

Investigating Racial Profiling

Diagnostic Plots

Chen Shi, Belle Xu

4/23/2021

Model Preparation

```
stop <- read.table("stop-and-frisk.dat", header = TRUE)

total_crime <- stop %>%
  mutate(eth = factor(eth, levels = c("3", "1", "2")),
        precinct = factor(precinct),
        crime = factor(crime)) %>%
  group_by(precinct) %>%
  mutate(total_pop = sum(pop) / 4,
        pop_prop = pop / total_pop,
        total_arrest = sum(past.arrests),
        crime_prop = past.arrests / total_arrest) %>%
  mutate(past.arrests = if_else(past.arrests != 0, past.arrests, as.integer(1)))
```

```
model.3.full <- stan_glm(data = total_crime,
                           formula = stops ~ log(past.arrests) +
                             eth * (pop_prop + crime) + pop_prop * crime +
                             eth*log(past.arrests) + pop_prop*log(past.arrests) +
                             crime*log(past.arrests),
                           family = neg_binomial_2(link = "log"),
                           prior = cauchy(0, 2.5),
                           prior_intercept = cauchy(0, 2.5),
                           seed = 360,
                           refresh = 0,
                           diagnostic_file = file.path(tempdir(), "glm3.csv"))
```

LOOIC Diagnostics

```
#Pareto-K Diagnostic
loo <- loo(model.3.full)
loo

##
## Computed from 4000 by 900 log-likelihood matrix
```

```

##          Estimate    SE
## elpd_loo   -4811.5 43.3
## p_loo       30.3  3.3
## looic      9623.1 86.6
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## Pareto k diagnostic values:
##                               Count Pct.  Min. n_eff
## (-Inf, 0.5]   (good)     899 99.9%  562
## (0.5, 0.7]   (ok)        1  0.1%  208
## (0.7, 1]     (bad)       0  0.0% <NA>
## (1, Inf)     (very bad)  0  0.0% <NA>
##
## All Pareto k estimates are ok (k < 0.7).
## See help('pareto-k-diagnostic') for details.

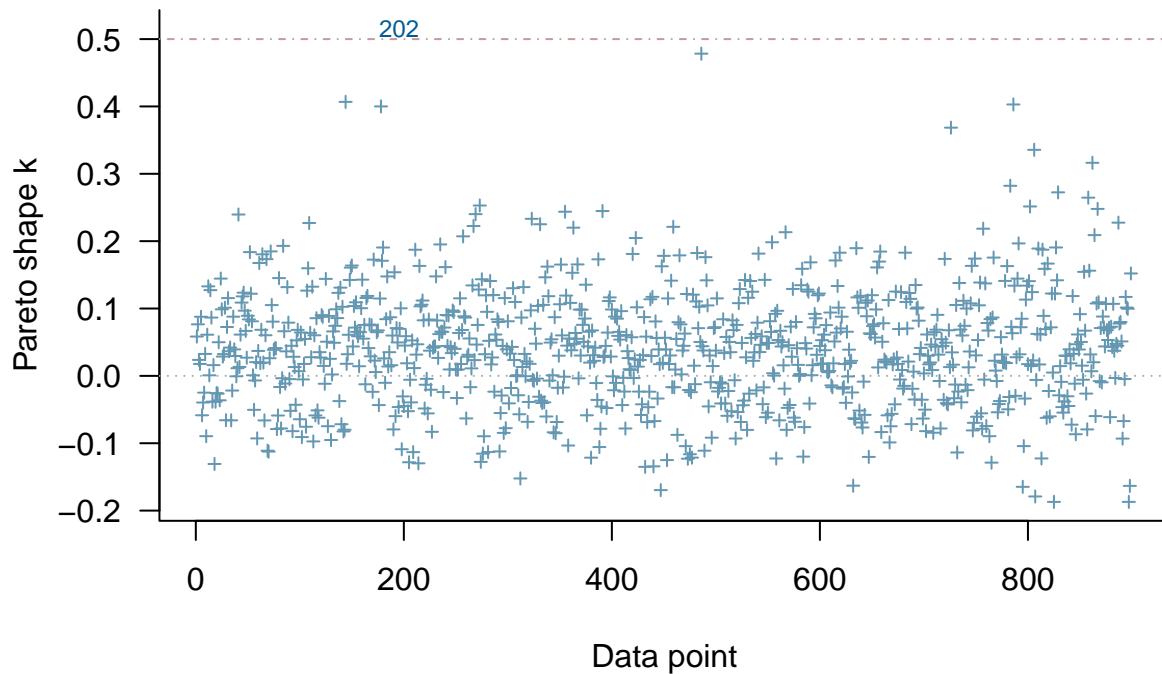
```

```

#check for outliers that could cause skew in our posterior prediction
plot(loo, label_points = TRUE)

```

PSIS diagnostic plot



MCMC Chain Diagnostic

```

summary(model.3.full)

## Model Info:
##   function: stan_glm
##   family: neg_binomial_2 [log]
##   formula: stops ~ log(past.arrests) + eth * (pop_prop + crime) + pop_prop *
##             crime + eth * log(past.arrests) + pop_prop * log(past.arrests) +
##             crime * log(past.arrests)
##   algorithm: sampling
##   sample: 4000 (posterior sample size)
##   priors: see help('prior_summary')
##   observations: 900
##   predictors: 25
##
## Estimates:
##               mean    sd   10%   50%   90%
## (Intercept) -1.3   0.4  -1.8  -1.3  -0.8
## log(past.arrests) 1.0   0.1   0.8   1.0   1.1
## eth1         1.7   0.4   1.2   1.7   2.2
## eth2         1.0   0.4   0.5   1.0   1.5
## pop_prop     2.5   0.6   1.7   2.5   3.4
## crime2       1.4   0.4   0.9   1.4   1.8
## crime3       2.5   0.4   2.0   2.5   3.0
## crime4       0.7   0.4   0.2   0.7   1.2
## eth1:pop_prop -0.5  0.4  -1.0  -0.5  0.0
## eth2:pop_prop -0.1  0.4  -0.6  -0.1  0.5
## eth1:crime2   -0.5  0.3  -0.9  -0.5  -0.2
## eth2:crime2   -0.3  0.2  -0.6  -0.3  0.0
## eth1:crime3   -0.4  0.3  -0.8  -0.4  -0.1
## eth2:crime3   0.1   0.2  -0.2  0.1   0.4
## eth1:crime4   -0.1  0.2  -0.4  -0.1  0.3
## eth2:crime4   0.3   0.2  0.0   0.3   0.5
## pop_prop:crime2 -0.4  0.4  -1.0  -0.4  0.1
## pop_prop:crime3  1.2   0.4   0.7   1.2   1.7
## pop_prop:crime4  1.2   0.4   0.7   1.2   1.7
## log(past.arrests):eth1 -0.1  0.1  -0.2  -0.1  0.0
## log(past.arrests):eth2  0.0   0.1  -0.1  0.0   0.1
## log(past.arrests):pop_prop -0.4  0.1  -0.5  -0.4  -0.2
## log(past.arrests):crime2  0.1   0.1  -0.1  0.1   0.2
## log(past.arrests):crime3 -0.5  0.1  -0.6  -0.5  -0.4
## log(past.arrests):crime4 -0.4  0.1  -0.5  -0.4  -0.2
## reciprocal_dispersion  2.0   0.1   1.8   2.0   2.1
##
## Fit Diagnostics:
##               mean    sd   10%   50%   90%
## mean_PPD 146.9   7.5 137.4 146.5 156.8
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)      0.0   1.0 1399

```

```

## log(past.arrests)          0.0  1.0 1170
## eth1                        0.0  1.0 1882
## eth2                        0.0  1.0 1960
## pop_prop                     0.0  1.0 2090
## crime2                       0.0  1.0 1466
## crime3                       0.0  1.0 1433
## crime4                       0.0  1.0 1646
## eth1:pop_prop                0.0  1.0 2226
## eth2:pop_prop                0.0  1.0 2717
## eth1:crime2                  0.0  1.0 1598
## eth2:crime2                  0.0  1.0 1804
## eth1:crime3                  0.0  1.0 1493
## eth2:crime3                  0.0  1.0 1688
## eth1:crime4                  0.0  1.0 1621
## eth2:crime4                  0.0  1.0 1779
## pop_prop:crime2              0.0  1.0 1545
## pop_prop:crime3              0.0  1.0 1488
## pop_prop:crime4              0.0  1.0 1501
## log(past.arrests):eth1       0.0  1.0 1879
## log(past.arrests):eth2       0.0  1.0 1998
## log(past.arrests):pop_prop   0.0  1.0 2406
## log(past.arrests):crime2     0.0  1.0 1235
## log(past.arrests):crime3     0.0  1.0 1146
## log(past.arrests):crime4     0.0  1.0 1081
## reciprocal_dispersion        0.0  1.0 5381
## mean_PPD                      0.1  1.0 3736
## log-posterior                 0.1  1.0 1427
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size

```

Posterior Predictive Checks

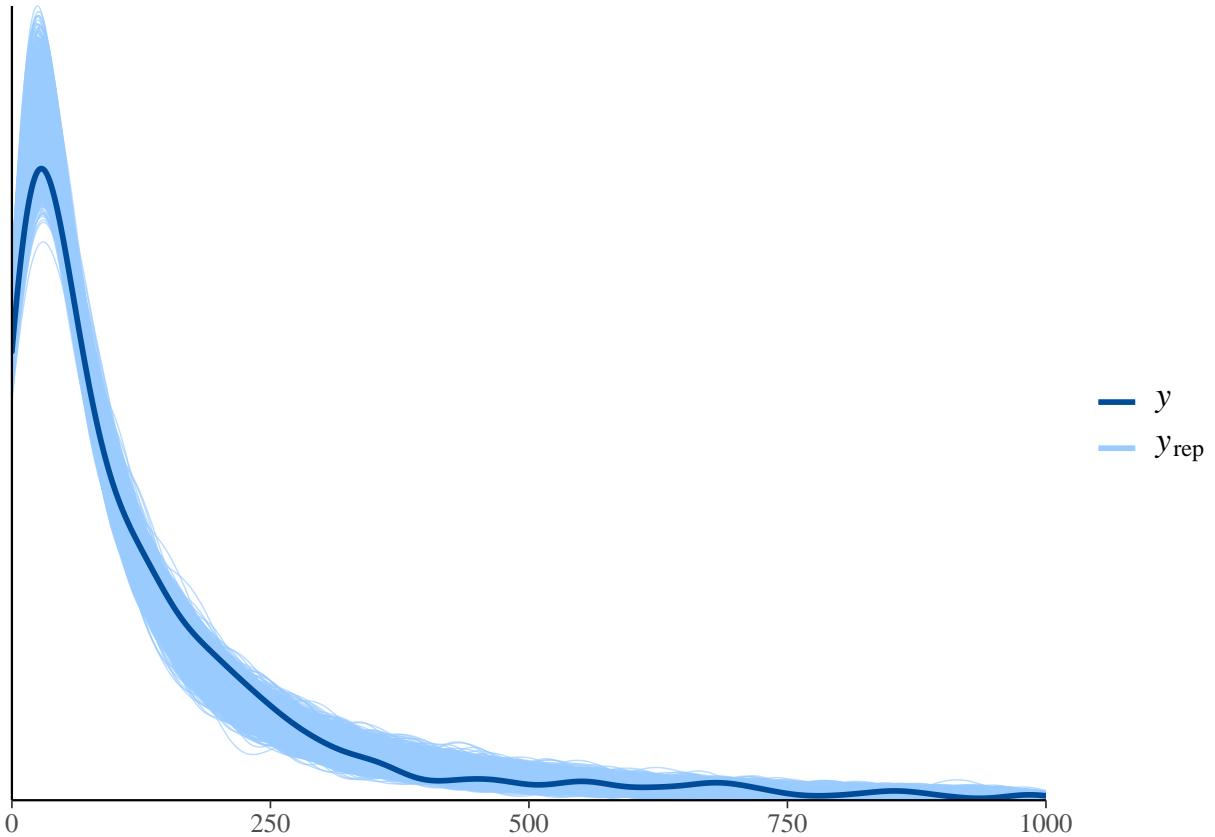
```

y.postpred <- posterior_predict(model.3.full)
color_scheme_set("brightblue")
ppc_dens_overlay(total_crime$stops, y.postpred) + xlim(0, 1000)

## Warning: Removed 52062 rows containing non-finite values (stat_density).

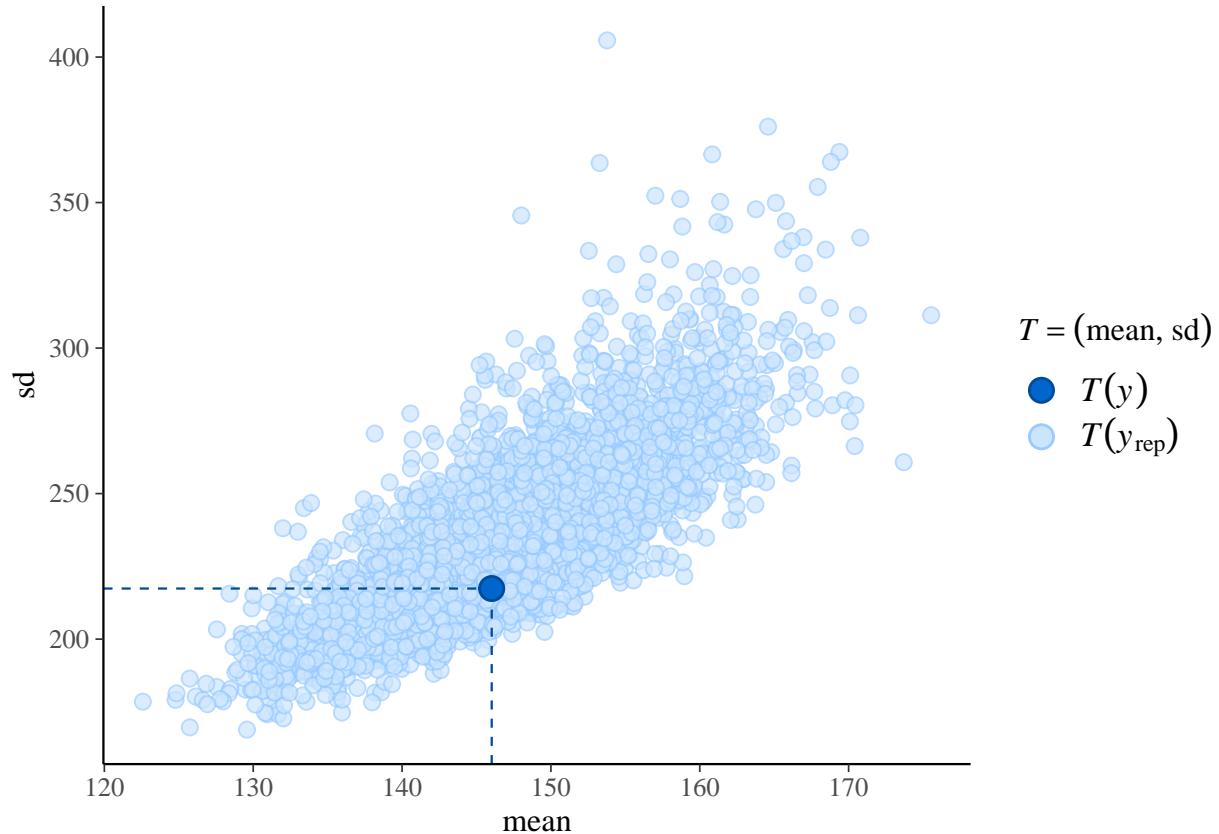
## Warning: Removed 11 rows containing non-finite values (stat_density).

