Modello Regione Lombardia con variabile apertira scuole

Laura Balasso

11/6/2020

Model for Lombardia from august

```
data_lombardia <- get_model_data(data_it, 'Lombardia', initial_date = as.Date('2020-07-30'))

### school effect

school_opening <- as.Date('2020-09-14')
school <- rep(0, length(data_lombardia$date))
school[which(data_lombardia$date > school_opening +10)] <- 1
grow_s <- which(data_lombardia$date>=school_opening &data_lombardia$date <= school_opening +10)
school[grow_s] <- (grow_s - which(data_lombardia$date ==school_opening))^2 /100

p_delay <- get_delay_distribution()

nonzero_days_l <- which(data_lombardia$total != 0)

stan_data_l <- list(N = nrow(data_lombardia),
                    conv_gt = get_gt_convolution(nrow(data_lombardia)),
                    length_delay = length(p_delay),
                    p_delay = p_delay,
                    exposures = exposures_from_total(data_lombardia$total),
                    N_nonzero = length(nonzero_days_l),
                    nonzero_positives = data_lombardia$positive[nonzero_days_l],
                    nonzero_days = nonzero_days_l,
                    school = school[nonzero_days_l]
)


compiled_model <- stan_model('rt_model_schools.stan')
```

```
## Trying to compile a simple C file
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
```

```
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: fa
## #include <complex>
##          ^~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
```

```
fit_model_lomb <- sampling(compiled_model, data=stan_data_l, iter = 2000, cores=getOption("mc.cores", 1L
```

```
##
## SAMPLING FOR MODEL 'rt_model_schools' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.002648 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 26.48 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 88.0476 seconds (Warm-up)
## Chain 1:                79.1347 seconds (Sampling)
## Chain 1:                167.182 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'rt_model_schools' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000807 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 8.07 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 94.7919 seconds (Warm-up)
## Chain 2:                76.9197 seconds (Sampling)
## Chain 2:                171.712 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'rt_model_schools' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.001153 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 11.53 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 89.9896 seconds (Warm-up)
## Chain 3:                80.0815 seconds (Sampling)
## Chain 3:                170.071 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'rt_model_schools' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000809 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 8.09 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
```

