How did it start?

Thibaut Jombart, Henry Mouysset

2024-03-22

Preamble

In this report, we investigate some of the key elements of the early stages of the bubonic Plague epidemic in Gévaudan, which started in 1720 in the village of Correjac, soon followed by the town of La Canourgue.

The analyses will rely on estimates of various epidemiological parameters, such as the case fatality ratio (CFR), the basic reproduction number (R_0), or delay distributions including the incubation period, the infectious period, or the serial interval. Where possible, these quantities are directly estimated from available date. Failing that, we use estimates from the literature.

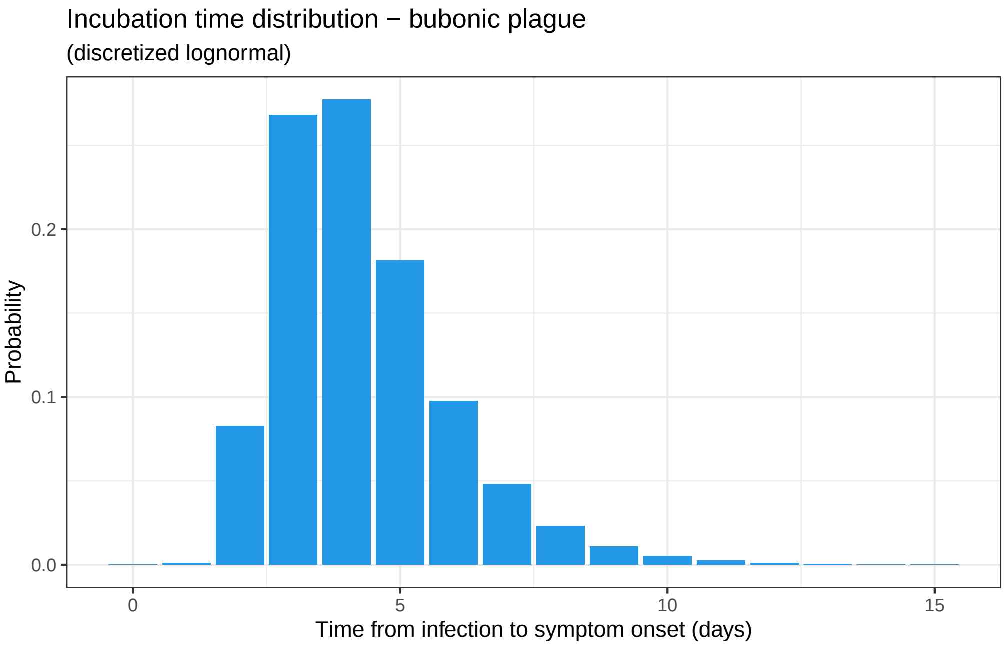
Estimates of key epidemiological features

Delay distributions

Incubation period

The incubation period is described in multiple papers but a general consensus seems to be around 2-6 days (see separate literature review). We build a discretized log-normal distribution compatible with these observations:

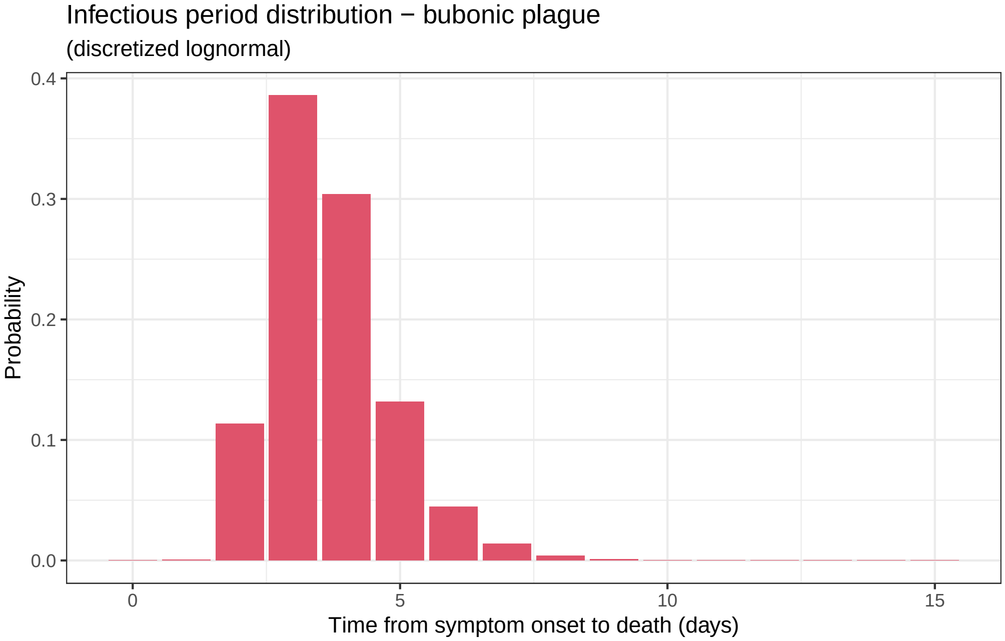
```r
library(distcrete)
library(tidyverse)

incub <- distcrete(
  "lnorm",
  meanlog = log(3.5),
  sdlog = log(1.5),
  interval = 1,
  w = 1
  )

incub_dat <- tibble(
  Day = 0:15,
  p = incub$d(0:15)
  )

incub_dat %>%
  ggplot(aes(x = Day, y = p)) +
  geom_col(fill = 4) +
  theme_bw() +
  labs(
    x = "Time from infection to symptom onset (days)",
    y = "Probability",
    title = "Incubation time distribution - bubonic plague",
    subtitle = "(discretized lognormal)")
```



Infectious period

The infectious period distribution is built similarly to match data from the literature, where for historical outbreaks death is reported to take place within 3 to 5 days post-symptom onset. Note that because this form had near 100% CFR, this is also the distribution of the time from onset of symptoms to death.

```
infec <- distcrete(
  "lnorm",
  meanlog = log(3),
  sdlog = log(1.4),
  interval = 1,
  w = 1
  )

infec_dat <- tibble(
  Day = 0:15,
  p = infec$d(0:15)
  )

infec_dat %>%
  ggplot(aes(x = Day, y = p)) +
  geom_col(fill = 2) +
  theme_bw() +
  labs(
    x = "Time from symptom onset to death (days)",
    y = "Probability",
    title = "Infectious period distribution - bubonic plague",
    subtitle = "(discretized lognormal)")
```



We note that the distribution is well in line with the mean duration of fatal, untreated bubonic plague of 3.6 days ("XXII. The Epidemiological Observations Made by the Commission in Bombay City" 1907):

```
## mean of 100 000 values from the distribution:
mean(infec$r(1e5))
```

```
## [1] 3.6716
```

Patient zero: the convict from Marseille

The theory of the convict walking all the way from Marseille with an sited bundle of wool to eventually sit Jean Quintin in Saint Laurent d'Olt is suspicious: the man would have had to not get sited during his entire trip, to then sit JQ after a fairly brief encounter.

The trip by foot via current roads is 273 km, and would have likely been longer using roads at the time. Especially since patrols were restricting movement in the area, and would have demanded some extra time to be avoided.

