Supplementary details on experiments

Olivia

2023-09-26

Experimental design

Synthetic reproduction numbers

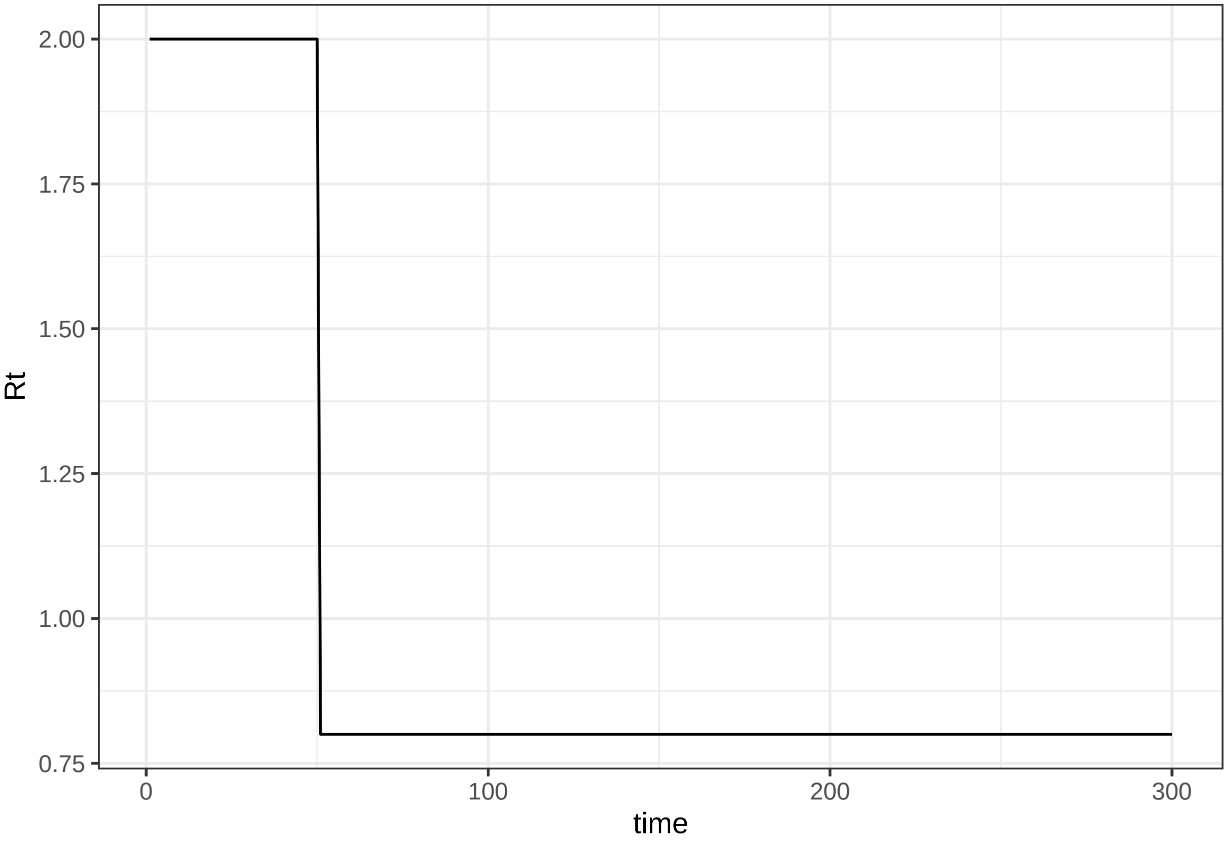
Overall, we argue our estimator is accurate, robust in model misspecification and computationally efficient. We can do a series of tests for each property. We may consider the following curvature of efficient reproduction numbers, and test the accuracy of our estimators compared to EpiEstim and EpiLPS.

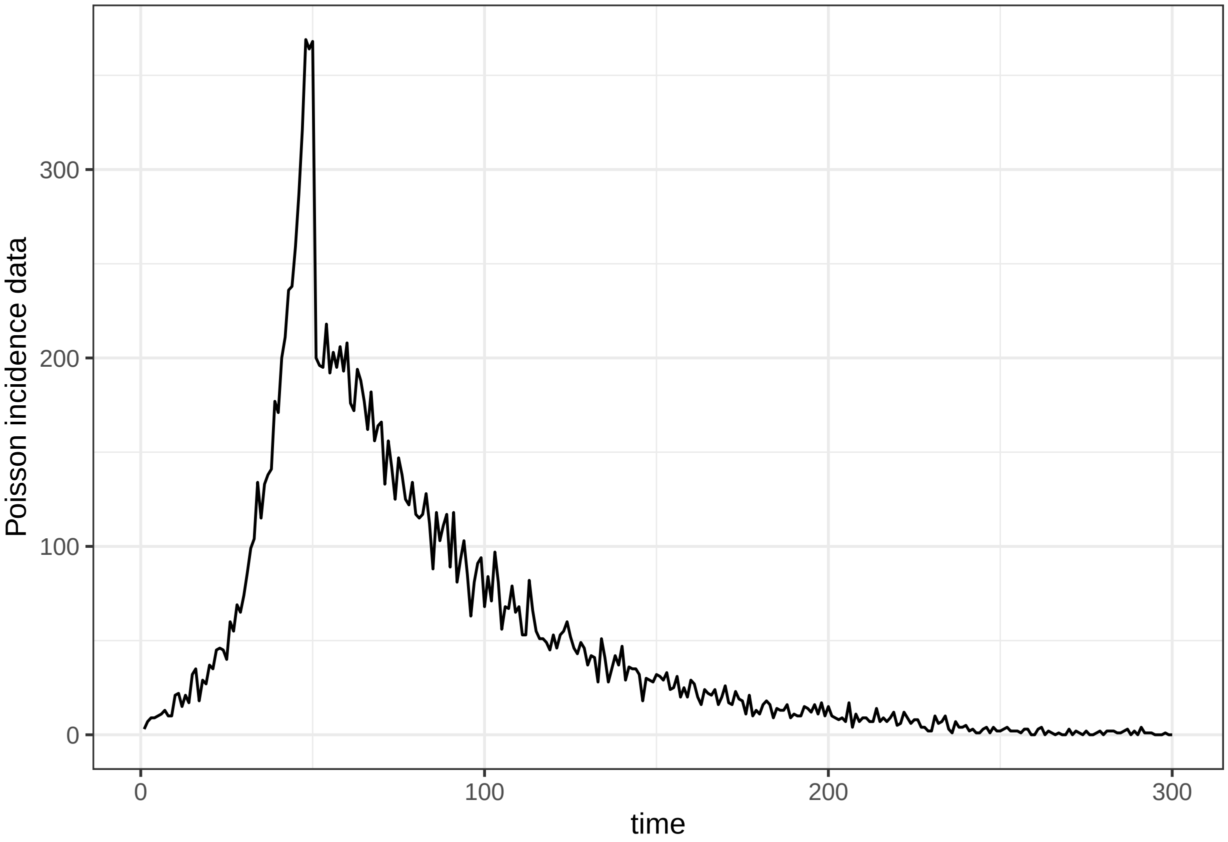
We may consider arbitrary reproduction numbers in a few scenarios: a) piecewise-constant epidemics with a drop at a certain time point to measure the effect of control measures, b) exponentially rising and falling epidemics with a change point, c) piecewise-constant with multiple segments to measure the initially controlled and resurged and the suppressed epidemics, d) periodic waves.

We may simulate the epidemics (with length T=300) 10 times for each scenario, estimate Rt, and compute the averaged KL divergence.

```
# General settings:
N1 = 2 # first incidence data
len = 300 # number of evenly spaced time points
library(rtestim)
# Get Poisson incidence cases:
get_pois_incidence <- function(N1, Rt, gamma_pars = c(2.5, 2.5)){
  len <- length(Rt)
  incidence <- numeric(len) # N_1:n
  poisson_count <- numeric(len) # y_1:n
  incidence[1] <- N1

  poisson_count[1] <- rpois(1, N1)
  if(poisson_count[1] == 0) poisson_count[1]=1
  for(t in 2:len){
    pi <- discretize_gamma(1:(t-1), gamma_pars[1], gamma_pars[2])
    incidence[t] <- Rt[t] * sum(rev(pi) * poisson_count[1:(t-1)])
    poisson_count[t] <- rpois(1, incidence[t])
  }

  return(poisson_count)
}
# Display the synthetic data:
library(ggplot2)
display_dat <- function(counts, Rt){
  len <- length(counts)
  if(length(counts) != length(Rt)) cli::cli_abort("Data lengths do not match.")
  dat <- data.frame(time = 1:len, count = counts, Rt = Rt)
  fig1 <- dat %>%
    ggplot(aes(y = Rt, x = time)) +
    geom_line() +
    theme_bw()
  print(fig1)
  fig2 <- dat %>%
    ggplot(aes(y = count, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    ylab("Poisson incidence data") + #(in log scale)
    theme_bw()
  print(fig2)
}
# Check data quality:
check_dat <- function(incidence, Rt){
  if(min(Rt) < 0) {cli::cli_abort("`Rt` must be non-negative.")}
  if(min(incidence) < 0) {cli::cli_abort("`incidence` cases must be nonnegative.")}
  if(max(incidence) > 1e4L) {cli::cli_alert_warning("`incidence` cases are too large.")}
  if(sum(incidence == 0) > 30) {cli::cli_warn("`incidence` data has more than 10% 0s.")}
}
```

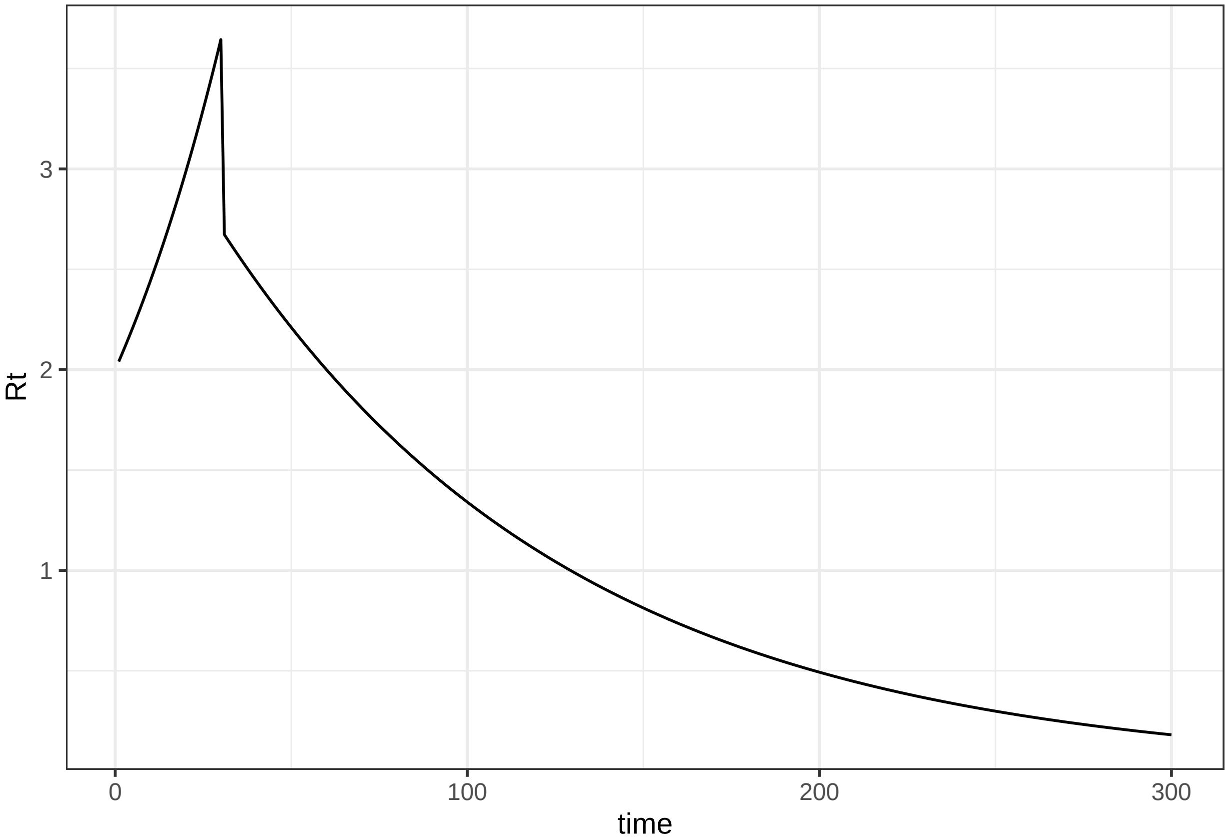
```
# Scenario 1: two-stage piecewise constant with one dropping point (similar as in EpiFilter)
Rt1 <- c(rep(2, 50), rep(0.8, len-50)) # arbitrary sequence of Rt
gamma_pars1 <- c(3, 3) # serial interval distribution parameters
seed <- 619
set.seed(seed)
incidence1 <- get_pois_incidence(N1, Rt1, gamma_pars1)
display_dat(incidence1, Rt1)
```

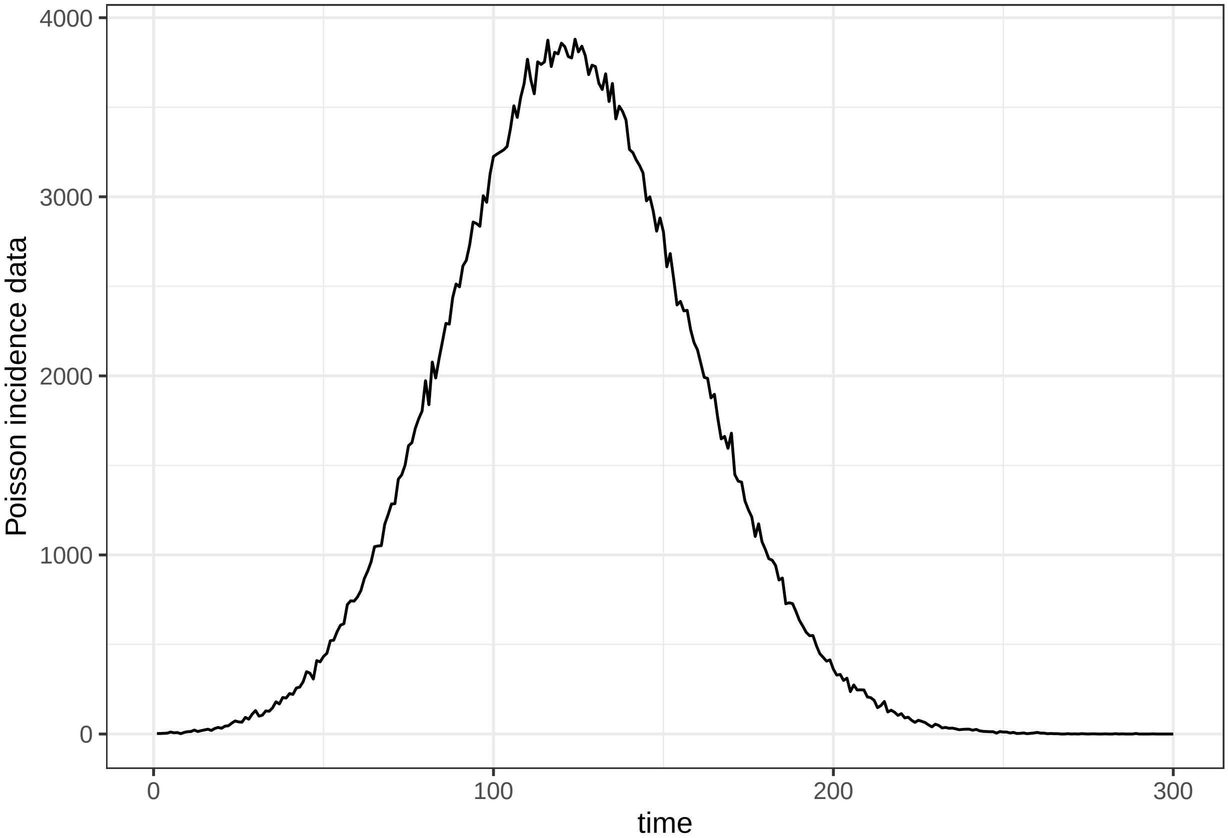




```
check_dat(incidence1, Rt1)
```

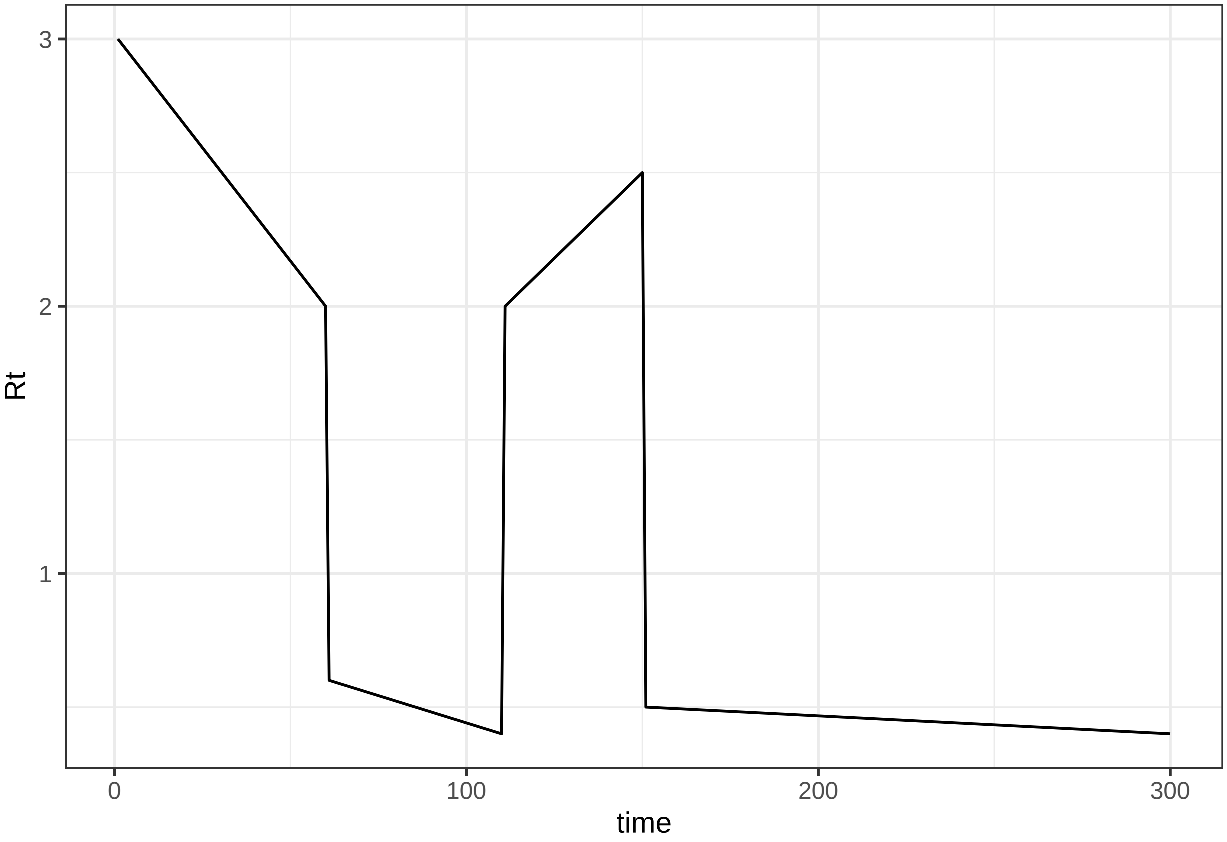
```
# Scenario 2: two-stage exponential growth and decay (similar as in EpiFilter)
rate <- c(.02, -.01)
Rt2 <- numeric(len)
knot <- 30
Rt2[1:knot] <- exp(rate[1] * (1:knot)) * 2
Rt2[(knot+1):len] <- exp((rate[2]) * ((knot+1):len))* Rt2[knot]
gamma_pars2 <- c(3.5, 3.5) # serial interval distribution parameters
set.seed(621)
incidence2 <- get_pois_incidence(N1, Rt2, gamma_pars2)
display_dat(incidence2, Rt2)
```

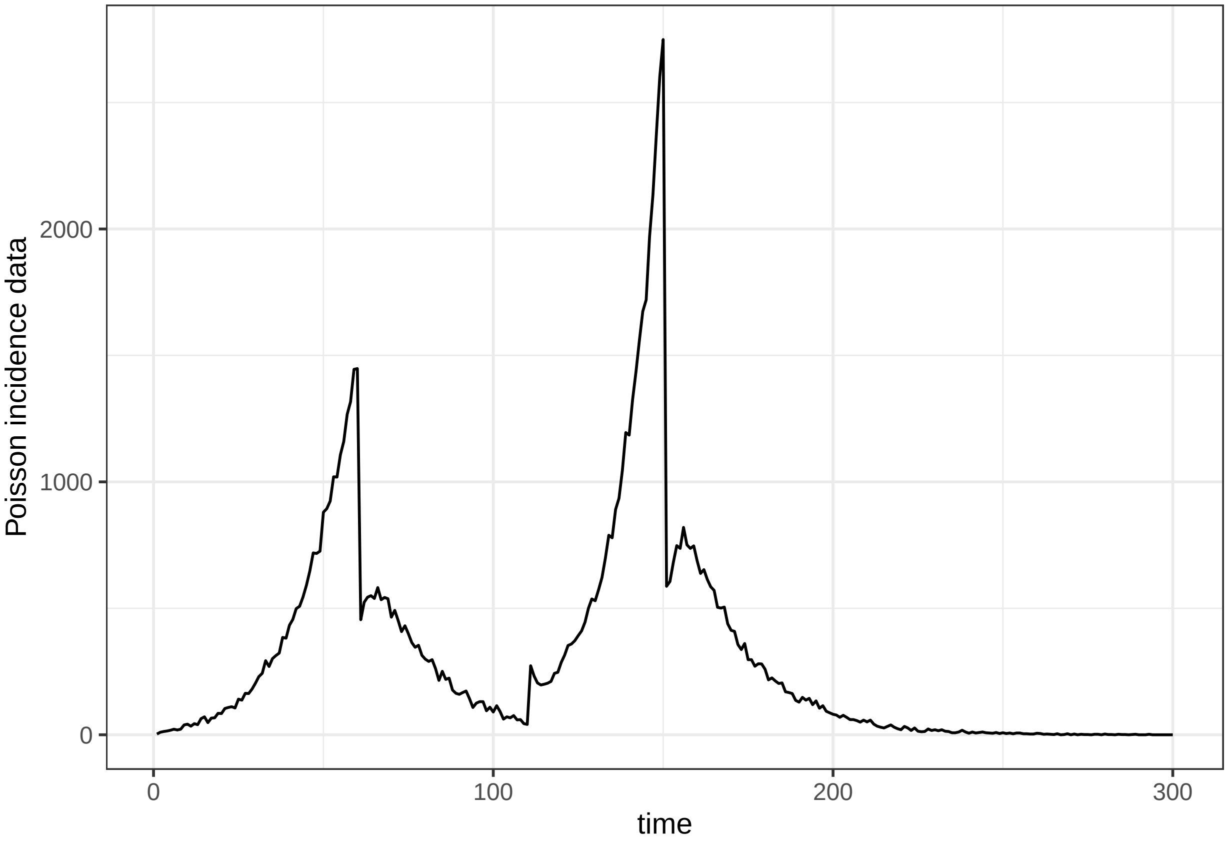

Rt
3
2
1
0
100
200
300
time


Poisson incidence data
4000
3000
2000
1000
0
0
100
200
300
time

```
check_dat(incidence2, Rt2)
```

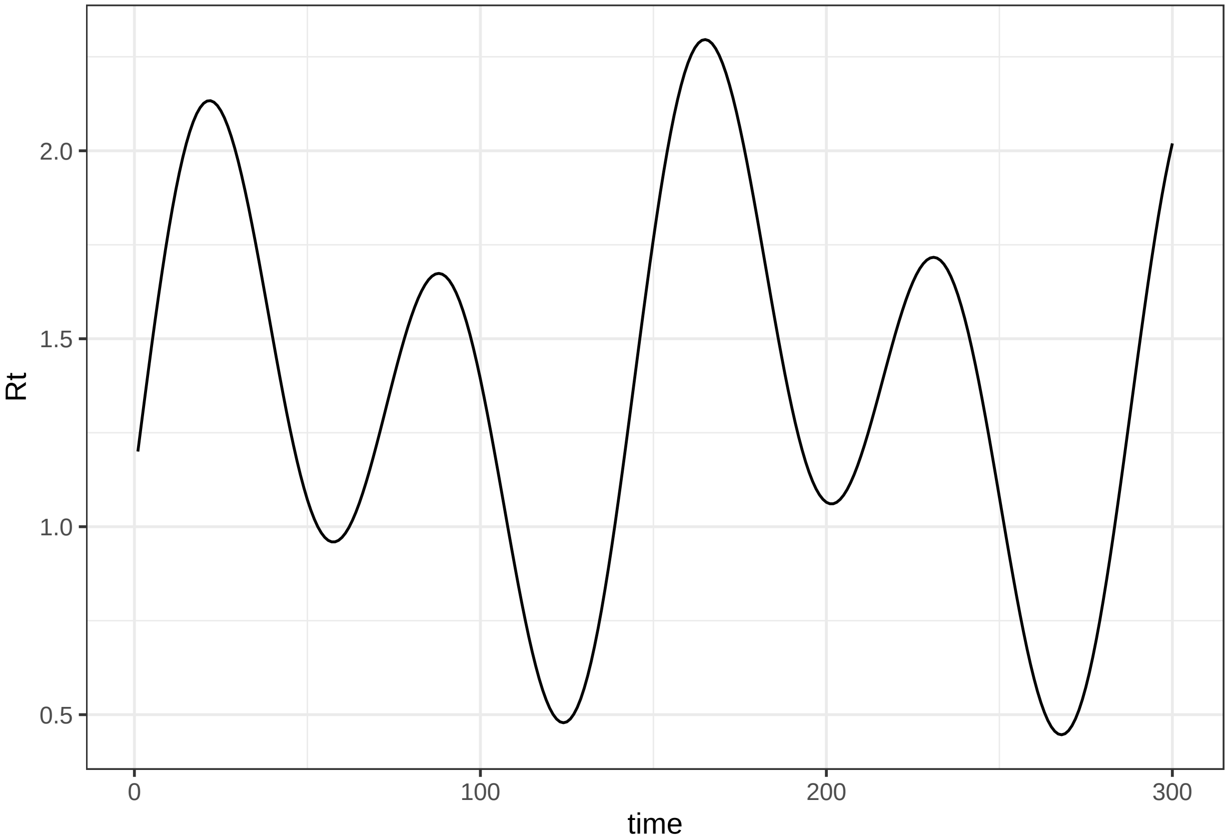
```
# Scenario 3: multi-stage piecewise constant (similar as in EpiFilter)
Rt3 <- c(seq(3, 2, length.out = 60), seq(0.6, 0.4, length.out = 50),
         seq(2, 2.5, length.out = 40), seq(0.5, 0.4, length.out = 150))
gamma_pars3 <- c(3.5, 3.5) # serial interval distribution parameters
set.seed(seed)
incidence3 <- get_pois_incidence(N1, Rt3, gamma_pars3)
display_dat(incidence3, Rt3)
```

