Introduction to epidemia

This vignette is very much a work in progress, and will be regularly updated. It aims to demonstrate basis usage of the **epidemia** package. The main work is done in the `epim` function. Before continuing, please read the documentation for a more detailed description of this function.

Model Overview

The models that can be fit using the epidemia package are extensions of the model introduced in Flaxman et al. (2020).

A Single Population

We begin by describing the model as it applies to a single population. Extensions to multiple populations are described in the next section. Let $(t_0, \dots, t_n)$ be a time index, representing the period over which to simulate the epidemic. The index t_0 is taken to be the first date at which infections are observed in the population.

Data is observed from L different observation processes $Y_{1,t}, \dots, Y_{L,t}$. Each process represents a different type of observation; i.e. daily death data (as in Flaxman et al. (2020)), hospitalisation rates, or recorded infections. The random variables $Y_{t,l}$ are assumed to follow a negative binomial distribution with positive mean $y_{t,l}$ and variance given by

$$y_{t,l} + \frac{y_{t,l}^2}{\phi_l},$$

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

- $I_1, \dots, I_{t-1}$, where I_i represents the cumulative number of infections at period i,
- $\pi^{(l)}$, a discrete distribution on $\mathbb{Z}^+$, representing the time 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 manifest as an observation of type l.

To give intuition on the above quantities, suppose the observations were death counts. Conditional on an individual dying on a given date, then $\pi^{(l)}(t)$ is the probability that this individual was infected t days prior. α_l is then the infection fatality ratio (IFR) in the population.

This functional form for y_{tl} is then simply

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

While the distribution $\pi^{(l)}$ is *assumed*, a prior is used on α_l. This is a normal distribution truncated to $[0, 1]$, i.e.

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

for hyperparameters μ_l and $\sigma_{\alpha_l}^2$. In future versions of epidemia, this may be replaced by a Beta distribution. We may also place a prior on the distributions $\pi^{(l)}$.

To model the sequence $\{I_t\}$ we begin with an extension to continuous time. Let $I(t)$ be an ODE satisfying

$$\frac{dI(t)}{dt} = \frac{P - I(t)}{P} R_{\lfloor t \rfloor} c_{\lfloor t \rfloor}.$$

with R_t being the *unadjusted* (not adjusted for the susceptible population) time-varying reproduction number, and c_t is a weighted sum over previous infections, defined by

$$c_t = \sum_{\tau=1}^{t} (I_\tau - I_{\tau-1}) g(t - \tau).$$

I_t is defined by the exact solution to the above ODE. This is easily shown to be

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

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.