```

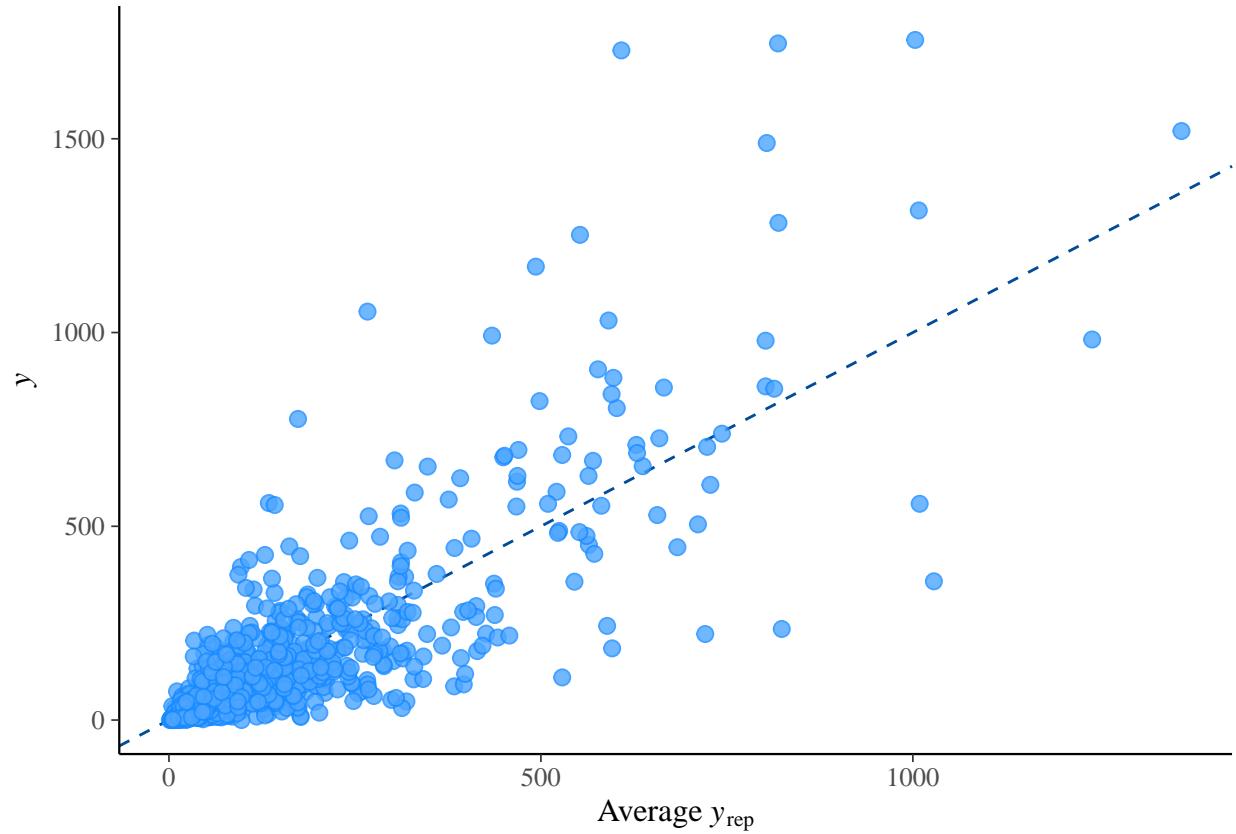


```
#xlim() truncates so that we focus on the part where x is less than 1000
```

```
pp_check(model.3.full, plotfun = "stat_2d", stat = c("mean", "sd"))
```

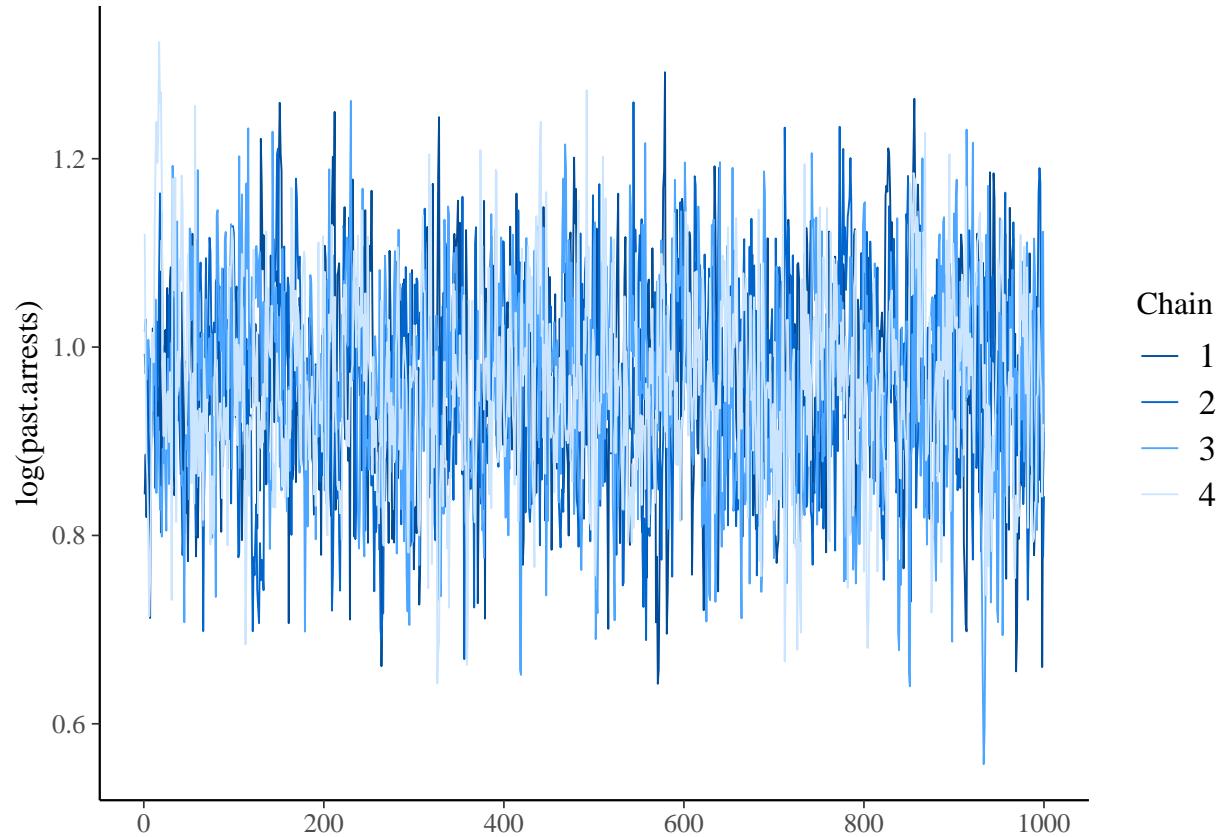


```
pp_check(model.3.full, plotfun = "scatter_avg")
```

