Johannes Bracher (johannes.bracher@kit.edu) 15 April 2024
NOZGEKA infectious disease modelling short course

Exercise sheet *Nowcasting, short-term forecasting and R-values*

Background: We will analyse a subset of the data set on the 2003 SARS epidemic in Hong Kong contained in the EpiEstim R package (type `?SARS2003` in `R`); see also Cori et al (2009, `https://doi.org/10.1371/journal.pcbi.1000471`). Some parts of this exercise sheet, like the reporting triangle, are not actually available and have been generated artificially by me for exercise purposes.

Access to files: Please download the files made available at `https://github.com/jbracher/NOZGEKA`. You can do that by clicking on **Code** → **Download ZIP**.

Use of software: I tried to conceive this exercise sheet in a way that it can be solved using `R` or purely the EpiEstim web app and a calculator. You may choose your preferred tool. Solutions refer to `R`, but the results should be the same using the web app.

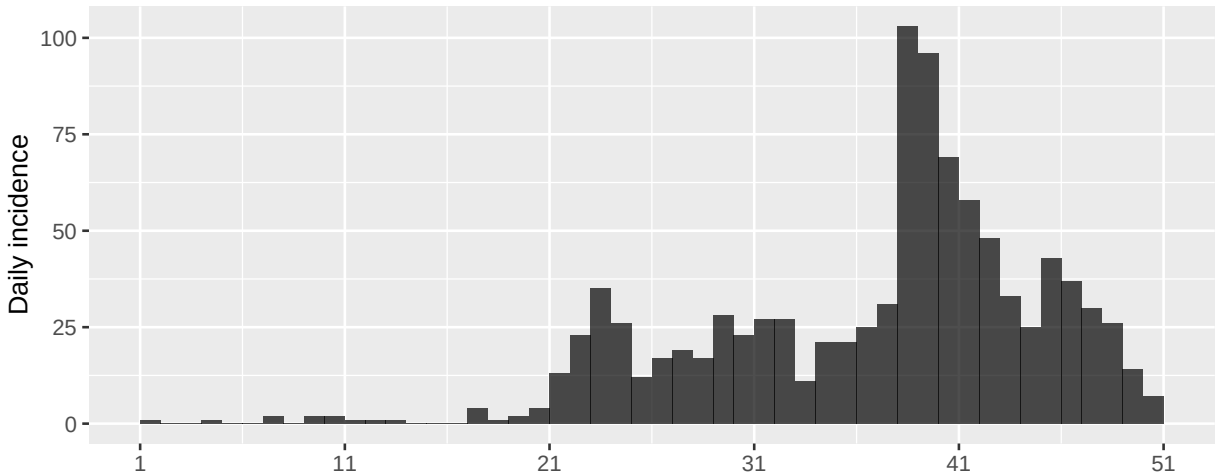
Solutions: Solutions are available in the file `exercise1_solution.pdf`. Do not hesitate to consult the solution if you are stuck somewhere.

1. Read the file `data_day_50_realtime.csv` into `R` or the EpiEstim web app at `https://shiny.dide.imperial.ac.uk/epiestim/`. Note: the latter may throw a security warning in your browser.
   If you are using the web app: You will need to tick the box "Header" when using the provided file. Also, you will only get a visualization once you are also estimating R_t. To this end you need to select "Distributional Estimate" and "Parametric without uncertainty (offset gamma)" in order to be able to specify the generation time distribution via its mean and standard deviation.

2. Generate a visualization (epicurve) of the data.

   Solution:

