Report COVID19 Model

Måns Magnusson

2020-05-05

Contents

Summary / Take Away

- Using the increased number of covariates seem to be better, model-wise compared to baseline. I think in many istuations our models now perform better than Imperial College Londons baseline.
- Mobility on transit station seem to be a good covariate to use.
- The regression coefficients are really sensitive. They need to be bounded for the model not to explode.

Current flaws to be fixed:

- IFR for Finland now set to IFR for Sweden

Potential ways forward:

- Use hospitalizations instead of deaths. I think this has better quality and is more data.
- Focus on Finland (and Sweden) at county level, although data may be missing.
- Include serological studies to estimate cumulative infections, weekly.
- Partial pooling of mobility variables.

Model descriptions

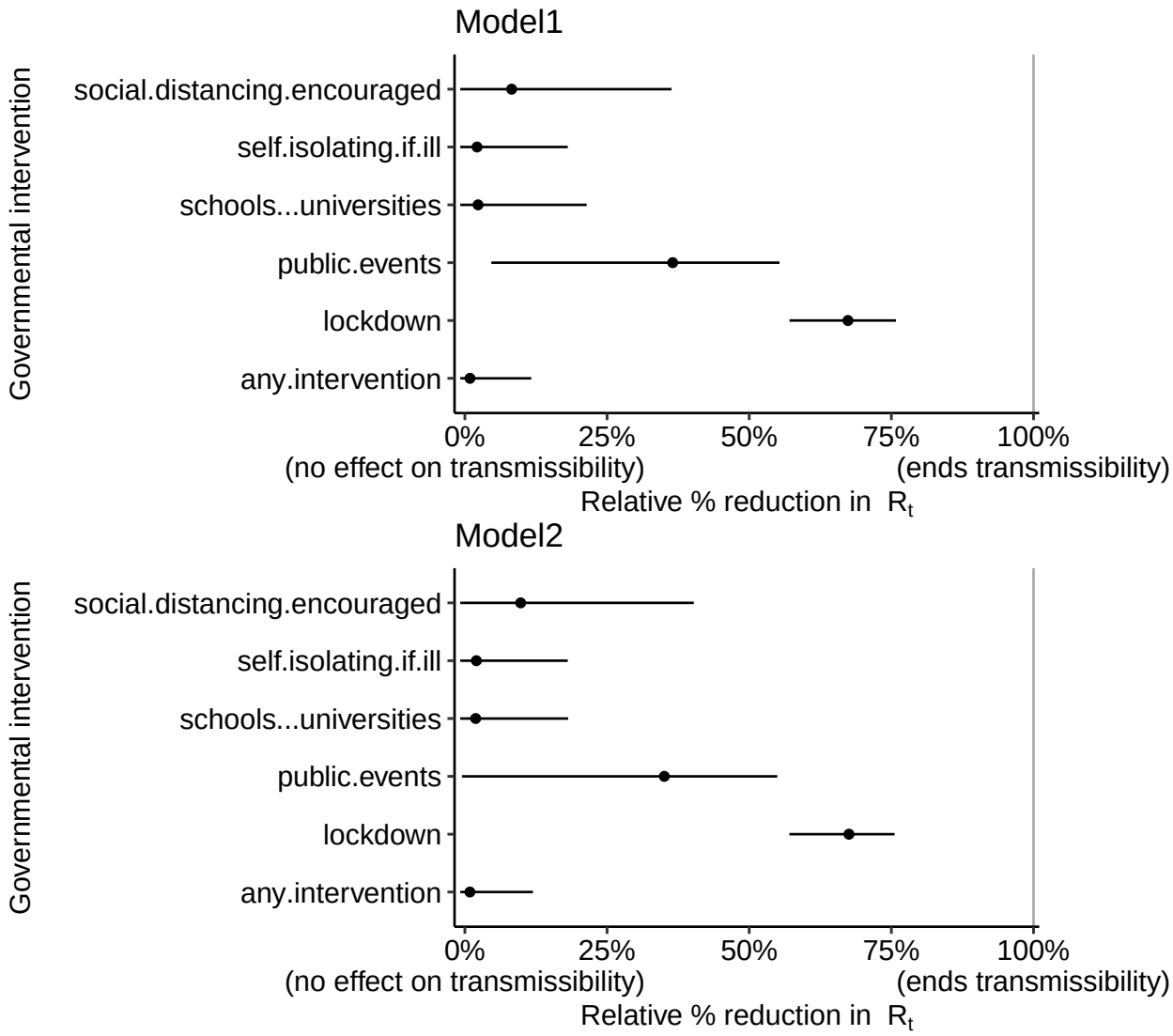
- Model 1: The baseline model, without partial pooling of lockdown
- Model 2: The baseline model, with partial pooling of lockdown variable
- Model 3: Using Oxford Response data as indicator variables
- Model 4: Google Mobility in transit stations
- Model 5: Google Mobility in transit stations and SeverityIndex
- Model 6: Google Mobility in transit stations and Oxford Binary data

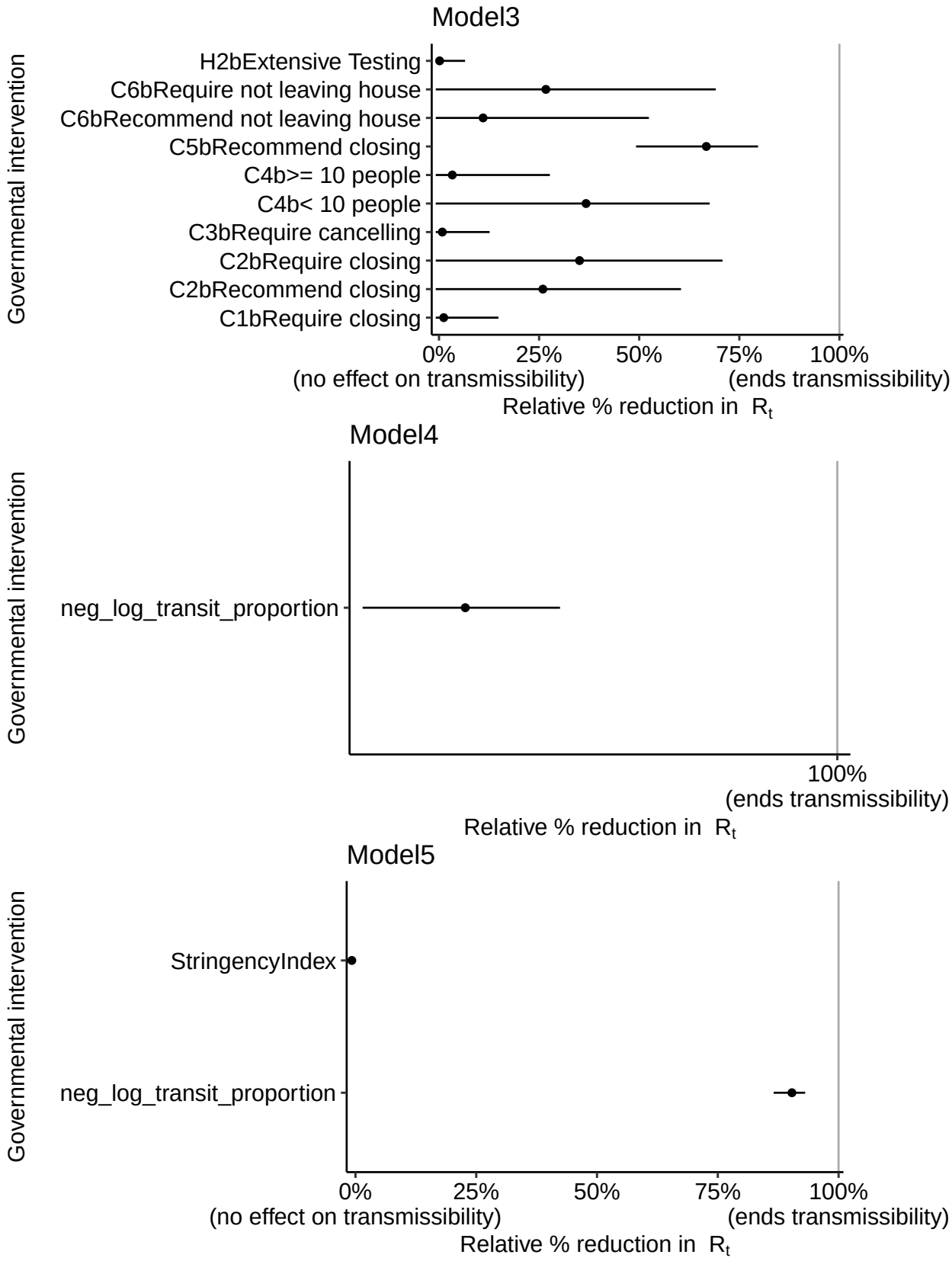
Oxford covariate descriptions

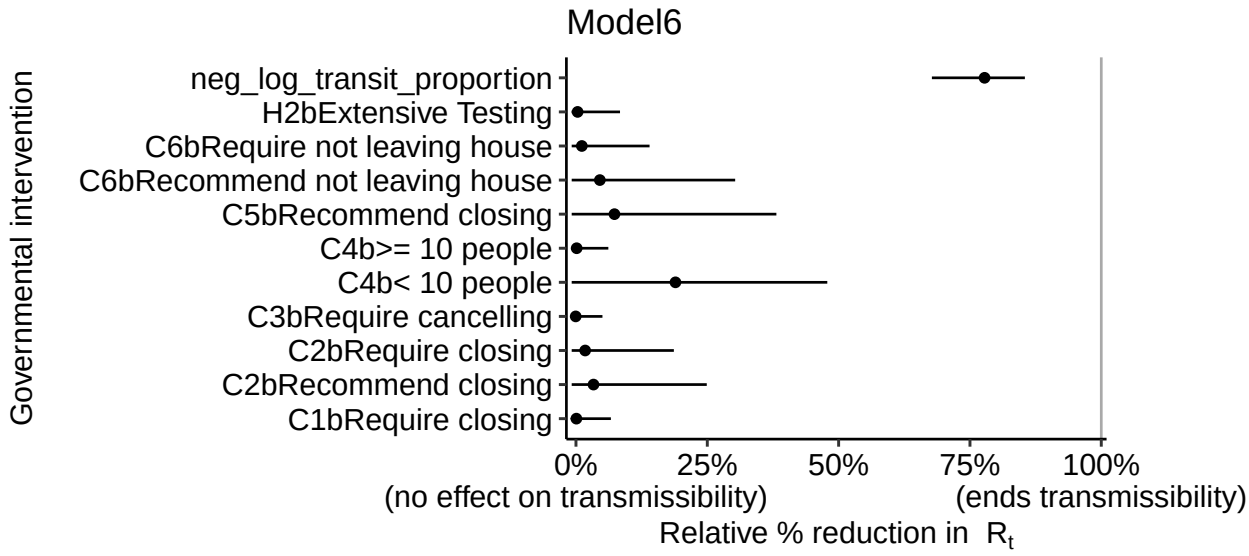
- C1: School closing
- C2: Workplace closing
- C3: Cancel public events
- C4: Restrictions on Gatherings
- C5: Close public transport
- C6: Stay at home requirements

