Flexible Epdiemic Modeling with epidemia

Contents

Note: *This vignette is very much a work in progress, and will be regularly updated.*

This vignette details the models fit by epidemia, and also demonstrates basic usage of **epidemia**. Most of the work is done in the `epim` function. Before continuing, please read the documentation for a more detailed description of this function.

The document is organized as follows. In Section 1, we provide basic commands to help the user to rapidly get a feel for the package. Section 2 digs deeper, and gives a formal framework for the epidemic models fit by the `epim` function. Understanding this framework is important for making the most of the flexibility of the package. Section 3 discusses the details of the `epim` function, and in particular describes its various arguments. Finally Section 4 attempts to demonstrate some of the more useful features of the package.

To fully understand how `epim` works, and in particular to make the most of its flexibility, it helps to formalize the model framework. This is done in 2.

1 Example Usage

The package contains an example dataset `EuropeCovid`, which contains data on the daily deaths from Covid-19 recorded in 11 European countries. Here we provide code snippets showing basic usage of **epidemia**.

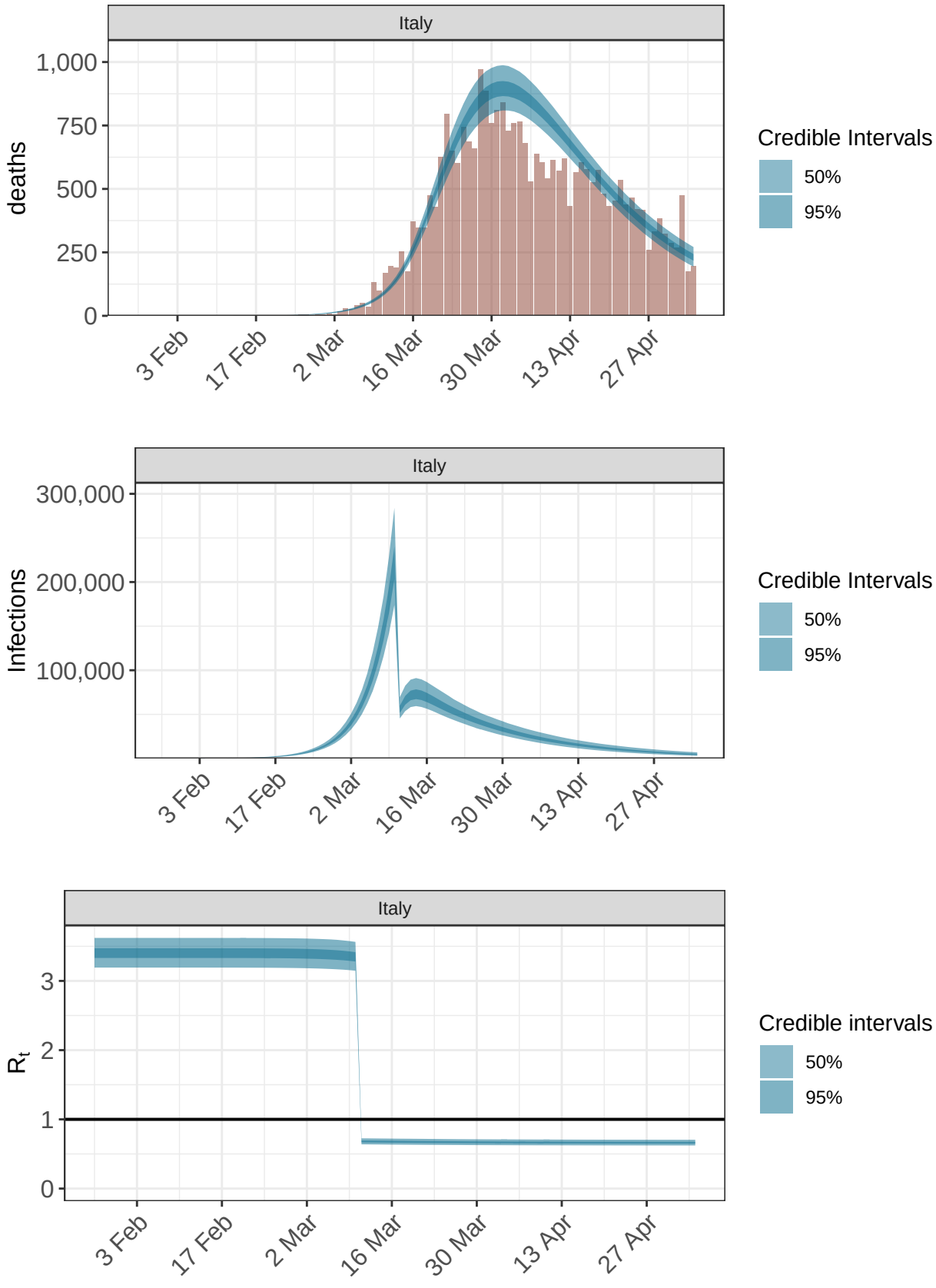
```
library(epidemia)
library(xfun)
data("EuropeCovid")
options(mc.cores = parallel::detectCores())
```

