SIR_example.r

luiz

2019-06-24

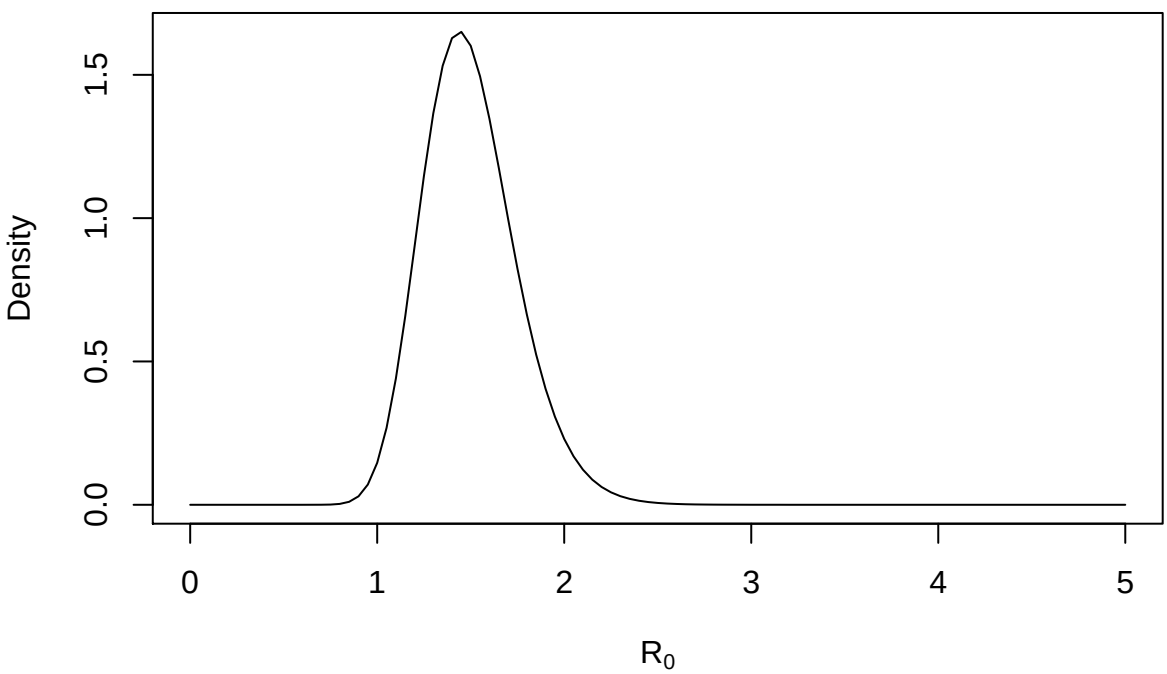
```r
library(outbreaks)
data(influenza_england_1978_school)
Ndata <- 763
sol <- influenza_england_1978_school
sol$time <- as.numeric(sol$date-min(sol$date)) + 2
sol$I <- sol$in_bed
forfit.sol <- sol
noisy_I <- forfit.sol$I/Ndata
iniTime <- 0
iniI <- 1/Ndata

#### Elicitation
devtools::source_url("https://raw.githubusercontent.com/maxbiostat/CODE/master/R/DISTRIBUTIONS/lognormal
```

```
## SHA-1 hash of file is c164d906f2e61bb2235d940b1621e771d53a224c
```

```r
muR <- LogMean(realMean = 1.5, realSD = .25)
sdR <- sqrt(LogVar(realMean = 1.5, realSD = .25))
qlnorm(c(.025, .975), meanlog = muR, sdlog = sdR)
```

```
## [1] 1.069659 2.046623
```

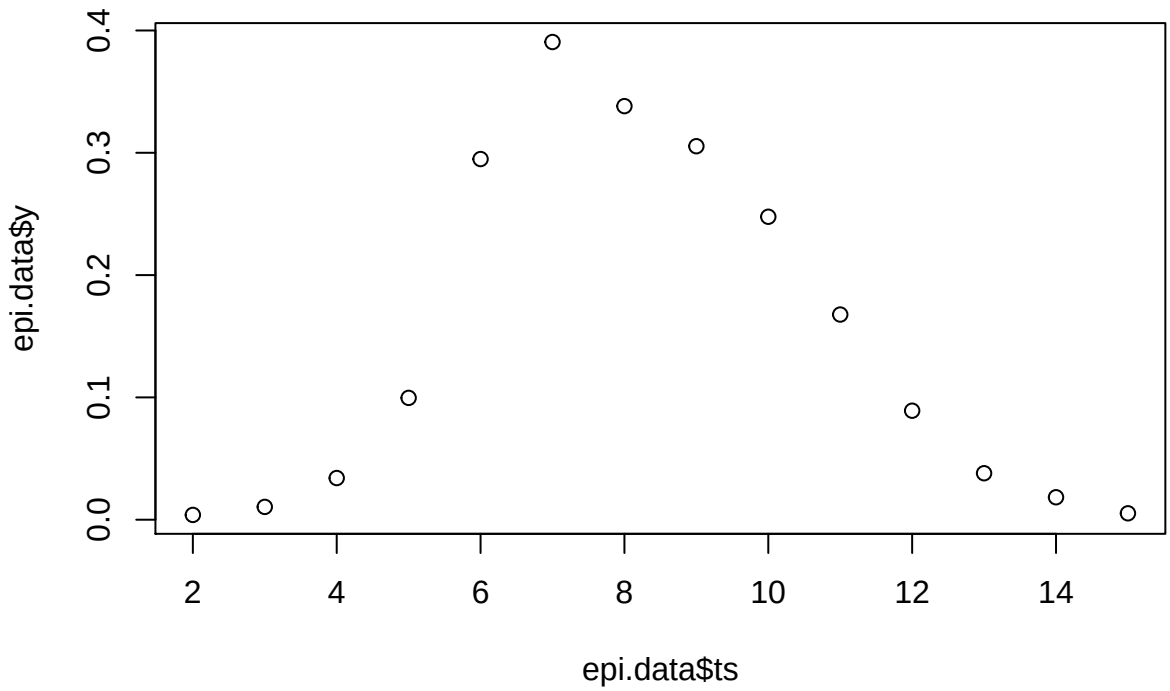
```r
curve(dlnorm(x, meanlog = muR, sdlog = sdR), 0, 5, xlab = expression(R[0]), ylab = "Density")
```




```r
epi.data <- list(
  n_obs = length(noisy_I),
  t0 = iniTime,
```

```r
  ts = forfit.sol$time,
  y_init = iniI,
  y = noisy_I,
  mu_beta = 0,
  sigma_beta = 1,
  mu_gamma = 0,
  sigma_gamma = 1,
  mu_r0 = muR,
  sigma_r0 = sdR,
  as = 99,
  bs = 1,
  a_alpha = 1,
  b_alpha = 1
)
plot(epi.data$ts, epi.data$y)

#### Inference

library(rstan)
```

```
## Loading required package: StanHeaders

## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang

## rstan (Version 2.19.1, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```