- H2: Testing policy

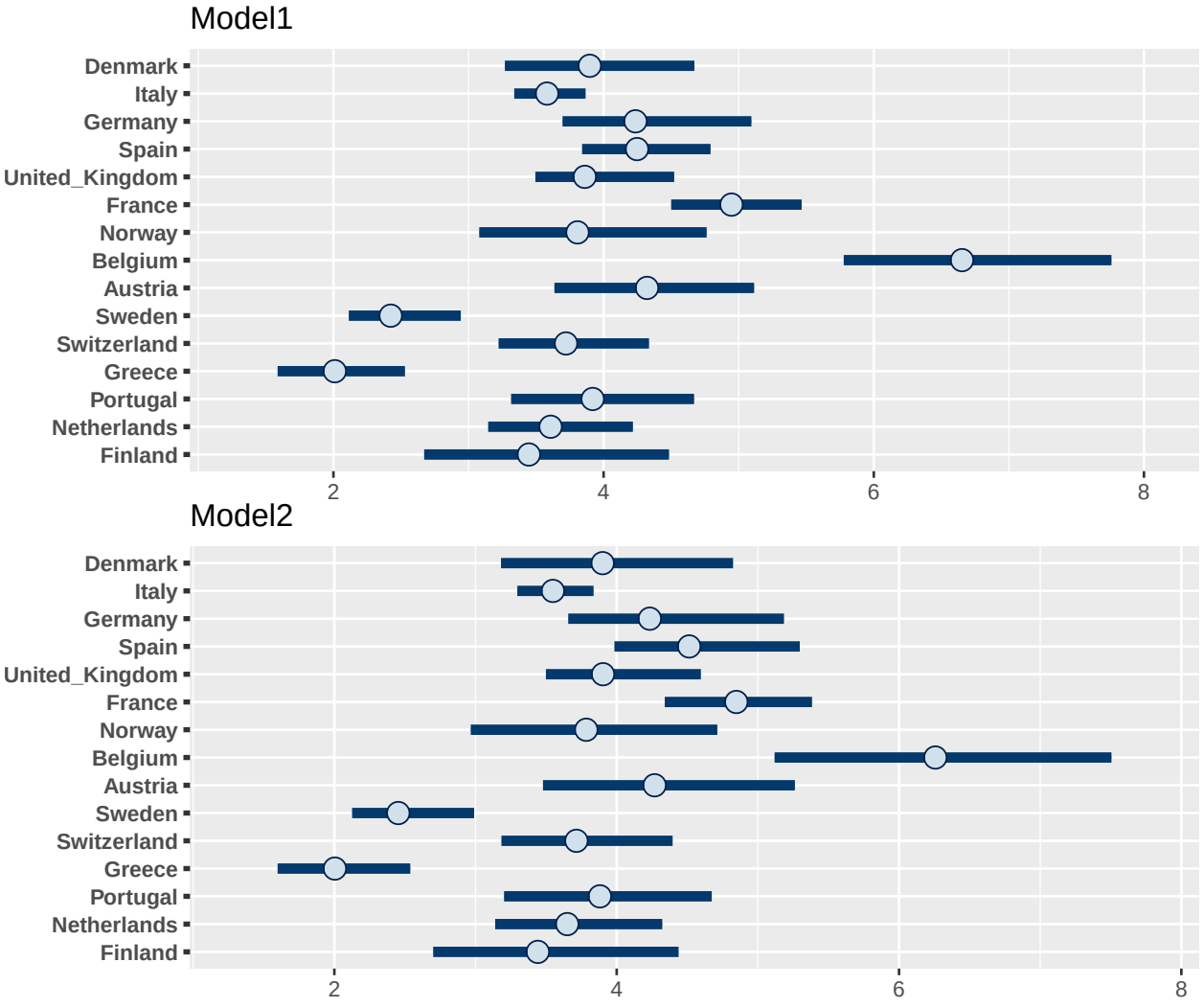
Covariate effects

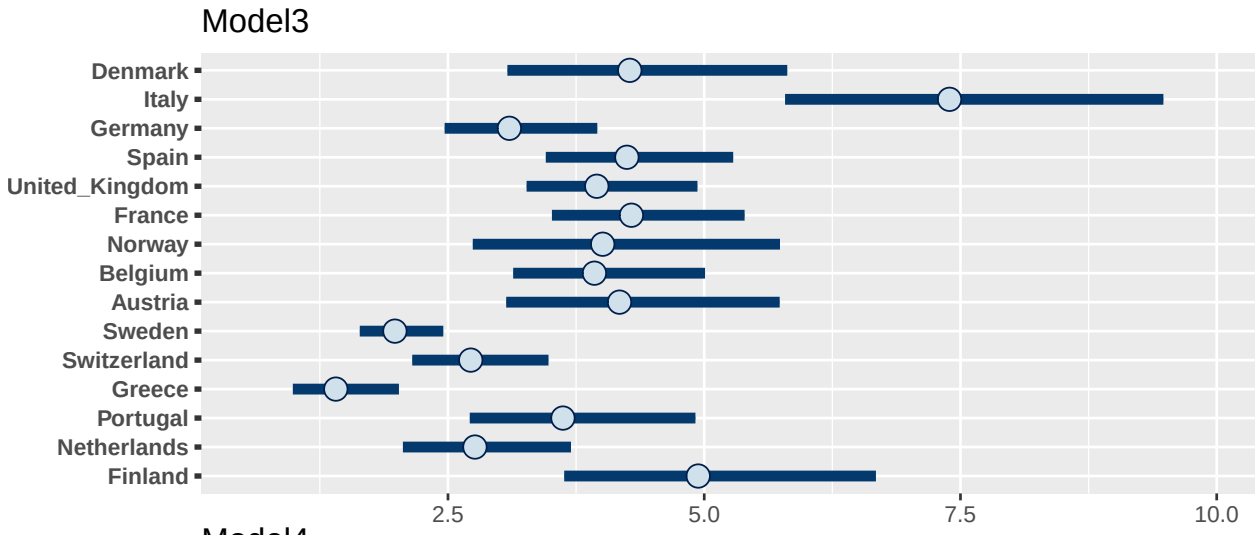


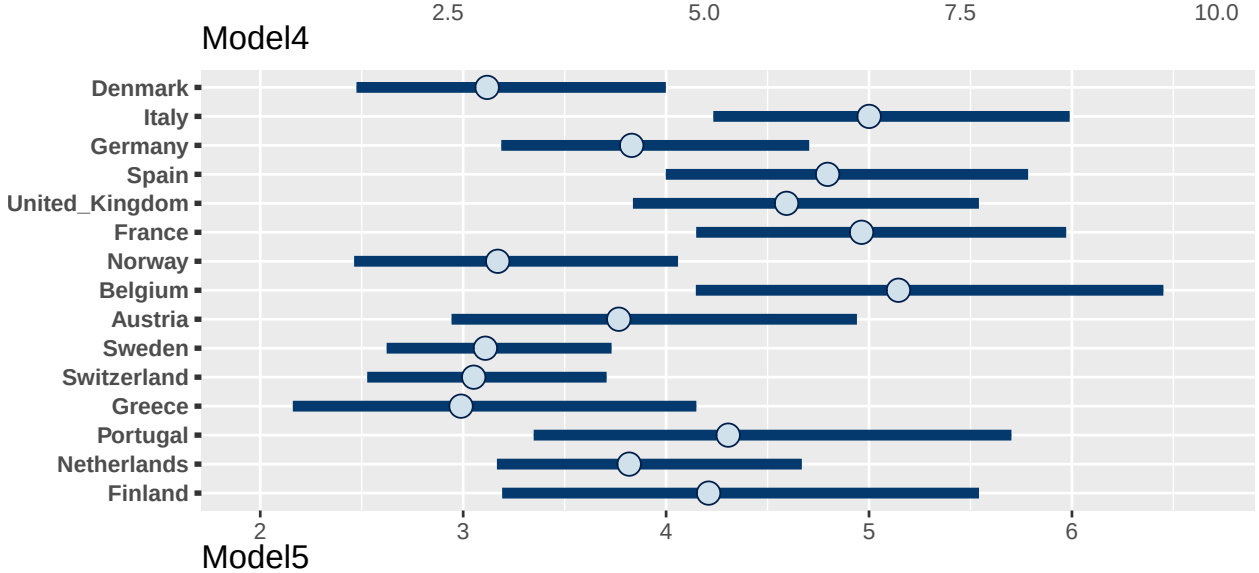

Model3
Governmental intervention
H2bExtensive Testing
C6bRequire not leaving house
C6bRecommend not leaving house
C5bRecommend closing
C4b>= 10 people
C4b< 10 people
C3bRequire cancelling
C2bRequire closing
C2bRecommend closing
C1bRequire closing
0%
25%
50%
75%
100%
(no effect on transmissibility)
(ends transmissibility)
Relative % reduction in R_t
Model4
Governmental intervention
neg_log_transit_proportion
100%
(ends transmissibility)
Relative % reduction in R_t
Model5
Governmental intervention
StringencyIndex
neg_log_transit_proportion
0%
25%
50%
75%
100%
(no effect on transmissibility)
(ends transmissibility)
Relative % reduction in R_t

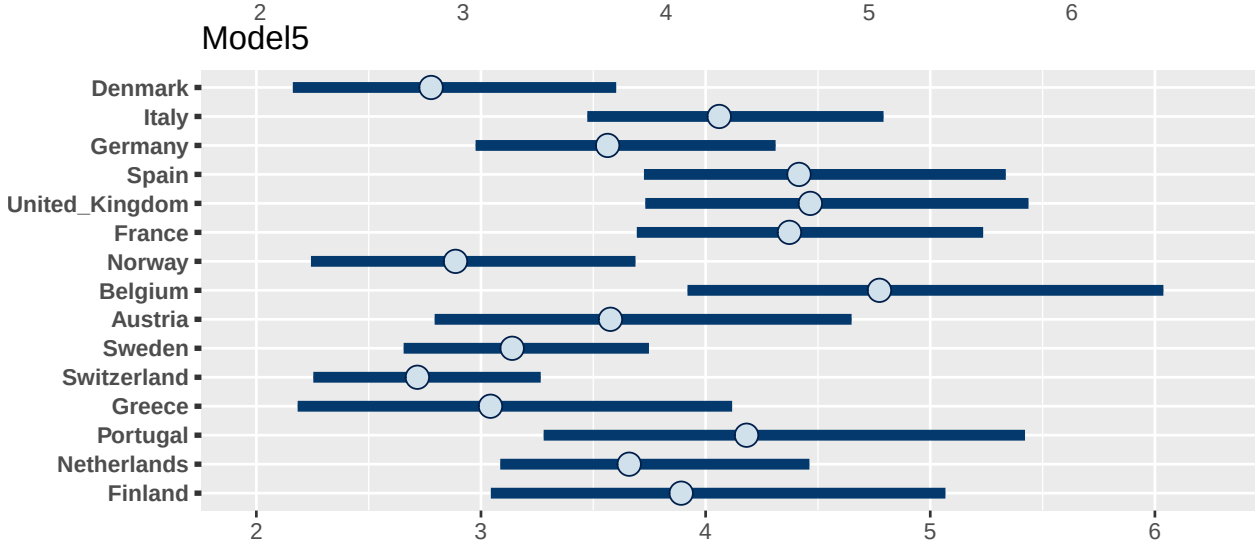


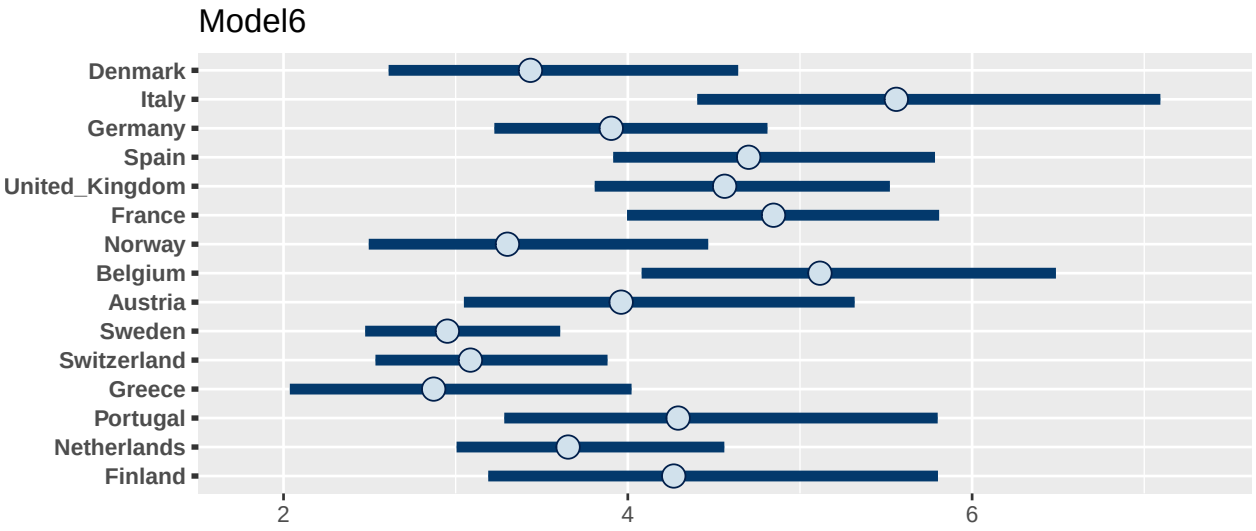
R0 for different countries



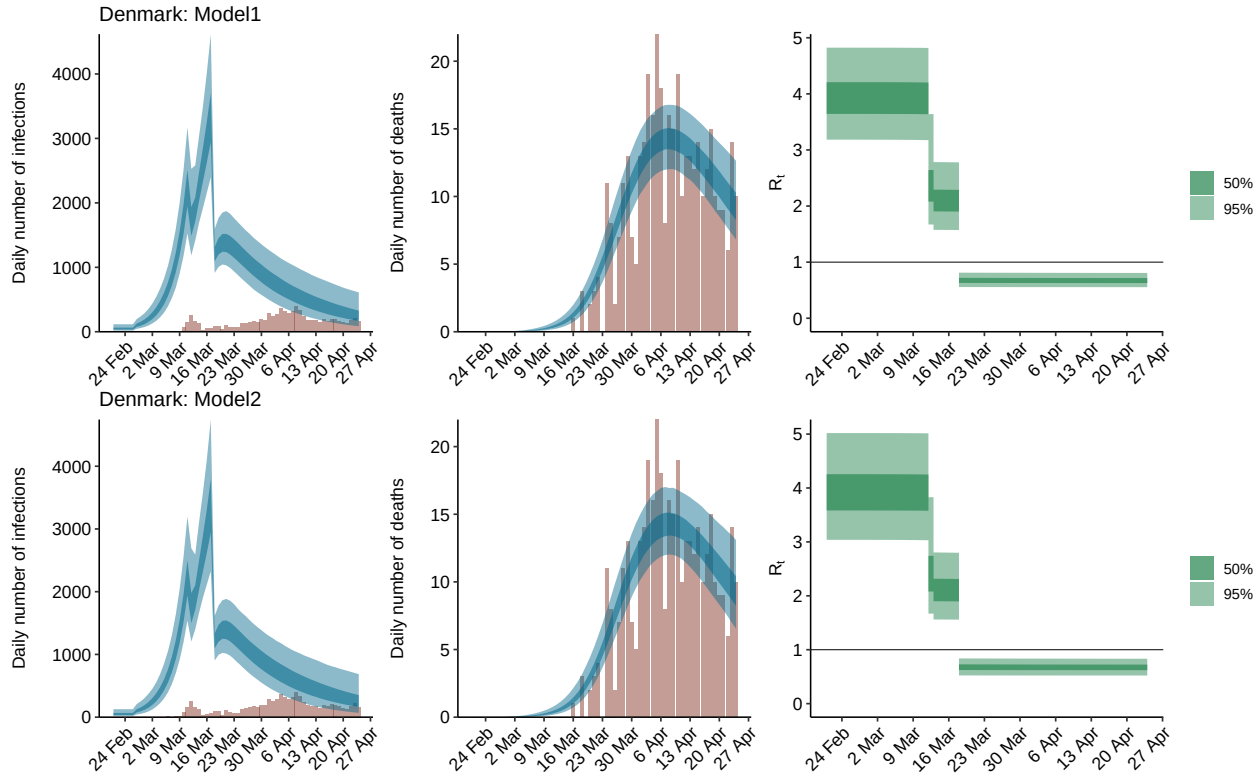

Model3
Denmark
Italy
Germany
Spain
United_Kingdom
France
Norway
Belgium
Austria
Sweden
Switzerland
Greece
Portugal
Netherlands
Finland
2.5
5.0
7.5
10.0

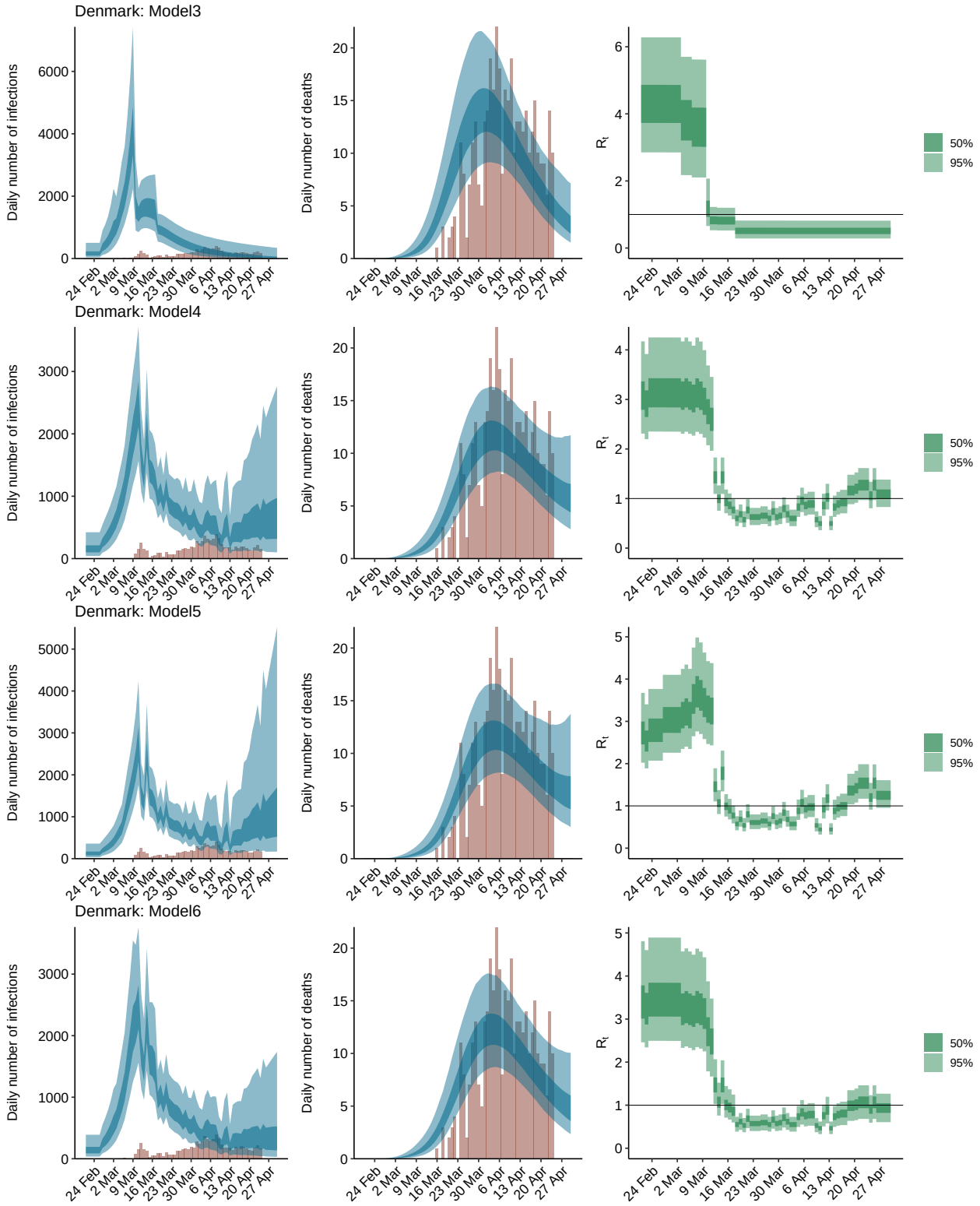

Model4
Denmark
Italy
Germany
Spain
United_Kingdom
France
Norway
Belgium
Austria
Sweden
Switzerland
Greece
Portugal
Netherlands
Finland
2
3
4
5
6