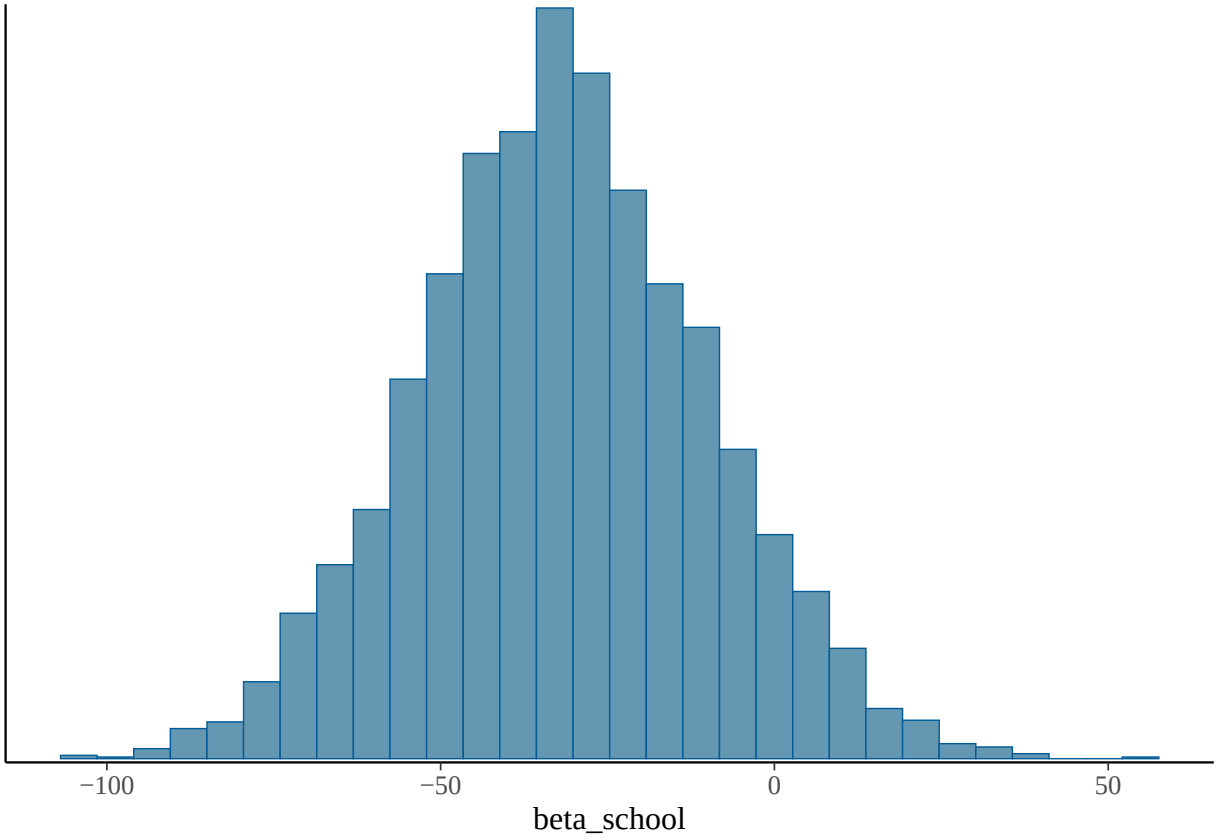
```
## Chain 4:  Elapsed Time: 93.26 seconds (Warm-up)
## Chain 4:                90.018 seconds (Sampling)
## Chain 4:                183.278 seconds (Total)
## Chain 4:
```

```
print(fit_model_lomb, pars = 'beta_school')
```

```
## Inference for Stan model: rt_model_schools.
## 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_school -31.73    0.38 21.59 -73.75 -46.02 -31.99 -17.21 10.81  3196    1
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  6 13:52:38 2020.
## 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).
```

```
mcmc_hist(fit_model_lomb, pars='beta_school')
```

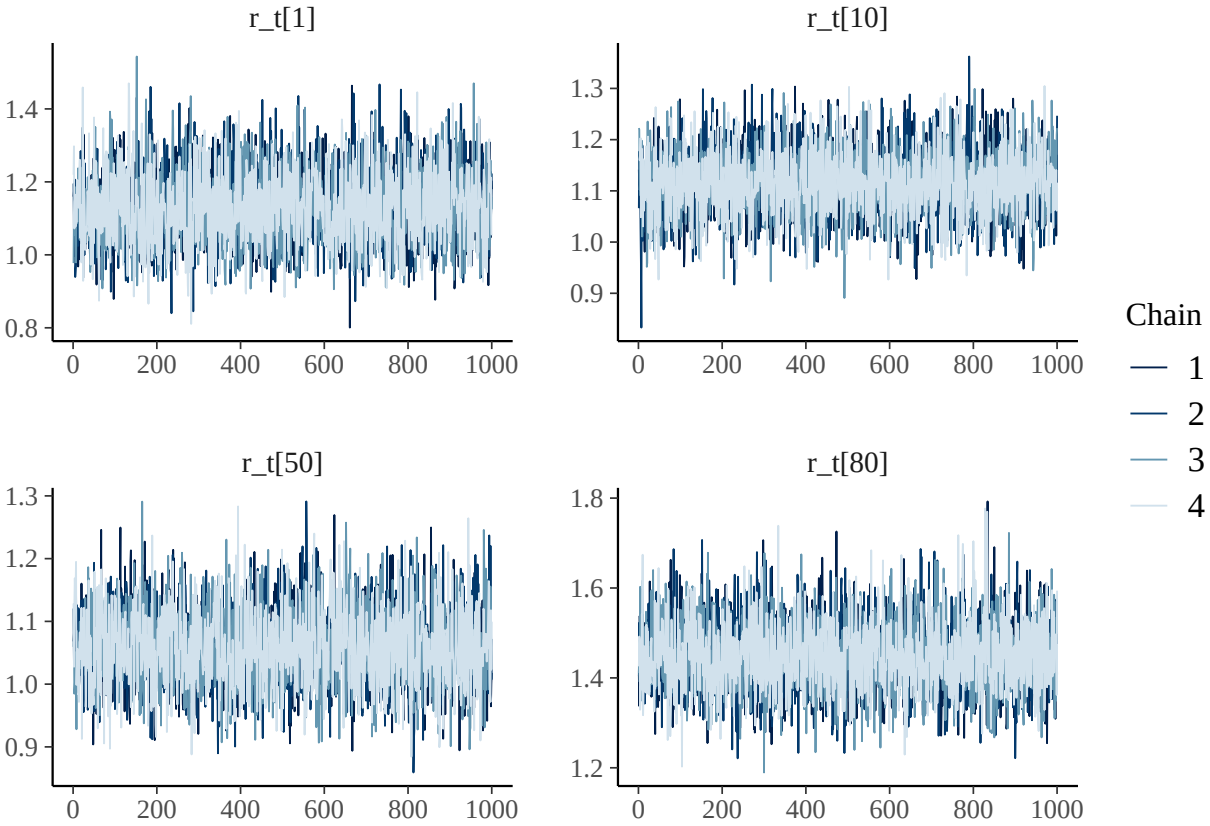
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Trace plots