```r
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

SIR_code_varying <- stan_model(file = "stan/sir_simple_varying_alpha.stan")
SIR_code_fixed <- stan_model(file = "stan/sir_simple_fixed_alpha.stan")

### Varying alpha
SIR.map.varying <- optimizing(SIR_code_varying, data = epi.data, hessian = TRUE, verbose = TRUE)
```

```
## Chain 1: Rejecting initial value:
## Chain 1:   Error evaluating the log probability at the initial value.
## Chain 1: Exception: lognormal_lpdf: Location parameter[8] is nan, but must be finite!  (in 'model70c
##
## Chain 1: Rejecting initial value:
## Chain 1:   Error evaluating the log probability at the initial value.
## Chain 1: Exception: lognormal_lpdf: Location parameter[7] is nan, but must be finite!  (in 'model70c
##
## Chain 1: Initial log joint probability = -13413.7
## Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
## Chain 1:       19       6.92672     0.0245908    0.00708765           1           1       20
## Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
## Chain 1:       36       6.92718      0.229325   6.28064e-05           1           1       37
## Chain 1: Optimization terminated normally:
## Chain 1:   Convergence detected: relative gradient magnitude is below tolerance
```

```r
SIR.posterior.varying <- sampling(SIR_code_varying, data = epi.data, chains = 4, control = list(adapt_d
check_hmc_diagnostics(SIR.posterior.varying)
```

```
##
## Divergences:

## 0 of 4000 iterations ended with a divergence.

##
```

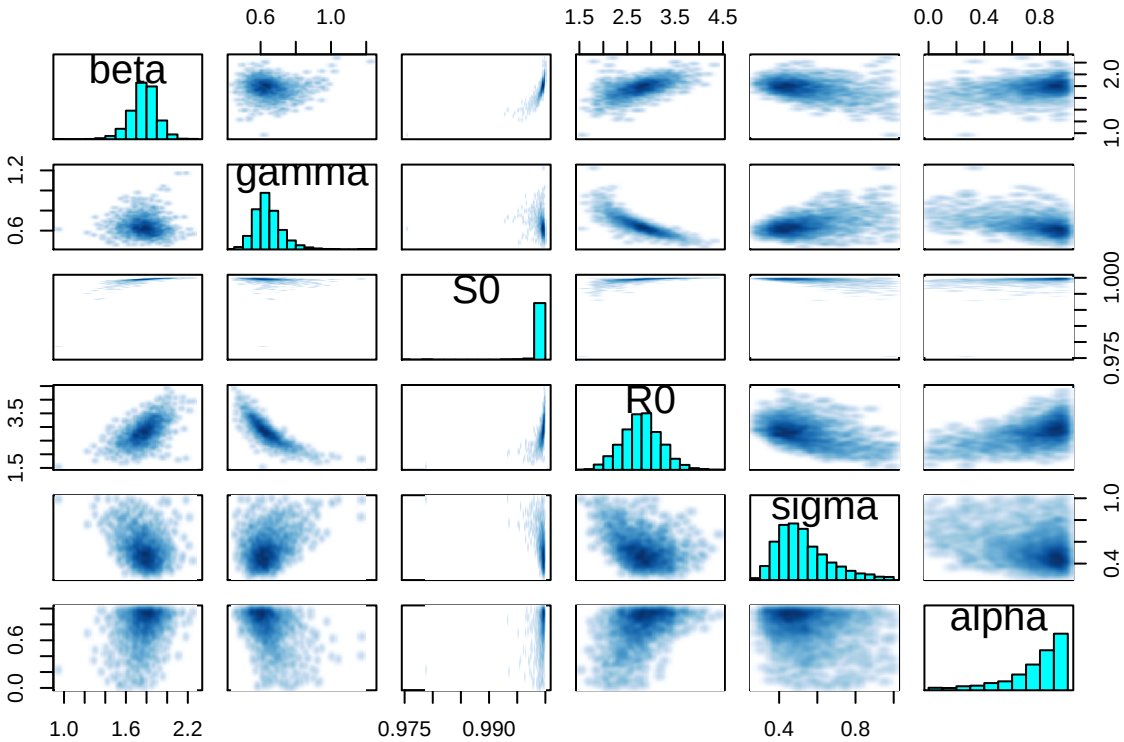
```
## Tree depth:

## 0 of 4000 iterations saturated the maximum tree depth of 10.

##
## Energy:

## E-BFMI indicated no pathological behavior.
```

```r
print(SIR.posterior.varying, pars = c("beta", "gamma", "S0", "R0", "sigma", "alpha"))
```

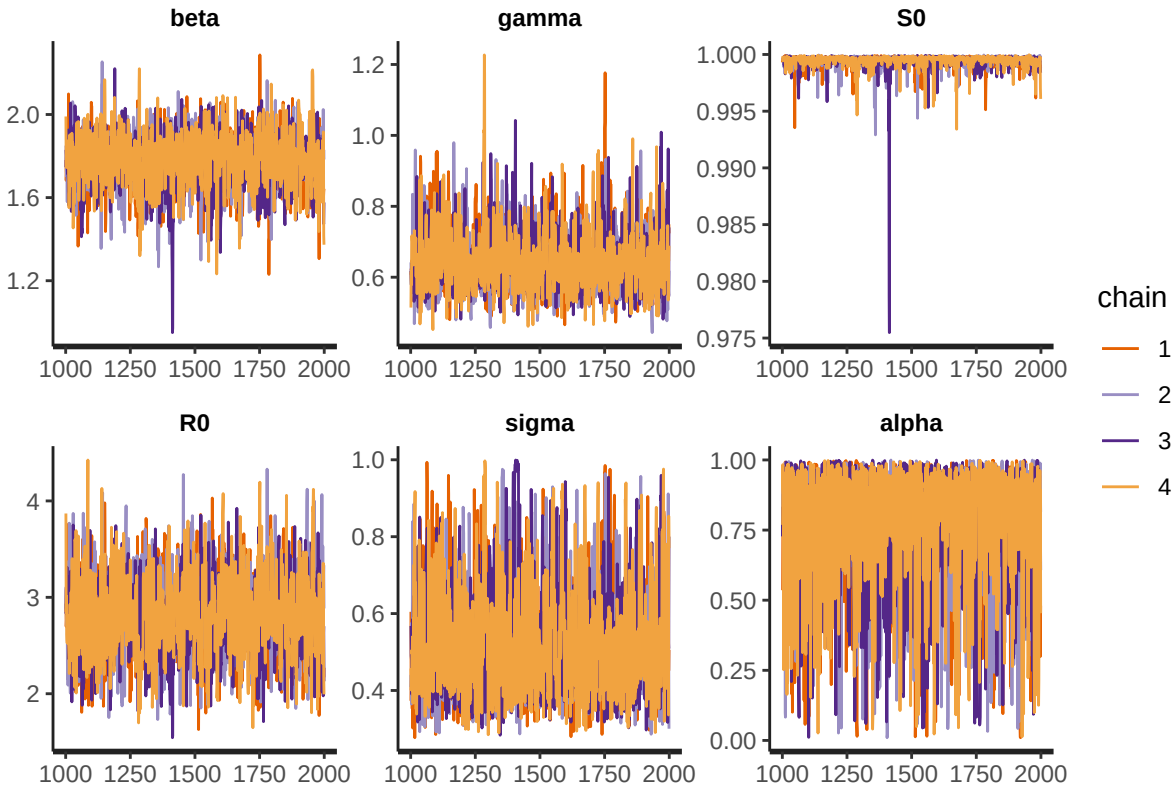
```
## Inference for Stan model: sir_simple_varying_alpha.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##       mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
## beta  1.77    0.00 0.13 1.50 1.70 1.78 1.85  2.01  1598    1
## gamma 0.64    0.00 0.08 0.51 0.59 0.63 0.68  0.84  1923    1
## S0    1.00    0.00 0.00 1.00 1.00 1.00 1.00  1.00  1201    1
## R0    2.80    0.01 0.40 2.01 2.53 2.80 3.05  3.63  1479    1
## sigma 0.53    0.00 0.14 0.33 0.43 0.50 0.60  0.88   894    1
## alpha 0.77    0.00 0.21 0.18 0.68 0.84 0.93  0.99  2241    1
##
## Samples were drawn using NUTS(diag_e) at Mon Jun 24 14:44:39 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```r
pairs(SIR.posterior.varying, pars = c("beta", "gamma", "S0", "R0", "sigma", "alpha"))
```

```
## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, bandwidth = bandwidth, gridsize = nbin, :
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'
```




```
stan_trace(SIR.posterior.varying, pars = c("beta", "gamma", "S0", "R0", "sigma", "alpha"))
```




```
simulated_trajectories.varying <- extract(SIR.posterior.varying, 'y_rep')$y_rep
predicted.varying <- data.frame(
  time = epi.data$ts,
```

```r
  lower = apply(simulated_trajectories.varying, 2, function(x) as.numeric(quantile(x, probs = .025))),
  post_mean = colMeans(simulated_trajectories.varying),
  upper = apply(simulated_trajectories.varying, 2, function(x) as.numeric(quantile(x, probs = .975))),
  alpha = "varying"
)

### Fixed alpha = 1/2
epi.data$alpha <- 1/2
SIR.map.fixedHalf <- optimizing(SIR_code_fixed, data = epi.data, hessian = TRUE, verbose = TRUE)
```