The line `options(mc.cores = parallel::detectCores())` is important if MCMC sampling is used, as it allows different chains to be run in parallel, rather than sequentially.

Below considers fitting a model to the Italian data, and demonstrates usage of various arguments to `epim`.

```
# collect arguments for 'epim'
args <- EuropeCovid
args$algorithm <- "sampling"
args$sampling_args <- list(iter=600,control=list(adapt_delta=0.95,max_treedepth=15),seed
args$group_subset <- c("Italy")
args$formula <- R(country,date) ~  1 + lockdown
args$prior <- rstanarm::normal(location=0,scale=.5)
args$prior_intercept <- rstanarm::normal(location=0,scale=2)
fit <- xfun::cache_rds({do.call("epim", args)})
```

```
library(gridExtra)
grid.arrange(plot_obs(fit, type="deaths"), plot_infections(fit), plot_rt(fit),nrow=3)
```



Here is an extension to multiple populations, with a group-specific intercept.

```
# collect arguments for 'epim'
args$group_subset <- c("Italy", "United_Kingdom")
args$formula <- R(country,date) ~  (1|country) + lockdown
fit <- xfun::cache_rds({do.call("epim", args)})
```

```
## Warning: extra argument(s) 'prior_intercept' disregarded
```

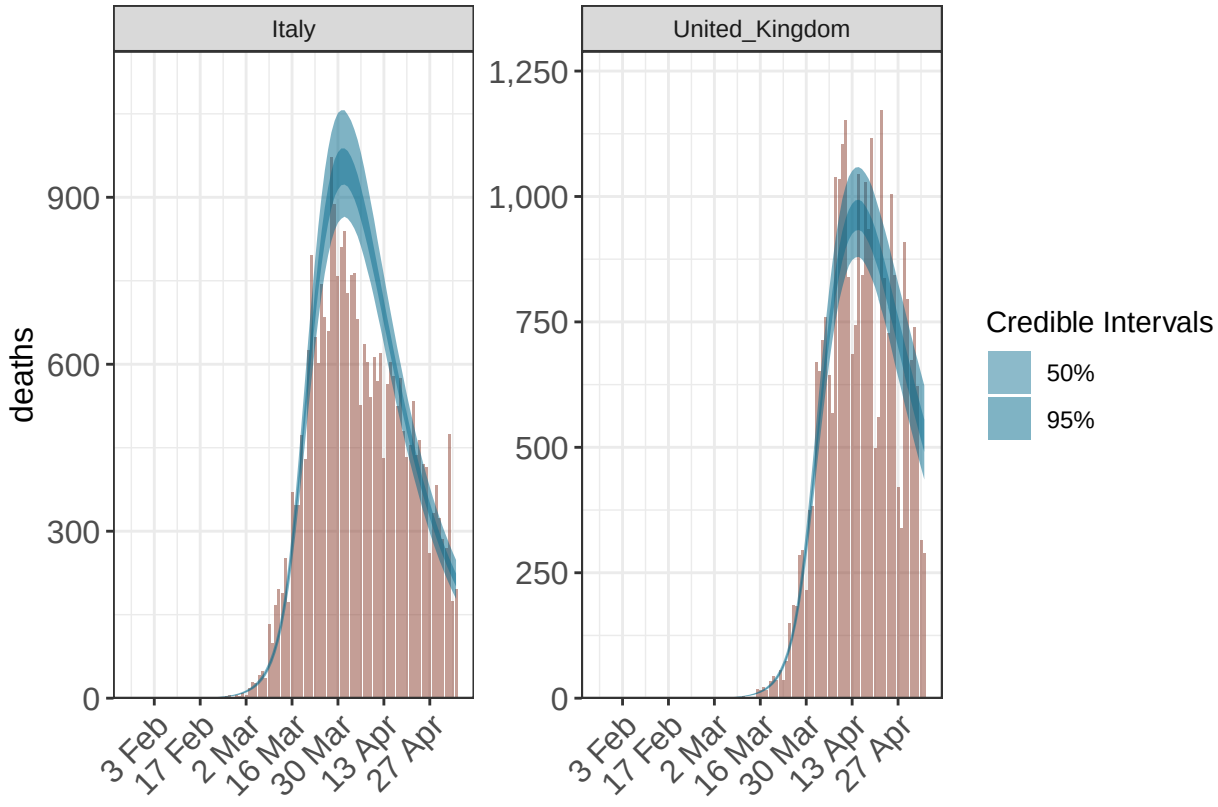
```
## Warning: There were 25 divergent transitions after warmup. Increasing adapt_delta abo
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

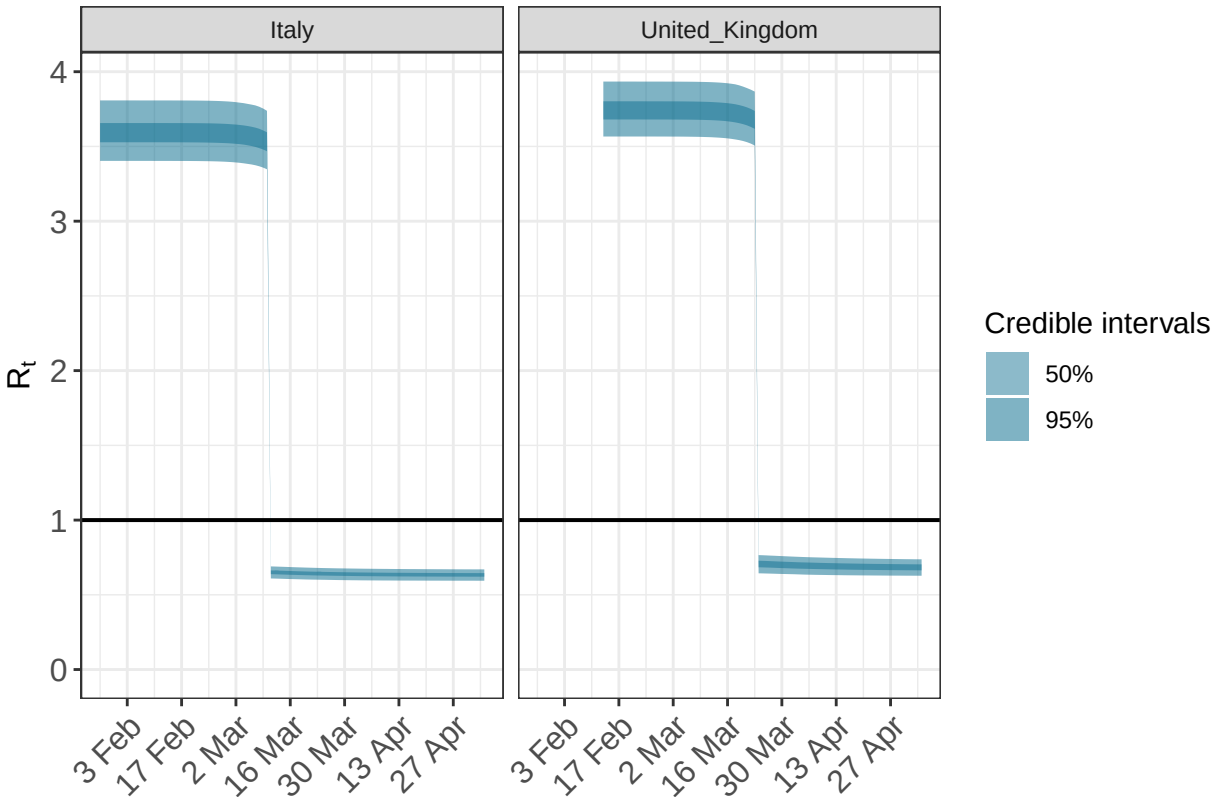
```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
plot_obs(fit, type = "deaths")
```