We will explore 3 scenarios, with varying durations for the trip, which we posit was at least about 300 km:

- 7 days (very optimistic, $>$ 40 km per day)
- 12 days ($\sim$ 25km per day)
- 20 days ($\sim$ 15 km per day)

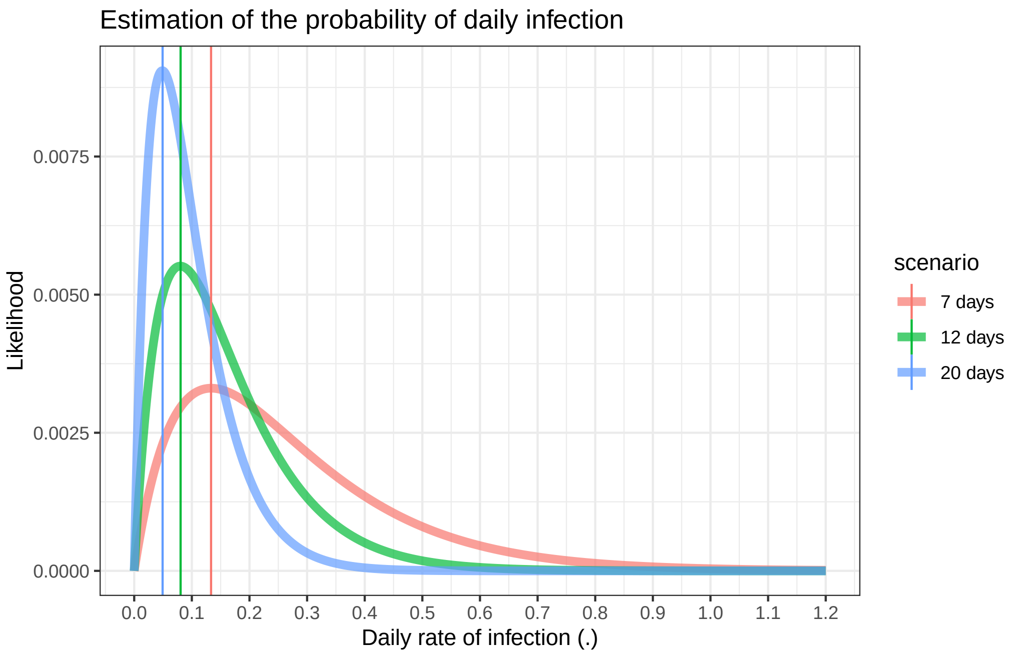
We do not know what the daily rate of infection λ from the wool bundle, but we can derive a likelihood profile for each scenario, using:

$$\mathcal{L}(\lambda) = p(T|\lambda) = (e^{-\lambda})^T(1 - e^{-\lambda})$$

where T is the number of days the trip took, and $(1 - e^{-\lambda})$ is the daily probability of infection from the sited bundle.

Let us look at these profiles, ensuring we re-standardise both densities to 1 to make them comparable, as they rely on datasets of different sizes:

```r
like_trip <- function(lambda, T) {
  p <- 1 - exp(-lambda)
  (1 - p)^T * p
}

lambda_res <- tibble(
  val = seq(0, 1.2, length = 1000),
  "7 days" = like_trip(val, 7),
  "12 days" = like_trip(val, 12),
  "20 days" = like_trip(val, 20)
) %>%
  mutate_at(
    vars(contains("days")),
    function(x) x / sum(x)
  )

lambda_res_long <-  lambda_res %>%
  pivot_longer(-1, names_to = "scenario", values_to = "likelihood") %>%
  mutate(scenario = factor(scenario, levels = paste(c(7, 12, 20), "days")))

lambda_res_mle <- lambda_res_long %>%
  group_by(scenario) %>%
  summarise(MLE = val[which.max(likelihood)])

lambda_res_long %>%
  ggplot(aes(x = val, y = likelihood, color = scenario)) +
  geom_line(linewidth = 2, alpha = 0.7) +
  geom_vline(data = lambda_res_mle, aes(xintercept = MLE, color = scenario)) +
  theme_bw() +
  labs(
    x = "Daily rate of infection (\U03BB)",
    y = "Likelihood",
    title = "Estimation of the probability of daily infection"
  ) +
  scale_x_continuous(n.breaks = 20)
```



```
lambda_res_mle
```

```
## # A tibble: 3 x 2
##   scenario    MLE
##   <fct>     <dbl>
## 1 7 days   0.133
## 2 12 days  0.0805
## 3 20 days  0.0492
```

We find some more likely values than others, and this leans towards fairly low rates ranging from 0.0492 to 0.1333. We calculate the corresponding probabilities that these events occurred using the MLE:

```
lambda_res_mle <- lambda_res_mle %>%
  mutate(
    days = as.integer(sub(" days", "", scenario)),
    proba = like_trip(lambda = MLE, T = days)
    )
lambda_res_mle
```

```
## # A tibble: 3 x 4
##   scenario    MLE  days  proba
##   <fct>     <dbl> <int>  <dbl>
## 1 7 days   0.133      7 0.0491
## 2 12 days  0.0805    12 0.0294
## 3 20 days  0.0492    20 0.0179
```

The resulting probabilities are quite low, ranging from 1.8% chances for a trip of 20 days, to 4.9% for 7 days. To assess the overall plausibility of these scenari, questions remain: how many such trips where made by people carrying the infection from Marseille at the time? And were such trips indeed possible in the first place, given the containment measures in place?

The first case after Jean

It is written that Jean showed symptoms on the 23rd November, and died 3 days later on the 26th. The first case in his household died on the 18th December. We can assess how likely this delay is by looking at the delay distributions for Bubonic plague, integrating over the possible dates of infection (24th, 25th, 26th November), and convolving the incubation period, and the survival period.

```
## calculate possible delays from infection to death
child_delay <- as.integer(as.Date("1720-12-18") - as.Date("1720-11-24") + 0:2)
child_delay
```

```
## [1] 24 25 26
```

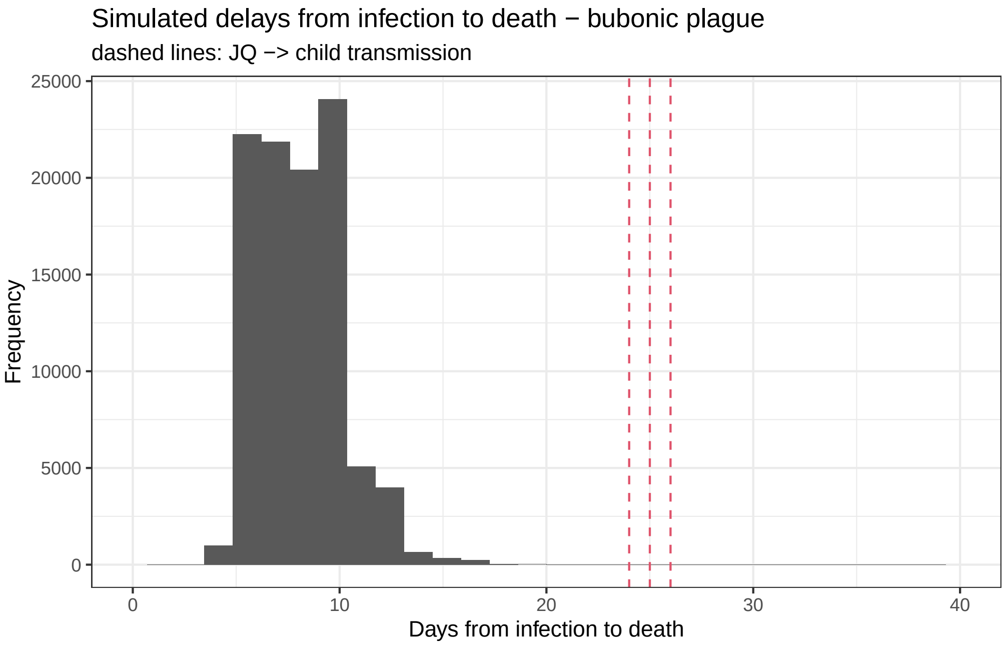
```
## convolve using 1e5 draws from distributions
sim_delay_inf_death <- incub$r(1e5) + infec$r(1e5)

## calculate p-values
child_delay_pval <- sapply(child_delay, function(x) mean(sim_delay_inf_death >= x))
child_delay_pval
```

```
## [1] 0 0 0
```

```
mean(child_delay_pval)
```

