Revisiting the early stage of the 1720 plague epidemic in Gévaudan

Thibaut Jombart, Anne Cori, Henry Mouysset

2024-11-08

Contents

1 Preamble

In this report, we revisit 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. Historical data suggest that the epidemic was initiated by a convict who travelled from Marseille to Saint-Laurent-d'Olt, where he infected Jean Quintin, who then seeded the epidemic which would first affect his village before spreading to the rest of Gévaudan.

We re-analyse this scenario by combining historical data on the dates of deaths of the first few cases, alongside published estimates of the incubation time or infectious period distributions. A branching process model with augmented data is used for jointly estimating the effective reproduction number and the rate of zoonotic introductions in the early stages of the epidemic.

2 Estimates of key epidemiological features

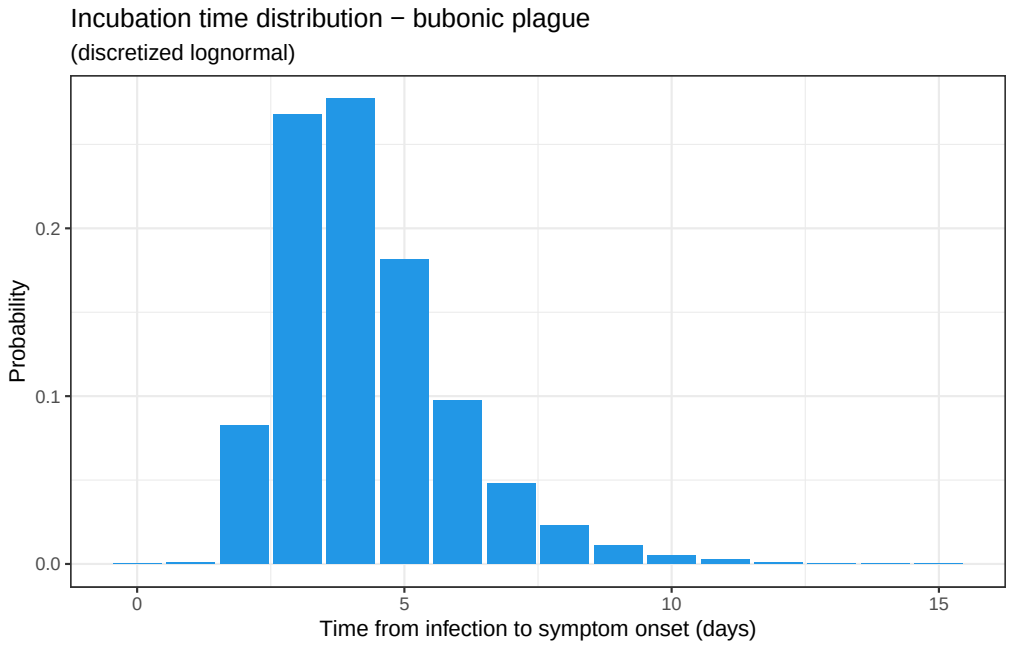
2.1 Delay distributions

2.1.1 Incubation period

The incubation period is described in multiple papers but a general consensus seems to be around 2-6 days [1–7].

We build a discretized log-normal distribution compatible with these observations:

```
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)")
```




```
## Some stats based upon 1 million draws
summary(incub$r(1e6))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   4.299   5.000  21.000
```

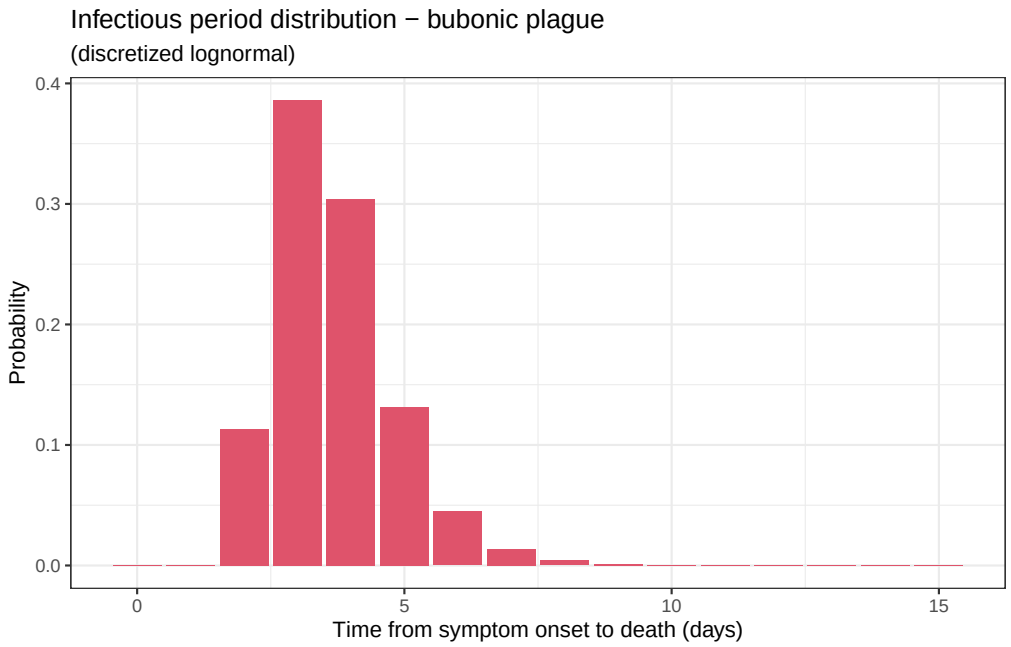
2.1.2 Infectious period

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 [1,6,8,9].

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)")
```




```
## Some stats based upon 10 million draws
summary(infec$r(1e6))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.674   4.000  15.000
```

We note that the distribution is well in line with the mean duration of fatal, untreated bubonic plague of 3.6 days [1].

```
## mean of 1 000 000 values from the distribution:
mean(infec$r(1e6))
```

```
## [1] 3.675218
```

3 Patient zero: the convict from Marseille

The theory of the convict walking all the way from Marseille with an infected bundle of wool to eventually infect Jean Quintin in Saint Laurent d'Olt is suspicious: the man would have had to not get infected during his entire trip, to then infect 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)

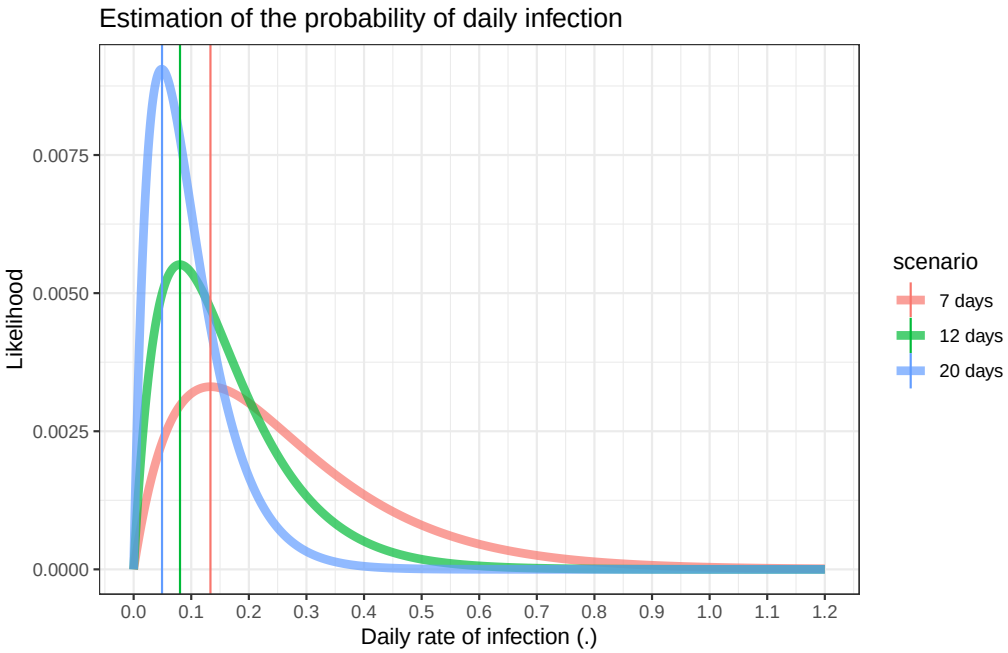
We do not know what the daily rate of infection λ from the wool bundle was, but we can derive a likelihood profile for each scenario, using:

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

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

```
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?