```
plot_rt(fit)
```



2 Model Overview

Flaxman et al. (2020) introduced a hierarchical Bayesian approach for epidemic modeling, and applied it to assessing the effect of non-pharmaceutical interventions on the covid-19 pandemic in 11 European countries. **epidemia** is designed to fit models which are largely extensions of this approach. There are however important differences which will be emphasized as we go along. The model is first described as it applies to a single population. Extensions to multiple populations are considered in the subsequent section.

2.1 A Single Population

Let t_0 be an integer representing the first date at which infections are non-zero in the population, and let T be the number of consecutive days over which to simulate the epidemic.

The basic idea behind the model is that the observed quantities (daily deaths, incidence rates etc.) are a function of the latent infections in the population over time. These infections are in turn a function of the *time-varying reproduction number*. This number may be explained by a number of covariates; for example non-pharmaceutical interventions, or changes in mobility over time.

The remainder of this section is organized as follows. First, the model for the sequence of

time-varying reproduction numbers is described. Infections are then modeled in terms of this sequence. Finally, the observed data is modeled as a function of these latent infections.

2.1.1 Time-varying reproduction number

Let $R := (R_0, \ldots, R_T)$ be a non-negative sequence representing the *unadjusted* time-varying reproduction number over the period considered. Intuitively, R_t measures the intensity of infectious interactions occuring in the population at time $t_0 + t$. The term *unadjusted* is used to make explicit that the sequence being modeled has not been adjusted for the size of the susceptible population. This adjustment is made later in the model for infections. The sequence is parameterized as

$$R = 2Af\left(X\beta + Zb\right) \tag{1}$$

where A is a constant, X and Z are $T \times p$ and $T \times q$ model matrices and where f is the logistic function $f(a) := \exp(a)/(1 + \exp(a))$. The p-dimensional β represents pooled coefficients, while the q-dimensional b represent group-specific effects. When both p and q are greater than zero, this is known as *partial pooling*.

The vector R is latent and unobserved. The quantities we do observe are linked to R through its effect on the underlying infections, which we now describe.

2.1.2 Infections

Let $I_0, \ldots I_T$ be a sequence where I_t is the count of infections that have occurred by date $t_0 + t$. This is modeled as the *exact* solution to an ordinary differential equation denoted by $I(t)$. Before describing this ODE, first define $s := (s_1, s_2, \ldots)$, a distribution on $\mathbb{Z}^+$ representing the serial interval of the disease being modeled.

The ODE $I(t)$ satisfies

$$I'(t) = \left(\frac{P - I(t)}{P}\right) R_{\lfloor t \rfloor} c_{\lfloor t \rfloor},$$

with initial condition $I(0) = 0$ and where c_t is a weighted summation over previous infections

$$c_t = \sum_{\tau=1}^{t} (I(\tau) - I(\tau - 1)) s_{t-\tau}.$$

The exact solution to the above ODE at $t = 1, \ldots$ is easily shown to be

$$I_t = I_{t-1} + (P - I_{t-1}) \left(1 - \exp\left(-\frac{R_t c_t}{P}\right)\right),$$

where P is the population size. This satisfies intuitive properties. If $R_t = 0$, then there are no new infections. Fixing $c_t > 0$ and letting $R_t \to \infty$ implies that $I_t \to P$, i.e. everyone is infected tomorrow.

2.1.3 Observations

Data is observed from L observation processes $Y_t^{(1)} \dots Y_t^{(L)}$, with the subscript t ranging between 1 and T. Each process represents a different type of observation. This could be daily death data as used in Flaxman et al. (2020), or hospitalization rates, or recorded infections for example. The random variables $Y_t^{(l)}$ are assumed to follow a negative binomial distribution with mean $y_t^{(l)}$ and variance

$$y_t^{(l)} + \frac{\left(y_t^{(l)}\right)^2}{\phi_l},$$

where $\phi_l \sim \mathcal{N}^+(0, \sigma_{\phi_l}^2)$ for some hyperparameter $\sigma_{\phi_l}^2 > 0$. The expected value $y_t^{(l)}$ is modeled as a function of