Multiple Infections

TODO

Example Usage

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

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

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

`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 + public_events + lo
```

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.

Trying out different priors.

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.

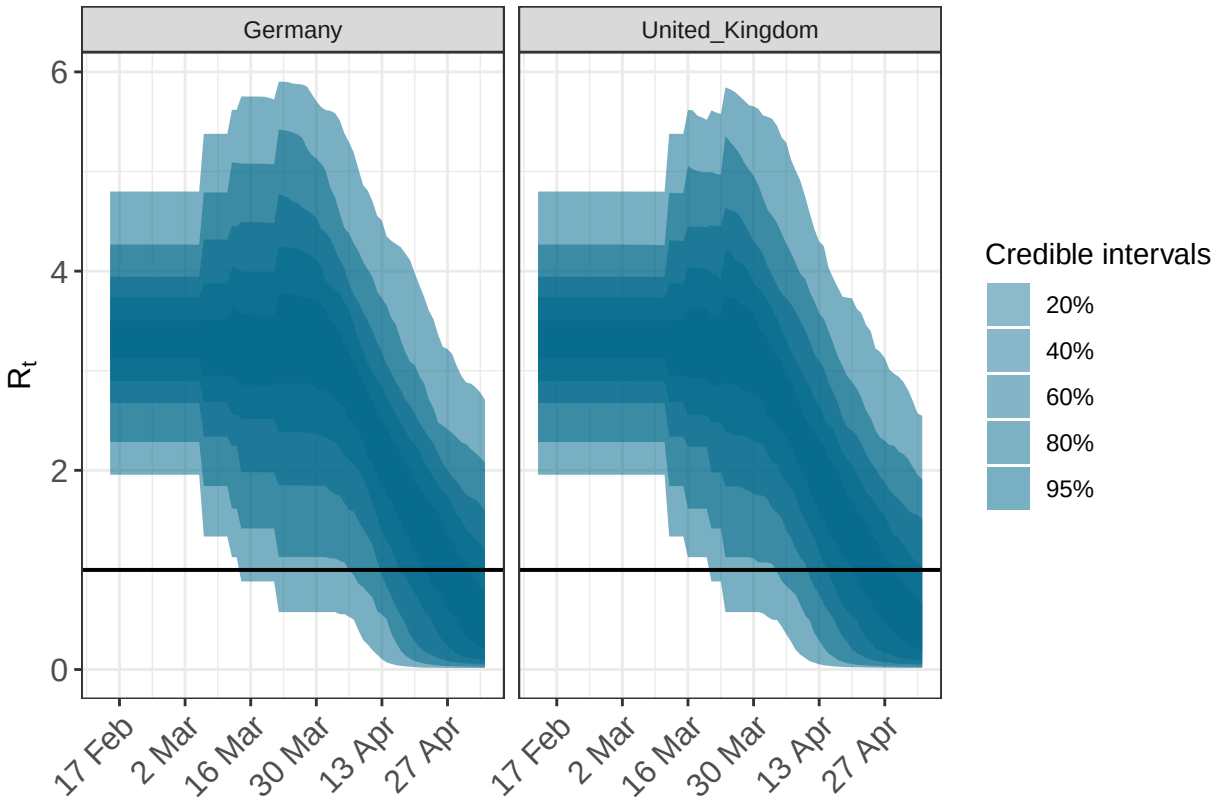
```
args$prior_intercept <- rstanarm::normal(location=0,scale = 0.5)
```

```
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 <- do.call("epim", args)
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## 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 tail quanti
## 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...

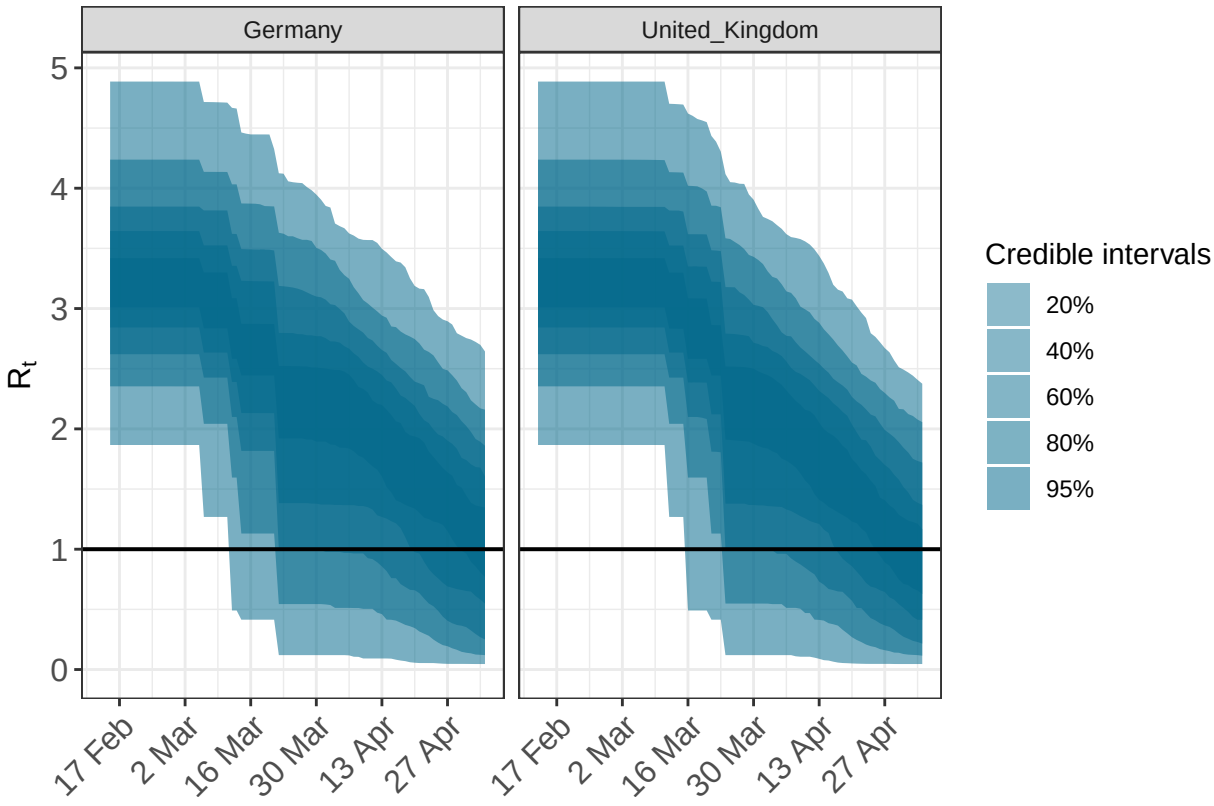
```
args$prior <- shifted_gamma(shape=1/6, scale=1, shift = -log(1.05)/6)
fit <- do.call("epim", args)
```

```
## Warning: The largest R-hat is 1.09, 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 medians may be
## 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 tail quanti
## 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 <- do.call("epim", args)
```

```
## Warning: There were 2 divergent transitions after warmup. Increasing adapt_delta above 0.95 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: There were 1 chains where the estimated Bayesian Fraction of Missing Information was low. S
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

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