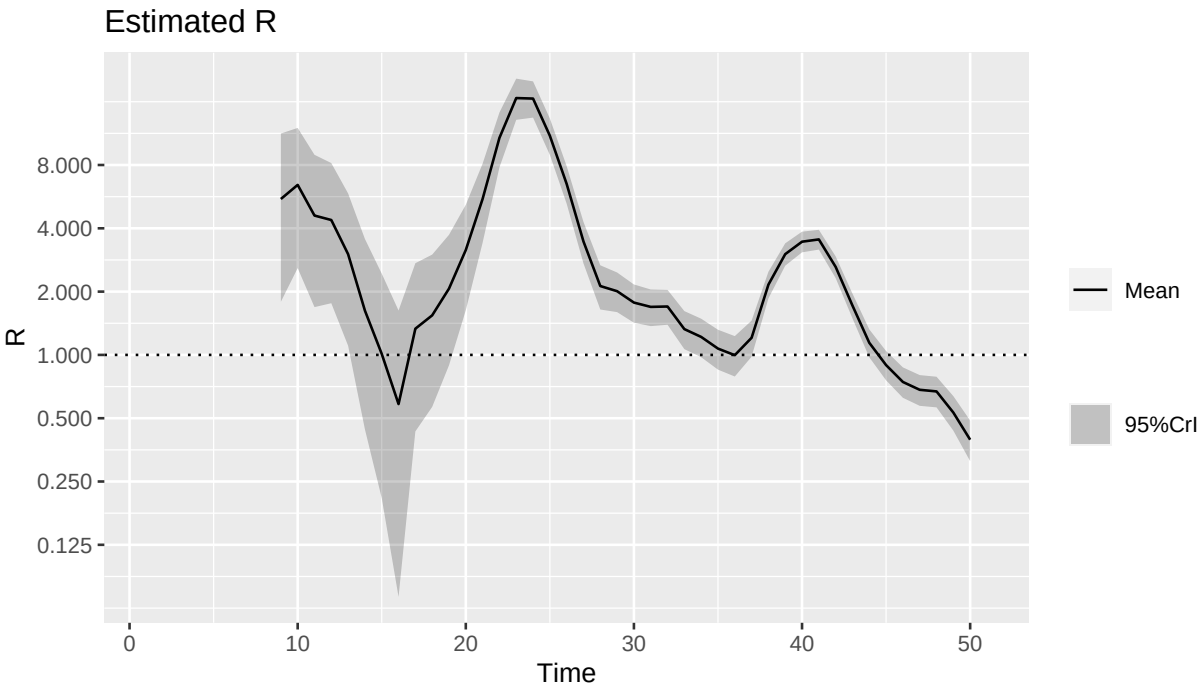
```
library(incidence)
inc <- read.csv("data_day50_realtime.csv")
# generate an incidence object which is easy to plot:
inc_to_plot <- as.incidence(inc$I)
plot(inc_to_plot)
```



3. Load the package `EpiEstim` and estimate the effective reproductive number R_t. Use a parametric (gamma) serial interval distribution with mean 8.4 days and standard deviation 3.8 days. For the window size use 4 days. Plot the result. (Note: as R_t fluctuates quite a bit, plotting it on a log scale may be helpful).
   Hint: the start and end dates for 4-day windows can be specified as follows.

```
# generate start and end days for 4-day windows
t_start <- seq(2, 47) # as there are 50 days in the data
t_end <- t_start + 3 # note: end days are included, thus +3 rather than +4
```

```
library(EpiEstim)
library(ggplot2)
# generate start and end days for 4-day windows
t_start <- seq(2, 47) # as there are 50 days in the data
t_end <- t_start + 3 # note: end days are included, thus +3 rather than +4
# generate a config list which we will re-use
config1 <- make_config(mean_si = 8.4,
                       std_si = 3.8,
                       t_start = t_start,
                       t_end = t_end)
R <- estimate_R(inc,
                method = "parametric_si",
                config = config1)

## Warning in estimate_R_func(incid = incid, method = method, si_sample = si_sample,
:  You're estimating R too early in the epidemic to get the desired
##             posterior CV.

# make a slightly customized plot:
pl <- plot(R, what = "R")
pl + scale_y_continuous(breaks = c(0, 0.125, 0.25, 0.5, 1, 2, 4, 8),
                        trans = "log2")

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Warning:  Removed 4 rows containing missing values (`geom_line()`).
```



4. Check the results visually and think about the two following questions:

   (a) Around day 24, the reproductive number is estimated to be above 10. According to Wikipedia (`https://en.wikipedia.org/wiki/SARS`), "R_0, ranges from 2 to 4 depending on different analyses" for SARS-COV-1. Discuss some possible explanations for these extreme estimates.

   (b) For the most recent days (47–50), the estimated R_t falls considerably below 1. Which potential issue do you see with this pattern?

   Solution:

   - A potential reason is that surveillance efforts were scaled up after the detection of some initial cases. This may have led to a more steep increase in reported cases than in actual cases (including undetected). Our R_t estimation method, which ignores the reporting fractions, then returns very large estimates.
   - A dip at the end of a series of estimated R_t values may indicate an issue with reporting delays. If the data are subject to such delays, the most recent values will be incomplete and R_t estimates may be biased downwards.