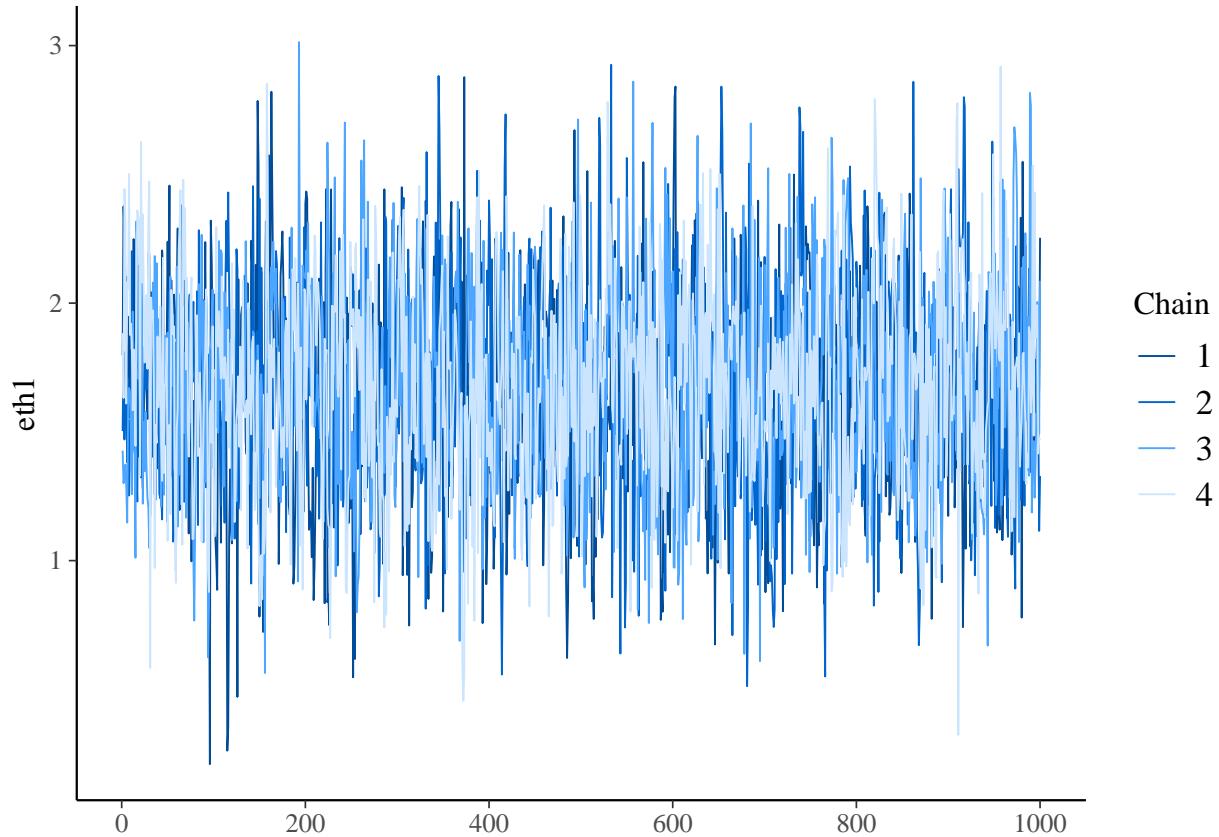


Traceplots

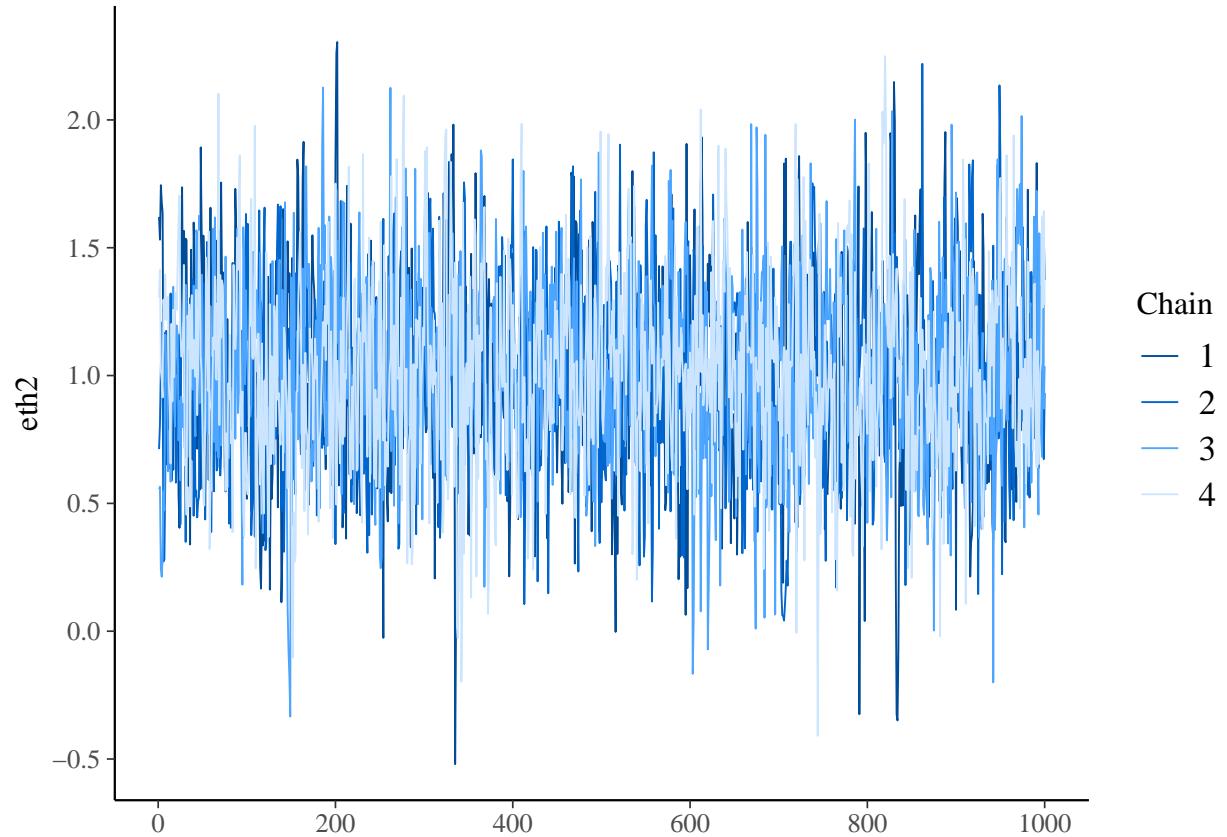
```
plot(model.3.full, "trace", pars = "log(past.arrests)")
```



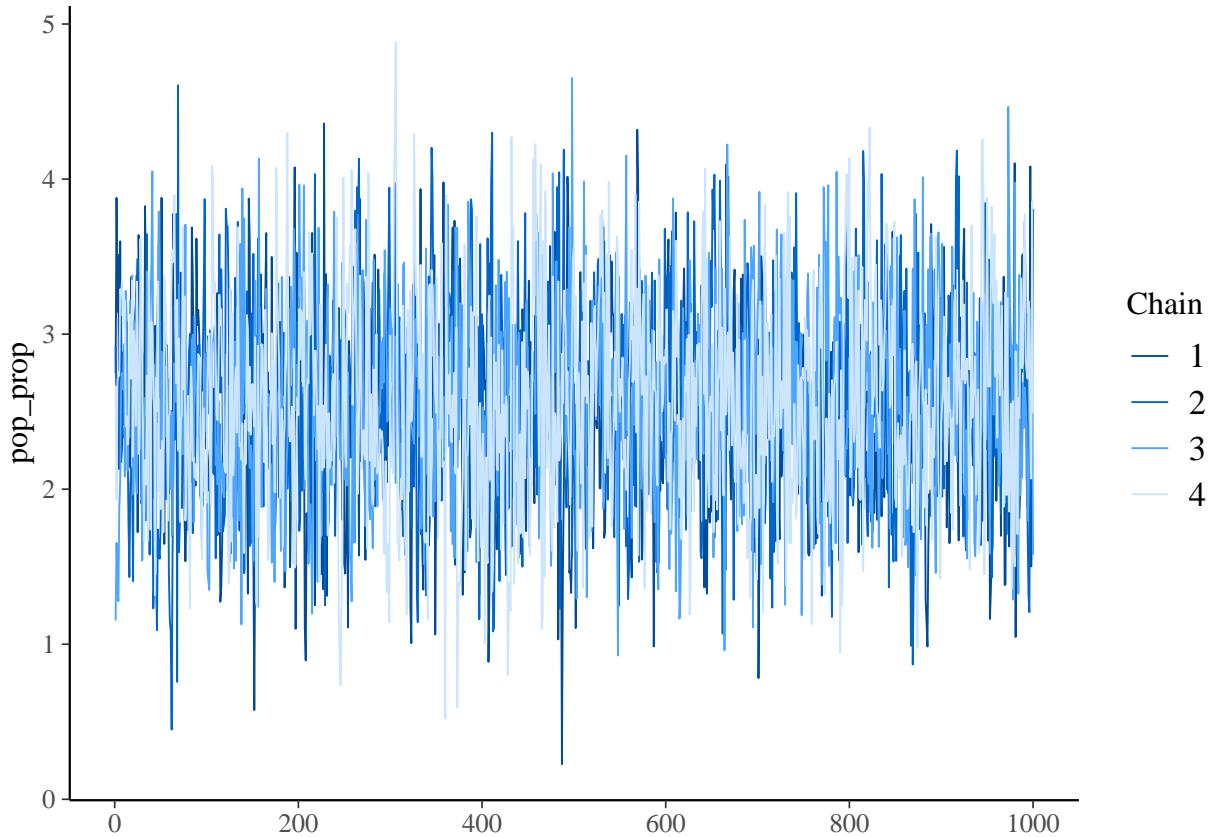
```
plot(model.3.full, "trace", pars = "eth1")
```



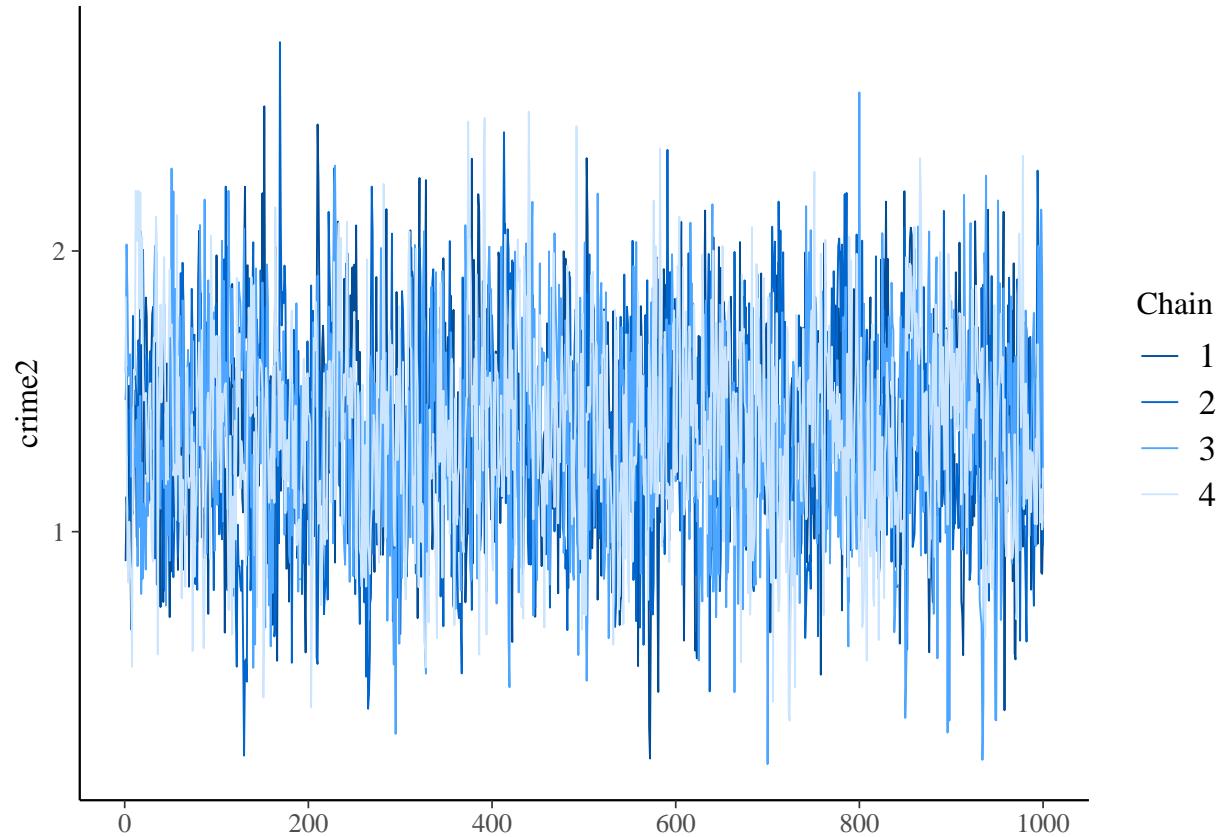
```
plot(model.3.full, "trace", pars = "eth2")
```