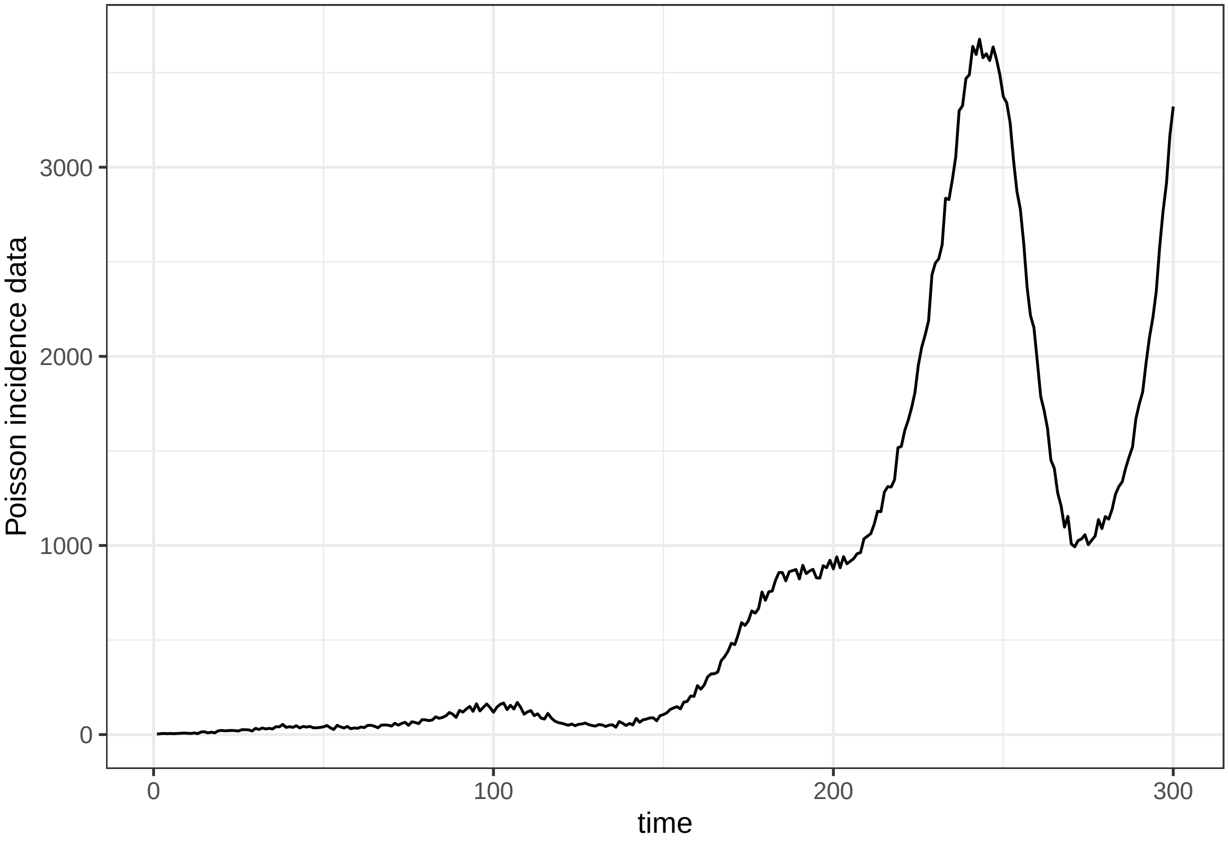




```
check_dat(incidence3, Rt3)
```

```
# Scenario 4: multi-stage with exponential-sinusoidal growth and decay
x <- seq(0, 10, length.out = len)
Rt4 <- numeric(len)
components <- list(
  list(freq = 0.1, amp = 1),
  list(freq = 0.5, amp = 2),
  list(freq = 1.0, amp = 3)
)
for (component in components) {
  Rt4 <- Rt4 + 0.2 * (component$amp * sin(pi * component$freq * x / 1.2) +
                       component$amp)
}
gamma_pars4 <- c(3.5, 3.5) # serial interval distribution parameters
set.seed(seed)
incidence4 <- get_pois_incidence(N1, Rt4, gamma_pars4)
display_dat(incidence4, Rt4)
```

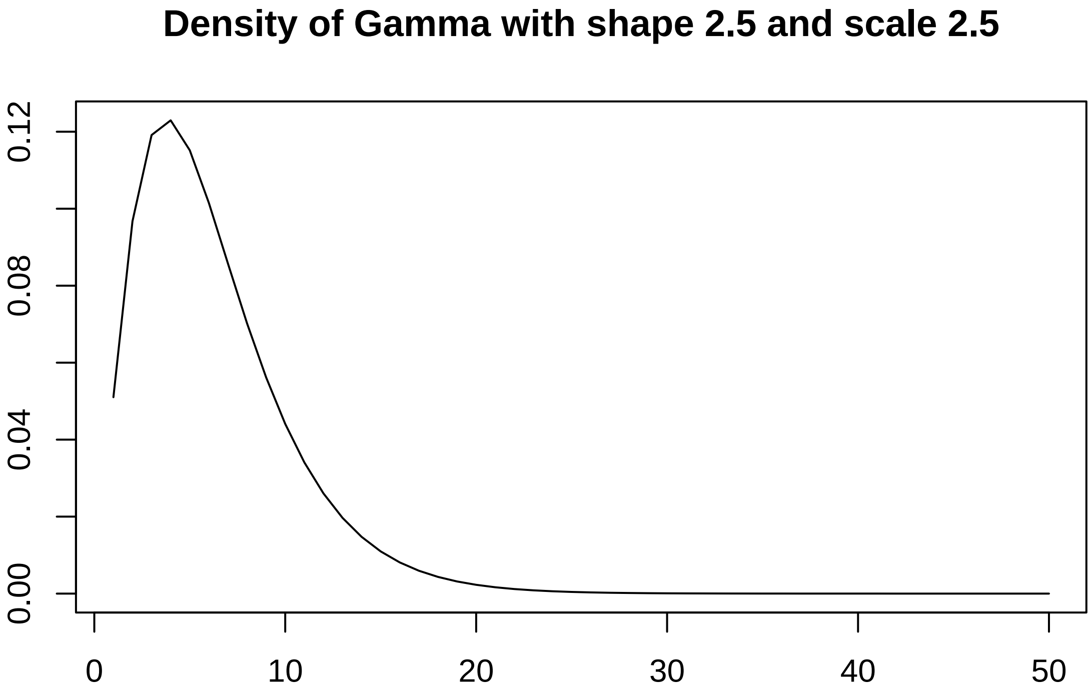

Rt
2.0
1.5
1.0
0.5
0
100
200
300
time


Poisson incidence data
3000
2000
1000
0
0
100
200
300
time

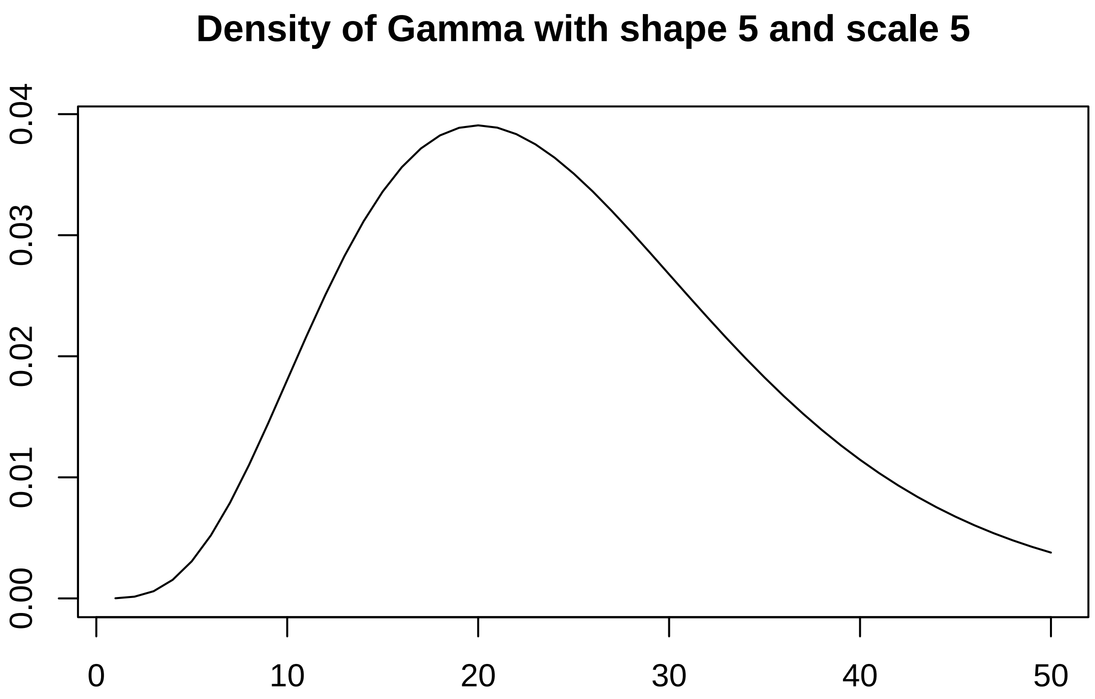
```r
check_dat(incidence4, Rt4)
```

Parameters of the serial interval distribution, i.e., shapes and scale/rates of Gamma distribution, can significantly influence the peak values of incidences and the smoothness of incidence curves. Here is a comparison of the densities of Gamma distribution with shape 2.5 and scale 2.5 and Gamma distribution with shape 5 and scale 5.

```r
plot(dgamma(1:50, 2.5, scale = 2.5), type="l", ylab="", xlab="",
     main="Density of Gamma with shape 2.5 and scale 2.5")
```



```r
plot(dgamma(1:50, 5, scale = 5), type="l", ylab="", xlab="",
     main="Density of Gamma with shape 5 and scale 5")
```



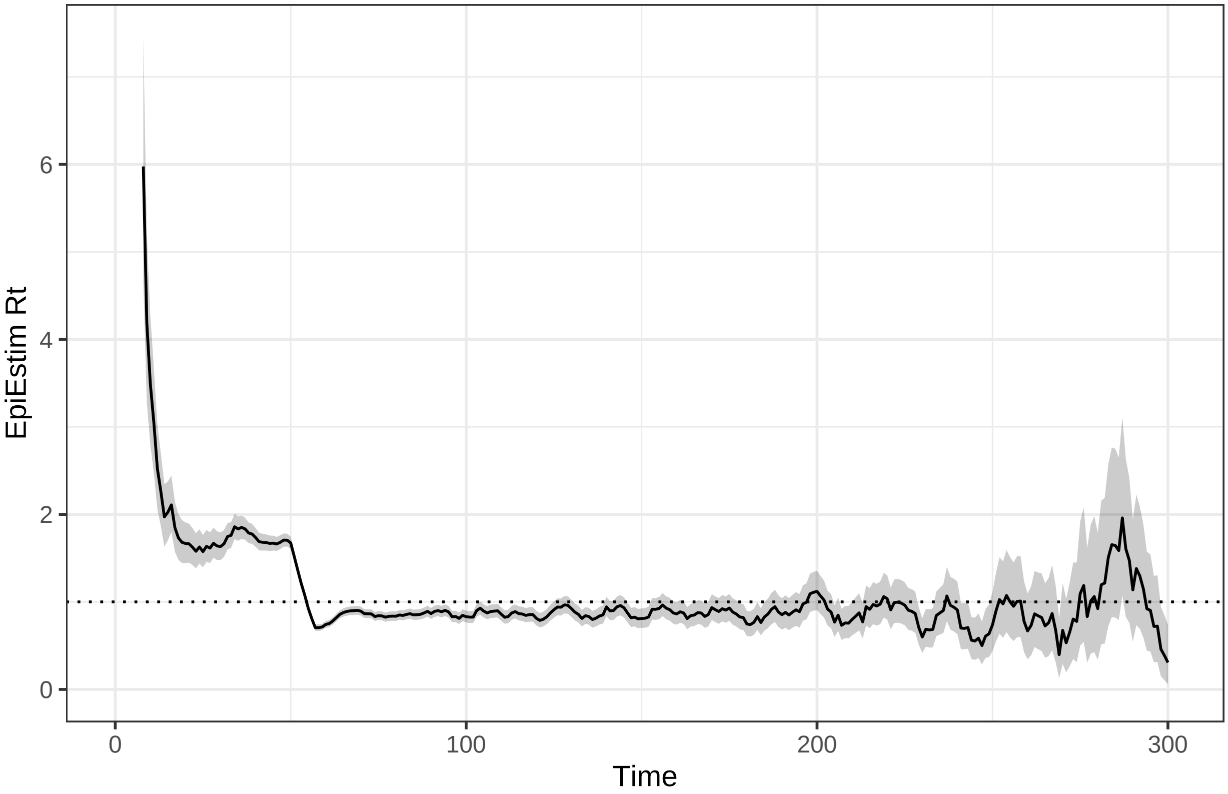
Density of Gamma with shape 5 and scale 5

Reproduction number estimation

```r
# EpiEstim with weekly sliding windows; specify the SI distribution
library(EpiEstim)
method <- "non_parametric_si"
# get gamma(3,3) probabilities with length 12; the 1st component is 0
prob_gamma1 <- c(0, diff(c(0, pgamma(1:11, 3, scale=3)))) / pgamma(11, 3, scale=3)
config1 <- make_config(list(si_distr = prob_gamma1))

mod_epiEstim1 <- EpiEstim::estimate_R(incid = incidence1, config = config1,
                                      method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

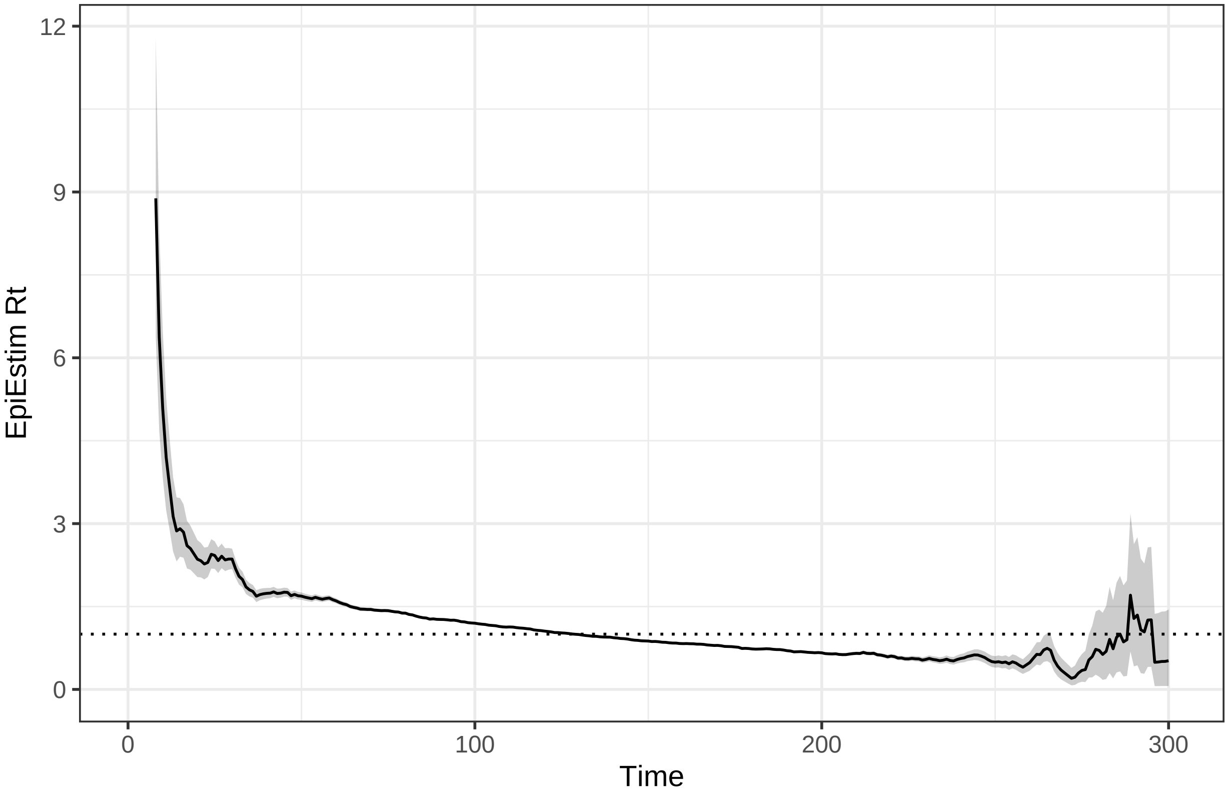
```r
plot(mod_epiEstim1, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none
```



```
prob_gamma2 <- c(0, diff(c(0, pgamma(1:11, 3.5, scale=3.5)))) / pgamma(11, 3.5, scale=3.5)
config2 <- make_config(list(si_distr = prob_gamma2))

mod_epiEstim2 <- EpiEstim::estimate_R(incid = incidence2, config = config2,
                                      method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

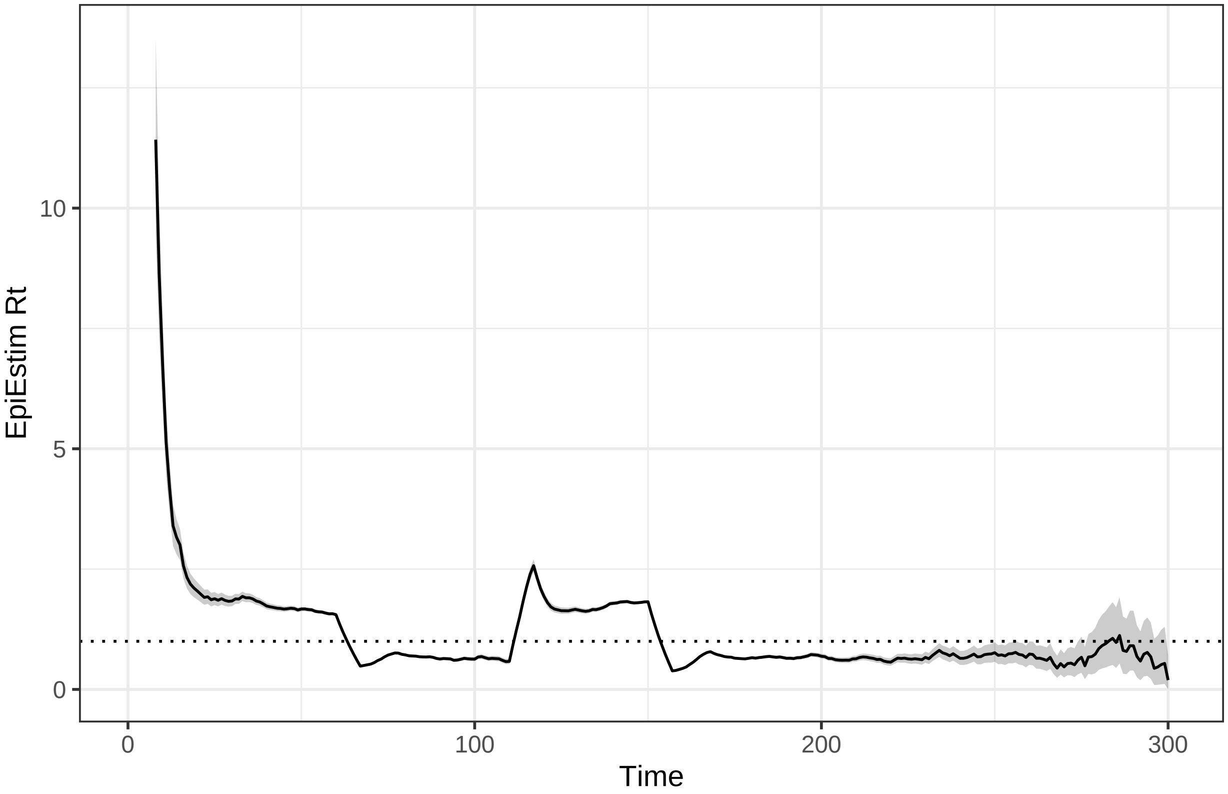
```
plot(mod_epiEstim2, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none
```



```
mod_epiEstim3 <- EpiEstim::estimate_R(incid = incidence3, config = config2,
                                      method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

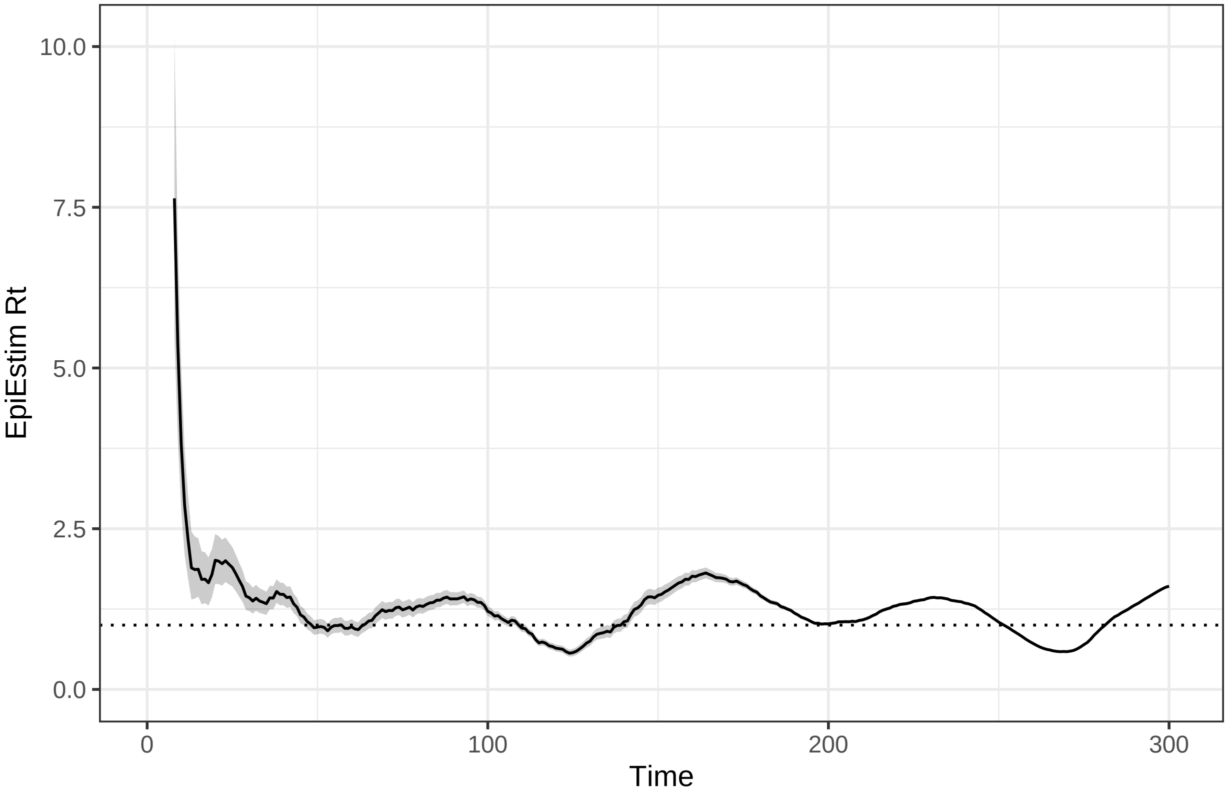
```
plot(mod_epiEstim3, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none
```



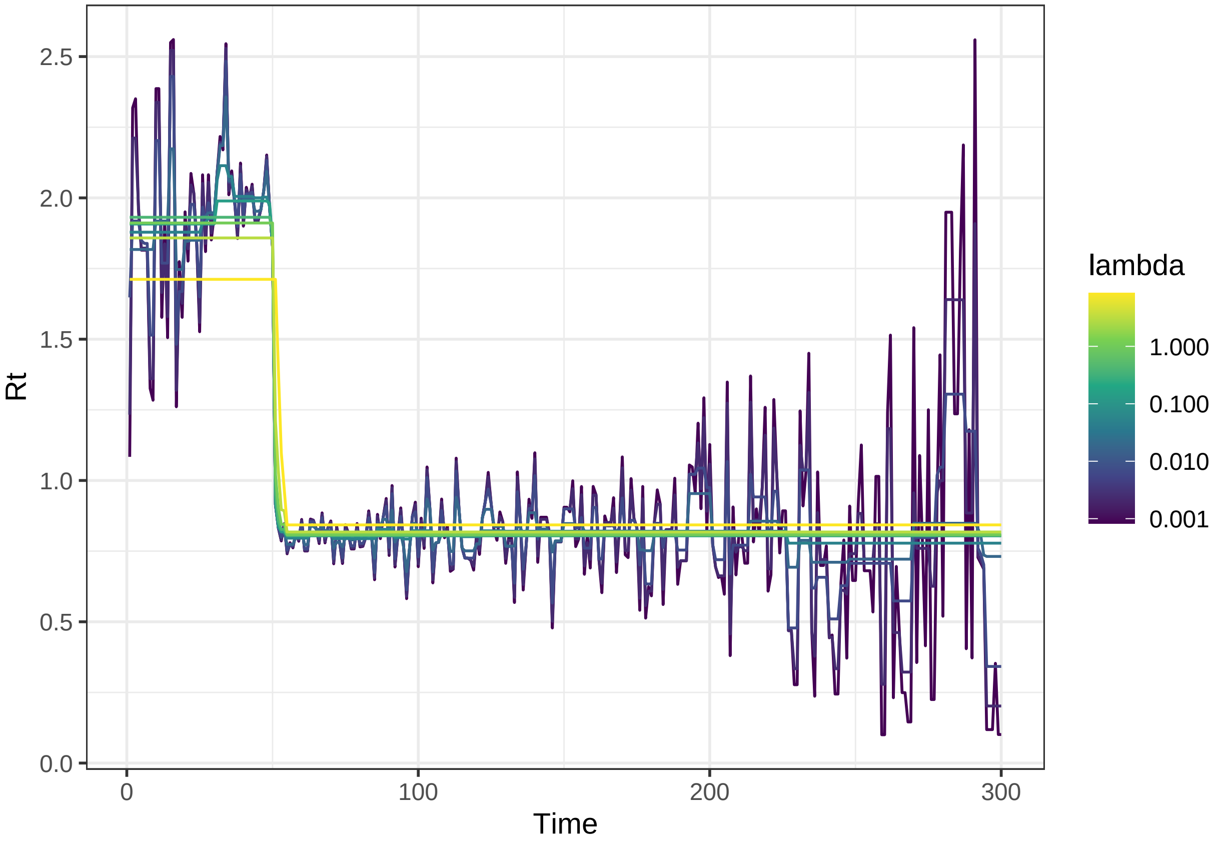
```
mod_epiEstim4 <- EpiEstim::estimate_R(incid = incidence4, config = config2,
                                       method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