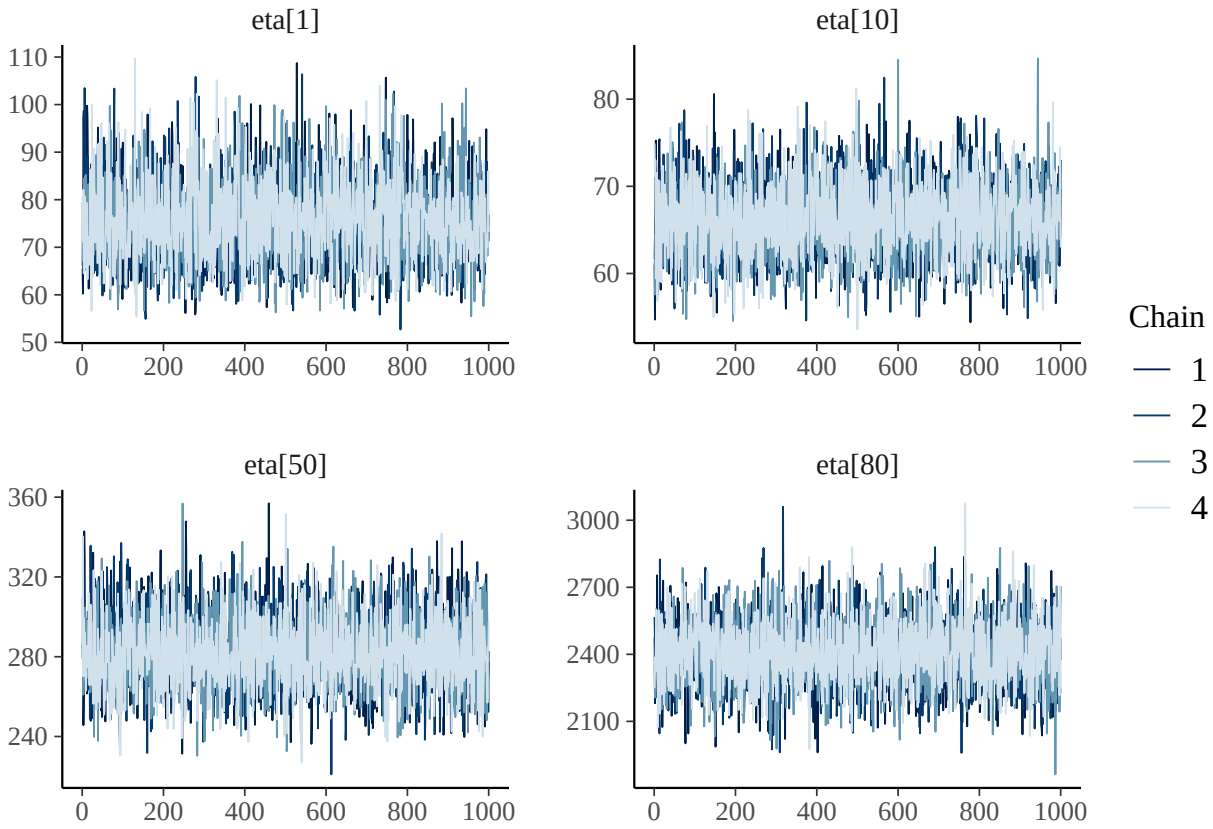
```
mcmc_trace(as.array(fit_model_lomb, pars = c('r_t[1]', 'r_t[10]', 'r_t[50]', 'r_t[80]')),
           np = nuts_params(fit_model_lomb)
           )
```

```
## No divergences to plot.
```