```
## Warning: The largest R-hat is 1.69, 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 medians may be
## 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 tail quanti
## 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)                   -0.61   0.30
## schools_universities           0.03   0.05
## self_isolating_if_ill          0.02   0.05
## public_events                  0.55   0.82
## lockdown                       0.53   0.80
## social_distancing_encouraged   0.15   0.23
##
## Other model parameters:
## -----
##                               Median MAD_SD
## seeds[Germany]                13.36   6.81
## seeds[United_Kingdom]         23.29  15.94
## tau                           19.31  18.06
## phi                            5.08   1.31
## noise[Germany,deaths]          0.97   0.16
## noise[United_Kingdom,deaths]   1.05   0.16
```

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,-0.0081,...])
## ------
## See help('prior_summary.epimodel') for more details
```

And get the matrix of parameter draws

```
draws <- as.matrix(fit)
head(draws)
```

```
##           parameters
## iterations (Intercept) schools_universities self_isolating_if_ill public_events
##       [1,]  -0.9786109         -0.008131694          0.1988957183    0.09182462
##       [2,]  -0.9450068         -0.008131694          0.4123287343    0.73262984
##       [3,]  -0.6209880         -0.008131694          0.0561883941    0.42363192
##       [4,]  -0.6592316         -0.008131694          0.0033551123    2.44933718
##       [5,]  -0.7579092         -0.008131694         -0.0007586313    2.22516357
##       [6,]  -0.9719819         -0.008131694          0.4645815991    2.49864921
##           parameters
## iterations     lockdown social_distancing_encouraged seeds[Germany]
##       [1,]  2.578450922                    0.3134185       13.31864
##       [2,]  1.677422958                    0.2740063       13.46890
##       [3,]  2.141584169                    0.1480759       18.79816
##       [4,]  0.173152289                    0.2564952       17.79847
##       [5,]  0.295905656                    0.3521887       14.07328
##       [6,] -0.004930905                    0.1562297       13.99347
##           parameters
```

```
## iterations seeds[United_Kingdom]       tau      phi noise[Germany,deaths]
##       [1,]              17.84638  7.641848 4.294052             0.8756980
##       [2,]              24.92813 65.615191 6.179389             1.0055258
##       [3,]              31.26074 14.410884 6.137804             0.9907169
##       [4,]              31.24582 21.074856 5.820341             1.0454357
##       [5,]              20.68987 17.613808 4.167918             0.8569008
##       [6,]              18.71703  8.921374 6.820748             1.0294607
##           parameters
## iterations noise[United_Kingdom,deaths]
##       [1,]                    1.0658092
##       [2,]                    0.9212967
##       [3,]                    1.0696043
##       [4,]                    1.0349465
##       [5,]                    1.1146828
##       [6,]                    1.0030996
```

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.