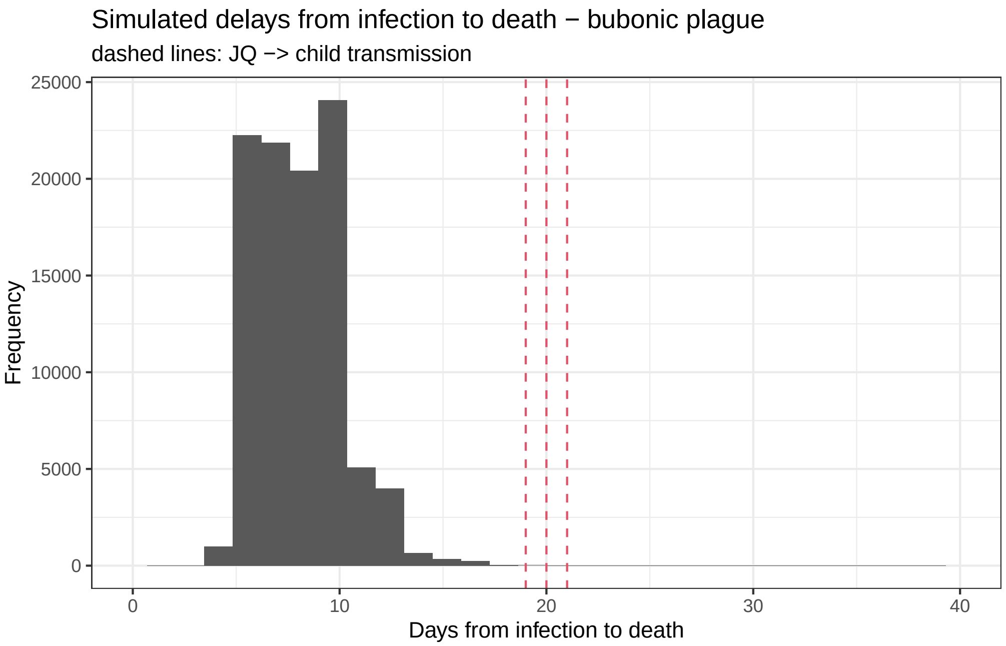
```
## [1] 0
```

```
qplot(sim_delay_inf_death) +
  theme_bw() +
  geom_vline(xintercept = child_delay, color = 2, linetype = 2) +
  xlim(0, 40) +
  labs(
    title = "Simulated delays from infection to death - bubonic plague",
    subtitle = "dashed lines: JQ -> child transmission",
    x = "Days from infection to death",
    y = "Frequency"
    )
```



It is very unlikely that the first secondary case was directly sited by Jean Quintin. There remains the possibility that sited surfaces (clothes, bedsheets) were sited by *Y. pestis*, which has been shown to be able to last up to 5 days (Rose et al. 2003). We can replicate the same analysis subtracting these 5 days from the overal delay:

```
qplot(sim_delay_inf_death) +
  theme_bw() +
  geom_vline(xintercept = child_delay - 5, color = 2, linetype = 2) +
  xlim(0, 40) +
  labs(
    title = "Simulated delays from infection to death - bubonic plague",
    subtitle = "dashed lines: JQ -> child transmission",
    x = "Days from infection to death",
    y = "Frequency"
    )
```



```
child_delay_pval_alt <- sapply(child_delay - 5, function(x) mean(sim_delay_inf_death >= x))
child_delay_pval_alt
```

```
## [1] 0.00031 0.00012 0.00006
```

```
mean(child_delay_pval_alt)
```

```
## [1] 0.0001633333
```

It seems we can rule out the hypothesis of a direct transmission of bubonic plague. Possible aternatives are:

- it was a bubonic plague, with a long duration of illness in the kid; might be unlikely seeing that patients seem to die very quickly - TBC
- the kid was **not** sited by his dad, but from a zoonotic source

Modelling the early epidemic

Epidemic curve

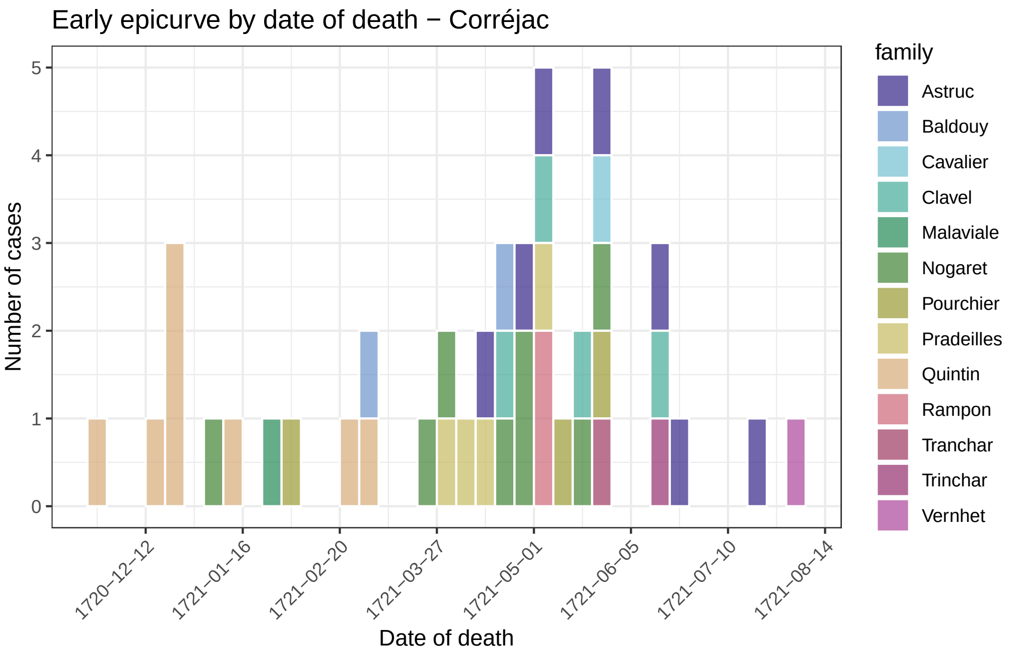
Here we analyse the linelist of the first few cases in Corréjac, followed by La Canourgue.

```
file_path <- here::here("data", "early_linelist.ods")
x <- rio::import(file_path, sheet = "linelist") %>%
  tibble() %>%
  mutate(date_of_death = as.Date(date_of_death, format = "%d/%m/%Y"))
x
```