```
## Chain 1: Initial log joint probability = -85.7879
## Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
## Chain 1: Exception: lognormal_lpdf: Location parameter[1] is nan, but must be finite!  (in 'model70c
##
## Exception: lognormal_lpdf: Location parameter[10] is nan, but must be finite!  (in 'model70cf1edd4aa
##
##
## Chain 1:       19        25.289     0.0301171      0.203989           1           1       28
## Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
## Chain 1:       37       30.9551   0.000150581     0.0013622           1           1       68
## Chain 1: Optimization terminated normally:
## Chain 1:   Convergence detected: relative gradient magnitude is below tolerance
```

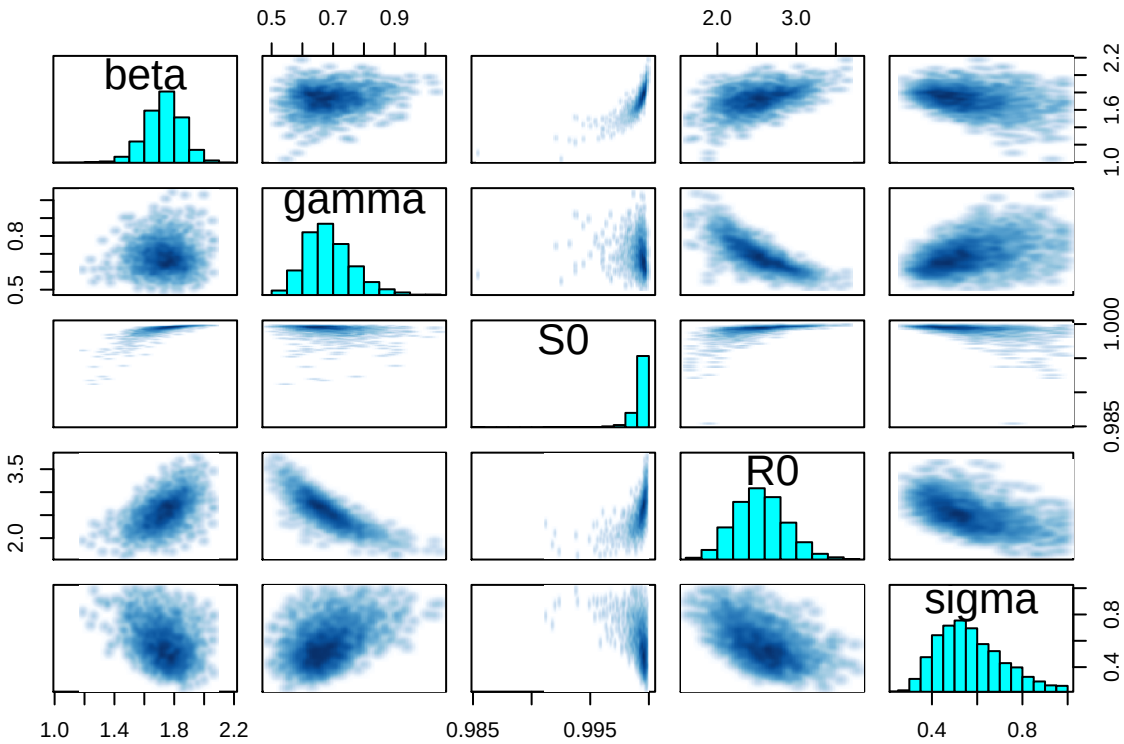
```r
SIR.posterior.fixedHalf <- sampling(SIR_code_fixed, data = epi.data, chains = 4, control = list(adapt_d
check_hmc_diagnostics(SIR.posterior.fixedHalf)
```

```
##
## Divergences:

## 0 of 4000 iterations ended with a divergence.

##
## Tree depth:

## 0 of 4000 iterations saturated the maximum tree depth of 10.

##
## Energy:

## E-BFMI indicated no pathological behavior.
```

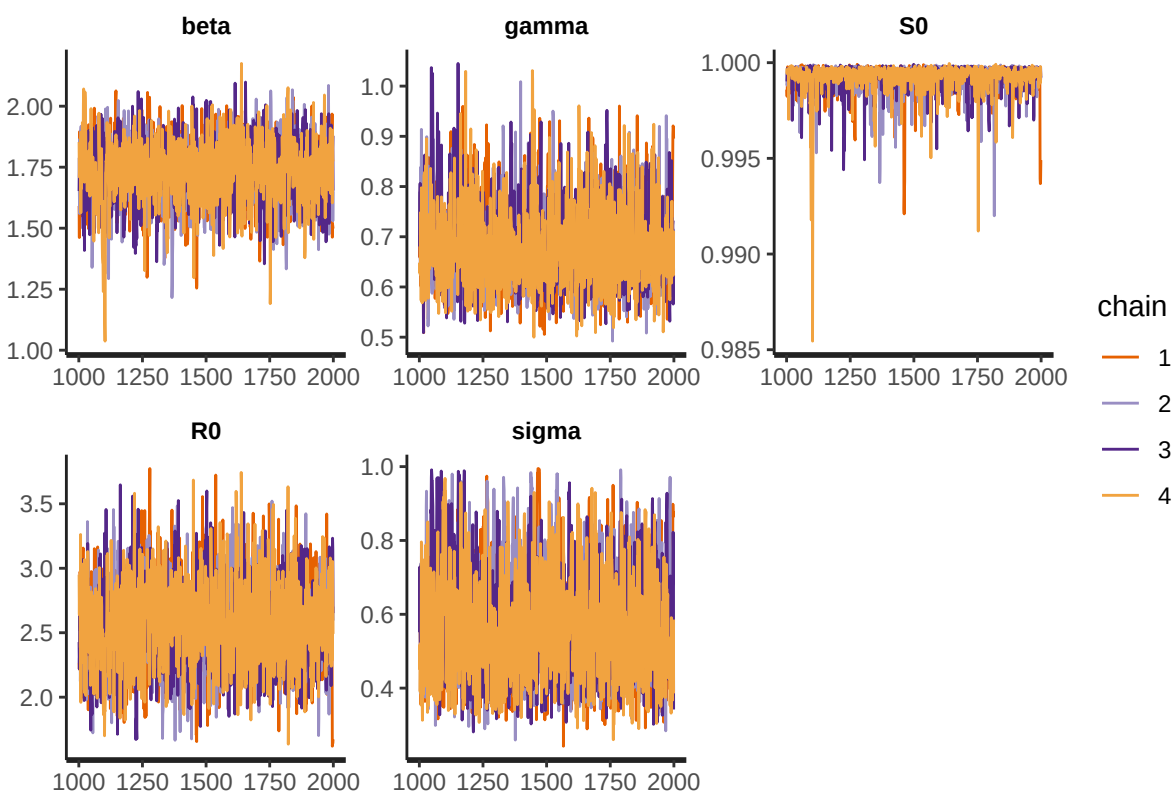
```r
print(SIR.posterior.fixedHalf, pars = c("beta", "gamma", "S0", "R0", "sigma"))
```