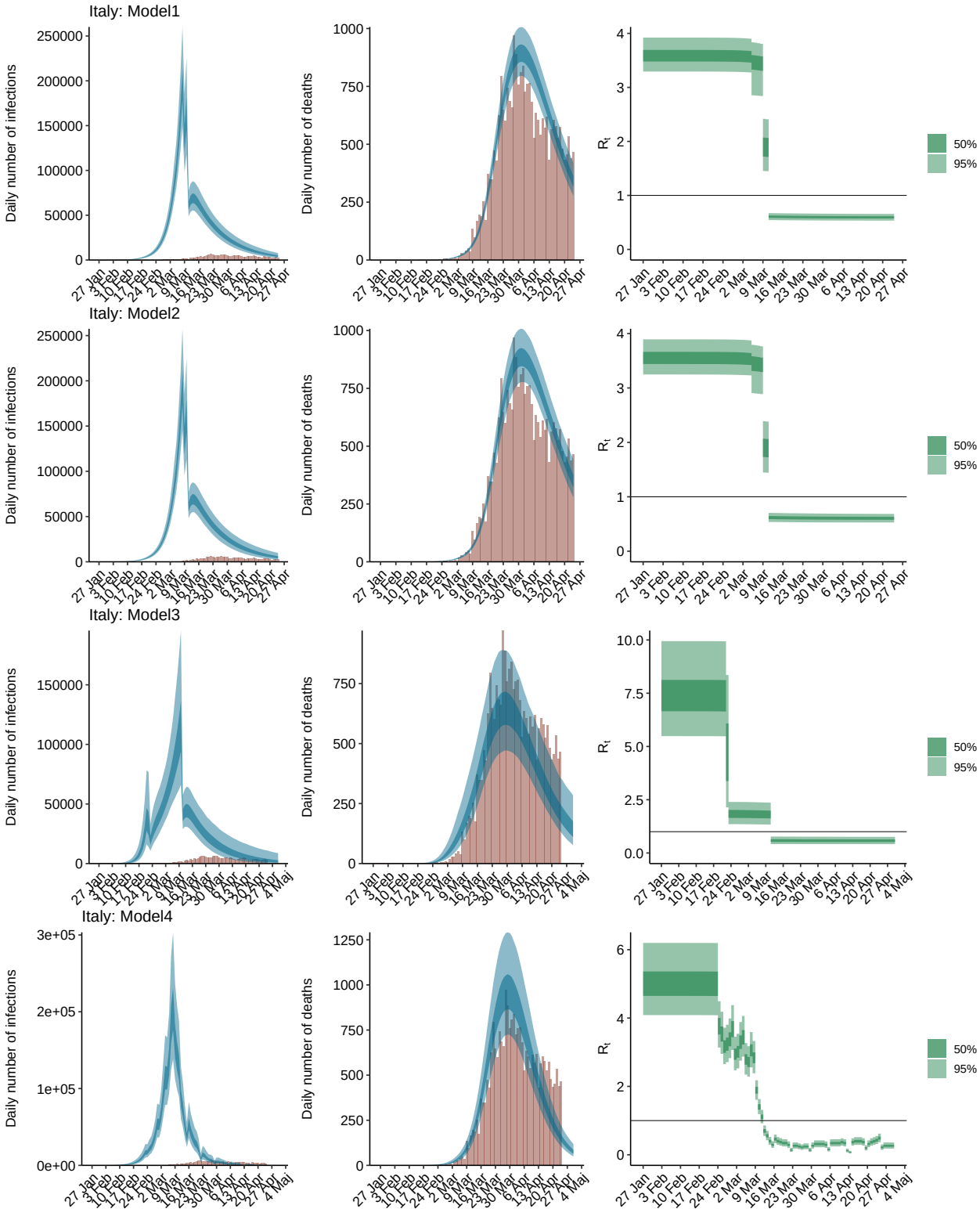

Model5
Denmark
Italy
Germany
Spain
United_Kingdom
France
Norway
Belgium
Austria
Sweden
Switzerland
Greece
Portugal
Netherlands
Finland
2
3
4
5
6

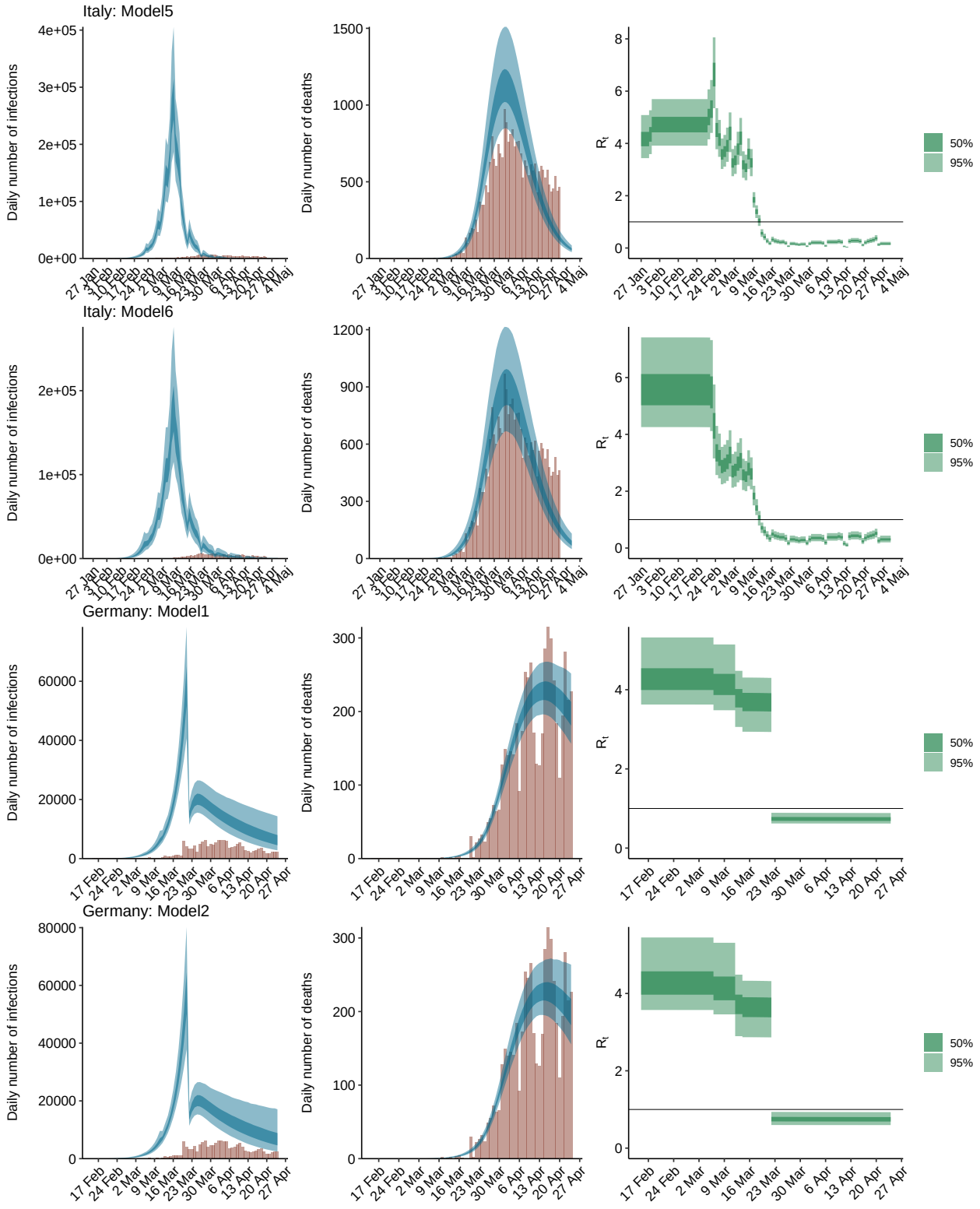


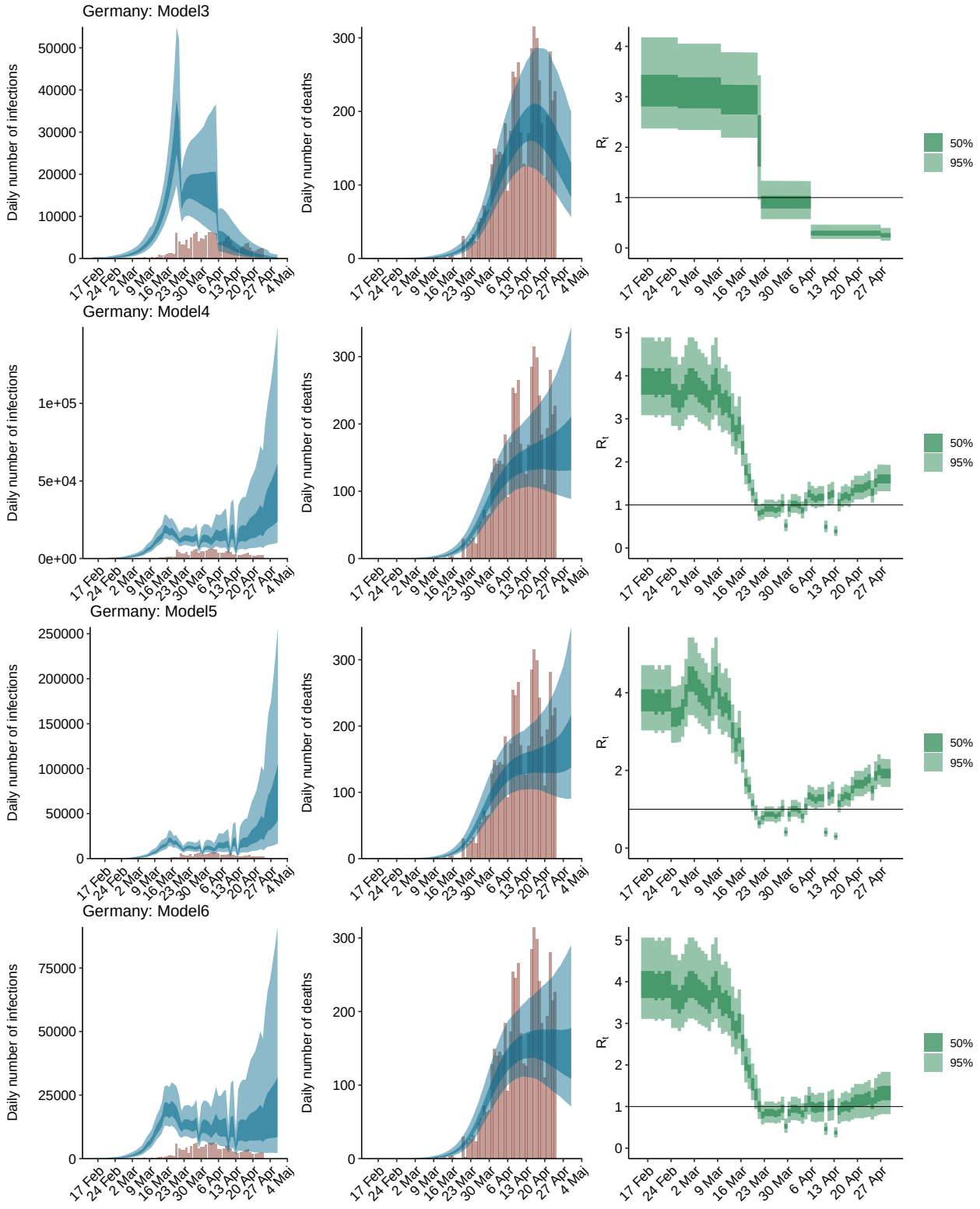
Infections, deaths and Rt by country and model