4 The first case after Jean

Jean Quintin 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 duration of the symptomatic 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 1e6 draws from distributions
sim_delay_inf_death <- incub$r(1e6) + infec$r(1e6)

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

```
## [1] 1.5e-05 8.0e-06 5.0e-06
```

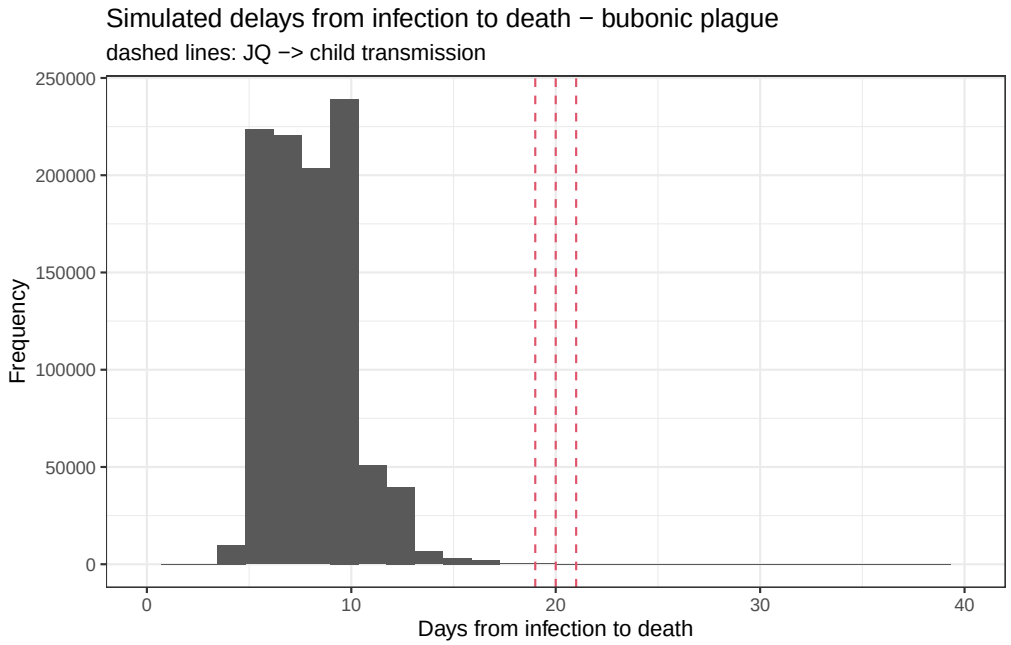
```
mean(child_delay_pval)
```

```
## [1] 9.333333e-06
```

```
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 infected by Jean Quintin. There remains the possibility that infected surfaces (clothes, bedsheets) were infected by *Y. pestis*, which has been shown to be able to last up to 5 days [10]. 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] 0.000391 0.000234 0.000130
```

```
mean(child_delay_pval_alt)
```

```
## [1] 0.0002516667
```

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** infected by his dad, but from a zoonotic source

5 Modelling the early epidemic

5.1 Epidemic curve

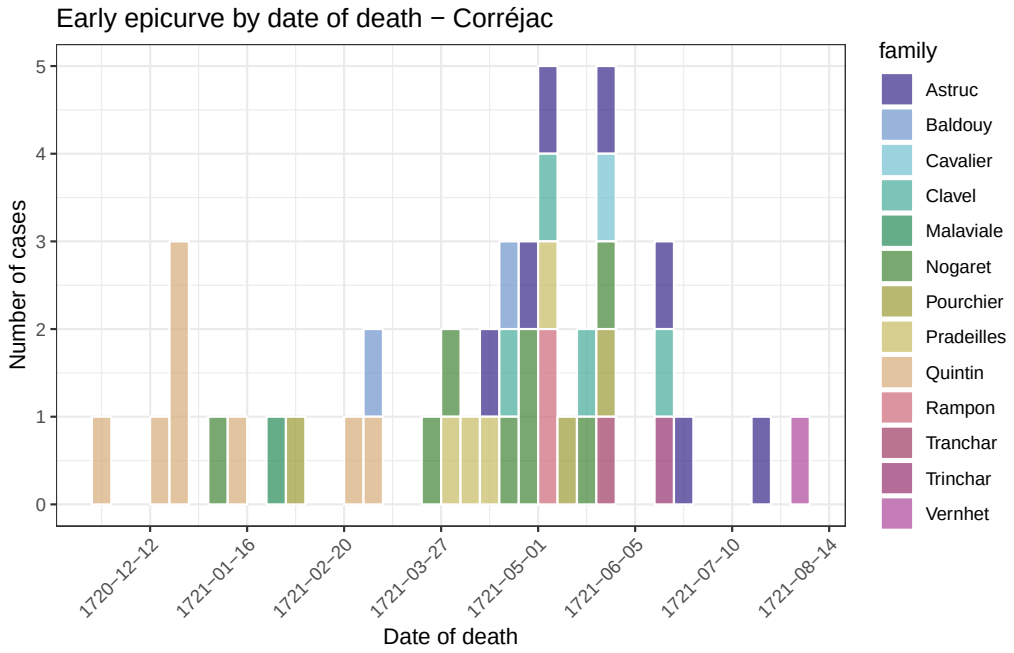
Here we analyse the linelist of the first few cases in Corréjac, followed by La Canourgue.

```r
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
```

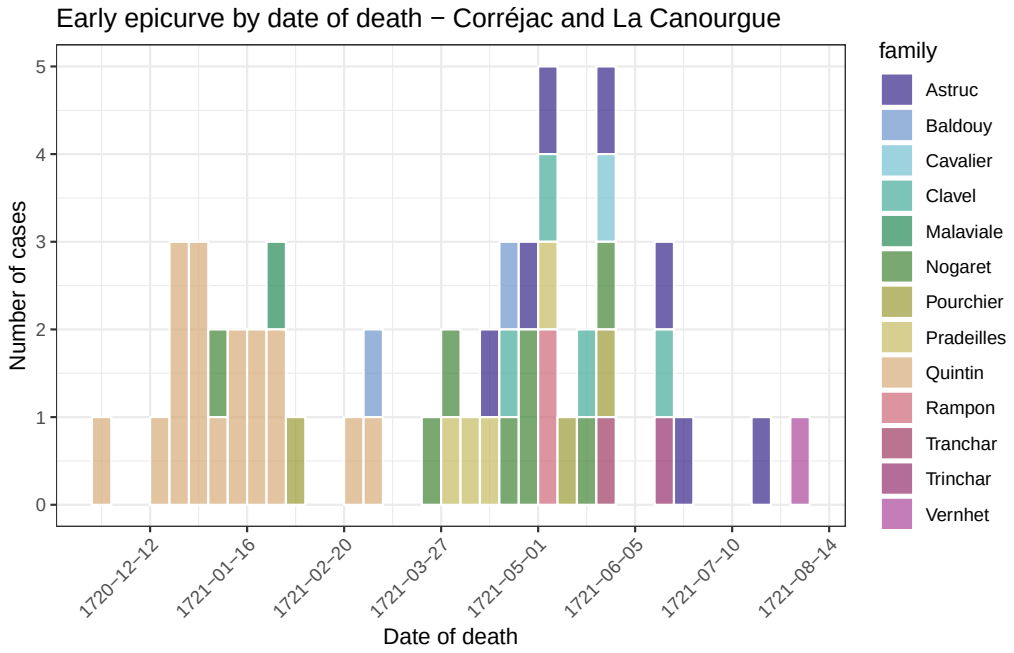
```
## # A tibble: 52 x 4
##    name                 date_of_death location family
##    <chr>                <date>        <chr>    <chr>
##  1 QUINTIN Jean         1720-11-26    Corréjac Quintin
##  2 QUINTIN Ambroise     1720-12-18    Corréjac Quintin
##  3 CASSANHES Marguerite 1720-12-25    Corréjac 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:

```r
library(incidence2)
x_cor <- x %>%
  filter(location == "Corréjac")

x_cor %>%
  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.

5.2 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 [11]. As such, we will be using a Hawkes process [12] 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.

5.2.1 Notations

5.2.1.1 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)
- S_t: the number of susceptible (*i.e.* non-infected) individuals at time t
- N: the total number of individuals (infected and non-infected) in the area considered

5.2.1.2 Distributions

- $\mathcal{F}$: probability mass function (pmf) of the infectious/symptomatic period distribution
- $\mathcal{G}$: pmf of the incubation period distribution
- $\mathcal{H}$: pmf of the generation time distribution, obtained from the convolution $\mathcal{F} * \mathcal{G} = \mathcal{H}$

5.2.1.3 Parameters

- λ_z: the rate of zoonotic introduction, assumed constant over time
- R_0: the basic reproduction number

5.2.2 The model

We use a classical Bayesian framework where the posterior distribution is defined for parameters θ and data (x) 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:

$$\begin{aligned}
p(x|\theta) &= p(d, O, I|\lambda_z, R_0) && (2)\\
&= p(d|O)p(O|I)p(I|\lambda_z, R_0) && (3)\\
&= \left(\prod_i p(d_i|O_i) \prod_i p(O_i|I_i)\right) p(I|\lambda_z, R_0) && (4)\\
&= \left(\prod_i \mathcal{F}(d_i - O_i)\mathcal{G}(O_i - I_i)\right) p(I|\lambda_z, R_0) && (5)
\end{aligned}$$

The calculation of $p(I|\lambda_z, R_0)$ is defined by the Hawkes process for the incidence Y:

$$\begin{aligned}
p(I|\lambda_z, R_0) &= p(Y|\lambda_z, R_0) && (6)\\
&= \prod_t p(Y_t|Y_1, ..., Y_{t-1}, \lambda_z, R_0) && (7)
\end{aligned}$$

The incidence Y_t is governed by:

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

where $\mathcal{P}(.)$ is the Poisson distribution, and with:

$$\lambda_t = \lambda_z + \sum_{s=1}^{t-1} R_0 \frac{S_t}{N} Y_s \mathcal{H}(t-s) \tag{9}$$

Finally, we assume independent priors for λ_z and R_0 such that:

$$p(\theta) = p(\lambda_z, R_0) = p(\lambda_z)p(R_0) \tag{10}$$

5.2.3 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_0, using Normal proposal distributions with standard deviations manually tailored to reach about 30-40% acceptance rates
4. go back to 1 until desired number of iterations reached

5.3 Distributions

5.3.1 Generation time

The generation time distribution $\mathcal{H}$ is estimated using the following procedure:

1. Sample a large number n of incubation periods X
2. Sample a large number n of infectious period durations Y
3. Sample n delays from onset to infections Z, uniformly distributed between 0 and Y
4. Derive the empirical distribution of $\mathcal{H}$ from $X + Z$

We note that this can be done using two alternative approaches:

a. sampling from the discretized distributions for steps 1-3
b. sampling from continuous distributions for steps 1-3, then discretizing

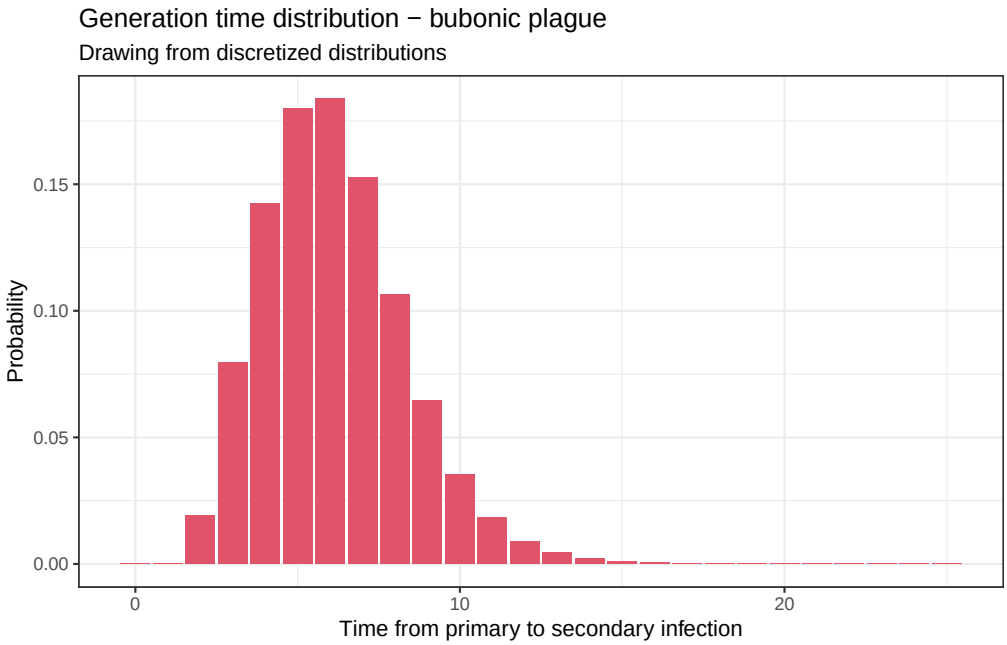
We will try both approaches out of curiosity (planting the seed of a new MRes project?) but will retain option **b** as the correct one.

We start with approach A, basing estimates on 10 million draws:

```
## RNG for delay from onset to infection
onset_to_inf_r_a <- function(n) {
  floor(runif(n, 0, infec$r(n) + 1))
}

## PMF for delays of 0, 1, ...
gentime_dat_a <- incub$r(1e7) + onset_to_inf_r_a(1e7)
range(gentime_dat_a)
```

```
## [1]  1 36
```

```
gentime_freq_a <- sapply(
  seq(0, max(gentime_dat_a), by = 1L),
  function(i) sum(i == gentime_dat_a)
)
gentime_pmf_a <- gentime_freq_a
gentime_pmf_a <- gentime_pmf_a / sum(gentime_pmf_a) # superfluous

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

## emulate rng function
gentime_r_a <- function(n) {
  sample(gentime_dat_a, size = n, replace = TRUE)
}

gentime_a <- list(d = gentime_d_a, r = gentime_r_a)

gt_dat_a <- tibble(
  Day = 0:25,
  p = gentime_a$d(0:25)
  )

gt_dat_a %>%
  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",
    subtitle = "Drawing from discretized distributions"
    )
```




```
## Summary stats of the GT - approach A
summary(gentime_dat_a)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   6.000   6.136   7.000  36.000
```

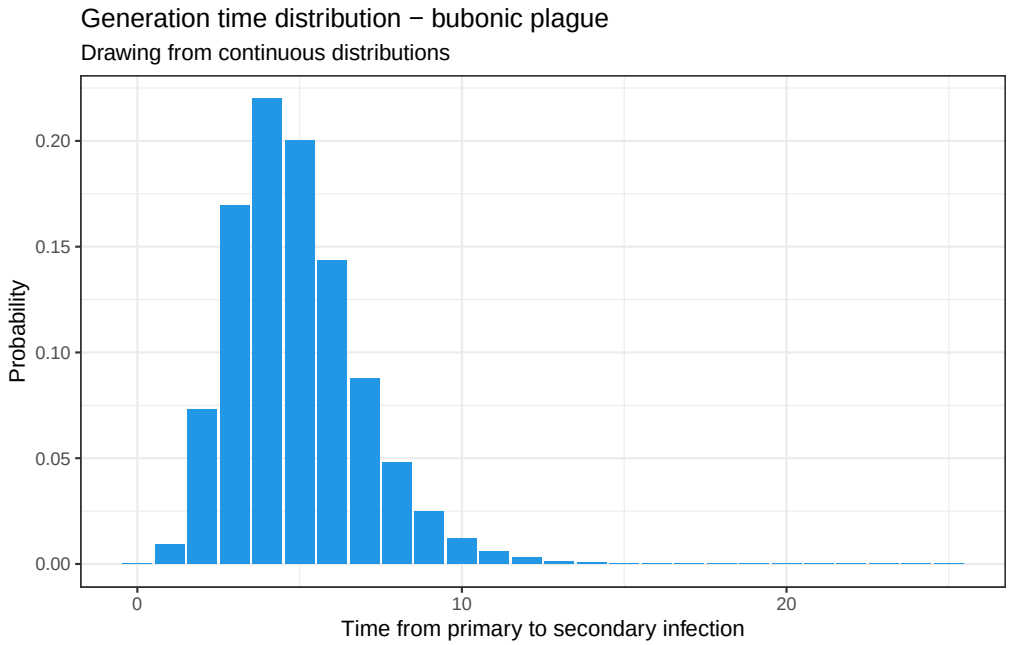
Let's now try the alternative approach 'B':

```
## Continuous version of the incubation and infectious period distributions
incub_cont_r <- function(n) rlnorm(n,
                                   meanlog = log(3.5),
                                   sdlog = log(1.5)
                                   )

infec_cont_r <- function(n) rlnorm(n,
                                   meanlog = log(3),
                                   sdlog = log(1.4)
                                   )

## RNG for delay from onset to infection
onset_to_inf_r_b <- function(n) {
  runif(n, 0, infec_cont_r(n))
}

## PMF for delays of 0, 1, ...
gentime_dat_b <- floor(incub_cont_r(1e7) + onset_to_inf_r_b(1e7))
range(gentime_dat_b)
```

```
## [1]  0 31
```

```
gentime_freq_b <- sapply(
  seq(0, max(gentime_dat_b), by = 1L),
  function(i) sum(i == gentime_dat_b)
)
gentime_pmf_b <- gentime_freq_b
```

```r
gentime_pmf_b[1] <- 0 # force p(0) = 0
gentime_pmf_b <- gentime_pmf_b / sum(gentime_pmf_b) # superfluous

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

## emulate rng function
gentime_r_b <- function(n) {
  sample(gentime_dat_b, size = n, replace = TRUE)
}

gentime_b <- list(d = gentime_d_b, r = gentime_r_b)

gt_dat_b <- tibble(
  Day = 0:25,
  p = gentime_b$d(0:25)
  )

gt_dat_b %>%
  ggplot(aes(x = Day, y = p)) +
  geom_col(fill = 4) +
  theme_bw() +
  labs(
    x = "Time from primary to secondary infection",
    y = "Probability",
    title = "Generation time distribution - bubonic plague",
    subtitle = "Drawing from continuous distributions"
    )
```

