

Supplementary details on experiments

Olivia

2023-10-04

Experimental design

Synthetic data with Poisson incidence

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 R_t , 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 %>%
```

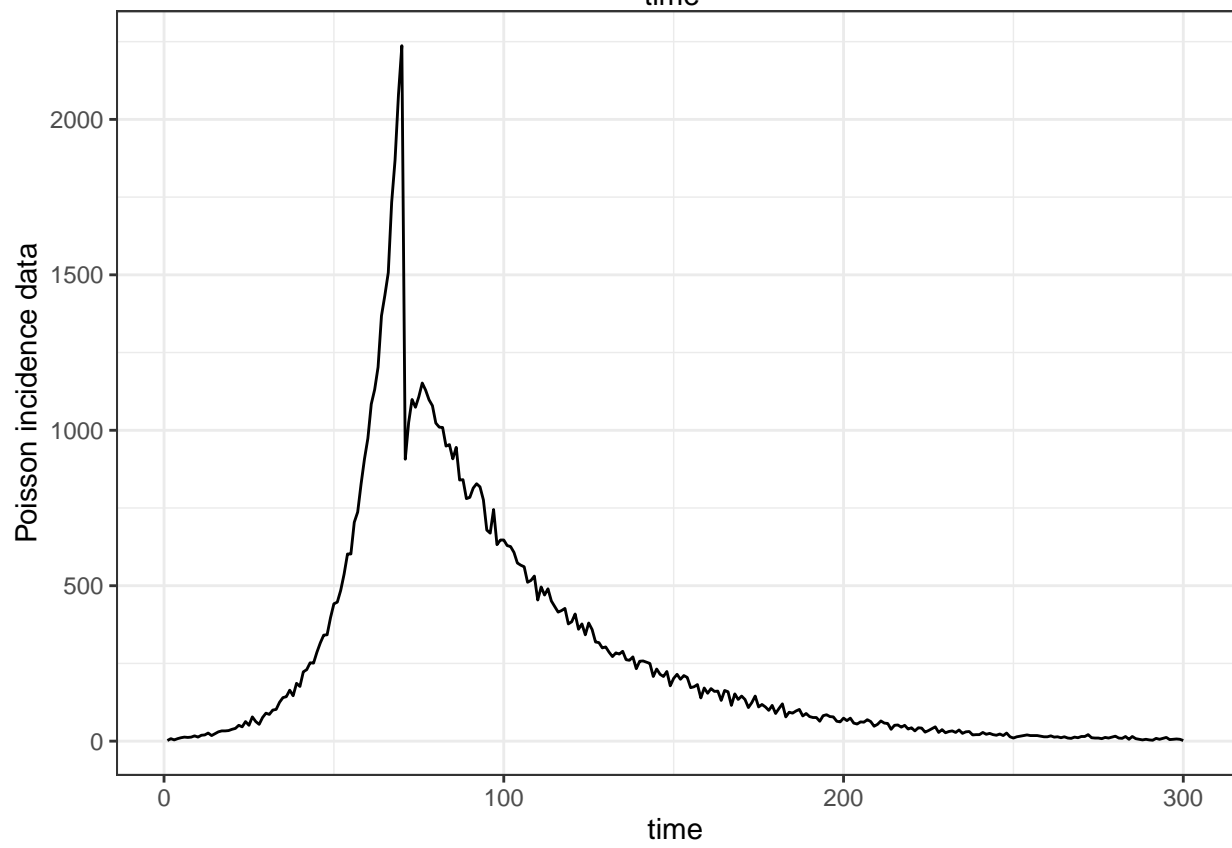
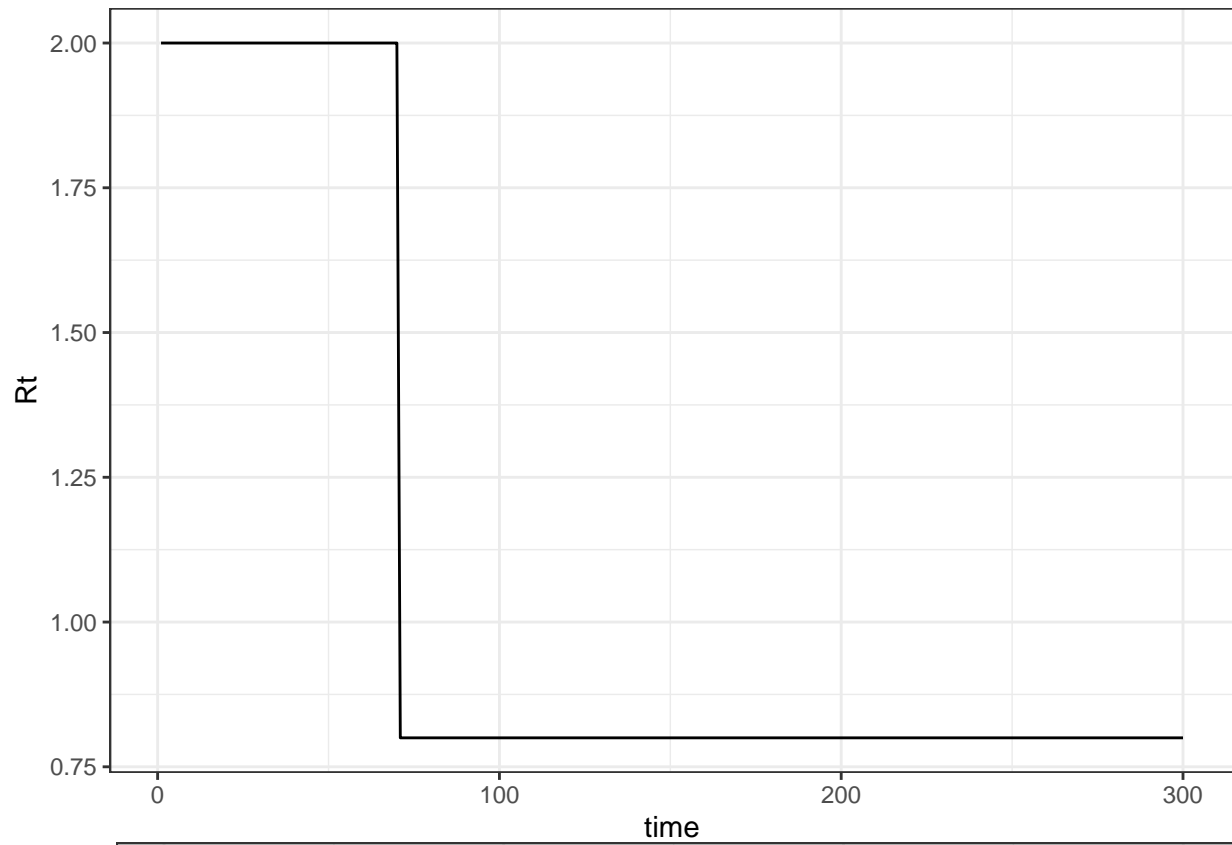
```

    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.")}
}

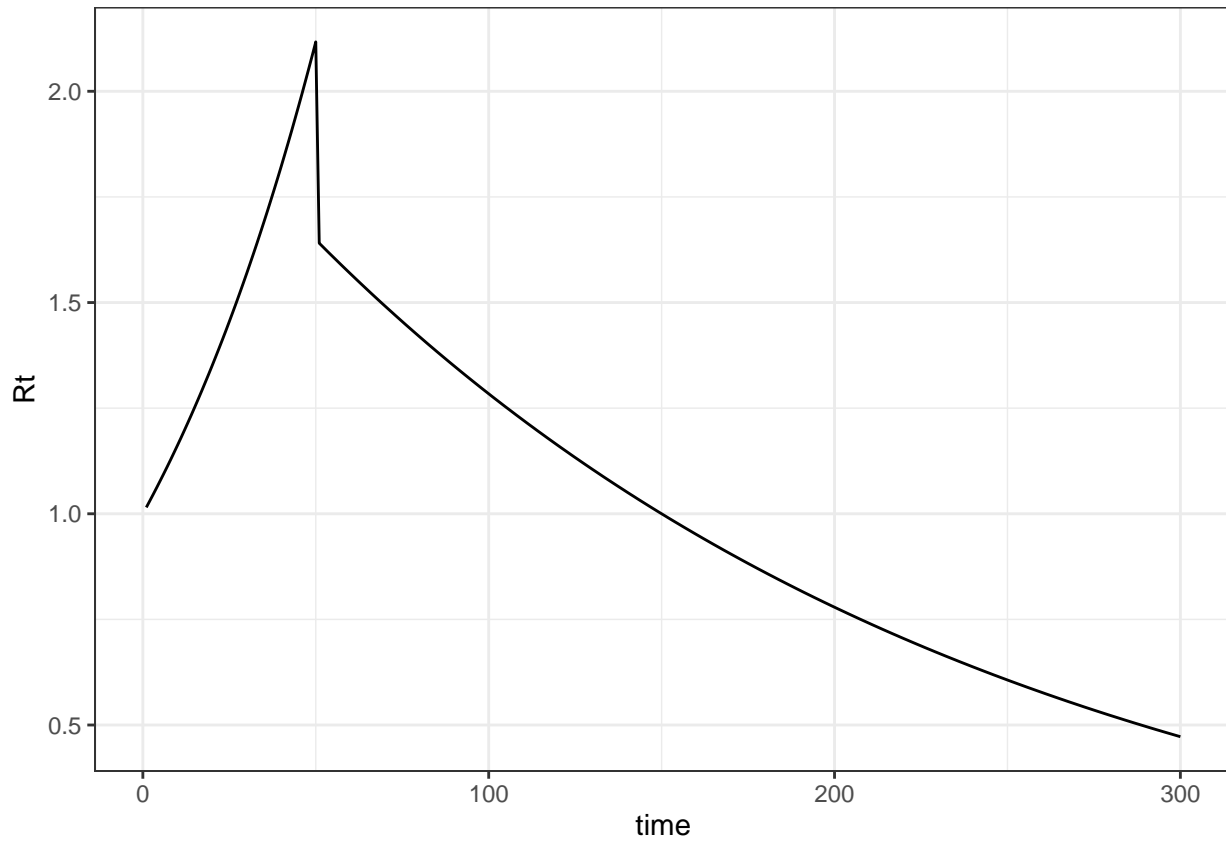
# Scenario 1: two-stage piecewise constant with one dropping point (similar as in EpiFilter)
Rt1 <- c(rep(2, 70), rep(0.8, len-70)) # extend first segment from 50 to 70
# to avoid too many 0s in the tail
gamma_pars1 <- c(3, 3) # serial interval distribution parameters
seed <- 420
set.seed(seed)
incidence1 <- get_pois_incidence(N1, Rt1, gamma_pars1)
display_dat(incidence1, Rt1)

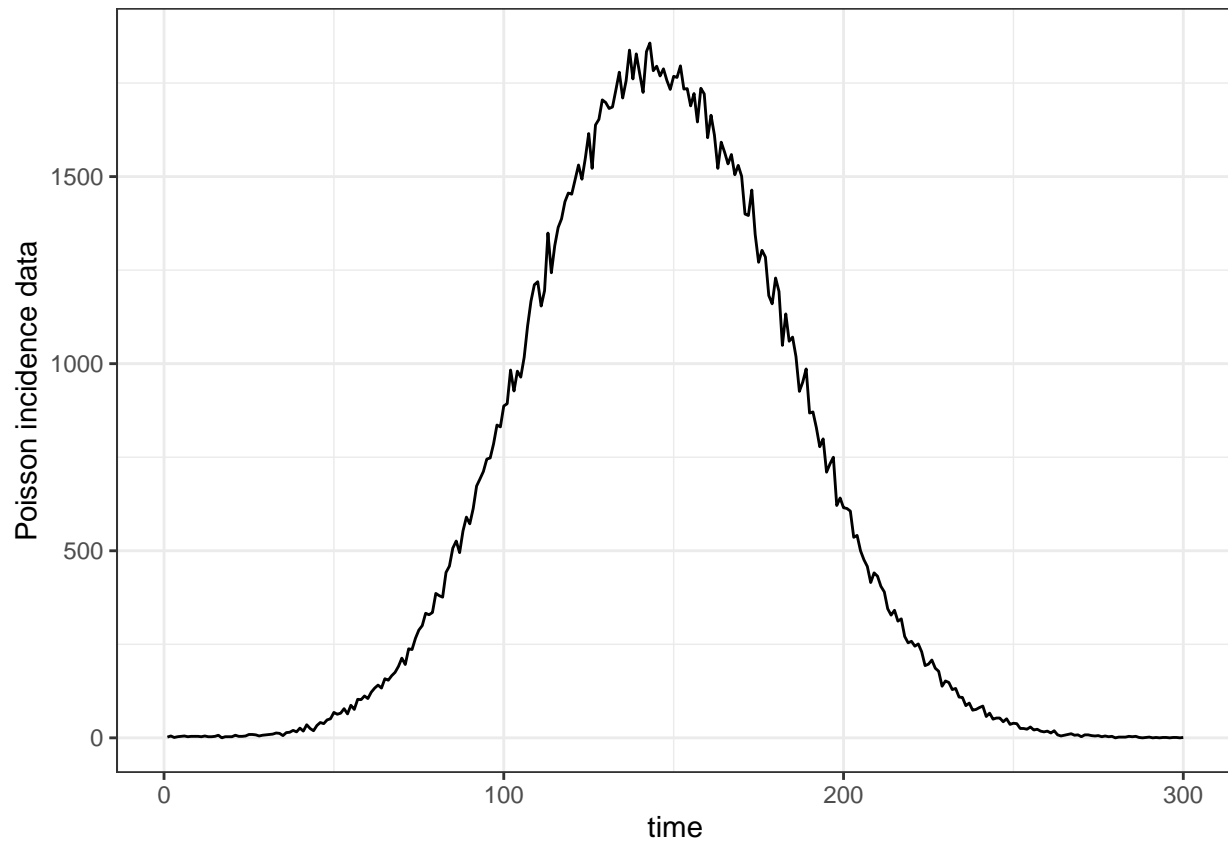
```



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

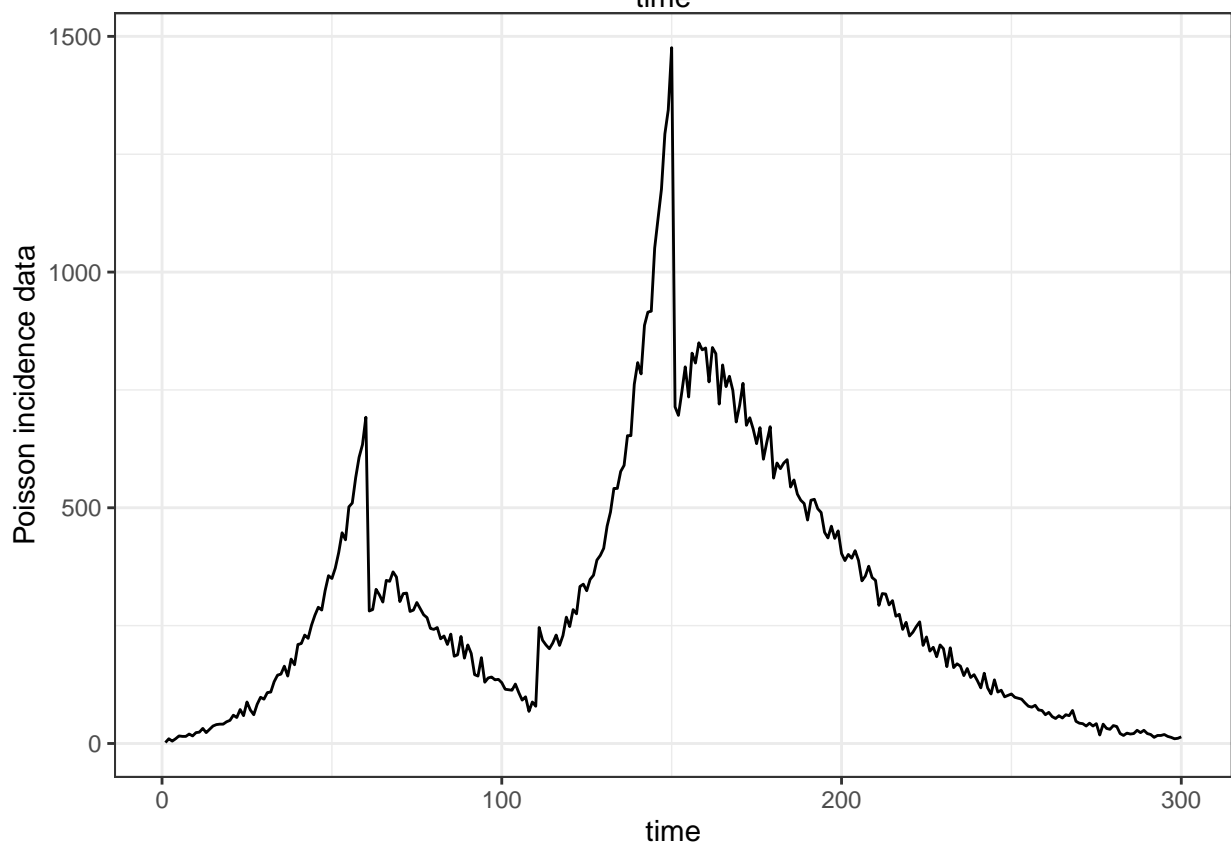
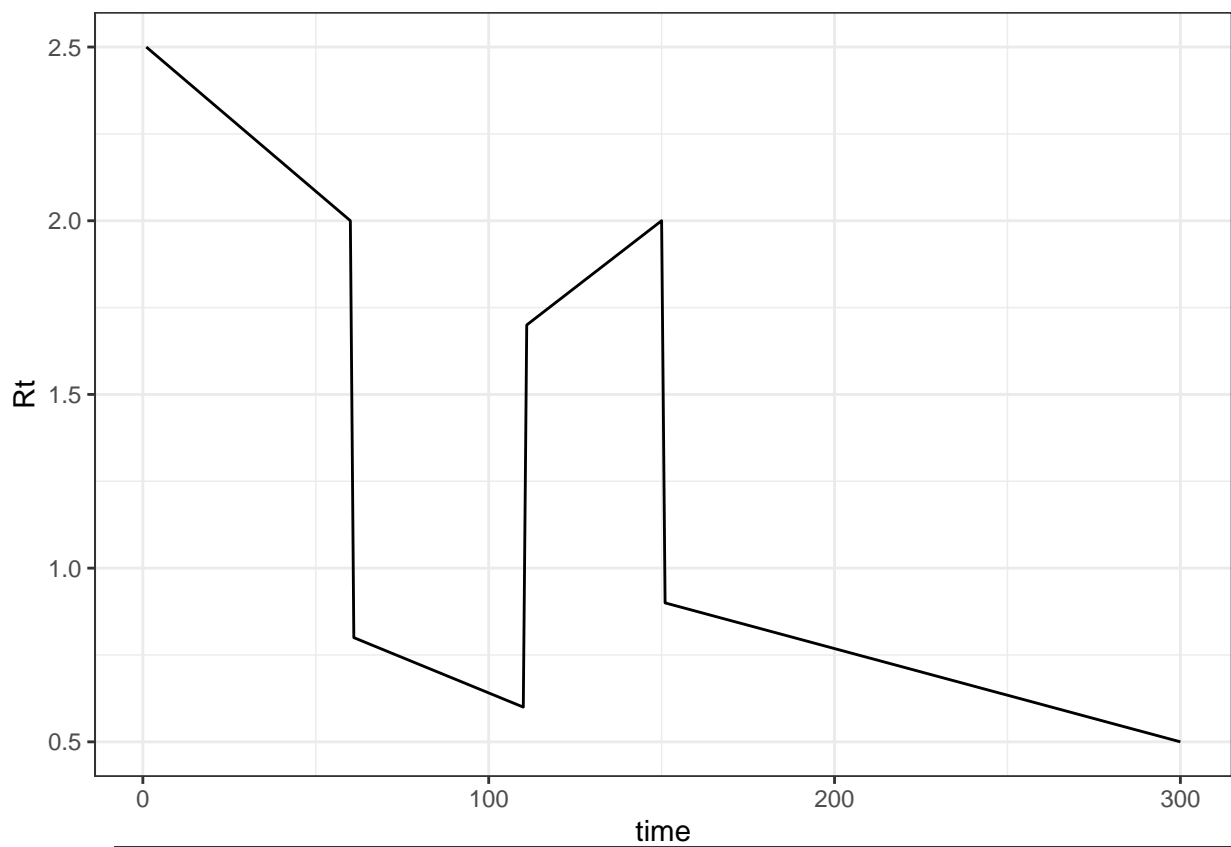
```
# Scenario 2: two-stage exponential growth and decay (similar as in EpiFilter)  
rate <- c(.015, -.005)  
Rt2 <- numeric(len)  
knot <- 50  
Rt2[1:knot] <- exp(rate[1] * (1:knot))  
Rt2[(knot+1):len] <- exp((rate[2]) * ((knot+1):len)) * Rt2[knot]  
gamma_pars2 <- c(2.5, 2.5) # serial interval distribution parameters  
set.seed(seed)  
incidence2 <- get_pois_incidence(N1, Rt2, gamma_pars2)  
display_dat(incidence2, Rt2)
```





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

