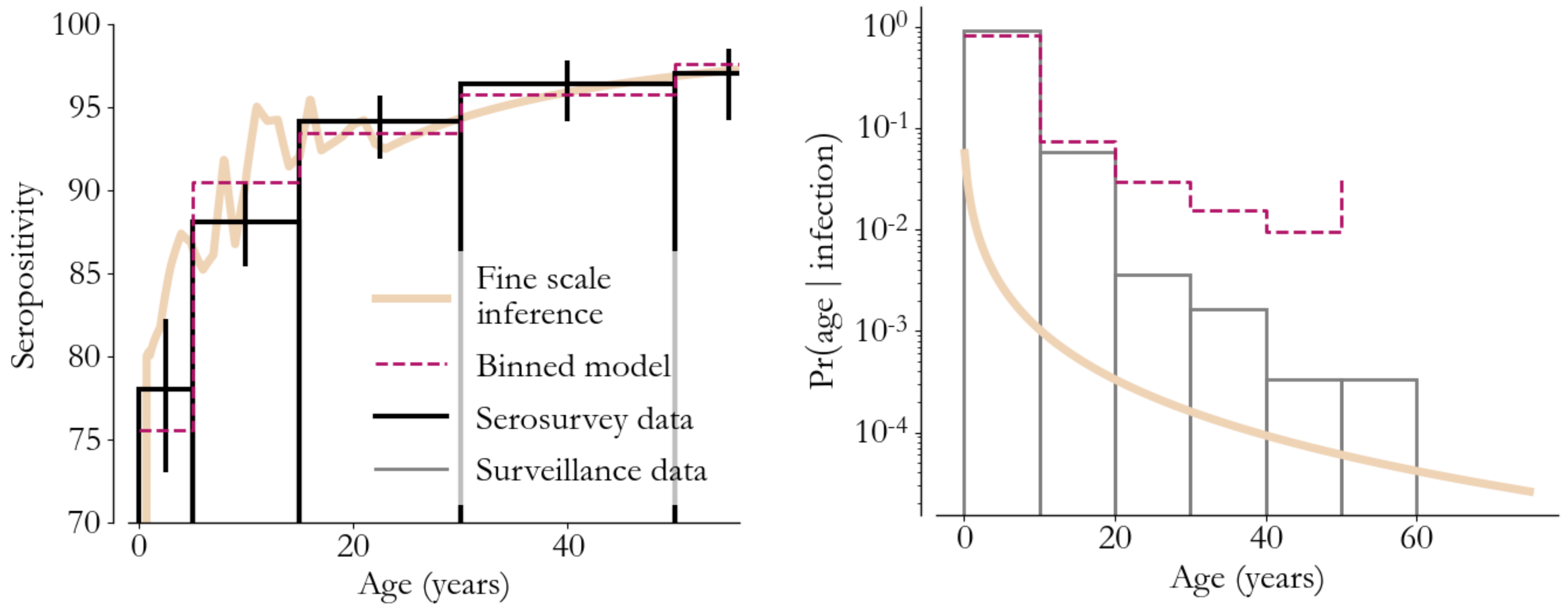


The seasonality profile is very flat!