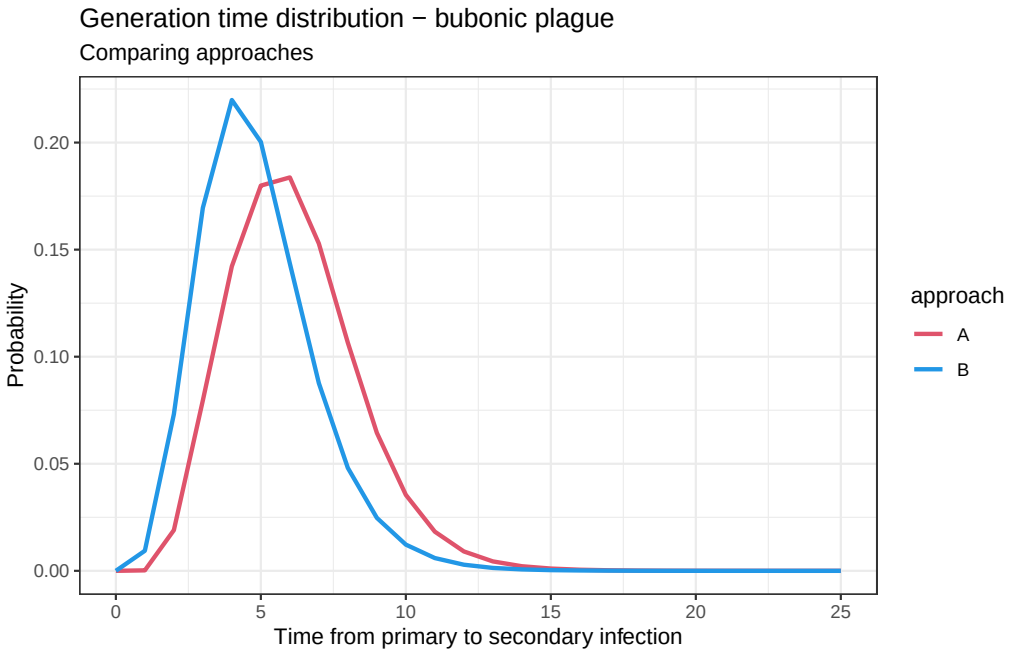



```
## Summary stats of the GT - approach B
summary(gentime_dat_b)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   5.000   4.888   6.000  31.000
```

Comparison of the two approaches:

```
gentime_dat_comp <- bind_rows(
    cbind(gt_dat_a, approach = "A"),
    cbind(gt_dat_b, approach = "B")
  )
gentime_dat_comp %>%
  ggplot(aes(x = Day, y = p, color = approach)) +
  geom_line(linewidth = 1) +
  scale_color_manual(values = c(A = 2, B = 4)) +
  theme_bw() +
  labs(
    x = "Time from primary to secondary infection",
    y = "Probability",
    title = "Generation time distribution - bubonic plague",
    subtitle = "Comparing approaches"
    )
```




```
## stats of the two distributions
summary(gentime_dat_a)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   6.000   6.136   7.000  36.000
```

```
summary(gentime_dat_b)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   5.000   4.888   6.000  31.000
```

```
## differences in infectious periods discretized/continous
summary(infec$r(1e6)) # discrete
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.676   4.000  15.000
```

```
summary(infec_cont_r(1e6)) # continuous
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5968  2.3907  2.9992  3.1749  3.7657 14.4213
```

We choose the 2nd approach:

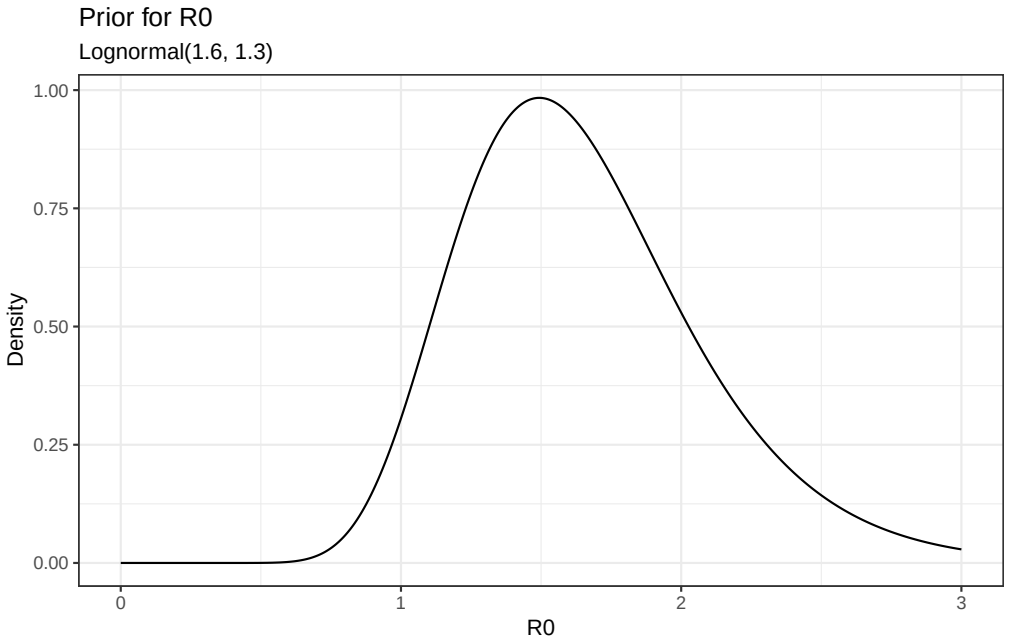
```
gentime <- gentime_b
```

5.3.2 Priors

We use a prior for R derived from [8] as a lognormal distribution with mean 1.6 and standard deviation 1.3:

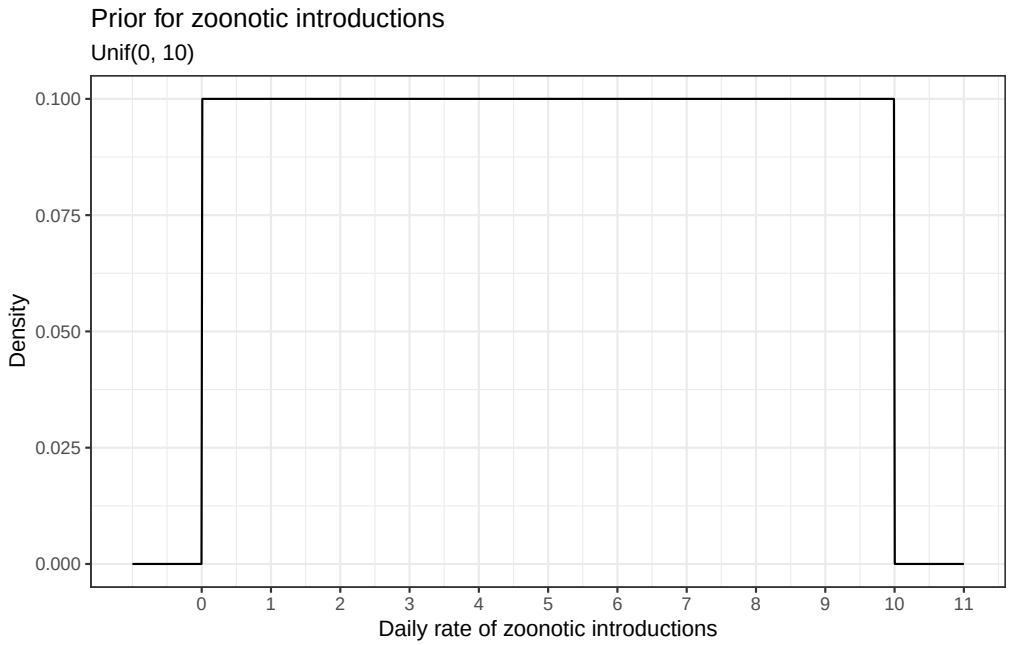
```
prior_R <- function(x, log = FALSE) dlnorm(x, log(1.6), log(1.3), log = log)
prior_R_dat <- tibble(
  x = seq(0, 3, length.out = 1000),
  d = prior_R(seq(0, 3, length.out = 1000))
  )

ggplot(prior_R_dat, aes(x = x, y = d)) +
```

```
  geom_line() +
  theme_bw() +
  labs(
    x = "R0",
    y = "Density",
    title = "Prior for R0",
    subtitle = "Lognormal(1.6, 1.3)"
  )
```



Rates of zoonotic introductions are usually harder to estimate, so we use a flat, uninformative priors for λ_z, uniformly distributed from 0 to 10:

```
prior_zoo <- function(x, log = FALSE) dunif(x, 0, 10, log = log)

prior_zoo_dat <- tibble(
  x = seq(-1, 11, length.out = 1000),
  d = prior_zoo(seq(-1, 11, length.out = 1000))
  )

ggplot(prior_zoo_dat, aes(x = x, y = d)) +
  geom_line() +
  theme_bw() +
  labs(
    x = "Daily rate of zoonotic introductions",
    y = "Density",
    title = "Prior for zoonotic introductions",
    subtitle = "Unif(0, 10)"
  ) +
  scale_x_continuous(breaks = 0:11)
```



5.4 Implementation

The following code implements the model. Since Corréjac and La Canourgue had vastly different populations (respectively 111 and 1370 inhabitants), we restrict the analysis of the early stage of the outbreak to Corréjac. Note that for such as small population, accounting for the depletion of susceptibles is a key feature of our 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,
                          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,
                            pop_size = 111) {

  ### 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)
  )

  #### calculate the corresponding number of susceptibles over time
  total_inf <- cumsum(incid)
  if (any(total_inf > pop_size)) stop("total infected > pop_size")
  total_sus <- pop_size - total_inf
  p_sus <- total_sus / pop_size