```
# Scenario 3: multi-stage piecewise constant (similar as in EpiFilter)  
Rt3 <- c(seq(2.5, 2, length.out = 60), seq(0.8, 0.6, length.out = 50),  
         seq(1.7, 2, length.out = 40), seq(0.9, 0.5, 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)
```

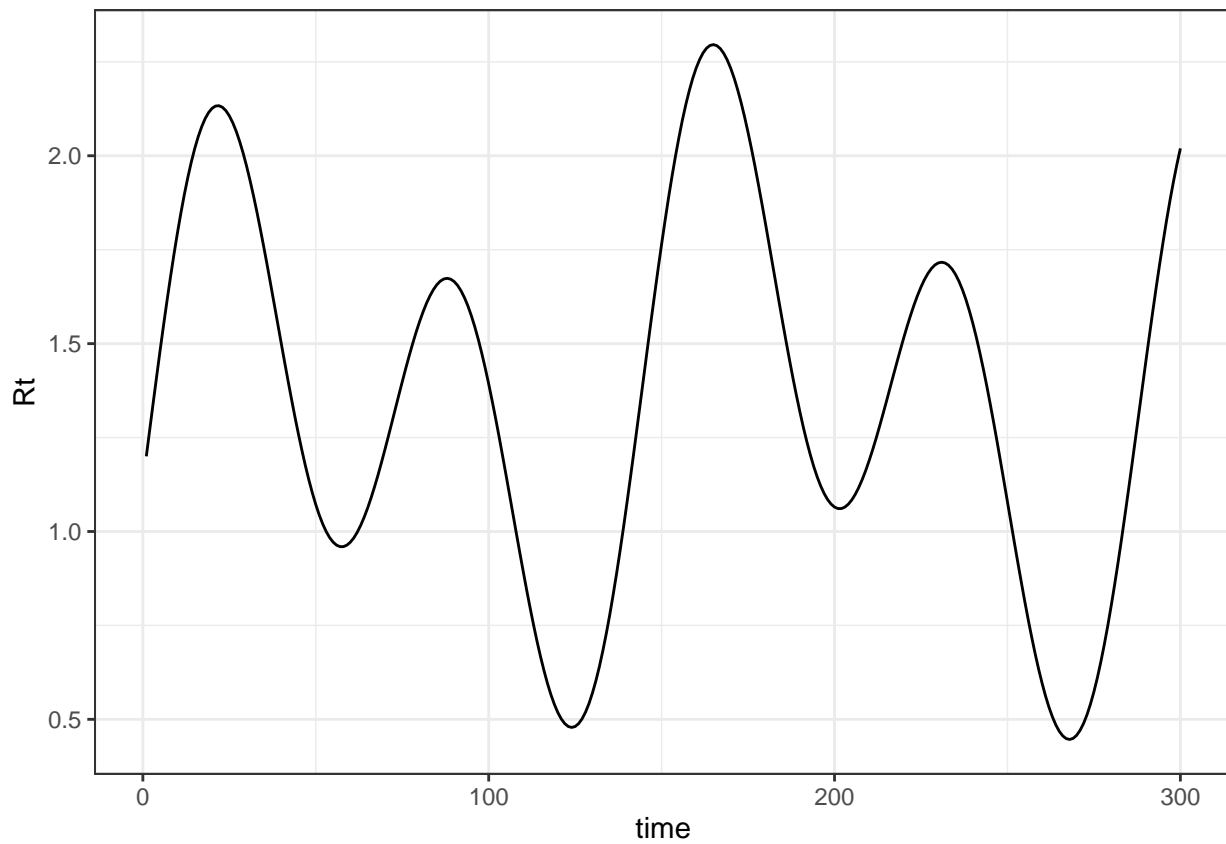


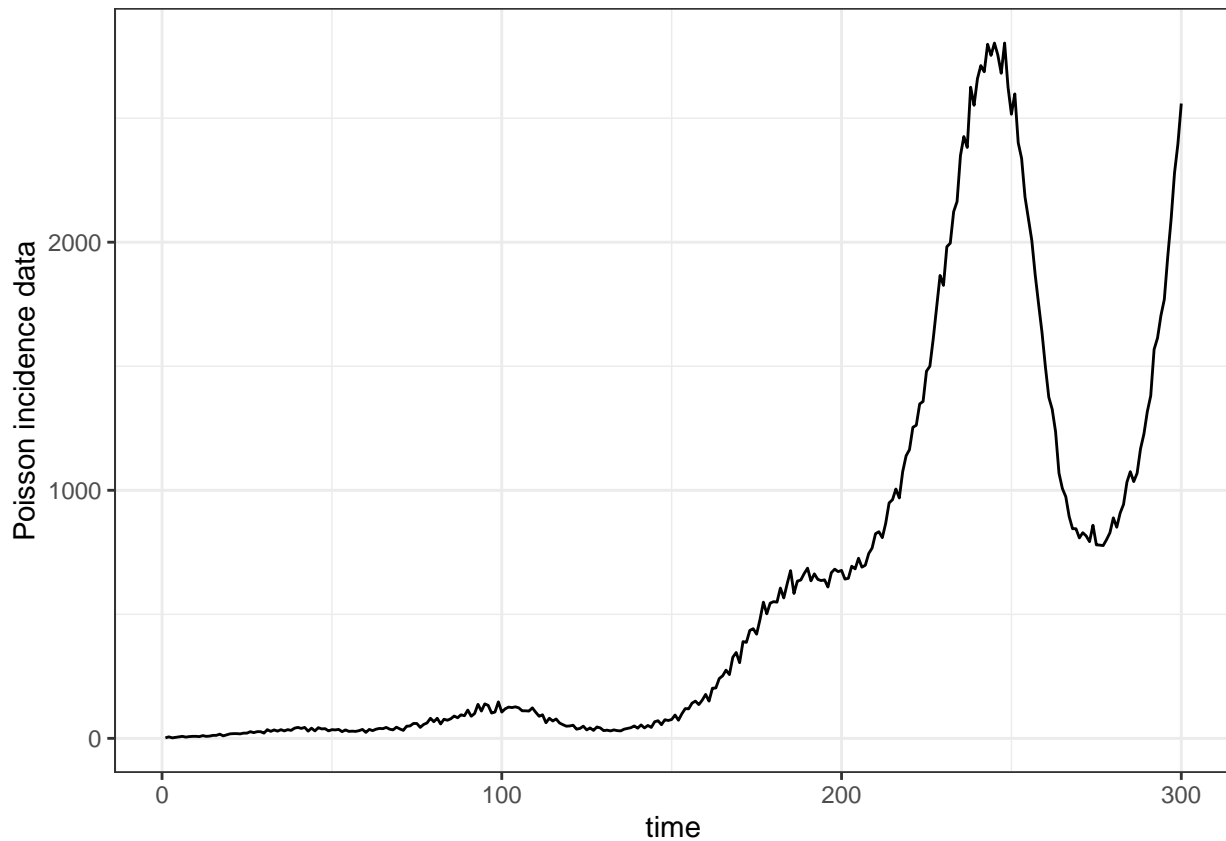
```

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)

```





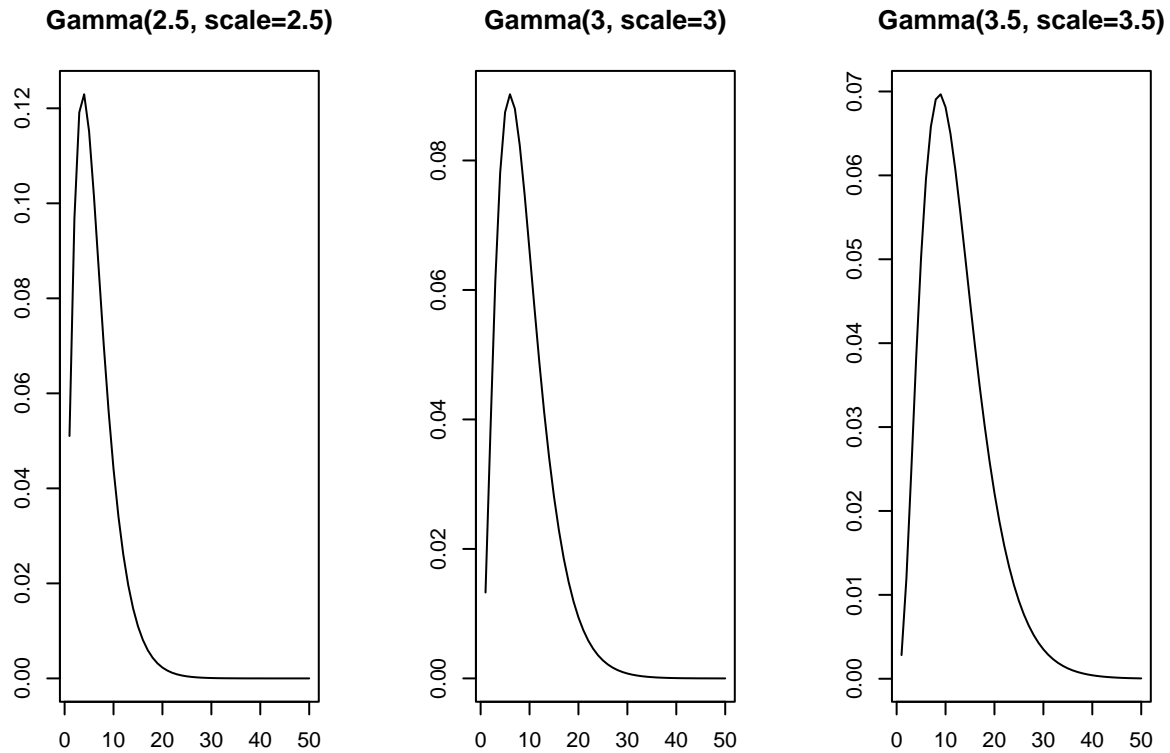
```
check_dat(incidence4, Rt4)
```

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.

```
par(mfrow = c(1, 3))
plot(dgamma(1:50, 2.5, scale = 2.5), type="l", ylab="", xlab="",
     main="Gamma(2.5, scale=2.5)")

plot(dgamma(1:50, 3, scale = 3), type="l", ylab="", xlab="",
     main="Gamma(3, scale=3)")

plot(dgamma(1:50, 3.5, scale = 3.5), type="l", ylab="", xlab="",
     main="Gamma(3.5, scale=3.5)")
```

```
par(mfrow = c(1, 1))
```

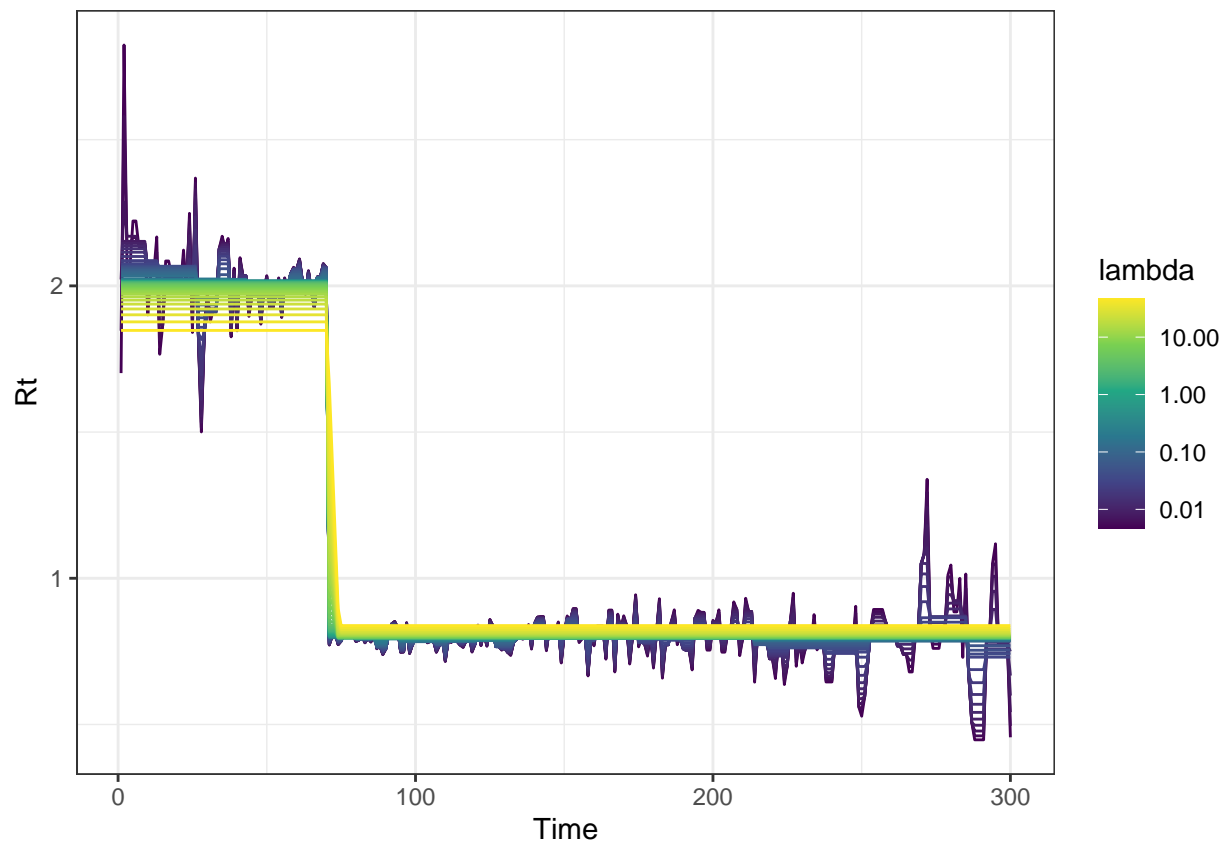
Rt estimates using three methods

Fit our poisson trend filtering using `RtEstim`. See an example of the first scenario. We first see an demonstration of estimates using 50 hyperparameters, and use cross validation to choose the “best” hyperparameter with the lowest score. The last plot of this scenario displays the R_t estimates using the chosen hyperparameter.

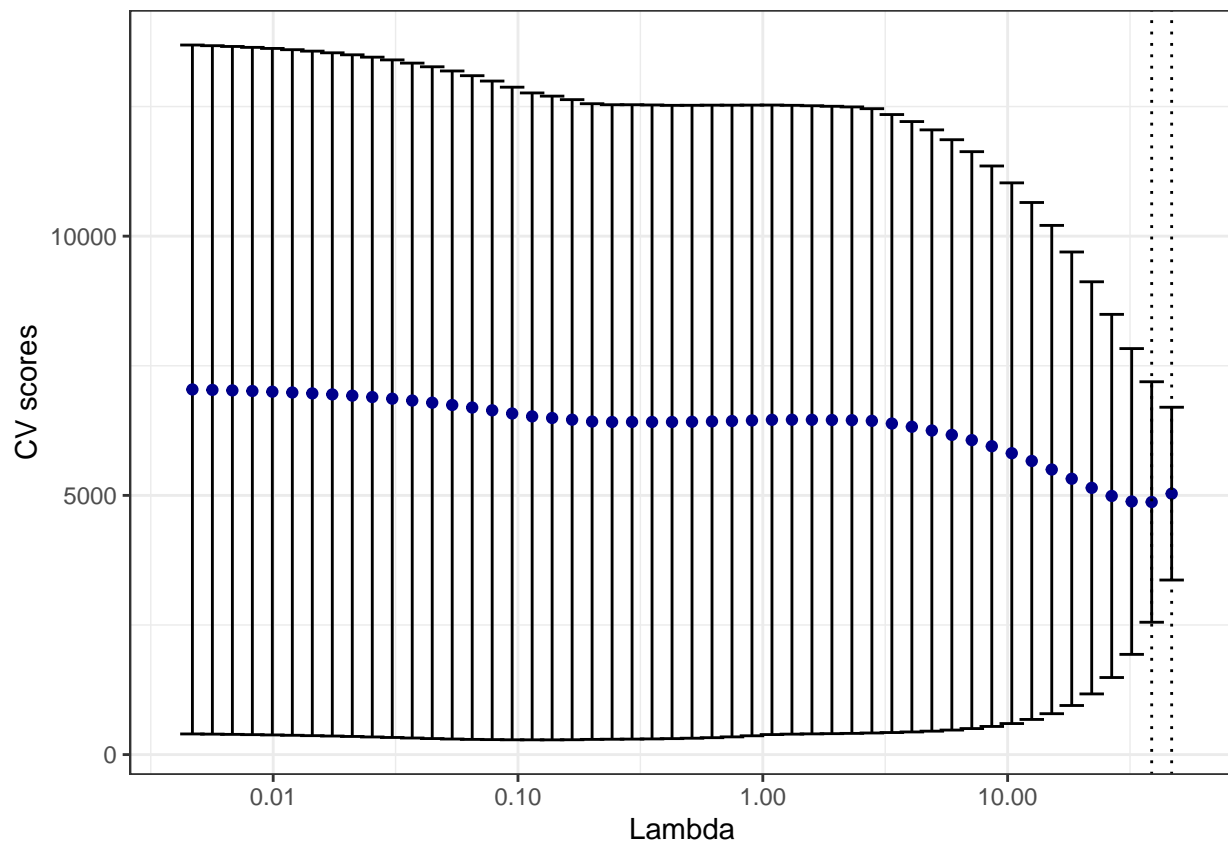
```
rtestim_mod1 <- estimate_rt(incidence1, korder = 0, nsol = 50,
                           dist_gamma = gamma_pars1)
sum(rtestim_mod1$convergence == FALSE) == 0
```

```
## [1] TRUE
```

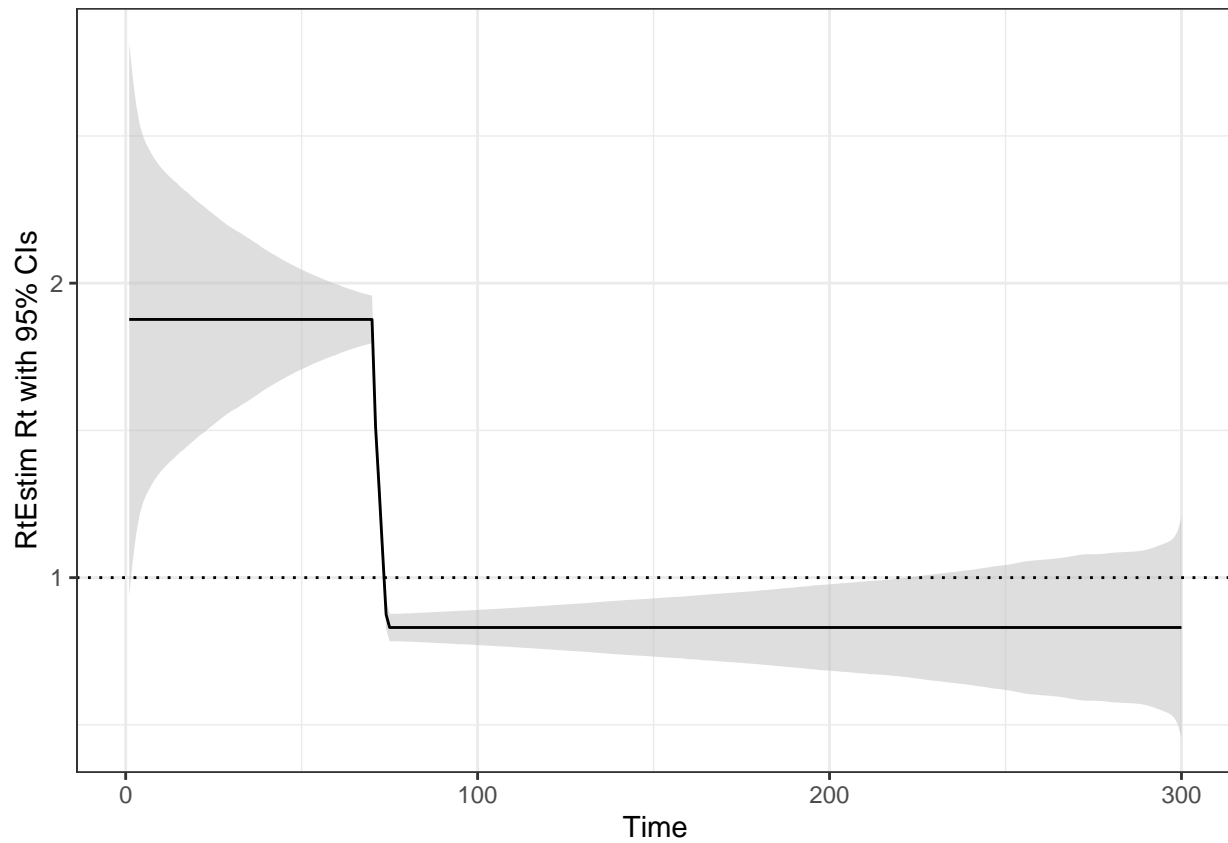
```
plot(rtestim_mod1)
```



```
# hyperparameter tuning using cross validation:
cv_mod1 <- cv_estimate_rt(incidence1, korder = 0, nfold = 3, nsol = 50,
                          dist_gamma = gamma_pars1, maxiter=1e7L)
plot(cv_mod1)
```