- $I_{t_0}, \dots, I_{t-1}$, where $I_{t'}$ is the count of infections that have occurred by period t',
- $\pi^{(l)} = (\pi_1^{(l)}, \pi_2^{(l)}, \dots)$, a discrete distribution on $\mathbb{Z}^+$ representing the time distribution from an infection event to an observation event,
- and a proportion α_l, assumed to be a time constant quantity representing the rate of infections which will eventually record as an observation of type l.

To give intuition on the above quantities, suppose for now that the l^{th} observation process recorded daily death counts. Then $\pi_t^{(l)}$ is the probability, conditional on a death being recorded today, that this individual was infected exactly t days prior. α_l is then the mean infection fatality ratio (IFR) for the given population.

The functional form for $y_t^{(l)}$ is then simply

$$y_t^{(l)} := \alpha_l \sum_{\tau=1}^{t-1} (I_\tau - I_{\tau-1}) \pi_{t-\tau}^{(l)}.$$

The distribution $\pi^{(l)}$ is *treated as known*, however in future versions of epidemia we may allow it to be learnt. One possibility in this direction would be to view $\pi^{(l)}$ as a discretization of a continuous parametric distribution. The parameters could be given priors and learnt. The Gamma distribution is a suitable candidate for this.

The proportion α_l is treated as unknown, and given a normal prior truncated to the unit interval. To be precise

$$\alpha_l \sim \mathcal{N}_{[0,1]}(\mu_l, \sigma_{\alpha_l}^2),$$

for given hyperparameters μ_l and $\sigma_{\alpha_l}^2$. In future versions of epidemia, these proportions may instead be assigned Beta priors.

2.2 Multiple Populations

Extending to multiple populations is straightforward. Suppose now that there are M non-overlapping populations with differing start dates $t_0^{(m)}$ and epidemic lengths T_m. Collect the

Table 1: Formal arguments for the epim function

Arguments	Description
formula	An R object of class 'formula'
data	Provides the data used in the model specified by 'formula'.
obs	A list providing all observed data.
pops	Population size for each group.
si	The serial interval of the disease.
seed_days	Number of days for which to seed infections
algorithm	The algorithm used to fit the model.
group_subset	(Optional)
center	Flag whether to center covariates.
prior	Prior on β, excluding the intercept.
prior_intercept	Prior on the intercept, if it exists in the model.
prior_covariance	Prior on the covariance matrix of b, the group-specific paramters.
r0	Set the constant A, described above.
prior_phi	Prior on the vector $(\phi_1, \ldots, \phi_L)$.
prior_tau	Prior on τ.
prior_PD	Flag whether to sample from the prior distribution.
sampling_args	Arguments to pass to the sampling algorithm.

time-varying reproduction numbers into a vector R with total length $T' := \sum_{m=1}^{M} T_m$, and consider the parameterization of R in terms of covariates given in (1). This approach allows flexible pooling of parameters through β, while still permitting between group variation using the group-effects b. Some populations may have little data; these groups could be at the early stages of the epidemic for example. This approach allows learning to be shared between groups, while helping to avoid overfitting to each group.

3 Implementation in R

epidemia is an R package allowing flexible specification of the models just outlined. In particular, **epidemia** utilizes R's formula interface to specify regression models for the time-varying reproduction number. The implementation of this shares many similarities with the `stan_glmer` function in **rstanarm** and the `glmer` function in **lme4**. The primary model fitting function in **epidemia** is `epim`. We now describe in detail the arguments

3.1 Model fitting using `epim`

4 Examples

The main model fitting function in **epidemia** is `epim`. This section demonstrates basic usage of this function.

4.1 Europe Covid

The package contains the dataset used in Flaxman et al. (2020). This data pertains to covid-19 in 11 European countries. Load this with

`EuropeCovid` is a list containing much of the information required for `epim`. These fields are named as follows.