```
## Inference for Stan model: sir_simple_fixed_alpha.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##       mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
## beta  1.73    0.00 0.12 1.47 1.65 1.73 1.81  1.96  1543    1
## gamma 0.69    0.00 0.08 0.56 0.63 0.68 0.73  0.87  1958    1
## S0    1.00    0.00 0.00 1.00 1.00 1.00 1.00  1.00  1091    1
## R0    2.55    0.01 0.33 1.96 2.32 2.54 2.76  3.22  1683    1
## sigma 0.57    0.00 0.14 0.35 0.46 0.55 0.66  0.90  1255    1
##
## Samples were drawn using NUTS(diag_e) at Mon Jun 24 14:45:38 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```r
pairs(SIR.posterior.fixedHalf, pars = c("beta", "gamma", "S0", "R0", "sigma"))
```




```r
stan_trace(SIR.posterior.fixedHalf, pars = c("beta", "gamma", "S0", "R0", "sigma"))
```



```r
simulated_trajectories.fixedHalf <- extract(SIR.posterior.fixedHalf, 'y_rep')$y_rep
predicted.fixedHalf <- data.frame(
  time = epi.data$ts,
  lower = apply(simulated_trajectories.fixedHalf, 2, function(x) as.numeric(quantile(x, probs = .025)))
  post_mean = colMeans(simulated_trajectories.fixedHalf),
  upper = apply(simulated_trajectories.fixedHalf, 2, function(x) as.numeric(quantile(x, probs = .975)))
  alpha = "0.5"
)

### Fixed alpha = 1
epi.data$alpha <- 1
SIR.map.fixedOne <- optimizing(SIR_code_fixed, data = epi.data, hessian = TRUE, verbose = TRUE)
```

```
## Chain 1: Initial log joint probability = -468.725
## Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
## Chain 1: Exception: Max number of iterations exceeded (1000000).  (in 'model70cf1edd4aa1_sir_simple_
##
## Exception: lognormal_lpdf: Location parameter[1] is nan, but must be finite!  (in 'model70cf1edd4aa1
##
## Exception: lognormal_lpdf: Location parameter[4] is nan, but must be finite!  (in 'model70cf1edd4aa1
##
## Exception: lognormal_lpdf: Location parameter[9] is nan, but must be finite!  (in 'model70cf1edd4aa1
##
##
## Chain 1:       19       22.8228      0.658234       15.0499           1           1       32
## Chain 1:     Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
## Chain 1:       27       25.5814   0.000735447   0.000129986           1           1       41
## Chain 1: Optimization terminated normally:
## Chain 1:   Convergence detected: relative gradient magnitude is below tolerance
```

```r
SIR.posterior.fixedOne <- sampling(SIR_code_fixed, data = epi.data, chains = 4, control = list(adapt_de
check_hmc_diagnostics(SIR.posterior.fixedOne)
```

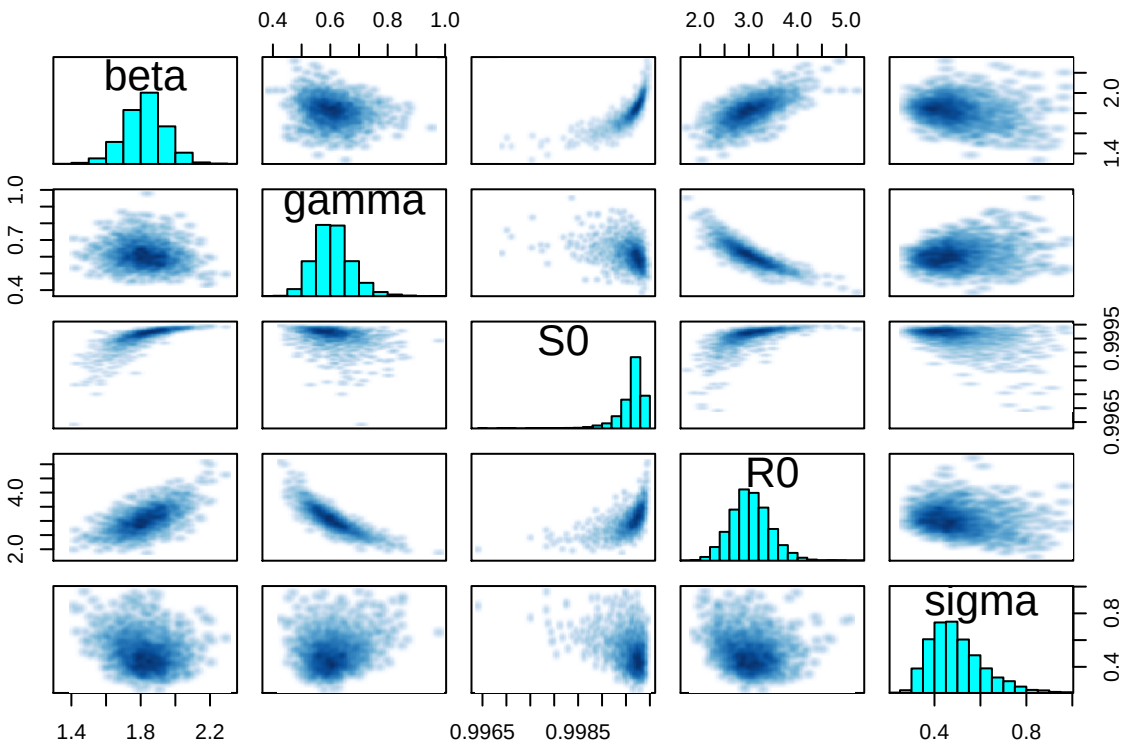
```
##
## Divergences:

## 0 of 4000 iterations ended with a divergence.

##
## Tree depth:

## 0 of 4000 iterations saturated the maximum tree depth of 10.

##
## Energy:

## E-BFMI indicated no pathological behavior.
```

```r
print(SIR.posterior.fixedOne, pars = c("beta", "gamma", "S0", "R0", "sigma"))
```

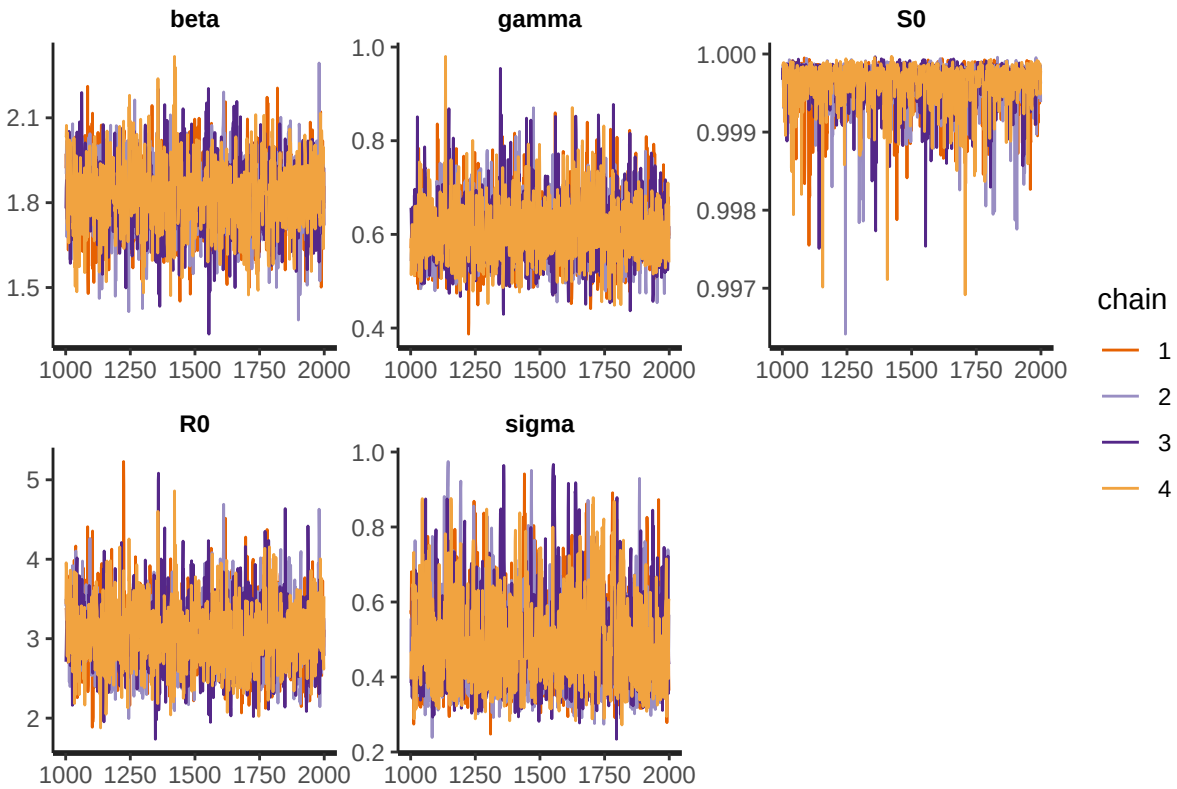
```
## Inference for Stan model: sir_simple_fixed_alpha.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##       mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
## beta  1.82    0.00 0.12 1.58 1.75 1.83 1.90  2.05  1986    1
## gamma 0.61    0.00 0.07 0.50 0.56 0.60 0.65  0.76  2332    1
## S0    1.00    0.00 0.00 1.00 1.00 1.00 1.00  1.00  1808    1
```

```
## R0    3.03    0.01 0.41 2.30 2.77 3.02 3.29  3.90  1918    1
## sigma 0.50    0.00 0.12 0.32 0.41 0.48 0.56  0.78  1376    1
##
## Samples were drawn using NUTS(diag_e) at Mon Jun 24 14:46:23 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

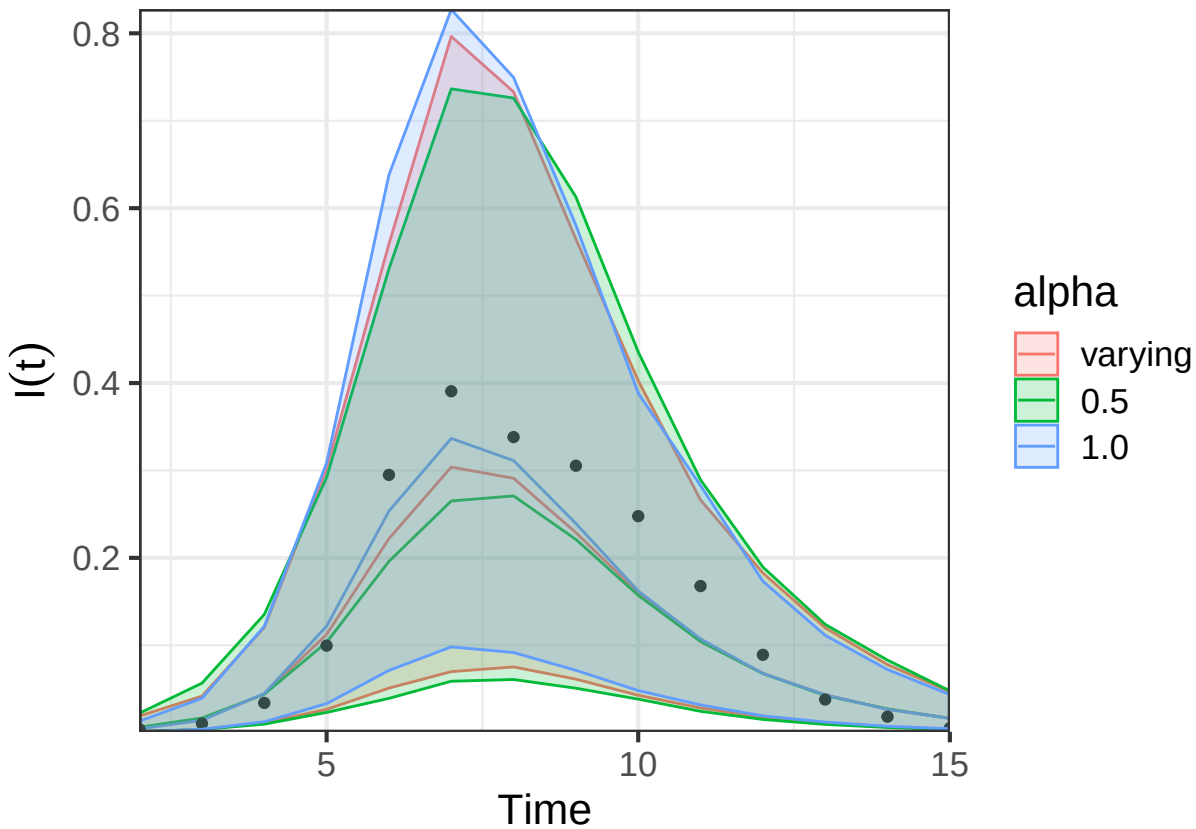
```
pairs(SIR.posterior.fixedOne, pars = c("beta", "gamma", "S0", "R0", "sigma"))
```




```
stan_trace(SIR.posterior.fixedOne, pars = c("beta", "gamma", "S0", "R0", "sigma"))
```