```
rtestim_tuned_mod1 <- cv_mod1$full_fit$Rt[, which.min(cv_mod1$cv_scores)]
rt_cil <- confband(cv_mod1, "lambda.min") # get 95% confidence band
rt_cil %>%
  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 with 95% CIs") +
  theme_bw()
```

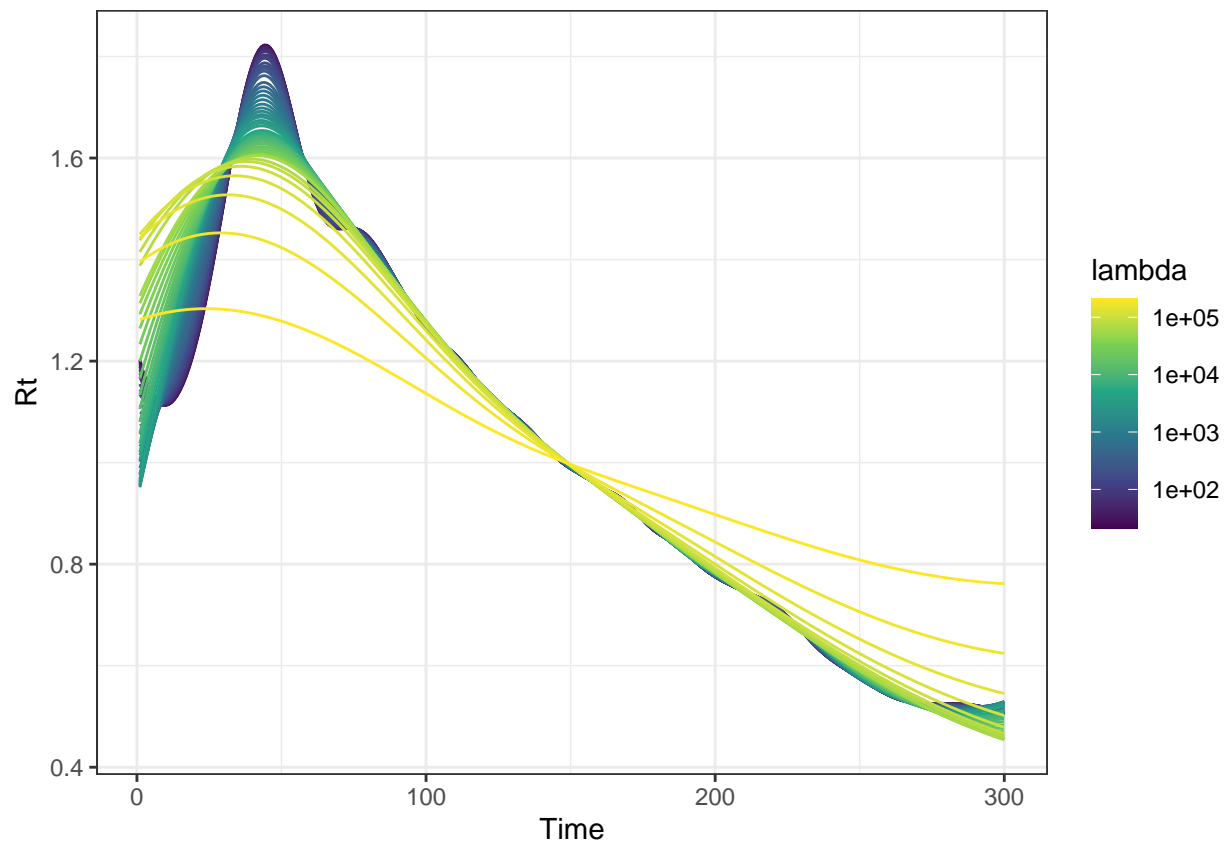


The Rt estimates for the other three scenarios are given in the following figures.

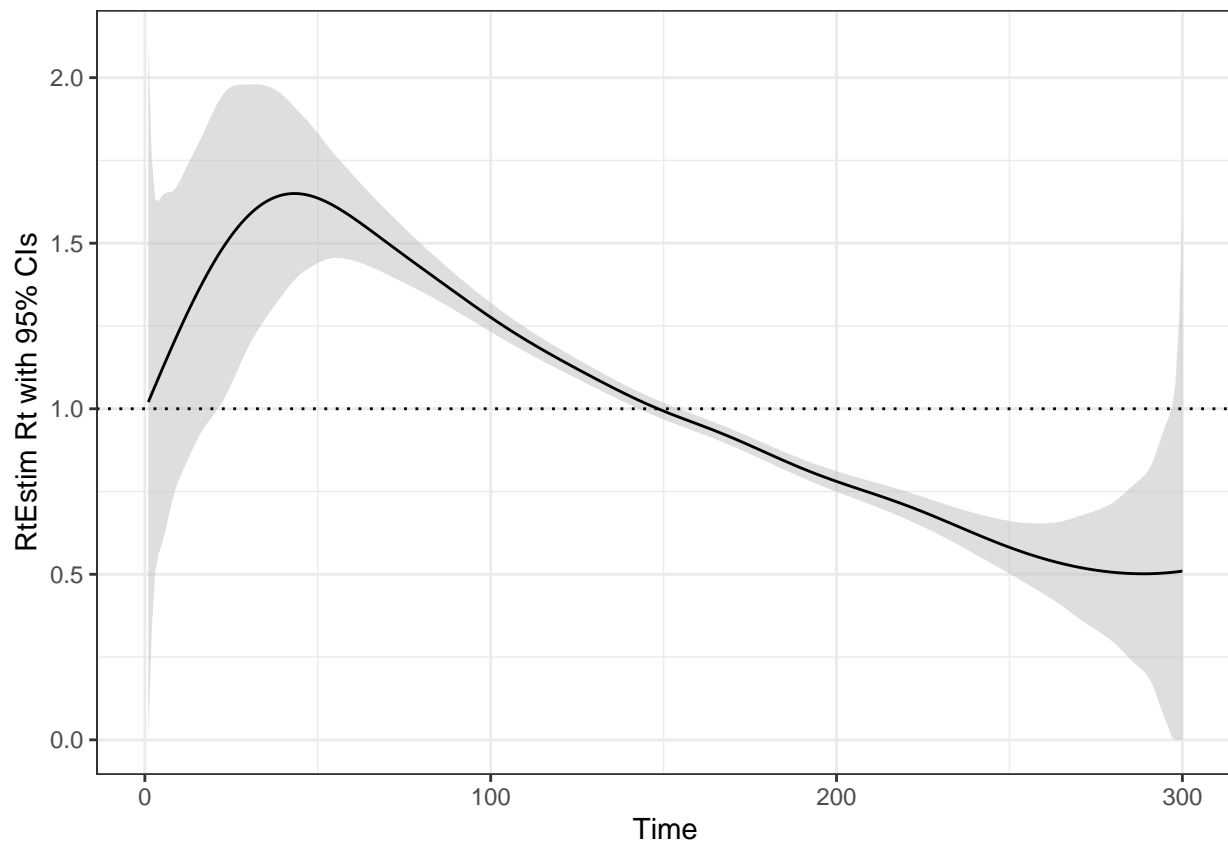
```
rtestim_mod2 <- estimate_rt(incidence2, korder=3, nsol=50, maxiter=1e7L,
                             dist_gamma = gamma_pars2)
sum(rtestim_mod2$convergence == FALSE) == 0
```

```
## [1] TRUE
```

```
plot(rtestim_mod2)
```



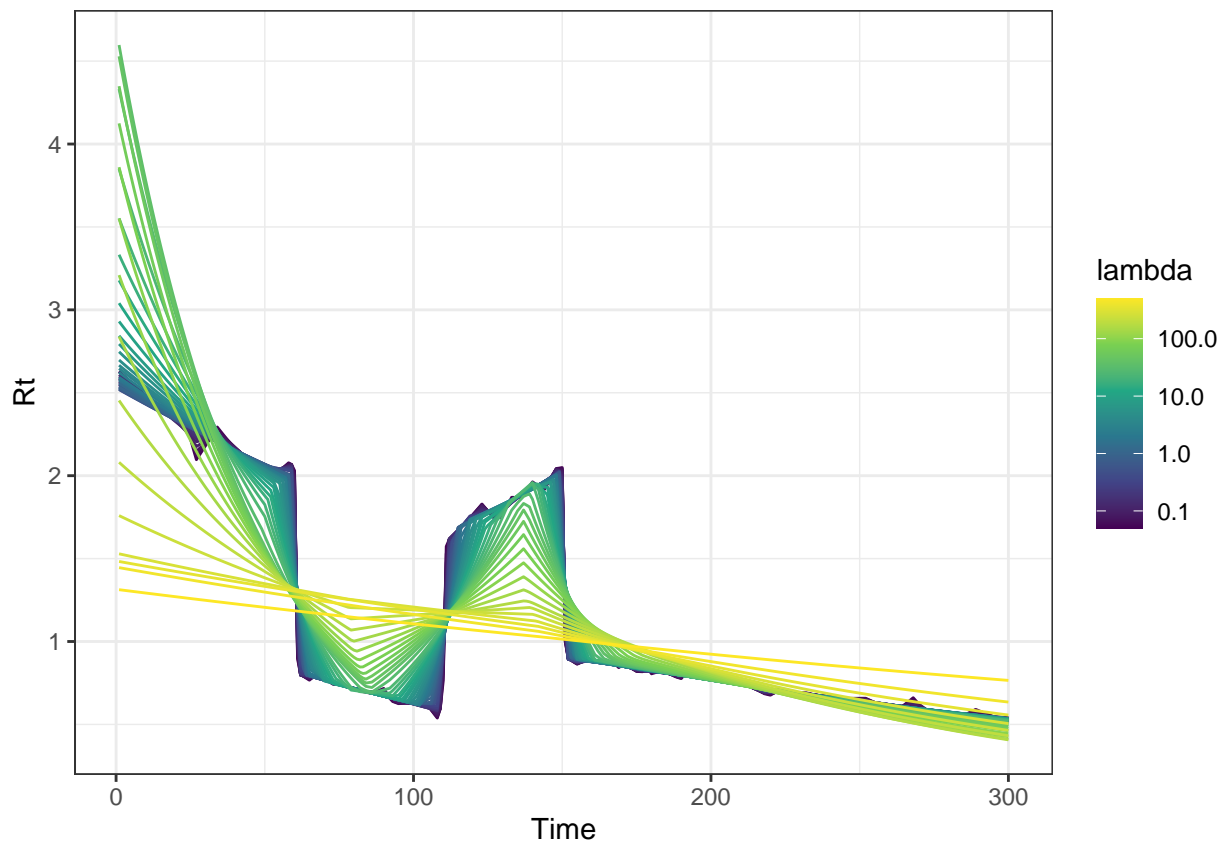
```
cv_mod2 <- cv_estimate_rt(incidence2, korder=3, nfold=3, nsol=50, maxiter=1e7L,
                          dist_gamma = gamma_pars2)
rtestim_tuned_mod2 <- cv_mod2$full_fit$Rt[, which.min(cv_mod2$cv_scores)]
rt_ci2 <- confband(cv_mod2, "lambda.min") # get 95% confidence band
rt_ci2 %>%
  ggplot(aes(x = 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 with 95% CIs") +
  theme_bw()
```



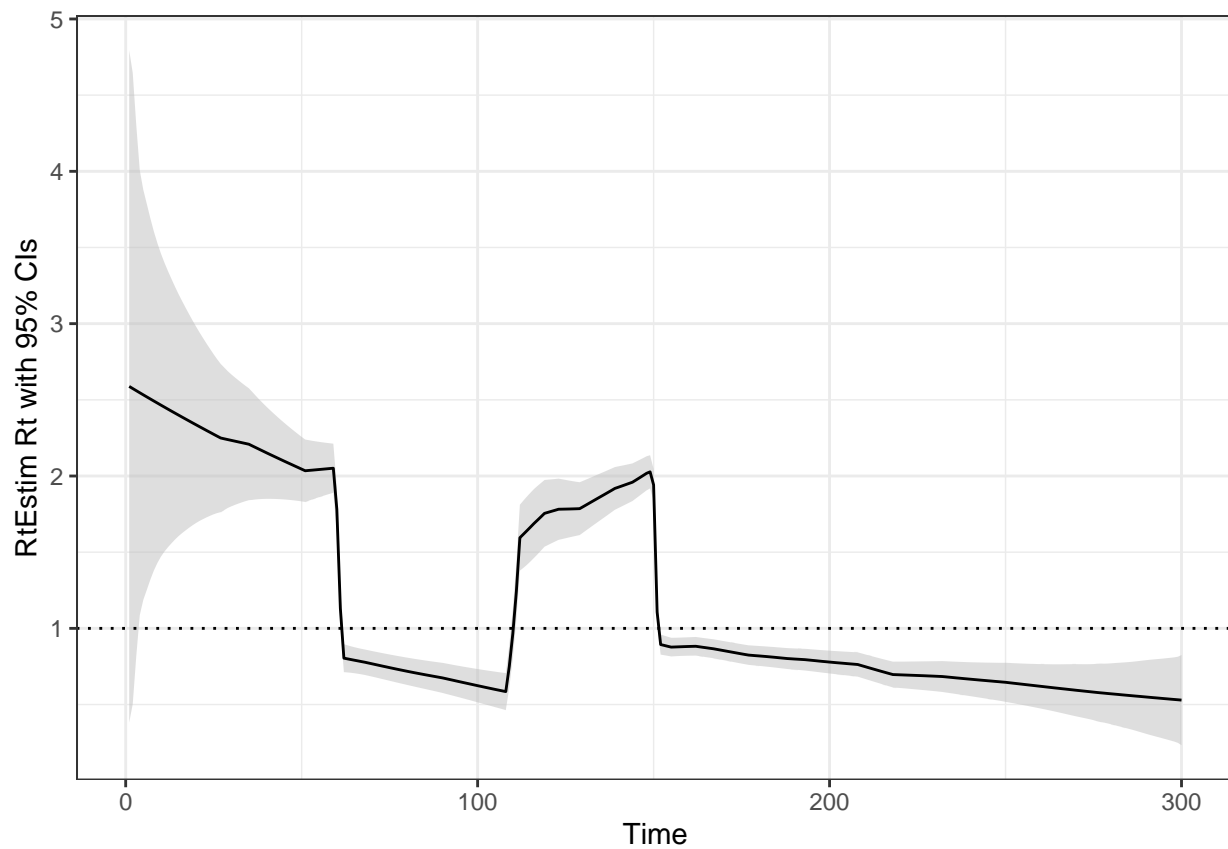
```
rtestim_mod3 <- estimate_rt(incidence3, korder=1, nsol=50, maxiter=1e7L,  
                           dist_gamma = gamma_pars3)  
sum(rtestim_mod3$convergence == FALSE) == 0
```

```
## [1] TRUE
```

```
plot(rtestim_mod3)
```



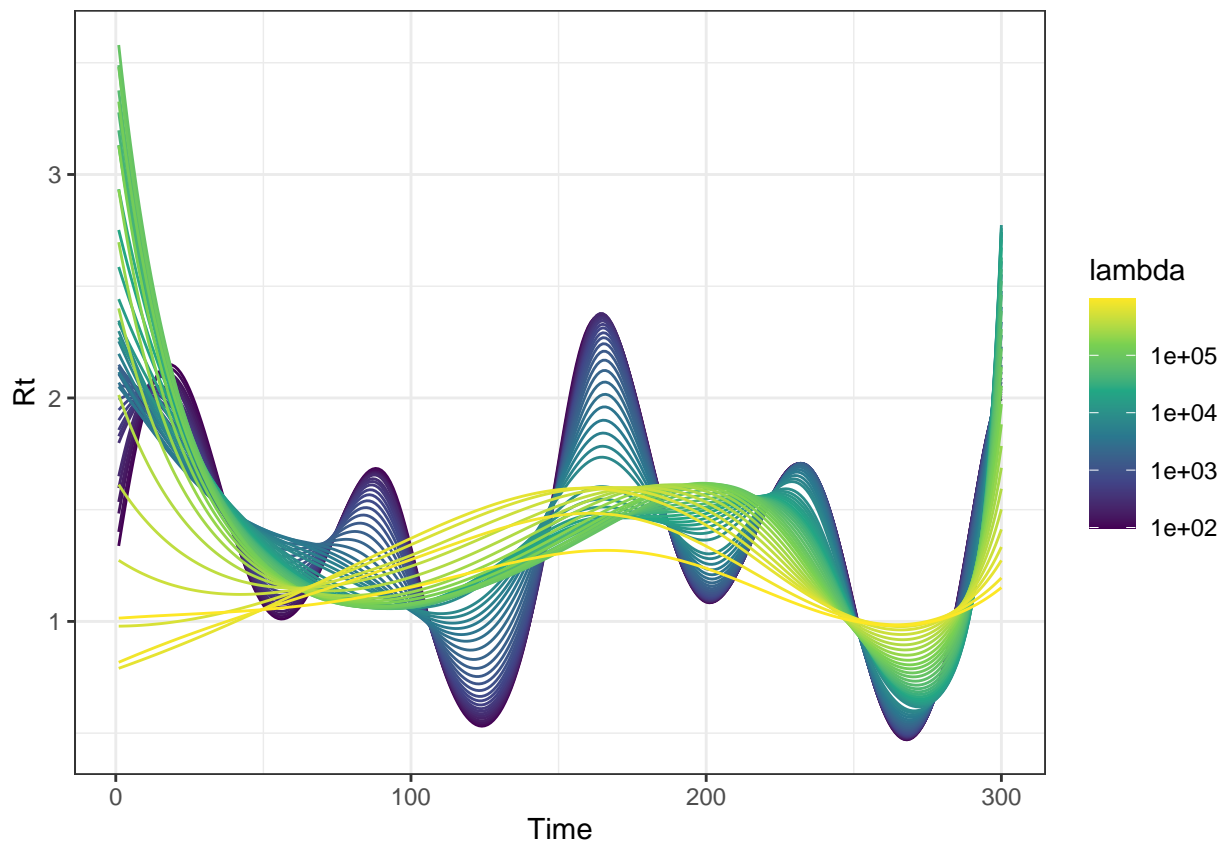
```
cv_mod3 <- cv_estimate_rt(incidence3, korder=1, nfold=3, nsol=50, maxiter=1e7L,
                          dist_gamma = gamma_pars3)
rtestim_tuned_mod3 <- cv_mod3$full_fit$Rt[, which.min(cv_mod3$cv_scores)]
rt_ci3 <- confband(cv_mod3, "lambda.min") # get 95% confidence band
rt_ci3 %>%
  ggplot(aes(x = 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 with 95% CIs") +
  theme_bw()
```



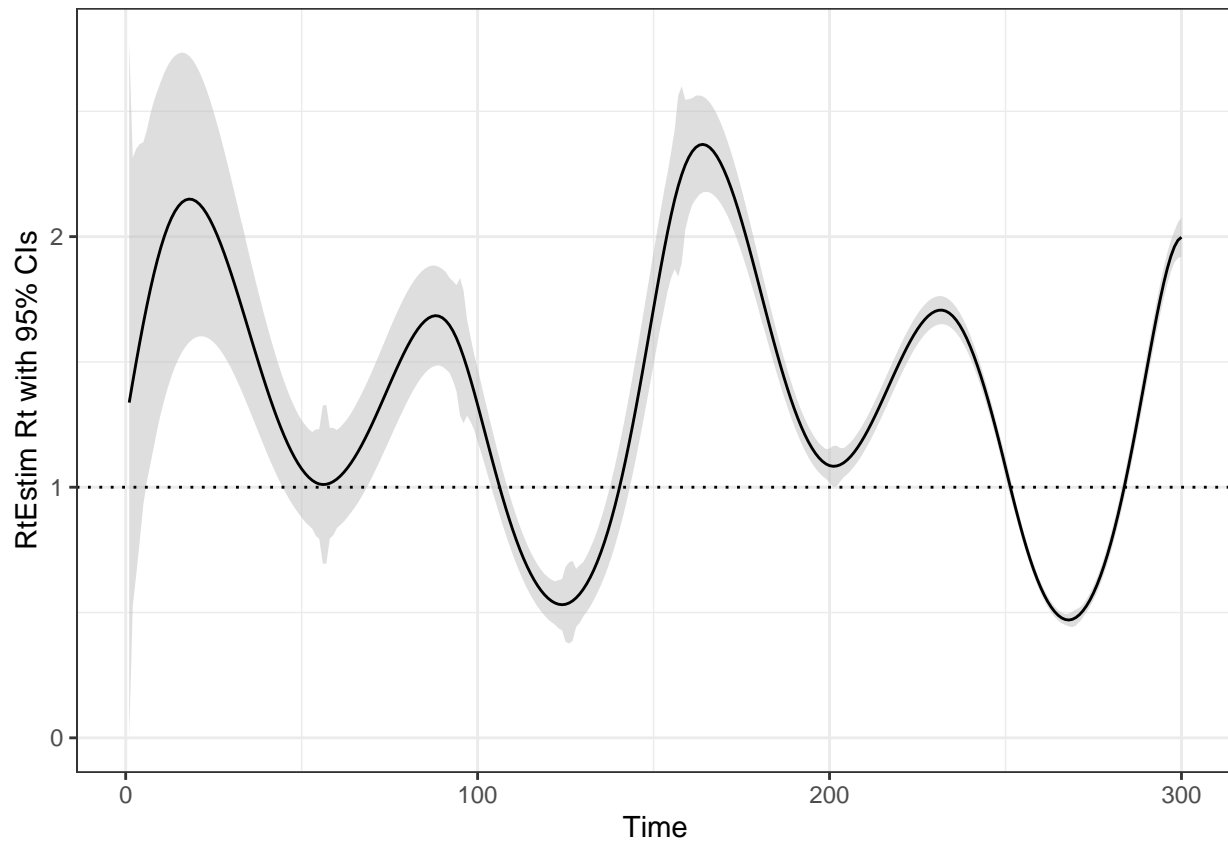
```
rtestim_mod4 <- estimate_rt(incidence4, korder=3, nsol=50, maxiter=1e7L,
                             dist_gamma = gamma_pars4)
sum(rtestim_mod4$convergence == FALSE) == 0 # check the convergence
```

```
## [1] TRUE
```

```
plot(rtestim_mod4)
```

```
cv_mod4 <- cv_estimate_rt(incidence4, korder=3, nfold=3, nsol=50, maxiter=1e7L,
                          dist_gamma = gamma_pars4)
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 with 95% CIs") +
  theme_bw()
```



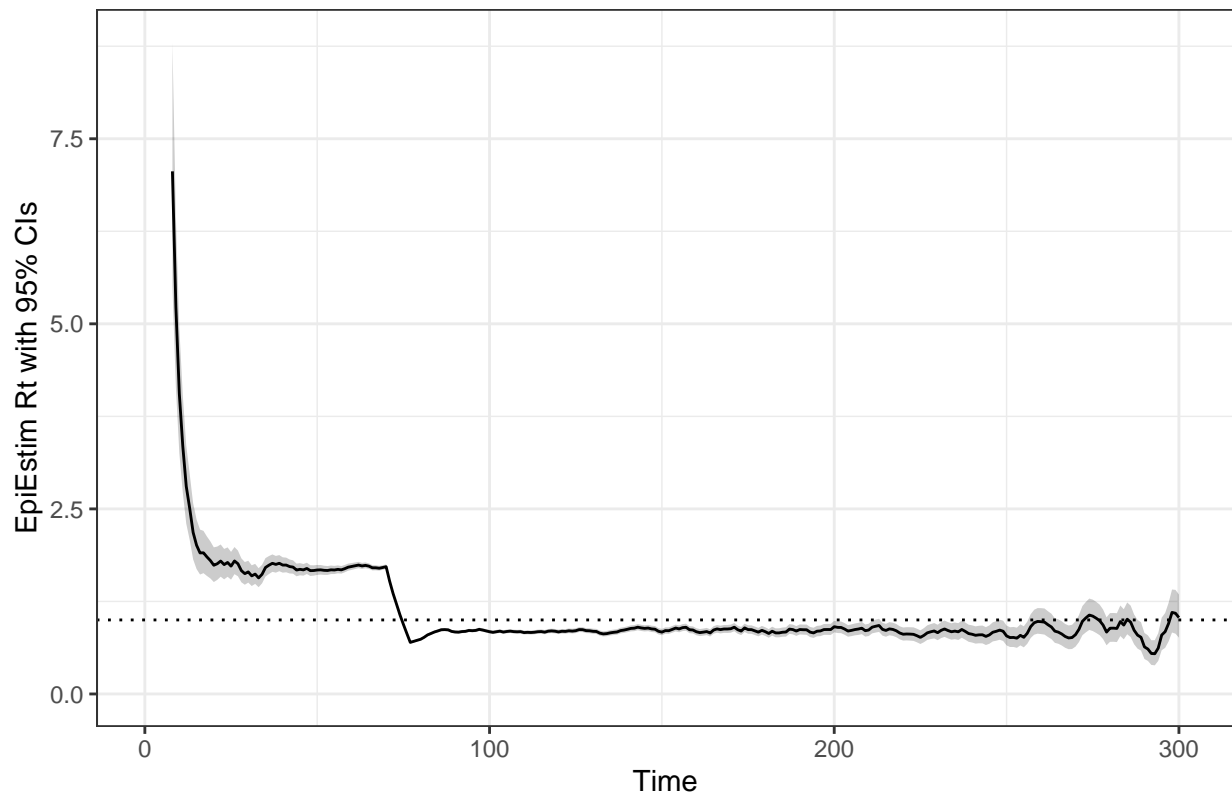
Fit EpiEstim with “true” serial interval.