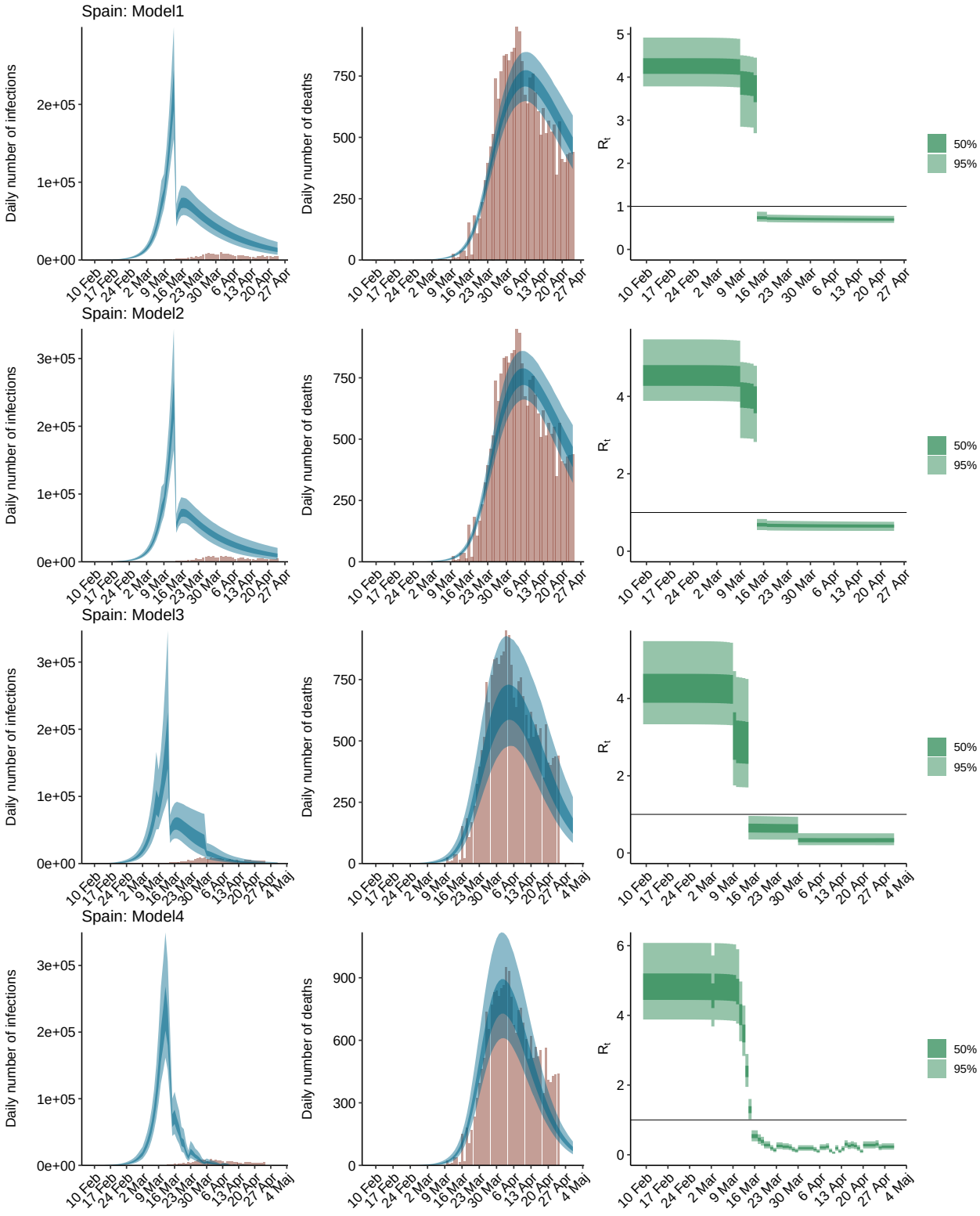


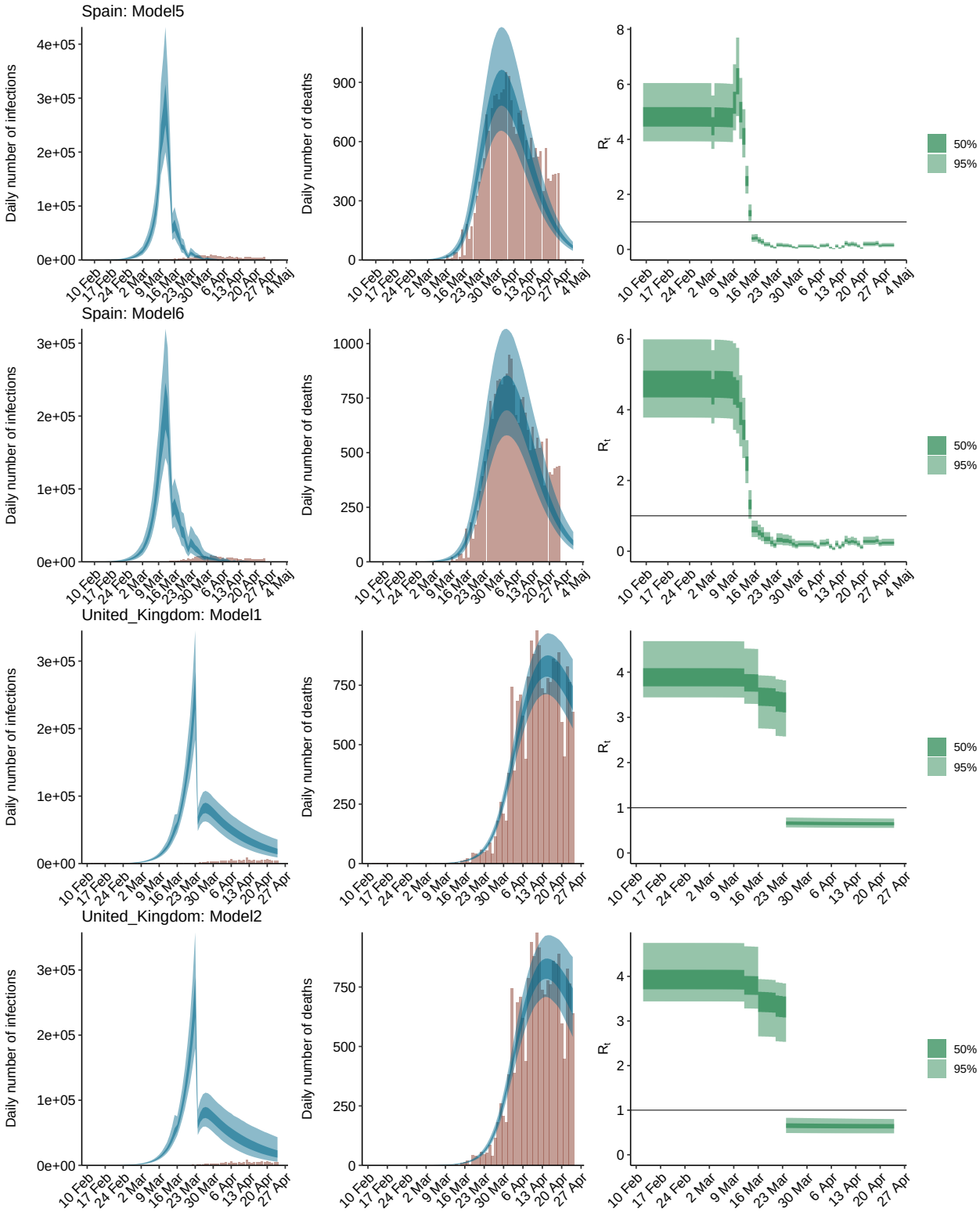

Denmark: Model3
Denmark: Model4
Denmark: Model5
Denmark: Model6
Daily number of infections
Daily number of deaths
R_t
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr
50%
95%

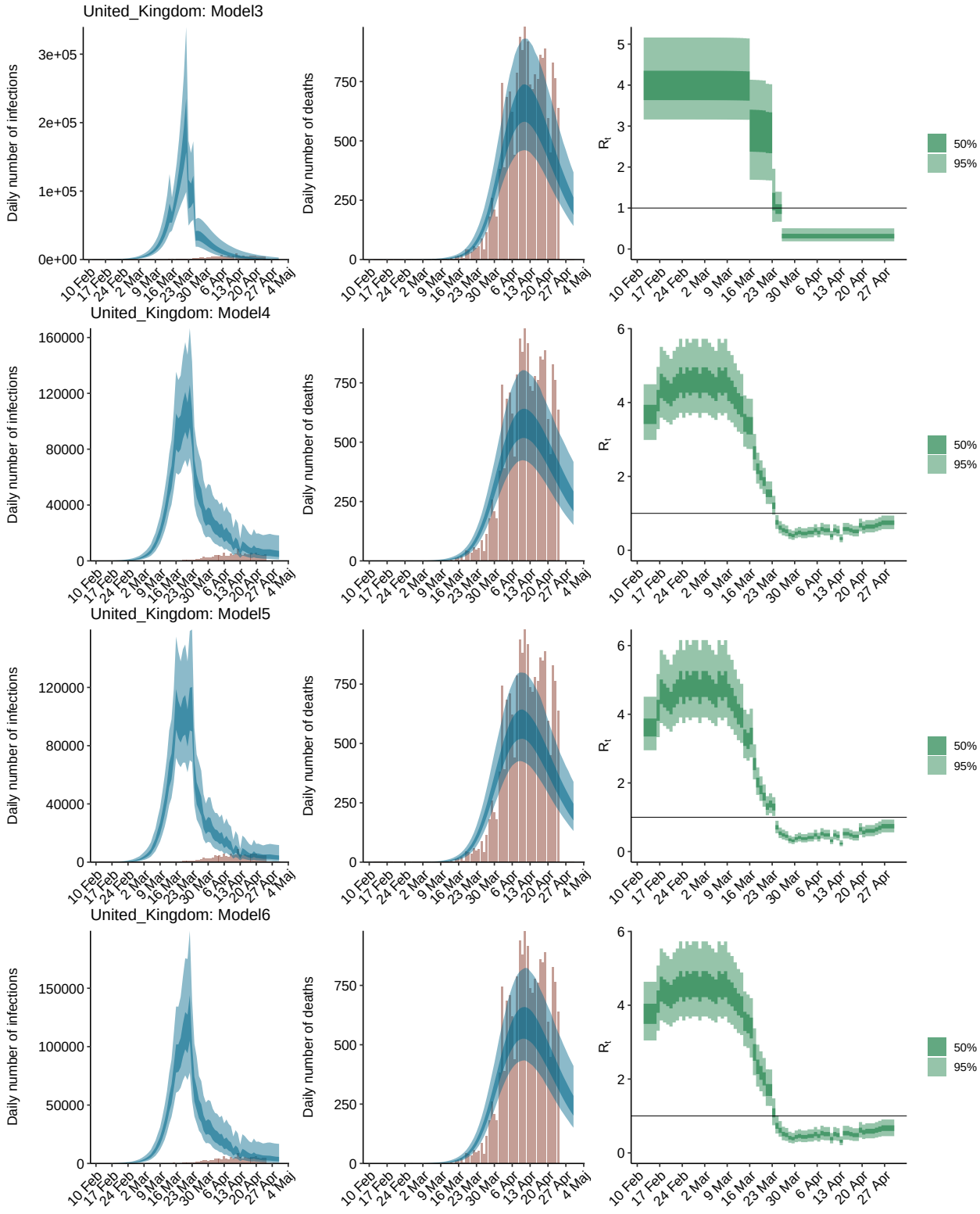

Italy: Model1
Italy: Model2
Italy: Model3
Italy: Model4
Daily number of infections
Daily number of deaths
R_t
50%
95%

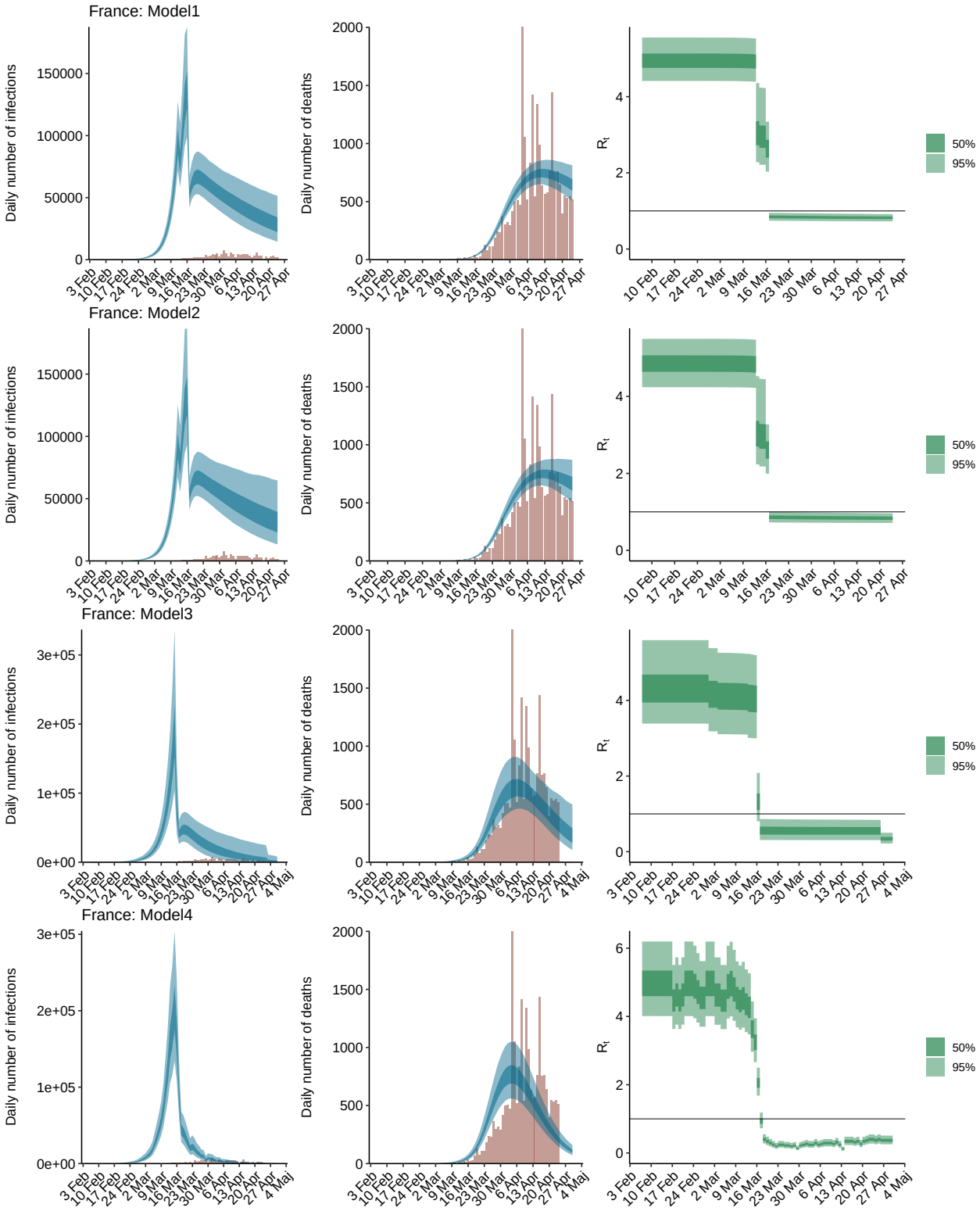

Italy: Model5
Daily number of infections
Daily number of deaths
R_t
Italy: Model6
Germany: Model1
Germany: Model2
50%
95%