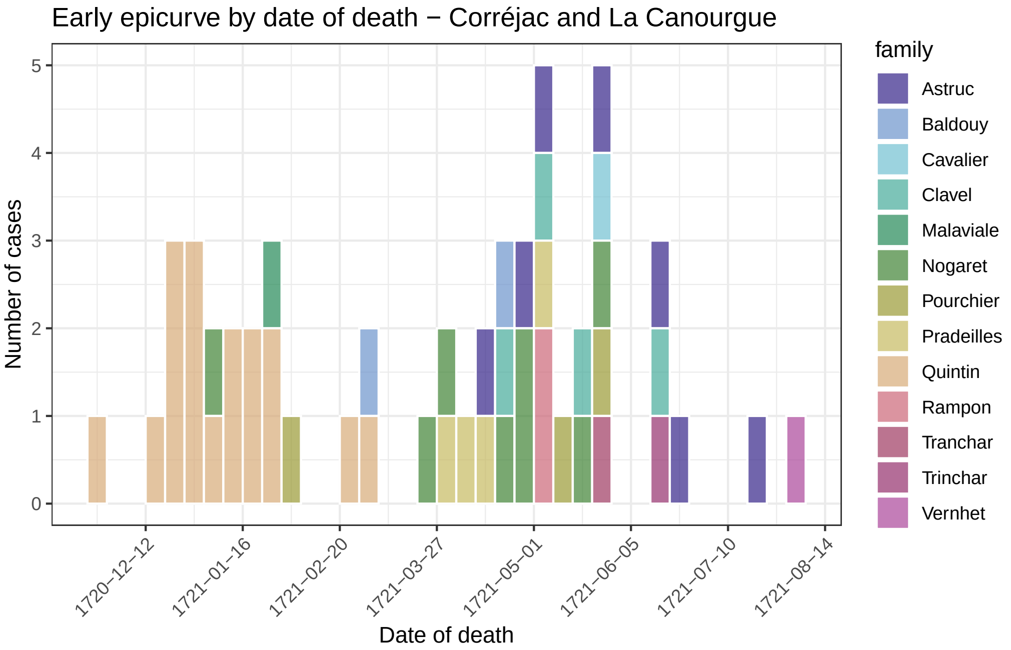
```
## # A tibble: 52 x 4
##    name                 date_of_death location family
##    <chr>                <date>        <chr>    <chr>
##  1 QUINTIN Jean         1720-11-26    Corréjac Quintin
##  2 QUINTIN Ambroise     1720-12-18    Corréjac Quintin
##  3 CASSANHES Marguerite 1720-12-25    Corréjac Quintin
##  4 QUINTIN enfant 1     1720-12-25    Corréjac Quintin
##  5 QUINTIN enfant 2     1720-12-25    Corréjac Quintin
##  6 NOGARET Pierre       1721-01-04    Corréjac Nogaret
##  7 DEROUCH Jean         1721-01-09    Corréjac Quintin
##  8 MALAVIALE Marguerite 1721-01-28    Corréjac Malaviale
##  9 POURCHIER Antoine    1721-02-02    Corréjac Pourchier
## 10 DEROUCH Jean fils    1721-02-22    Corréjac Quintin
## # i 42 more rows
```

We build a simple epidemic curve for Corréjac, and for both locations together:

```
library(incidence2)

x %>%
  filter(location == "Corréjac") %>%
  incidence("date_of_death", groups = "family", interval = 7) %>%
  plot(fill = "family", colour_palette = muted, angle = 45, border = "white") +
  labs(
    title = "Early epicurve by date of death - Corréjac",
    x = "Date of death",
    y = "Number of cases"
    )
```



```
x %>%
  incidence("date_of_death", groups = "family", interval = 7) %>%
  plot(fill = "family", colour_palette = muted, angle = 45, border = "white") +
  labs(
    title = "Early epicurve by date of death - Corréjac and La Canourgue",
    x = "Date of death",
    y = "Number of cases"
    )
```



There is one key question here: do we assume Corréjac and La Canourgue were disconnected at the time, or do we treat these as a single location.

A transmission model for bubonic plague

A branching process will likely be best at capturing small fluctuations in case incidence over time, whilst neglecting the impact of the depletion of susceptible individuals. However we need to allow for a background rate of zoonotic introduction in addition to the classical person-to-person transmission (Wallinga and Teunis 2004). As such, we will be using a Hawkes process (Hawkes 1971) to model the case incidence over time, using augmented data to make up for the lack of precise information on dates of symptom onset and infection.

Notations

Data and augmented data

- $i = 1, ..., n$: index of individuals
- $t = 1, ..., T$: time index
- d_i: date of death of case i (data)
- O_i: date of symptom onset of case i (augmented data)
- I_i: date of infection of case i *(augmented data)
- Y_t: incidence of new infections at time t (augmented data, derived from I_i)

Distributions

- $\mathcal{F}$: incubation period distribution (so we have $(O_i - I_i) \sim \mathcal{F}(.)$)
- $\mathcal{G}$: infectious period distribution (so we have$(D_i - O_i) \sim \mathcal{G}(.)$)
- $\mathcal{H}$: generation time distribution, obtained from the convolution $\mathcal{F} * \mathcal{G} = \mathcal{H}$

Both $\mathcal{F}$ and $\mathcal{G}$ are drawn from the literature.

Parameters

- λ_z: the rate of zoonotic introduction, assumed constant over time
- R: the effective reproduction number, assumed constant over time

The model

We use a Bayesian framework to describe the distribution of parameters (θ), data (x), augmented data (X), where the posterior distribution is defined as:

$$p(\theta|x) \propto p(x|\theta)p(\theta) \tag{1}$$

where $p(x|\theta)$ is the likelihood function and $p(\theta)$ the prior distributions.

The likelihood can be written as:

$$\begin{aligned} p(x|\theta) &= p(d, O, I|\lambda_z, R) & (2) \\ &= p(d|O)p(O|I)p(I|\lambda_z, R) & (3) \\ &= \left(\prod_i p(d_i|O_i) \prod_i p(O_i|I_i)\right) p(I|\lambda_z, R) & (4) \\ &= \left(\prod_i \mathcal{F}(d_i - O_i)\mathcal{G}(O_i - I_i)\right) p(I|\lambda_z, R) & (5) \end{aligned}$$

The calculation of $p(I|\lambda_z, R)$ is defined by the Hawkes process for the incidence Y:

$$\begin{aligned} p(I|\lambda_z, R) &= p(Y|\lambda_z, R) & (6) \\ &= \prod_t p(Y_t|Y_1, ..., Y_{t-1}, \lambda_z, R) & (7) \end{aligned}$$

where the incidence Y_t is governed by:

$$Y_t \sim \mathcal{P}(\lambda_t) \tag{8}$$

where $\mathcal{P}(.)$ is the probability mass function of a Poisson distribution, and with:

$$\lambda_t = \lambda_z + \sum_{s=1}^{t-1} RY_s\mathcal{H}(t - s) \tag{9}$$

Finally, we assume independent priors for λ_z and R such that:

$$p(\theta) = p(\lambda_z, R) = p(\lambda_z)p(R) \tag{10}$$

Estimation process

We can sample from the posterior distribution using the Metropolis algorithm with augmented data with the following process:

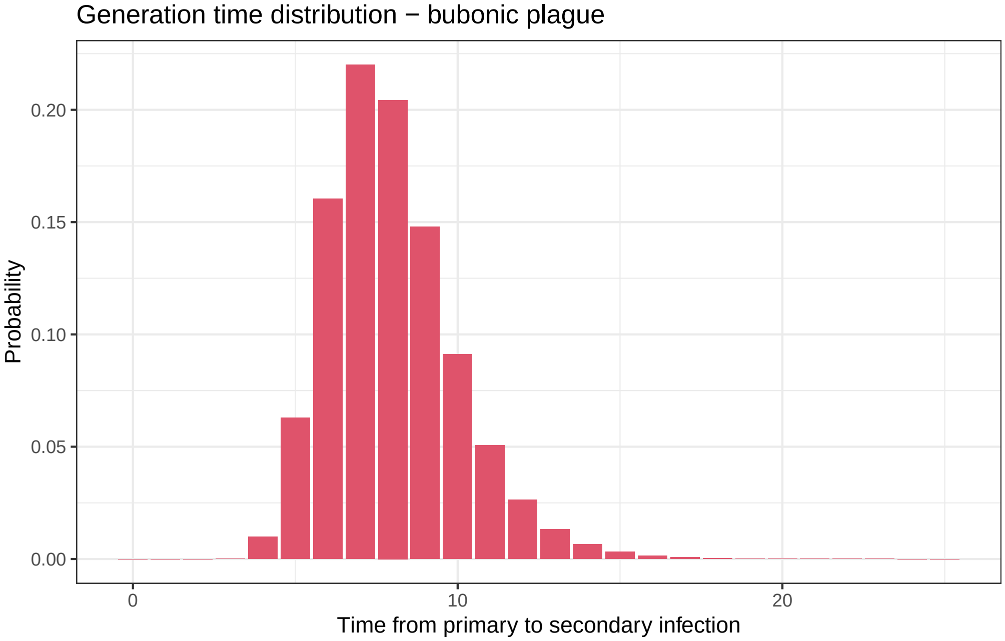
1. draw augmented data O using $d_i - O_i \sim \mathcal{F}$
2. draw augmented data I using $O_i - I_i \sim \mathcal{G}$
3. propose (using symmetric proposal distributions) new values for θ^* and accept/reject these values with probability: $max(1, \frac{p(\theta^*|x)}{p(\theta|x)})$, where θ represents the previous parameter state; in practice, separate movements are used for λ_z and R
4. go back to 1 until desired number of iterations reached

Distributions

Generation time

The generation time distribution $\mathcal{H}$ can be calculated by convolving the incubation time $\mathcal{F}$ and the infectious period distribution $\mathcal{G}$.

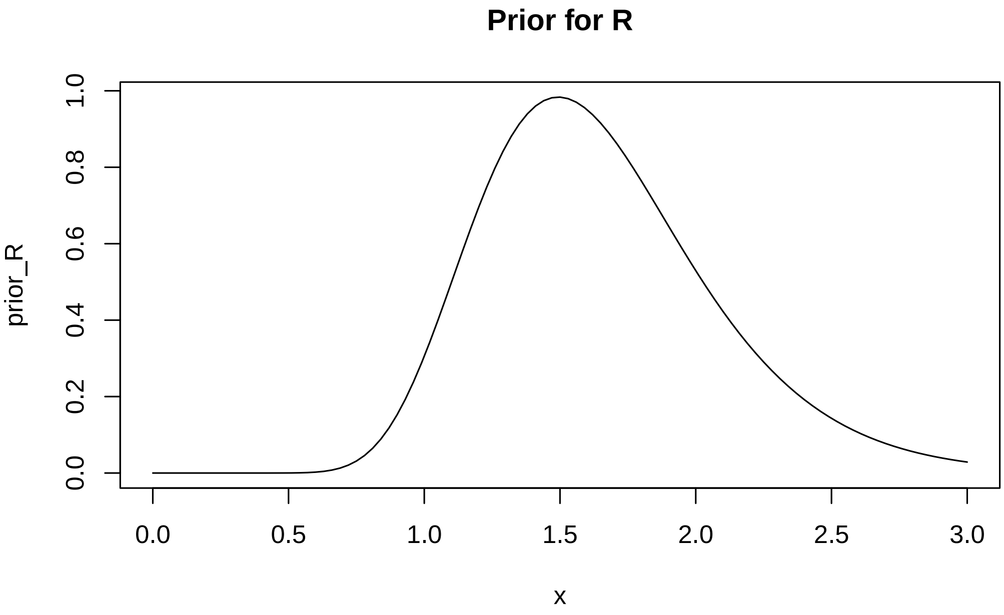
```r
## PMF for delays of 0, 1, ...
gentime_pmf <- convolve(incub$d(0:100), rev(infec$d(0:100)), type = "open")
gentime_pmf[1] <- 0
gentime_pmf <- gentime_pmf / sum(gentime_pmf) # superfluous

## emulate density function
gentime_d <- function(x, log = FALSE) {
  ## make sure we don't get out of bounds
  x <- as.integer(x)
  x[x < 0] <- 0
  x[x > 100] <- 100
  out <- gentime_pmf[x+1]
  if (log) {
    out <- log(out)
  }
  out
}

## emulate rng function
gentime_r <- function(n) {
  sample(0:100, size = n, replace = TRUE, prob = gentime_d(0:100))
}

gentime <- list(d = gentime_d, r = gentime_r)

gt_dat <- tibble(
  Day = 0:25,
  p = gentime$d(0:25)
  )

gt_dat %>%
  ggplot(aes(x = Day, y = p)) +
  geom_col(fill = 2) +
  theme_bw() +
  labs(
    x = "Time from primary to secondary infection",
    y = "Probability",
    title = "Generation time distribution - bubonic plague"
    )
```



Priors

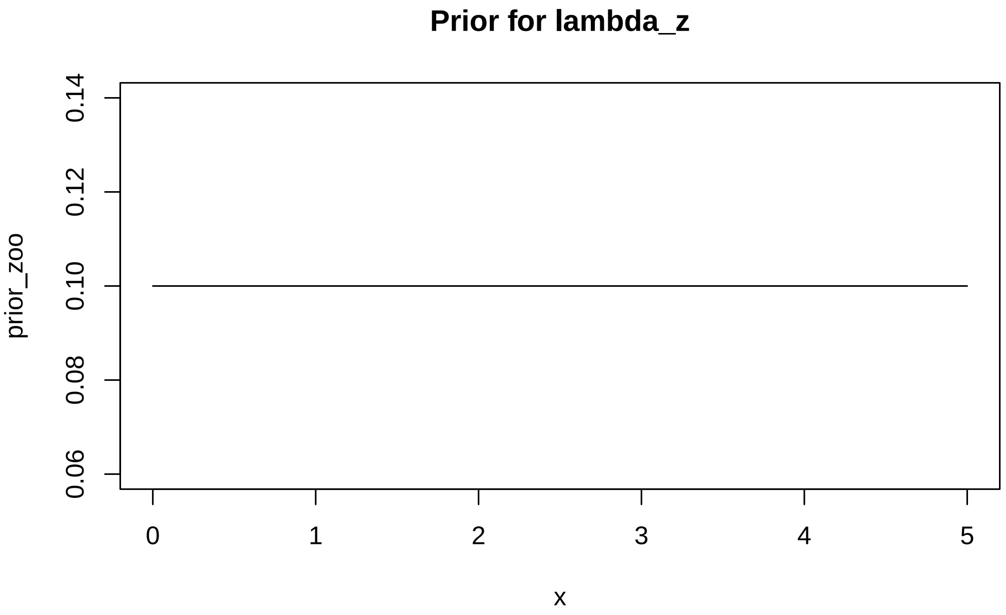
We use a prior for R derived from (Dean, Krauer, and Schmid 2019) as a lognormal distribution with mean 1.6 and standard deviation 1.3:

```
prior_R <- function(x, log = FALSE) dlnorm(x, log(1.6), log(1.3), log = log)
plot(prior_R, xlim = c(0, 3), main = "Prior for R")
```