```
# EpiEstim with weekly sliding windows; specify the SI distribution
library(EpiEstim)
method <- "non_parametric_si"
# Get "true" serial interval: get gamma 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))

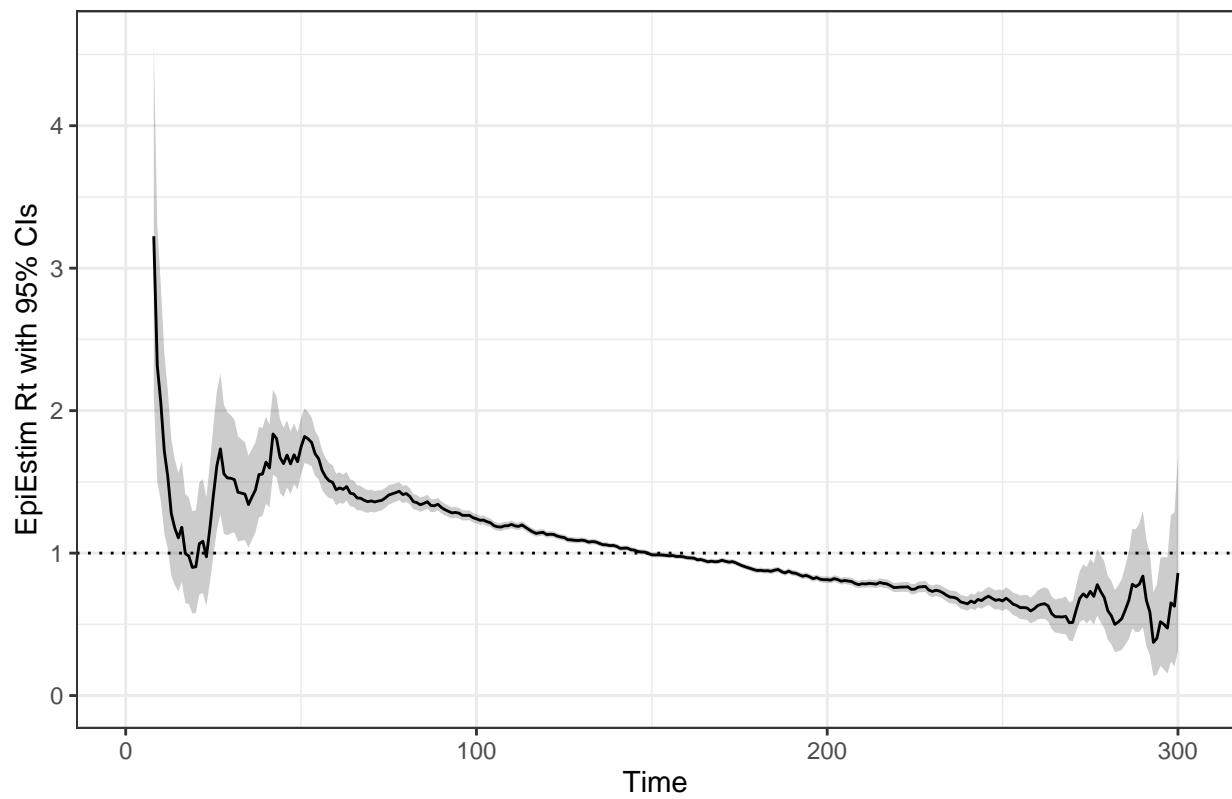
prob_gamma2 <- c(0, diff(c(0, pgamma(1:11, 2.5, scale=2.5)))) / pgamma(11, 2.5, scale=2.5)
config2 <- make_config(list(si_distr = prob_gamma2))

prob_gamma3 <- c(0, diff(c(0, pgamma(1:11, 3.5, scale=3.5)))) / pgamma(11, 3.5, scale=3.5)
config3 <- make_config(list(si_distr = prob_gamma3))

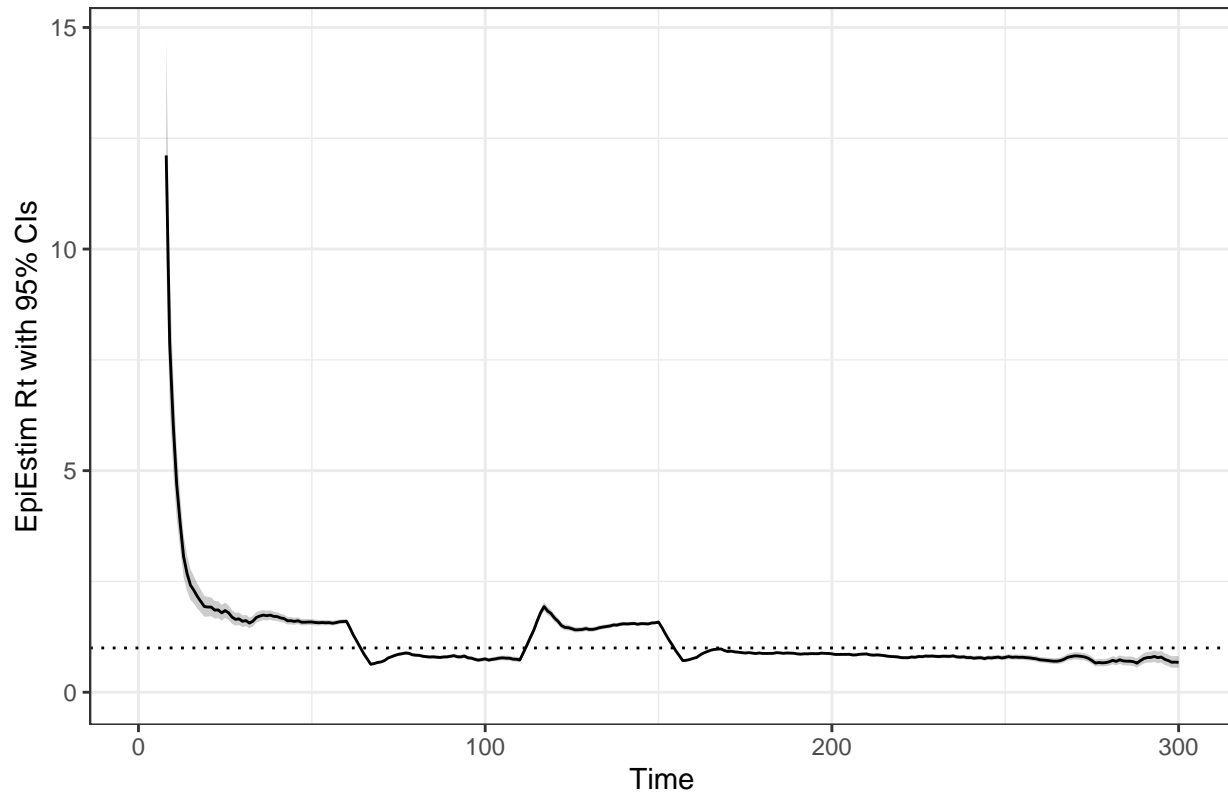
# Fit EpiEstim models
mod_epiEstim1 <- EpiEstim::estimate_R(incid = incidence1, config = config1,
                                     method = method)
plot(mod_epiEstim1, "R") + labs(y="EpiEstim Rt with 95% CIs", title="") + theme_bw() + theme(legend.pos=
```



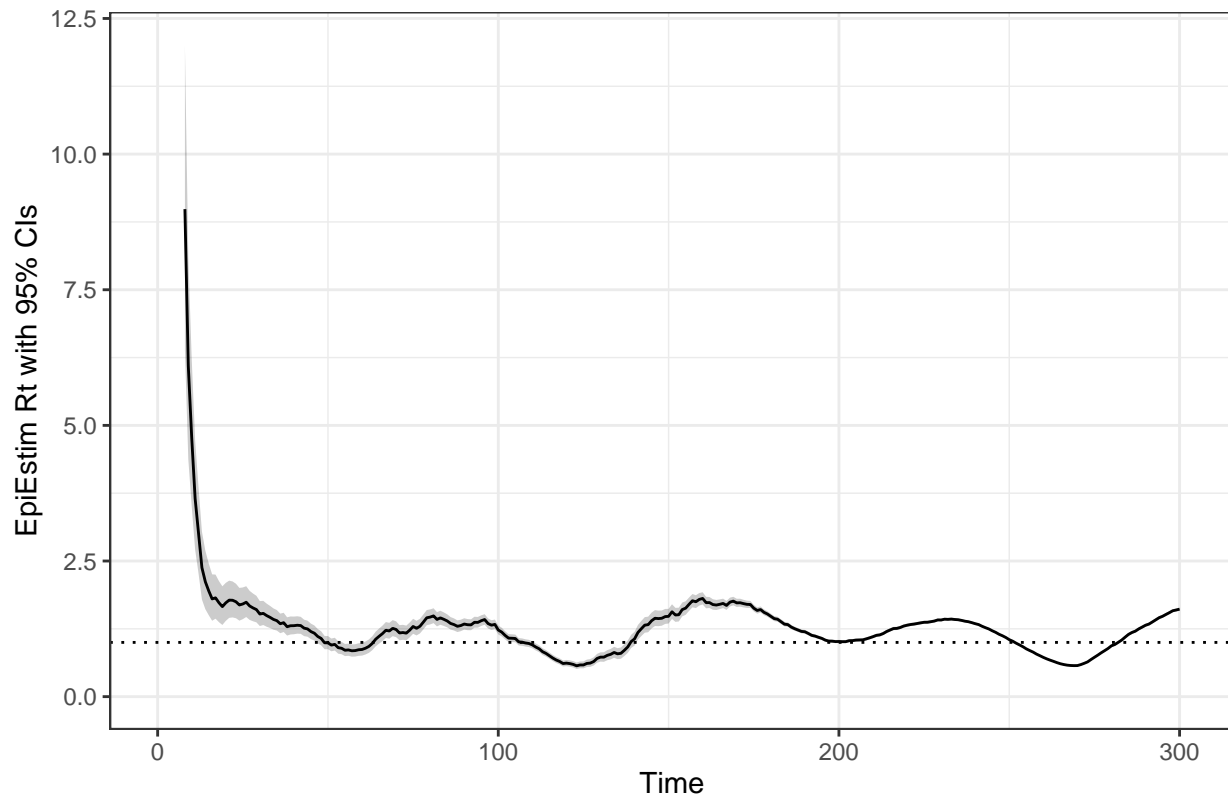
```
mod_epiEstim2 <- EpiEstim::estimate_R(incid = incidence2, config = config2,
                                     method = method)
plot(mod_epiEstim2, "R") + labs(y="EpiEstim Rt with 95% CIs", title="") + theme_bw() + theme(legend.pos = "bottom")
```



```
mod_epiEstim3 <- EpiEstim::estimate_R(incid = incidence3, config = config3,
                                     method = method)
plot(mod_epiEstim3, "R") + labs(y="EpiEstim Rt with 95% CIs", title="") + theme_bw() + theme(legend.pos=
```



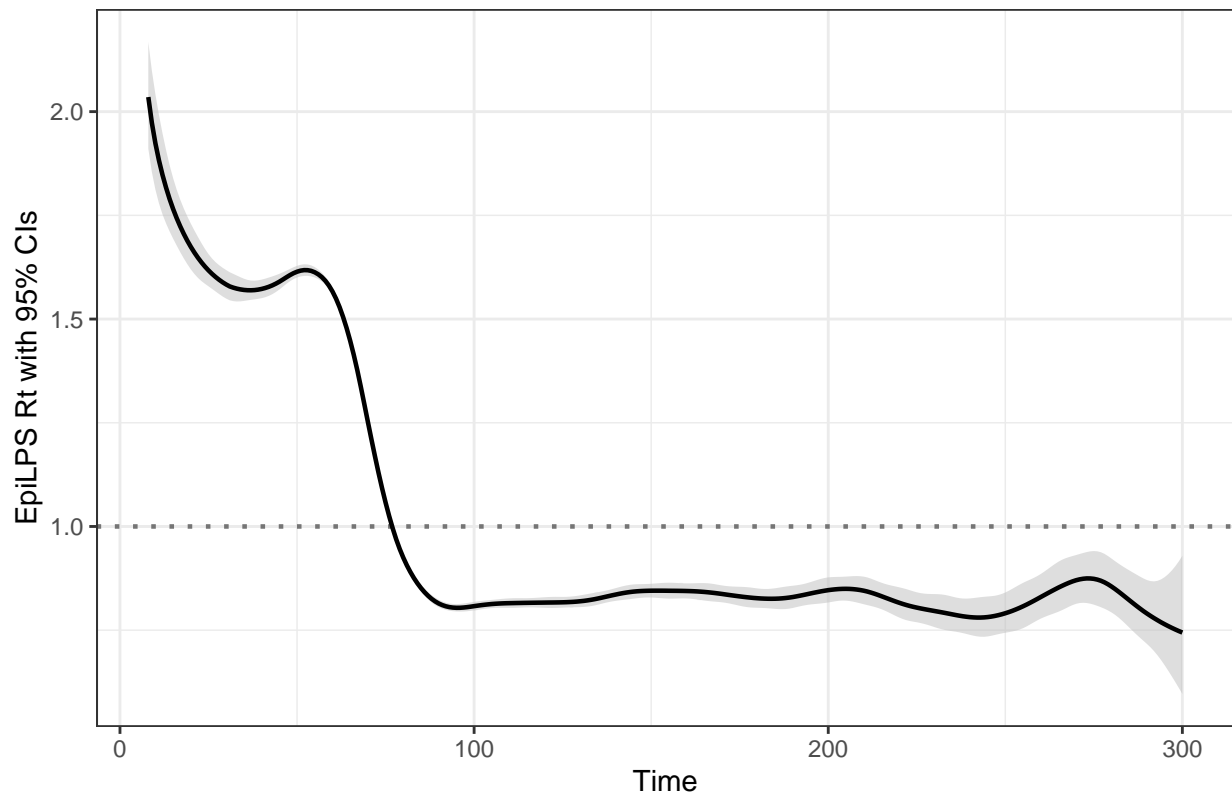
```
mod_epiEstim4 <- EpiEstim::estimate_R(incid = incidence4, config = config3,
                                     method = method)
plot(mod_epiEstim4, "R") + labs(y="EpiEstim Rt with 95% CIs", title="") + theme_bw() + theme(legend.pos=
```



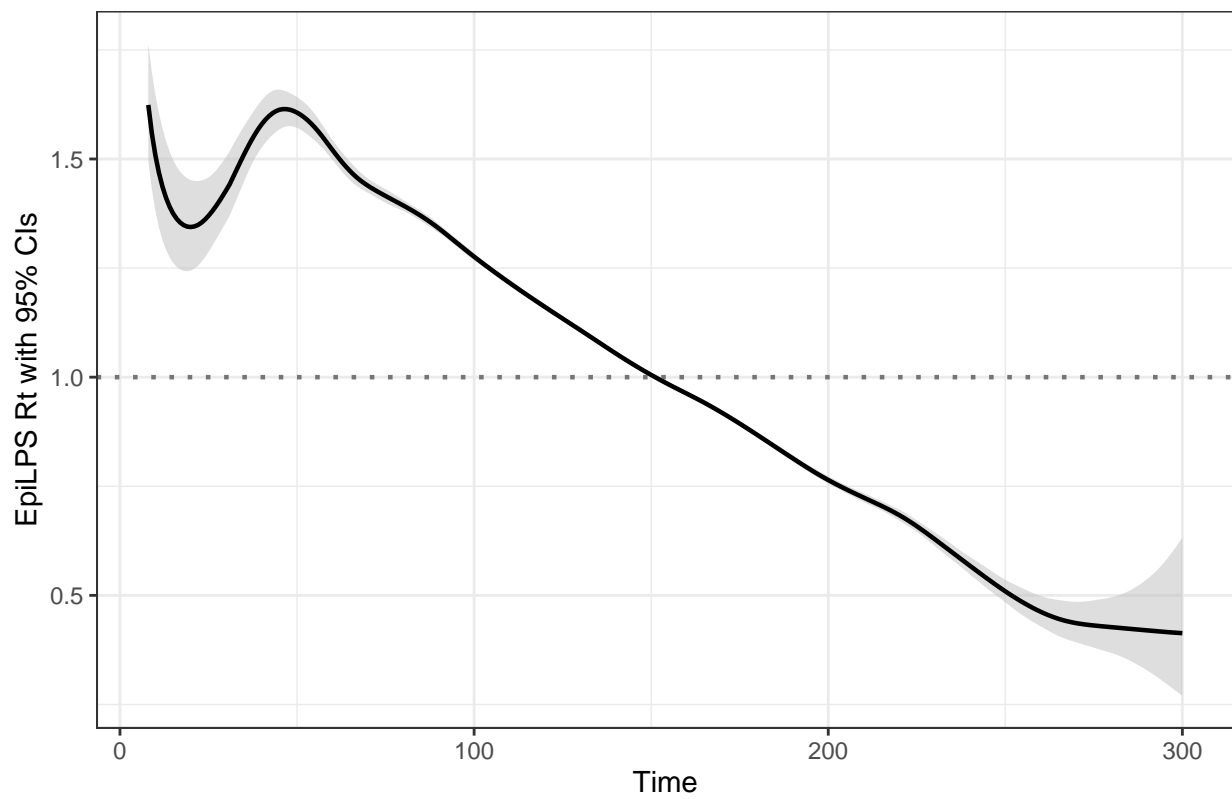
Fit EpiLPS using “true” serial interval.

```
library(EpiLPS)
si1 <- Idist(mean = 9, sd = 27, dist="gamma")$pvec[1:30]
si1 <- si1 / sum(si1)
si2 <- Idist(mean = 2.5^2, sd = 2.5^3, dist="gamma")$pvec[1:30]
si2 <- si2 / sum(si2)
si3 <- Idist(mean = 3.5^2, sd = 3.5^3, dist="gamma")$pvec[1:30]
si3 <- si3 / sum(si3)
si4 <- si3