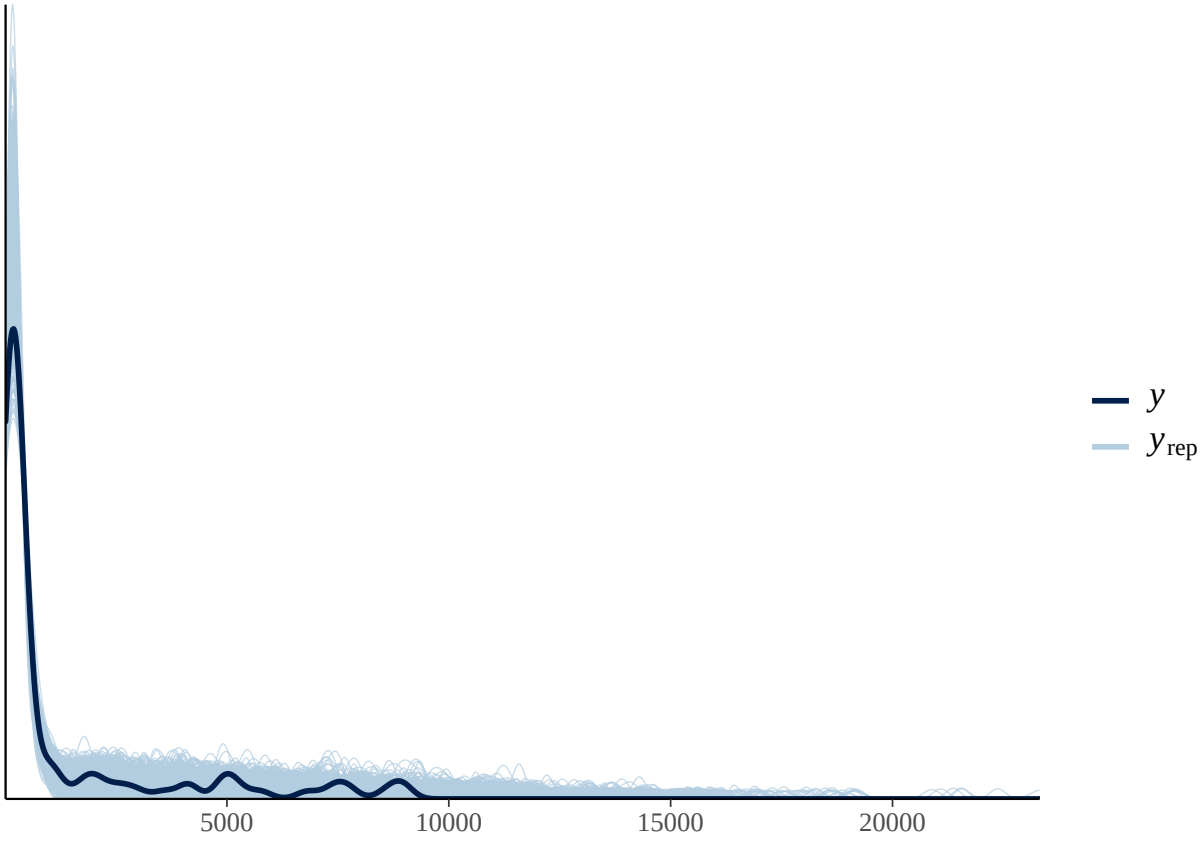

```
mcmc_trace(as.array(fit_model_lomb, pars = c('eta[1]', 'eta[10]', 'eta[50]', 'eta[80]')),
           np = nuts_params(fit_model_lomb)
)
```

```
## No divergences to plot.
```