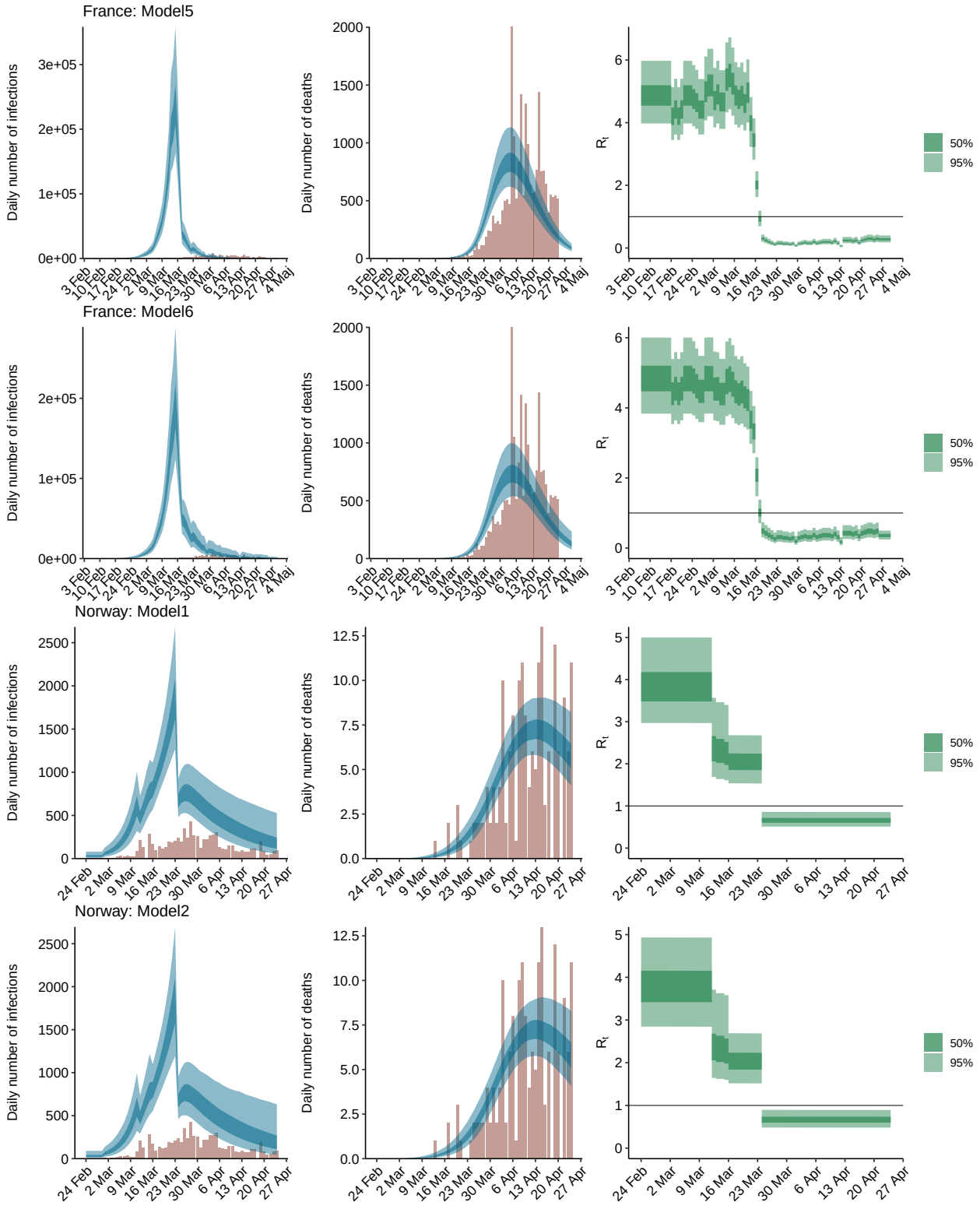

Germany: Model3
Germany: Model4
Germany: Model5
Germany: Model6
Daily number of infections
Daily number of deaths
R_t
50%
95%
17 Feb
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr
4 Maj

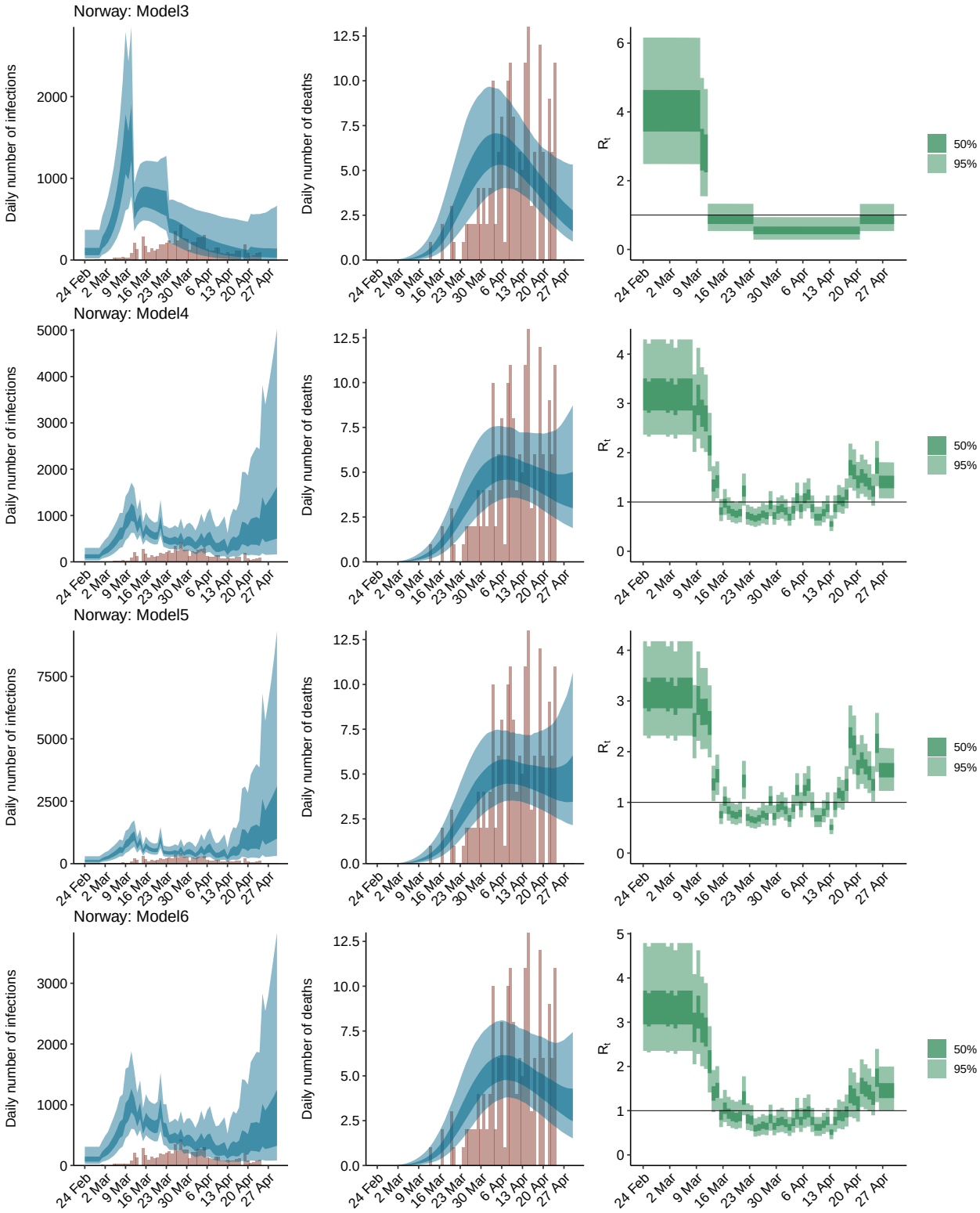

Spain: Model1
Spain: Model2
Spain: Model3
Spain: Model4
Daily number of infections
Daily number of deaths
R_t
50%
95%

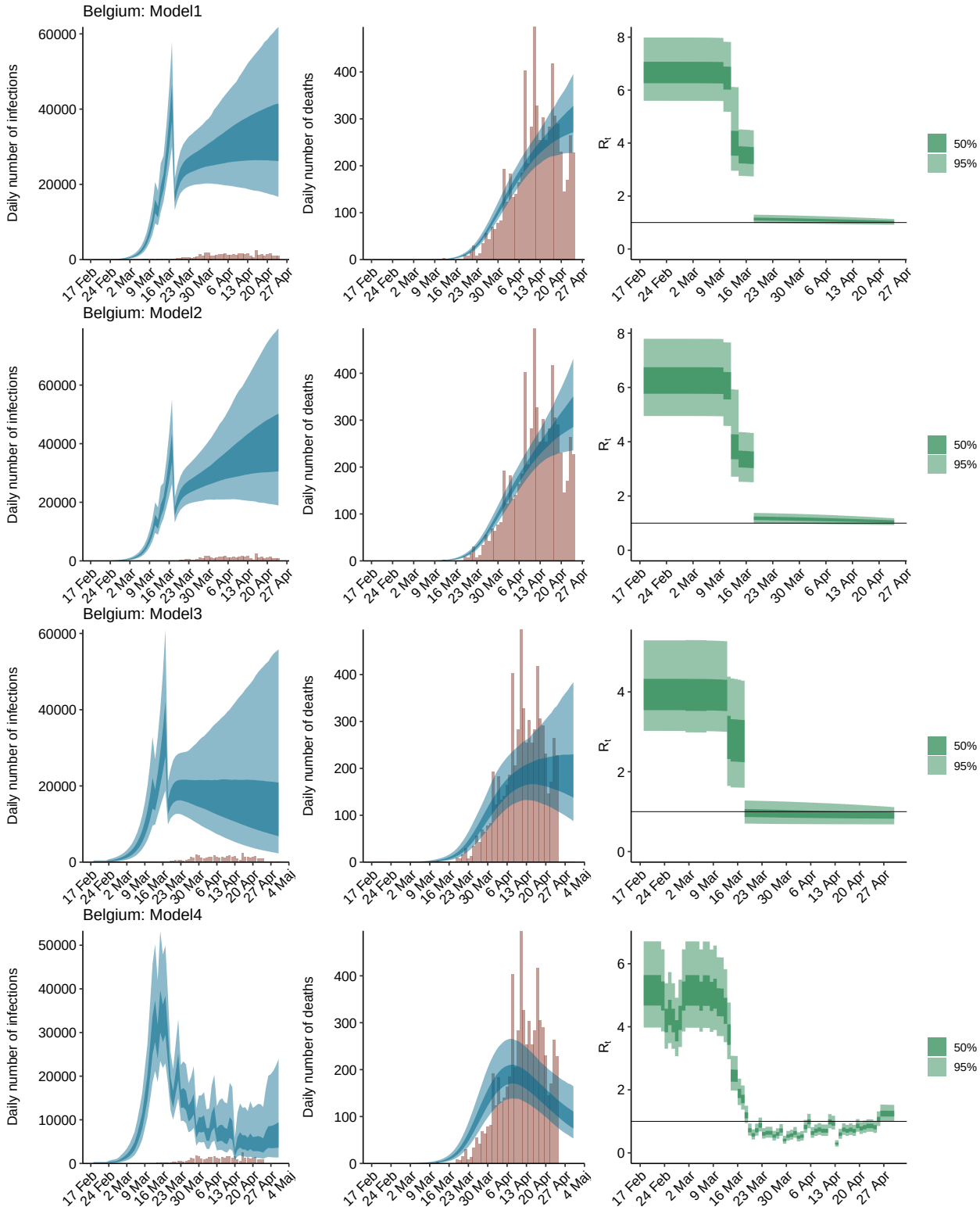

Spain: Model5
Spain: Model6
United_Kingdom: Model1
United_Kingdom: Model2
Daily number of infections
Daily number of deaths
R_t
50%
95%

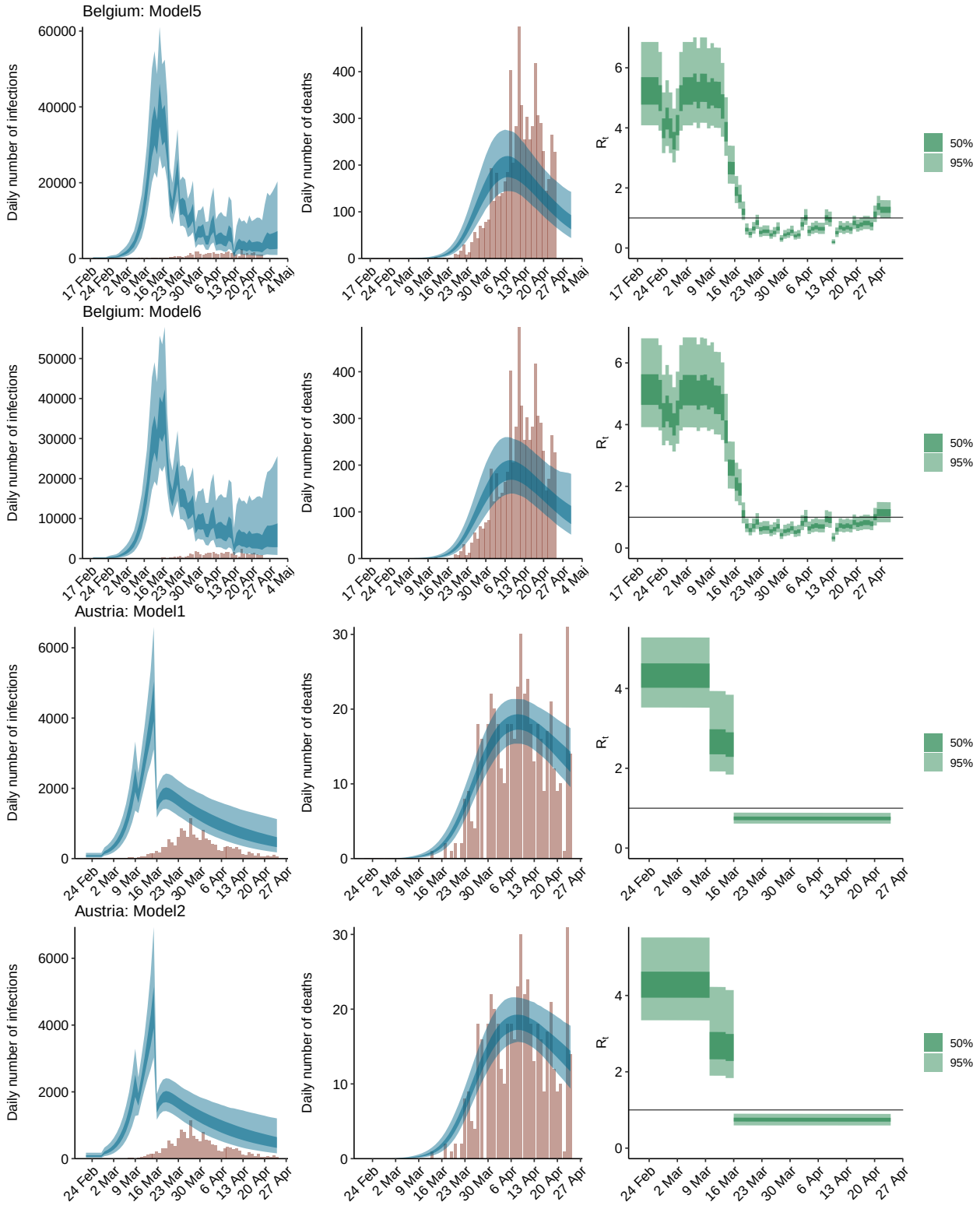

United_Kingdom: Model3
United_Kingdom: Model4
United_Kingdom: Model5
United_Kingdom: Model6
Daily number of infections
Daily number of deaths
R_t
50%
95%