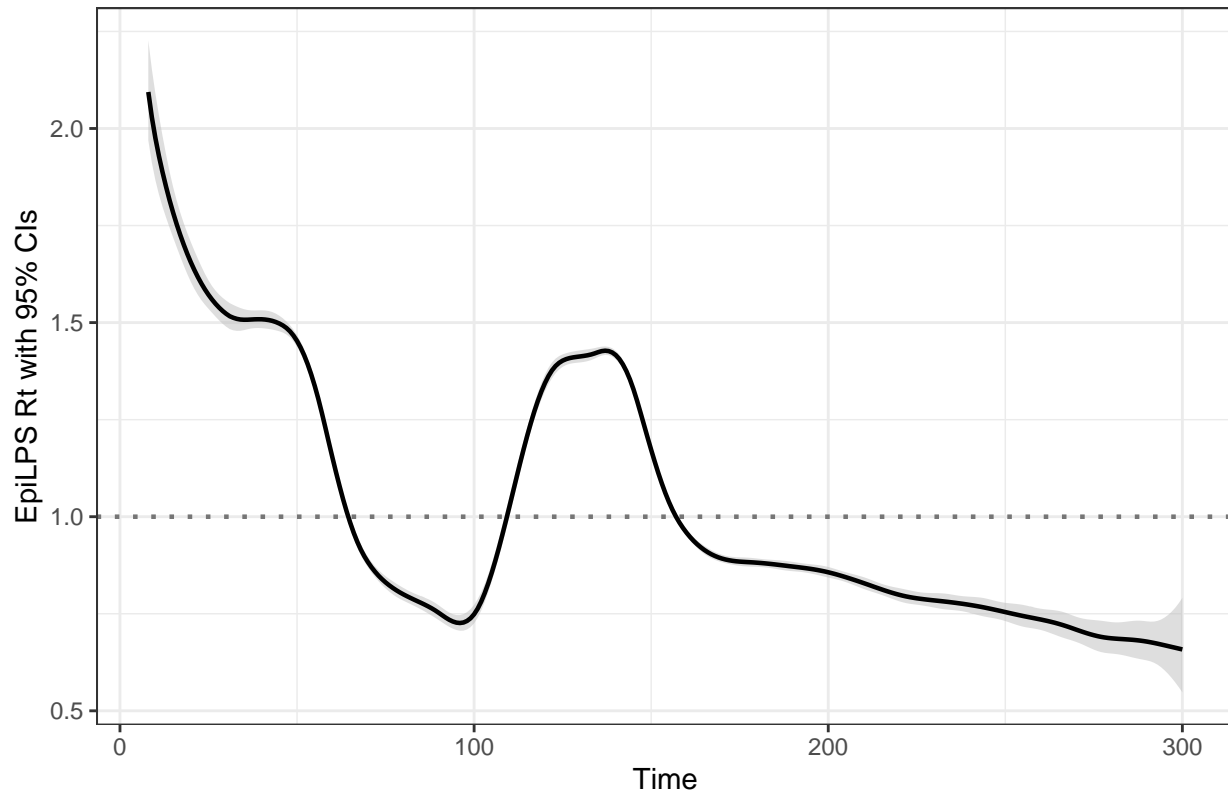
mod_epilst_pois1 <- estimR(incidence = incidence1, si = si1)
plot(mod_epilst_pois1) + labs(y="EpiLPS Rt with 95% CIs", title="") + theme_bw() + theme(legend.position = "bottom")
```



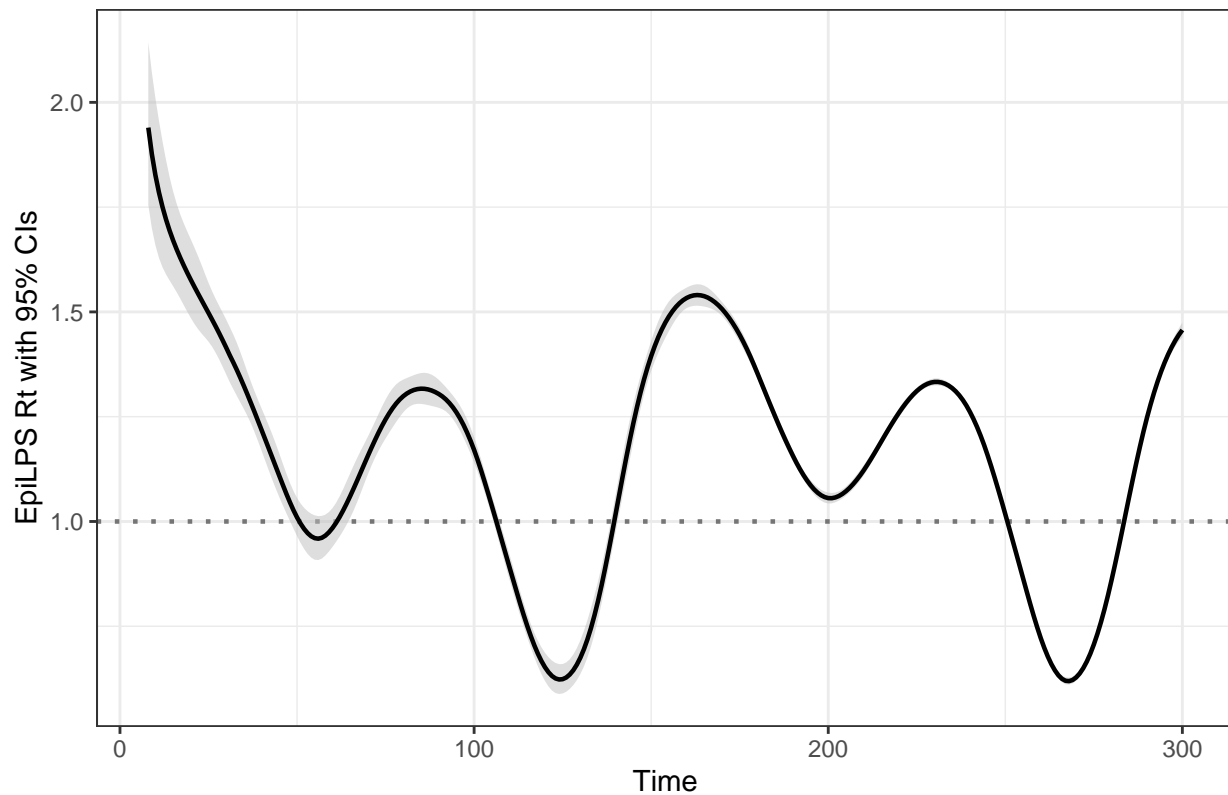
```
mod_epilst_pois2 <- estimR(incidence = incidence2, si = si2)
plot(mod_epilst_pois2) + labs(y="EpiLPS Rt with 95% CIs", title="") + theme_bw() + theme(legend.position = "bottom")
```



```
mod_epilst_pois3 <- estimR(incidence = incidence3, si = si3)
plot(mod_epilst_pois3) + labs(y="EpiLPS Rt with 95% CIs", title="") + theme_bw() + theme(legend.position="bottom")
```



```
mod_epilst_pois4 <- estimR(incidence = incidence4, si = si4)
plot(mod_epilst_pois4) + labs(y="EpiLPS Rt with 95% CIs", title="") + theme_bw() + theme(legend.position="bottom")
```



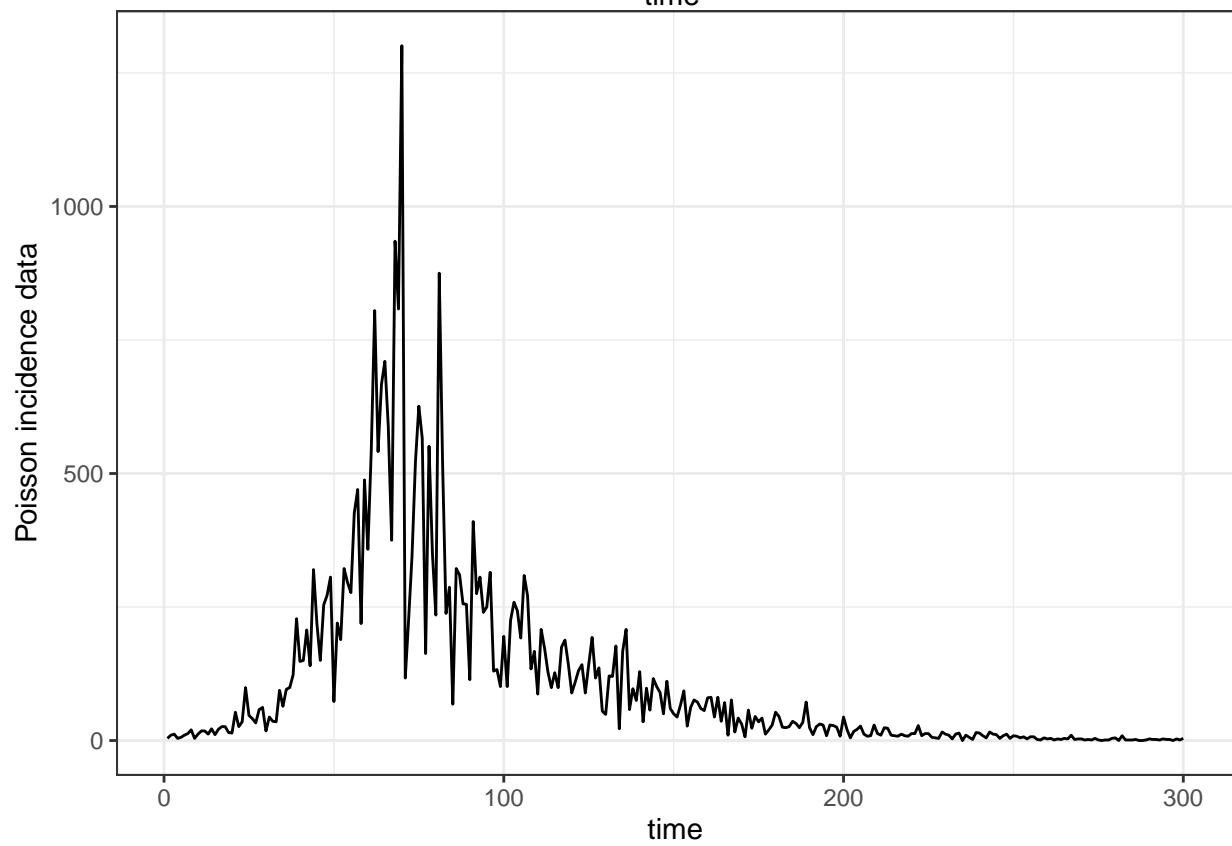
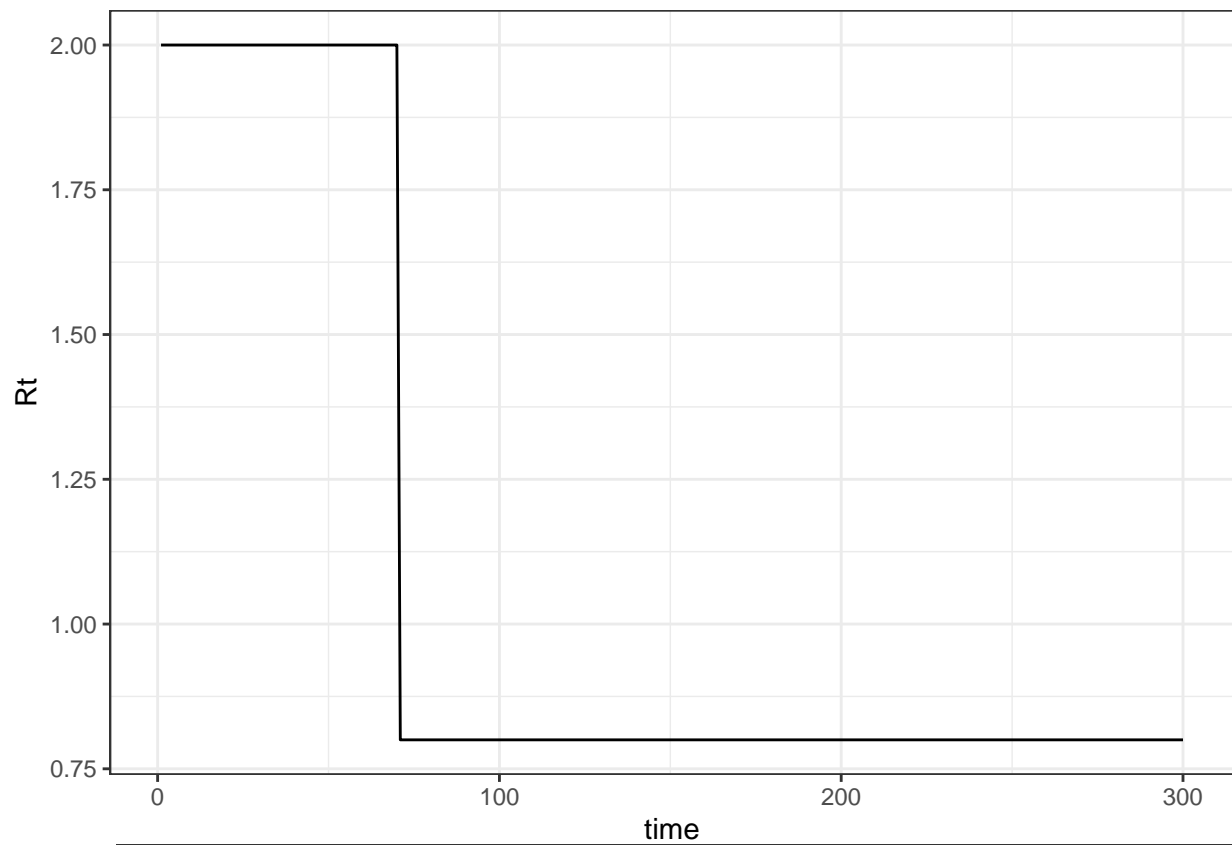
Synthetic data with negative Binomial incidence

```
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)
  }

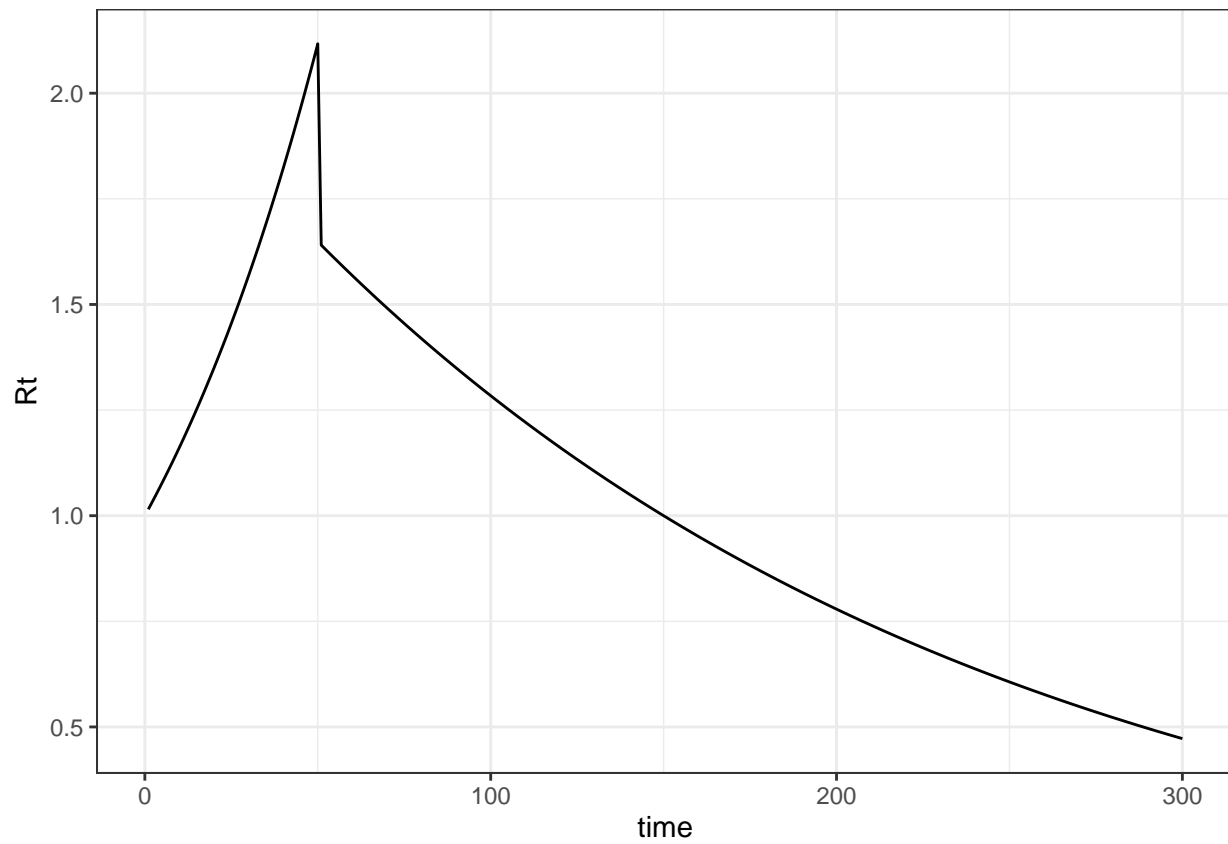
  return(NB_count)
}

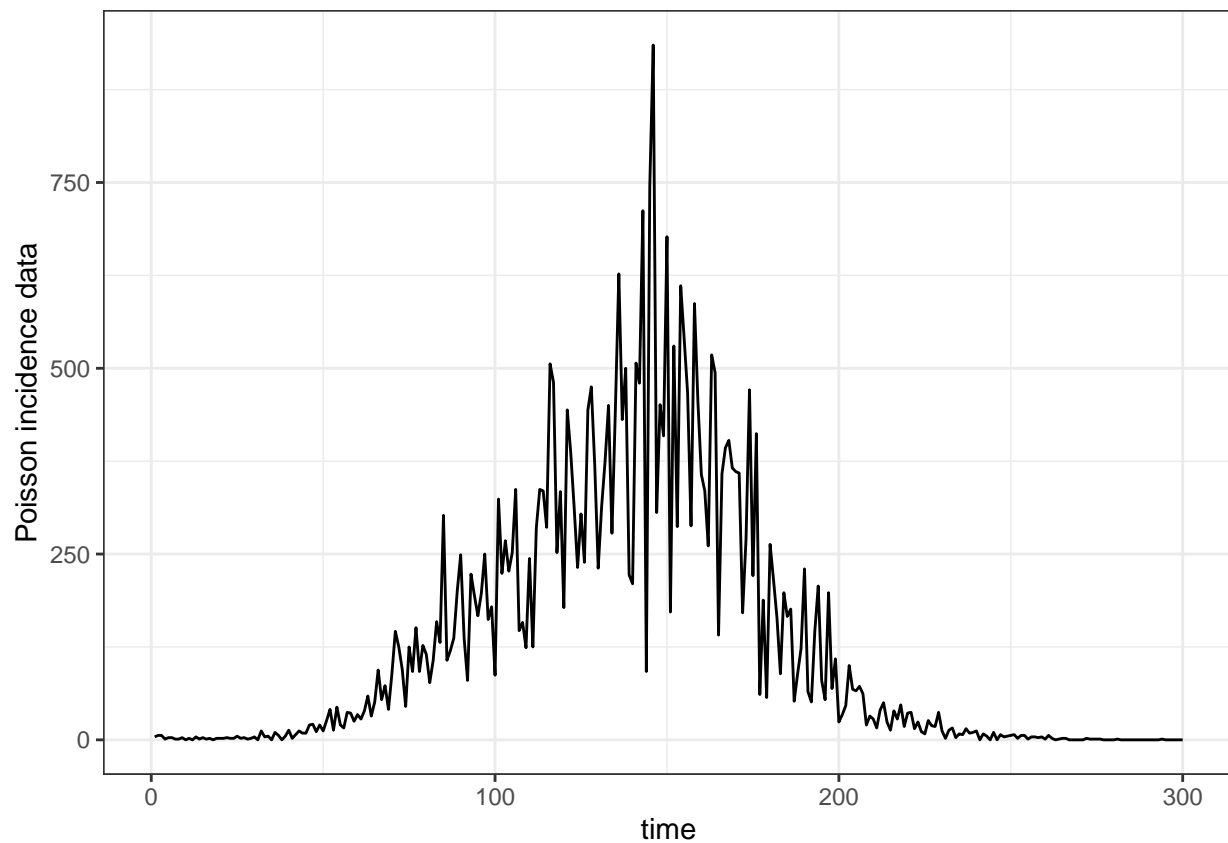
# case 1
seed <- 629
set.seed(seed)
nb_incidence1 <- get_nb_incidence(N1, Rt1, gamma_pars1)
display_dat(nb_incidence1, Rt1)
```

```
check_dat(nb_incidence1, Rt1)

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





```
check_dat(nb_incidence2, Rt2)
```

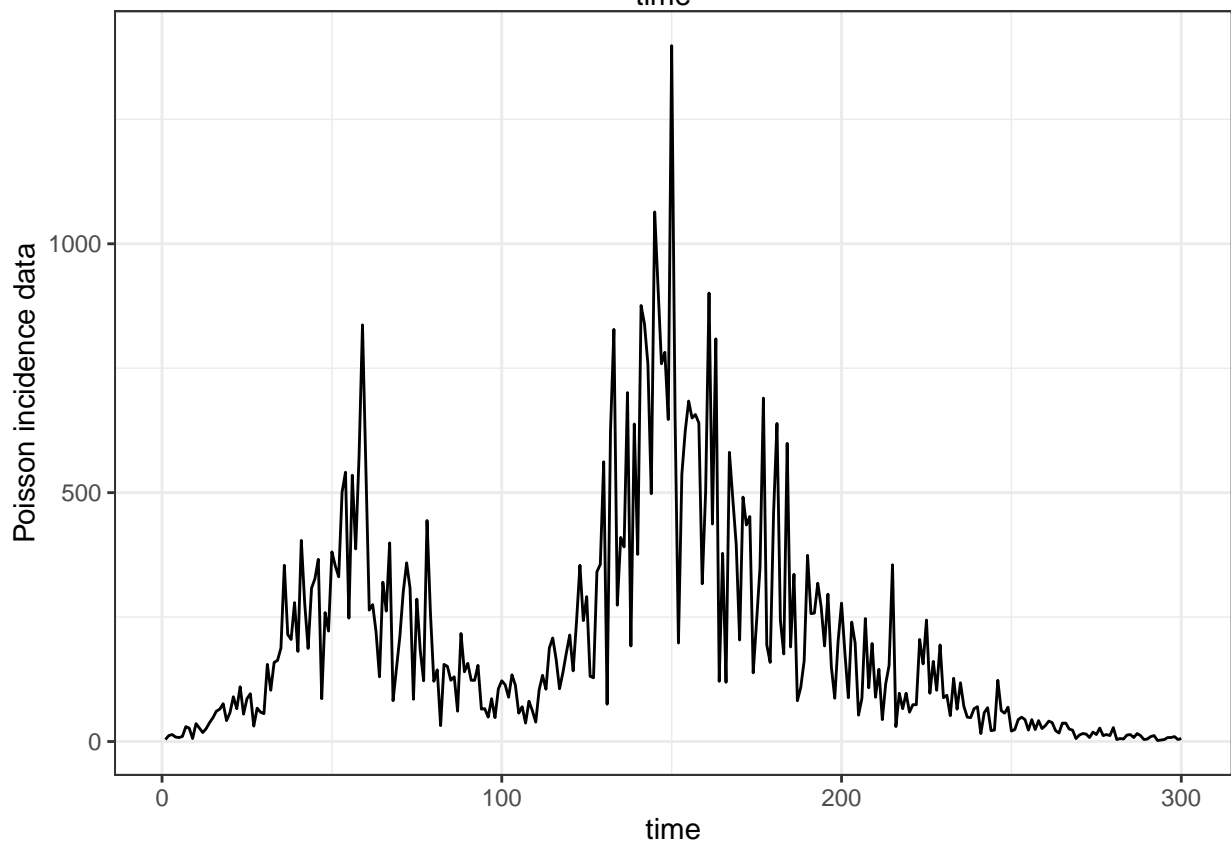
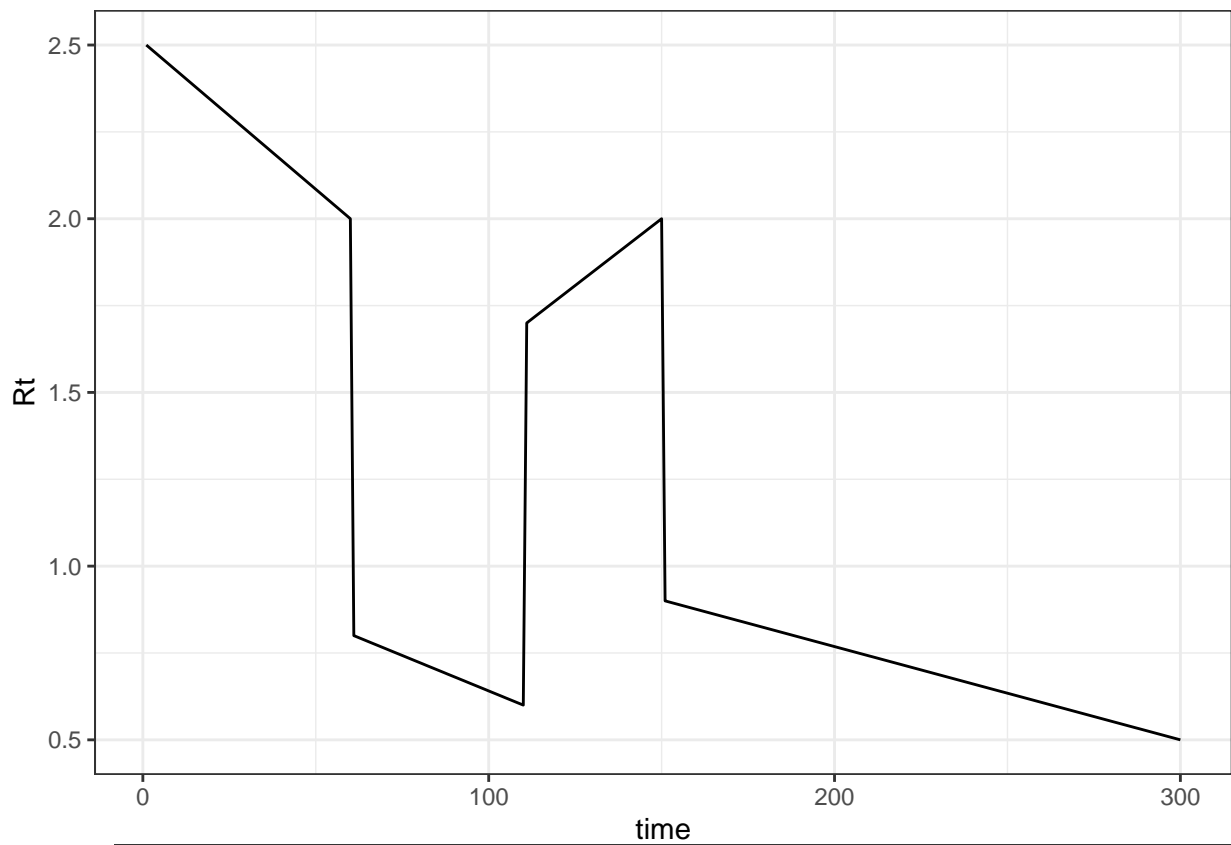
```
## Warning: `incidence` data has more than 10% 0s.
```

```
# case 3
```

```
set.seed(seed)
```