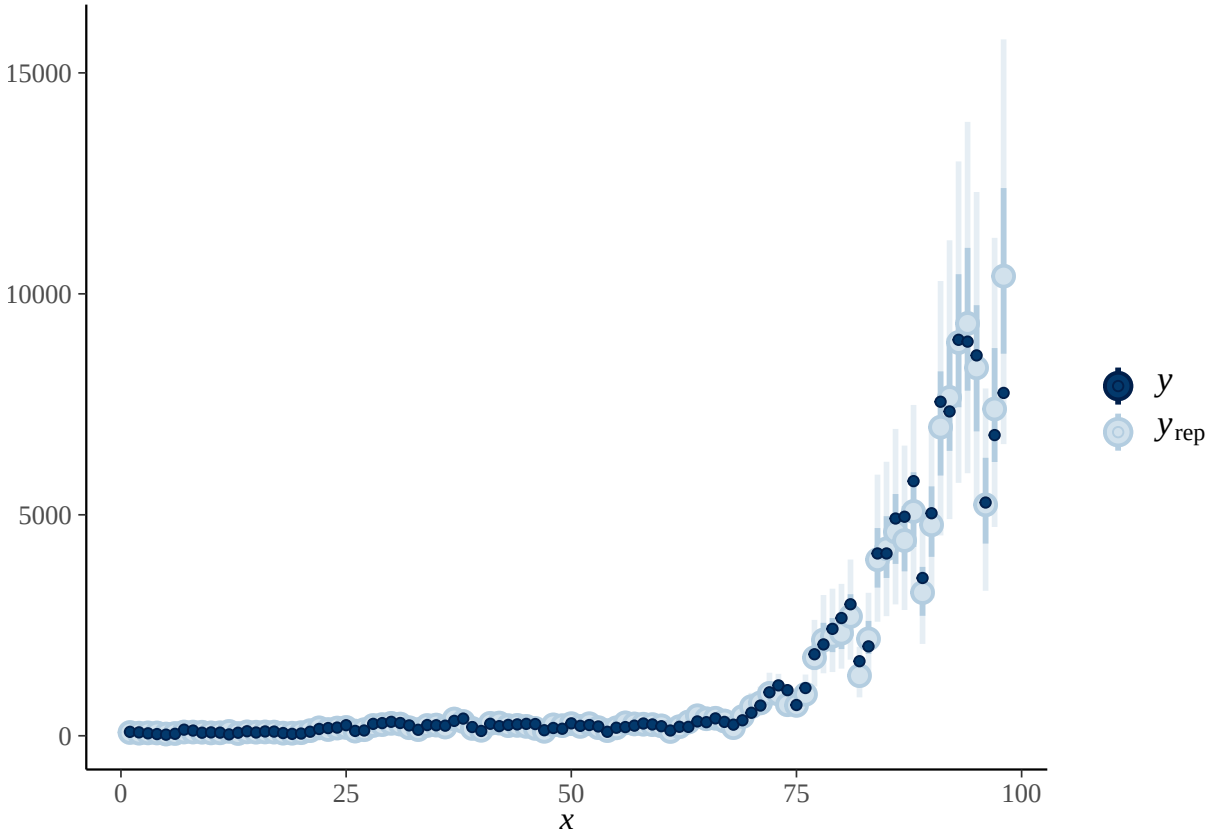


Posterior predictive check

```
y_rep <- as.matrix(fit_model_lomb, pars = "y_rep")
ppc_dens_overlay(y = data_lombardia$positive[nonzero_days_1], y_rep[1:1000, ])
```