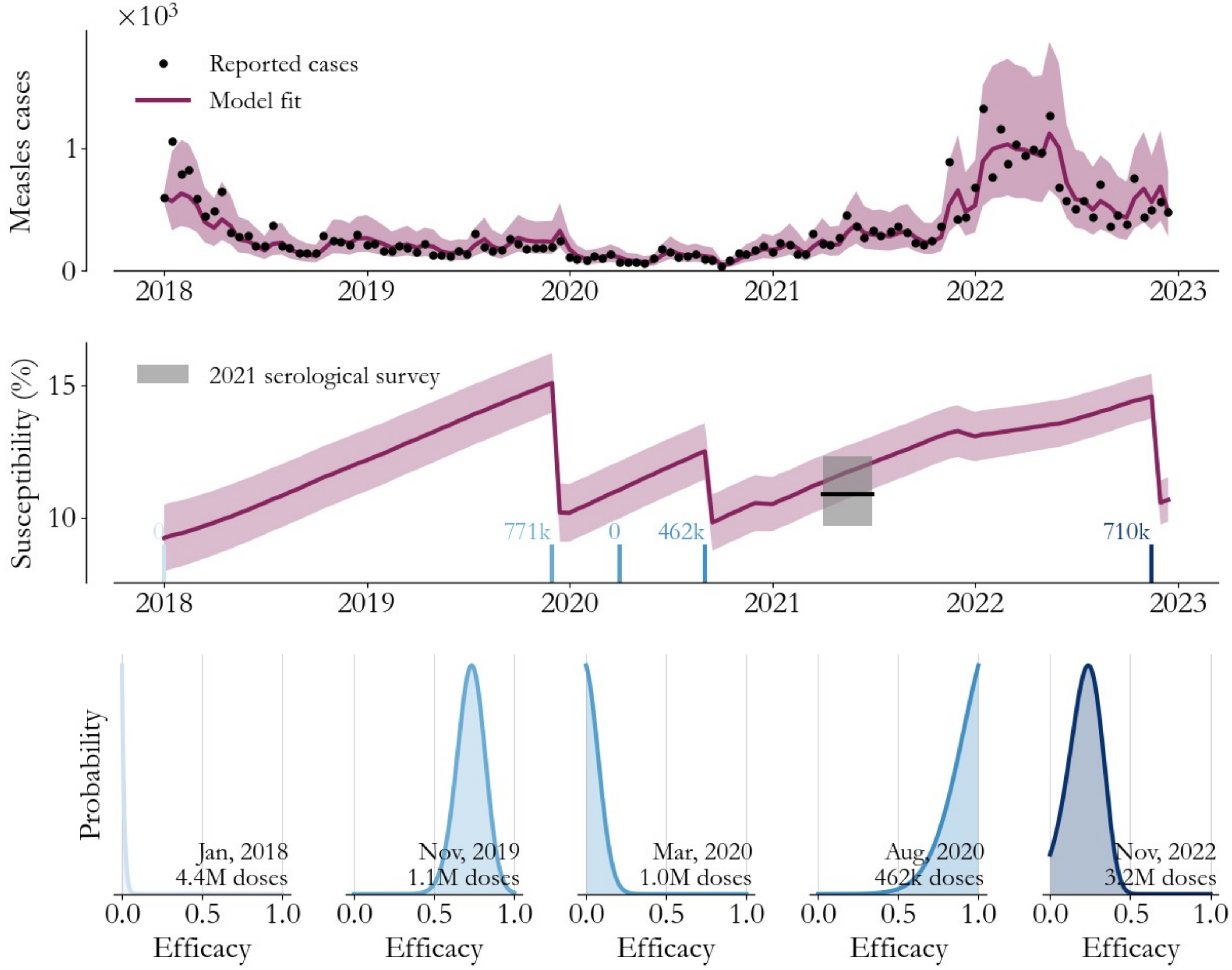
A susceptibility estimate from a 2021 serological survey is much lower!

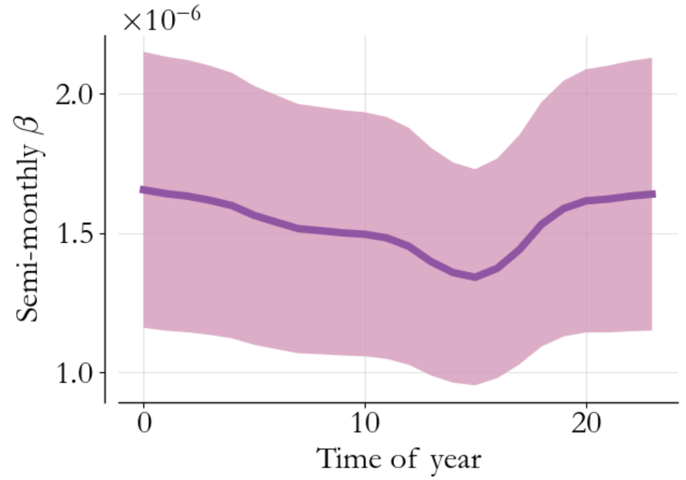


We can again incorporate age information, this time from a serological survey in 2021