```
names(EuropeCovid)
```

```
## [1] "data" "obs"  "pops" "si"
```

Each of these names correspond to an argument required for `epim`. The 'data' argument is a dataframe with columns referring to possible covariates for modelling the time-varying reproduction number. It contains one column which will specify the `groups` to be modelled, and an additional column giving the dates corresponding to the covariate data. Note that the covariates included here will not be used unless specified in the formula argument of `epim` – more on this below.

```
args <- EuropeCovid
data <- args$data
head(data)
```

```
##   country       date schools_universities self_isolating_if_ill public_events
## 1 Austria 2020-02-22                     0                      0             0
## 2 Austria 2020-02-23                     0                      0             0
## 3 Austria 2020-02-24                     0                      0             0
## 4 Austria 2020-02-25                     0                      0             0
## 5 Austria 2020-02-26                     0                      0             0
## 6 Austria 2020-02-27                     0                      0             0
##   lockdown social_distancing_encouraged
## 1        0                            0
## 2        0                            0
## 3        0                            0
## 4        0                            0
## 5        0                            0
## 6        0                            0
```

The `obs` argument is itself a list of lists. Each element of which corresponds to a different type of observation. This could for example be death, incidence, or hospitalisation counts. Following Flaxman et al. (2020), we only consider death counts here.

```
deaths <- args$obs$deaths
names(deaths)
```

```
## [1] "odata" "pvec"  "rates"
```

`epim` requires a formula, which specifies the model to be fit. In the current version of the package, the terms in the formila must correspond to the names in `data`. This may be relaxed in future versions, in line with other model fitting functions like `lm` or `glm`.

Although `data$country` contains 11 different populations, here we consider, for simplicity, only two. Specifically we will look at Germany and the United Kingdim. `epim` makes this simple by providing a `group_subset` argument.

```
args$group_subset <- c("Germany", "United_Kingdom")
```

A model is specified using the `formula` argument. The LHS of the formula always takes the form `R(x,y)` for some columns `x` and `y` in `data`. Unless `group_subset` is specified explicitly, `epim` will use the factor levels in `data$x` as the groups to model, and will use `data$y` to specify the modeled dates for each group. The dates must be a consecutive range, and there must be no missing covarite data in the columns specified on the RHS of the formula. The first date found for each group is assumed to be the beginning of the epidemic, and seeding of infections begins from this date.

We briefly give an intepretation of the models specified by different fomulas. Suppose for simplicity that the value of all covariates at the start date is zero, so that the intercept can be interpreted as specifying the R_0. Then

- `R(country, date) ~ 0 + ...` This is a no-intercept model. The effect is to set an exact starting R_0 which is the same for all countries. This is then modified at dates for which covariates become non-zero (i.e. after interventions come into place). This starting value is controlled by the `r0` argument to `epim`, which defaults to 3.28.
- `R(country, date) ~ 1 + ...` This gives a common intercept for all countries. The distribution for R_0 is the same for all countries. This distribution can be modified by specifying the `prior_intercept` argument for `epim`
- `R(country, date) ~ (1 | country) + ...` The random effects term allows the distribution for R_0 to depend on the country. The prior for these is controlled by both `prior_intercept` and `prior_covariance`.

Similar to Flaxman et al. (2020), we specify the following model.