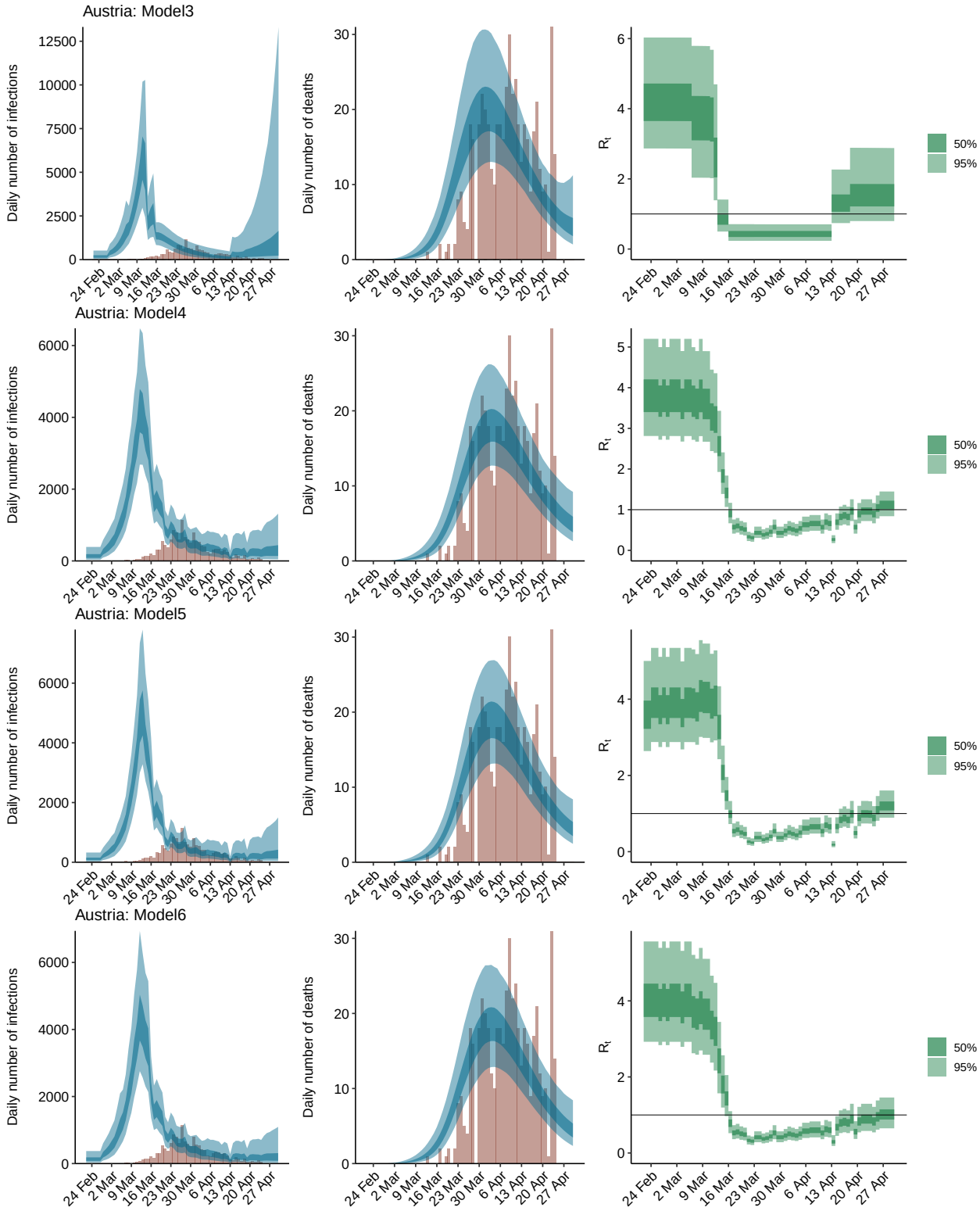

France: Model1
France: Model2
France: Model3
France: Model4
Daily number of infections
Daily number of deaths
R_t
50%
95%

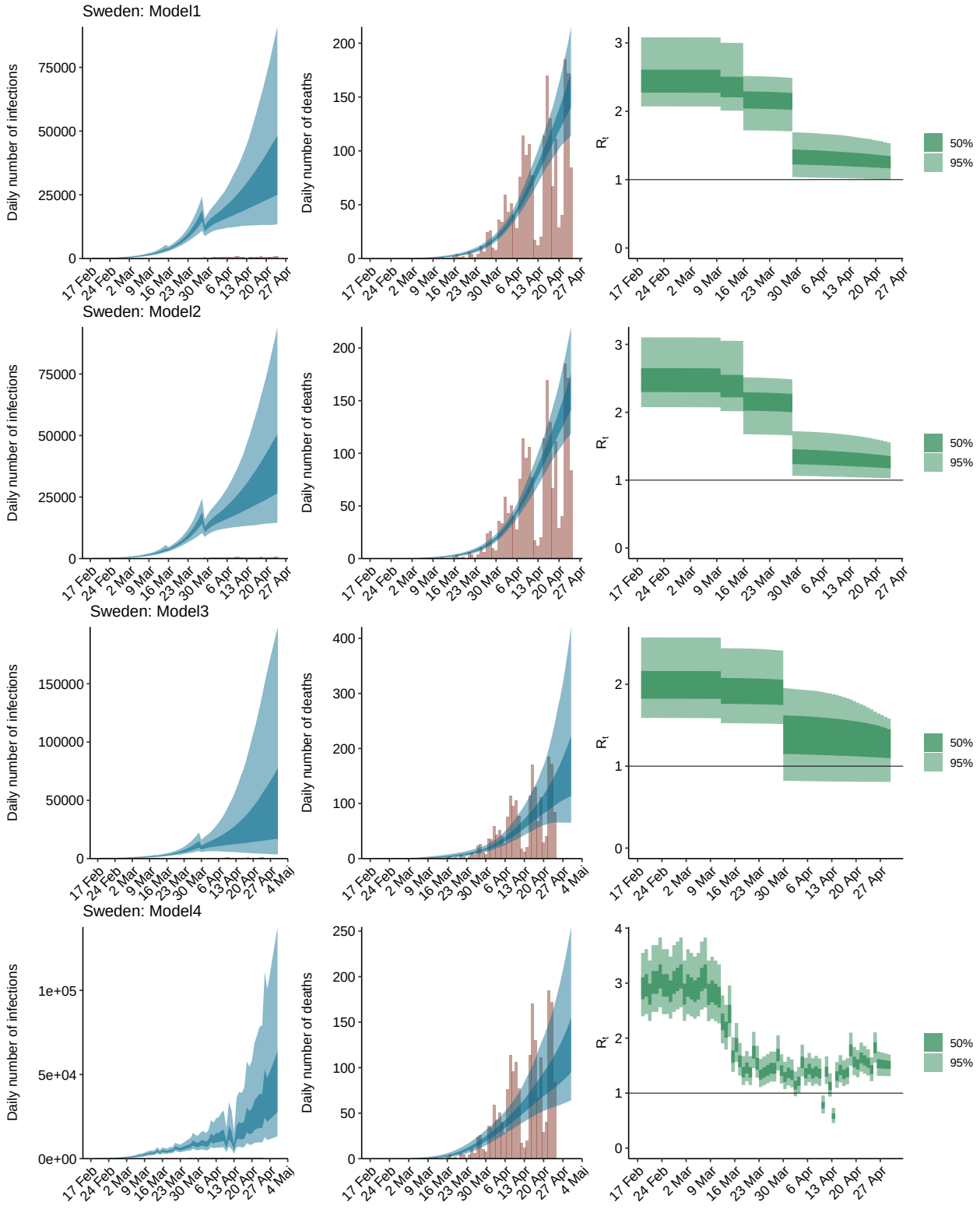

France: Model5
France: Model6
Norway: Model1
Norway: Model2
Daily number of infections
Daily number of deaths
R_t
50%
95%

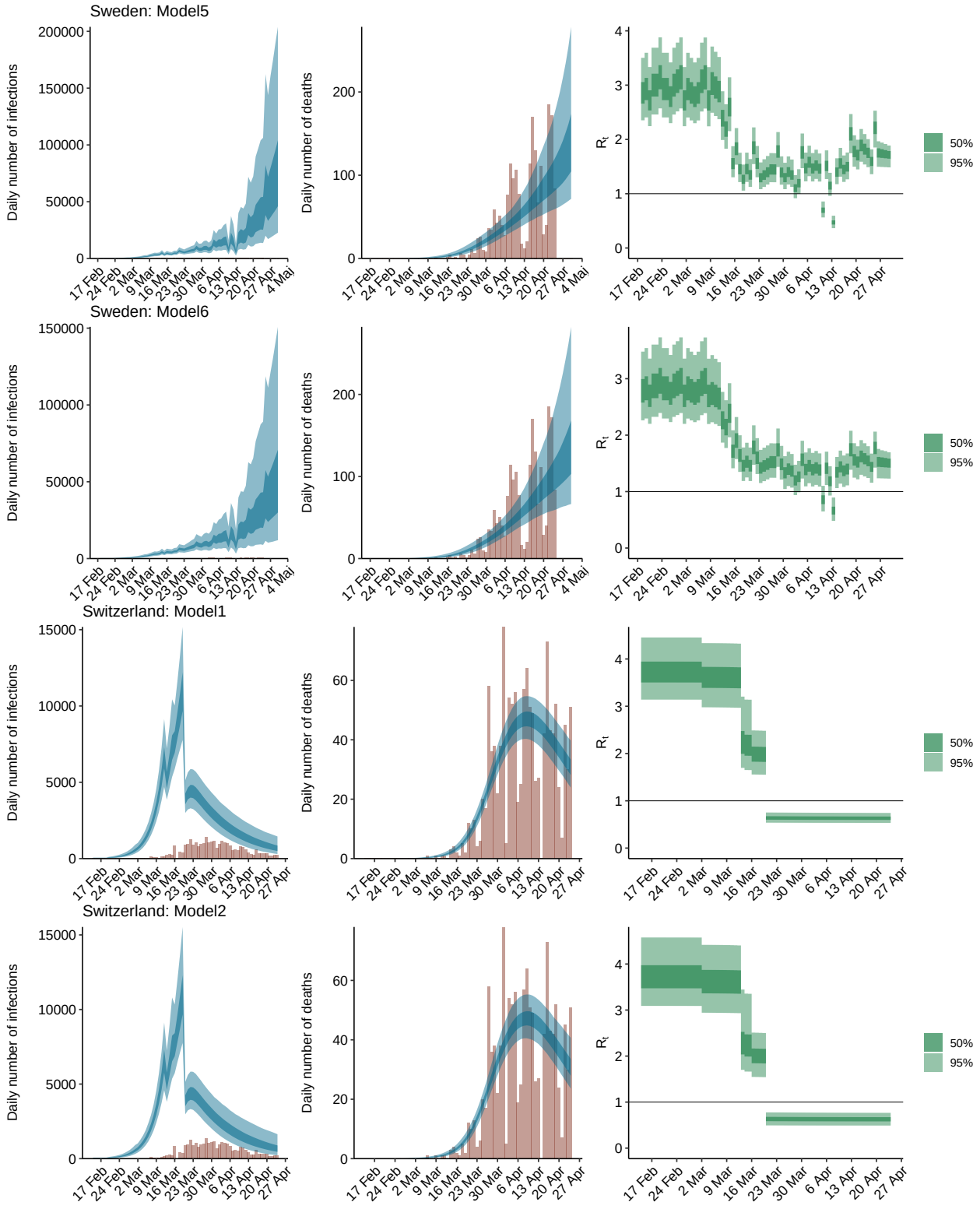

Norway: Model3
Norway: Model4
Norway: Model5
Norway: Model6
Daily number of infections
Daily number of deaths
R_t
50%
95%
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr

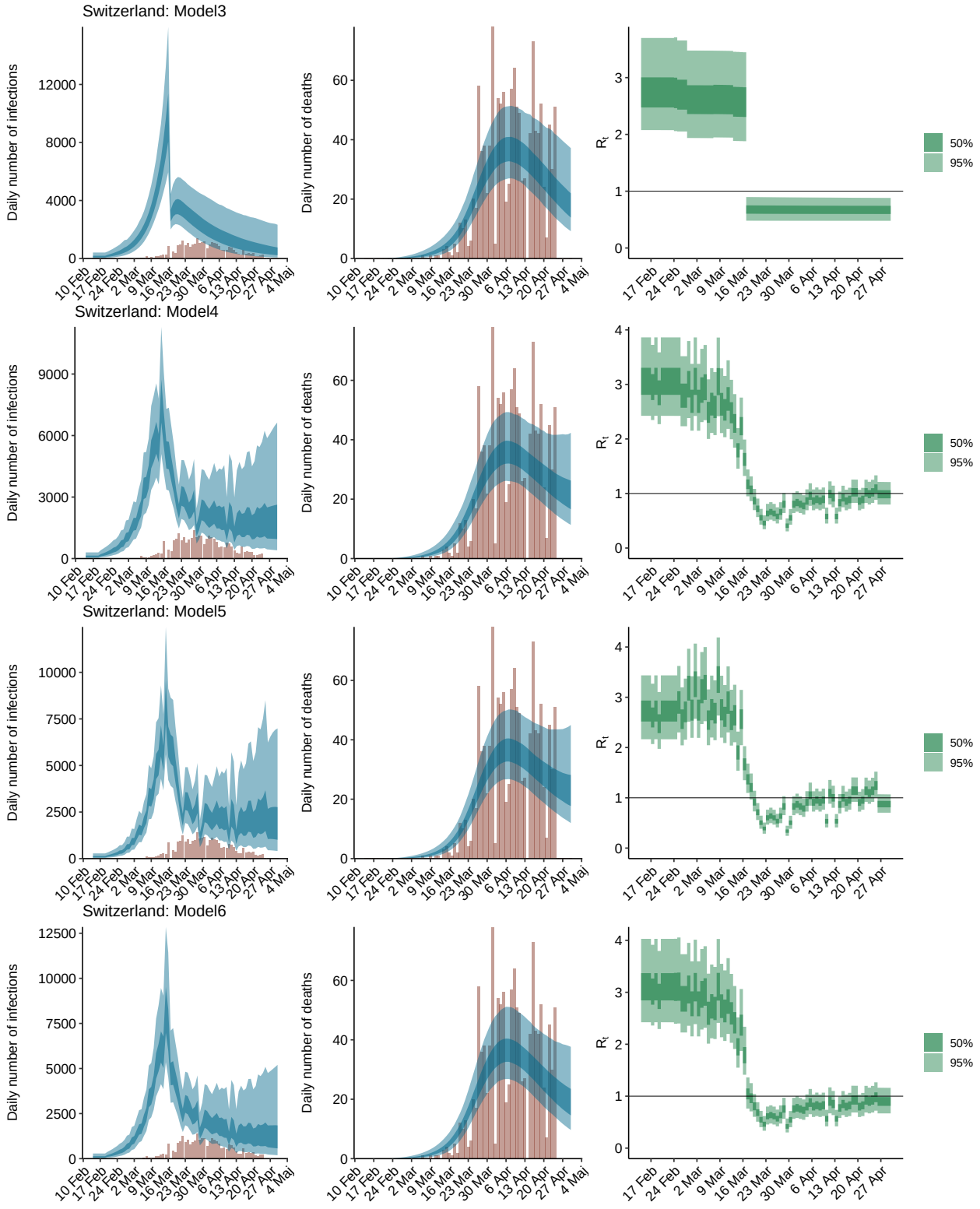

Belgium: Model1
Belgium: Model2
Belgium: Model3
Belgium: Model4
Daily number of infections
Daily number of deaths
R_t
50%
95%
17 Feb
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr
4 Mai