5. As the epidemiologists in charge are aware of the delay problem commonly occurring in surveillance data, they preserved daily snapshots of the epicurve and regularly generate reporting triangles. Here is the reporting triangle they generated on day 49:

t	$d=0$	$d=1$	$d=2$	$d=3$
41	22	23	11	2
42	16	19	8	5
43	9	16	4	4
44	5	12	5	3
45	14	18	7	4
46	16	12	4	5
47	12	10	6	?
48	6	10	?	
49	5	?		
50	?			

On day 50 the cells containing question marks can be filled in. Use the time series data as available on days 49 and 50 to compute the missing cells:

t	41	42	43	44	45	46	47	48	49	50
data as of day 49	58	48	33	25	43	37	28	16	5	
data as of day 50	58	48	33	25	43	37	30	26	14	7

Solution: The reporting triangle for day 50 is:

t	$d=0$	$d=1$	$d=2$	$d=3$
41	22	23	11	2
42	16	19	8	5
43	9	16	4	4
44	5	12	5	3
45	14	18	7	4
46	16	12	4	5
47	12	10	6	2
48	6	10	10	
49	5	9		
50	7			

Example computation: The entry for $t = 49, d = 1$ results from $14 - 5 = 12$.

6. The empty cells in your updated reporting triangle can be filled in using the simple nowcasting method from the slides. Perform these computations using a calculator or R. **Note:** The reporting triangle for day 50 is available in the file `reporting_triangle_day50.csv`, but to keep things simple you can just note your results by hand. The incidence time series with nowcasted and complete data are available in `data_day50_nowcast.csv` and `data_day50_complete.csv` for further use. So you are not obliged to perform the computations for all six cells if you feel you got the principle.

   Solution:

t	$d=0$	$d=1$	$d=2$	$d=3$	total (rounded)
41	22	23	11	2	58
42	16	19	8	5	48
43	9	16	4	4	33
44	5	12	5	3	25
45	14	18	7	4	43
46	16	12	4	5	37
47	12	10	6	2	30
48	6	10	10	2.96	29
49	5	9	3.5	2.00	20
50	7	8.6	3.9	2.22	22

   Example computation: For $t = 50$, $d = 1$ we have

$$7 \times \frac{23 + 19 + 16 + 12 + 18 + 12 + 10 + 10 + 9}{22 + 16 + 9 + 5 + 14 + 16 + 12 + 6 + 5} = \frac{129}{105} = 8.6$$

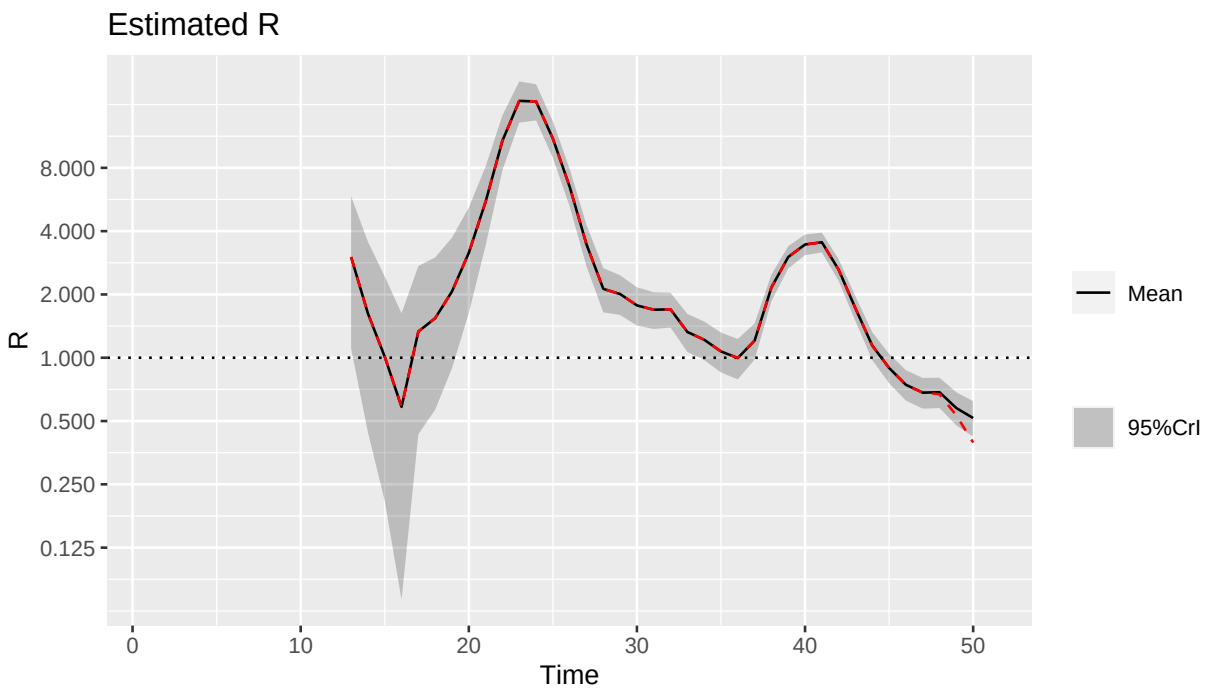
   The nowcasted values can be compared to the uncorrected ones shown at the end of question 5. It can be seen that at least for the last two values, the correction is substantial.