Rates of zoonotic introductions are usually harder to estimate, so we use fairly uninformative priors for λ_z:

```
#prior_zoo <- function(x, log = FALSE) dexp(x, 0.2, log = log)
prior_zoo <- function(x, log = FALSE) dunif(x, 0, 10, log = log)
plot(prior_zoo, xlim = c(0, 5), main = "Prior for lambda_z")
```



Implementation

The following code implements the model.

```
## Function to generate augmented data
## This will return a list with all the data needed for likelihood calculation.
## Note that dates of deaths are converted to integers, with 0 the earliest death.

make_aug_data <- function(date_death = x$date_of_death,
                          r_incub = incub$r,
                          r_infec = infec$r
                          ) {
  if (any(is.na(date_death))) {
    msg <- "Some dates of death are missing"
    stop(msg)
  }
  n <- length(date_death)
  date_death <- as.integer(date_death - min(date_death))
  date_onset <- date_death - r_infec(n)
  date_infection <- date_onset - r_incub(n)
  out <- data.frame(
    infection = date_infection,
    onset = date_onset,
    death = date_death
  )

  ## Use this to have 0 as the earliest date; otherwise we do get some negative
  ## dates:
  ## data.frame(lapply(a, function(x) x - min(a)))
  out
}



## Function to calculate the log-likelihood
```

```
##
## @param data: a data.frame with 3 columns as returned by make_aug_data()
## @param params: a list with two items: zoo, and R, in this order
## @param d_incub: a PMF function for the incubation period
## @param d_infec: a PMF function for the infectious period
## @param d_gentime: a PMF function for the generation time

compute_loglike <- function(data,
                            params,
                            d_incub = incub$d,
                            d_infec = infec$d,
                            d_gentime = gentime$d) {

  ### There are 3 components to the likelihood:^
  ###
  ### 1. delay from onset to death (infectious period)
  ### 2. delay from infection to onset (incubation time)
  ### 3. incidence of infections from Hawkes process

  ### Component 1
  p_1 <- sum(d_infec(data$death - data$onset, log = TRUE))

  ### Component 2
  p_2 <- sum(d_incub(data$onset - data$infection, log = TRUE))

  ### Component 3
  first_day <- min(data$infection)
  last_day <- max(data$infection)

  #### calculate incidence for the whole time period, do no miss the zeros
  incid <- sapply(
    seq(first_day, last_day, by = 1),
    function(i) sum(data$infection == i)
  )

  #### get relative FOIs - check that w indeed start at 1 in EpiEstim
  lambdas_p2p <- EpiEstim::overall_infectivity(incid, d_gentime(0:100))
  lambdas <- params$zoo + (lambdas_p2p * params$R)
  p_3 <- sum(na.omit(dpois(incid, lambdas, log = TRUE)))

  p_1 + p_2 + p_3
}


## Function to calculate log-priors
compute_priors <- function(params,
                           d_zoo = prior_zoo,
                           d_R = prior_R) {
  d_zoo(params$zoo, log = TRUE) + d_R(params$R, log = TRUE)
}

## Compute log-posterior
compute_post <- function(data,
```

```r
                         params,
                         d_incub = incub$d,
                         d_infec = infec$d,
                         d_gentime = gentime$d,
                         d_zoo = prior_zoo,
                         d_R = prior_R
                         ) {
  compute_loglike(data, params, d_incub, d_infec, d_gentime) +
    compute_priors(params, d_zoo, d_R)
}

## Function implementing the whole MCMC procedure

## @param n_iter the number of iterations of the MCMC; defaults to 1000
## @param sd_zoo the standard deviation of the normal proposal distribution for
##   the rate of zoonotic introductions
## @oaram sd_R the standard deviation of the normal proposal distribution for
##   the effective reproduction number
## @param ini_zoo the initial value of the daily rate of zoonotic introduction
## @param ini_R the initial value of the effective reproduction number

estimate_params <- function(date_death = x$date_of_death,
                            n_iter = 1e3,
                            d_incub = incub$d,
                            d_infec = infec$d,
                            d_gentime = gentime$d,
                            d_zoo = prior_zoo,
                            d_R = prior_R,
                            r_incub = incub$r,
                            r_infec = infec$r,
                            sd_zoo = 0.1,
                            sd_R = 0.4,
                            ini_zoo = runif(1, 0, 1),
                            ini_R = runif(1, 0, 3)) {

  ### Initialize MCMC
  params <- list(zoo = ini_zoo, R = ini_R)
  aug_data <- make_aug_data(date_death, r_incub, r_infec)
  ini_post <- compute_post(aug_data,
                           params,
                           d_incub,
                           d_infec,
                           d_gentime,
                           d_zoo,
                           d_R)
  new_params <- current_params <- params

  ### Build output structure
  mcmc <- list(
    step = seq_len(n_iter),
    post = double(n_iter),
    zoo = double(n_iter),
    R = double(n_iter)
  )
```

```
accept_zoo <- 0
accept_R <- 0

for (i in seq_len(n_iter)) {

  #### make new augmented data
  aug_data <- make_aug_data(date_death, r_incub, r_infec)
  current_post <- compute_post(aug_data,
                               current_params,
                               d_incub,
                               d_infec,
                               d_gentime,
                               d_zoo,
                               d_R)

  #### propose new zoo
  new_params$zoo <- current_params$zoo + rnorm(1, sd = sd_zoo)

  #### accept/reject zoo
  new_post <- compute_post(aug_data,
                           new_params,
                           d_incub,
                           d_infec,
                           d_gentime,
                           d_zoo,
                           d_R)
  p_accept_zoo <- exp(new_post - current_post)
  if (runif(1) <= p_accept_zoo) { # accept move
    current_params$zoo <- new_params$zoo
    current_post <- new_post
    accept_zoo <- accept_zoo + 1
  } else { # reject move
    new_params$zoo <- current_params$zoo
  }

  #### propose new R
  new_params$R <- current_params$R + rnorm(1, sd = sd_R)

  #### accept/reject R
  new_post <- compute_post(aug_data,
                           new_params,
                           d_incub,
                           d_infec,
                           d_gentime,
                           d_zoo,
                           d_R)
  p_accept_R <- exp(new_post - current_post)
  if (runif(1) <= p_accept_R) { # accept move
    current_params$R <- new_params$R
    current_post <- new_post
    accept_R <- accept_R + 1
  } else { # reject move
    new_params$R <- current_params$R
    }


    ### store info from this iteration
    mcmc$post[i] <- current_post
    mcmc$zoo[i] <- current_params$zoo
    mcmc$R[i] <- current_params$R
  }

  list(
    mcmc = data.frame(mcmc),
    accept_zoo = accept_zoo / n_iter,
    accept_R = accept_R / n_iter
  )

}
```

We run the chains for a few iterations:

```
system.time(res <- estimate_params(n_iter = 10))
```

```
##    user  system elapsed
##   2.308   0.279   2.598
```

```
head(res)
```