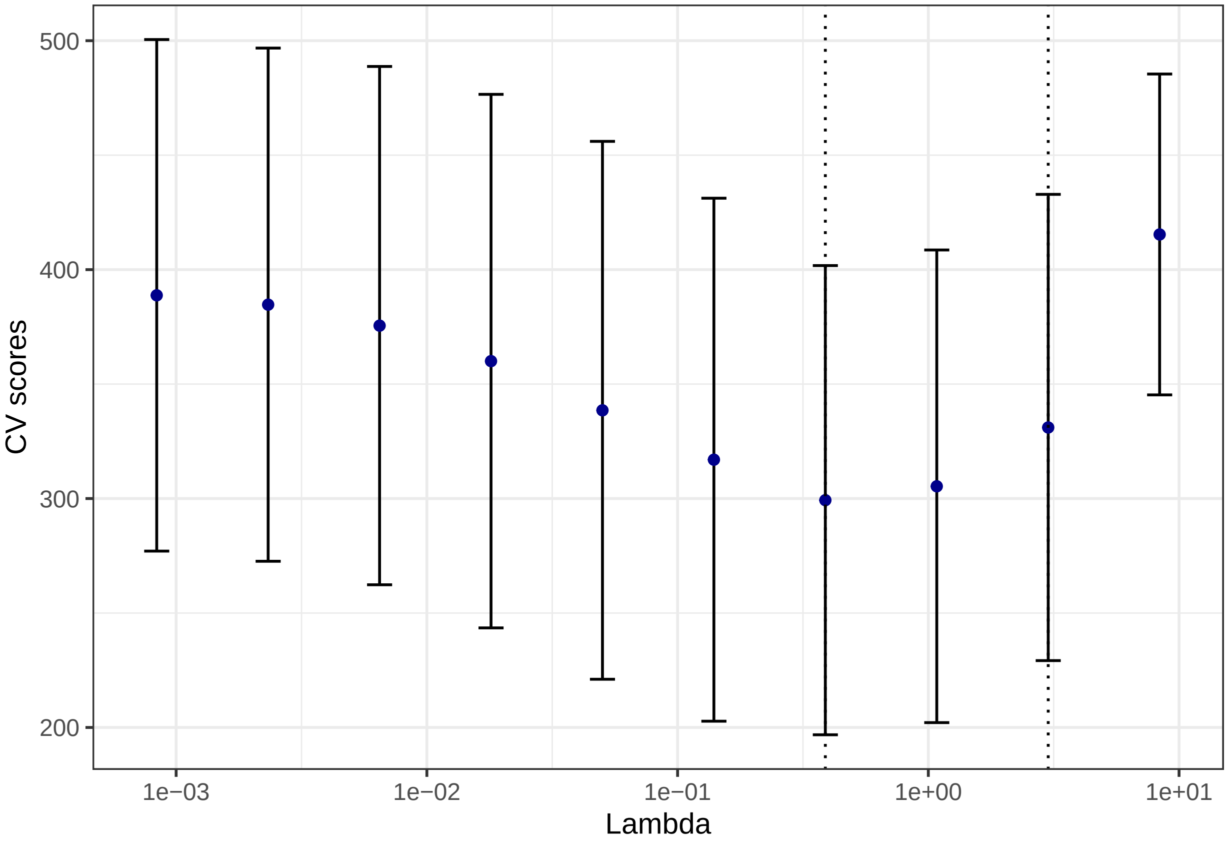
```
plot(mod_epiEstim4, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none
```



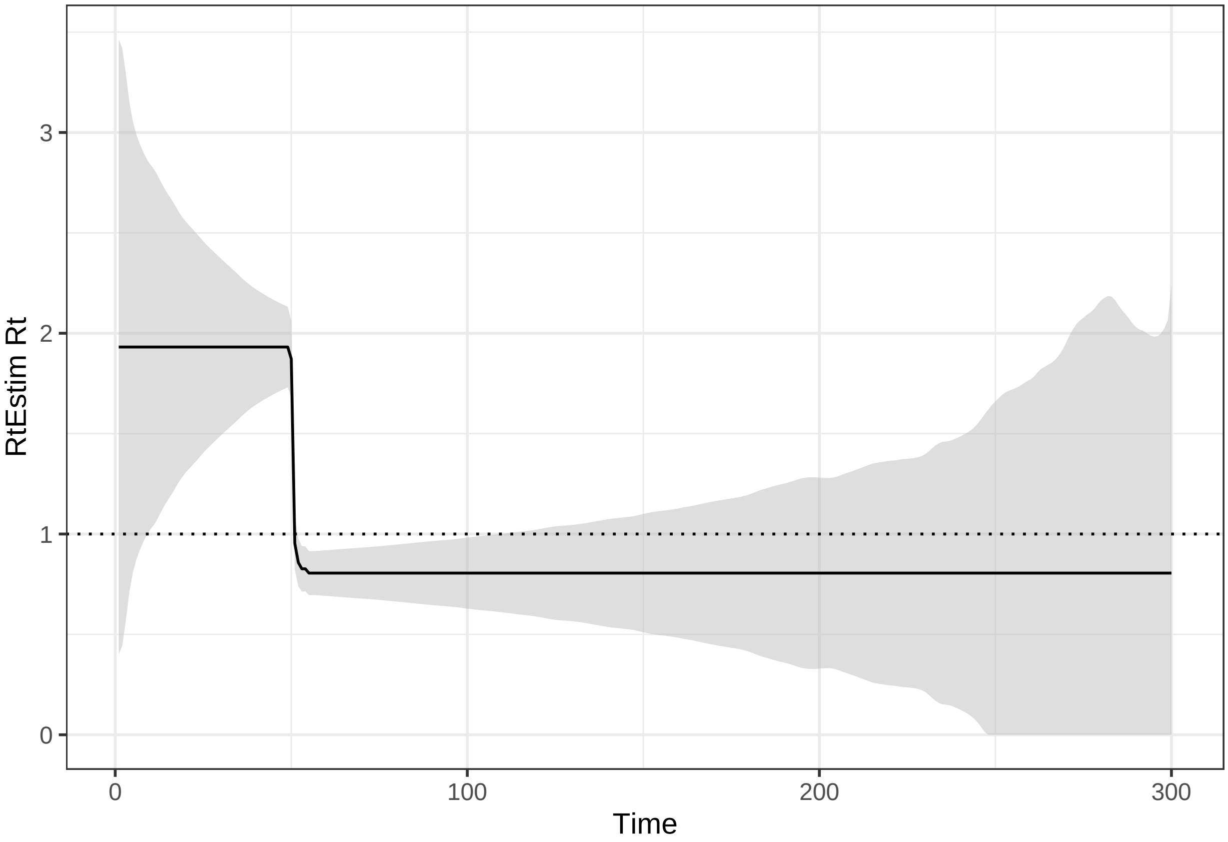
```
rtestim_mod1 <- estimate_rt(incidence1, korder = 0, nsol = 10,
                            dist_gamma = c(3, 3))
plot(rtestim_mod1)
```



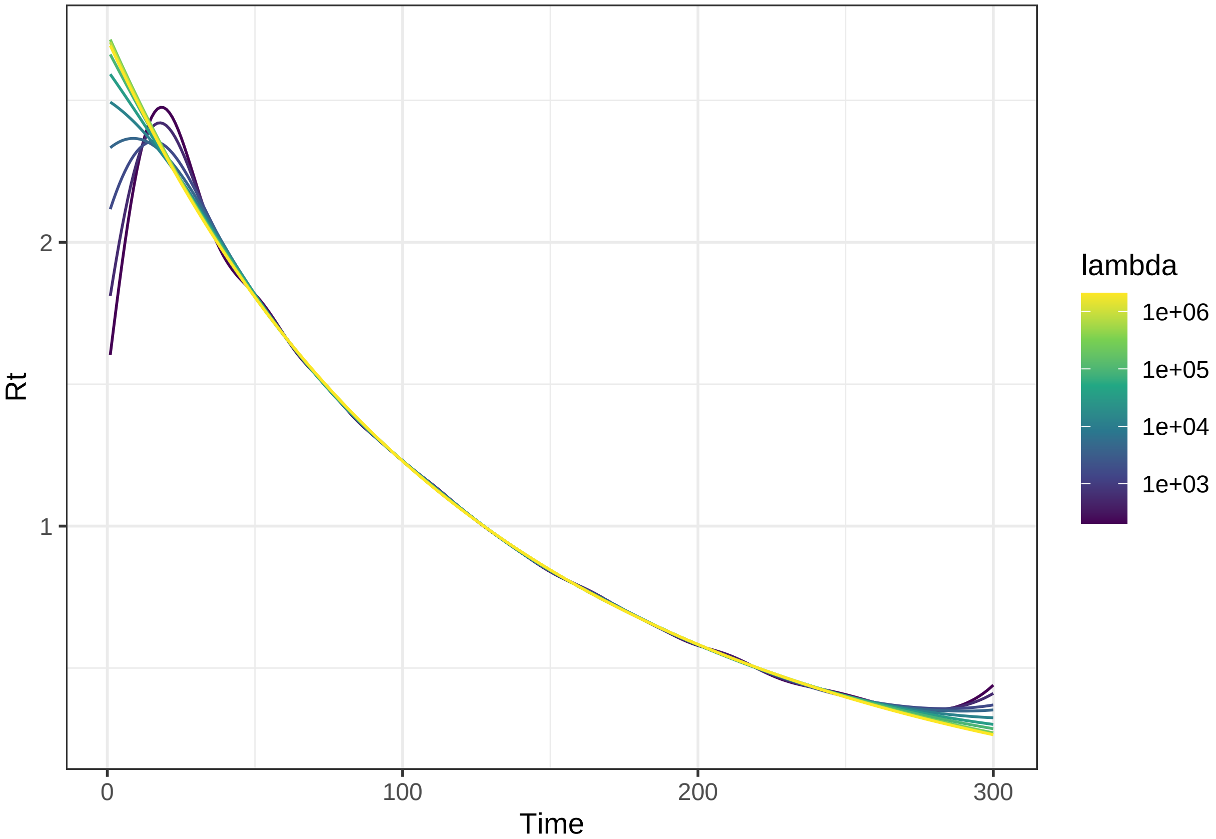
```r
# hyperparameter tuning using cross validation:
cv_mod1 <- cv_estimate_rt(incidence1, korder = 0, nfold = 3, nsol = 10,
                          dist_gamma = c(3, 3), maxiter=1e7L)
plot(cv_mod1)
```



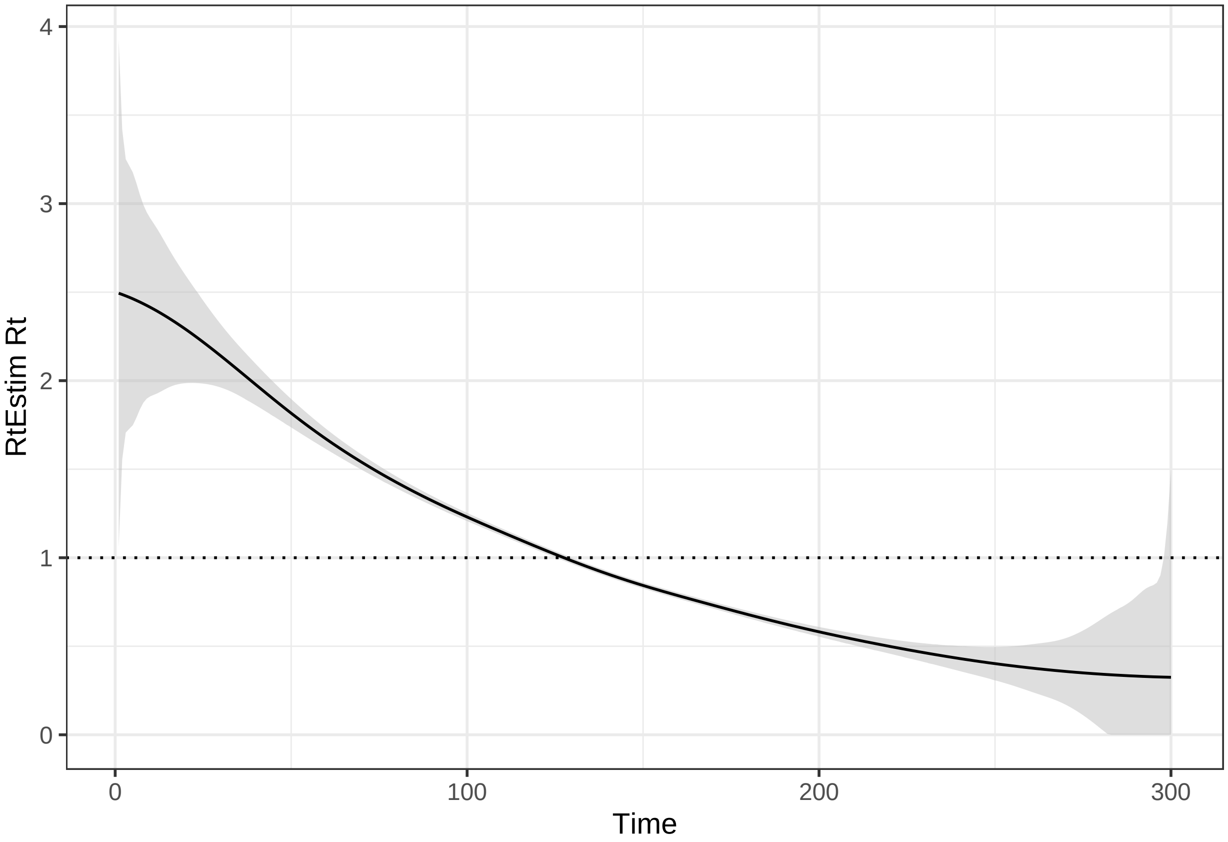
```
rtestim_tuned_mod1 <- cv_mod1$full_fit$Rt[, which.min(cv_mod1$cv_scores)]
rt_ci1 <- confband(cv_mod1, "lambda.min") # get 95% confidence band
rt_ci1 %>%
  ggplot(aes(x = 1:length(incidence1), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



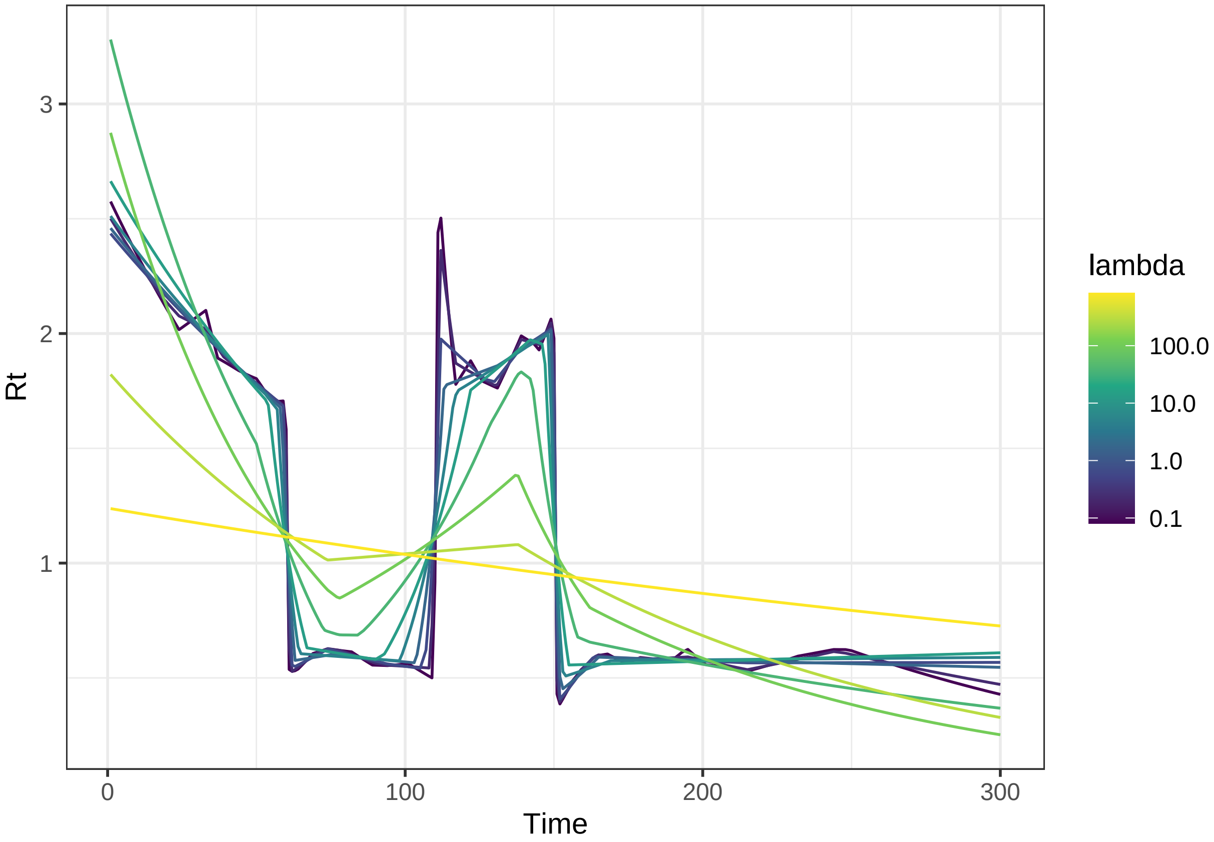
```
rtestim_mod2 <- estimate_rt(incidence2, korder=3, nsol=10,
                            dist_gamma = c(3, 3))
plot(rtestim_mod2)
```



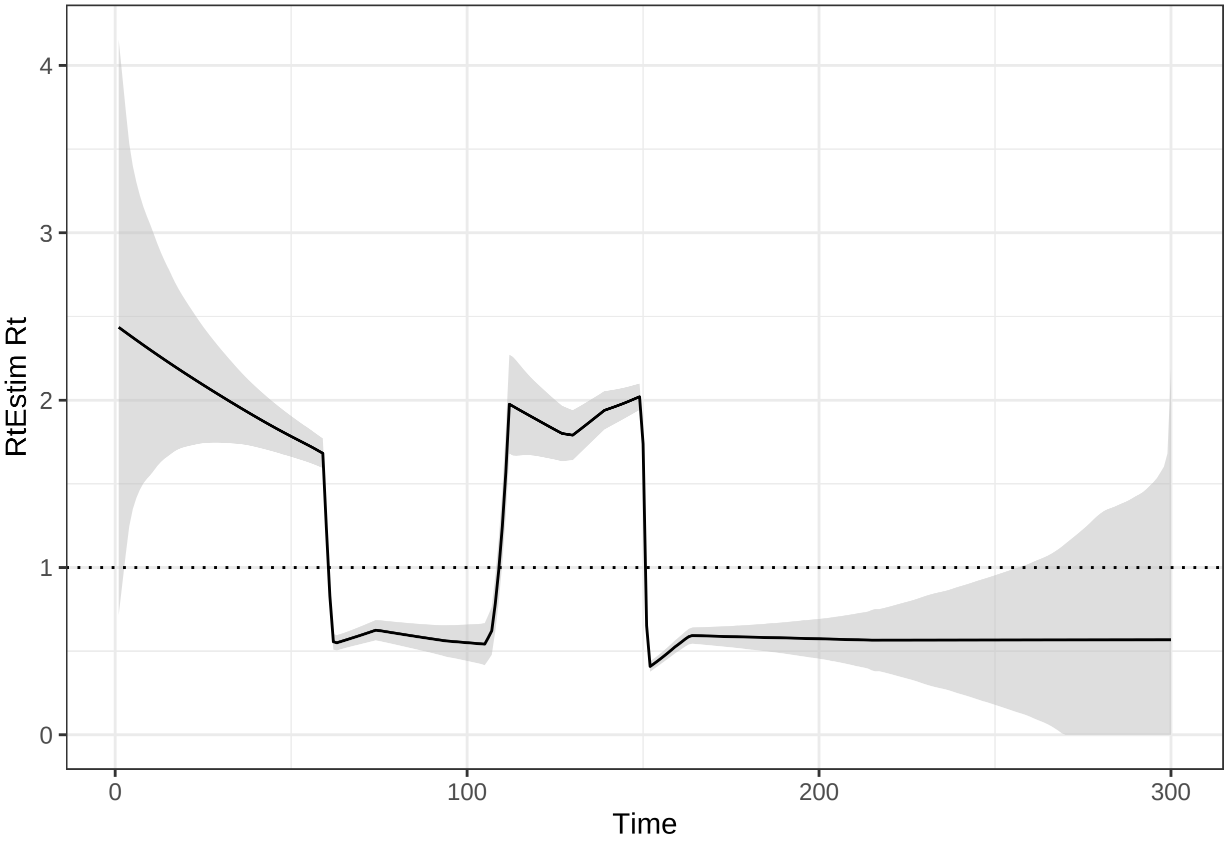
```r
cv_mod2 <- cv_estimate_rt(incidence2, korder=3, nfold=3, nsol=10,
                          dist_gamma = c(3, 3))
rtestim_tuned_mod2 <- cv_mod2$full_fit$Rt[, which.min(cv_mod2$cv_scores)]
rt_ci2 <- confband(cv_mod2, "lambda.min") # get 95% confidence band
rt_ci2 %>%
  ggplot(aes(x = 1:length(incidence2), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



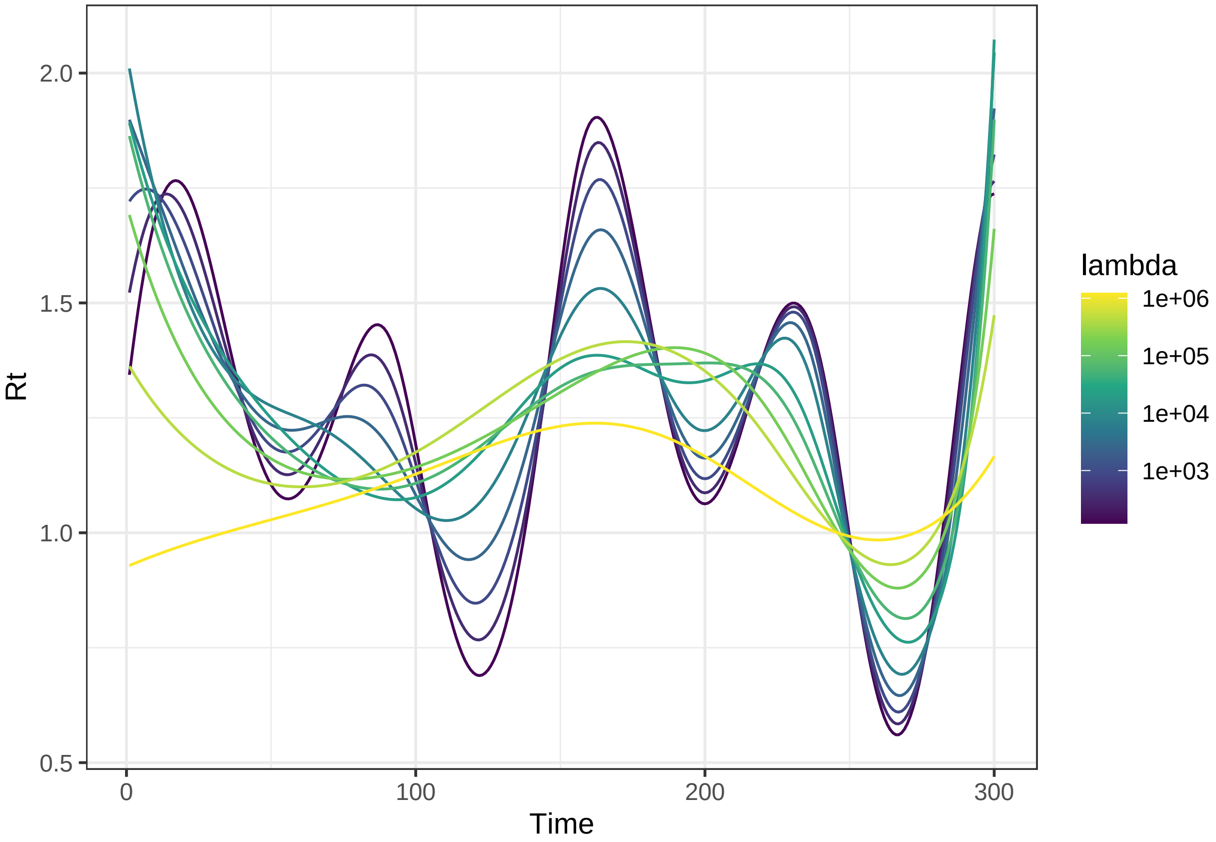
```
rtestim_mod3 <- estimate_rt(incidence3, korder=1, nsol=10,
                              dist_gamma = c(3, 3))
plot(rtestim_mod3)
```