7. Use the incidence time series with nowcasting correction(file `data_day50_nowcast.csv`) and the final time series (file `data_day50_complete.csv`) to re-run the nowcasting computations. Compare the results to those from question 4.

Solution: The estimated R_t for the last three days increase quite a bit when accounting for reporting delays. Specifically, we get a value of around 0.62 rather than 0.40. The estimates based on the complete data look quite similar (i.e., the nowcasting worked). The qualitative finding that $R_t < 1$, however, does not change in this setting. In this case, the dip in R_t at the end of the series is thus not purely an artefact.

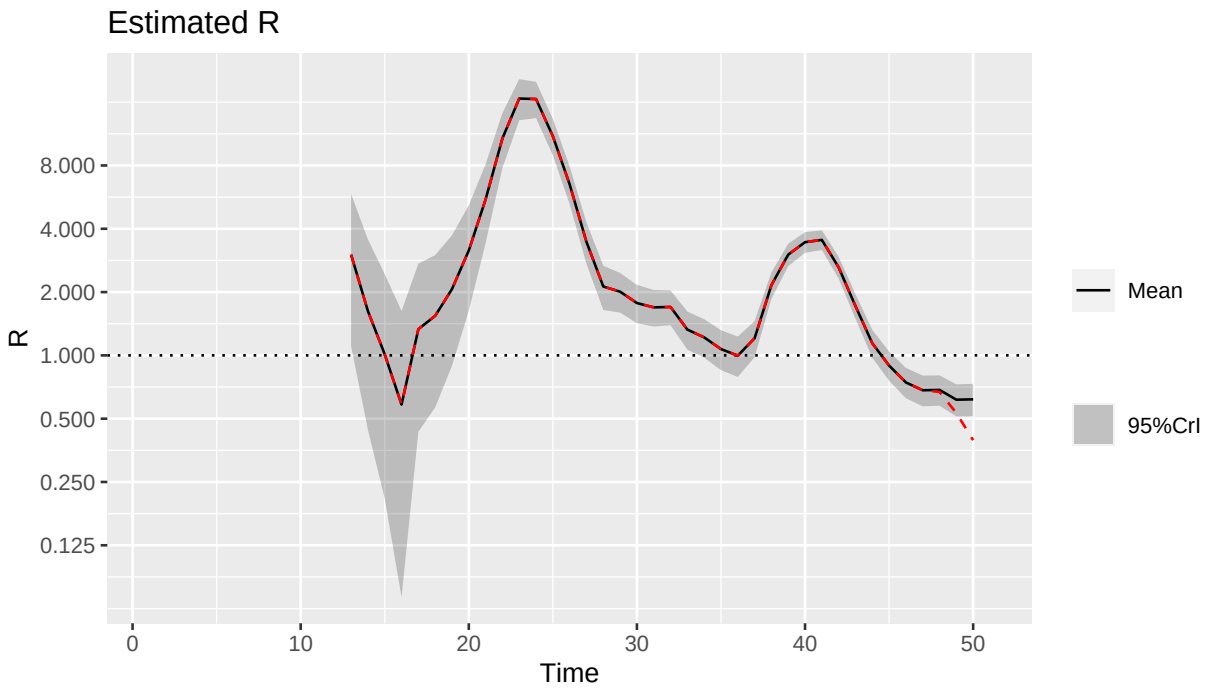
In the below plots, the dashed red line shows the result without nowcasting.

```
# nowcasted data:
# load the data:
inc_nowcast <- read.csv("data_day50_nowcast.csv")
# estimate R:
R_nowcast <- estimate_R(inc_nowcast,
                        method = "parametric_si",
                        config = config1)
# make a slightly customized plot:
pl_nowcast <- plot(R_nowcast, what = "R")
pl_nowcast +
  scale_y_continuous(breaks = c(0, 0.125, 0.25, 0.5, 1, 2, 4, 8), trans = "log2") +
  geom_line(aes(y = R$R$`Mean(R)`), col = "red", lty = 2)

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```