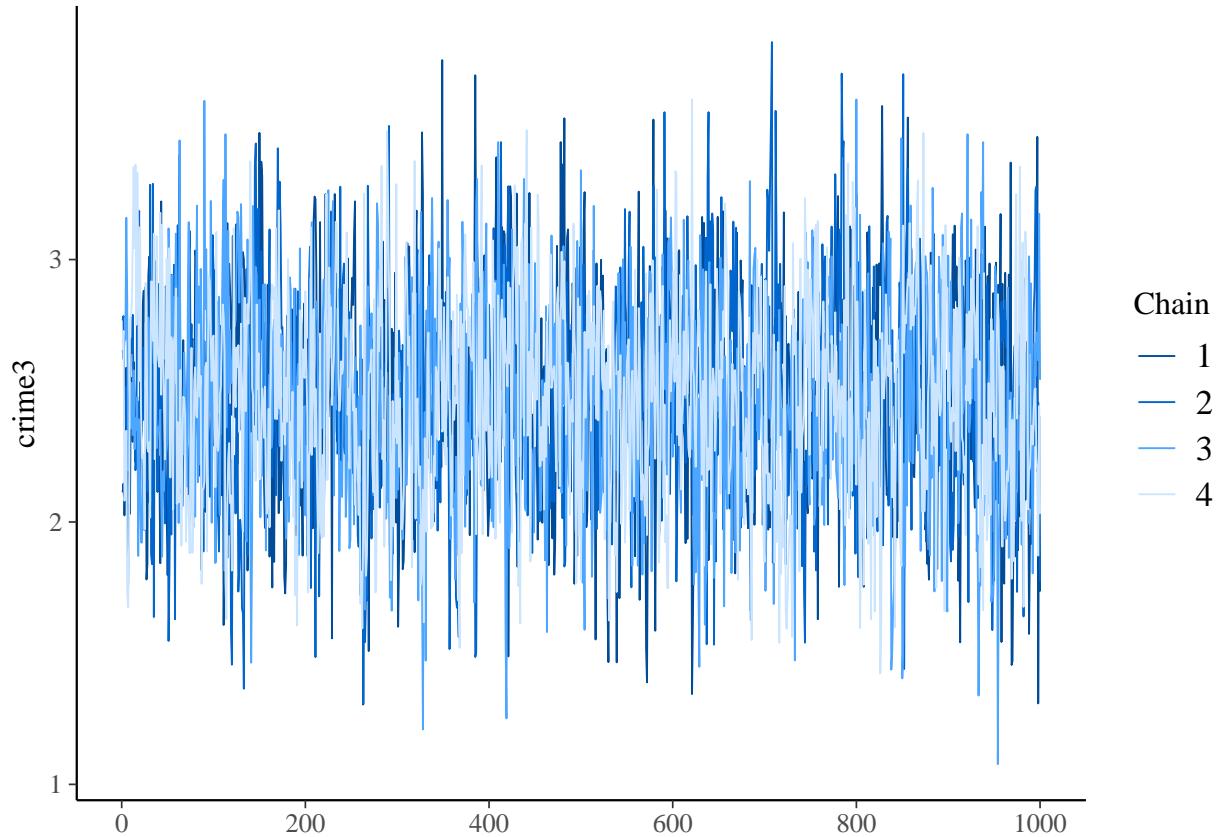
```
plot(model.3.full, "trace", pars = "pop_prop")
```



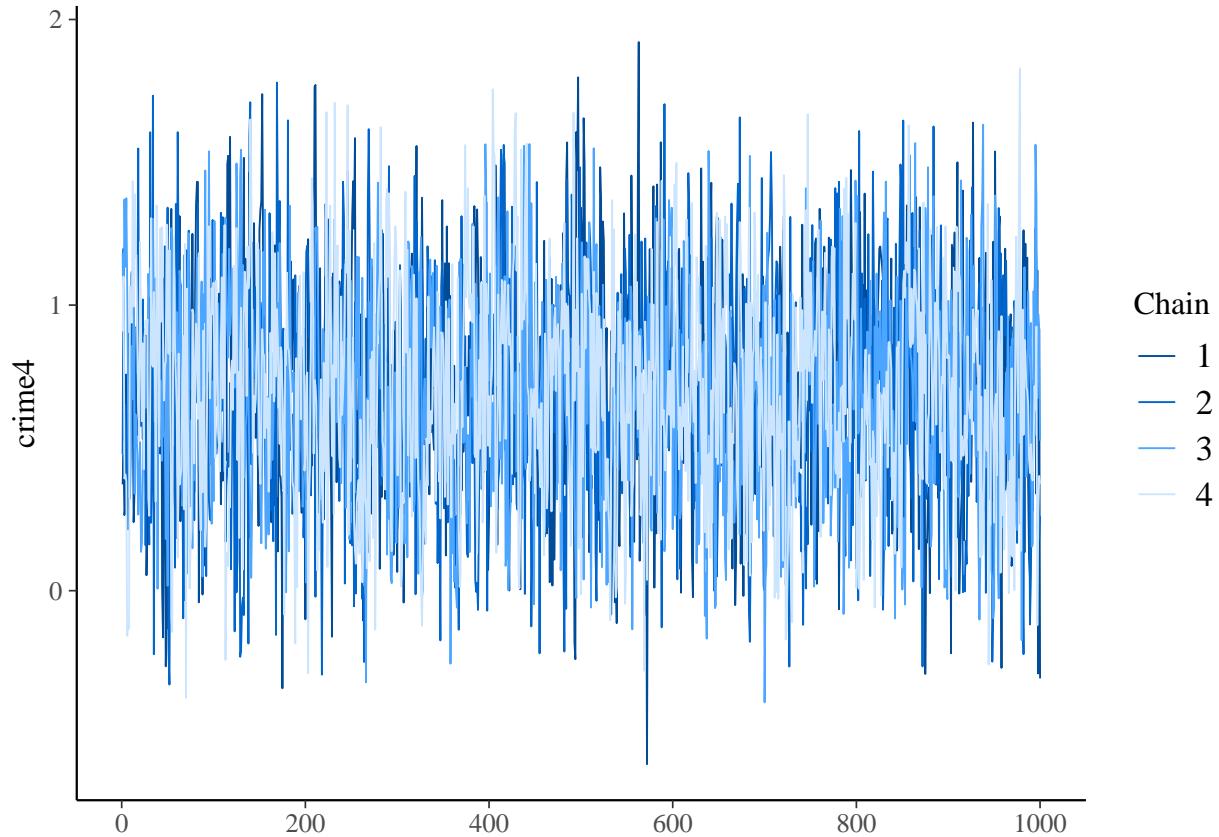
```
plot(model.3.full, "trace", pars = "crime2")
```



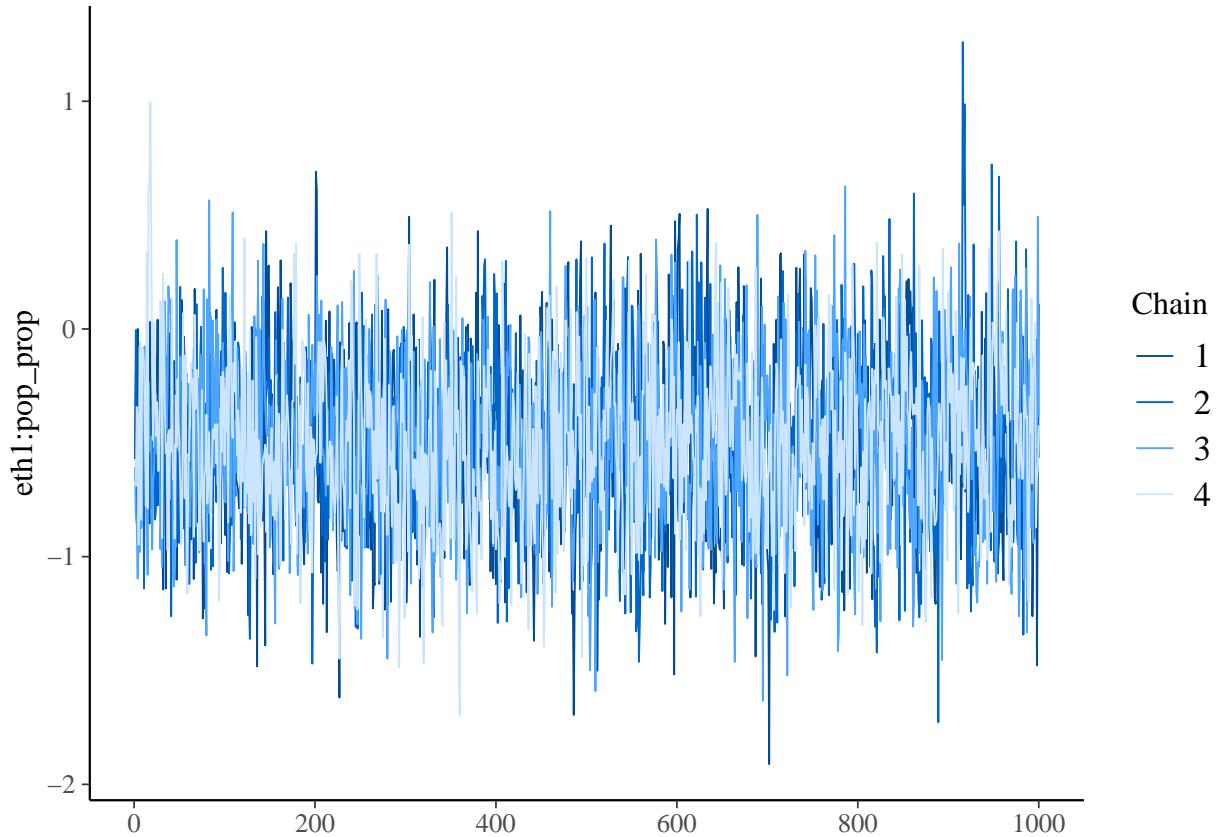
```
plot(model.3.full, "trace", pars = "crime3")
```



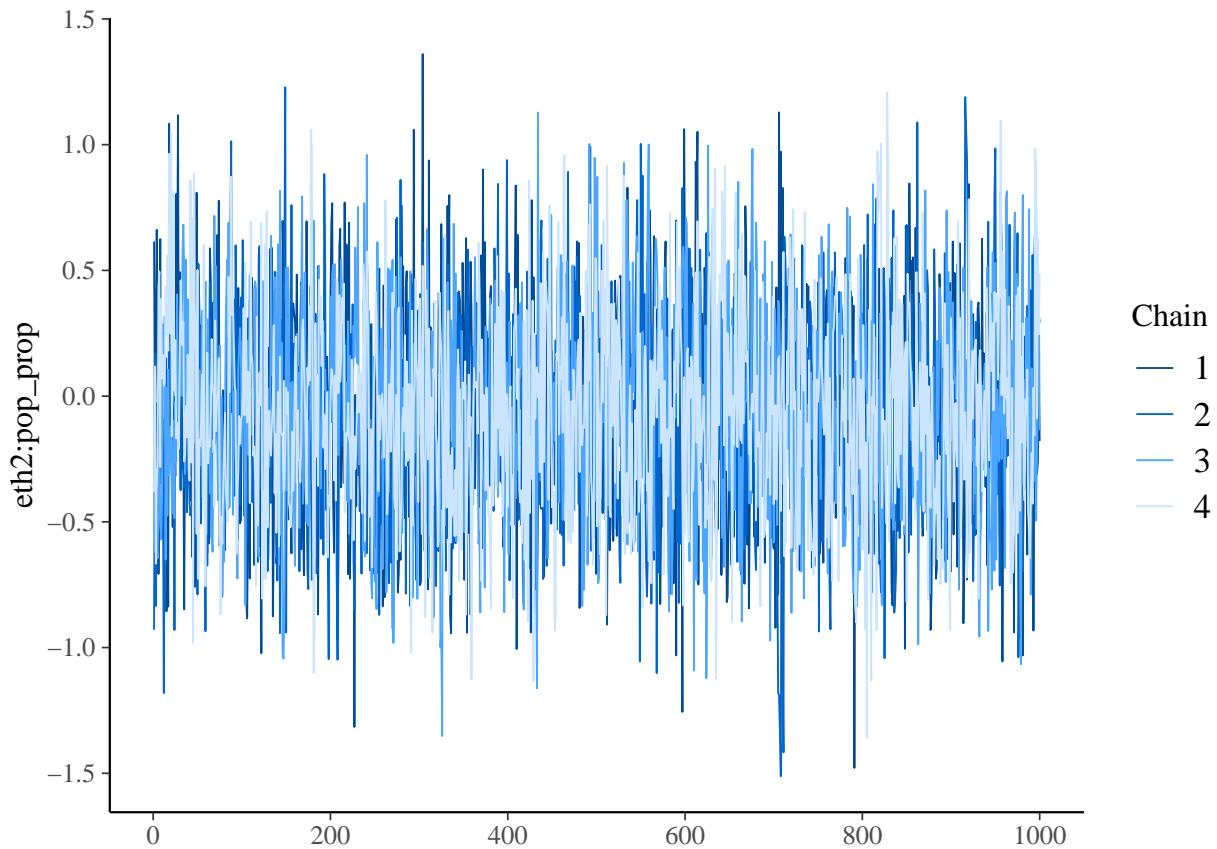
```
plot(model.3.full, "trace", pars = "crime4")
```