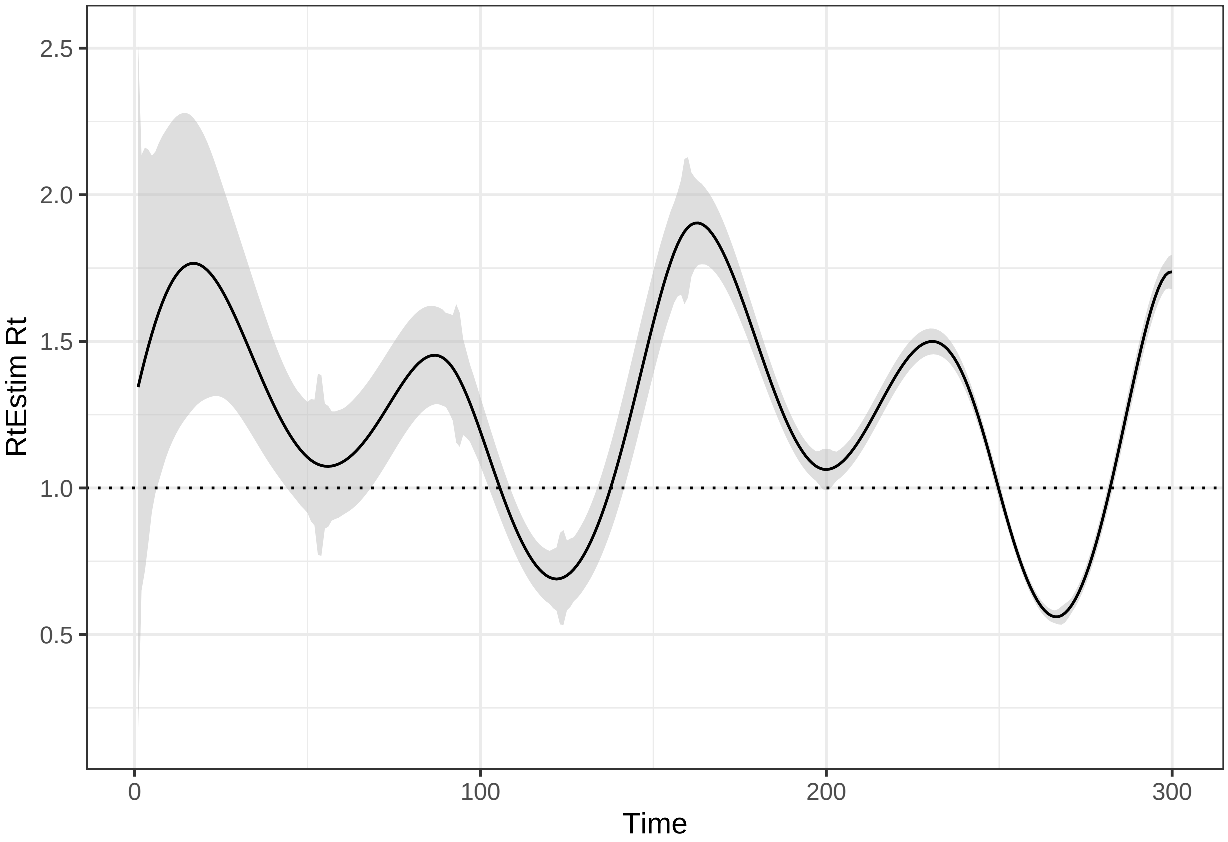
```
cv_mod3 <- cv_estimate_rt(incidence3, korder=1, nfold=3, nsol=10,
                          dist_gamma = c(3, 3))
rtestim_tuned_mod3 <- cv_mod3$full_fit$Rt[, which.min(cv_mod3$cv_scores)]
rt_ci3 <- confband(cv_mod3, "lambda.min") # get 95% confidence band
rt_ci3 %>%
  ggplot(aes(x = 1:length(incidence2), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



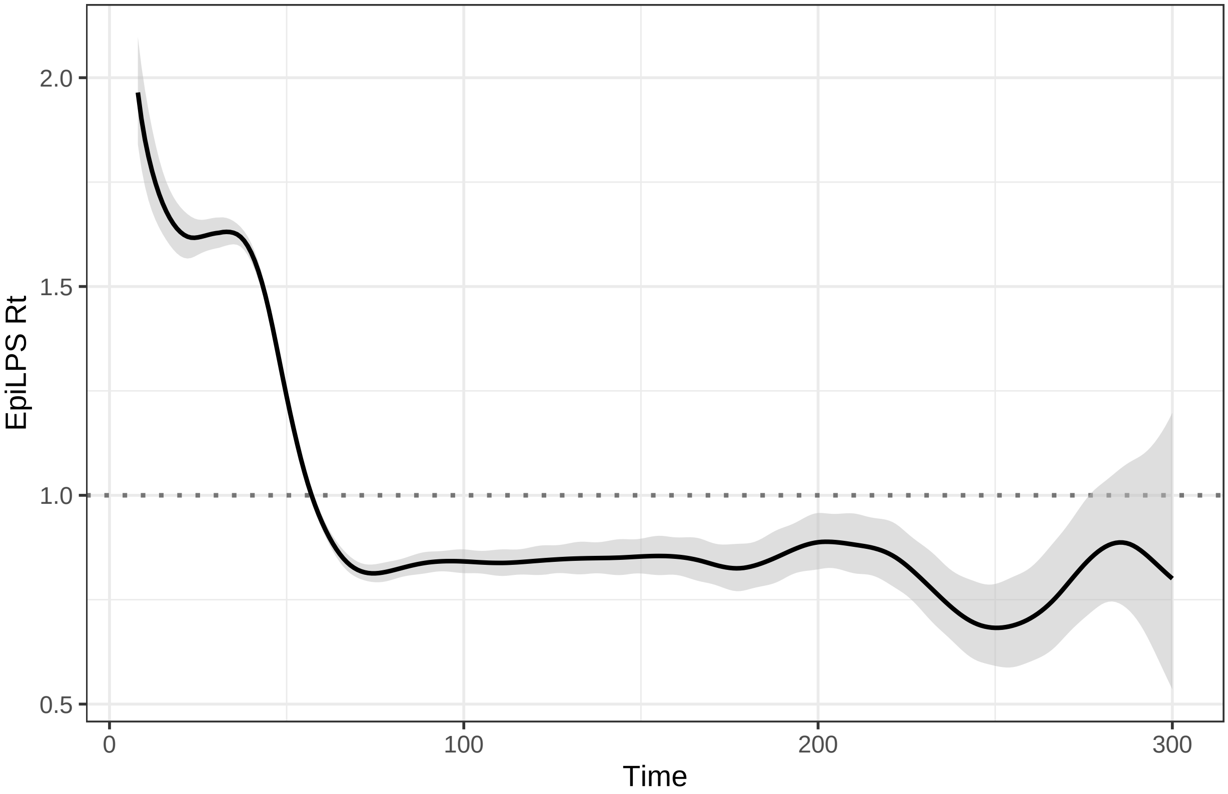
```r
rtestim_mod4 <- estimate_rt(incidence4, korder=3, nsol=10,
                            dist_gamma = c(3, 3))
plot(rtestim_mod4)
```



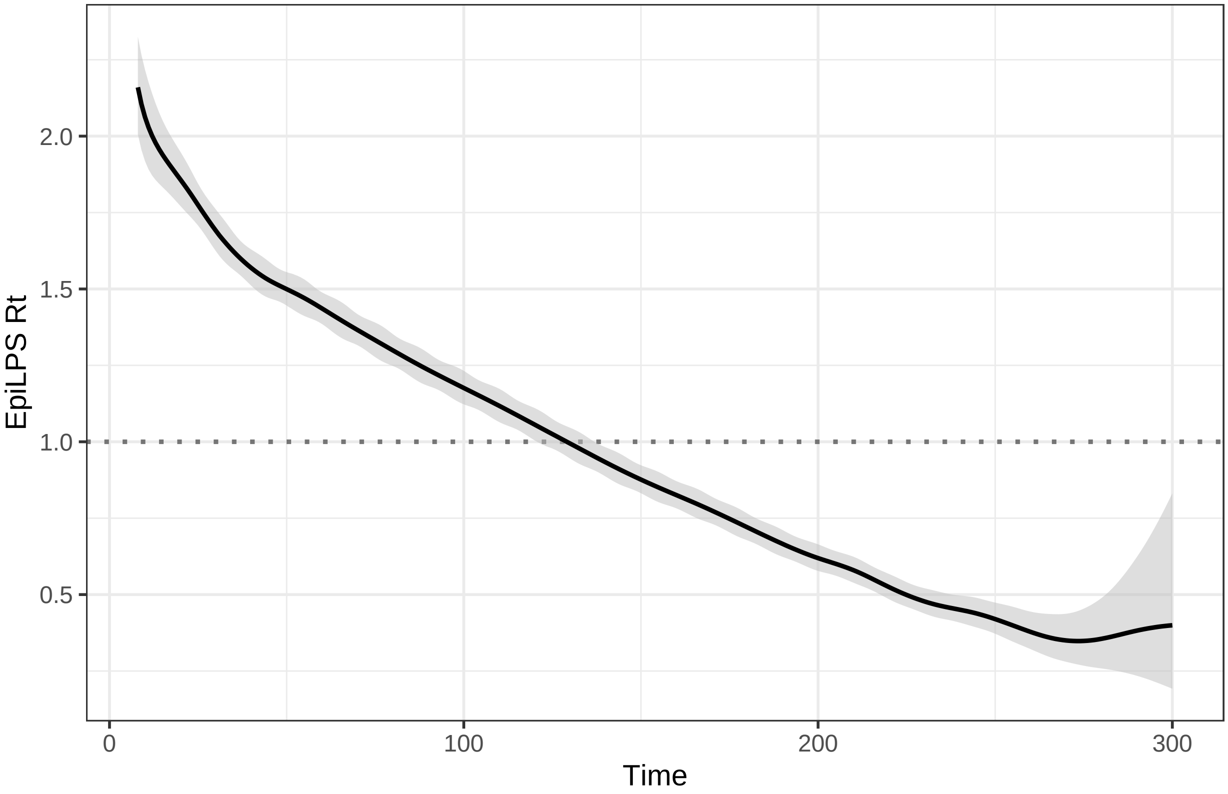
```r
cv_mod4 <- cv_estimate_rt(incidence4, korder=3, nfold=3, nsol=10, maxiter=1e7L,
                          dist_gamma = c(3, 3))
rtestim_tuned_mod4 <- cv_mod4$full_fit$Rt[, which.min(cv_mod4$cv_scores)]
rt_ci4 <- confband(cv_mod4, "lambda.min") # get 95% confidence band
rt_ci4 %>%
  ggplot(aes(x = 1:length(incidence2), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



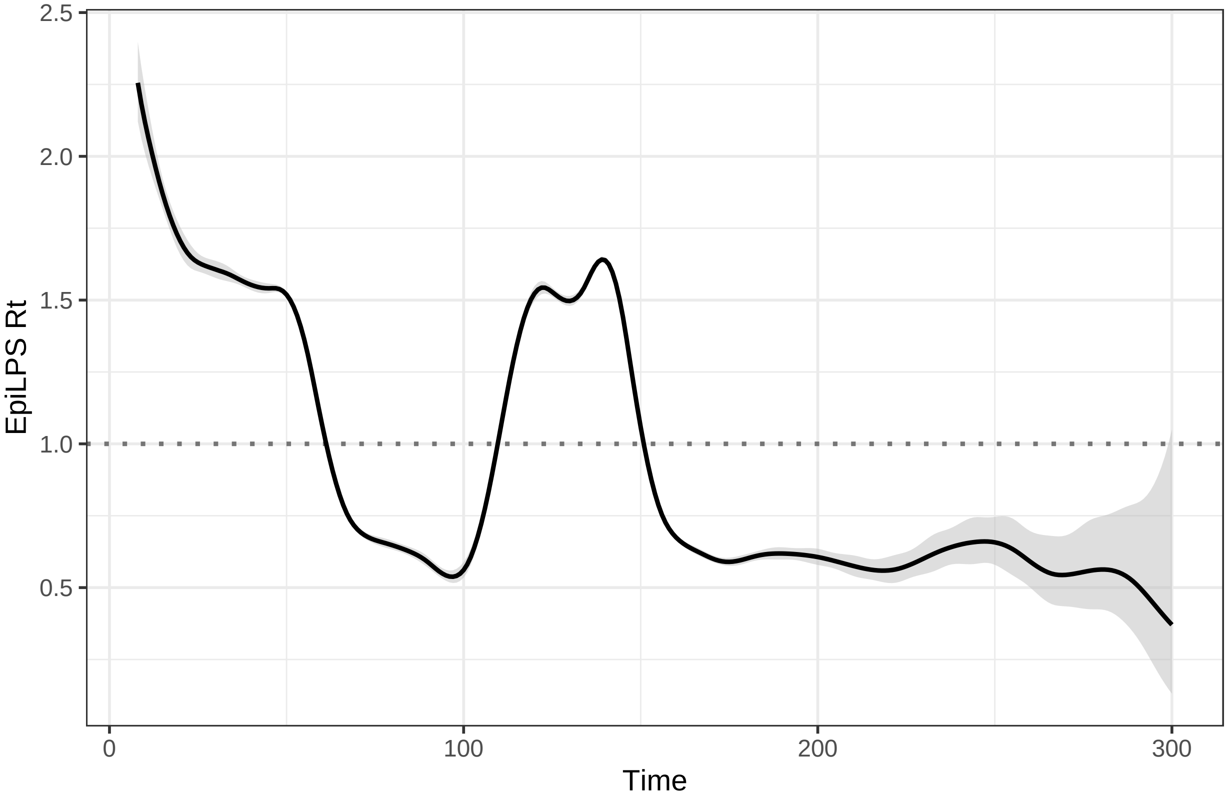
```
library(EpiLPS)
si1 <- Idist(mean = 9, sd = 27, dist="gamma")$pvec[1:30]
si1 <- si1 / sum(si1)
si2 <- Idist(mean = 3.5^2, sd = 3.5^3, dist="gamma")$pvec[1:30]
si2 <- si2 / sum(si2)

mod_epilst_pois1 <- estimR(incidence = incidence1, si = si1)
plot(mod_epilst_pois1) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```



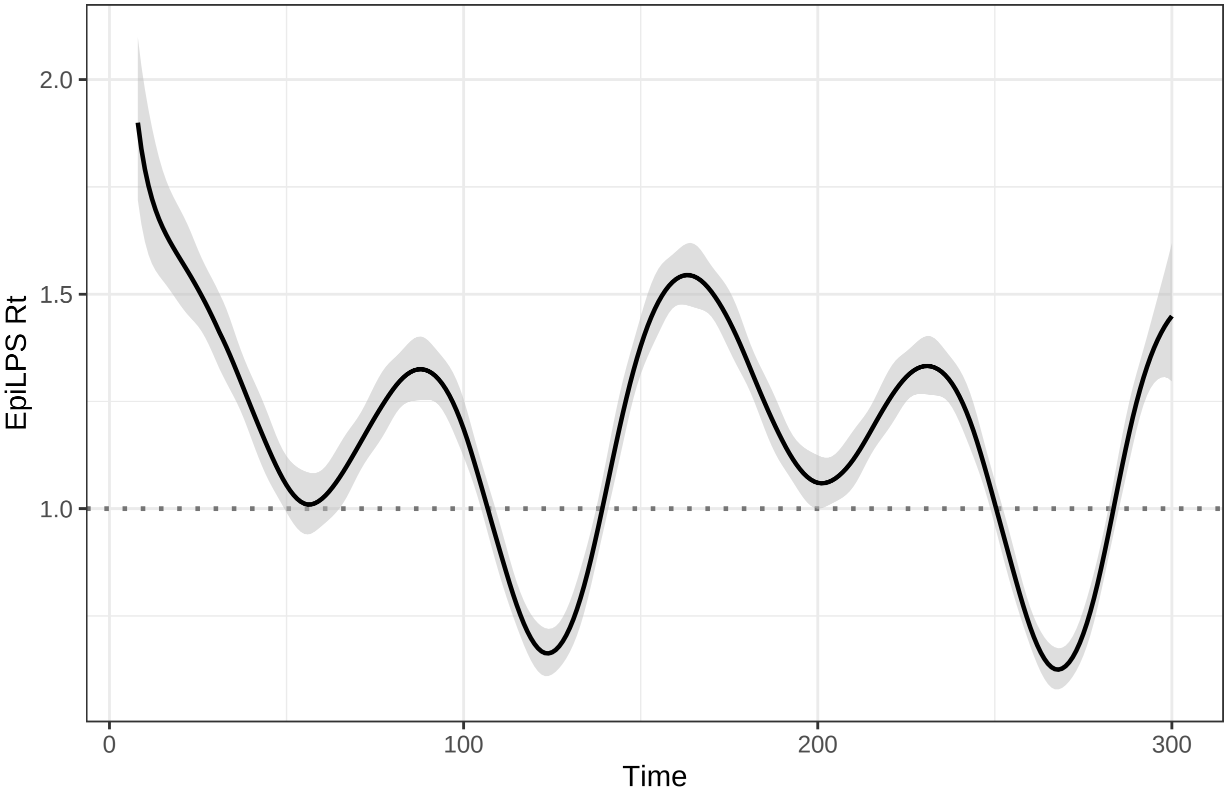
```
mod_epilst_pois2 <- estimR(incidence = incidence2, si = si2)
plot(mod_epilst_pois2) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```



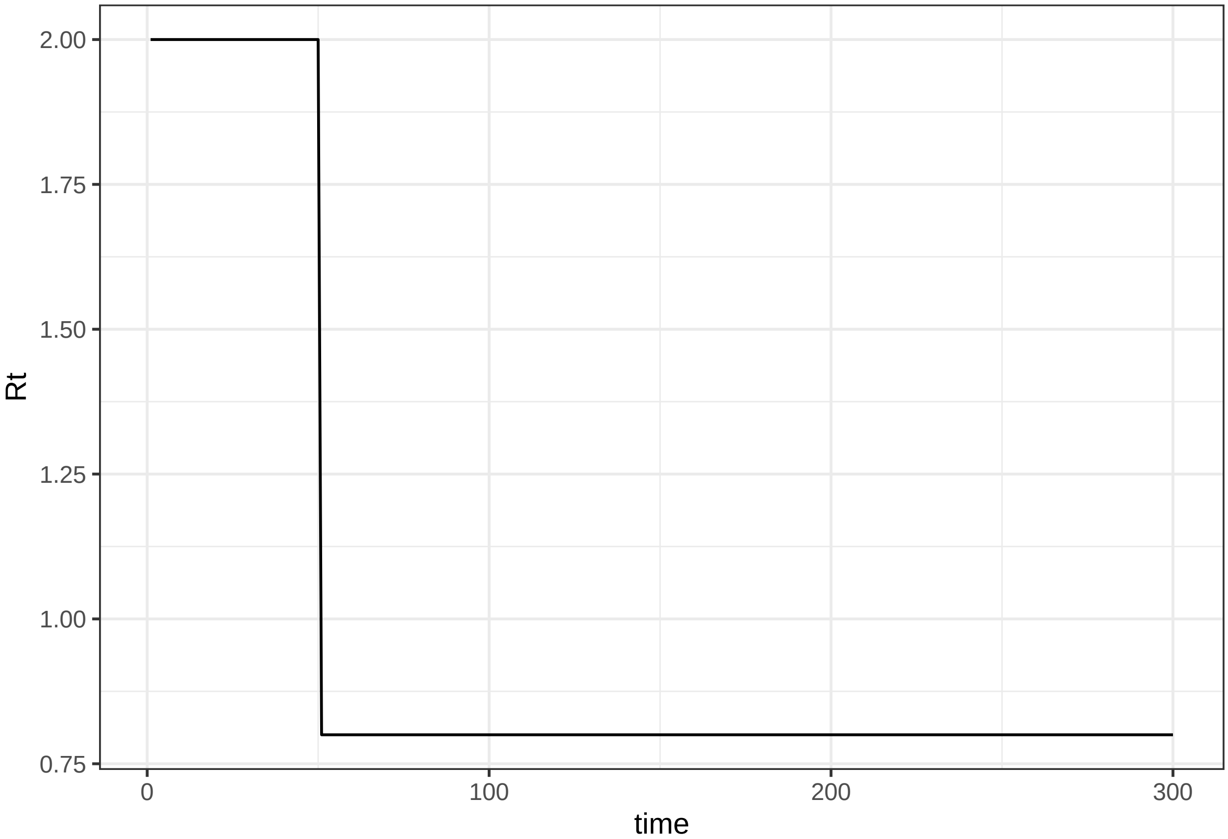
```
mod_epilst_pois3 <- estimR(incidence = incidence3, si = si2)
plot(mod_epilst_pois3) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```

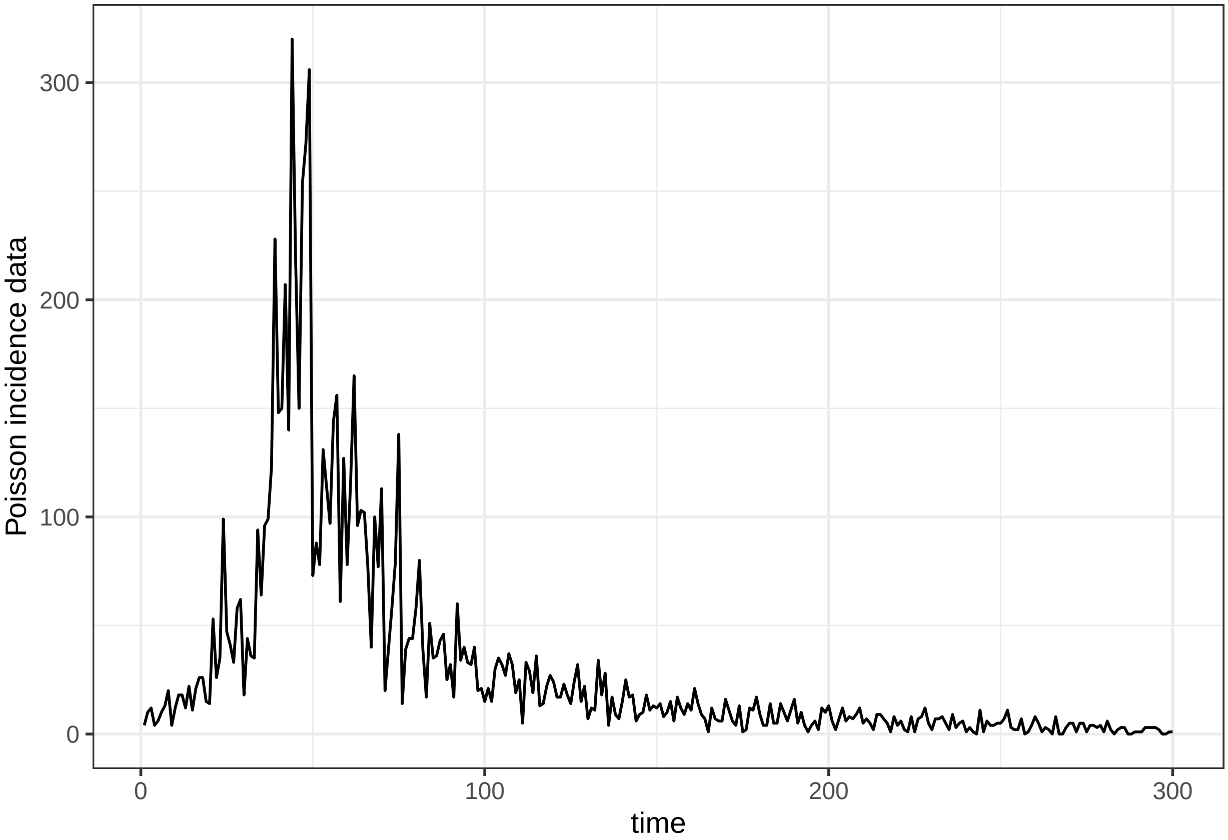


```
mod_epilst_pois4 <- estimR(incidence = incidence4, si = si2)
plot(mod_epilst_pois4) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```