```
# complete data:
# load the data:
inc_complete <- read.csv("data_day50_complete.csv")
# estimate R:
R_complete <- estimate_R(inc_complete,
                         method = "parametric_si",
                         config = config1)
# make a slightly customized plot:
pl_complete <- plot(R_complete, what = "R")
```

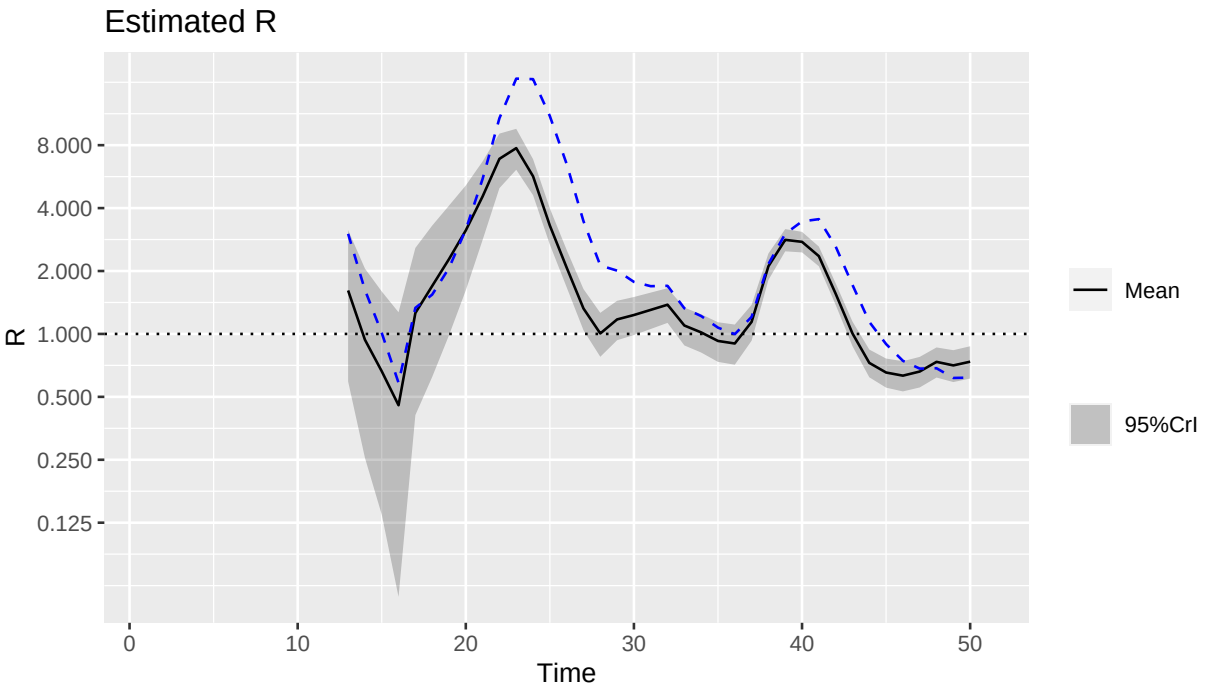
```
pl_complete +
  scale_y_continuous(breaks = c(0, 0.125, 0.25, 0.5, 1, 2, 4, 8), trans = "log2") +
  geom_line(aes(y = R$R$`Mean(R)`), col = "red", lty = 2)

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



8. Re-run the R_t estimation using the complete data and an adapted serial interval distribution with mean 5 days and standard deviation 3 days. How do the results compare to those from the previous question?

   Solution: This specification leads to weaker amplitudes in the estimated R_t values (as discussed in the slides).