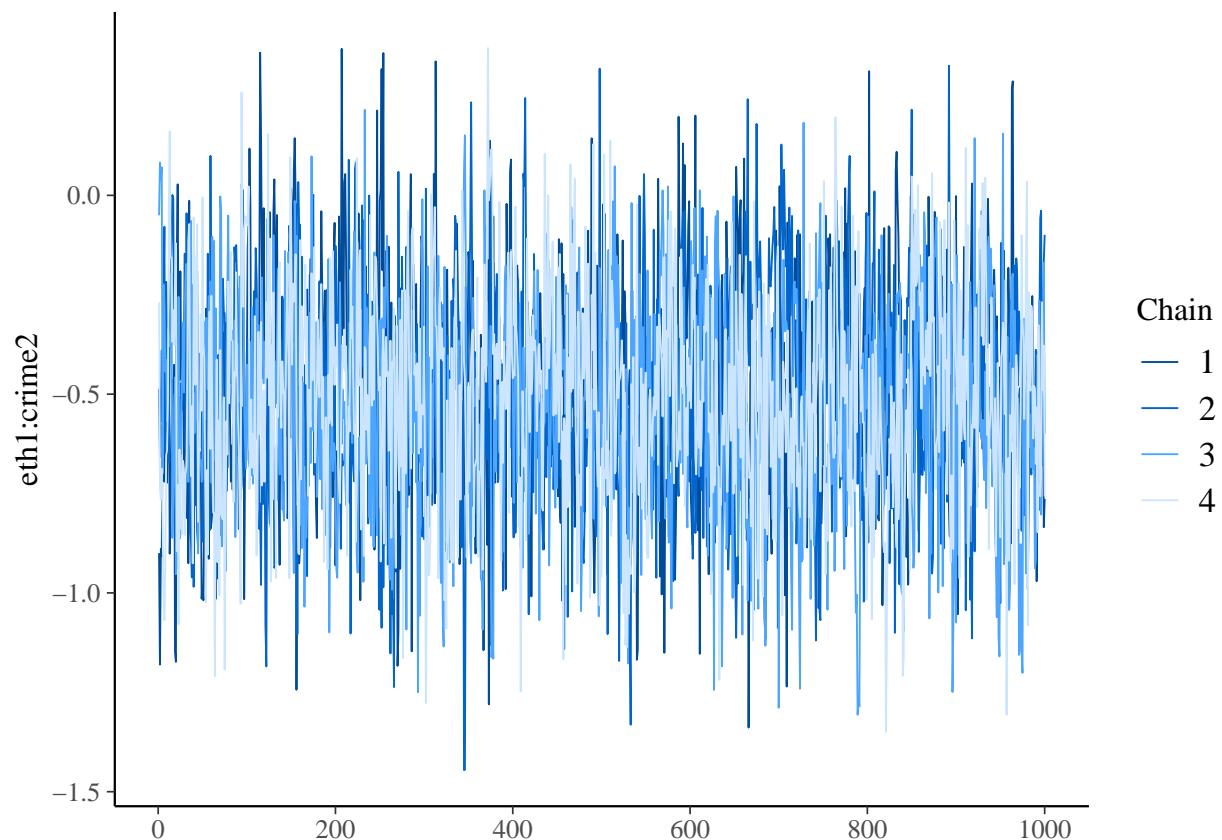
```
plot(model.3.full, "trace", pars = "eth1:pop_prop")
```



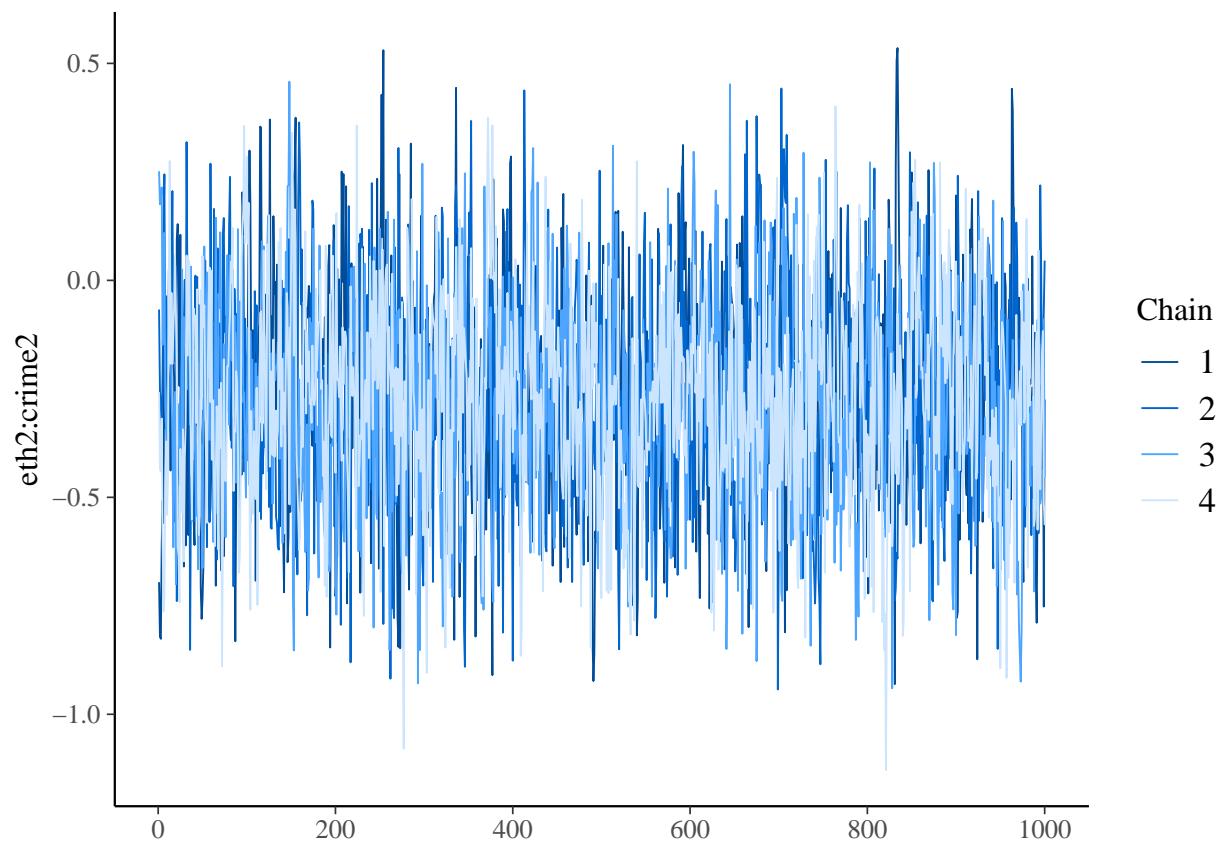
```
plot(model.3.full, "trace", pars = "eth2:pop_prop")
```



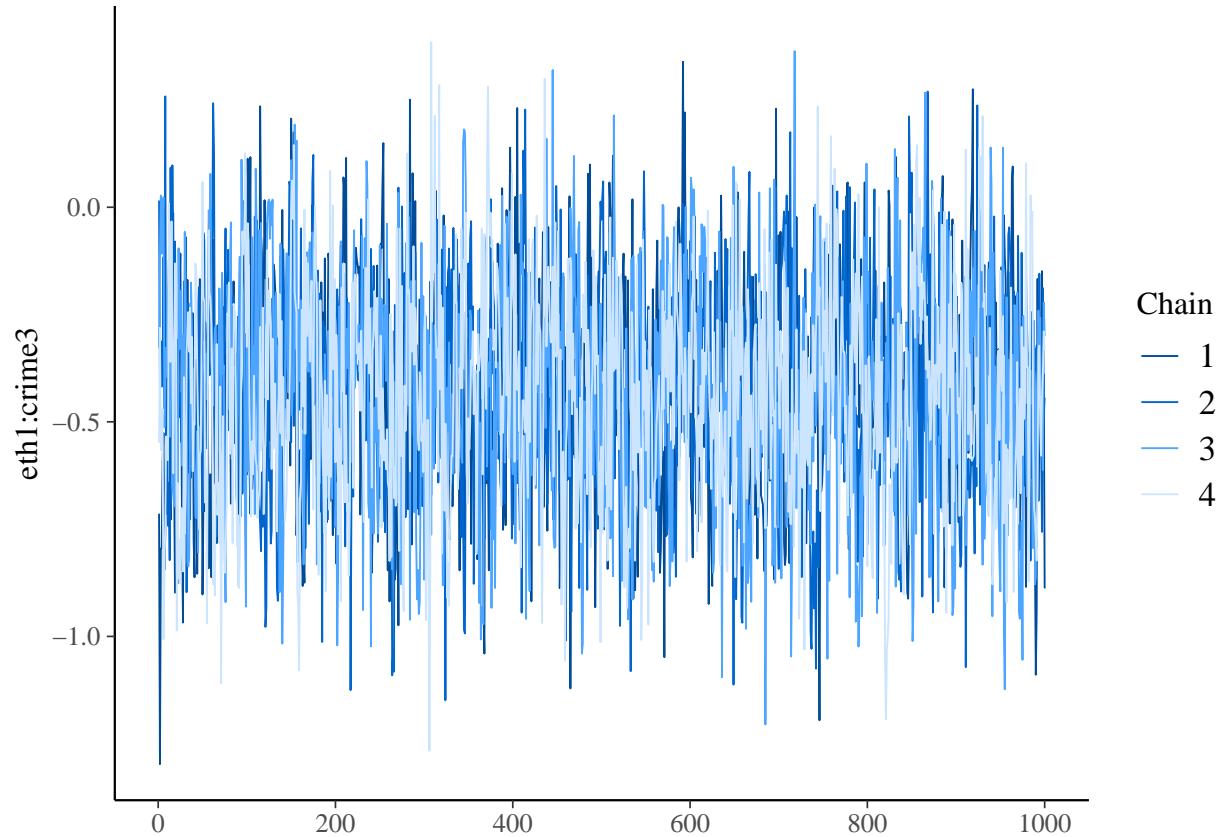
```
plot(model.3.full, "trace", pars = "eth1:crime2")
```



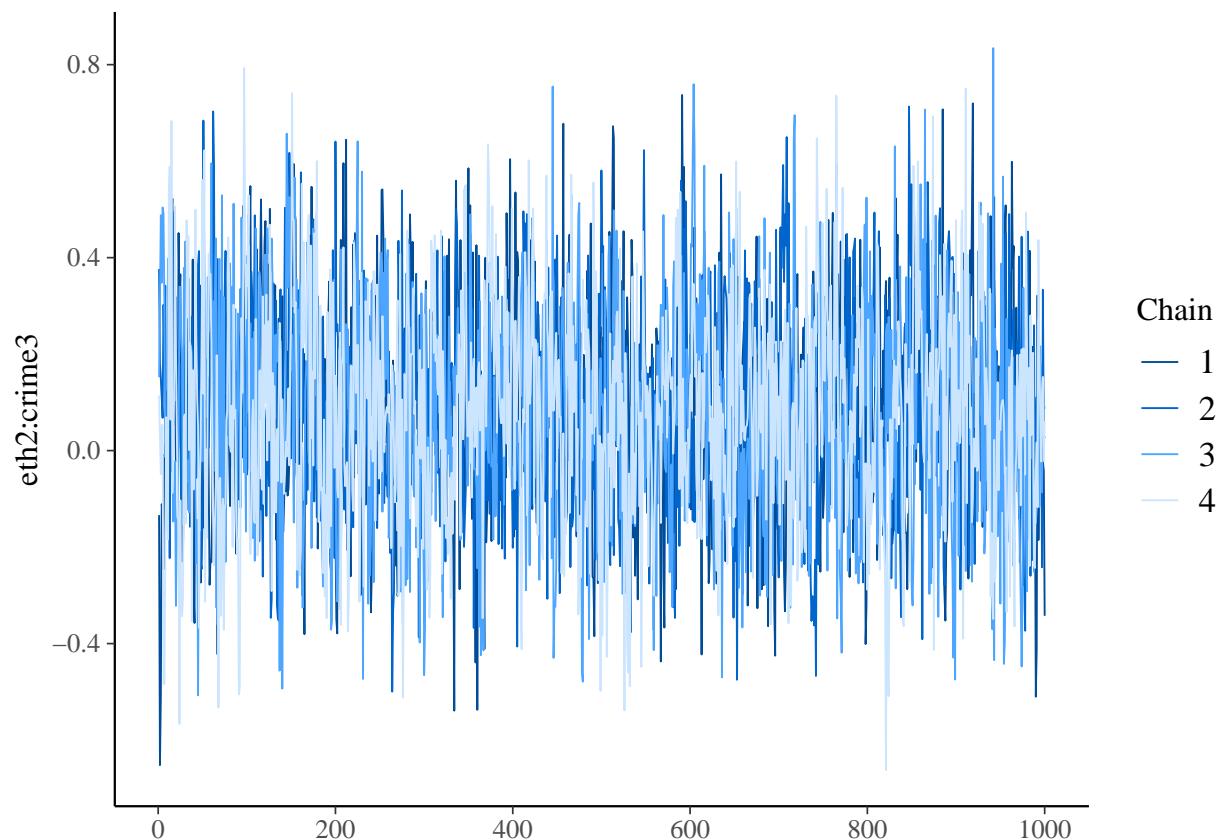
```
plot(model.3.full, "trace", pars = "eth2:crime2")
```