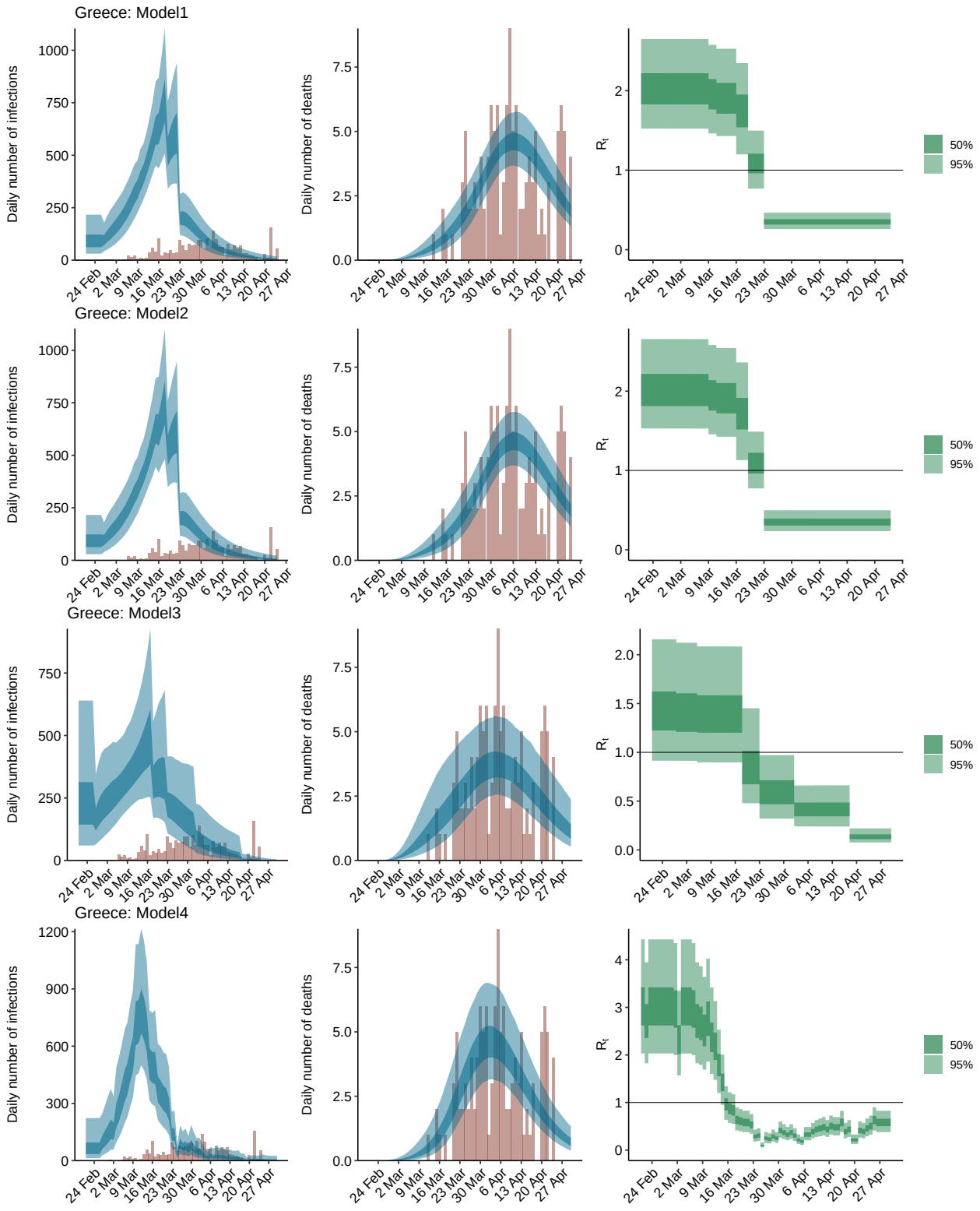

Belgium: Model5
Belgium: Model6
Austria: Model1
Austria: Model2
Daily number of infections
Daily number of deaths
R_t
50%
95%

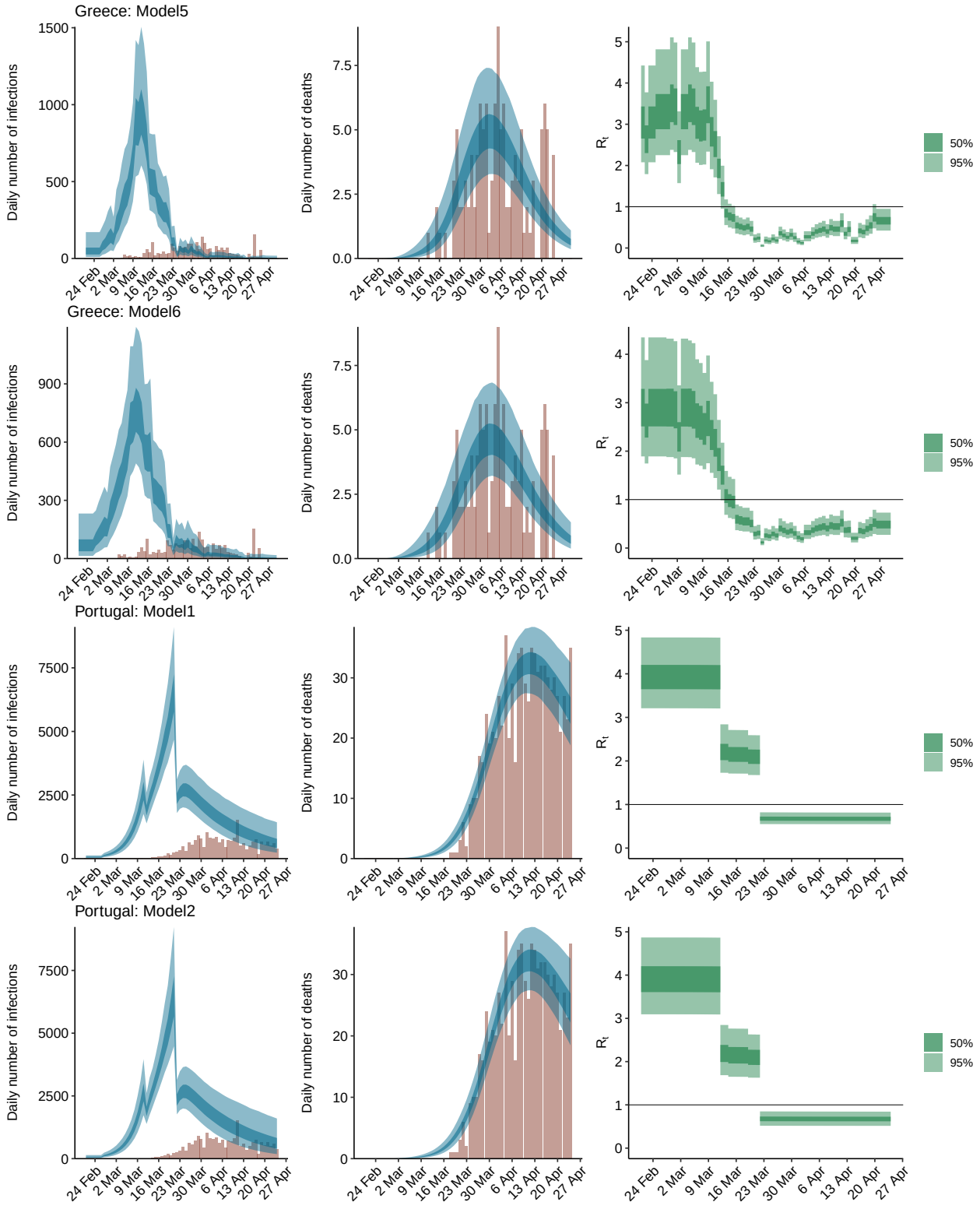

Austria: Model3
Austria: Model4
Austria: Model5
Austria: Model6
Daily number of infections
Daily number of deaths
R_t
50%
95%
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr

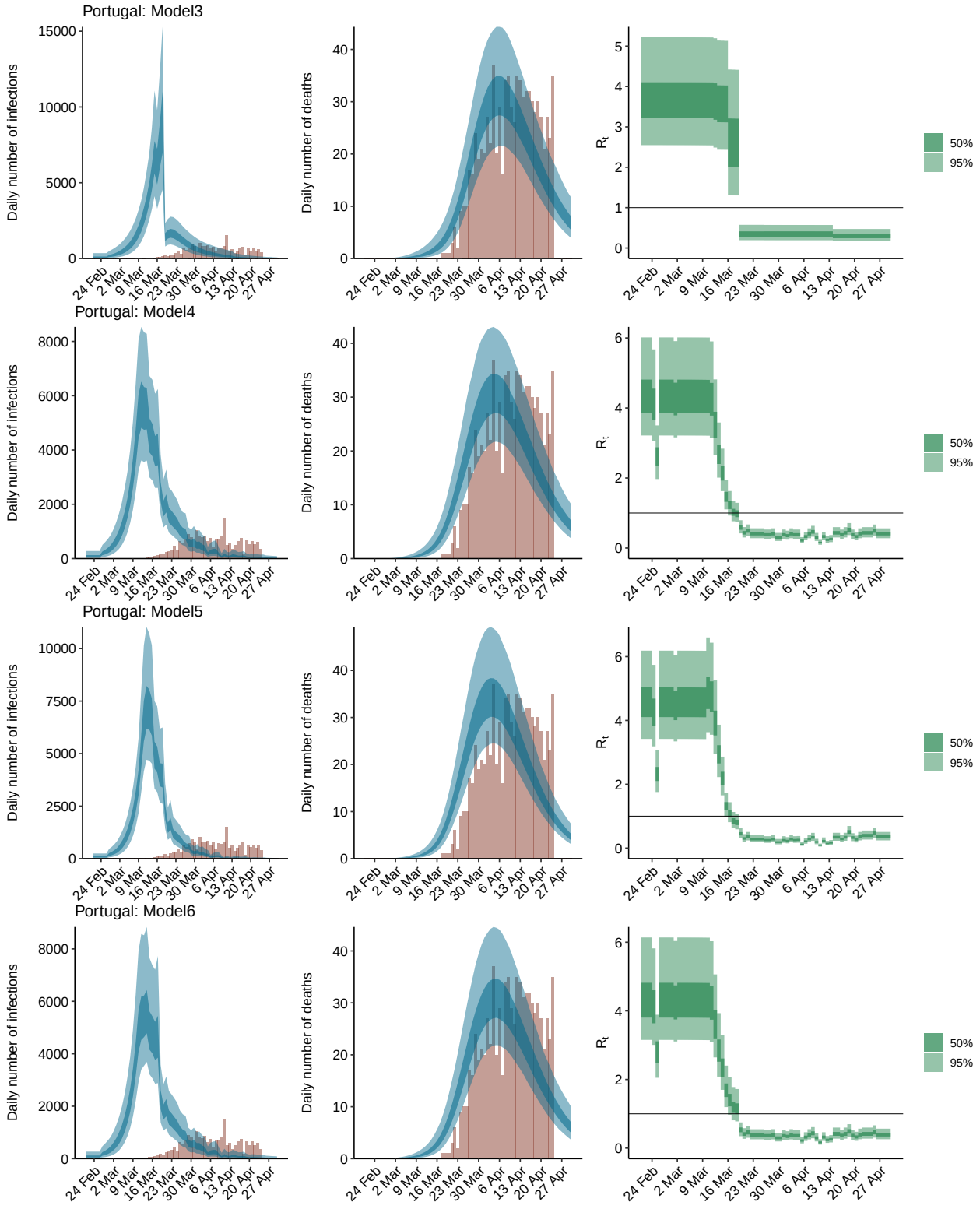

Sweden: Model1
Sweden: Model2
Sweden: Model3
Sweden: Model4
Daily number of infections
Daily number of deaths
R_t
50%
95%
17 Feb
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr
4 Mai

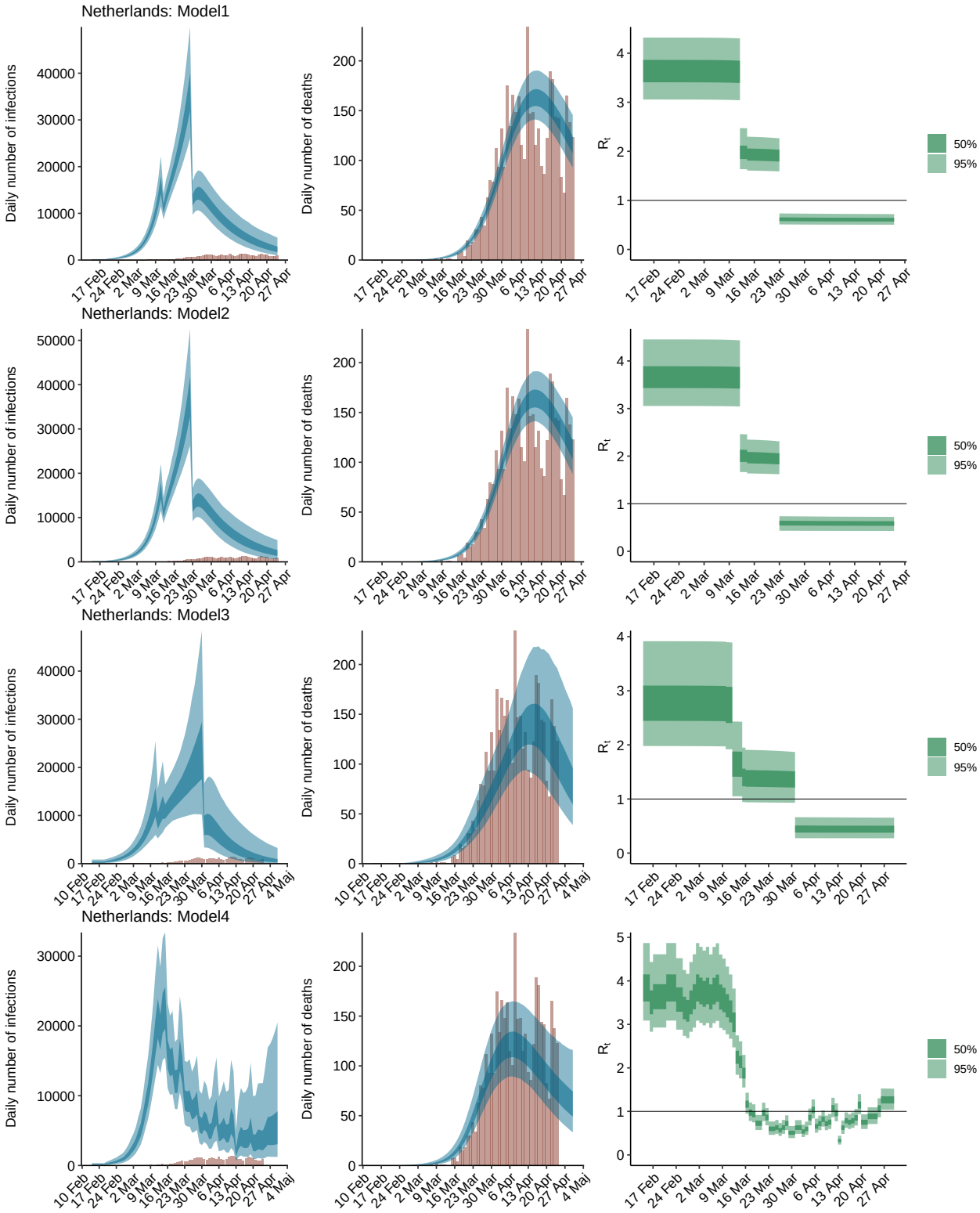

Sweden: Model5
Sweden: Model6
Switzerland: Model1
Switzerland: Model2
Daily number of infections
Daily number of deaths
R_t
50%
95%
17 Feb
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr
4 Mai