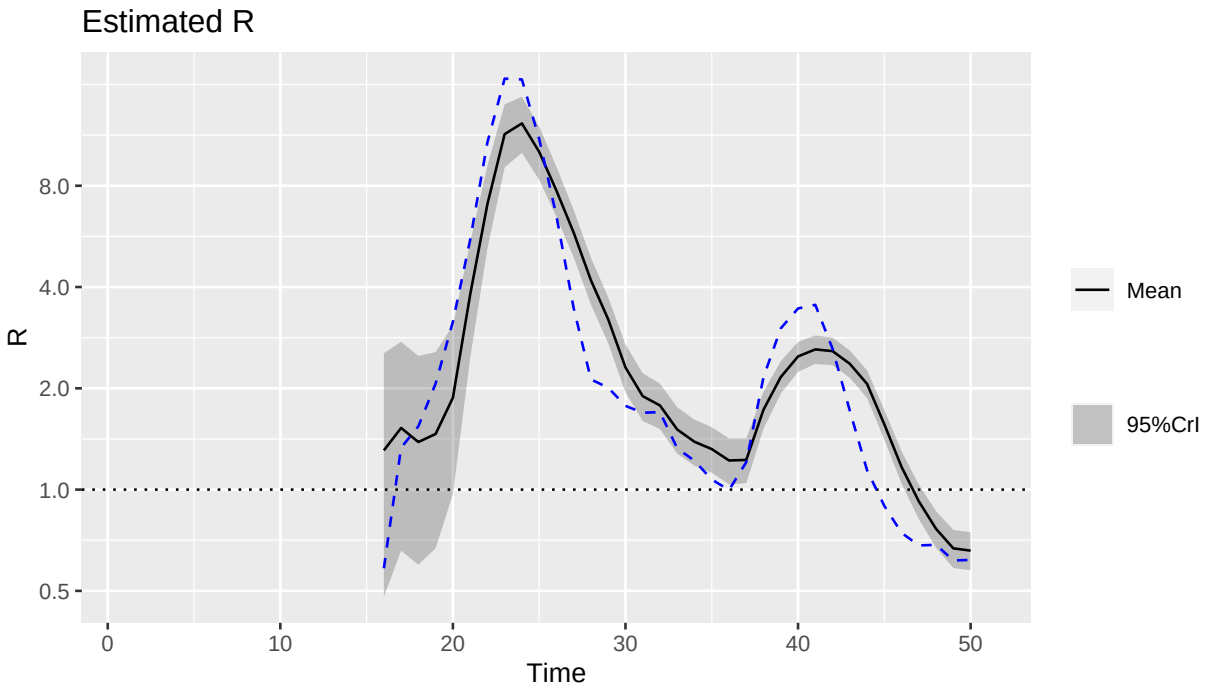
```
# estimate R:
config2 <- make_config(mean_si = 5,
                       std_si = 3,
                       t_start = t_start,
                       t_end = t_end)
R_complete2 <- estimate_R(inc_complete,
                          method = "parametric_si",
                          config = config2)
# make a slightly customized plot:
pl_complete2 <- plot(R_complete2, what = "R")
pl_complete2 +
  scale_y_continuous(breaks = c(0, 0.125, 0.25, 0.5, 1, 2, 4, 8), trans = "log2") +
  geom_line(aes(y = R_complete$R$`Mean(R)`), col = "blue", lty = 2)

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



9. Re-run the R_t estimation using a window size of 7 rather than 4 days. How do the results compare to those from question 8?

   Solution: This specification leads to smoother and sightly shifted estimates (as the estimates refer to longer windows which thus start earlier for the same end date).

```
# estimate R:
t_start3 <- seq(10, 44)
t_end3 <- t_start3 + 6
config3 <- make_config(mean_si = 8.4,
                       std_si = 3.8,
                       t_start = t_start3,
                       t_end = t_end3)
R_complete3 <- estimate_R(inc_complete,
                          method = "parametric_si",
                          config = config3)
# make a slightly customized plot:
pl_complete3 <- plot(R_complete3, what = "R")
pl_complete3 +
  scale_y_continuous(breaks = c(0, 0.125, 0.25, 0.5, 1, 2, 4, 8), trans = "log2") +
  geom_line(aes(y = tail(R_complete$R$`Mean(R)`, nrow(R_complete3$R))), col = "blue", lty

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



10. The incidence time series is aggregated by the date of symptom onset. It is assumed that the mean incubation period of SARS-COV1 is around 5 days. How can you take this into account when interpreting the results? Use the value 0.62 (uncertainty interval 0.5–0.71) estimated for day 50 as an example.

    Solution: Over a time window of length 4 days which ended on day $50-5=45$, the instantaneous effective reproductive number was estimated to be 0.62 (and within the interval 0.5–0.71 with good confidence). This means that on average (and in a backward-looking sense, compare slides), one infected caused 0.62 secondary cases.