  #### 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_sus)

  #### we ommit the first entry as it is by definition NA
  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,
                         params,
                         d_incub = incub$d,
                         d_infec = infec$d,
                         d_gentime = gentime$d,
                         d_zoo = prior_zoo,
                         d_R = prior_R,
                         pop_size = 111
                         ) {
  compute_loglike(data, params, d_incub, d_infec, d_gentime, pop_size) +
    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,
                            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),
                            pop_size = 111) {

  ### 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,
                         pop_size)
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,
                               pop_size)

  #### 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,
                           pop_size)
  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,
                             pop_size)
    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 to check all runs smoothly:

```
system.time(res <- estimate_params(
  x_cor$date_of_death,
  pop_size = 111,
  n_iter = 30)
)
```

```
##    user  system elapsed
##   4.276   0.000   4.277
```

```
head(res)
```

```
## $mcmc
##    step      post        zoo         R
## 1     1 -277.5824 0.21940831 0.2553579
## 2     2 -297.7359 0.07991889 0.2553579
## 3     3 -286.8339 0.07991889 0.2553579
## 4     4 -281.6622 0.10582099 0.2553579
```

```
## 5     5 -278.5713 0.14419924 0.2553579
## 6     6 -287.4241 0.16141774 0.2553579
## 7     7 -289.7483 0.16141774 0.2670288
## 8     8 -258.7166 0.14104295 0.5711002
## 9     9 -268.4876 0.12042883 0.5711002
## 10   10 -278.5904 0.12042883 0.5711002
## 11   11 -265.9056 0.06940813 0.7434928
## 12   12 -274.7668 0.14207337 0.7434928
## 13   13 -263.6828 0.10996631 0.7434928
## 14   14 -280.8993 0.10996631 0.7434928
## 15   15 -266.6163 0.06573591 0.9181644
## 16   16 -264.5206 0.06573591 1.0173176
## 17   17 -278.1260 0.06573591 0.7172216
## 18   18 -265.0931 0.06573591 0.7172216
## 19   19 -265.4927 0.06573591 0.7417444
## 20   20 -264.7431 0.06573591 0.7417444
## 21   21 -275.2106 0.06573591 1.3283508
## 22   22 -273.1222 0.06573591 1.0484876
## 23   23 -267.8429 0.06573591 0.9212117
## 24   24 -264.4725 0.06573591 0.9212117
## 25   25 -268.2550 0.06573591 1.1302763
## 26   26 -263.4825 0.06573591 1.1302763
## 27   27 -264.2256 0.06573591 1.1302763
## 28   28 -249.0580 0.06573591 1.1302763
## 29   29 -284.6914 0.06573591 1.1302763
## 30   30 -282.3666 0.06573591 0.8306455
##
## $accept_zoo
## [1] 0.3666667
##
## $accept_R
## [1] 0.4333333
```

```
tail(res)
```

```
## $mcmc
##    step      post        zoo         R
## 1     1 -277.5824 0.21940831 0.2553579
## 2     2 -297.7359 0.07991889 0.2553579
## 3     3 -286.8339 0.07991889 0.2553579
## 4     4 -281.6622 0.10582099 0.2553579
## 5     5 -278.5713 0.14419924 0.2553579
## 6     6 -287.4241 0.16141774 0.2553579
## 7     7 -289.7483 0.16141774 0.2670288
## 8     8 -258.7166 0.14104295 0.5711002
## 9     9 -268.4876 0.12042883 0.5711002
## 10   10 -278.5904 0.12042883 0.5711002
## 11   11 -265.9056 0.06940813 0.7434928
## 12   12 -274.7668 0.14207337 0.7434928
## 13   13 -263.6828 0.10996631 0.7434928
## 14   14 -280.8993 0.10996631 0.7434928
## 15   15 -266.6163 0.06573591 0.9181644
## 16   16 -264.5206 0.06573591 1.0173176
## 17   17 -278.1260 0.06573591 0.7172216
## 18   18 -265.0931 0.06573591 0.7172216
```

```
## 19   19 -265.4927 0.06573591 0.7417444
## 20   20 -264.7431 0.06573591 0.7417444
## 21   21 -275.2106 0.06573591 1.3283508
## 22   22 -273.1222 0.06573591 1.0484876
## 23   23 -267.8429 0.06573591 0.9212117
## 24   24 -264.4725 0.06573591 0.9212117
## 25   25 -268.2550 0.06573591 1.1302763
## 26   26 -263.4825 0.06573591 1.1302763
## 27   27 -264.2256 0.06573591 1.1302763
## 28   28 -249.0580 0.06573591 1.1302763
## 29   29 -284.6914 0.06573591 1.1302763
## 30   30 -282.3666 0.06573591 0.8306455
##
## $accept_zoo
## [1] 0.3666667
##
## $accept_R
## [1] 0.4333333
```

5.5 Resuts for Corréjac

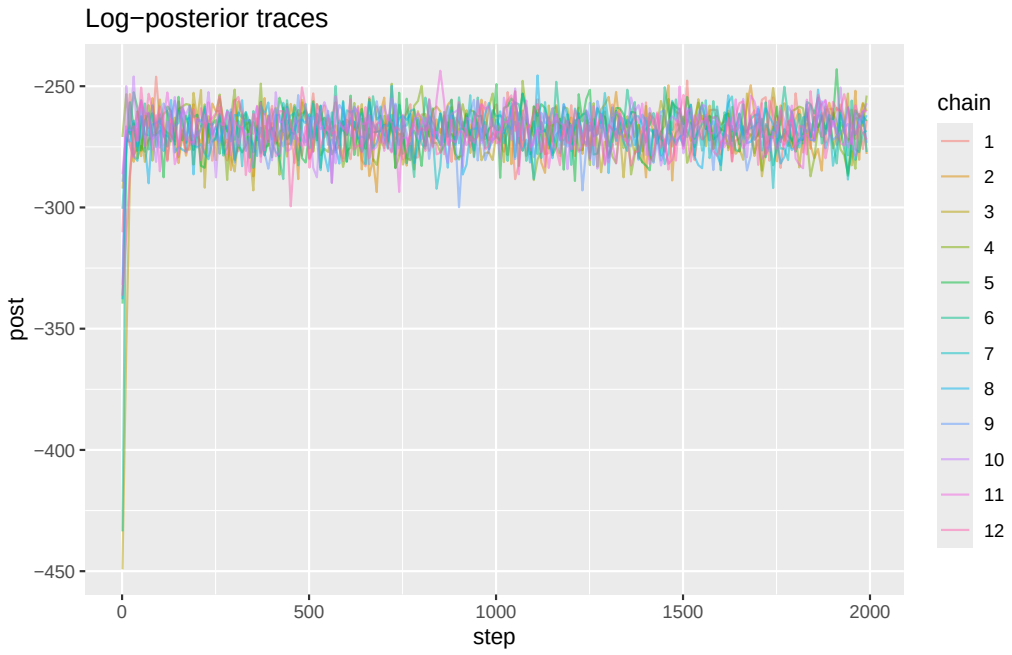
Parameters are estimated for all cases from Corréjac, using 12 separate chains run in parallel (works only on linux), with a burn-in of 100 iterations. For simplicity, chains have been saved in a separate RDS file.

```
n_iter <- 2000

## library(parallel)
## res <- mclapply(1:12, function(i)
##   cbind.data.frame(
##     estimate_params(
##       x_cor$date_of_death,
##       pop_size = 111,
##       n_iter = n_iter),
##     chain = i),
##   mc.cores = 12
##   )
## saveRDS(res, file = "res_2000iter_12chains_correjac.rds")

res <- readRDS("res_2000iter_12chains_correjac.rds")

## Some diagnostics
library(coda)
lapply(res, function(e) effectiveSize(mcmc(e))) %>%
  bind_rows()
```