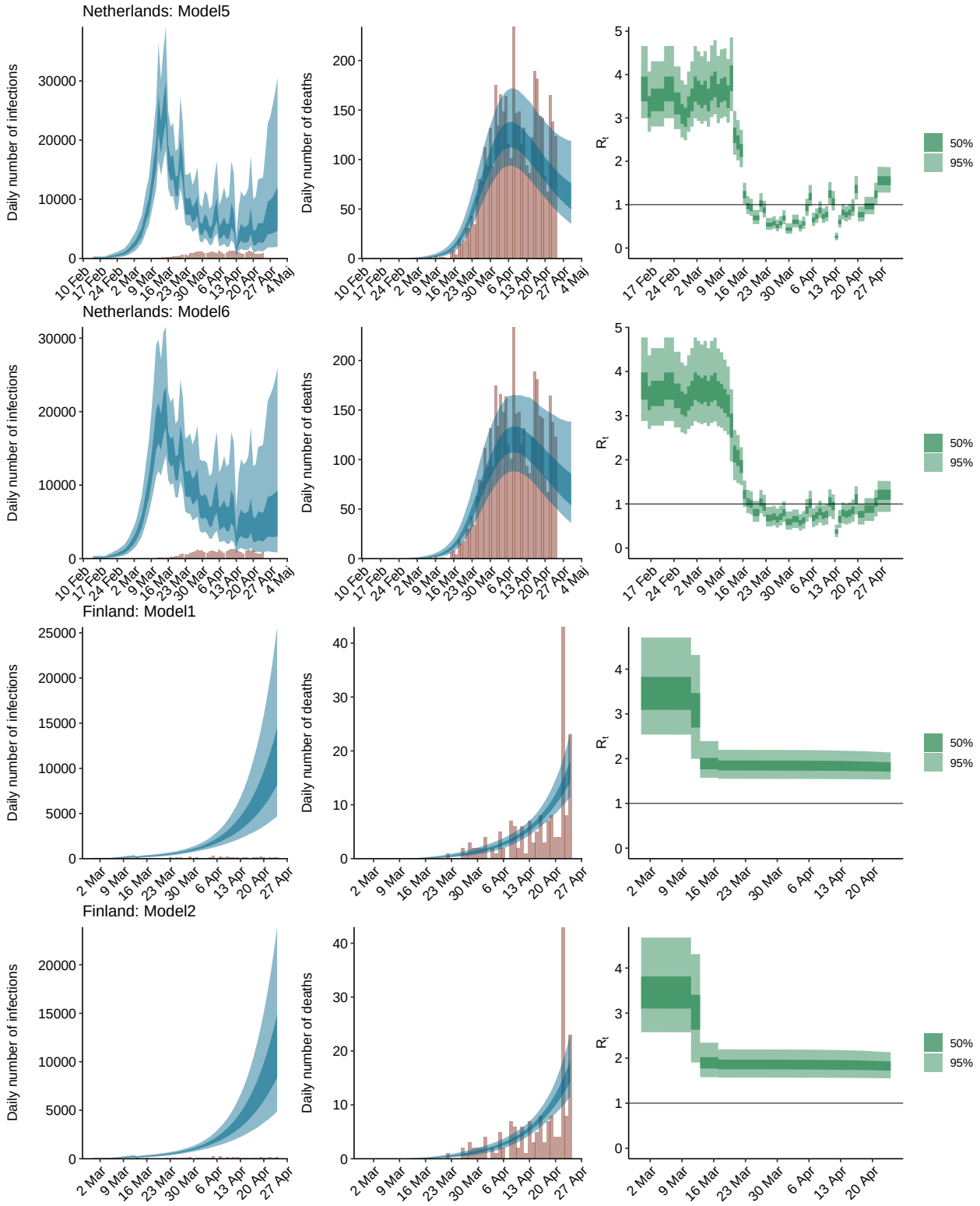

Switzerland: Model3
Switzerland: Model4
Switzerland: Model5
Switzerland: Model6
Daily number of infections
Daily number of deaths
R_t
50%
95%
10 Feb
17 Feb
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr
4 Maj

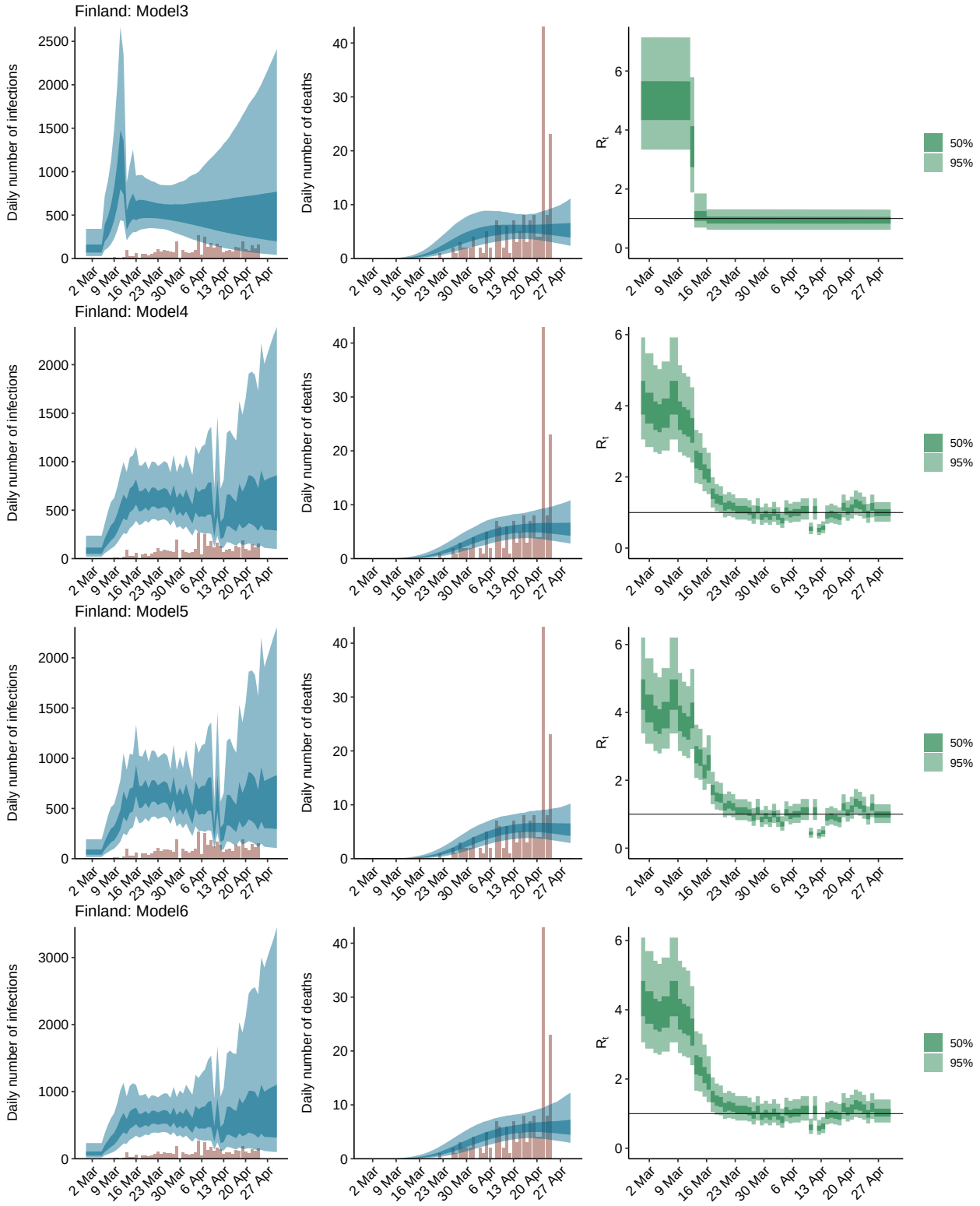

Greece: Model1
Greece: Model2
Greece: Model3
Greece: Model4
Daily number of infections
Daily number of deaths
R_t
50%
95%
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr


Greece: Model5
Daily number of infections
Daily number of deaths
R_t
Greece: Model6
Portugal: Model1
Portugal: Model2
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr
50%
95%


Portugal: Model3
Portugal: Model4
Portugal: Model5
Portugal: Model6
Daily number of infections
Daily number of deaths
R_t
50%
95%
24 Feb
2 Mar
9 Mar
16 Mar
23 Mar
30 Mar
6 Apr
13 Apr
20 Apr
27 Apr


Netherlands: Model1
Daily number of infections
Daily number of deaths
R_t
50%
95%
Netherlands: Model2
Daily number of infections
Daily number of deaths
R_t
50%
95%
Netherlands: Model3
Daily number of infections
Daily number of deaths
R_t
50%
95%
Netherlands: Model4
Daily number of infections
Daily number of deaths
R_t
50%
95%


Netherlands: Model5
Daily number of infections
Daily number of deaths
R_t
50%
95%
Netherlands: Model6
Daily number of infections
Daily number of deaths
R_t
50%
95%
Finland: Model1
Daily number of infections
Daily number of deaths
R_t
50%
95%
Finland: Model2
Daily number of infections
Daily number of deaths
R_t
50%
95%



Stan code

```
## S4 class stanmodel 'base_general_speed' coded as follows:
## data {
```

```
##   int <lower=1> M; // number of countries
##   int <lower=1> P; // number of covariates
##   int <lower=1> N0; // number of days for which to impute infections
##   int<lower=1> N[M]; // days of observed data for country m. each entry must be <= N2
##   int<lower=1> N2; // days of observed data + # of days to forecast
##   int cases[N2,M]; // reported cases
##   int deaths[N2, M]; // reported deaths -- the rows with i > N contain -1 and should be ignored
##   matrix[N2, M] f; // h * s
##   matrix[N2, P] X[M];
##   int EpidemicStart[M];
##   real pop[M];
##   real SI[N2]; // fixed pre-calculated SI using emprical data from Neil
## }
##
## transformed data {
##   vector[N2] SI_rev; // SI in reverse order
##   vector[N2] f_rev[M]; // f in reversed order
##
##   for(i in 1:N2)
##     SI_rev[i] = SI[N2-i+1];
##
##   for(m in 1:M){
##     for(i in 1:N2) {
##      f_rev[m, i] = f[N2-i+1,m];
##     }
##   }
## }
##
##
## parameters {
##   real<lower=0> mu[M]; // intercept for Rt
##   real<lower=0> alpha_hier[P]; // sudo parameter for the hier term for alpha
##   real<lower=0> kappa;
##   real<lower=0> y[M];
##   real<lower=0> phi;
##   real<lower=0> tau;
##   real <lower=0> ifr_noise[M];
## }
##
## transformed parameters {
##     vector[P] alpha;
##     matrix[N2, M] prediction = rep_matrix(0,N2,M);
##     matrix[N2, M] E_deaths  = rep_matrix(0,N2,M);
##     matrix[N2, M] Rt = rep_matrix(0,N2,M);
##     matrix[N2, M] Rt_adj = Rt;
##
##     {
##       matrix[N2,M] cumm_sum = rep_matrix(0,N2,M);
##       for(i in 1:P){
##         alpha[i] = alpha_hier[i] - ( log(1.05) / 6.0 );
##       }
##       for (m in 1:M){
##         /*
##         for (i in 2:N0){
```

```
##           cumm_sum[i,m] = cumm_sum[i-1,m] + y[m];
##         }
##         */
##         prediction[1:N0,m] = rep_vector(y[m],N0); // learn the number of cases in the first N0 days
##         cumm_sum[2:N0,m] = cumulative_sum(prediction[2:N0,m]);
##
##         Rt[,m] = mu[m] * exp(-X[m] * alpha);
##           Rt_adj[1:N0,m] = Rt[1:N0,m];
##         for (i in (N0+1):N2) {
##           /*
##           real convolution=0;
##           for(j in 1:(i-1)) {
##             convolution += prediction[j, m] * SI[i-j];
##           }
##           */
##           real convolution = dot_product(sub_col(prediction, 1, m, i-1), tail(SI_rev, i-1));
##
##           cumm_sum[i,m] = cumm_sum[i-1,m] + prediction[i-1,m];
##           Rt_adj[i,m] = ((pop[m]-cumm_sum[i,m]) / pop[m]) * Rt[i,m];
##           prediction[i, m] = Rt_adj[i,m] * convolution;
##         }
##
##         E_deaths[1, m]= 1e-15 * prediction[1,m];
##         for (i in 2:N2){
##           // for(j in 1:(i-1)){
##           //   E_deaths[i,m] += prediction[j,m] * f[i-j,m] * ifr_noise[m];
##           // }
##           E_deaths[i,m] = ifr_noise[m] * dot_product(sub_col(prediction, 1, m, i-1), tail(f_rev[m], i
##         }
##       }
##     }
## }
## model {
##   tau ~ exponential(0.03);
##   for (m in 1:M){
##       y[m] ~ exponential(1/tau);
##   }
##   phi ~ normal(0,5);
##   kappa ~ normal(0,0.5);
##   mu ~ normal(3.28, kappa); // citation: https://academic.oup.com/jtm/article/27/2/taaa021/5735319
##   alpha_hier ~ gamma(.1667,1);
##   ifr_noise ~ normal(1,0.1);
##   for(m in 1:M){
##     deaths[EpidemicStart[m]:N[m], m] ~ neg_binomial_2(E_deaths[EpidemicStart[m]:N[m], m], phi);
##    }
## }
##
## generated quantities {
##     matrix[N2, M] prediction0 = rep_matrix(0,N2,M);
##     matrix[N2, M] E_deaths0  = rep_matrix(0,N2,M);
##
##     {
##       matrix[N2,M] cumm_sum0 = rep_matrix(0,N2,M);
##       for (m in 1:M){
```

```
##          for (i in 2:N0){
##            cumm_sum0[i,m] = cumm_sum0[i-1,m] + y[m];
##          }
##          prediction0[1:N0,m] = rep_vector(y[m],N0);
##          for (i in (N0+1):N2) {
##            real convolution0 = 0;
##            for(j in 1:(i-1)) {
##              convolution0 += prediction0[j, m] * SI[i-j];
##            }
##            cumm_sum0[i,m] = cumm_sum0[i-1,m] + prediction0[i-1,m];
##            prediction0[i, m] =  ((pop[m]-cumm_sum0[i,m]) / pop[m]) * mu[m] * convolution0;
##          }
##
##          E_deaths0[1, m] = uniform_rng(1e-16, 1e-15);
##          for (i in 2:N2){
##            for(j in 1:(i-1)){
##              E_deaths0[i,m] += prediction0[j,m] * f[i-j,m] * ifr_noise[m];
##            }
##          }
##        }
##      }
## }
##
```