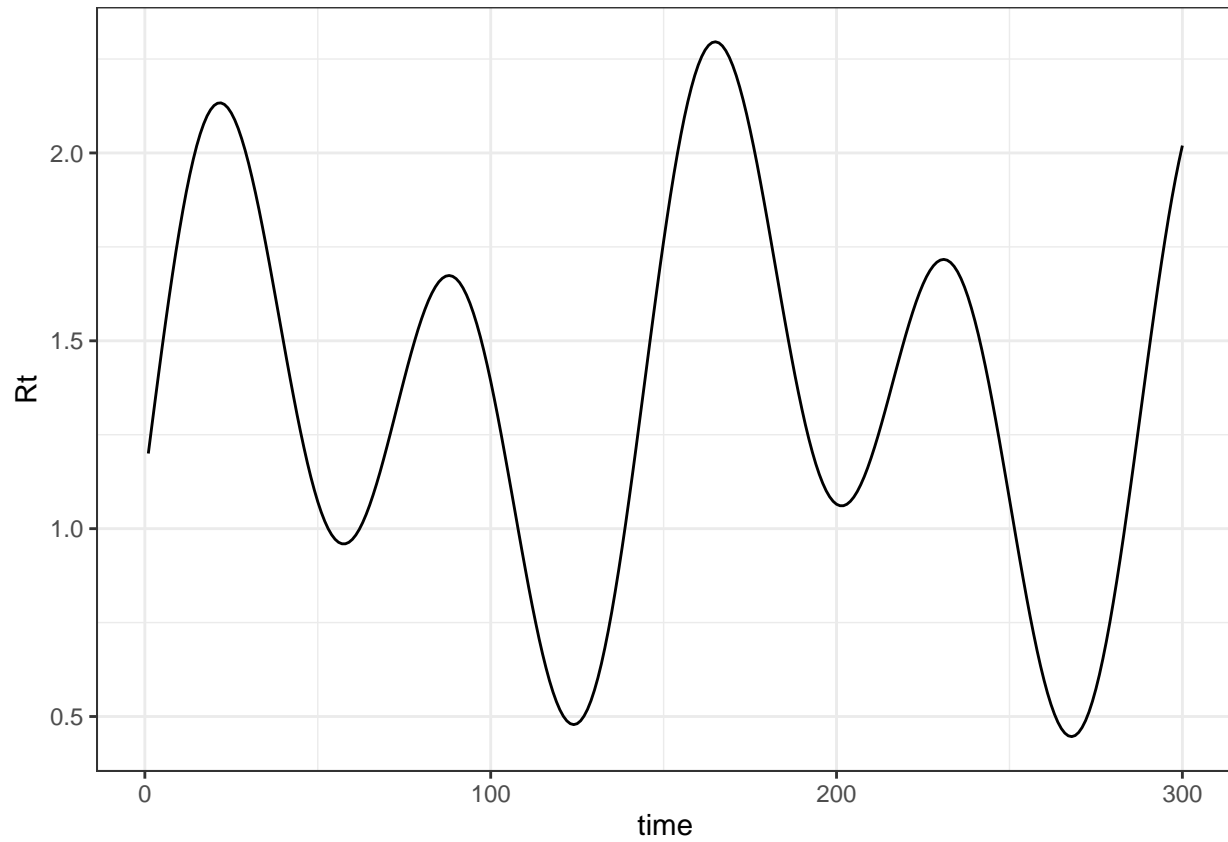
```
nb_incidence3 <- get_nb_incidence(N1, Rt3, gamma_pars3)
```

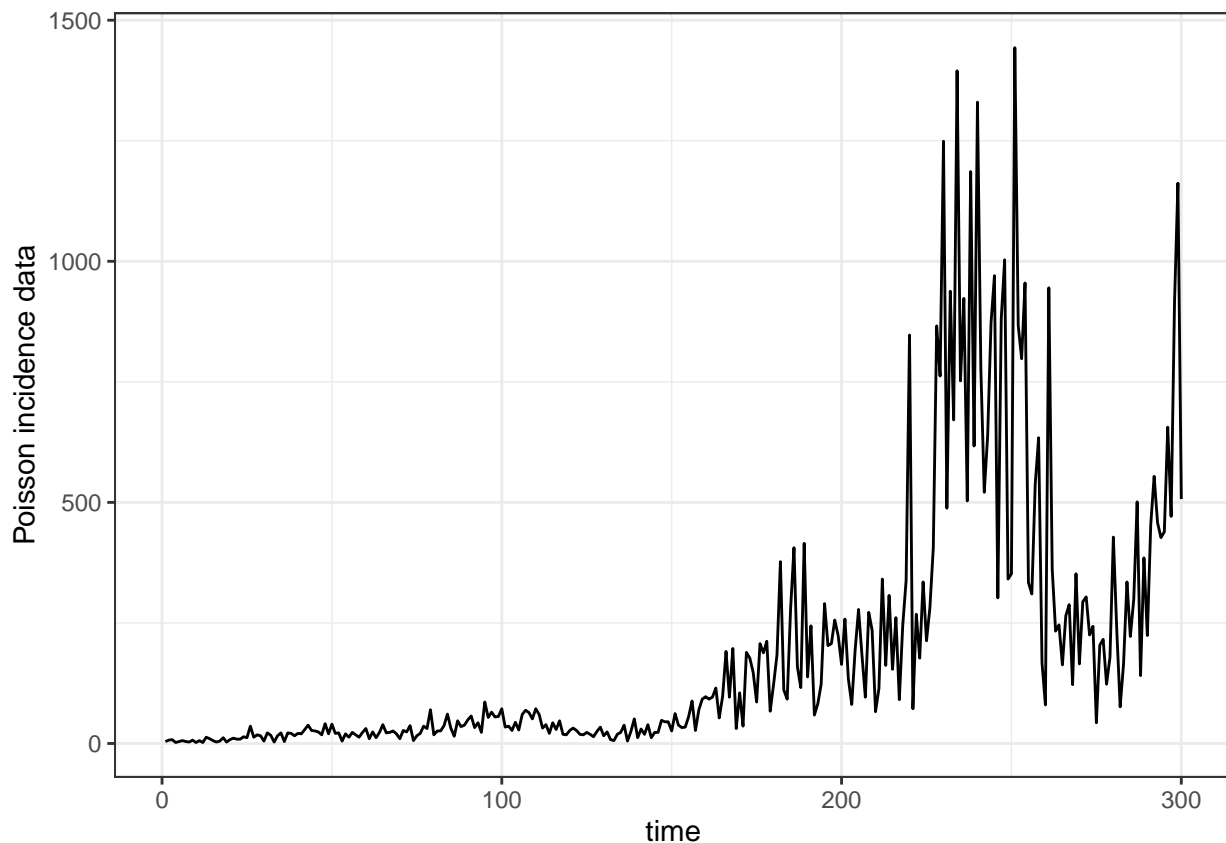
```
display_dat(nb_incidence3, Rt3)
```



```
check_dat(nb_incidence3, Rt3)

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

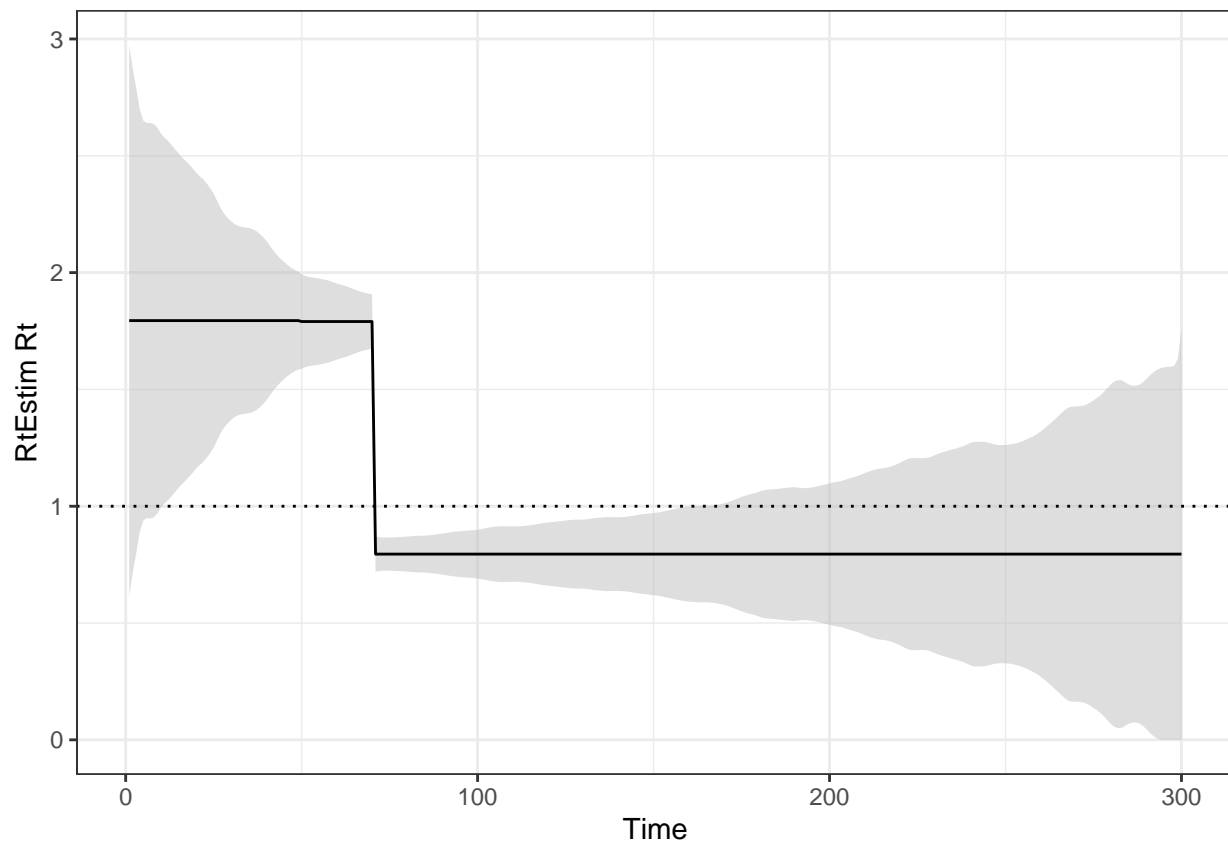




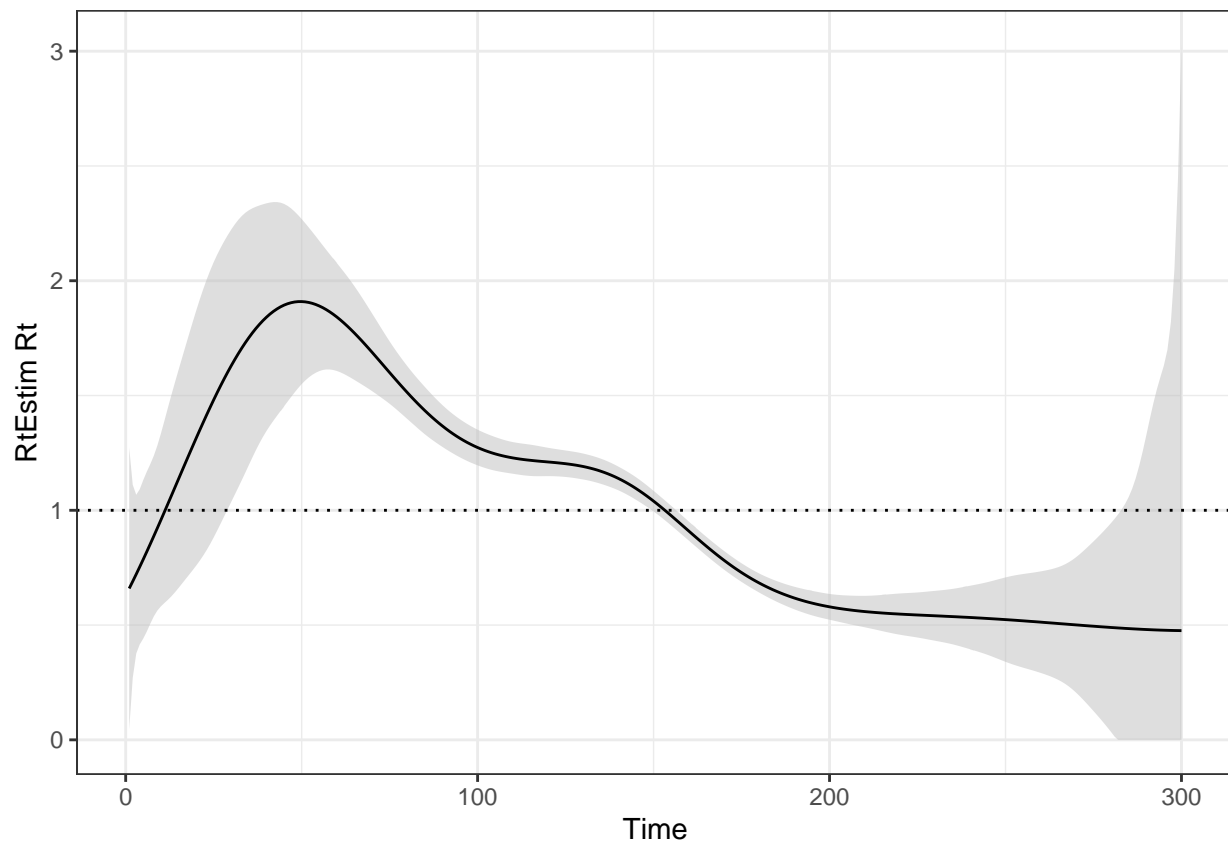
```
check_dat(nb_incidence4, Rt4)
```

Estimate R_t using our Poisson trend filtering using `RtEstim`.

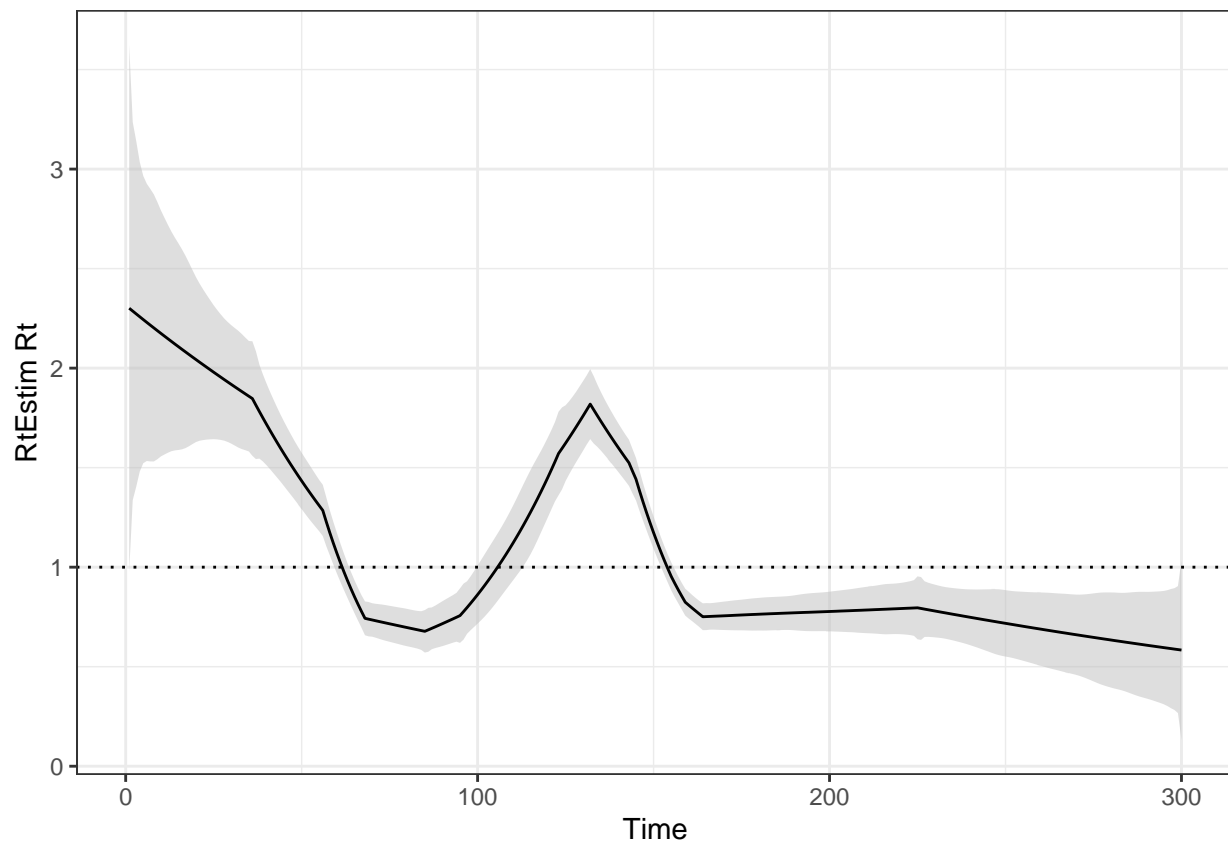
```
cv_mod_nb1 <- cv_estimate_rt(nb_incidence1, korder=0, nfold=3, nsol=50,
                             dist_gamma = c(3, 3))
rtestim_tuned_mod_nb1 <- cv_mod_nb1$full_fit$Rt[,which.min(cv_mod_nb1$cv_scores)]
rt_ci_nb1 <- confband(cv_mod_nb1, "lambda.min") # get 95% confidence band
rt_ci_nb1 %>%
  ggplot(aes(x = 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()
```



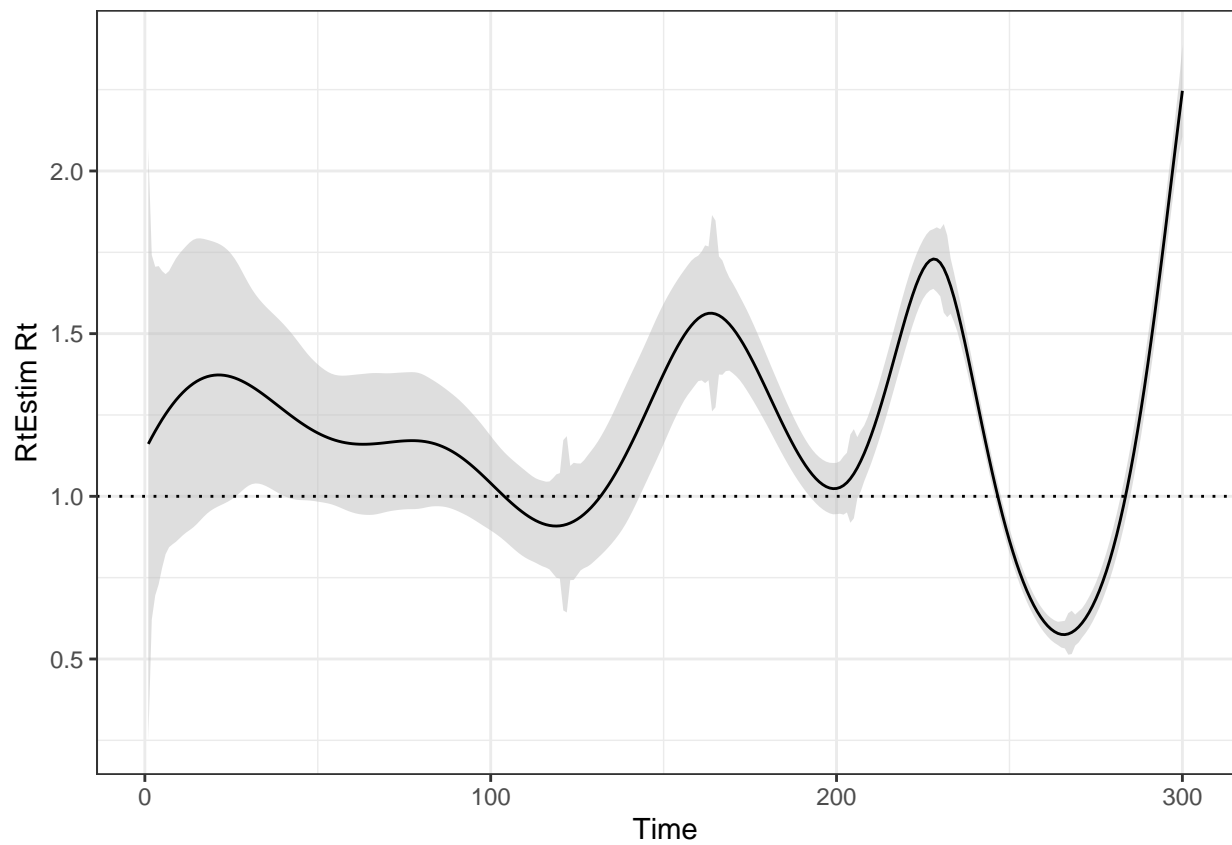
```
cv_mod_nb2 <- cv_estimate_rt(nb_incidence2, korder=3, nfold=3, nsol=50,
                             dist_gamma = c(3, 3))
rtestim_tuned_mod_nb2 <- cv_mod_nb2$full_fit$Rt[,which.min(cv_mod_nb2$cv_scores)]
rt_ci_nb2 <- confband(cv_mod_nb2, "lambda.min") # get 95% confidence band
rt_ci_nb2 %>%
  ggplot(aes(x = 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=50,
                             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()
```

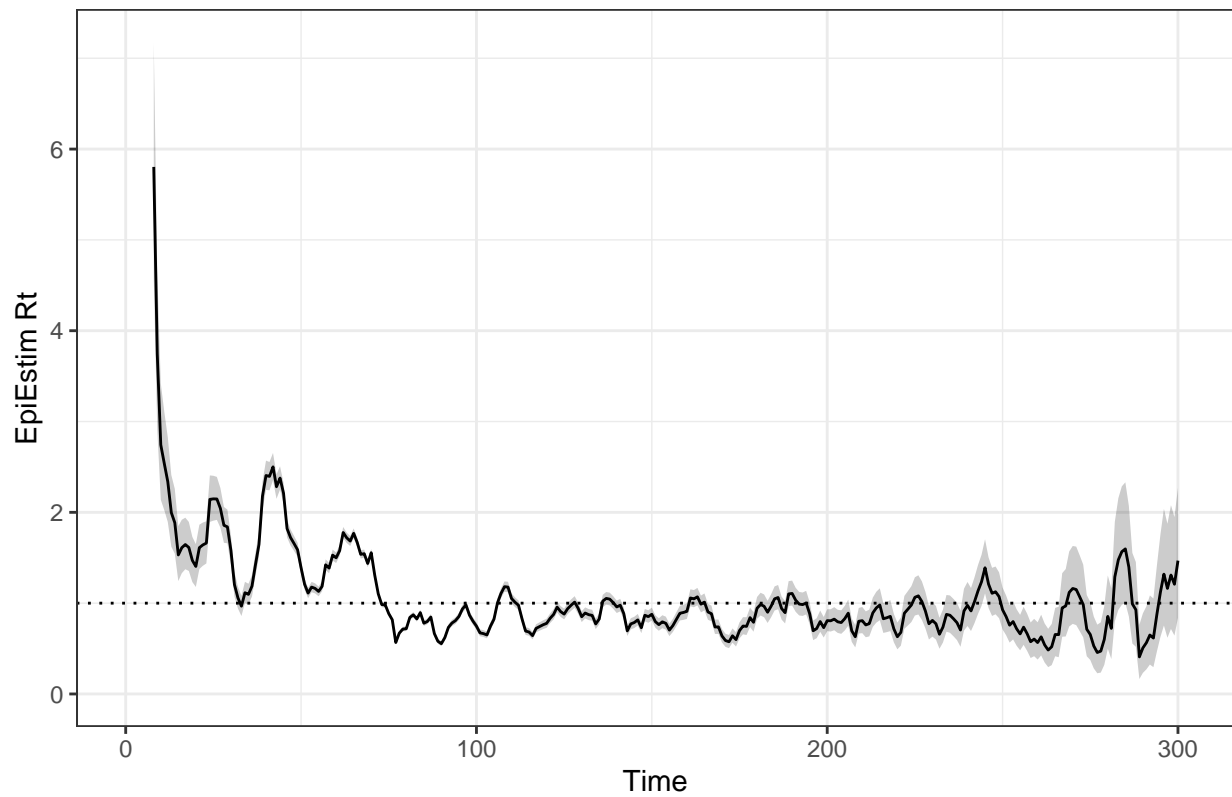