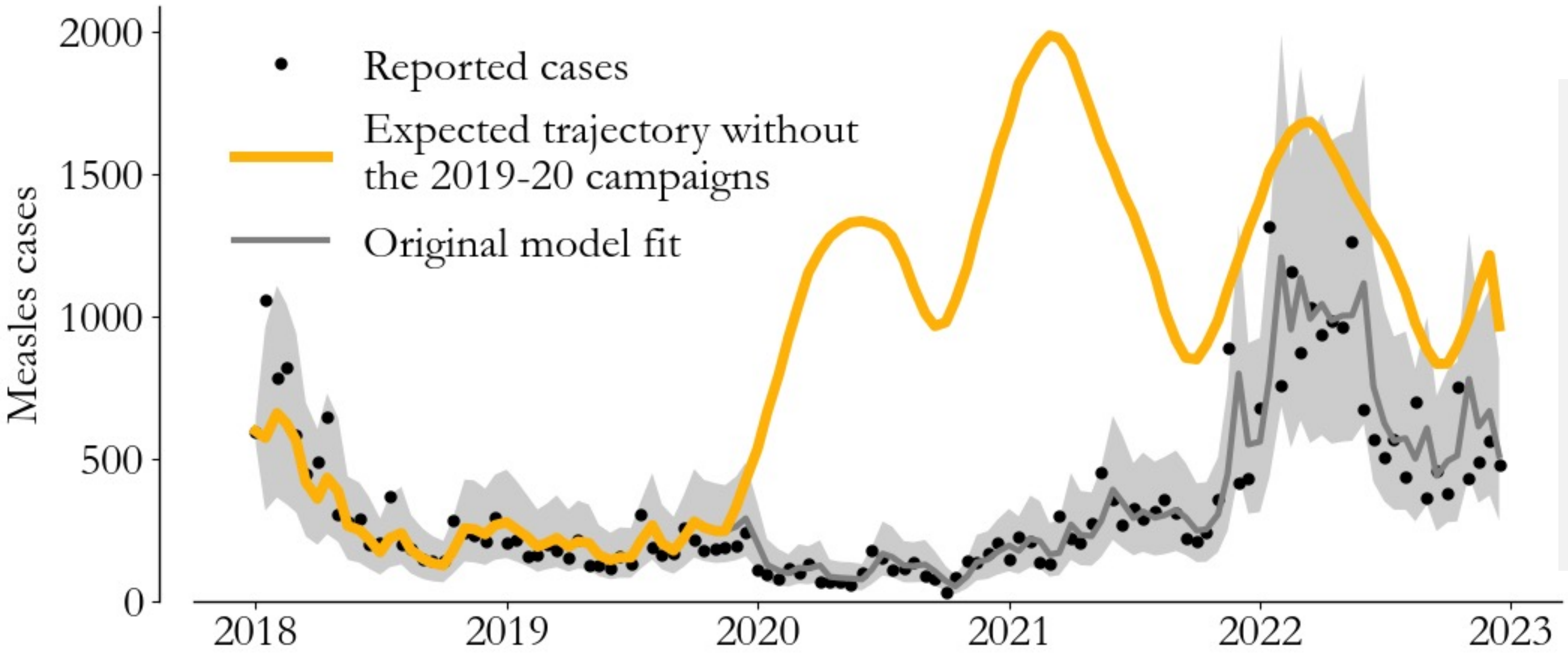
P(sero +) = P(vaccinated before infection)
+ P(infected and not vaccinated)





Independent SIA effects with prior information from the serosurvey gives us a more stable model with intuitive inferences.

To illustrate the importance of the 2019-2020 SIAs, we can **simulate epidemics in their absence**.



Comparing the model with and without the 2019-2020 SIAs, **we estimate that the campaigns prevented 64 thousand (52 to 77 95% interval) measles case reports**, corresponding to 1.3 million (1.1 to 1.6 95% interval) measles infections.

Examples in this block