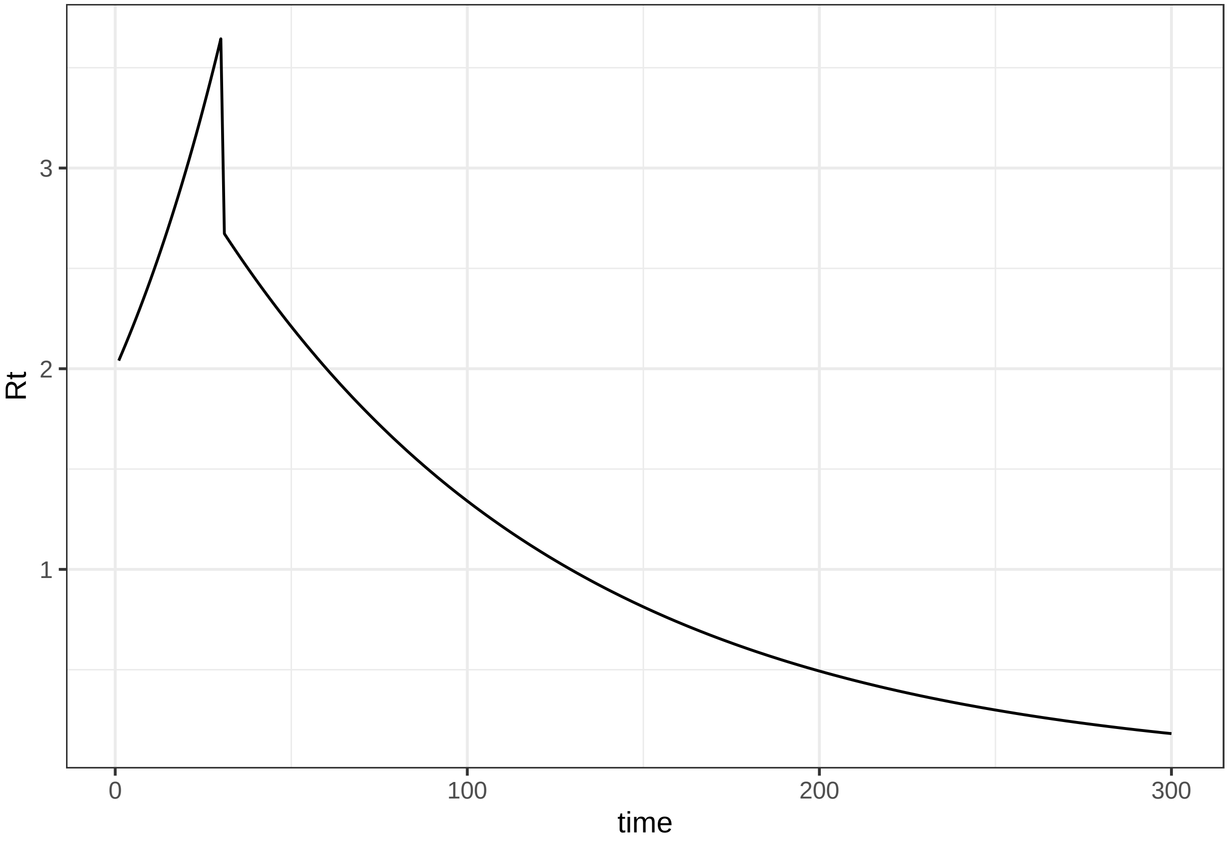


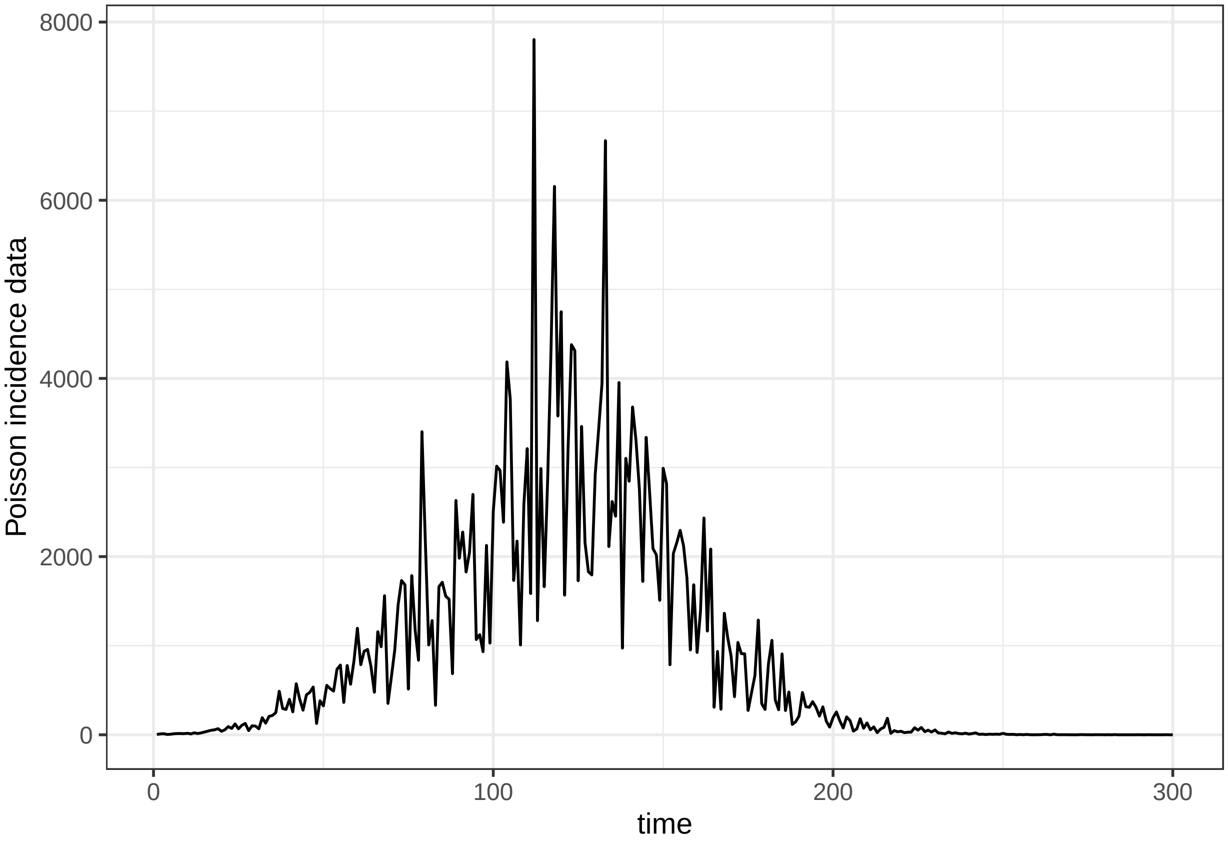
```r
get_nb_incidence <- function(N1, Rt, gamma_pars = c(2.5, 2.5)){
  len <- length(Rt)
  incidence <- numeric(len) # N_1:n
  NB_count <- numeric(len) # y_1:n
  incidence[1] <- N1
  size = 5

  NB_count[1] <- rnbinom(1, mu=N1, size=size)
  if(NB_count[1] == 0) NB_count[1] = 1
  for(t in 2:len){
    pi <- discretize_gamma(1:(t-1), gamma_pars[1], gamma_pars[2])
    incidence[t] <- Rt[t] * sum(rev(pi) * NB_count[1:(t-1)])
    NB_count[t] <- rnbinom(1, mu = incidence[t], size=size)
  }

  return(NB_count)
}

# case 1
seed <- 629
set.seed(seed)
nb_incidence1 <- get_nb_incidence(N1, Rt1, gamma_pars1)
display_dat(nb_incidence1, Rt1)
```


2.00
1.75
1.50
1.25
1.00
0.75
Rt
0
100
200
300
time

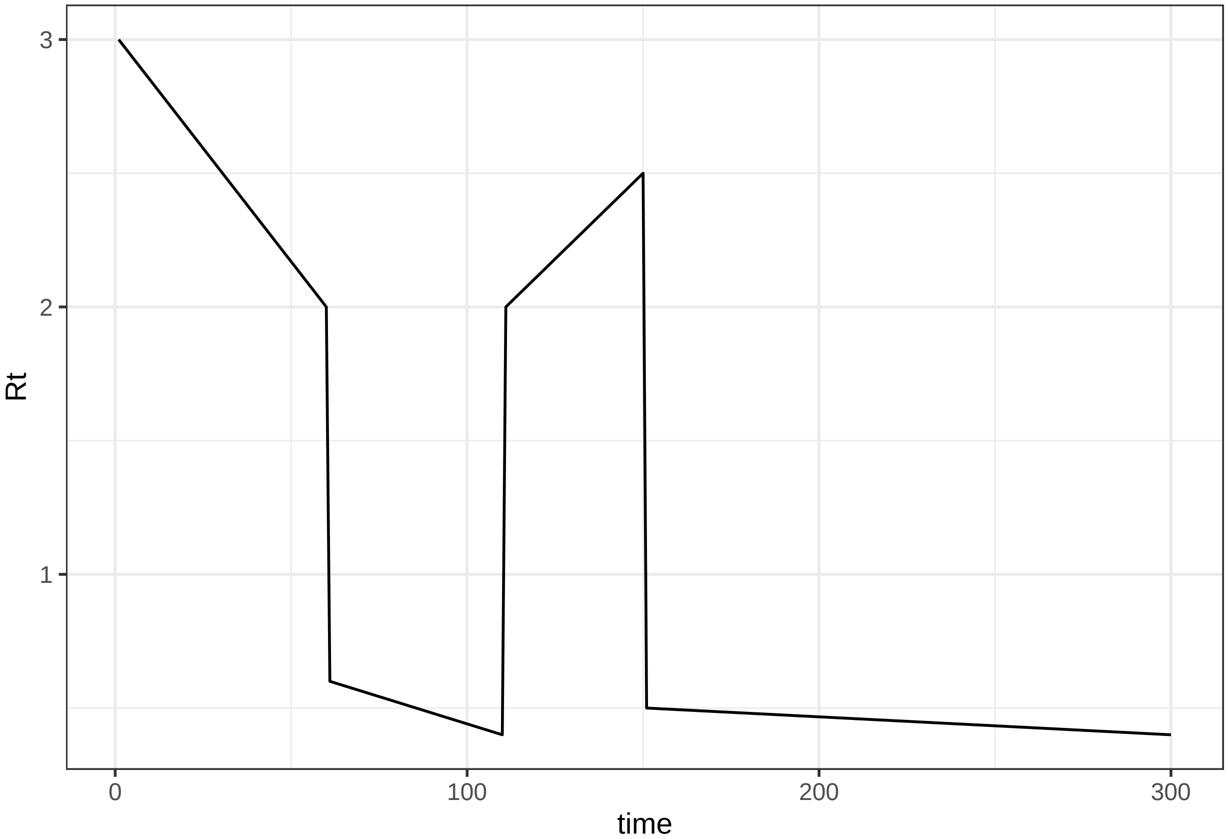

300
200
100
0
Poisson incidence data
0
100
200
300
time

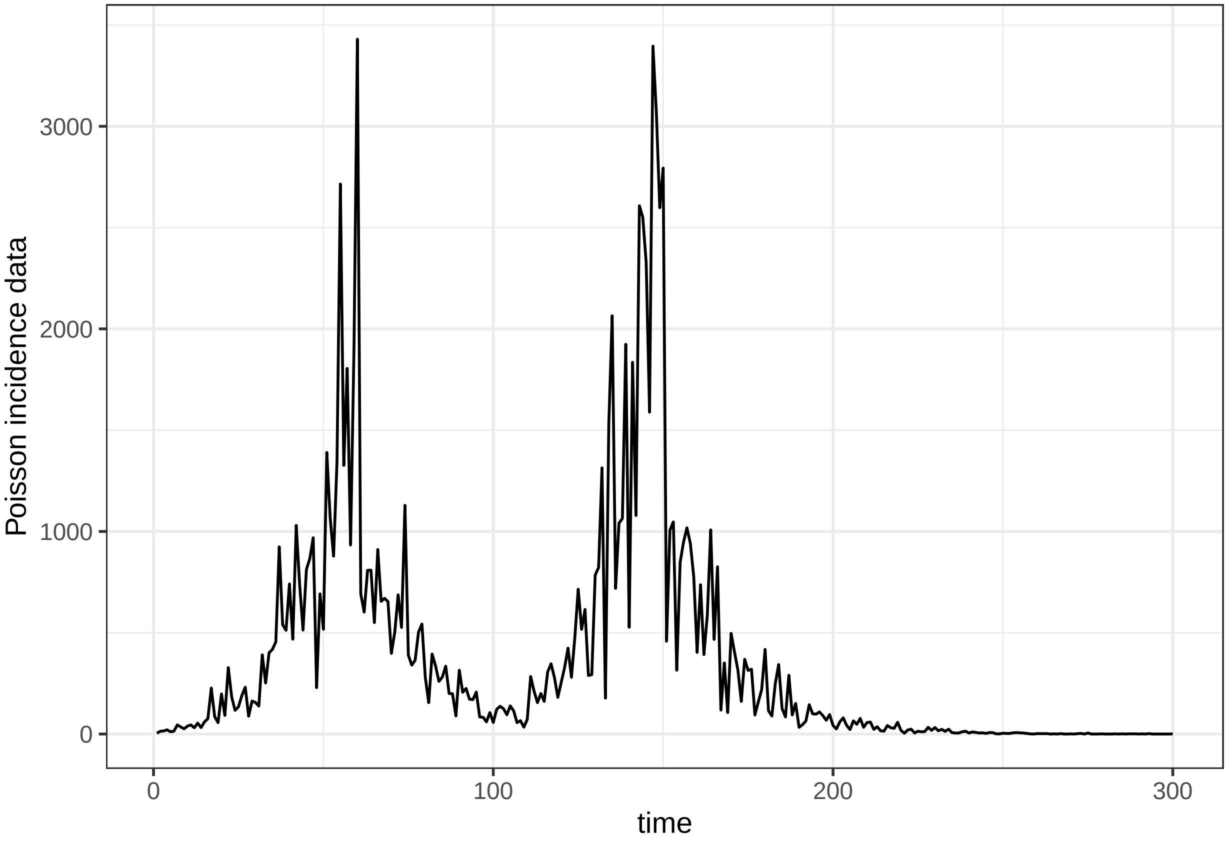
```
check_dat(nb_incidence1, Rt1)

# case 2
set.seed(seed)
nb_incidence2 <- get_nb_incidence(N1, Rt2, gamma_pars2)
display_dat(nb_incidence2, Rt2)
```



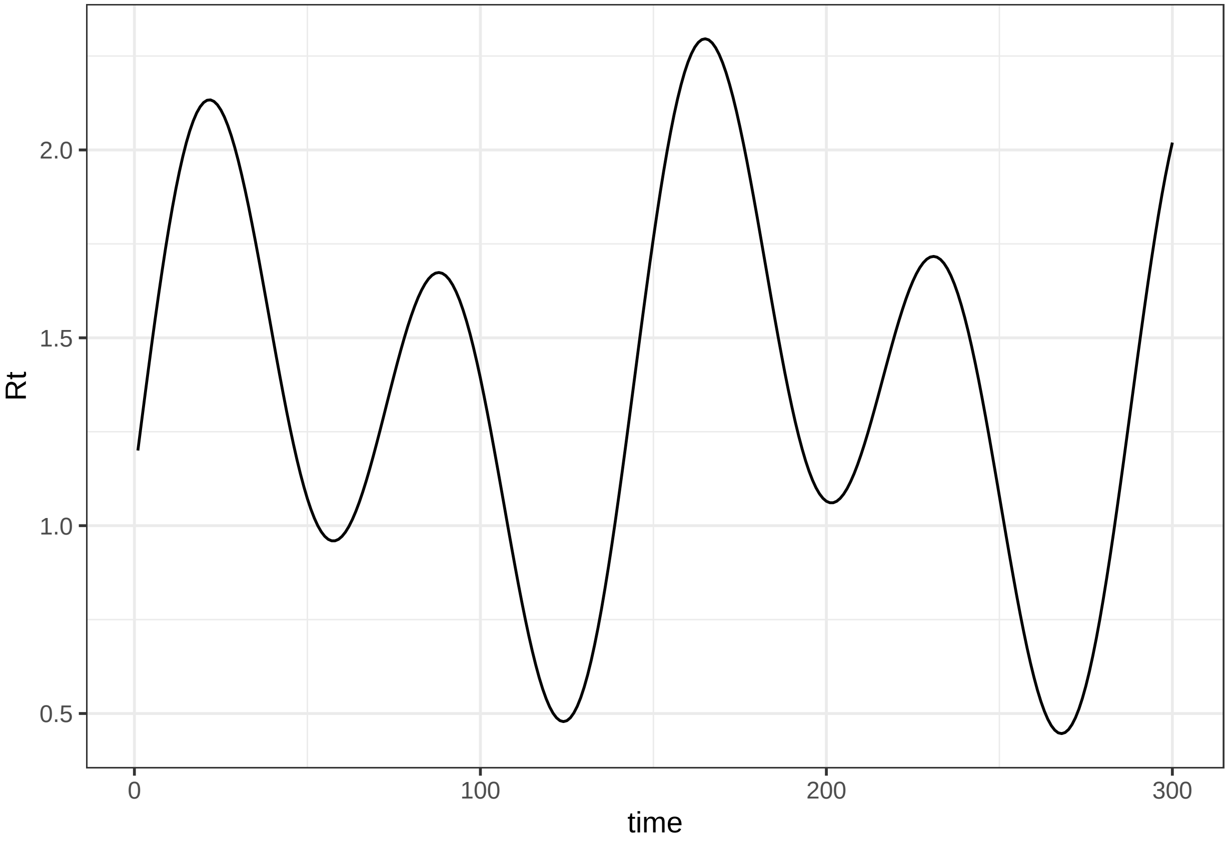


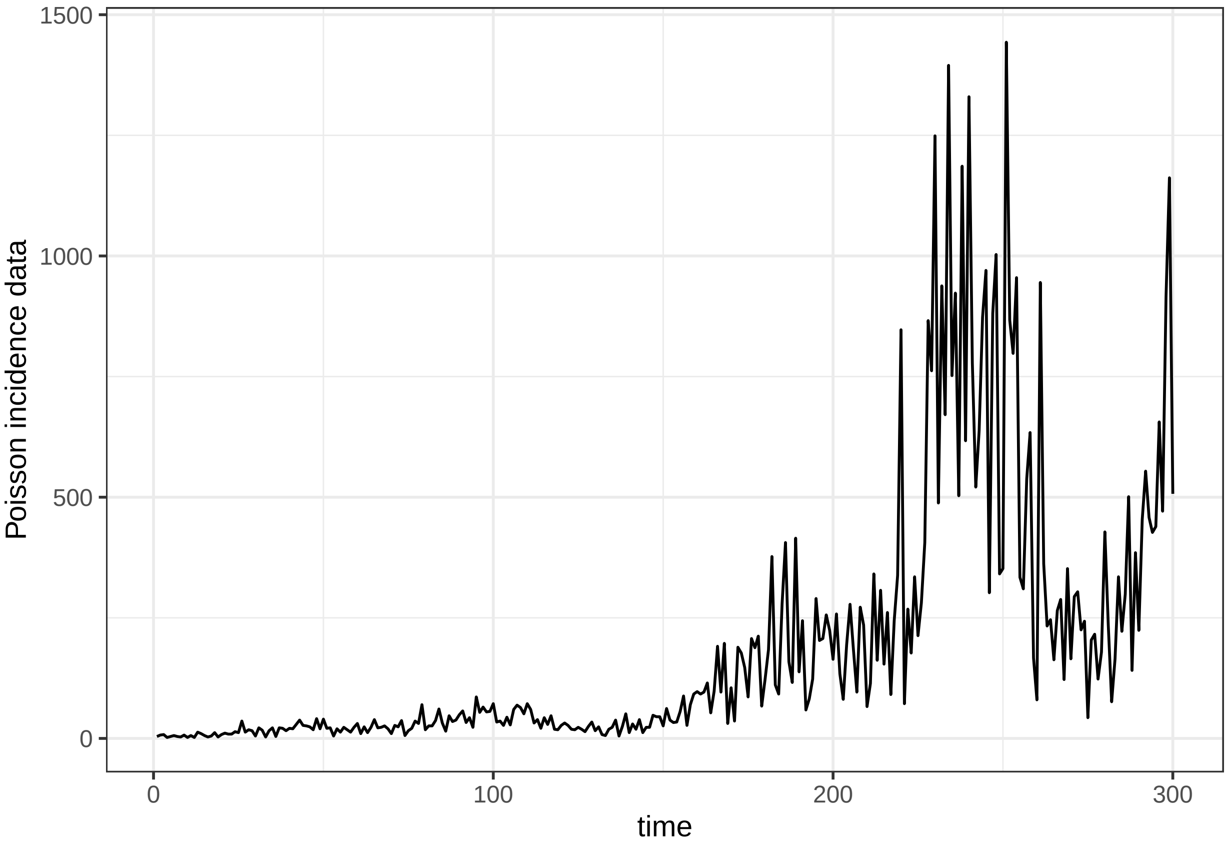
```r
check_dat(nb_incidence2, Rt2)

# case 3
set.seed(seed)
nb_incidence3 <- get_nb_incidence(N1, Rt3, gamma_pars3)
display_dat(nb_incidence3, Rt3)
```


3
2
1
Rt
0
100
200
300
time


3000
2000
1000
0
Poisson incidence data
0
100
200
300
time

```
check_dat(nb_incidence3, Rt3)

# case 4
set.seed(seed)
nb_incidence4 <- get_nb_incidence(N1, Rt4, gamma_pars4)
display_dat(nb_incidence4, Rt4)
```



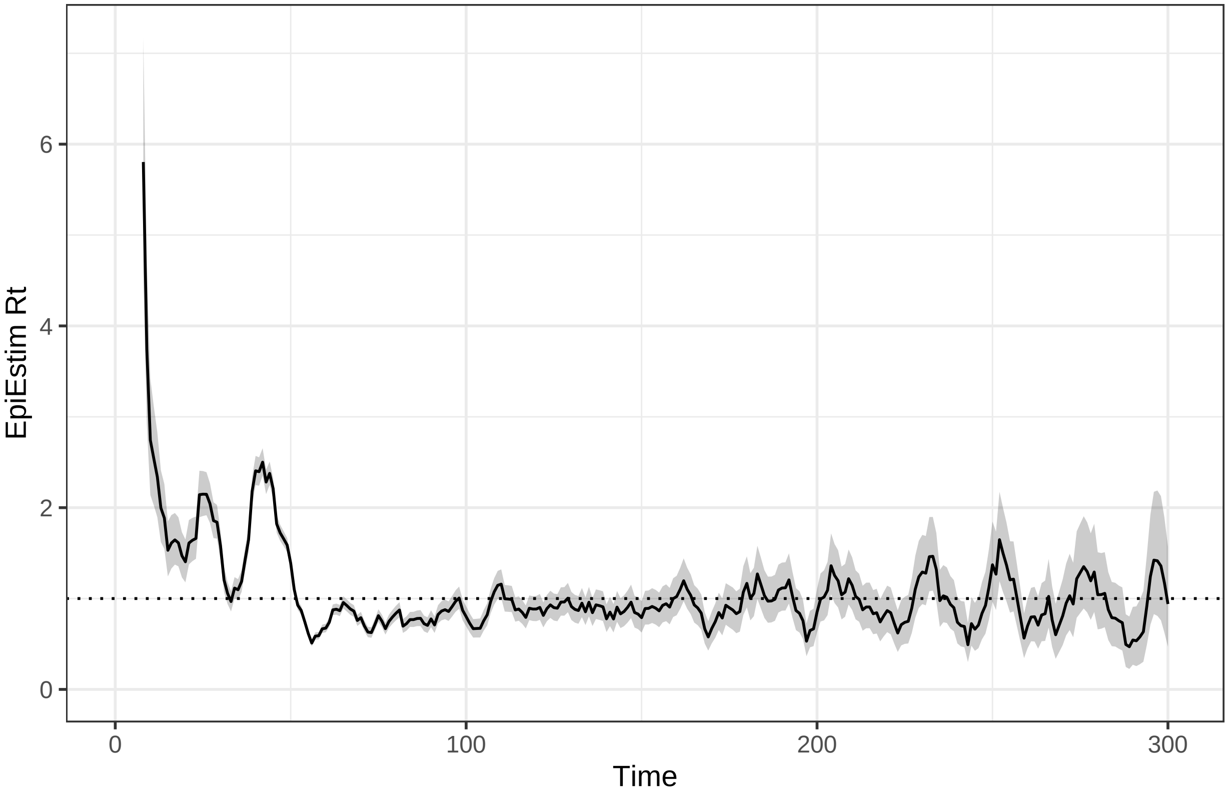


```
check_dat(nb_incidence4, Rt4)
```

```
mod_epi_nb1 <- EpiEstim::estimate_R(incid = nb_incidence1, config = config1, method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

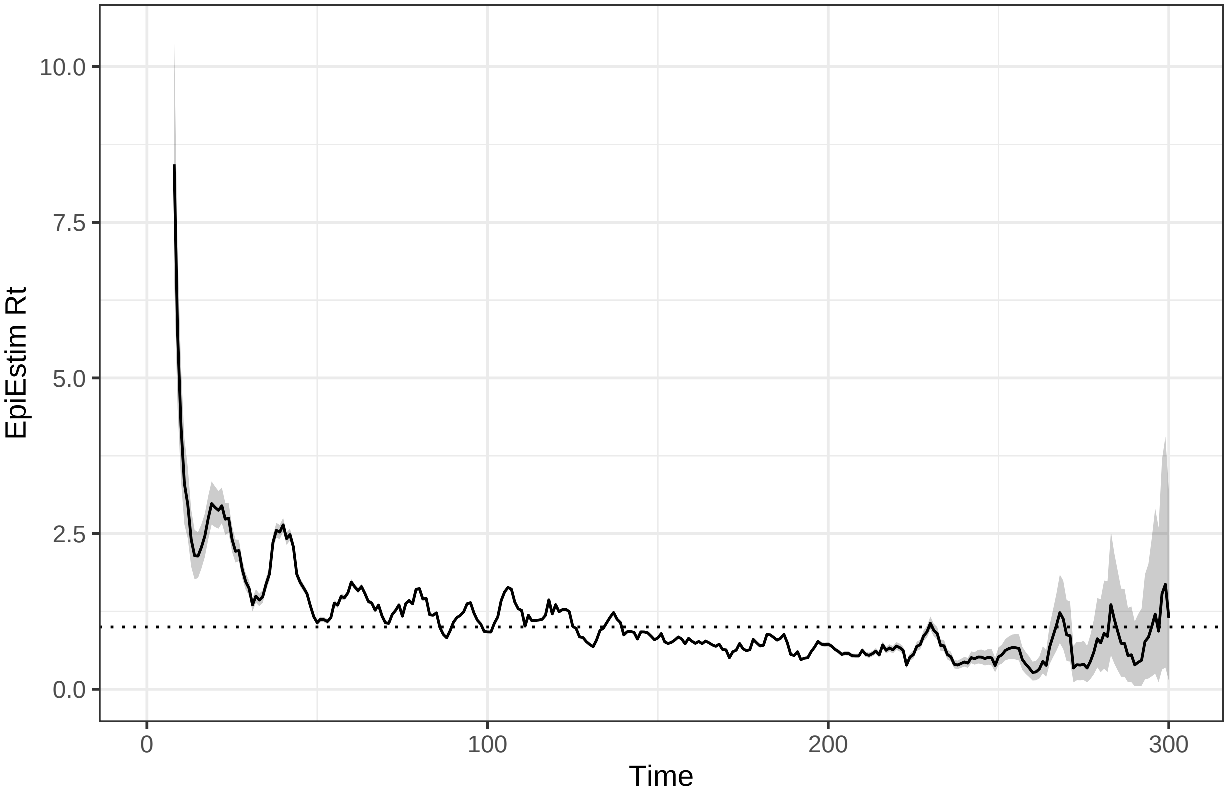
```
plot(mod_epi_nb1, "R")  + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none"
```