```
args$formula <- R(country,date) ~ 1 + schools_universities + self_isolating_if_ill + pub
```

This model allows a separate R_0 for each country, and includes 6 different non-pharamceutical interventions (NPIs) to explain the changes in the time-varying reproduction number.

4.2 Prior Specification

The priors on the coefficient in the regressions, and on other parameters are controlled by the arguments `prior`, `prior_intercept`, `prior_covariance`, `prior_tau` and `prior_phi`. Please read the documentation for `epim` for a precise interpretation of these arguments.

Here we focus on specifying the `prior` argument. This controls the prior distribution of the coefficients in the regression. Any of the rstanarm priors can be used. We have also added a `shifted_gamma` prior to replicate the prior in Flaxman et al. (2020).

To quickly visualise the effect of the prior distribution we can used the `prior_PD` flag to `epim`. If `TRUE` epim will sample all parameters from their prior distribution. We specify the prior for the intercept as follows.

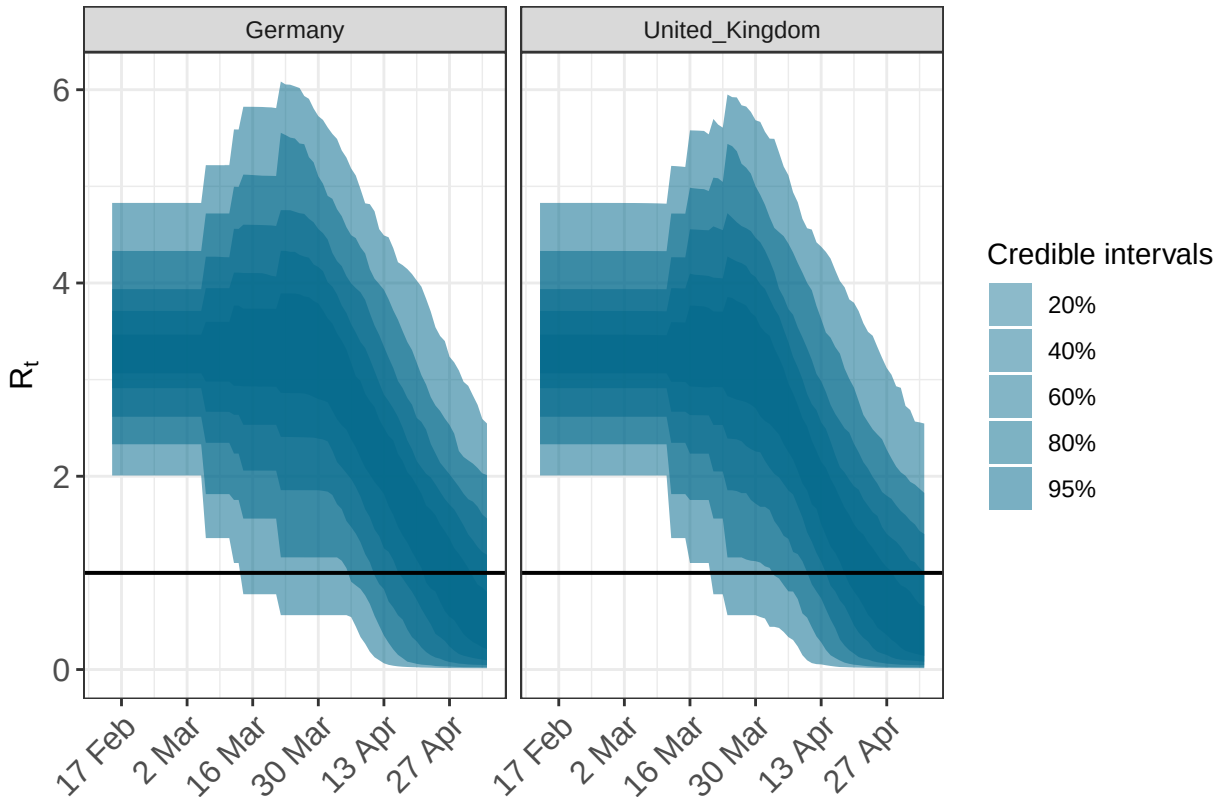
```
args$prior_intercept <- rstanarm::normal(location=0,scale = 0.5)
```

```
args$prior <- rstanarm::normal(scale = 0.5)
args$algorithm <- "sampling"
args$sampling_args <- list(iter=200,control=list(adapt_delta=0.95,max_treedepth=15))
args$prior_PD <- TRUE
fit <- xfun::cache_rds({do.call("epim", args)})
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and media
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and t
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
plot_rt(fit, levels = c(20,40,60,80,95))
```



And example of using a shifted gamma prior...

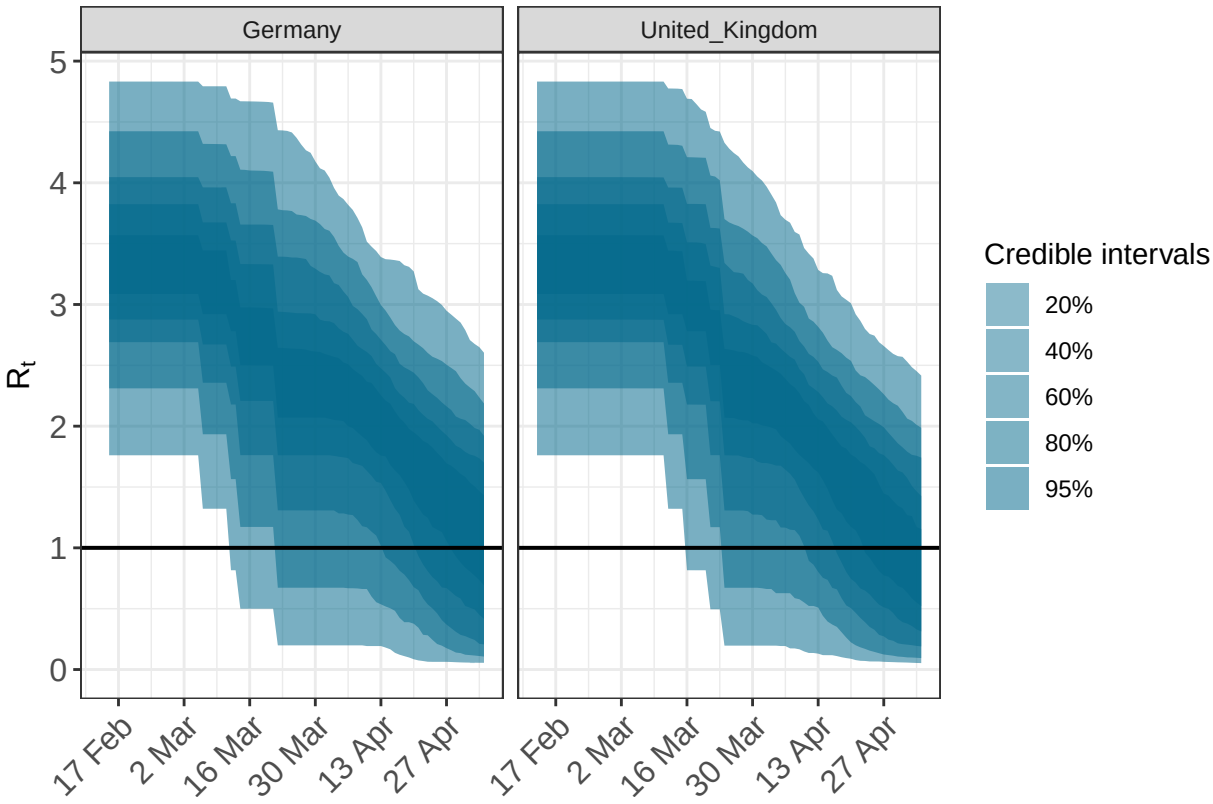
```
args$prior <- shifted_gamma(shape=1/6, scale=1, shift = -log(1.05)/6)
fit <- xfun::cache_rds({ do.call("epim", args)})
```