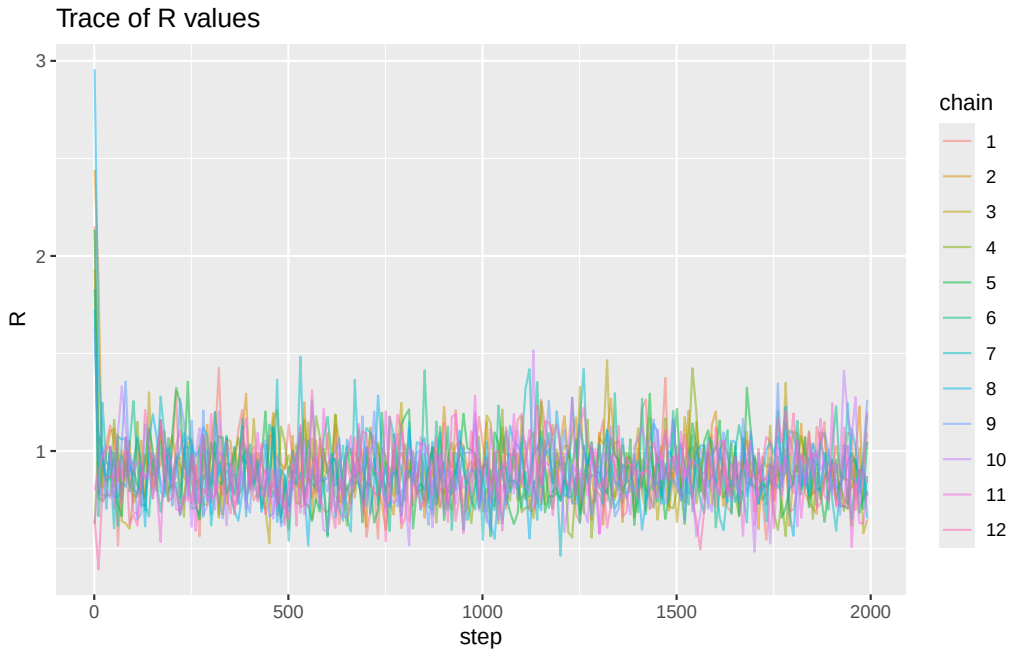
```
## # A tibble: 12 x 7
##    mcmc.step mcmc.post mcmc.zoo mcmc.R accept_zoo accept_R chain
##        <dbl>     <dbl>    <dbl>  <dbl>      <dbl>    <dbl> <dbl>
##  1         0     1861.     261.   412.          0        0     0
##  2         0     1298.     318.   233.          0        0     0
##  3         0      161.     103.   288.          0        0     0
##  4         0     2000      341.   294.          0        0     0
##  5         0     1480.     302.   318.          0        0     0
##  6         0      395.     221.   368.          0        0     0
##  7         0     2000.     404.   406.          0        0     0
```

```
##  8         0      857.     313.   193.          0        0     0
##  9         0     2000.     246.   274.          0        0     0
## 10         0     1471.     328.   362.          0        0     0
## 11         0     1860.     326.   435.          0        0     0
## 12         0      500.     147.   384.          0        0     0
```

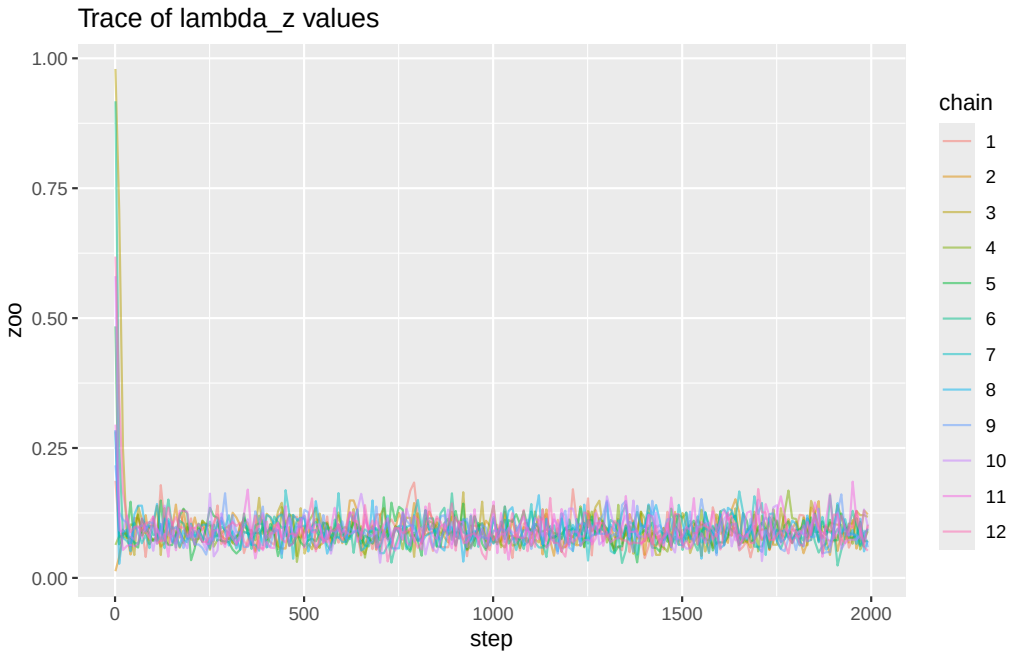
```
## Putting chains together and thining
## Given the reported ESS a thining of 1/10 seems reasonable
to_keep <- seq(from = 1, to = n_iter, by = 10)
chains <- Reduce(rbind, lapply(1:length(res), function(i) res[[i]][to_keep, ]))
names(chains) <- gsub("mcmc.", "", names(chains))
chains$chain <- factor(chains$chain)


## Plots
ggplot(chains) +
  geom_line(aes(x = step, y = post, color = chain), alpha = 0.5) +
  labs(title = "Log-posterior traces")
```




```
## Plots
ggplot(chains) +
  geom_line(aes(x = step, y = R, color = chain), alpha = 0.5) +
  labs(title = "Trace of R values")
```

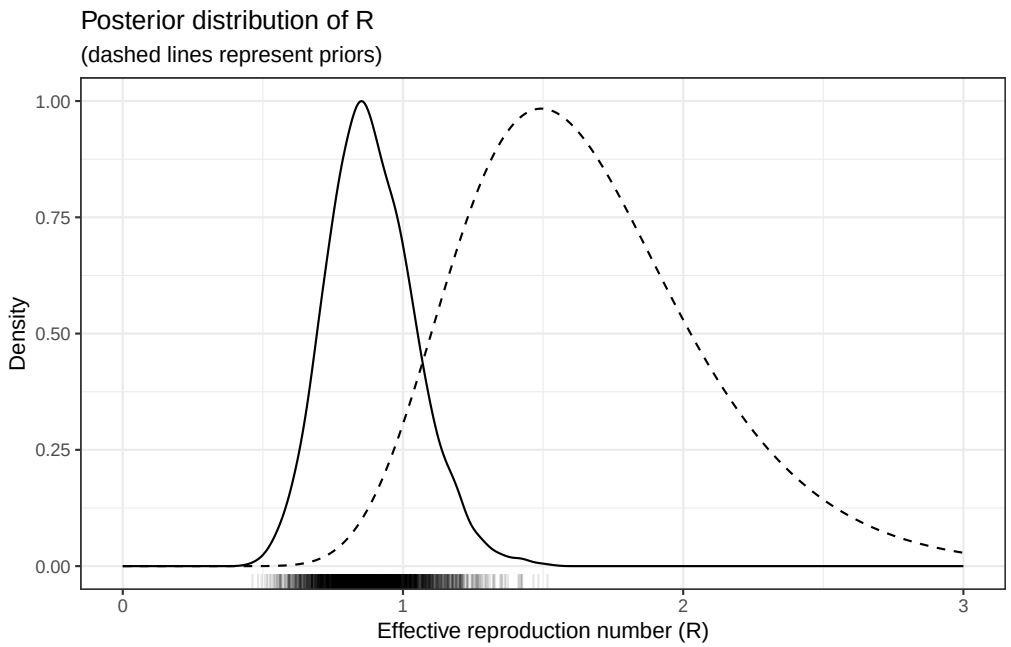



```
ggplot(chains) +
  geom_line(aes(x = step, y = zoo, color = chain), alpha = 0.5) +
  labs(title = "Trace of lambda_z values")
```

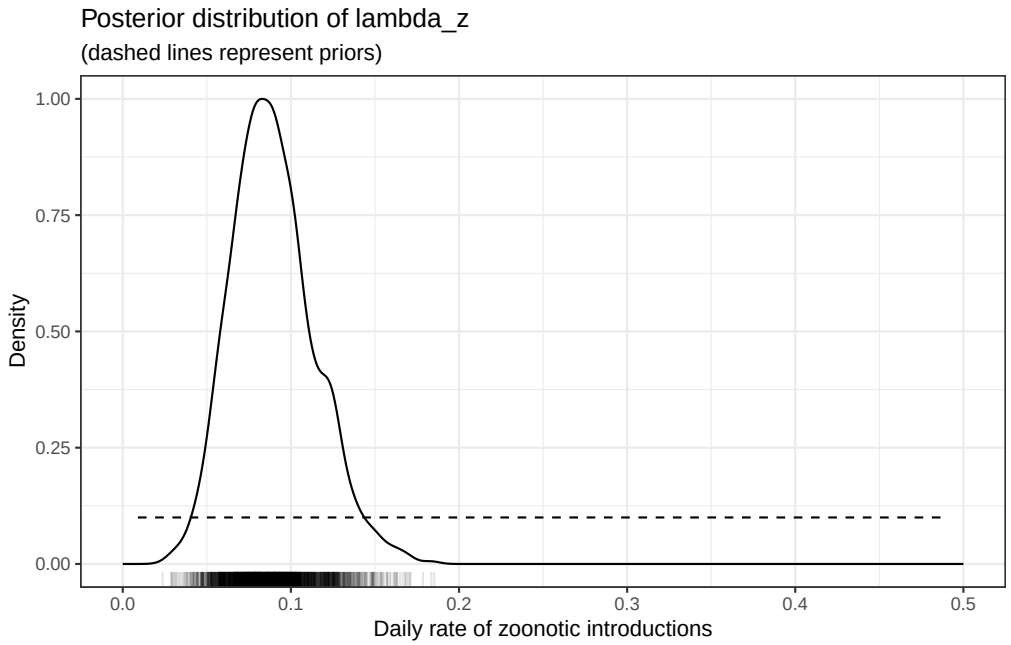



```
chains %>%
  filter(step > 100) %>%
  ggplot(aes(x = R)) +
  stat_density(aes(y = after_stat(scaled)), geom = "line") +
  geom_rug(alpha = .1) +
  geom_line(data = prior_R_dat, aes(x = x, y = d), linetype = 2) +
  theme_bw() +
  labs(
```

```
    x = "Effective reproduction number (R)",
    y = "Density",
    title = "Posterior distribution of R",
    subtitle = "(dashed lines represent priors)")
```




```
chains %>%
  filter(step > 100) %>%
  ggplot(aes(x = zoo)) +
  stat_density(aes(y = after_stat(scaled)), geom = "line") +
  geom_rug(alpha = .1) +
  geom_line(data = prior_zoo_dat, aes(x = x, y = d), linetype = 2) +
  theme_bw() +
  labs(
    x = "Daily rate of zoonotic introductions",
    y = "Density",
    title = "Posterior distribution of lambda_z",
    subtitle = "(dashed lines represent priors)") +
  xlim(0, 0.5)
```