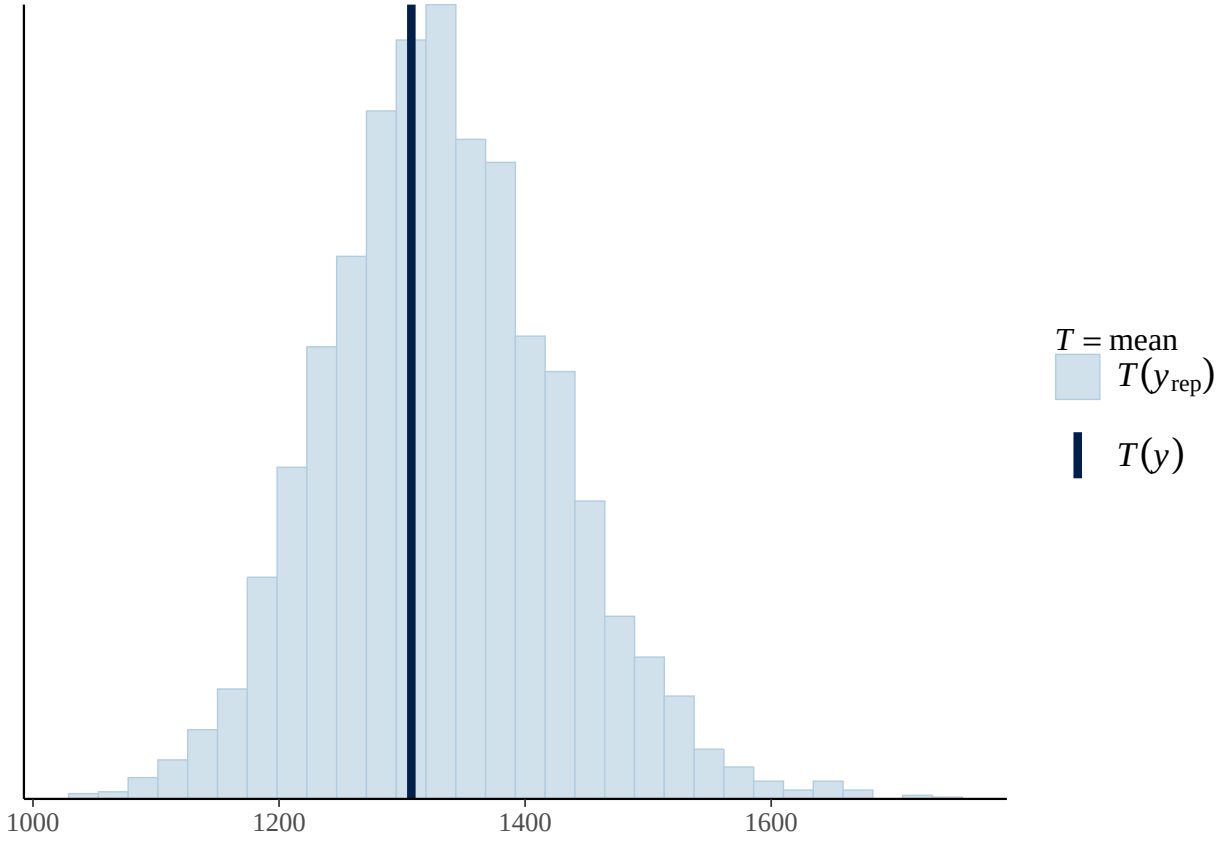



```
ppc_intervals(
  y = data_lombardia$positive[nonzero_days_l],
  yrep = y_rep
)
```




```
ppc_stat(y = stan_data_l$nonzero_positives, yrep = y_rep, stat = 'mean')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

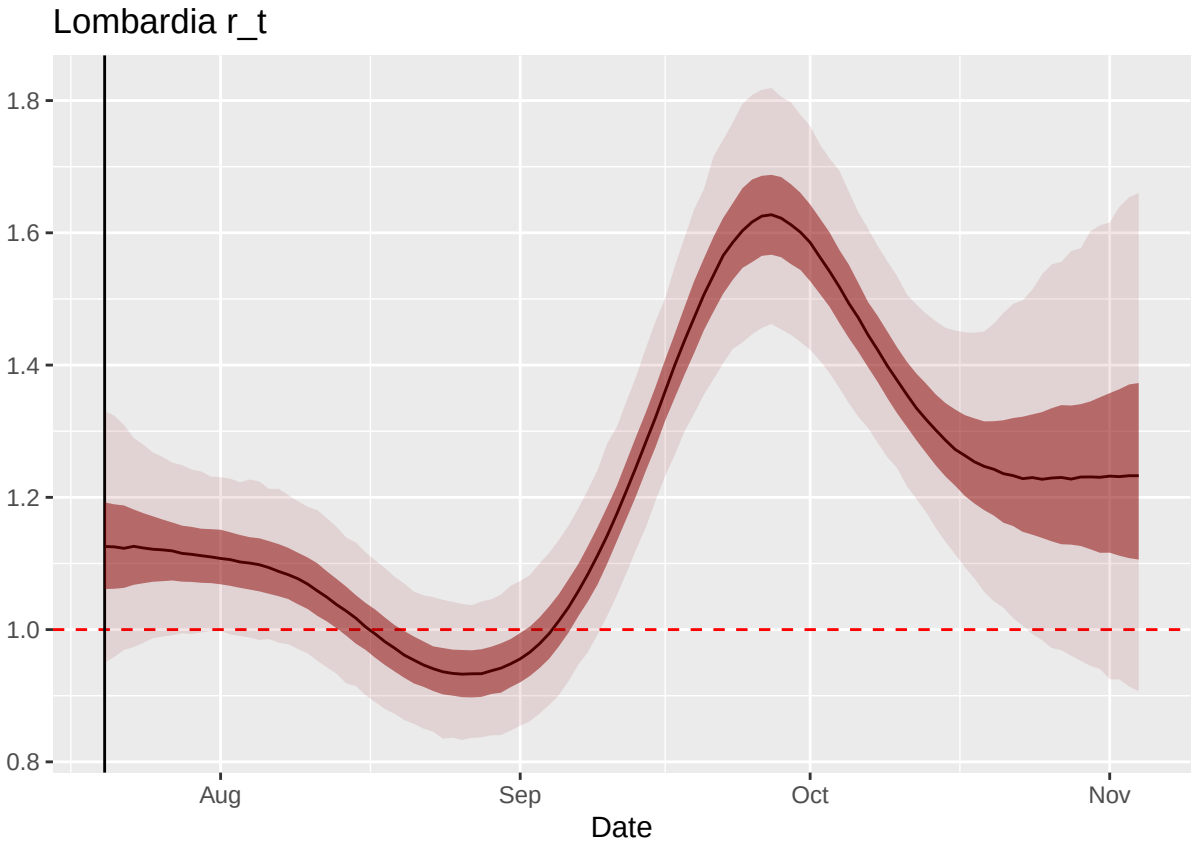


R_t curve