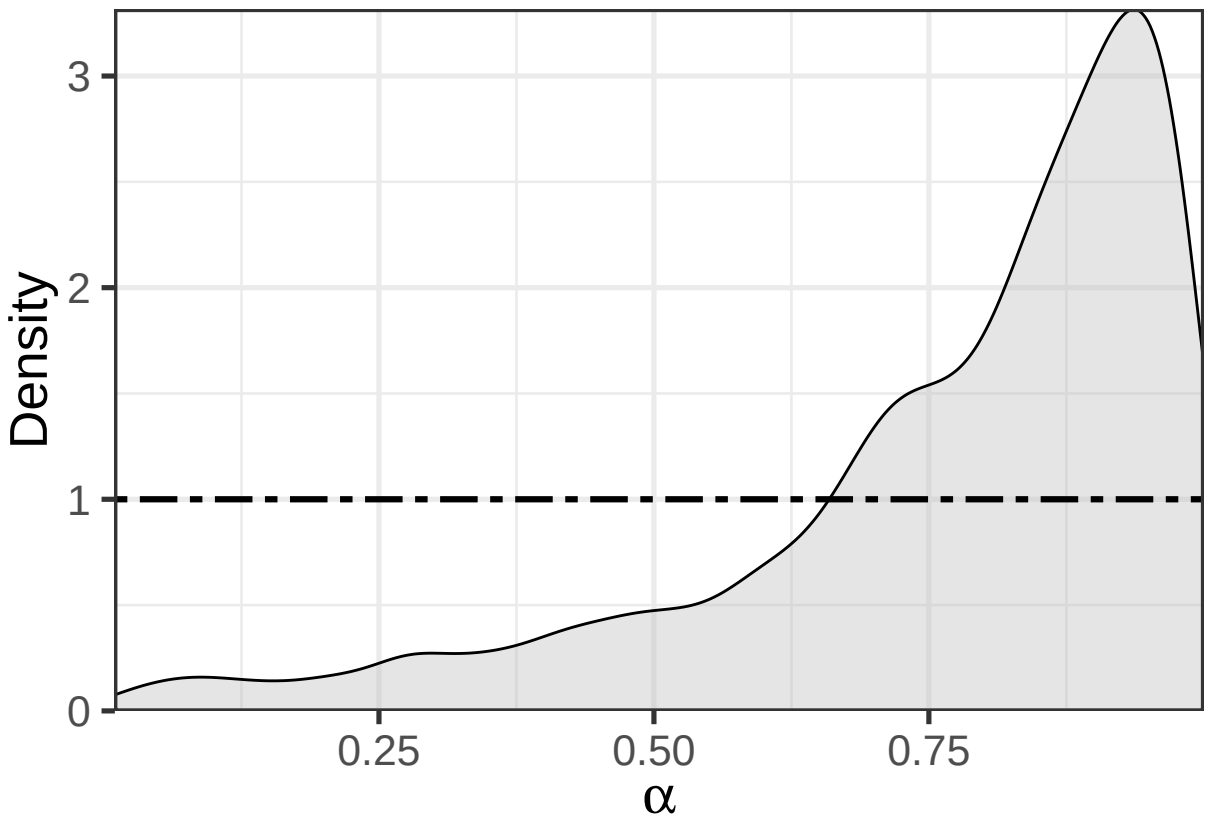

```
simulated_trajectories.fixedOne <- extract(SIR.posterior.fixedOne, 'y_rep')$y_rep
predicted.fixedOne <- data.frame(
  time = epi.data$ts,
  lower = apply(simulated_trajectories.fixedOne, 2, function(x) as.numeric(quantile(x, probs = .025))),
  post_mean = colMeans(simulated_trajectories.fixedOne),
  upper = apply(simulated_trajectories.fixedOne, 2, function(x) as.numeric(quantile(x, probs = .975))),
  alpha = "1.0"
)

#### Plotting and annotating

prediction.bands.SIR <- do.call(rbind, list(predicted.varying, predicted.fixedHalf, predicted.fixedOne)

library(ggplot2)

predictions_SIR <- ggplot(data = prediction.bands.SIR, aes(x = time, y = post_mean, colour = alpha, fil
  geom_line() +
  geom_point(data = data.frame(time = epi.data$ts, I = epi.data$y), aes(x = time, y = I), inherit.aes =
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = .2) +
  scale_x_continuous("Time", expand = c(0, 0)) +
  scale_y_continuous(expression(I(t)), expand = c(0, 0)) +
  theme_bw(base_size = 16)

predictions_SIR
```




```
ggsave(plot = predictions_SIR, file = "../plots/SIR_bands.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

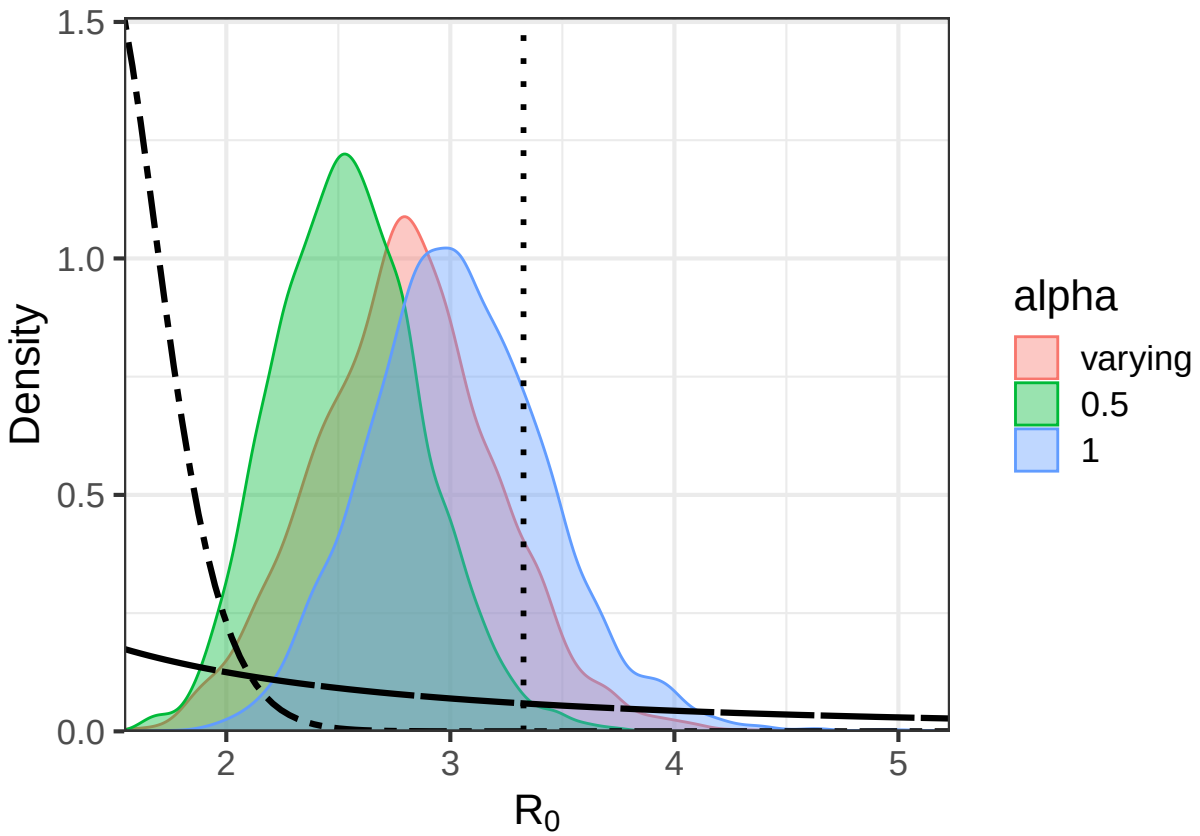
```
alpha.posterior <- data.frame(alpha = extract(SIR.posterior.varying, 'alpha')$alpha)