```
mod_epi_nb2 <- EpiEstim::estimate_R(incid = nb_incidence2, config = config2, method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

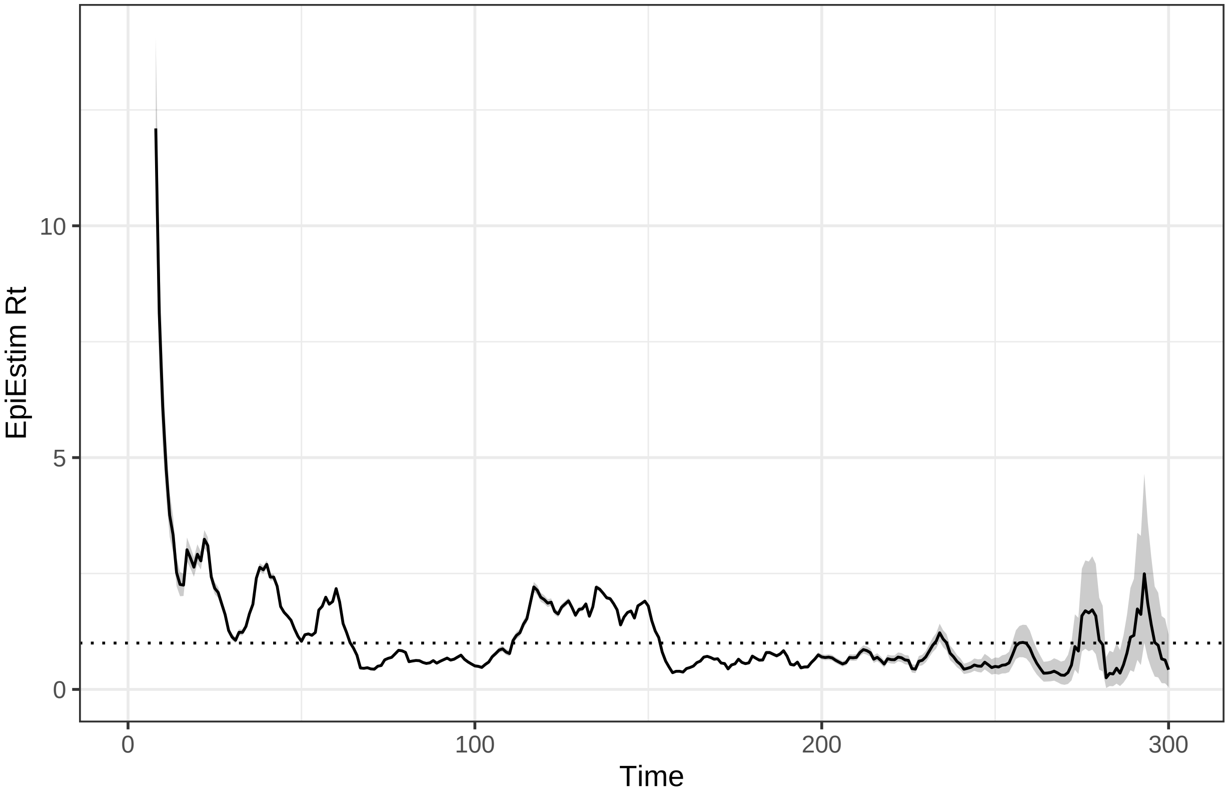
```
plot(mod_epi_nb2, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none")
```



```
mod_epi_nb3 <- EpiEstim::estimate_R(incid = nb_incidence3, config = config2, method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

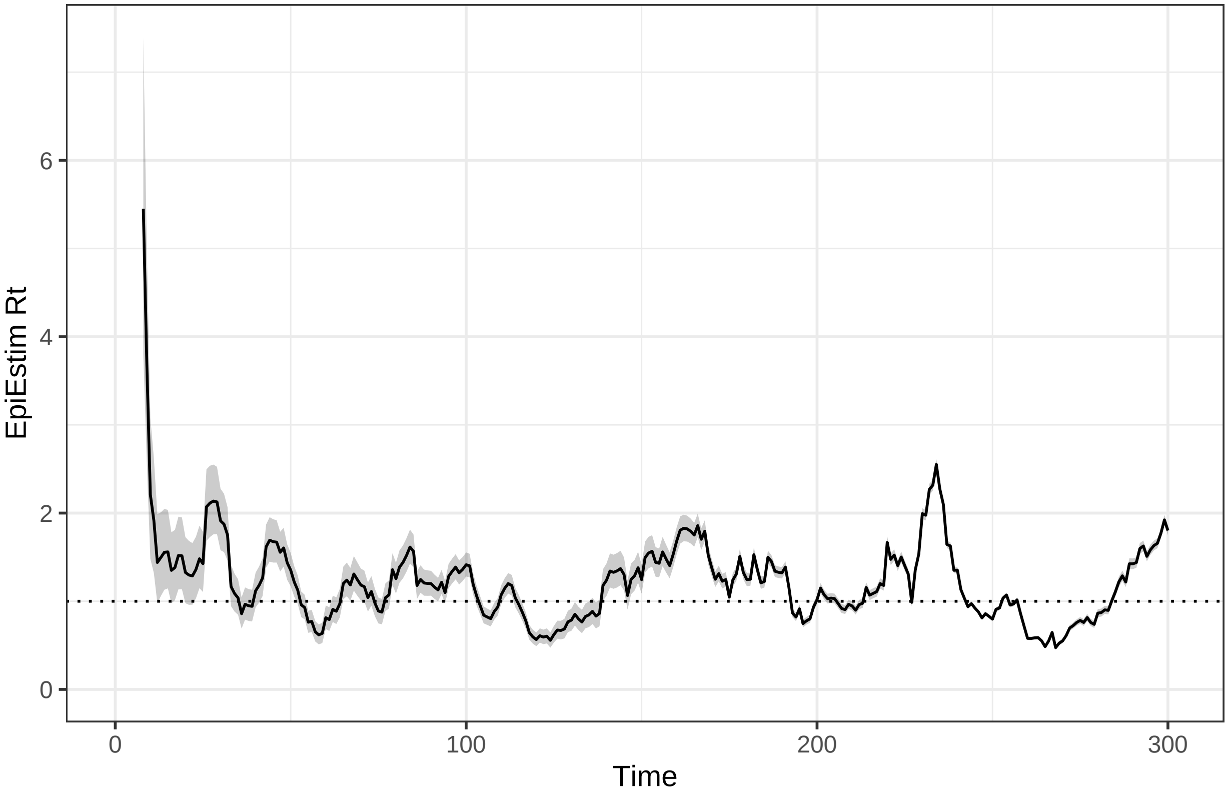
```
plot(mod_epi_nb3, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none")
```



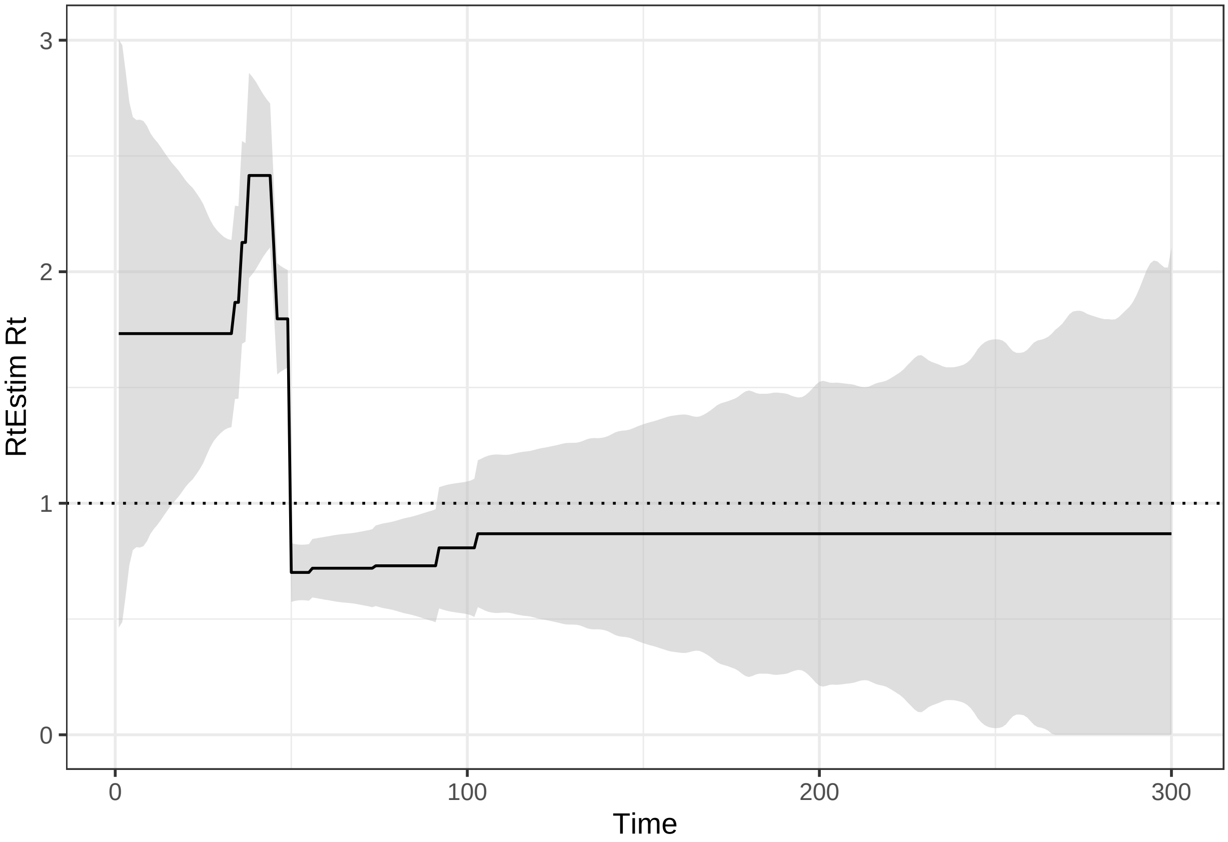
```
mod_epi_nb4 <- EpiEstim::estimate_R(incid = nb_incidence4, config = config2, method = method)
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

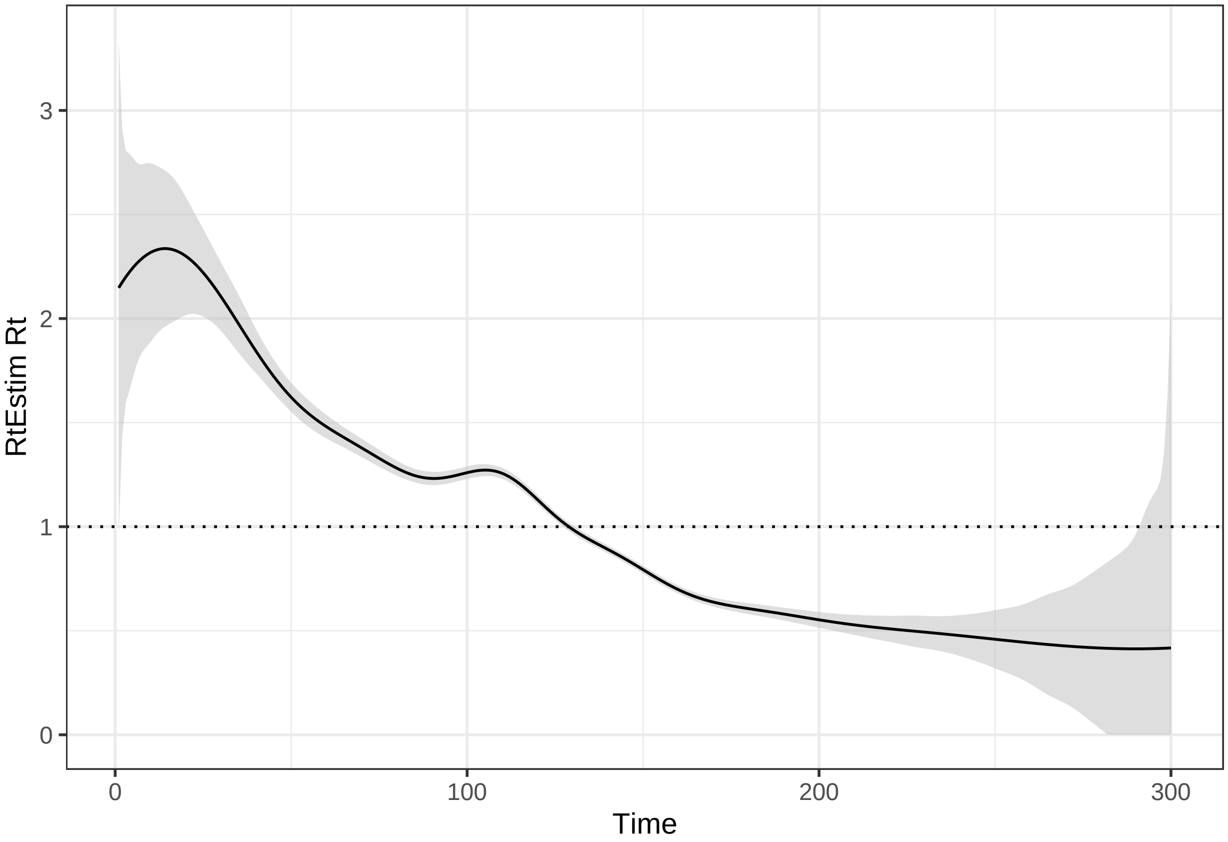
```
plot(mod_epi_nb4, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none")
```



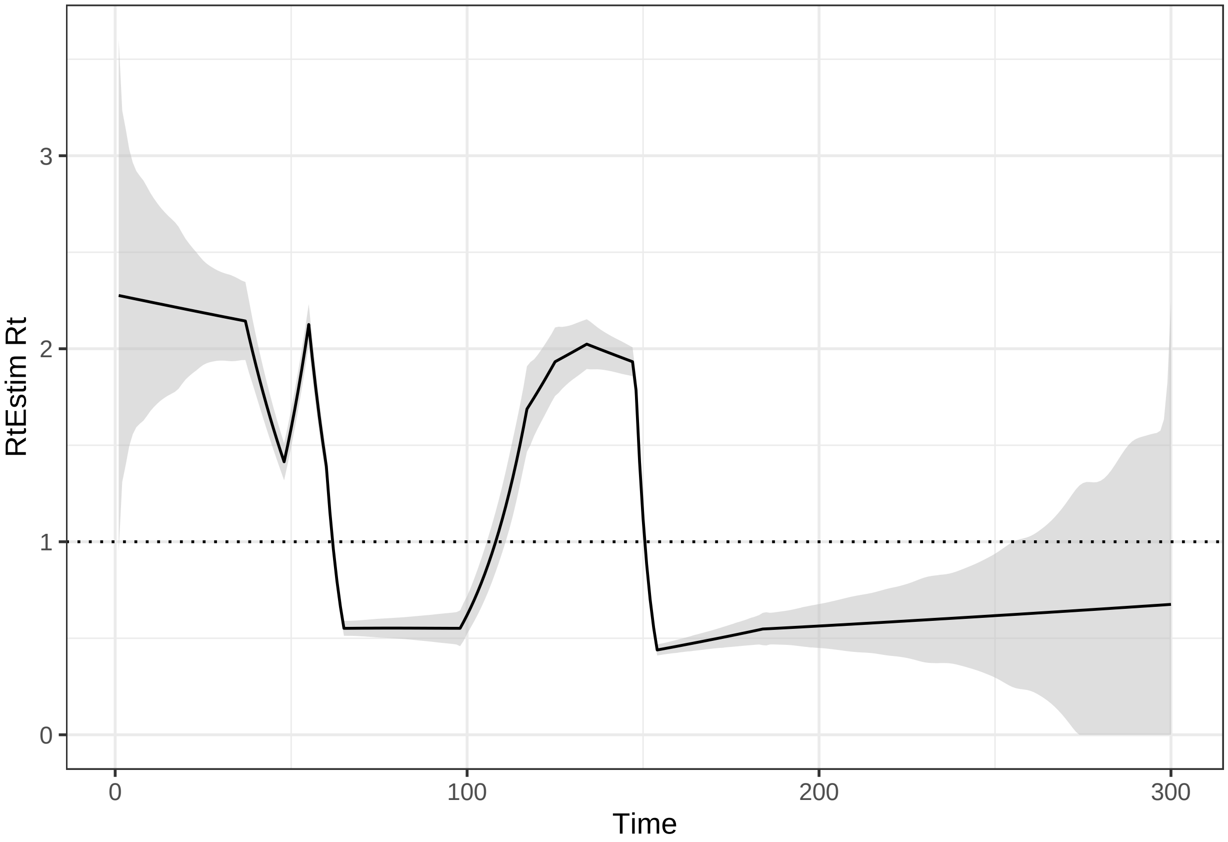
```
cv_mod_nb1 <- cv_estimate_rt(nb_incidence1, korder=0, nfold=3, nsol=10,
                             dist_gamma = c(3, 3))
rtestim_tuned_mod_nb1 <- cv_mod_nb1$full_fit$Rt[ ,which.min(cv_mod_nb1$cv_scores)]
rt_ci_nb1 <- confband(cv_mod_nb1, "lambda.min") # get 95% confidence band
rt_ci_nb1 %>%
  ggplot(aes(x = 1:length(nb_incidence1), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



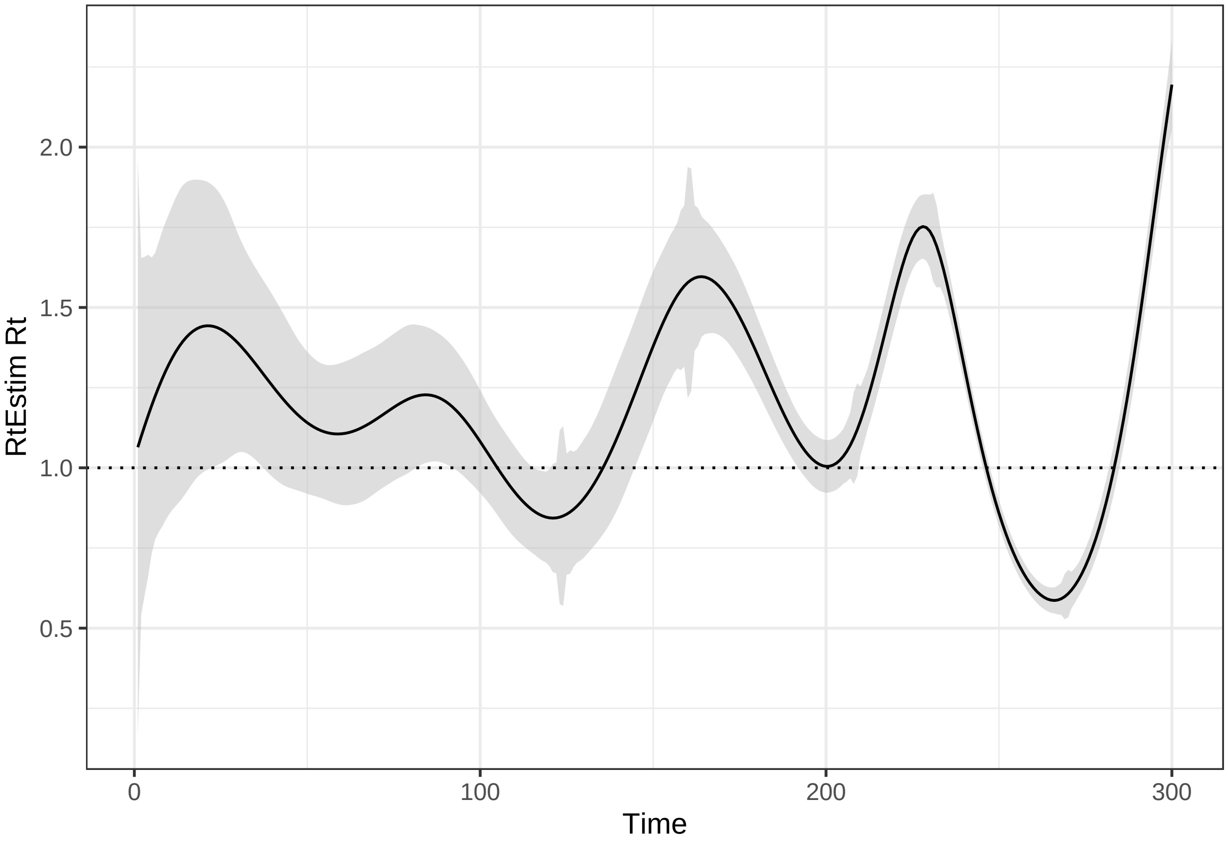
```r
cv_mod_nb2 <- cv_estimate_rt(nb_incidence2, korder=3, nfold=3, nsol=10,
                             dist_gamma = c(3, 3))
rtestim_tuned_mod_nb2 <- cv_mod_nb2$full_fit$Rt[ ,which.min(cv_mod_nb2$cv_scores)]
rt_ci_nb2 <- confband(cv_mod_nb2, "lambda.min") # get 95% confidence band
rt_ci_nb2 %>%
  ggplot(aes(x = 1:length(nb_incidence2), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