We can report the mean, median and 95% CrI for each parameter:

```
chains_smry <- chains %>%
  filter(step > 100) %>%
  select(R, zoo) %>%
  lapply(function(x) data.frame(
    mean = mean(x),
    median = median(x),
    CrI_low = quantile(x, 0.025),
    CrI_up = quantile(x, 0.975))
    ) %>%
  bind_rows()

rownames(chains_smry) <- c("R", "Zoo")
chains_smry
```

```
##           mean     median    CrI_low    CrI_up
## R   0.88832734 0.87628125 0.61252164 1.2132946
## Zoo 0.08942776 0.08737024 0.04702142 0.1434559
```

We can also check acceptance rates:

```
chains %>%
  filter(step > 100) %>%
  select(accept_R, accept_zoo) %>%
  summarise_all(mean)
```

```
##   accept_R accept_zoo
## 1 0.397375     0.2755
```

We finally check the proportion of R_0 above 1:

```
p_R_above_1 <- chains %>%
  filter(step > 100) %>%
  summarise(mean(R > 1))
p_R_above_1
```

```
##   mean(R > 1)
## 1   0.2263158
```

Results suggest that:

- the data is informative on R_0 and λ_z, with posterior distributions well different from the priors
- the average value of estimated R_0 is 0.89 (95% CrI: 0.61 - 1.21), with only 0.23% of posterior values above 1, suggesting that person-to-person transmission only is unlikely to have sustained the epidemic
- the estimated rate of zoonotic introductions confirms that re-introduction played a substantial role in transmission, with a mean rate of introduction of 0.09 (95% CrI: 0.05 - 0.14), corresponding to an average of one introduction every 11.2 days

5.6 Resuts on all data

As a sensitivity study, we re-run the same analyses but including all cases.

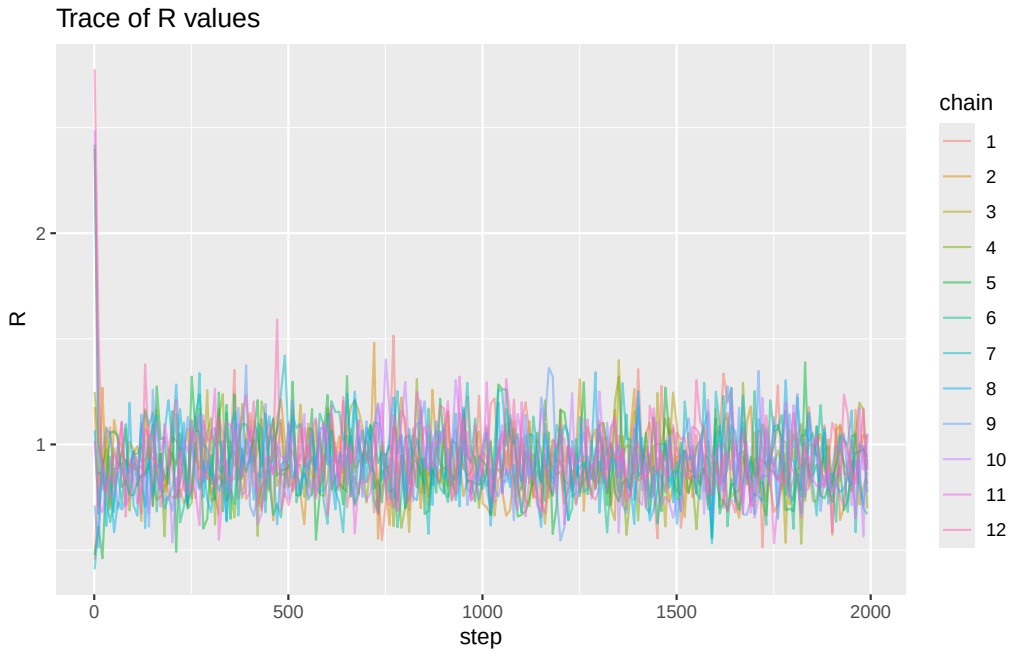
```
n_iter <- 2000

## library(parallel)
## res <- mclapply(1:12, function(i)
##   cbind.data.frame(
##     estimate_params(
##       x$date_of_death,
##       pop_size = 111,
##       n_iter = n_iter),
##     chain = i),
##   mc.cores = 12
##   )
## saveRDS(res, file = "res_2000iter_12chains_all_cases.rds")

res <- readRDS("res_2000iter_12chains_all_cases.rds")

## Some diagnostics
lapply(res, function(e) effectiveSize(mcmc(e))) %>%
  bind_rows()
```

```
## # A tibble: 12 x 7
##    mcmc.step mcmc.post mcmc.zoo mcmc.R accept_zoo accept_R chain
##        <dbl>     <dbl>    <dbl>  <dbl>      <dbl>    <dbl> <dbl>
##  1         0     2177.     269.   357.          0        0     0
##  2         0     2000.     299.   334.          0        0     0
##  3         0     1224.     297.   385.          0        0     0
##  4         0     1731.     379.   395.          0        0     0
##  5         0      202.     103.   269.          0        0     0
##  6         0      597.     225.   308.          0        0     0
##  7         0      899.     167.   302.          0        0     0
##  8         0     2000.     240.   406.          0        0     0
##  9         0      428.     110.   356.          0        0     0
## 10         0     1345.     323.   234.          0        0     0
## 11         0     2000      304.   420.          0        0     0
## 12         0     1134.     341.   348.          0        0     0
```

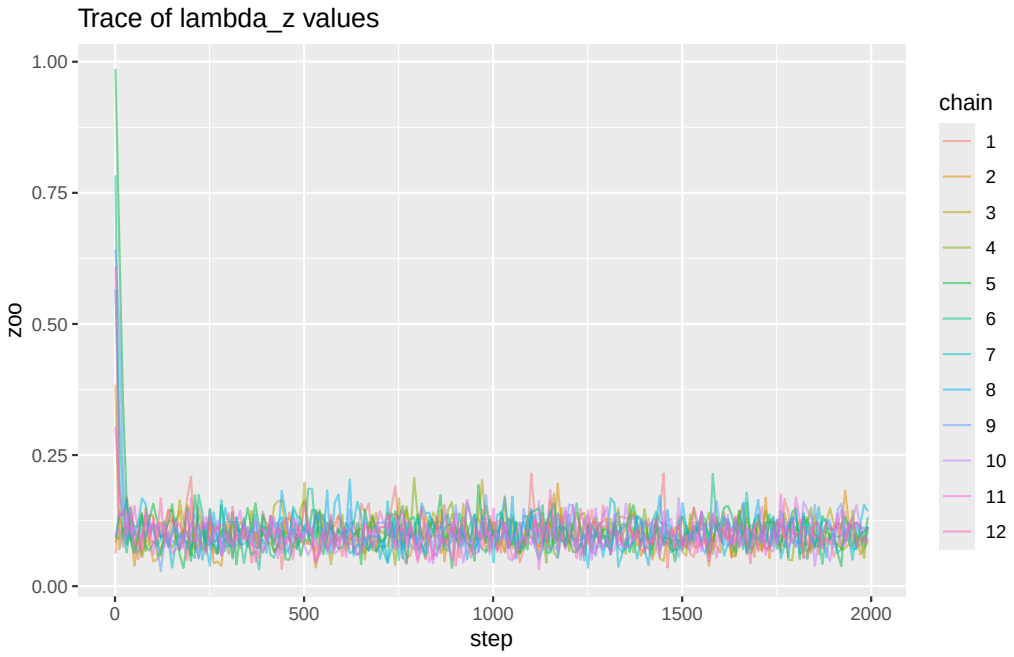
```
## Putting chains together and thining
## Given the reported ESS a thining of 1/10 seems reasonable
to_keep <- seq(from = 1, to = n_iter, by = 10)
```

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


## Plots
ggplot(chains) +
  geom_line(aes(x = step, y = post, color = chain), alpha = 0.5) +
  labs(title = "Log-posterior traces")
```