alpha_post_plot <- ggplot(data = alpha.posterior, aes (x = alpha)) +
  geom_density(alpha = .4, fill = "grey") +
  stat_function(fun = function(x) dbeta(x, epi.data$a_alpha, epi.data$b_alpha), linetype = "twodash", s
  scale_y_continuous("Density", expand = c(0, 0)) +
  scale_x_continuous(expression(alpha), expand = c(0, 0)) +
  theme_bw(base_size = 20)
alpha_post_plot
```




```
ggsave(plot = alpha_post_plot, filename = "../plots/posterior_alpha_SIR_example.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
R0.varying <- data.frame(R0 = extract(SIR.posterior.varying, 'R0')$R0, alpha = "varying")
R0.fixedHalf <- data.frame(R0 = extract(SIR.posterior.fixedHalf, 'R0')$R0, alpha = "0.5")
R0.fixedOne <- data.frame(R0 = extract(SIR.posterior.fixedOne, 'R0')$R0, alpha = "1")

R0.posteriors <- do.call(rbind, list(R0.varying, R0.fixedHalf, R0.fixedOne))

R0_posterior <- ggplot(data = R0.posteriors, aes(x = R0, colour = alpha, fill = alpha)) +
  geom_density(alpha = .4) +
  geom_vline(xintercept = 0.00218/.5 * Ndata, linetype = "dotted", size = 1.01) +
  stat_function(fun = function(x) dlnorm(x, meanlog = muR, sdlog = sdR), inherit.aes = FALSE, linetype =
  stat_function(fun = function(x) dlnorm(x, meanlog = 0, sdlog = sqrt(epi.data$sigma_beta^2 + epi.data$s
                inherit.aes = FALSE, linetype = "F1", size = 1.10) +
  scale_x_continuous(expression(R[0]), expand = c(0, 0)) +
  scale_y_continuous("Density", expand = c(0, 0)) +
  theme_bw(base_size = 16)
R0_posterior
```




```
ggsave(plot = R0_posterior, file = "../plots/R0_posteriors_SIR_boardingSchool.pdf")
```

```
## Saving 6.5 x 4.5 in image
```