```
fit_summary_lomb <- summary(fit_model_lomb)

rt_idx <- which(rownames(fit_summary_lomb$summary) == 'r_t[1]')
medians_rt <- fit_summary_lomb$summary[rt_idx: (rt_idx + stan_data_l$N - 1), '50%']
min_rt_50_interval <- fit_summary_lomb$summary[rt_idx: (rt_idx + stan_data_l$N - 1), '25%']
max_rt_50_interval <- fit_summary_lomb$summary[rt_idx: (rt_idx + stan_data_l$N - 1), '75%']
min_rt_95_interval <- fit_summary_lomb$summary[rt_idx: (rt_idx + stan_data_l$N - 1), '2.5%']
max_rt_95_interval <- fit_summary_lomb$summary[rt_idx: (rt_idx + stan_data_l$N - 1), '97.5%']


ggplot(data = NULL, aes(x = data_lombardia$date, y = medians_rt)) +
  geom_line() +
  xlab('Date') +
  ylab('') +
  ggtitle( 'Lombardia r_t')+
  geom_hline(yintercept=1, linetype="dashed", color = "red") +
  geom_vline(xintercept = data_lombardia$date[1]) +
  geom_ribbon(aes(ymin = min_rt_50_interval, ymax = max_rt_50_interval), alpha= 0.5, fill = 'darkred') +
  geom_ribbon(aes(ymin = min_rt_95_interval, ymax = max_rt_95_interval), alpha= 0.1, fill = 'darkred')
```


Lombardia r_t
1.8
1.6
1.4
1.2
1.0
0.8
Aug
Sep
Oct
Nov
Date