```
cv_mod_nb4 <- cv_estimate_rt(nb_incidence4, korder=3, nfold=3, nsol=50,
                             dist_gamma = c(3, 3)) # k=2 or 3?
rtestim_tuned_mod_nb4 <- cv_mod_nb4$full_fit$Rt[,which.min(cv_mod_nb4$cv_scores)]
rt_ci_nb4 <- confband(cv_mod_nb4, "lambda.min") # get 95% confidence band
rt_ci_nb4 %>%
  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()
```

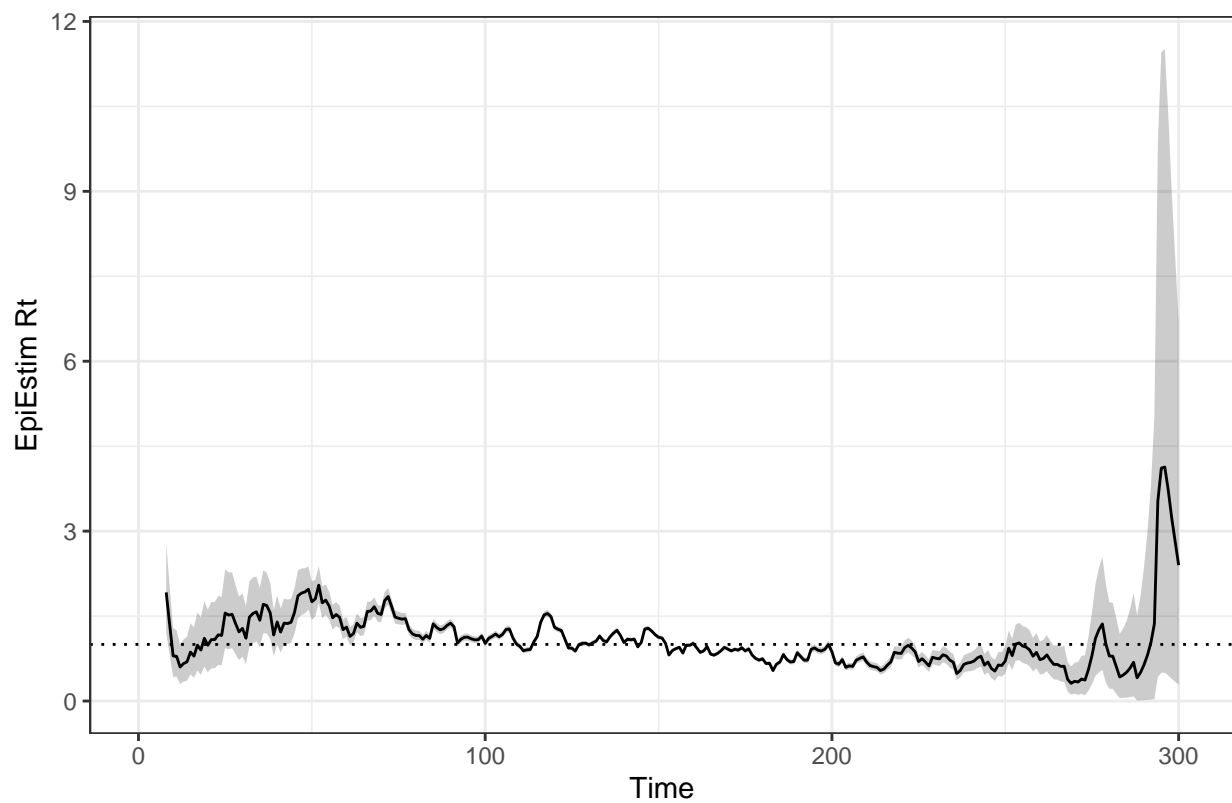


Estimate Rt using EpiEstim.

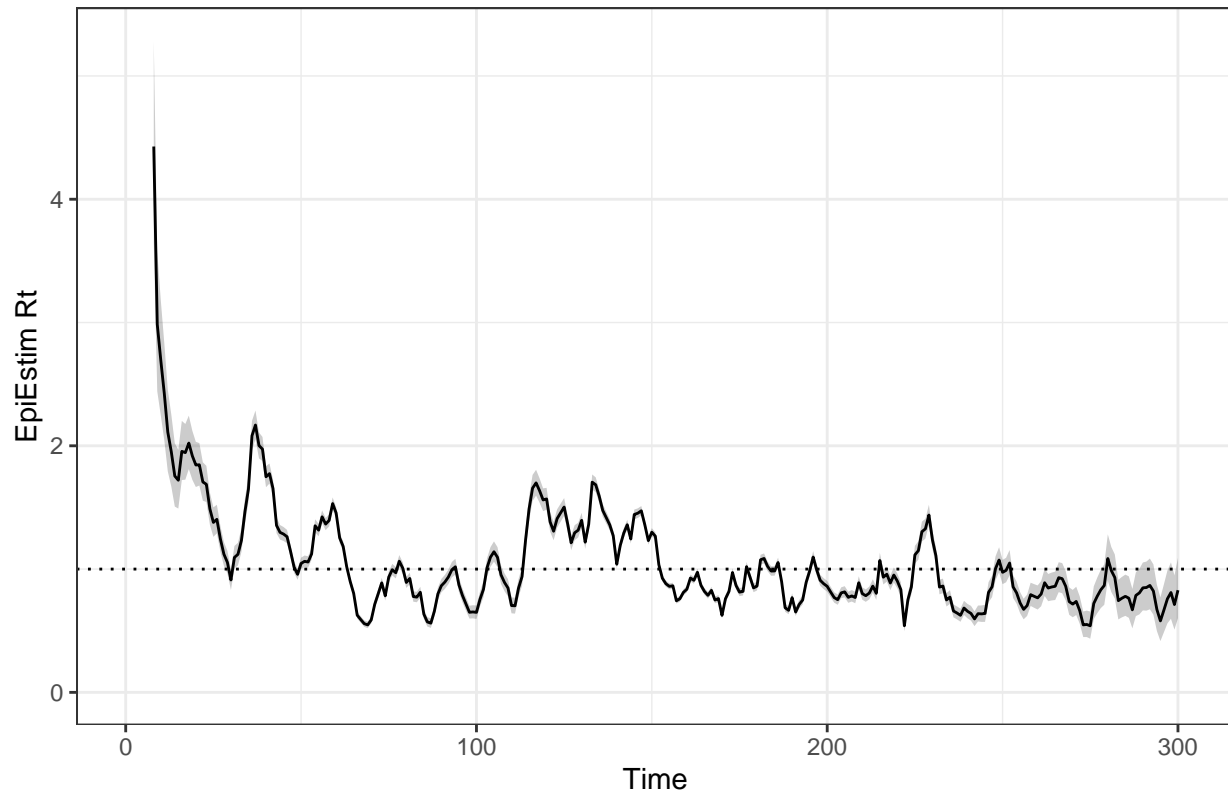
```
mod_epi_nb1 <- EpiEstim::estimate_R(incid = nb_incidence1, config = config1, method = method)
plot(mod_epi_nb1, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none")
```



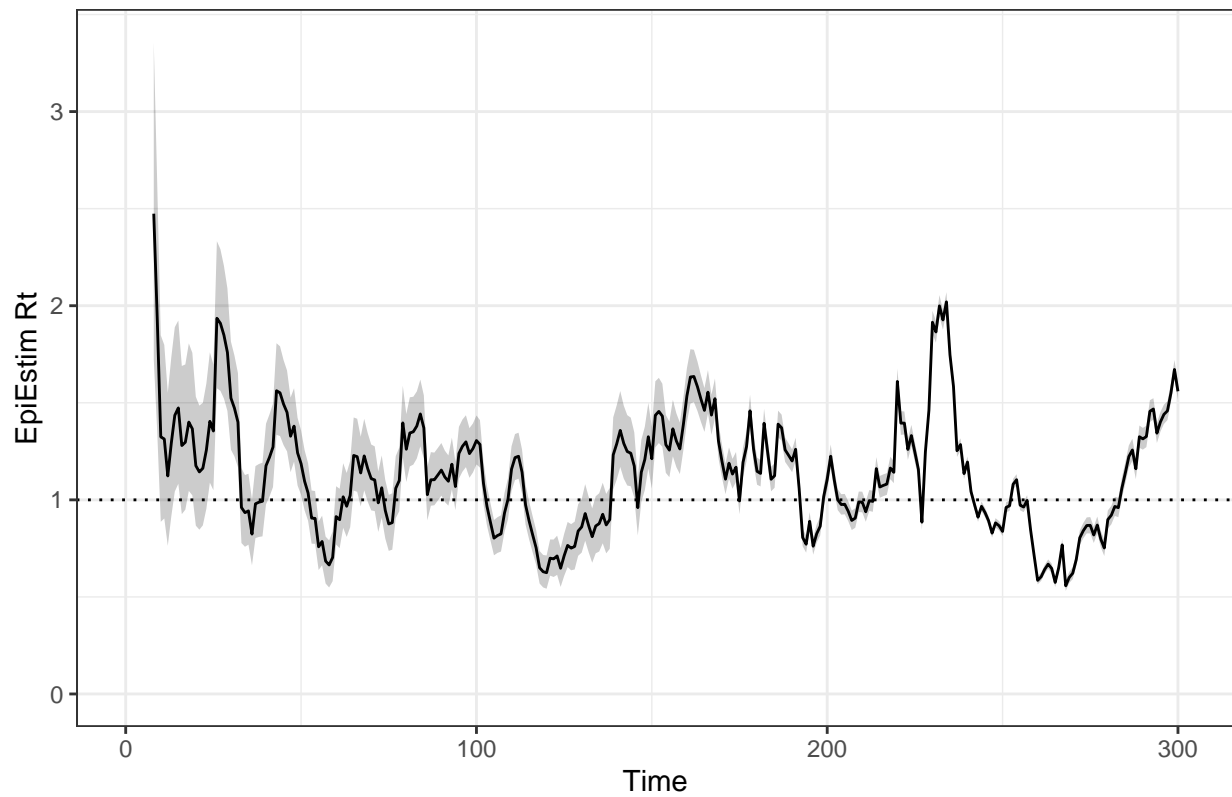
```
mod_epi_nb2 <- EpiEstim::estimate_R(incid = nb_incidence2, config = config2, method = method)
plot(mod_epi_nb2, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none")
```



```
mod_epi_nb3 <- EpiEstim::estimate_R(incid = nb_incidence3, config = config2, method = method)
plot(mod_epi_nb3, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none")
```

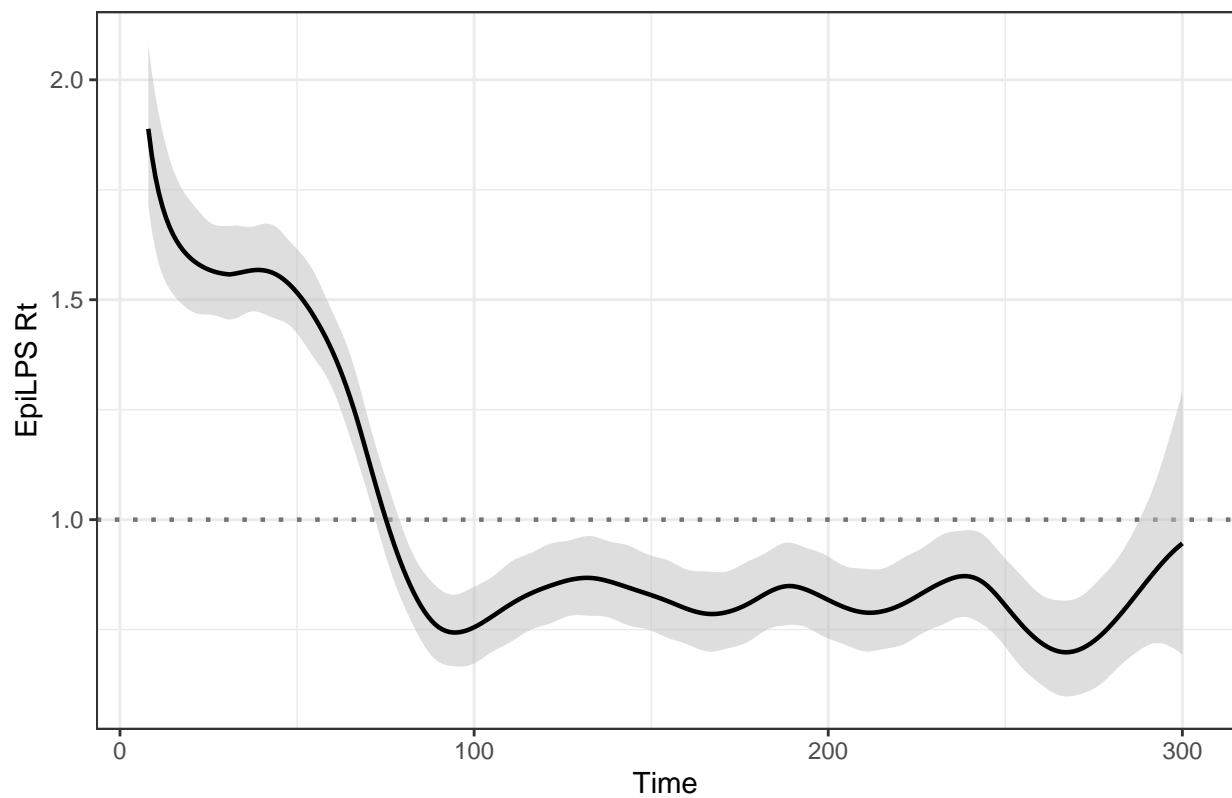


```
mod_epi_nb4 <- EpiEstim::estimate_R(incid = nb_incidence4, config = config2, method = method)
plot(mod_epi_nb4, "R") + labs(y="EpiEstim Rt", title="") + theme_bw() + theme(legend.position = "none")
```

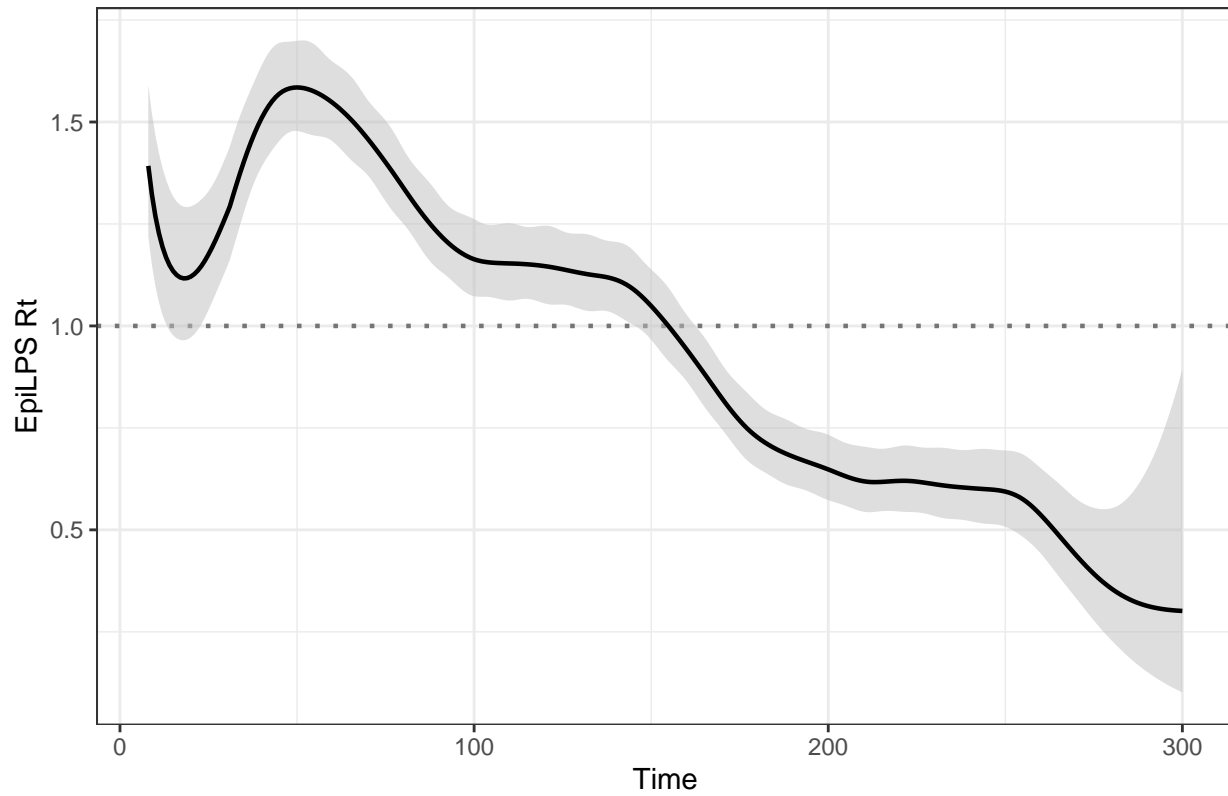


Estimate Rt using EpiLPS.

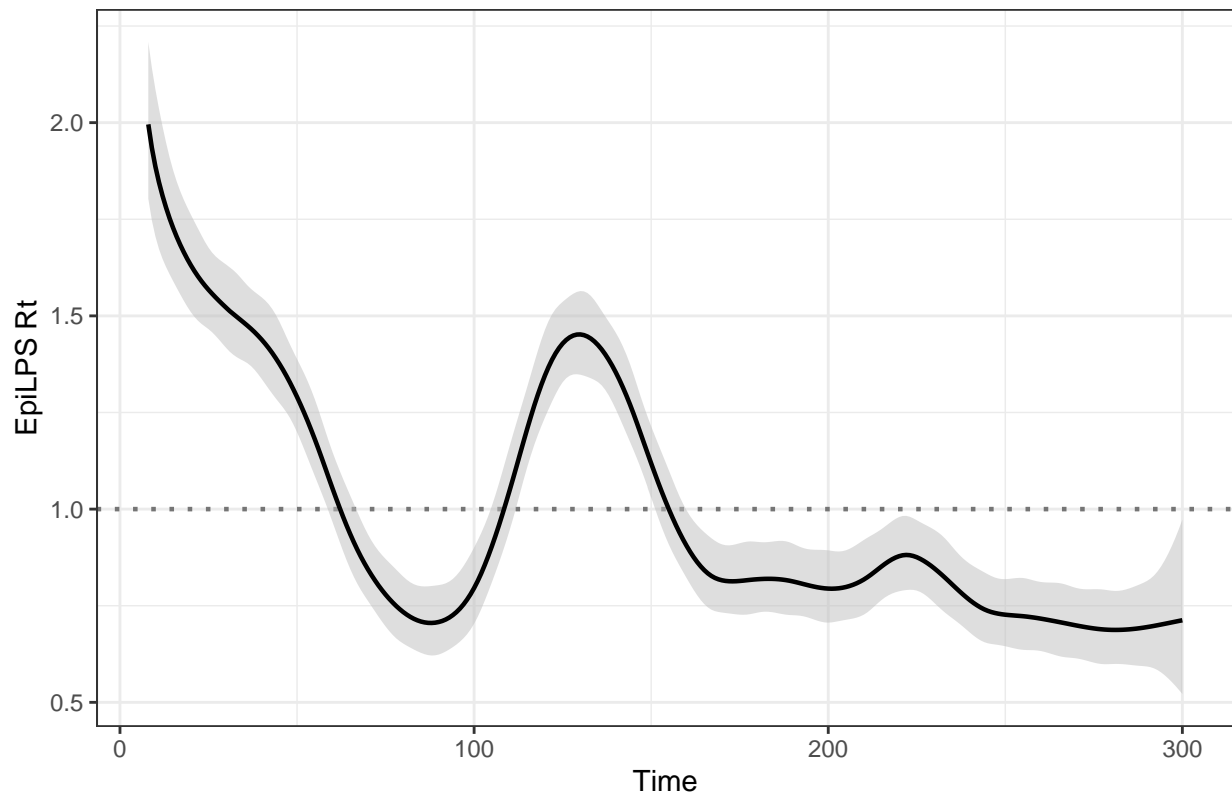
```
mod_epilst_nb1 <- estimR(incidence = nb_incidence1, si = si1, CoriR = TRUE)
plot(mod_epilst_nb1) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```