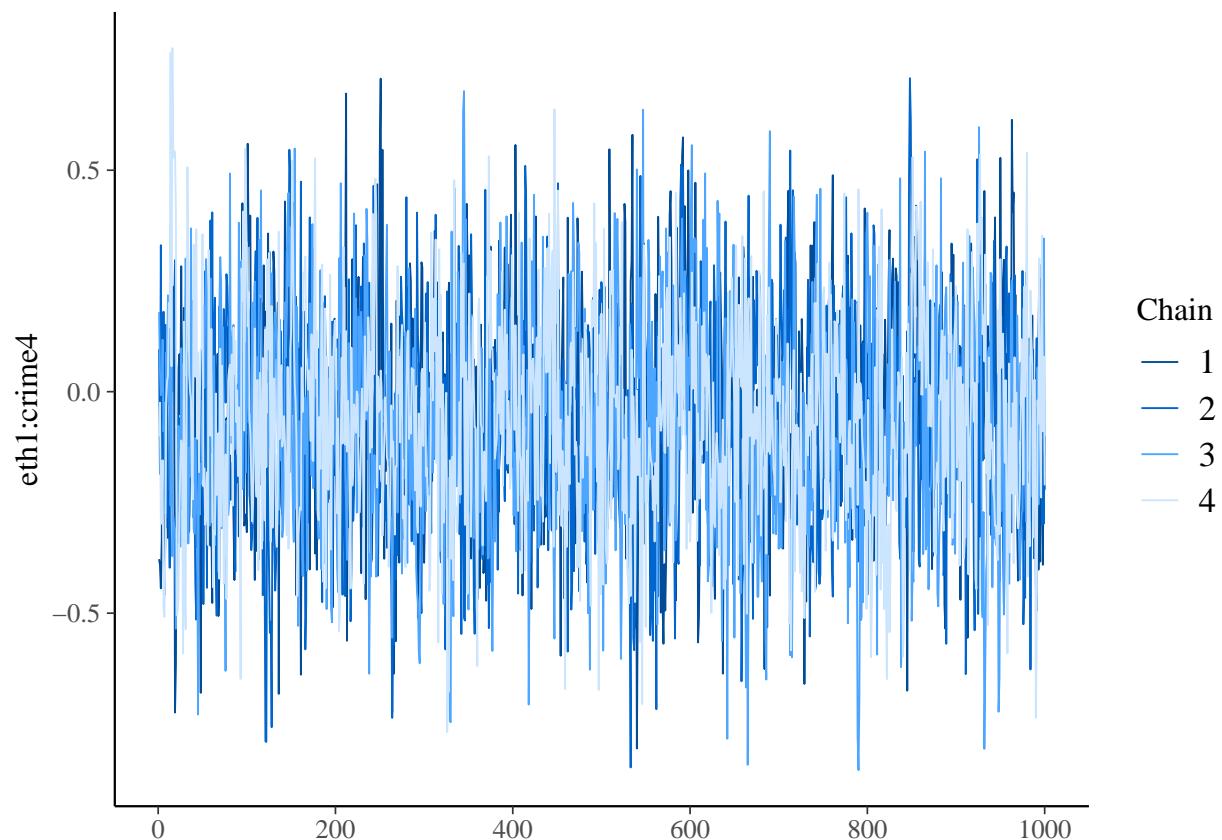
```
plot(model.3.full, "trace", pars = "eth1:crime3")
```



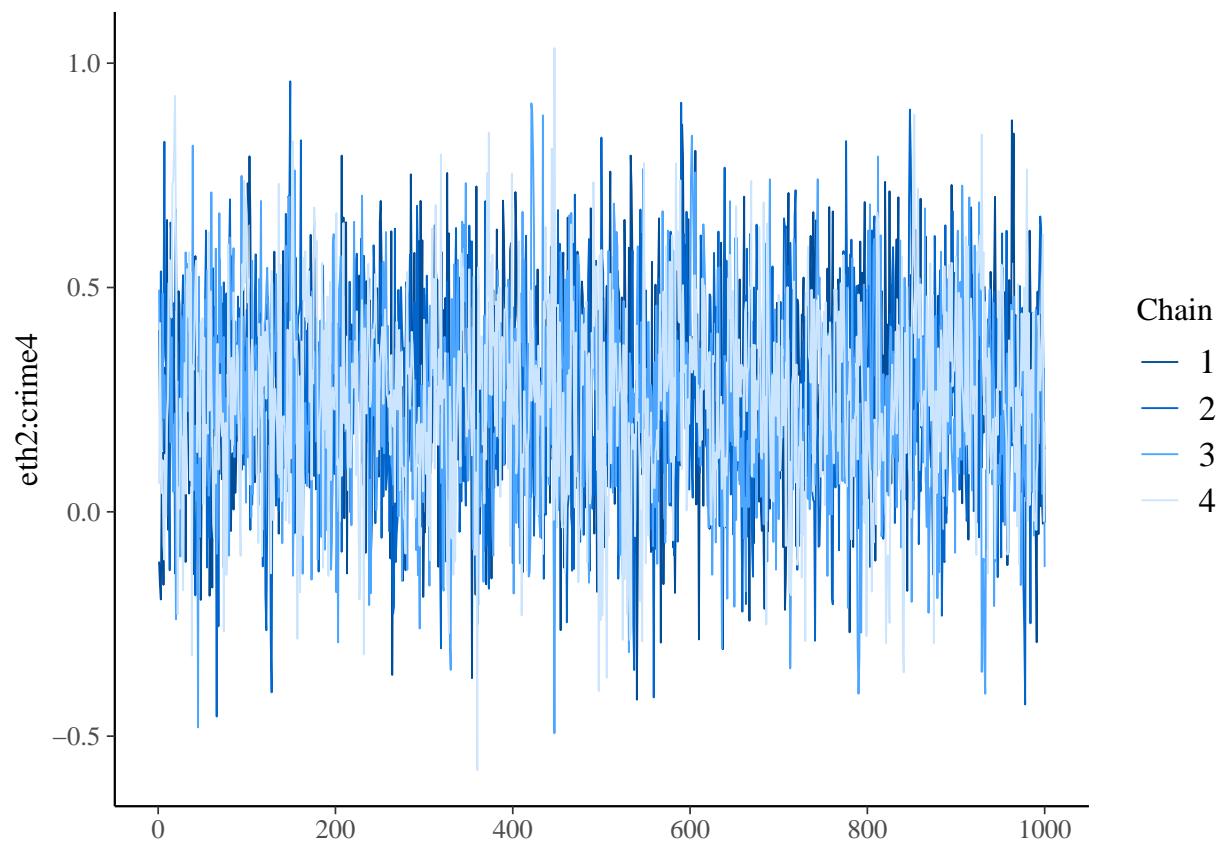
```
plot(model.3.full, "trace", pars = "eth2:crime3")
```



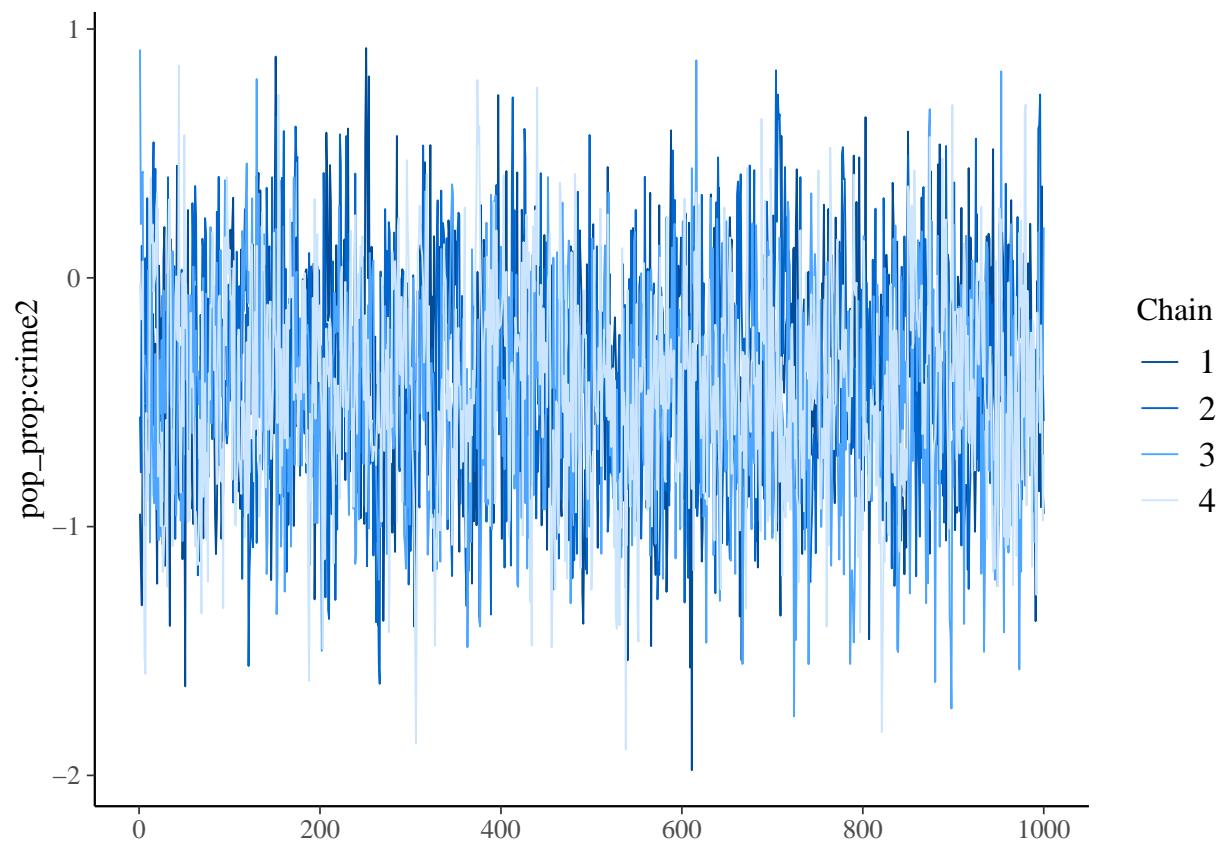
```
plot(model.3.full, "trace", pars = "eth1:crime4")
```



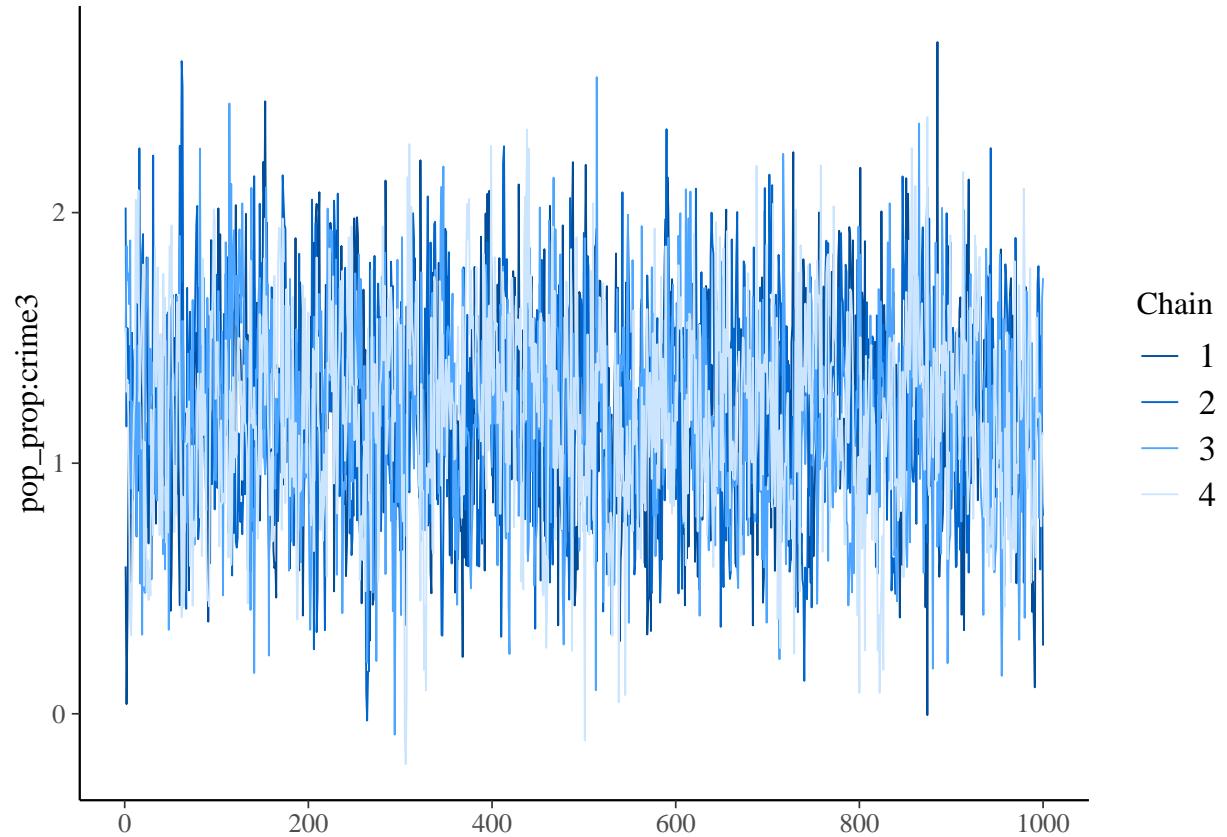
```
plot(model.3.full, "trace", pars = "eth2:crime4")
```