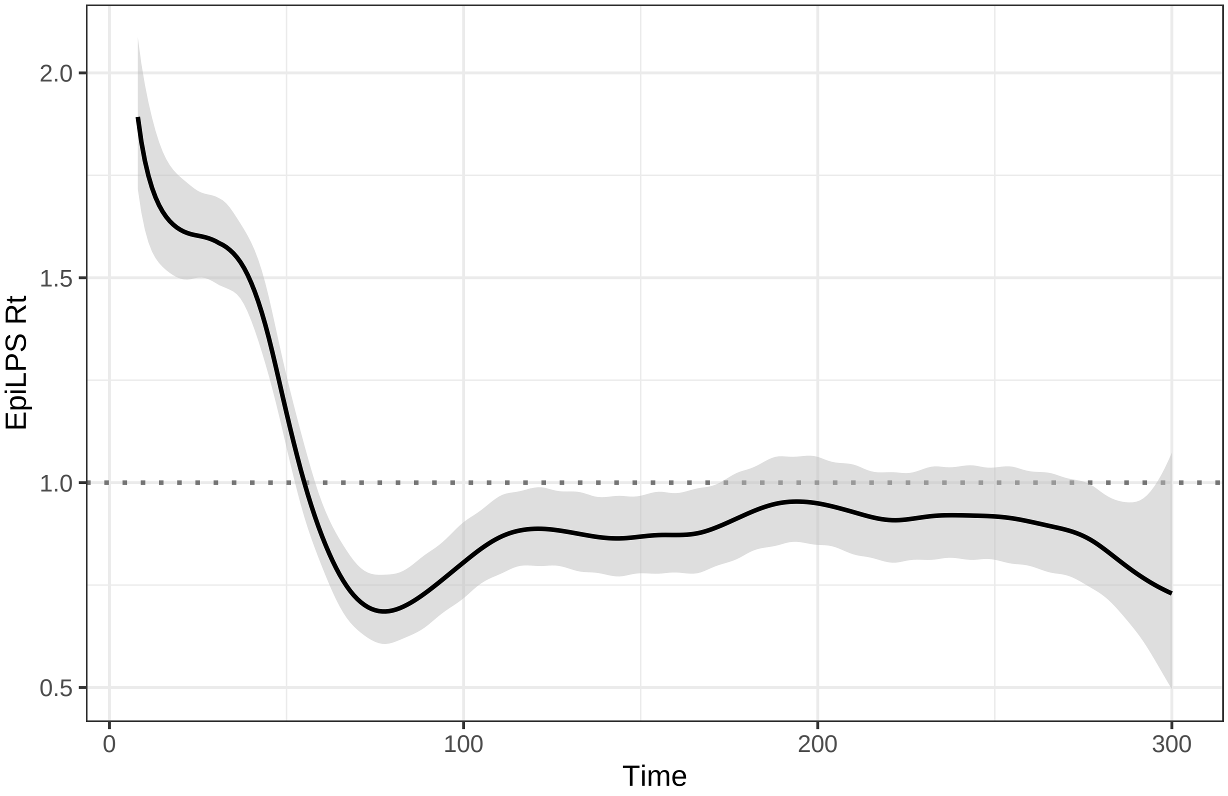
```
cv_mod_nb3 <- cv_estimate_rt(nb_incidence3, korder=1, nfold=3, nsol=10,
                             dist_gamma = c(3, 3))
rtestim_tuned_mod_nb3 <- cv_mod_nb3$full_fit$Rt[ ,which.min(cv_mod_nb3$cv_scores)]
rt_ci_nb3 <- confband(cv_mod_nb3, "lambda.min") # get 95% confidence band
rt_ci_nb3 %>%
  ggplot(aes(x = 1:length(nb_incidence3), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



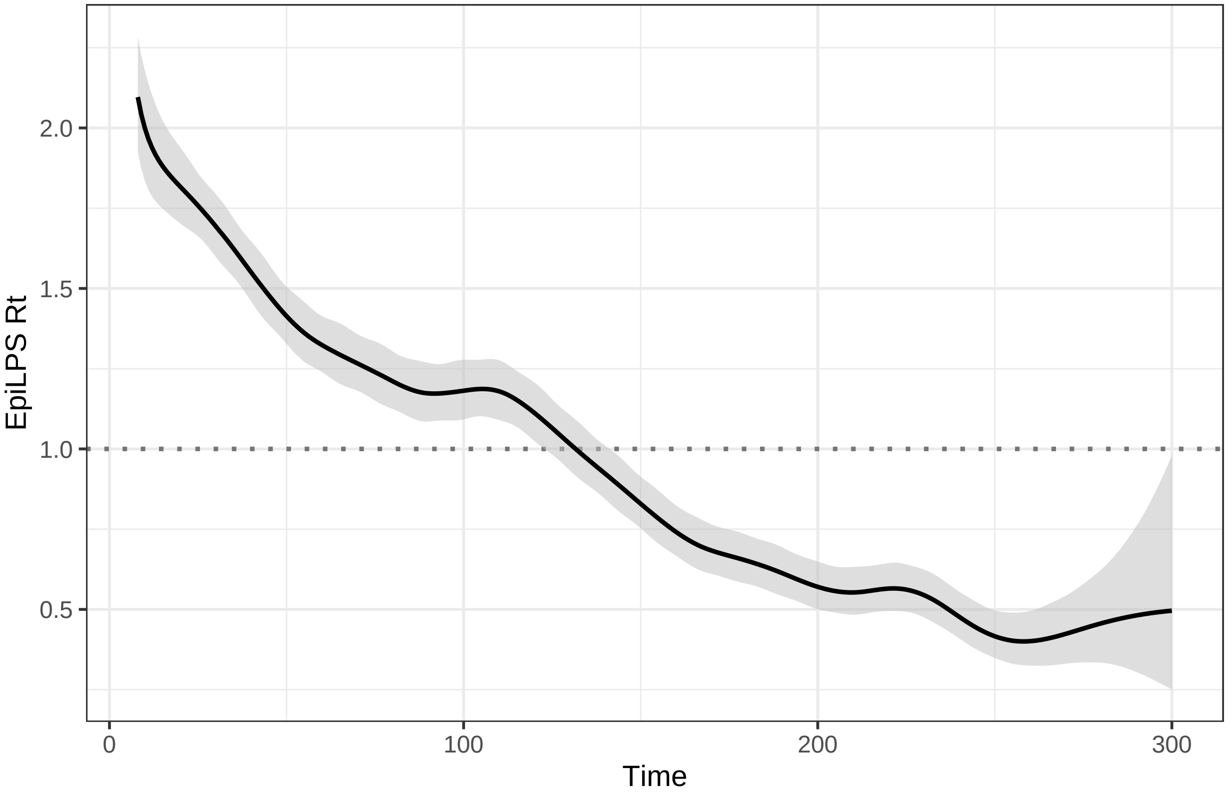
```
cv_mod_nb4 <- cv_estimate_rt(nb_incidence4, korder=3, nfold=3, nsol=10,
                             dist_gamma = c(3, 3)) # k=2 or 3?
rtestim_tuned_mod_nb4 <- cv_mod_nb4$full_fit$Rt[ ,which.min(cv_mod_nb4$cv_scores)]
rt_ci_nb4 <- confband(cv_mod_nb4, "lambda.min") # get 95% confidence band
rt_ci_nb4 %>%
  ggplot(aes(x = 1:length(nb_incidence4), y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
              fill = "gray", alpha = 0.5) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



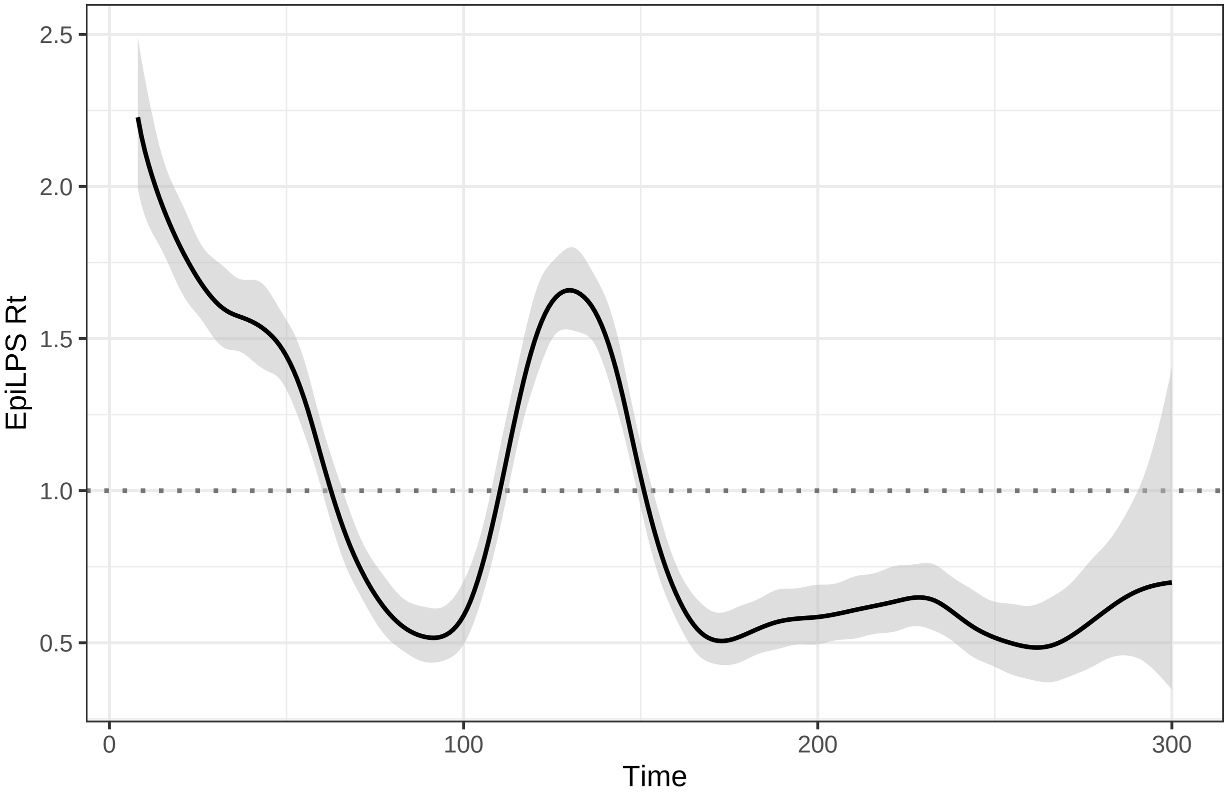
```
mod_epilst_nb1 <- estimR(incidence = nb_incidence1, si = si1, CoriR = TRUE)
plot(mod_epilst_nb1) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```



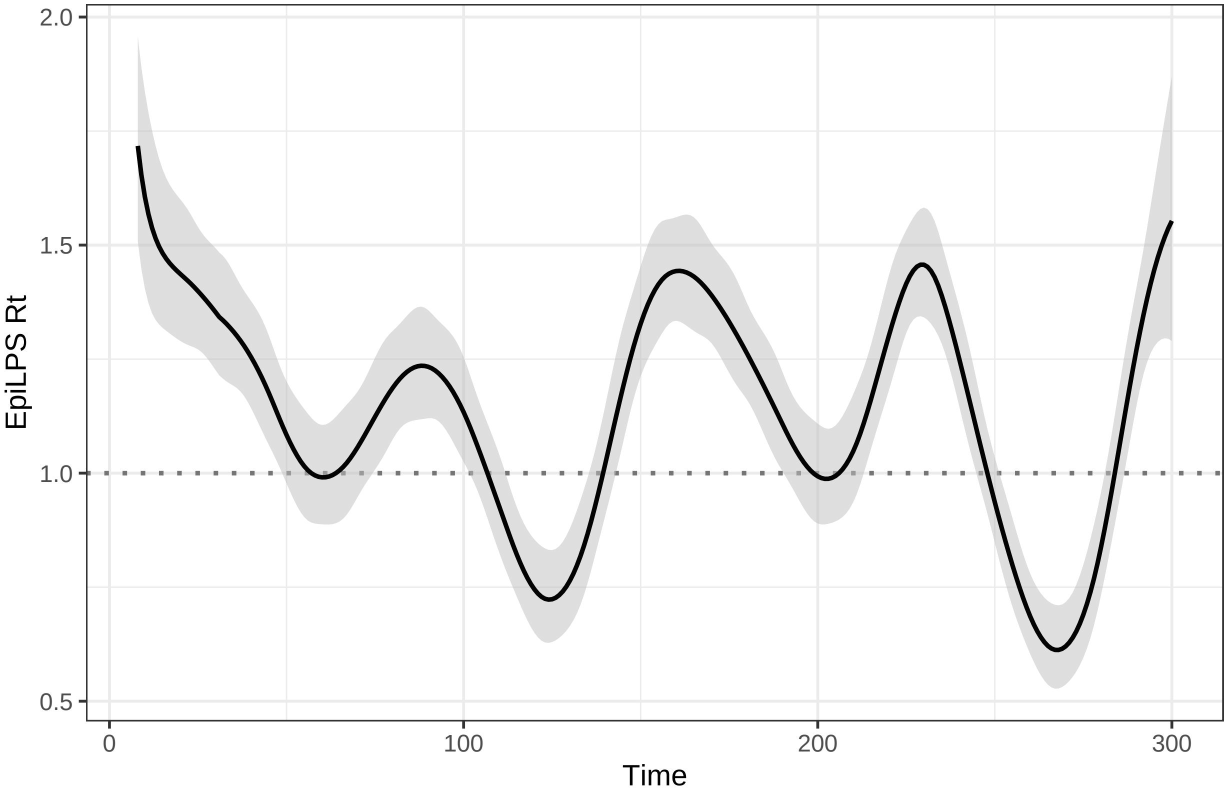
```
mod_epilst_nb2 <- estimR(incidence = nb_incidence2, si = si2, CoriR = TRUE)
plot(mod_epilst_nb2) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```



```
mod_epilst_nb3 <- estimR(incidence = nb_incidence3, si = si2, CoriR = TRUE)
plot(mod_epilst_nb3) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```



```
mod_epilst_nb4 <- estimR(incidence = nb_incidence4, si = si2, CoriR = TRUE)
plot(mod_epilst_nb4) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```

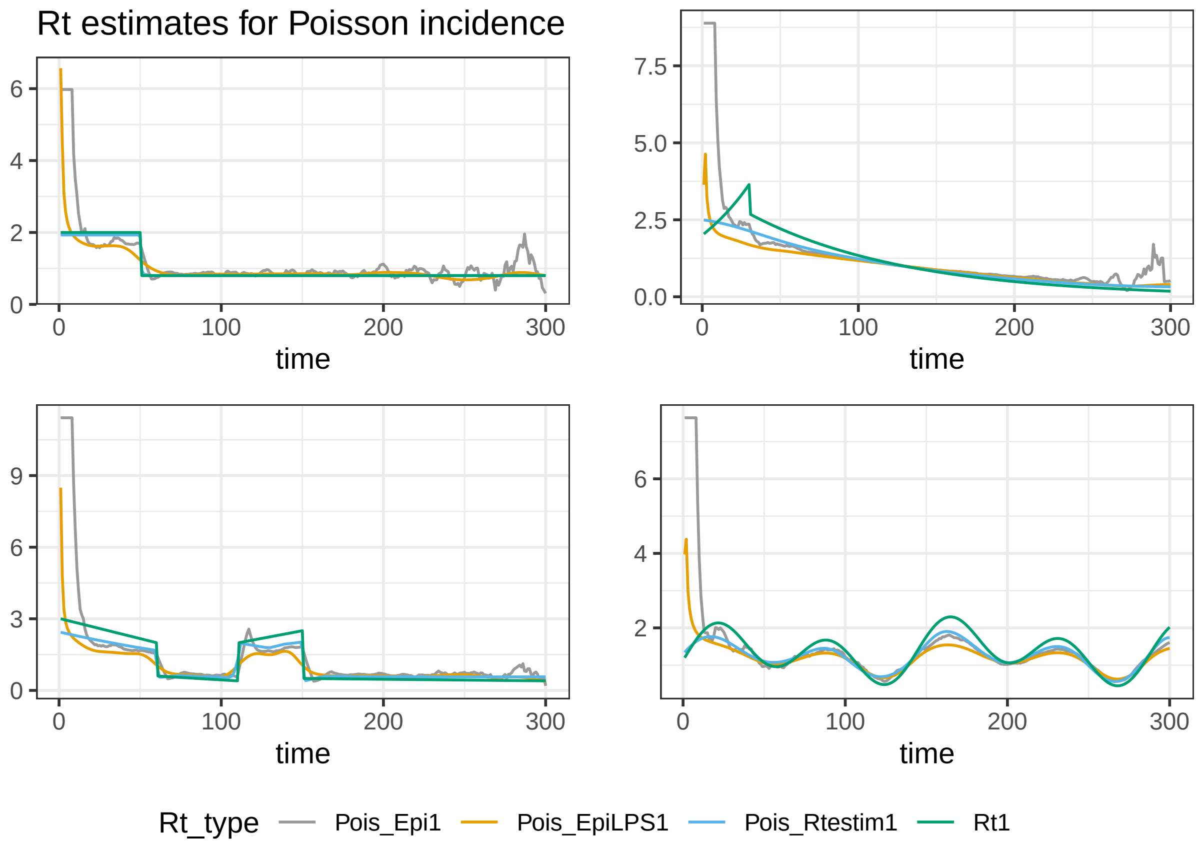


A combined graphical display of both algorithms.

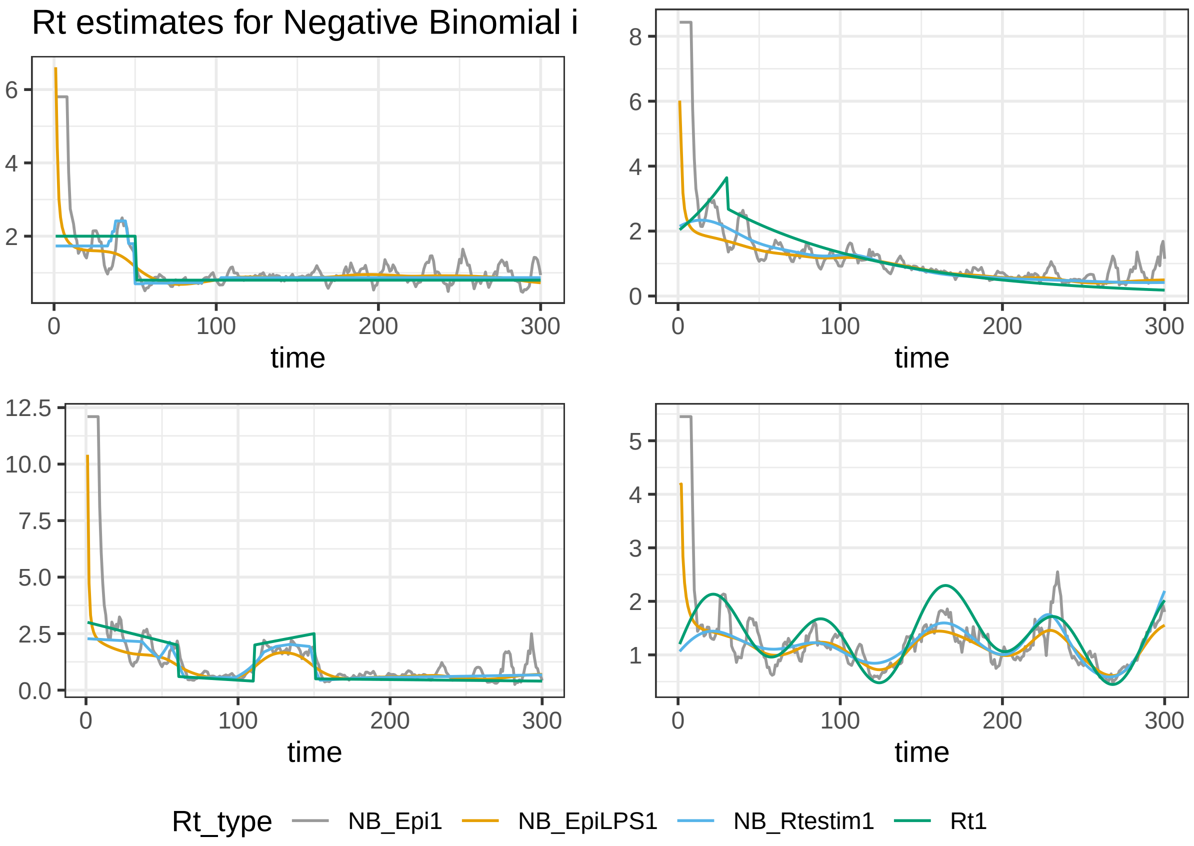
```
res_dat <- data.table(time = 1:len, Rt1 = Rt1, Rt2 = Rt2, Rt3 = Rt3, Rt4 = Rt4,
                      Pois_count1 = incidence1, Pois_count2 = incidence2,
                      Pois_count3 = incidence3, Pois_count4 = incidence4,
                      NB_count1 = nb_incidence1, NB_count2 = nb_incidence2,
                      NB_count3 = nb_incidence3, NB_count4 = nb_incidence4,
                      Pois_Epi1 = c(rep(mod_epiEstim1$R$`Mean(R)`[1], 7),
                                    mod_epiEstim1$R$`Mean(R)`),
                      Pois_Epi2 = c(rep(mod_epiEstim2$R$`Mean(R)`[1], 7),
                                    mod_epiEstim2$R$`Mean(R)`),
                      Pois_Epi3 = c(rep(mod_epiEstim3$R$`Mean(R)`[1], 7),
                                    mod_epiEstim3$R$`Mean(R)`),
                      Pois_Epi4 = c(rep(mod_epiEstim4$R$`Mean(R)`[1], 7),
                                    mod_epiEstim4$R$`Mean(R)`),
                      Pois_Rtestim1 = rtestim_tuned_mod1,
                      Pois_Rtestim2 = rtestim_tuned_mod2,
                      Pois_Rtestim3 = rtestim_tuned_mod3,
                      Pois_Rtestim4 = rtestim_tuned_mod4,
                      Pois_EpiLPS1 = mod_epilst_pois1$RLPS$R,
                      Pois_EpiLPS2 = mod_epilst_pois2$RLPS$R,
                      Pois_EpiLPS3 = mod_epilst_pois3$RLPS$R,
                      Pois_EpiLPS4 = mod_epilst_pois4$RLPS$R,

                      NB_Epi1 = c(rep(mod_epi_nb1$R$`Mean(R)`[1], 7),
                                  mod_epi_nb1$R$`Mean(R)`),
                      NB_Epi2 = c(rep(mod_epi_nb2$R$`Mean(R)`[1], 7),
                                  mod_epi_nb2$R$`Mean(R)`),
                      NB_Epi3 = c(rep(mod_epi_nb3$R$`Mean(R)`[1], 7),
                                  mod_epi_nb3$R$`Mean(R)`),
                      NB_Epi4 = c(rep(mod_epi_nb4$R$`Mean(R)`[1], 7),
                                  mod_epi_nb4$R$`Mean(R)`),
                      NB_Rtestim1 = rtestim_tuned_mod_nb1,
                      NB_Rtestim2 = rtestim_tuned_mod_nb2,
                      NB_Rtestim3 = rtestim_tuned_mod_nb3,
                      NB_Rtestim4 = rtestim_tuned_mod_nb4,
                      NB_EpiLPS1 = mod_epilst_nb1$RLPS$R,
                      NB_EpiLPS2 = mod_epilst_nb2$RLPS$R,
                      NB_EpiLPS3 = mod_epilst_nb3$RLPS$R,
                      NB_EpiLPS4 = mod_epilst_nb4$RLPS$R
)
```

```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73",
               "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
fig1 <- res_dat %>%
  select(time, Rt1, Pois_Epi1, Pois_Rtestim1, Pois_EpiLPS1) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(title="Rt estimates for Poisson incidence", y="") +
  theme(color = "methods") +
  theme_bw()

fig2 <- res_dat %>%
```

```r
  select(time, Rt2, Pois_Epi2, Pois_Rtestim2, Pois_EpiLPS2) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="") +
  theme_bw()

fig3 <- res_dat %>%
  select(time, Rt3, Pois_Epi3, Pois_Rtestim3, Pois_EpiLPS3) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="") +
  theme_bw()

fig4 <- res_dat %>%
  select(time, Rt4, Pois_Epi4, Pois_Rtestim4, Pois_EpiLPS4) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="") +
  theme_bw()

library(ggpubr)
ggpubr::ggarrange(fig1, fig2, fig3, fig4, ncol=2, nrow=2,
          common.legend = TRUE, legend = "bottom")
```



```
fig21 <- res_dat %>%
  select(time, Rt1, NB_Epi1, NB_Rtestim1, NB_EpiLPS1) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
   labs(title="Rt estimates for Negative Binomial incidence", y="") +
  theme_bw()

fig22 <- res_dat %>%
  select(time, Rt2, NB_Epi2, NB_Rtestim2, NB_EpiLPS2) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="") +
  theme_bw()

fig23 <- res_dat %>%
  select(time, Rt3, NB_Epi3, NB_Rtestim3, NB_EpiLPS3) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="") +
  theme_bw()

fig24 <- res_dat %>%
  select(time, Rt4, NB_Epi4, NB_Rtestim4, NB_EpiLPS4) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="") +
  theme_bw()

#library(cowplot)
#nb_res_plot <- plot_grid(fig21, fig22, fig23, fig24)
ggarrange(fig21, fig22, fig23, fig24, ncol=2, nrow=2,
          common.legend = TRUE, legend = "bottom")
```