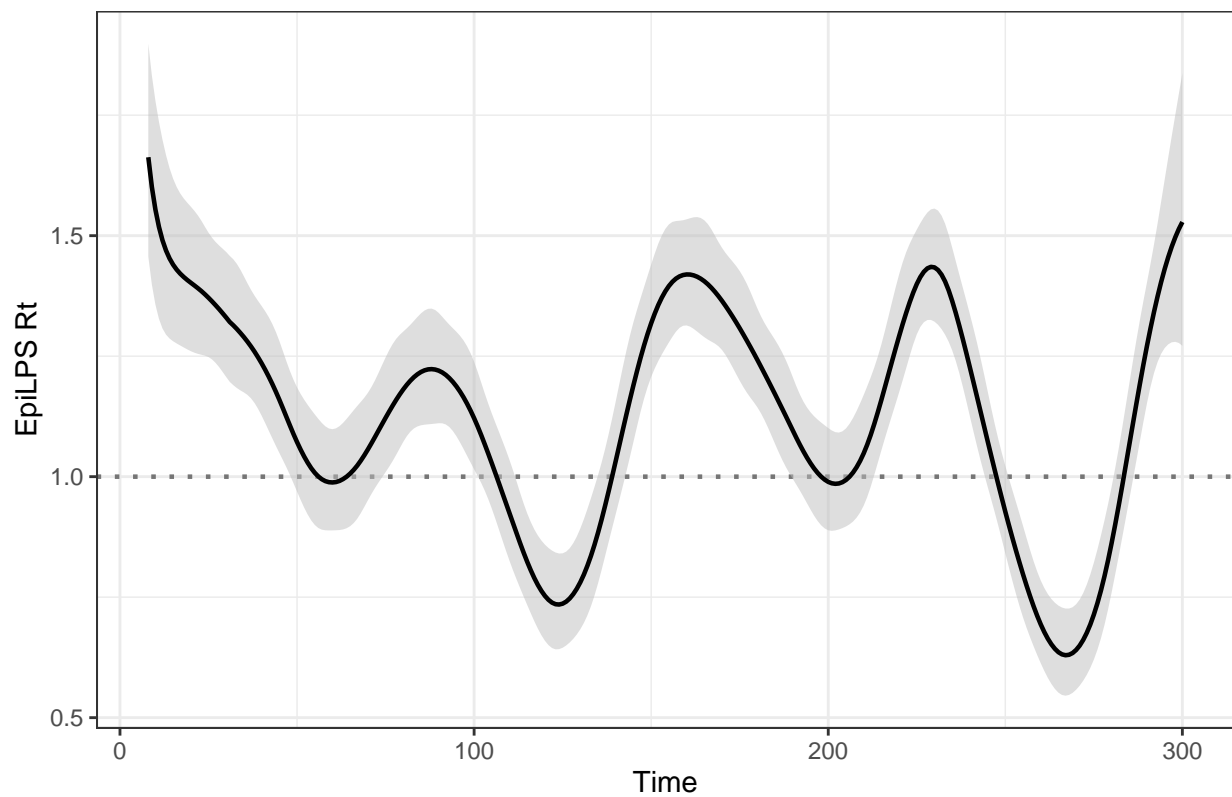
```
mod_epilst_nb2 <- estimR(incidence = nb_incidence2, si = si2, CoriR = TRUE)
plot(mod_epilst_nb2) + labs(y="EpiLPS Rt", title="") + theme_bw() + theme(legend.position = "none")
```



```
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")
```



Display and save results for synthetic data

Create a results table to save all Rt and incidence samples, and Rt estimates using three methods. Save the results table.

```
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
)

# save Rt, incidence and estimates
saveRDS(res_dat, here::here("dat/def_examples.RDS"))
```

Display results of Poisson incidence.

```
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") %>%
```



```

mutate(Rt_type = fct_recode(Rt_type, "True Rt" = "Rt1",
                             "RtEstim" = "Pois_Rtestim1",
                             "EpiEstim" = "Pois_Epi1",
                             "EpiLPS" = "Pois_EpiLPS1"
                             )) %>%

mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                             "RtEstim",
                             "EpiEstim",
                             "EpiLPS"
                             )) %>%

group_by(Rt_type) %>%
ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
geom_line() +
scale_colour_manual(values = cbPalette) +
labs(y="", color = "Rt methods", x="", title="Scenario 1") +
theme_bw()

fig2 <- res_dat %>%
select(time, Rt2, Pois_Epi2, Pois_Rtestim2, Pois_EpiLPS2) %>%
pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
mutate(Rt_type = fct_recode(Rt_type, "True Rt" = "Rt2",
                             "RtEstim" = "Pois_Rtestim2",
                             "EpiEstim" = "Pois_Epi2",
                             "EpiLPS" = "Pois_EpiLPS2"
                             )) %>%

mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                             "RtEstim",
                             "EpiEstim",
                             "EpiLPS"
                             )) %>%

group_by(Rt_type) %>%
ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
geom_line() +
scale_colour_manual(values = cbPalette) +
labs(y="", color="Rt methods", x="", title="Scenario 2") +
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") %>%
mutate(Rt_type = fct_recode(Rt_type, "True Rt" = "Rt3",
                             "RtEstim" = "Pois_Rtestim3",
                             "EpiEstim" = "Pois_Epi3",
                             "EpiLPS" = "Pois_EpiLPS3"
                             )) %>%

mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                             "RtEstim",
                             "EpiEstim",
                             "EpiLPS"
                             )) %>%

group_by(Rt_type) %>%
ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
geom_line() +

```

```

scale_colour_manual(values = cbPalette) +
labs(y="", color="Rt methods", x="", title="Scenario 3") +
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") %>%
  mutate(Rt_type = fct_recode(Rt_type, "True Rt" = "Rt4",
                              "RtEstim" = "Pois_Rtestim4",
                              "EpiEstim" = "Pois_Epi4",
                              "EpiLPS" = "Pois_EpiLPS4"
                              )) %>%
  mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                              "RtEstim",
                              "EpiEstim",
                              "EpiLPS"
                              )) %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="", color="Rt methods", x="", title="Scenario 4") +
  theme_bw()

library(ggpubr)
fig_Pois_res <- ggpubr::ggarrange(
  fig1, fig2, fig3, fig4, ncol=2, nrow=2,
  common.legend = TRUE, legend = "bottom",
  font.label = list(size = 14)
)

fig_Pois_res <- ggpubr::annotate_figure(
  fig_Pois_res,
  left = grid::textGrob("Rt estimates for Poisson incidences",
                        rot = 90, vjust = 1, gp = grid::gpar(cex = 1)))

ggsave(here::here("fig/Pois-res-plot.png"), fig_Pois_res, width = 5.71, height = 5.41)

```

Display results of negative Binomial incidence.

```

fig21 <- res_dat %>%
  select(time, Rt1, NB_Epi1, NB_Rtestim1, NB_EpiLPS1) %>%
  pivot_longer(!time, names_to = "Rt_type", values_to = "Rt_value") %>%
  mutate(Rt_type = fct_recode(Rt_type, "EpiEstim" = "NB_Epi1",
                              "RtEstim" = "NB_Rtestim1",
                              "EpiLPS" = "NB_EpiLPS1",
                              "True Rt" = "Rt1")) %>%
  mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                              "RtEstim",
                              "EpiEstim",
                              "EpiLPS"
                              )) %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +

```

```

geom_line() +
scale_colour_manual(values = cbPalette) +
labs(y="", color="Rt methods", x="", title="Scenario 1") +
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") %>%
  mutate(Rt_type = fct_recode(Rt_type, "EpiEstim" = "NB_Epi2",
                              "RtEstim" = "NB_Rtestim2",
                              "EpiLPS" = "NB_EpiLPS2",
                              "True Rt" = "Rt2")) %>%
  mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                              "RtEstim",
                              "EpiEstim",
                              "EpiLPS"
                              )) %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="", color="Rt methods", x="", title="Scenario 2") +
  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") %>%
  mutate(Rt_type = fct_recode(Rt_type, "EpiEstim" = "NB_Epi3",
                              "RtEstim" = "NB_Rtestim3",
                              "EpiLPS" = "NB_EpiLPS3",
                              "True Rt" = "Rt3")) %>%
  mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                              "RtEstim",
                              "EpiEstim",
                              "EpiLPS"
                              )) %>%
  group_by(Rt_type) %>%
  ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
  geom_line() +
  scale_colour_manual(values = cbPalette) +
  labs(y="", color="Rt methods", x="", title="Scenario 3") +
  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") %>%
  mutate(Rt_type = fct_recode(Rt_type, "EpiEstim" = "NB_Epi4",
                              "RtEstim" = "NB_Rtestim4",
                              "EpiLPS" = "NB_EpiLPS4",
                              "True Rt" = "Rt4")) %>%
  mutate(Rt_type = fct_relevel(Rt_type, "True Rt" ,
                              "RtEstim",
                              "EpiEstim",

```

```

                                "EpiLPS"
                                )) %>%

group_by(Rt_type) %>%
ggplot(aes(x = time, y = Rt_value, col=Rt_type)) +
geom_line() +
scale_colour_manual(values = cbPalette) +
labs(y="", color="Rt methods", x="", title="Scenario 4") +
theme_bw()

#library(cowplot)
#nb_res_plot <- plot_grid(fig21, fig22, fig23, fig24)
fig_NB_res <- ggarrange(fig21, fig22, fig23, fig24, ncol=2, nrow=2,
                        common.legend = TRUE, legend = "bottom",
                        font.label = list(size = 14))
fig_NB_res <- ggpubr::annotate_figure(
  fig_NB_res,
  left = grid::textGrob("Rt estimates for negative Binomial incidences",
                        rot = 90, vjust = 1, gp = grid::gpar(cex = 1)))

ggsave(here::here("fig/NB-res-plot.png"), fig_NB_res, width = 5.71, height = 5.41)

```

Save graphical display of estimates across all Rt scenarios for both incidence distributional assumptions.

```

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)

```

Save graphical display of all Rt and incidence samples across both incidence distributional assumptions.

```

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 1.015113 2.500000 1.200000          2          2          2
## 2:      2  2 1.030455 2.491525 1.271725          8          5         10
## 3:      3  2 1.046028 2.483051 1.343015          4          1          5
## 4:      4  2 1.061837 2.474576 1.413436          8          3         10
## 5:      5  2 1.077884 2.466102 1.482563         11          4         16
## 6:      6  2 1.094174 2.457627 1.549978         13          5         15
##      Pois_count4 NB_count1 NB_count2 NB_count3 NB_count4 Pois_Epi1 Pois_Epi2
## 1:              2          4          4          4          4 7.059698 3.224088
## 2:              6         10          6         12          7 7.059698 3.224088
## 3:              2         12          6         14          8 7.059698 3.224088
## 4:              4          4          1          9          2 7.059698 3.224088
## 5:              6          6          3          8          4 7.059698 3.224088
## 6:              8         10          3         10          6 7.059698 3.224088
##      Pois_Epi3 Pois_Epi4 Pois_Rtestim1 Pois_Rtestim2 Pois_Rtestim3 Pois_Rtestim4
## 1: 12.11352 8.985446      1.876867      1.020317      2.588113      1.337800
## 2: 12.11352 8.985446      1.876867      1.045198      2.574192      1.420209
## 3: 12.11352 8.985446      1.876867      1.069970      2.560348      1.500582
## 4: 12.11352 8.985446      1.876867      1.094601      2.546581      1.578264
## 5: 12.11352 8.985446      1.876867      1.119059      2.532891      1.652640
## 6: 12.11352 8.985446      1.876867      1.143310      2.519275      1.723146
##      Pois_EpiLPS1 Pois_EpiLPS2 Pois_EpiLPS3 Pois_EpiLPS4 NB_Epi1 NB_Epi2
## 1:      5.956857      2.577492      7.211589      3.594243 5.803456 1.918394
## 2:      4.578125      4.161077      4.606643      4.435372 5.803456 1.918394
## 3:      3.157738      2.772014      3.206680      3.054421 5.803456 1.918394
## 4:      2.652835      2.266517      2.709453      2.558458 5.803456 1.918394
## 5:      2.389542      1.996772      2.449661      2.297065 5.803456 1.918394
## 6:      2.226892      1.826883      2.288177      2.133985 5.803456 1.918394
##      NB_Epi3 NB_Epi4 NB_Rtestim1 NB_Rtestim2 NB_Rtestim3 NB_Rtestim4 NB_EpiLPS1
## 1: 4.427581 2.474002      1.794514      0.6593982      2.301004      1.161091      6.703414
## 2: 4.427581 2.474002      1.794514      0.6902454      2.286578      1.181212      4.418239
## 3: 4.427581 2.474002      1.794514      0.7217602      2.272249      1.200461      3.015958
## 4: 4.427581 2.474002      1.794514      0.7539066      2.258016      1.218800      2.512148
## 5: 4.427581 2.474002      1.794514      0.7866459      2.243880      1.236198      2.246999
## 6: 4.427581 2.474002      1.794514      0.8199357      2.229840      1.252622      2.082266
##      NB_EpiLPS2 NB_EpiLPS3 NB_EpiLPS4
## 1:      2.831848      8.974085      4.176513
## 2:      3.915898      4.545046      4.188413

```

```
## 3: 2.557328 3.113151 2.797712
## 4: 2.054081 2.607131 2.293761
## 5: 1.780714 2.344918 2.026428
## 6: 1.605661 2.183943 1.859368
```

```
save_example(res_dat)
```

Real case

Covid-19 Canada

Get latest Covid-19 incidence data in Canada.

```
library(CanCovidData)
dat <- get_british_columbia_case_data() %>%
  rename(HA = `Health Authority`, Date = `Reported Date`,
         Age = `Age group`) %>%
  filter(Date >= ymd("2020-03-01"))

cases <- dat %>%
  count(Date, name = "Cases")

time_diff <- difftime(cases$Date, min(cases$Date), units = "days")
natural_numbers <- as.numeric(time_diff) + 1

covid_cases <- data.table(cases, x = natural_numbers)
covid_dat <- covid_cases %>%
  ggplot(aes(x=Date)) +
  geom_line(aes(y=Cases), col="black") +
  labs(x="Date", y='Observed counts') +
  scale_x_date(date_breaks = "6 month") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(hjust = 0.5),
        panel.spacing = unit(0.8, "lines"),
        text = element_text(size = 12))
ggsave(here::here("fig/covid_dat.png"), covid_dat, width = 6.67, height = 4.06)

n <- length(covid_cases$Cases)
```

Estimate Rt using RtEstim.

```
# rtestim
mod_rt_covid <- estimate_rt(x=cases$Date, covid_cases$Cases, korder = 1, nsol = 50,
                          maxiter=1e7L)
covid_fig1 <- plot(mod_rt_covid) + labs(y="", x="Date")
cv_mod_covid <- cv_estimate_rt(x=cases$Date, covid_cases$Cases, korder = 1, nfold = 3,
                              maxiter=1e7L, nsol = 50)
mod_rtestim_covid <- cv_mod_covid$full_fit$Rt[,which.min(cv_mod_covid$cv_scores)]
rt_ci_covid <- confband(cv_mod_covid, "lambda.min") # get 95% confidence band
covid_fig_cv1 <- rt_ci_covid %>%
  ggplot(aes(x = covid_cases$Date)) +
  geom_line(aes(y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
            fill = "gray", alpha = 0.5) +
```

```

geom_hline(yintercept = 1, linetype = "dotted") +
labs(x="Date", y = "Piecewise linear Rt") +
theme_bw()

# k=2
mod_rt_covid <- estimate_rt(x=cases$Date, covid_cases$Cases, korder = 2, nsol = 50,
                           maxiter=1e7L)
covid_fig2 <- plot(mod_rt_covid) + labs(y="", x="Date")
cv_mod_covid <- cv_estimate_rt(x=cases$Date, covid_cases$Cases, korder = 2, nfold = 3,
                              maxiter=1e7L, nsol = 50)
mod_rtestim_covid <- cv_mod_covid$full_fit$Rt[,which.min(cv_mod_covid$cv_scores)]
rt_ci_covid <- confband(cv_mod_covid, "lambda.min") # get 95% confidence band
covid_fig_cv2 <- rt_ci_covid %>%
  ggplot(aes(x = covid_cases$Date)) +
  geom_line(aes(y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
            fill = "gray", alpha = 0.5) +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Date", y = "Piecewise quadratic Rt") +
  theme_bw()

# k=3
mod_rt_covid <- estimate_rt(x=cases$Date, covid_cases$Cases, korder = 3, nsol = 50,
                           maxiter=1e7L)
covid_fig3 <- plot(mod_rt_covid) + labs(y="", x="Date")
cv_mod_covid <- cv_estimate_rt(x=cases$Date, covid_cases$Cases, korder = 3, nfold = 3,
                              maxiter=1e7L, nsol = 50)
mod_rtestim_covid <- cv_mod_covid$full_fit$Rt[,which.min(cv_mod_covid$cv_scores)]
rt_ci_covid <- confband(cv_mod_covid, "lambda.min") # get 95% confidence band
covid_fig_cv3 <- rt_ci_covid %>%
  ggplot(aes(x = covid_cases$Date)) +
  geom_line(aes(y = Rt)) +
  geom_ribbon(aes(ymin = `2.5%`, ymax = `97.5%`),
            fill = "gray", alpha = 0.5) +
  geom_hline(yintercept = 1, linetype = "dotted") +
  labs(x="Date", y = "Piecewise cubic Rt") +
  theme_bw()

covid_all <- ggarrange(covid_fig_cv1, covid_fig1, covid_fig_cv2, covid_fig2,
                      covid_fig_cv3, covid_fig3, ncol=2, nrow=3,
                      common.legend = TRUE, legend = "bottom")

ggsave(here::here("fig/covid_full_res.png"), covid_all, width = 6.91, height = 6.17)

```

1918 H1N1 influenza in the USA

Obtain 1918 H1N1 influenza data in the US from EpiEtim package.

```

# grab the flu data from EpiEstim package
data("Flu1918")
n <- length(Flu1918$incidence)
flu_dat <- data.frame(Flu_incidence = Flu1918$incidence, Time = 1:n)
flu_fig <- flu_dat %>%
  ggplot(aes(y=Flu_incidence, x=Time)) +

```



```

  geom_line() +
  theme_bw()
ggsave(here::here("fig/flu_dat.png"), flu_fig, width = 5.18, height = 3.34)

mod_rt <- estimate_rt(Flu1918$incidence, korder = 3, nsol = 50)
fig_flu3 <- plot(mod_rt) + labs(y="", x="Date")
cv_mod_flu <- cv_estimate_rt(Flu1918$incidence, korder = 3,
                             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.1se") # get 95% confidence band
fig_flu_cv3 <- 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="Date", y = "Piecewise cubic Rt") +
  theme_bw()

# k=2
mod_rt <- estimate_rt(Flu1918$incidence, korder = 2, nsol = 50)
fig_flu2 <- plot(mod_rt) + labs(y="", x="Date")
cv_mod_flu <- cv_estimate_rt(Flu1918$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.1se") # get 95% confidence band
fig_flu_cv2 <- 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="Date", y = "Piecewise quadratic Rt") +
  theme_bw()

mod_rt <- estimate_rt(Flu1918$incidence, korder = 1, nsol = 50)
fig_flu1 <- plot(mod_rt) + labs(y="", x="Date")
cv_mod_flu <- cv_estimate_rt(Flu1918$incidence, korder = 1,
                             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.1se") # get 95% confidence band
fig_flu_cv1 <- 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="Date", y = "Piecewise linear Rt") +
  theme_bw()

flu_all <- ggarrange(fig_flu_cv1, fig_flu1, fig_flu_cv2, fig_flu2,

```

```
fig_flu_cv3, fig_flu3, ncol=2, nrow=3,  
common.legend = TRUE, legend = "bottom")  
ggsave(here::here("fig/flu_full_res.png"), flu_all, width = 6.91, height = 6.17)
```