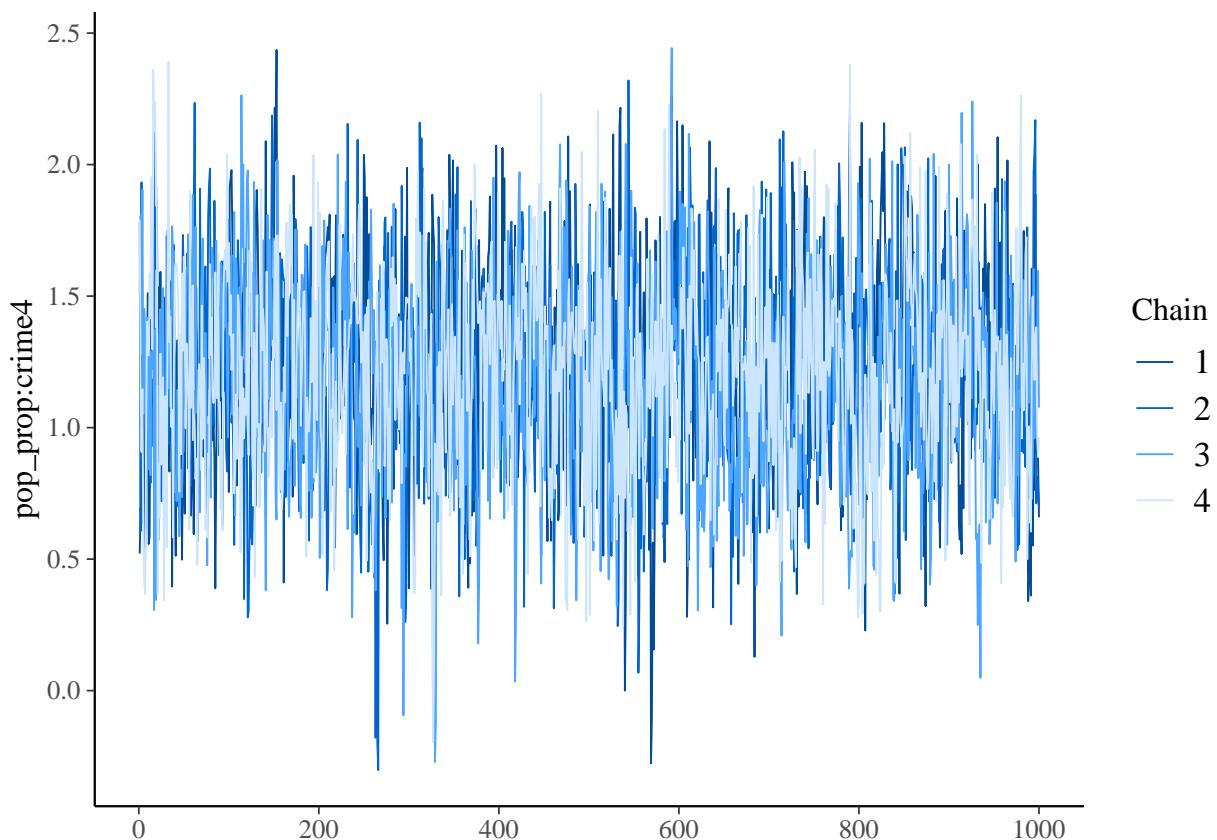
```
plot(model.3.full, "trace", pars = "pop_prop:crime2")
```



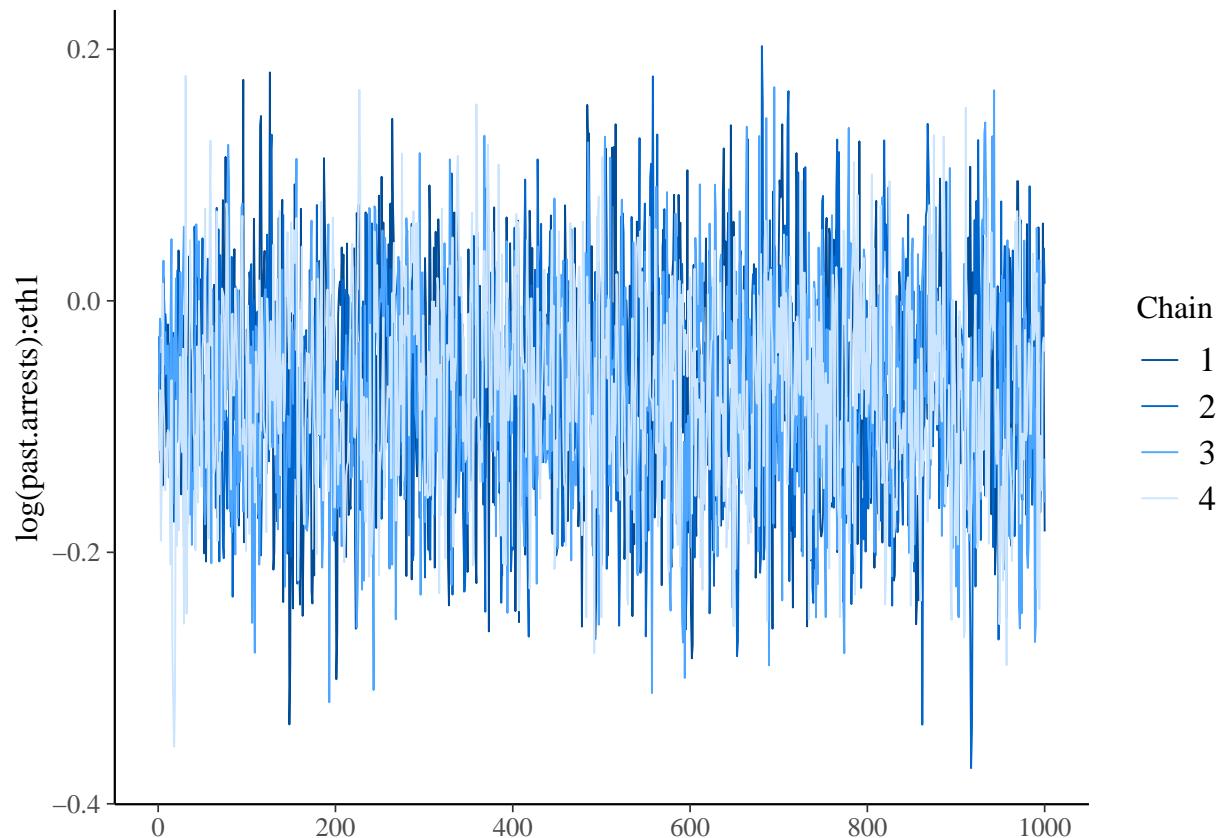
```
plot(model.3.full, "trace", pars = "pop_prop:crime3")
```



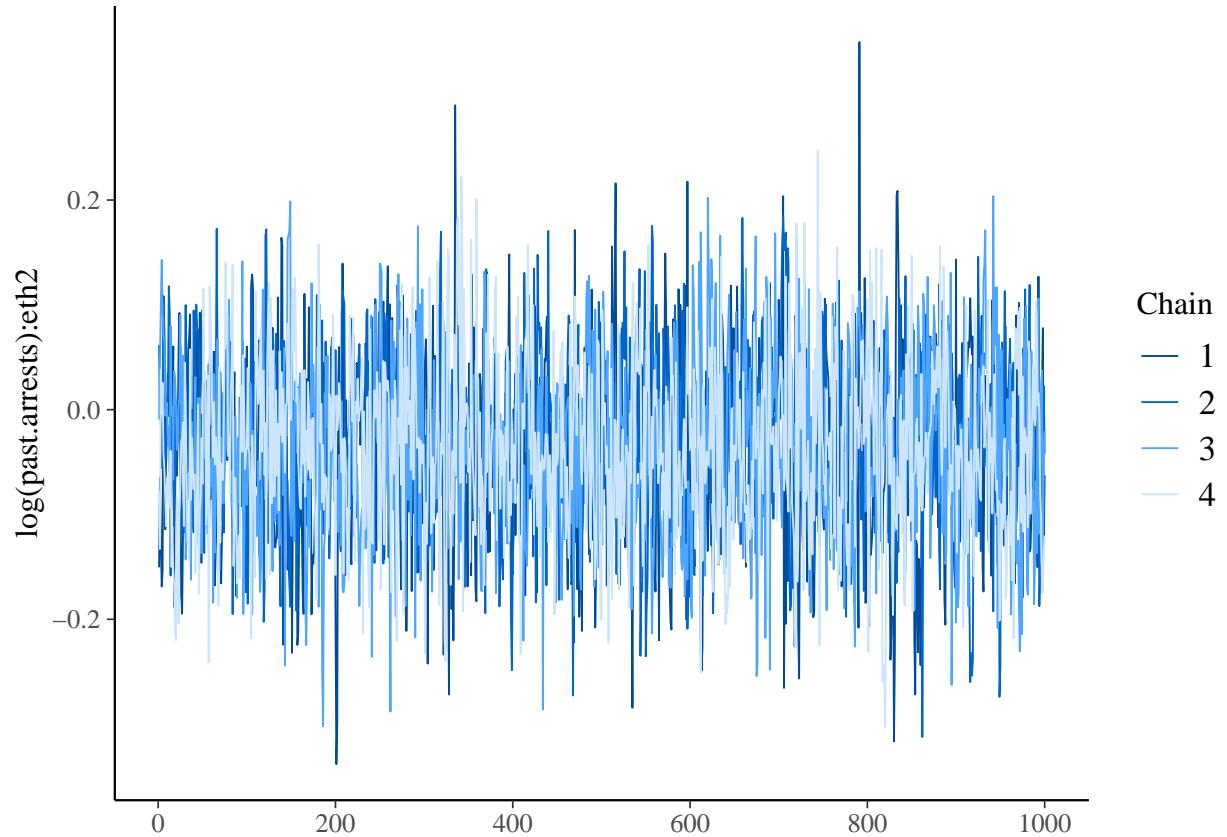
```
plot(model.3.full, "trace", pars = "pop_prop:crime4")
```