```r
## Plots
ggplot(chains) +
  geom_line(aes(x = step, y = R, color = chain), alpha = 0.5) +
  labs(title = "Trace of R values")
```

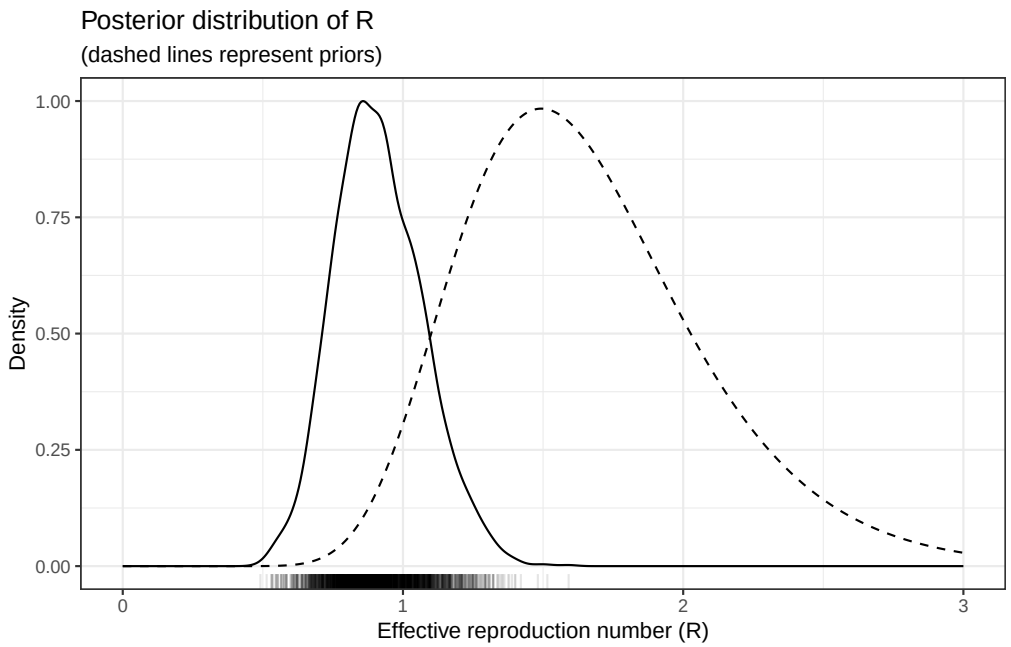



```
ggplot(chains) +
  geom_line(aes(x = step, y = zoo, color = chain), alpha = 0.5) +
  labs(title = "Trace of lambda_z values")
```

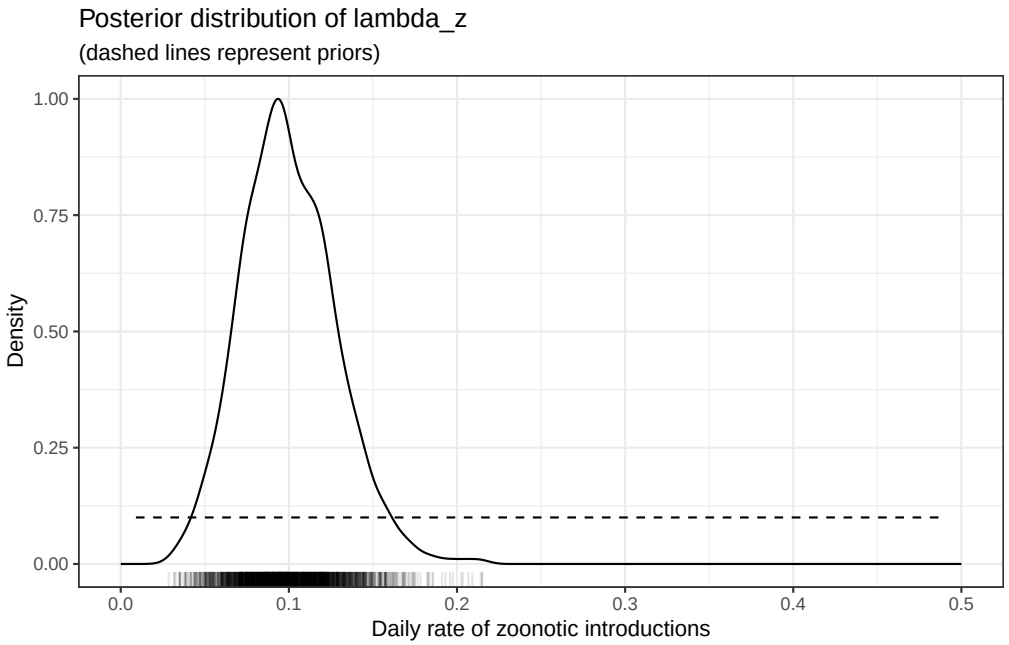



```
chains %>%
  filter(step > 100) %>%
  ggplot(aes(x = R)) +
  stat_density(aes(y = after_stat(scaled)), geom = "line") +
  geom_rug(alpha = .1) +
  geom_line(data = prior_R_dat, aes(x = x, y = d), linetype = 2) +
  theme_bw() +
  labs(
```

```
    x = "Effective reproduction number (R)",
    y = "Density",
    title = "Posterior distribution of R",
    subtitle = "(dashed lines represent priors)")
```




```
chains %>%
  filter(step > 100) %>%
  ggplot(aes(x = zoo)) +
  stat_density(aes(y = after_stat(scaled)), geom = "line") +
  geom_rug(alpha = .1) +
  geom_line(data = prior_zoo_dat, aes(x = x, y = d), linetype = 2) +
  theme_bw() +
  labs(
    x = "Daily rate of zoonotic introductions",
    y = "Density",
    title = "Posterior distribution of lambda_z",
    subtitle = "(dashed lines represent priors)") +
  xlim(0, 0.5)
```



We can report the mean, median and 95% CrI for each parameter:

```
chains_smry <- chains %>%
  filter(step > 100) %>%
  select(R, zoo) %>%
  lapply(function(x) data.frame(
    mean = mean(x),
    median = median(x),
    CrI_low = quantile(x, 0.025),
    CrI_up = quantile(x, 0.975))
    ) %>%
  bind_rows()

rownames(chains_smry) <- c("R", "Zoo")
chains_smry
```

```
##           mean     median    CrI_low    CrI_up
## R   0.91414122 0.90332743 0.63381058 1.2496595
## Zoo 0.09993969 0.09781525 0.05008894 0.1585335
```

We can also check acceptance rates:

```
chains %>%
  filter(step > 100) %>%
  select(accept_R, accept_zoo) %>%
  summarise_all(mean)
```

```
##    accept_R accept_zoo
## 1 0.3980833  0.2979583
```

We finally check the proportion of R_0 above 1:

```
p_R_above_1 <- chains %>%
  filter(step > 100) %>%
  summarise(mean(R > 1))
p_R_above_1
```

```
##   mean(R > 1)
## 1   0.2877193
```

Results of the sensitivity analysis on all data:

- the average value of estimated R_0 using all cases is 0.91 (95% CrI: 0.63 - 1.25), with 0.29% of posterior values above 1
- the mean rate of zoonotic introductions estimated on all data was 0.1 (95% CrI: 0.05 - 0.16), corresponding to an average of one introduction every 10 days

6 Conclusions

The statistical analysis of historical data suggests the established scenario of how the Plague epidemic in Gévaudan started may not be accurate. In short:

- It is rather unlikely that an individual would have travel with an infected wood bundle without getting infected the entire trip, to then infect a new person overnight.
- The delays between Jean Quintin's infection and his son's death are not compatible with direct transmission.
- Epidemic modelling suggests that person-to-person transmission was insufficient to sustain the epidemic, and that zoonotic transmission, while low, indeed played a role in the epidemic.

References

1. XXII. The epidemiological observations made by the commission in bombay city. Epidemiology & Infection. 1907;7: 724–798.

2. Dennis DT, Mead PS. Plague. Richard L, Guerrant, editors. Tropical Infectious Diseases. 2006; 471–481.

3. Kitasato S. THE BACILLUS OF BUBONIC PLAGUE. Lancet. 1894;144: 428–430.

4. Clemow F. The incubation period of plague. Lancet. 1900;155: 1508–1510.

5. Siegrist M. Bubonic plague: History and epidemiology. International Journal of Global Health and Health Disparities. 2009;6: 132–142.

6. Walløe L. 3 medieval and modern bubonic plague: Some clinical continuities. Med Hist. 2008;52: 59–73.

7. Scott S, Duncan CJ. Biology of plagues: Evidence from historical populations. Cambridge University Press; 2001.

8. Dean KR, Krauer F, Schmid BV. Epidemiology of a bubonic plague outbreak in glasgow, scotland in 1900. R Soc Open Sci. 2019;6: 181695.

9. McEvedy C. The bubonic plague. Sci Am. 1988;258: 118–123.

10. Rose LJ, Donlan R, Banerjee SN, Arduino MJ. Survival of yersinia pestis on environmental surfaces. Appl Environ Microbiol. 2003;69: 2166–2171.

11. Wallinga J, Teunis P. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. Am J Epidemiol. 2004;160: 509–516.

12. Hawkes AG. Spectra of some self-exciting and mutually exciting point processes. Biometrika. 1971;58: 83–90.