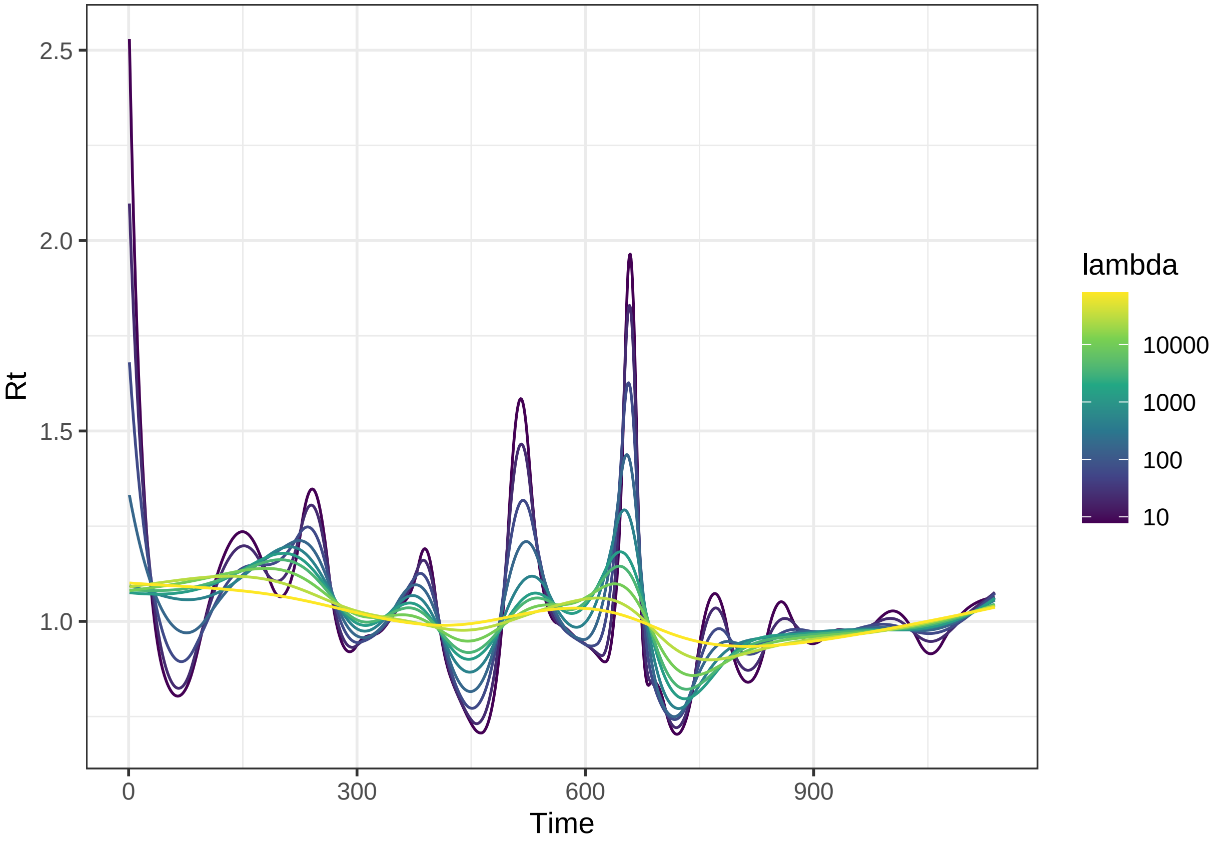


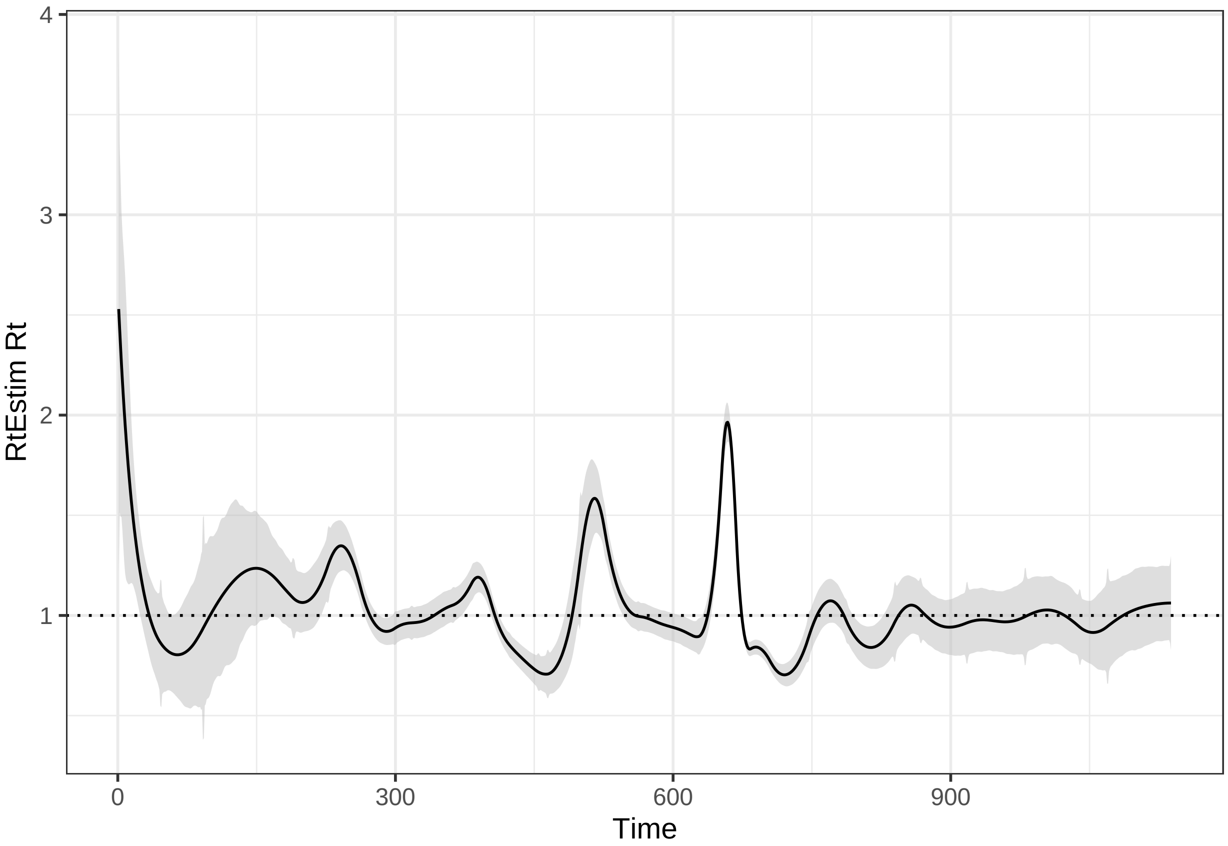
```
fig_res <- ggarrange(fig1, fig21,
          fig2, fig22,
          fig3, fig23,
          fig4, fig24, ncol=2, nrow=4,
          common.legend = TRUE, legend = "bottom")
ggsave(here::here("fig/res-plot.png"), fig_res, width = 9.77, height = 10.39)
```

Real case

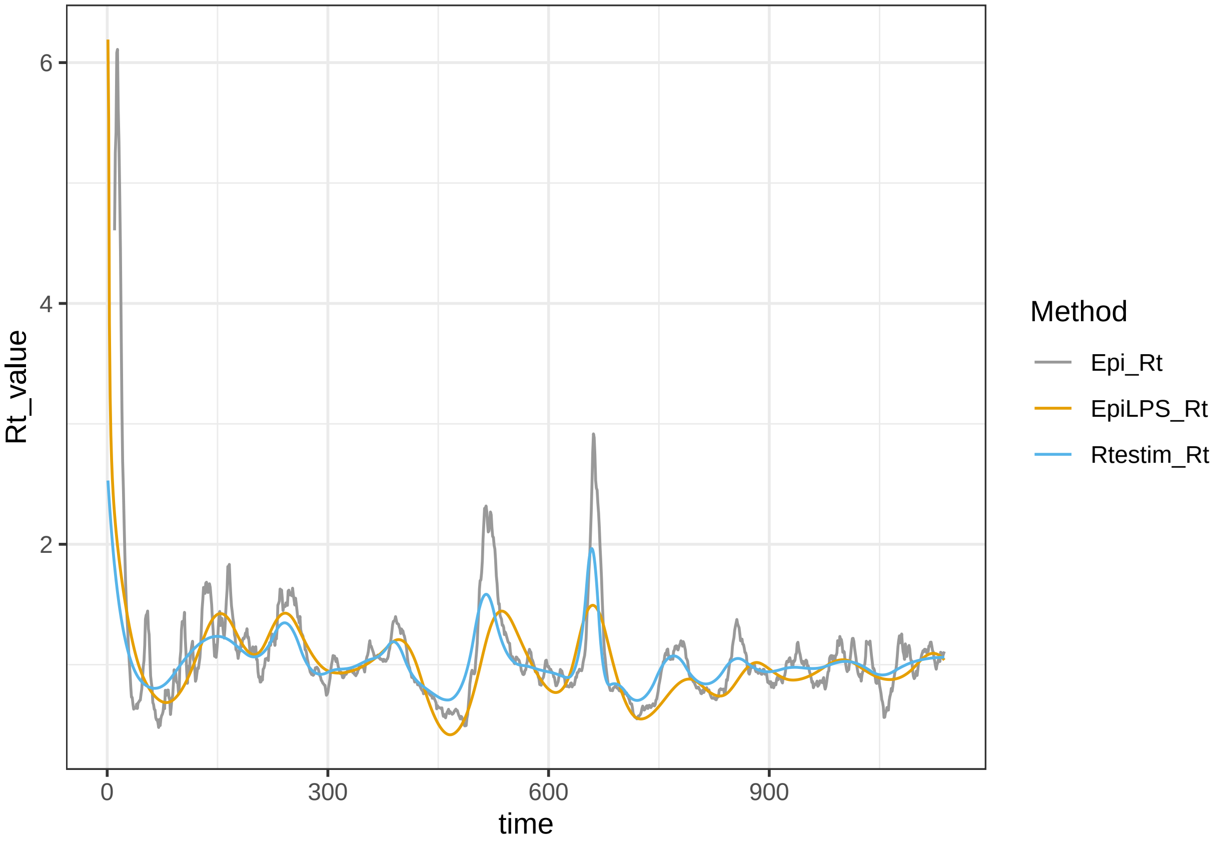
Covid-19 Canada

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```





```
## Warning: Removed 9 rows containing missing values (`geom_line()`).
```

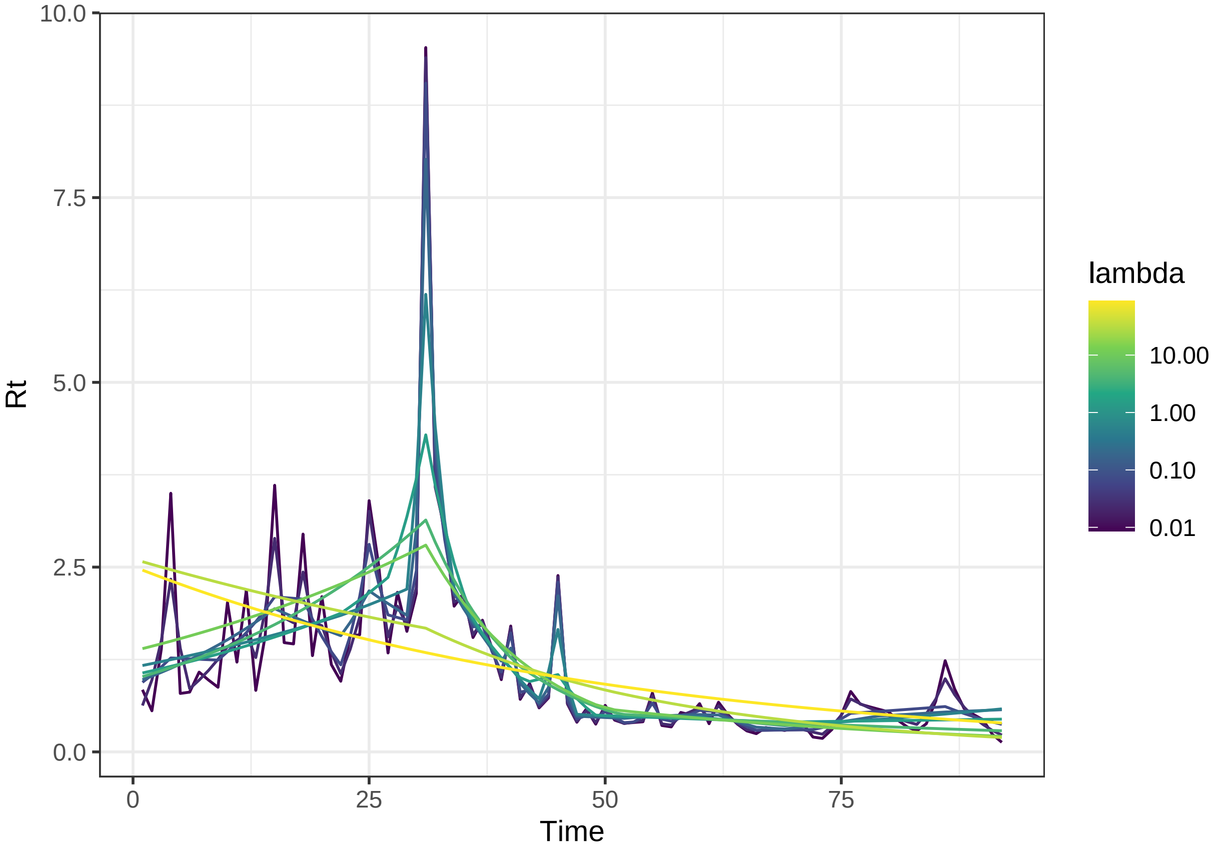


1918 H1N1 influenza in the USA

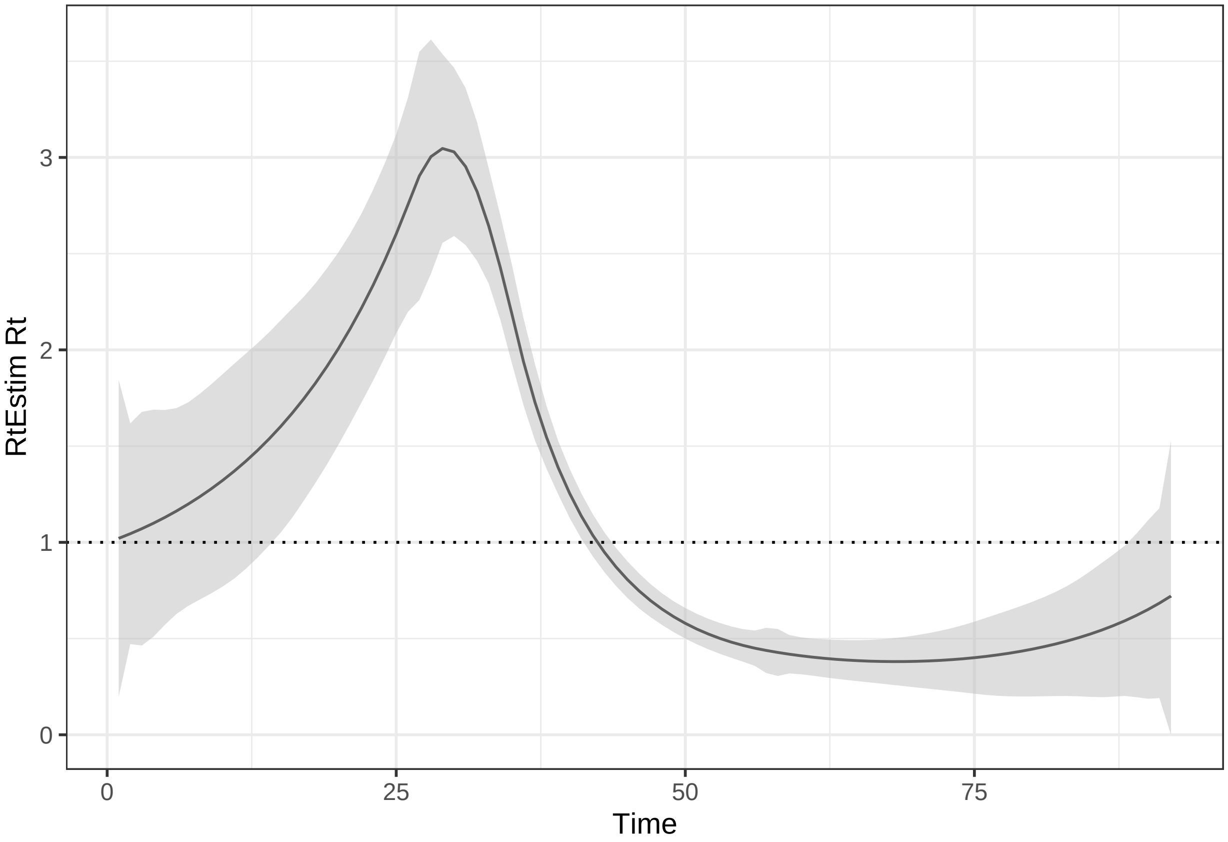
```r
# grab the flu data from EpiEstim package
data("Flu1918")
n <- length(Flu1918$incidence)

mod_epi_flu <- estimate_R(Flu1918$incidence,
          method = "non_parametric_si",
          config = make_config(list(si_distr = Flu1918$si_distr)))
```

```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

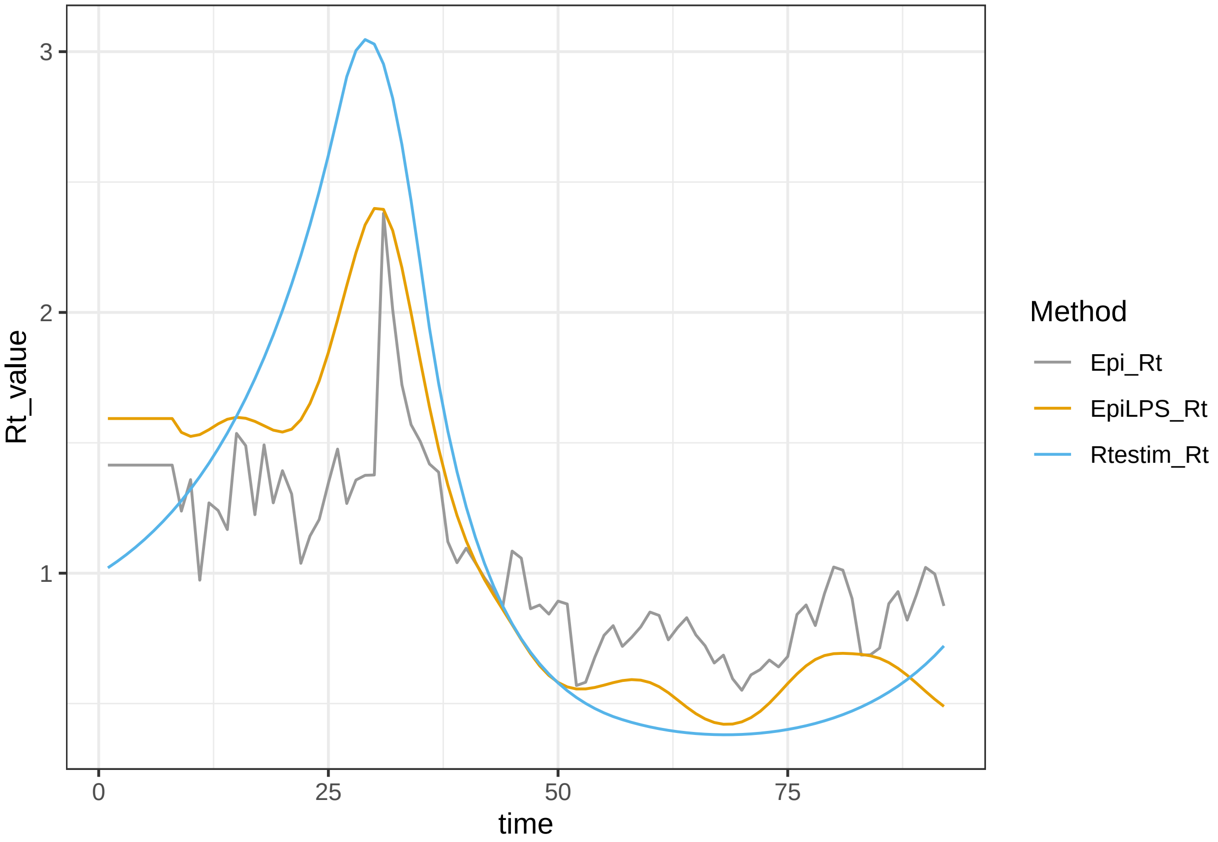
```r
rt_epi_flu <- mod_epi_flu$R$`Mean(R)`

mod_rt <- estimate_rt(Flu1918$incidence, korder = 1, nsol = 10)
plot(mod_rt)
```



```r
cv_mod_flu <- cv_estimate_rt(Flu1918$incidence, korder = 2,
                             nfold = 3, nsol = 50)
mod_rtestim_flu <- cv_mod_flu$full_fit$Rt[,which.min(cv_mod_flu$cv_scores)]

rt_ci_flu <- confband(cv_mod_flu, "lambda.min") # get 95% confidence band
rt_ci_flu %>%
  ggplot(aes(x = 1:n, y = Rt)) +
  geom_line() +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`), fill = "gray", alpha = 0.5) +

  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Time", y = "RtEstim Rt") +
  theme_bw()
```



```
si <- EpiLPS::Idist(mean = 5, sd = 3)$pvec
epiLPSfit_flu <- estimR(incidence = Flu1918$incidence, si = si, CoriR = TRUE)
#plot(epiLPSfit_flu, addfit = "Cori")

flu_res <- data.table(time = 1:length(Flu1918$incidence),
                      Epi_Rt = c(rep(rt_epi_flu[1], 7), rt_epi_flu),
                      Rtestim_Rt = mod_rtestim_flu,
                      EpiLPS_Rt = c(rep(epiLPSfit_flu$RLPS$R[8], 7),
                                    epiLPSfit_flu$RLPS$R[-1:-7])
                      )
flu_res %>%
  pivot_longer(!time, names_to = "Method", values_to = "Rt_value") %>%
  group_by(Method) %>%
  ggplot(aes(x = time, y = Rt_value, col = Method)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  theme_bw()
```



```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73",
               "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

save_example <- function(res_dat, palette = cbPalette){
  len <- dim(res_dat)[1]
  fig1 <- res_dat %>%
    ggplot(aes(y = Rt1, x = time)) +
    geom_line() +
    labs(y = "case 1", title = "Rt", x="") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme_bw()
  fig2 <- res_dat %>%
    ggplot(aes(y = Pois_count1, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(title = "Poisson incidence", y=" ", x="") + #(in log scale)
    theme(plot.title = element_text(hjust = 0.5)) +
    theme_bw()
  fig3 <- res_dat %>%
    ggplot(aes(y = NB_count1, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(title = "Negative Binomial incidence", y=" ", x="") + #(in log scale)
    theme(plot.title = element_text(hjust = 0.5)) +
    theme_bw()

  fig21 <- res_dat %>%
    ggplot(aes(y = Rt2, x = time)) +
    geom_line() +
    labs(y = "case 2", x="") +
    theme_bw()
  fig22 <- res_dat %>%
    ggplot(aes(y = Pois_count2, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(y=" ", x="") +  #(in log scale)
    theme_bw()
  fig23 <- res_dat %>%
    ggplot(aes(y = NB_count2, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(y=" ", x="") +  #(in log scale)
    theme_bw()

  fig31 <- res_dat %>%
    ggplot(aes(y = Rt3, x = time)) +
    geom_line() +
    labs(y = "case 3", x="") +
    theme_bw()
  fig32 <- res_dat %>%
    ggplot(aes(y = Pois_count3, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(y =" ", x="") +  #(in log scale)
    theme_bw()
  fig33 <- res_dat %>%
    ggplot(aes(y = NB_count3, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(y =" ", x="") +  #(in log scale)
    theme_bw()

  fig41 <- res_dat %>%
    ggplot(aes(y = Rt4, x = time)) +
    geom_line() +
    labs(y = "case 4") +
    theme_bw()
  fig42 <- res_dat %>%
    ggplot(aes(y = Pois_count4, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(y=" ") +  #(in log scale)
    theme_bw()
  fig43 <- res_dat %>%
    ggplot(aes(y = NB_count4, x = time)) +
    geom_line() +
    #scale_y_log10() + # axis of incidence in log scale
    labs(y=" ") +  #(in log scale)
    theme_bw()
  figfull <- ggpubr::ggarrange(fig1, fig2, fig3,
            fig21, fig22, fig23,
            fig31, fig32, fig33,
            fig41, fig42, fig43,
            ncol=3, nrow=4,
          common.legend = TRUE, legend = "bottom")
  ggplot2::ggsave(here::here("fig/plot_samples.png"), figfull,
                  width = 9.04, height = 10.47)
}

head(res_dat)
```

```
##    time Rt1      Rt2      Rt3      Rt4 Pois_count1 Pois_count2 Pois_count3
## 1:    1   2 2.040403 3.000000 1.200000           3           3           3
## 2:    2   2 2.081622 2.983051 1.271725           7           3          10
## 3:    3   2 2.123673 2.966102 1.343015           9           4          13
## 4:    4   2 2.166574 2.949153 1.413436           9           5          15
## 5:    5   2 2.210342 2.932203 1.482563          10          11          18
## 6:    6   2 2.254994 2.915254 1.549978          11           7          22
##    Pois_count4 NB_count1 NB_count2 NB_count3 NB_count4 Pois_Epi1 Pois_Epi2
## 1:           3         4         4         4         4  5.976307  8.884755
## 2:           5        10        10        14         7  5.976307  8.884755
## 3:           6        12        12        15         8  5.976307  8.884755
## 4:           5         4         4        21         2  5.976307  8.884755
## 5:           6         6         6        11         4  5.976307  8.884755
## 6:           5        10        12        14         6  5.976307  8.884755
##    Pois_Epi3 Pois_Epi4 Pois_Rtestim1 Pois_Rtestim2 Pois_Rtestim3 Pois_Rtestim4
## 1:   11.4225  7.638985      1.931343      2.493922      2.435425      1.343692
## 2:   11.4225  7.638985      1.931343      2.486704      2.419948      1.393026
## 3:   11.4225  7.638985      1.931343      2.479056      2.404570      1.439947
## 4:   11.4225  7.638985      1.931343      2.470982      2.389291      1.484230
## 5:   11.4225  7.638985      1.931343      2.462491      2.374112      1.525672
## 6:   11.4225  7.638985      1.931343      2.453589      2.359031      1.564103
##    Pois_EpiLPS1 Pois_EpiLPS2 Pois_EpiLPS3 Pois_EpiLPS4  NB_Epi1  NB_Epi2
## 1:     6.569282     3.636231     8.495863     3.971075 5.803456 8.431216
## 2:     4.506061     4.634144     4.820802     4.382102 5.803456 8.431216
## 3:     3.093234     3.235555     3.392683     3.008339 5.803456 8.431216
## 4:     2.588211     2.743194     2.889053     2.513971 5.803456 8.431216
## 5:     2.323392     2.489772     2.625958     2.253333 5.803456 8.431216
## 6:     2.159021     2.335793     2.460871     2.091076 5.803456 8.431216
##     NB_Epi3  NB_Epi4 NB_Rtestim1 NB_Rtestim2 NB_Rtestim3 NB_Rtestim4 NB_EpiLPS1
## 1: 12.10068 5.450343    1.732839    2.147560    2.275916    1.064130   6.608708
## 2: 12.10068 5.450343    1.732839    2.174457    2.272132    1.098302   4.419586
## 3: 12.10068 5.450343    1.732839    2.199464    2.268353    1.131394   3.017330
## 4: 12.10068 5.450343    1.732839    2.222517    2.264581    1.163279   2.513770
## 5: 12.10068 5.450343    1.732839    2.243558    2.260814    1.193839   2.249000
## 6: 12.10068 5.450343    1.732839    2.262535    2.257052    1.222961   2.084767
##    NB_EpiLPS2 NB_EpiLPS3 NB_EpiLPS4
## 1:   6.018287  10.412442   4.176513
## 2:   4.568950   4.727214   4.193291
## 3:   3.177039   3.317337   2.840407
## 4:   2.684567   2.822726   2.346464
## 5:   2.429845   2.567937   2.082341
## 6:   2.274451   2.411878   1.915893
```

```r
save_example(res_dat)
```