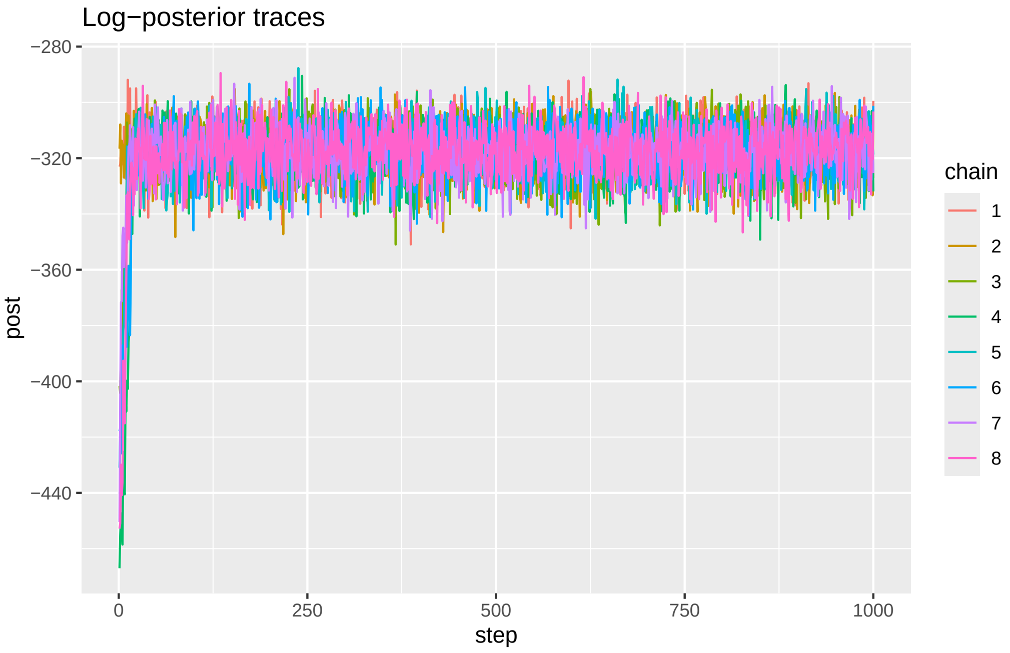
```
## $mcmc
##    step      post        zoo         R
## 1     1 -363.5310 0.11478158 2.3465806
## 2     2 -350.0599 0.11478158 1.9436122
## 3     3 -355.0849 0.11478158 1.9436122
## 4     4 -355.7194 0.02163069 1.7545463
## 5     5 -336.0286 0.02163069 1.7545463
## 6     6 -334.1726 0.02163069 1.7545463
## 7     7 -350.0838 0.02163069 1.7545463
## 8     8 -336.2709 0.02163069 1.6188067
## 9     9 -320.0484 0.02163069 0.7312108
## 10   10 -334.6498 0.02163069 0.7312108
##
## $accept_zoo
## [1] 0.2
##
## $accept_R
## [1] 0.5
```

```
tail(res)
```

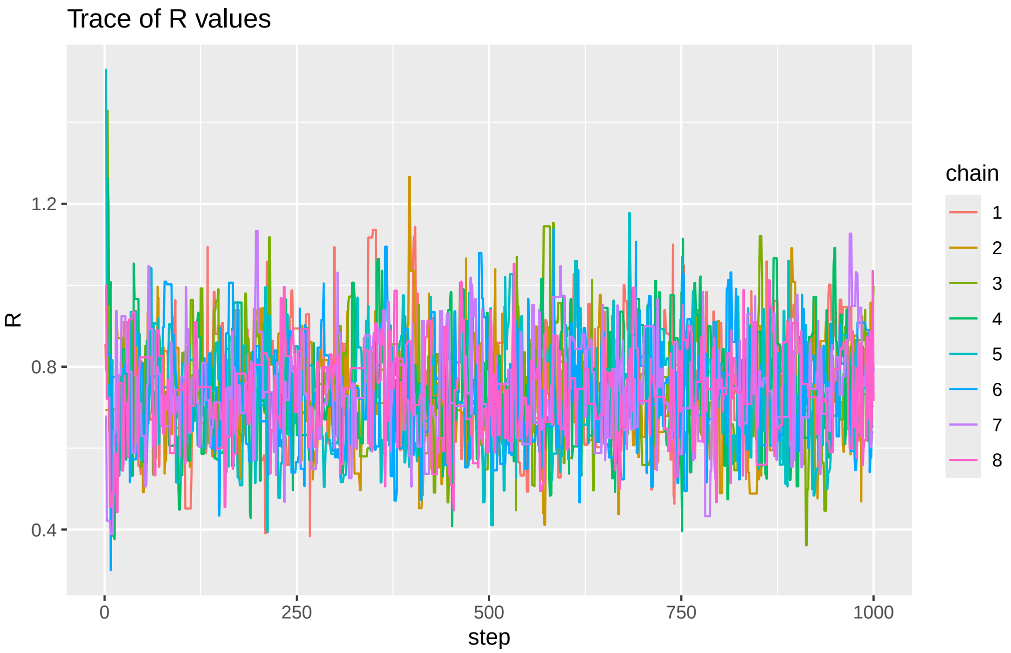
```
## $mcmc
##    step      post        zoo         R
## 1     1 -363.5310 0.11478158 2.3465806
## 2     2 -350.0599 0.11478158 1.9436122
## 3     3 -355.0849 0.11478158 1.9436122
## 4     4 -355.7194 0.02163069 1.7545463
## 5     5 -336.0286 0.02163069 1.7545463
## 6     6 -334.1726 0.02163069 1.7545463
## 7     7 -350.0838 0.02163069 1.7545463
## 8      8 -336.2709 0.02163069 1.6188067
## 9      9 -320.0484 0.02163069 0.7312108
## 10    10 -334.6498 0.02163069 0.7312108
##
## $accept_zoo
## [1] 0.2
##
## $accept_R
## [1] 0.5
```

We can try to run several chains in parallel:

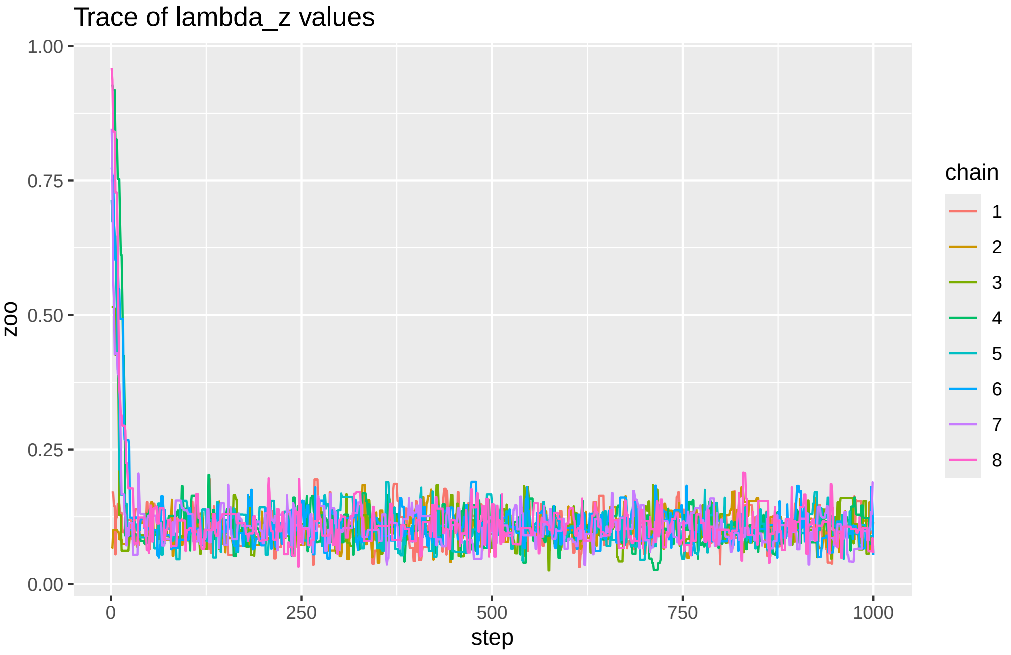
```
# library(future.apply)
# plan(multisession, workers = 8)
# res <- future_lapply(1:8, function(i) estimate_params(n_iter = 100))

#res <- lapply(1:8, function(i)
#  cbind.data.frame(estimate_params(n_iter = 100), chain = i)
#  )

res <- readRDS("res_1000iter_8chains_04-10-2024.rds")

#plan(sequential)

chains <- Reduce(rbind, lapply(1:length(res), function(i) res[[i]]))
names(chains) <- gsub("mcmc.", "", names(chains))
chains$chain <- factor(chains$chain)


## Plots
ggplot(chains) +
  geom_line(aes(x = step, y = post, color = chain)) +
  labs(title = "Log-posterior traces")
```