```
## Warning: The largest R-hat is 1.07, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
plot_rt(fit, levels = c(20,40,60,80,95))
```



Fitting the model to the data…

```
args$prior_PD = FALSE
fit <- xfun::cache_rds({do.call("epim", args)})
```

```
## Warning: There were 24 divergent transitions after warmup. Increasing adapt_delta abo
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: There were 2 chains where the estimated Bayesian Fraction of Missing Informa
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: The largest R-hat is 3.07, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
## Warning: Markov chains did not converge! Do not analyze results!
```

Printing the object gives a brief summary.

```
print(fit, digits = 2)
```

```
##
## Rt regression parameters:
## -----
## coefficients:
##                               Median MAD_SD
## (Intercept)                   -0.66   1.01
## schools_universities           0.28   0.43
## self_isolating_if_ill          0.66   0.98
## public_events                  1.64   1.52
## lockdown                       0.84   1.26
## social_distancing_encouraged   0.40   0.60
##
## Other model parameters:
## -----
##                               Median MAD_SD
## seeds[Germany]                3.66   5.18
## seeds[United_Kingdom]         4.90   6.81
## tau                           4.87   1.20
## phi                           1.76   2.61
## noise[Germany,deaths]         0.99   0.39
## noise[United_Kingdom,deaths] 0.99   0.36
```

Can extract the priors used...

```
prior_summary(fit)
```

```
## Priors for model 'fit'
## ------
## Intercept (after predictors centered)
##  ~ normal(location = 0, scale = 0.5)
##
## Coefficients
##  ~ gamma(shape = [0.17,0.17,0.17,...], scale = [1,1,1,...], shift = [-0.0081,-0.0081,-0.008
## ------
## See help('prior_summary.epimodel') for more details
```

And get the matrix of parameter draws

```
draws <- as.matrix(fit)
draws[1:4,1:4]
```

```
##          parameters
## iterations (Intercept) schools_universities self_isolating_if_ill public_events
##       [1,]  0.02275102            1.213675             0.6641195      3.330600
##       [2,]  0.02273181            1.213672             0.6641164      3.330597
##       [3,]  0.02273859            1.213681             0.6641380      3.330516
```

```
##       [4,]  0.02270380             1.213652             0.6641575      3.330426
```

References

Flaxman, Seth, Swapnil Mishra, Axel Gandy, H Juliette T Unwin, Thomas A Mellan, Helen Coupland, Charles Whittaker, et al. 2020. "Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe." *Nature*. https://doi.org/10.1038/s41586-020-2405-7.