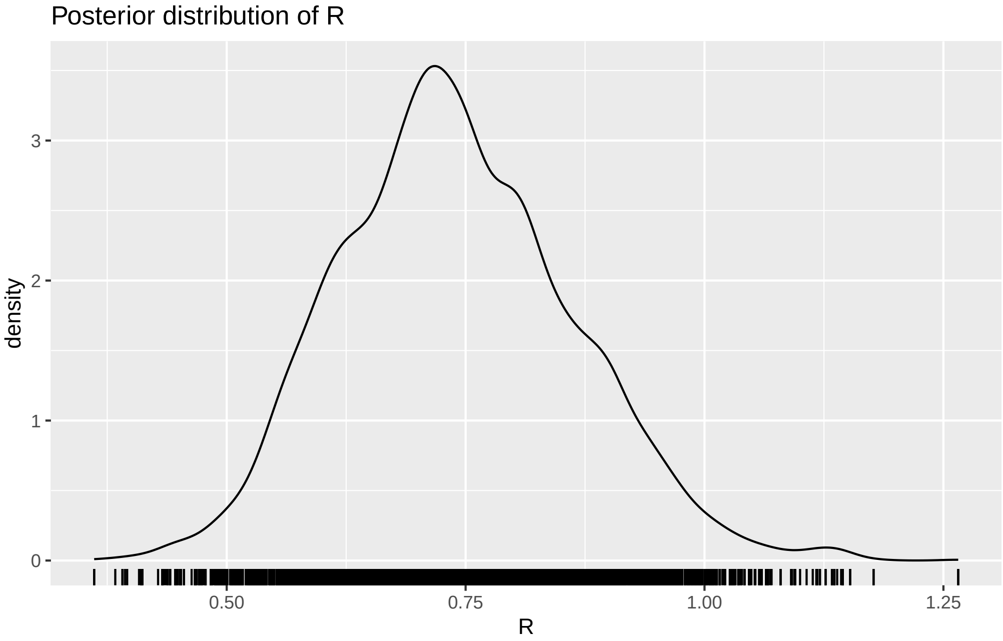
```r
## Plots
ggplot(chains) +
  geom_line(aes(x = step, y = R, color = chain)) +
  labs(title = "Trace of R values")
```



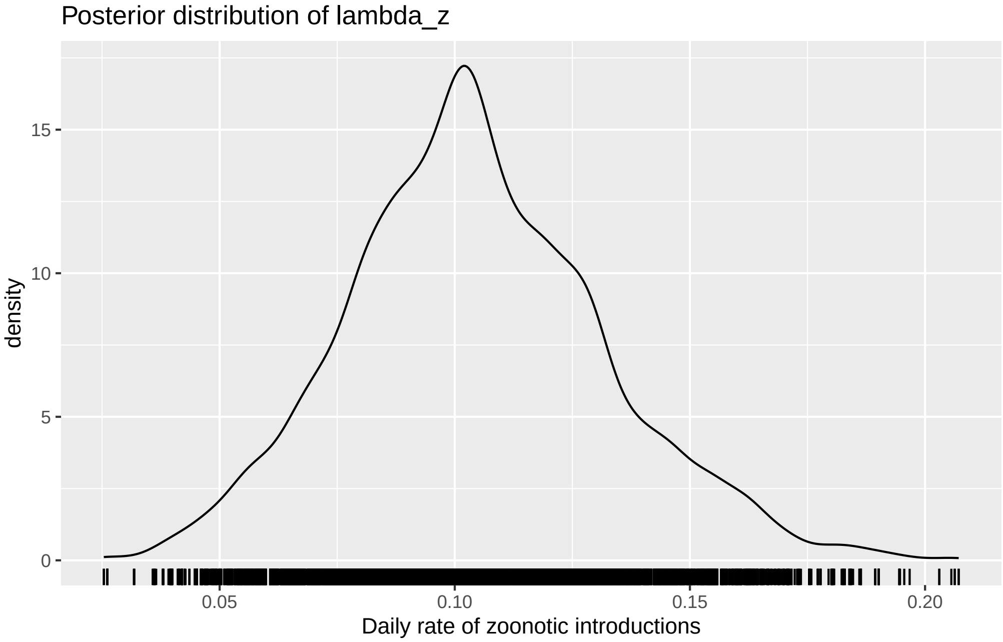
```r
ggplot(chains) +
  geom_line(aes(x = step, y = zoo, color = chain)) +
  labs(title = "Trace of lambda_z values")
```



```r
chains %>%
  filter(step > 25) %>%
  ggplot(aes(x = R)) +
  geom_density() +
  geom_rug() +
  labs(title = "Posterior distribution of R")
```


Posterior distribution of R

```
chains %>%
  filter(step > 25) %>%
  ggplot(aes(x = zoo)) +
  geom_density() +
  geom_rug() +
  labs(
    title = "Posterior distribution of lambda_z",
    x = "Daily rate of zoonotic introductions")
```


Posterior distribution of lambda_z

References

Dean, Katharine R, Fabienne Krauer, and Boris V Schmid. 2019. “Epidemiology of a Bubonic Plague Outbreak in Glasgow, Scotland in 1900.” *R Soc Open Sci* 6 (1): 181695.

Hawkes, Alan G. 1971. “Spectra of Some Self-Exciting and Mutually Exciting Point Processes.” *Biometrika* 58 (1): 83–90.

Rose, Laura J, Rodney Donlan, Shailen N Banerjee, and Matthew J Arduino. 2003. “Survival of Yersinia Pestis on Environmental Surfaces.” *Appl. Environ. Microbiol.* 69 (4): 2166–71.

Wallinga, Jacco, and Peter Teunis. 2004. “Different Epidemic Curves for Severe Acute Respiratory Syndrome Reveal Similar Impacts of Control Measures.” *Am. J. Epidemiol.* 160 (6): 509–16.

“XXII. The Epidemiological Observations Made by the Commission in Bombay City.” 1907. *Epidemiology & Infection* 7 (6): 724–98.