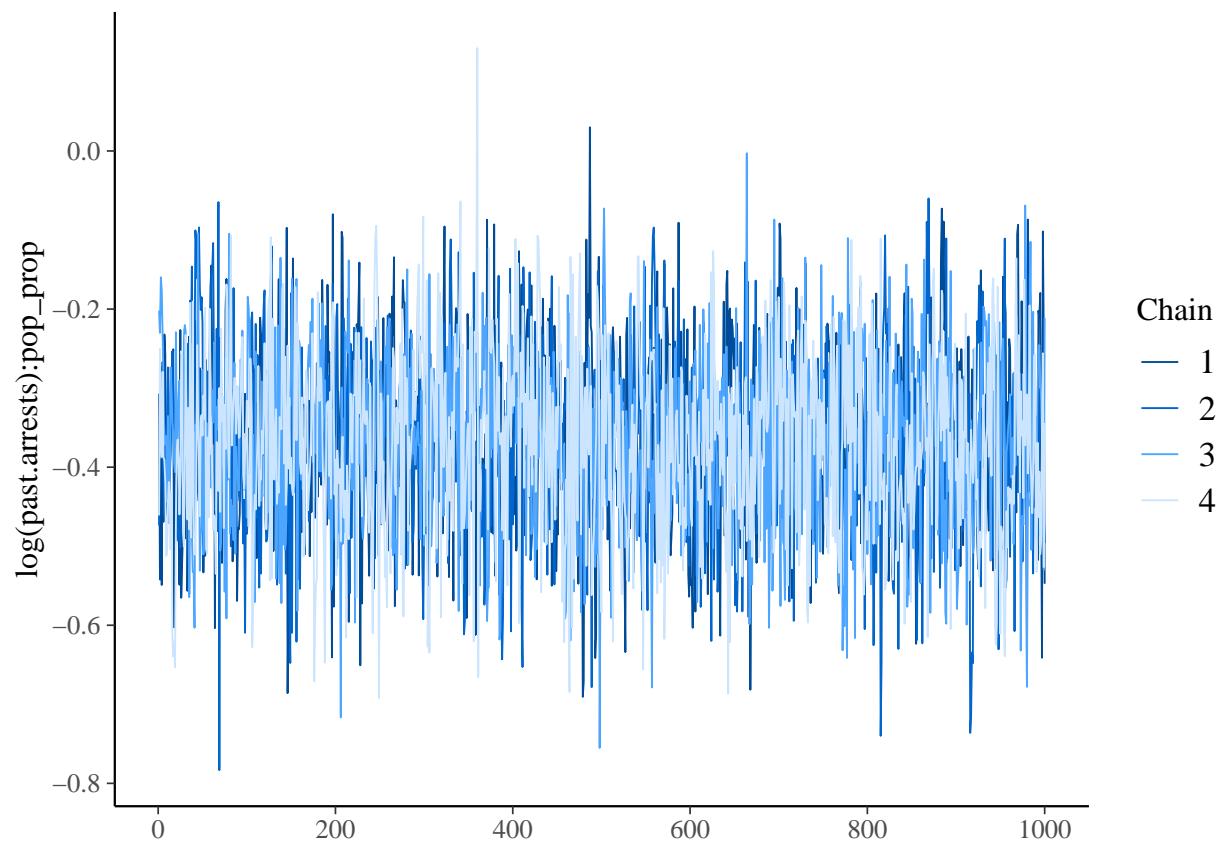
```
plot(model.3.full, "trace", pars = "log(past.arrests):eth1")
```



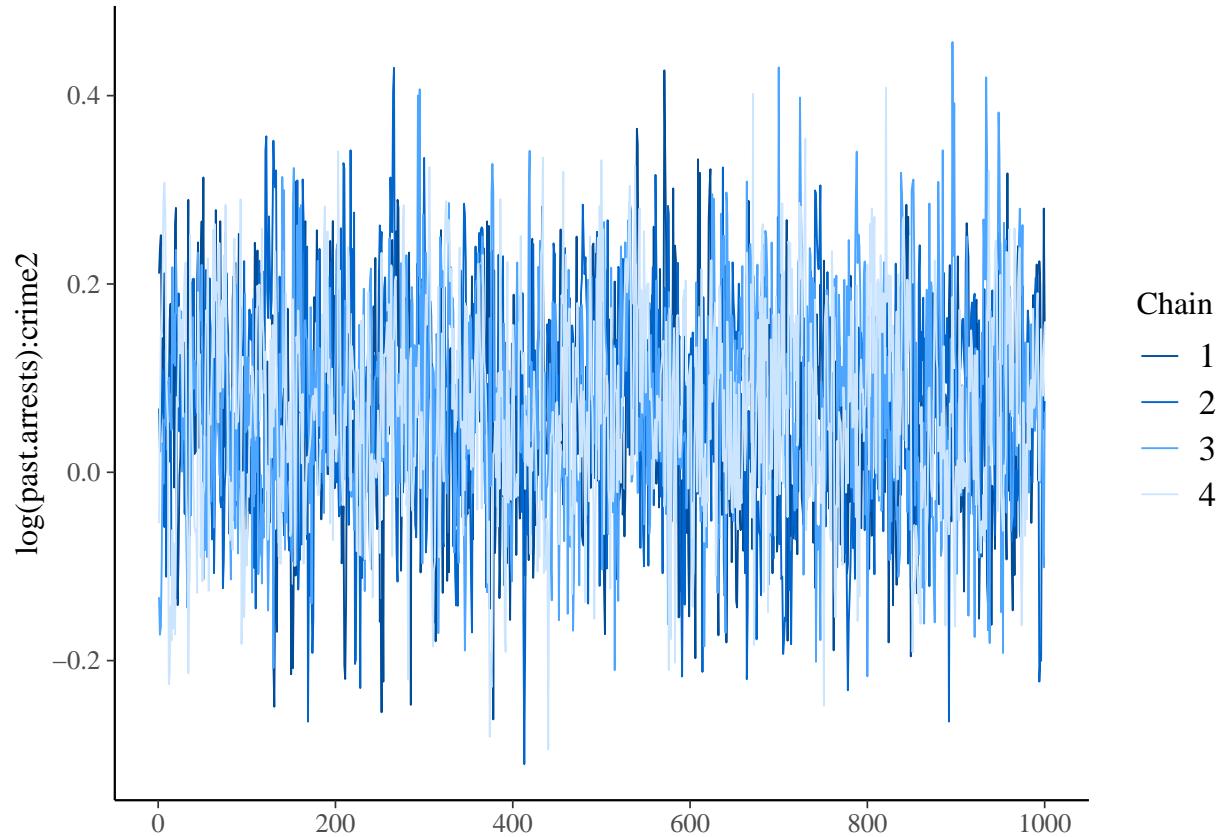
```
plot(model.3.full, "trace", pars = "log(past.arrests):eth2")
```



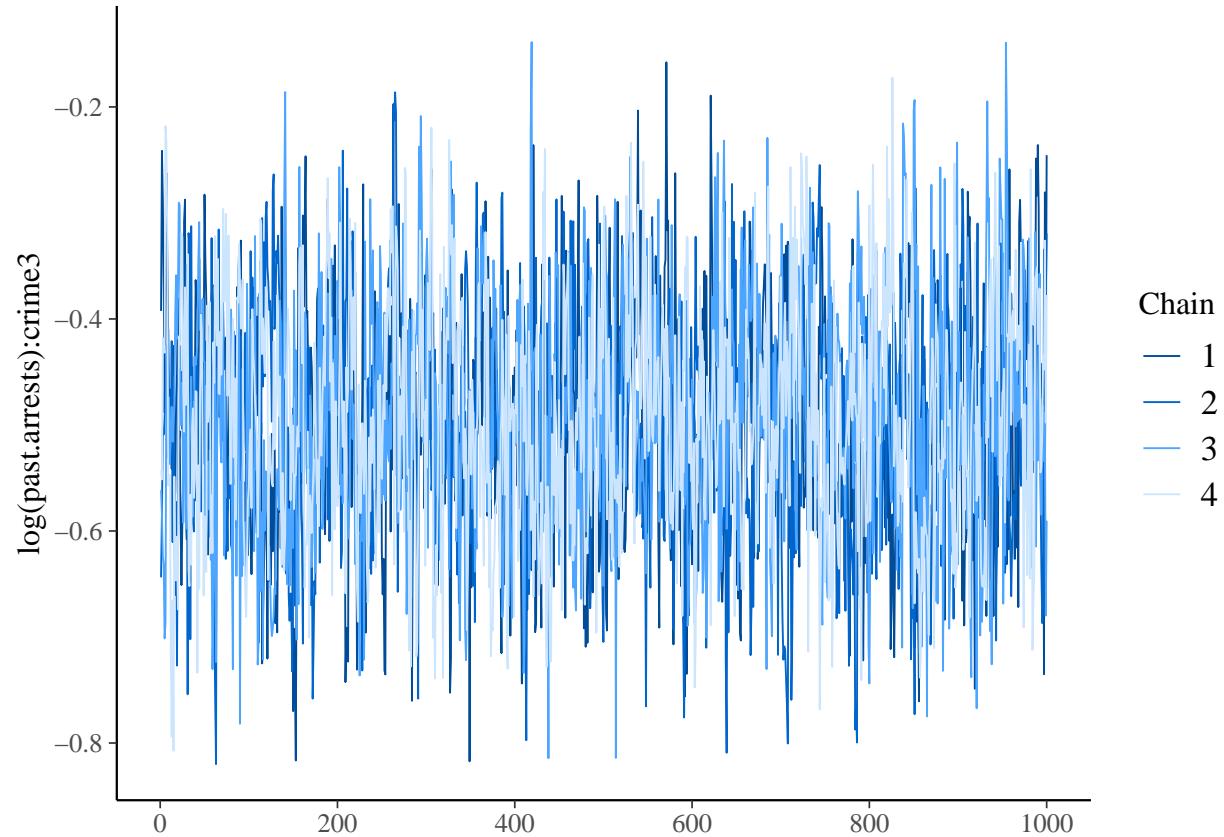
```
plot(model.3.full, "trace", pars = "log(past.arrests):pop_prop")
```



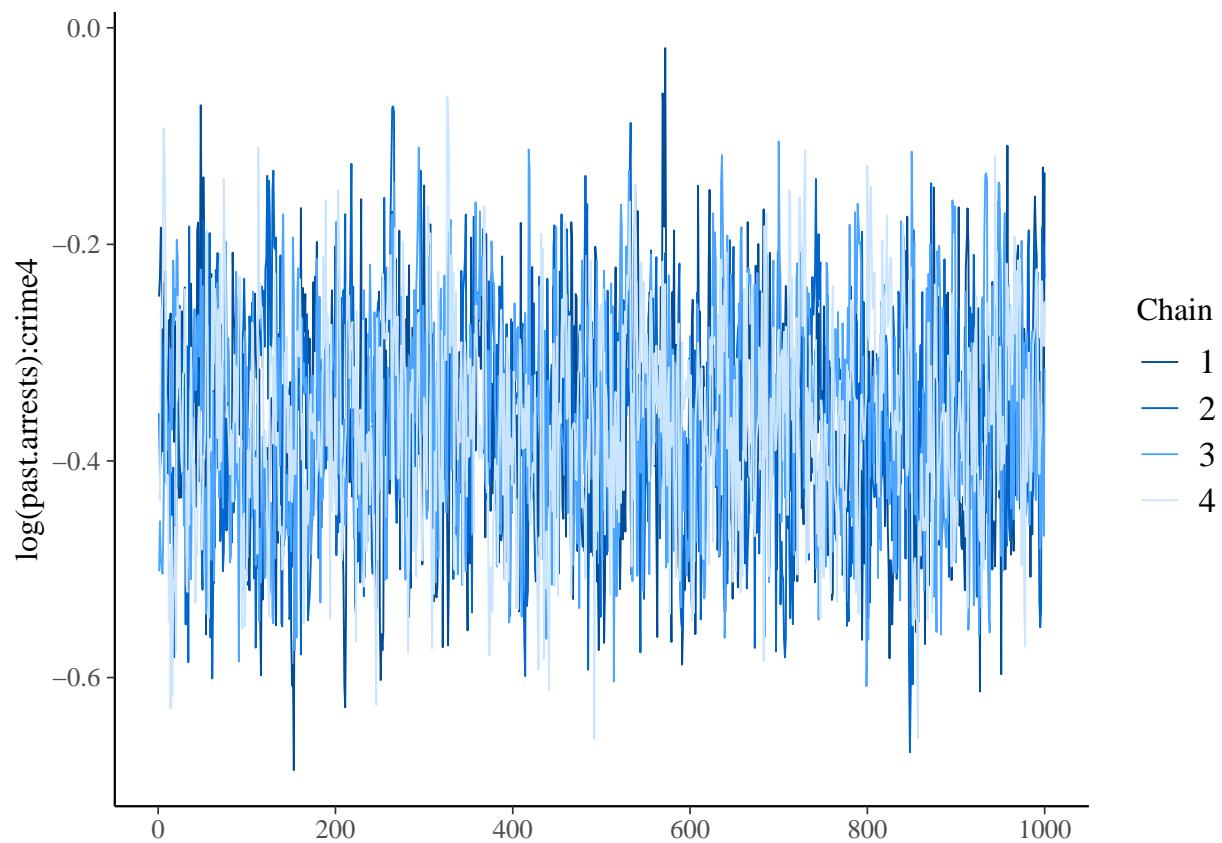
```
plot(model.3.full, "trace", pars = "log(past.arrests):crime2")
```



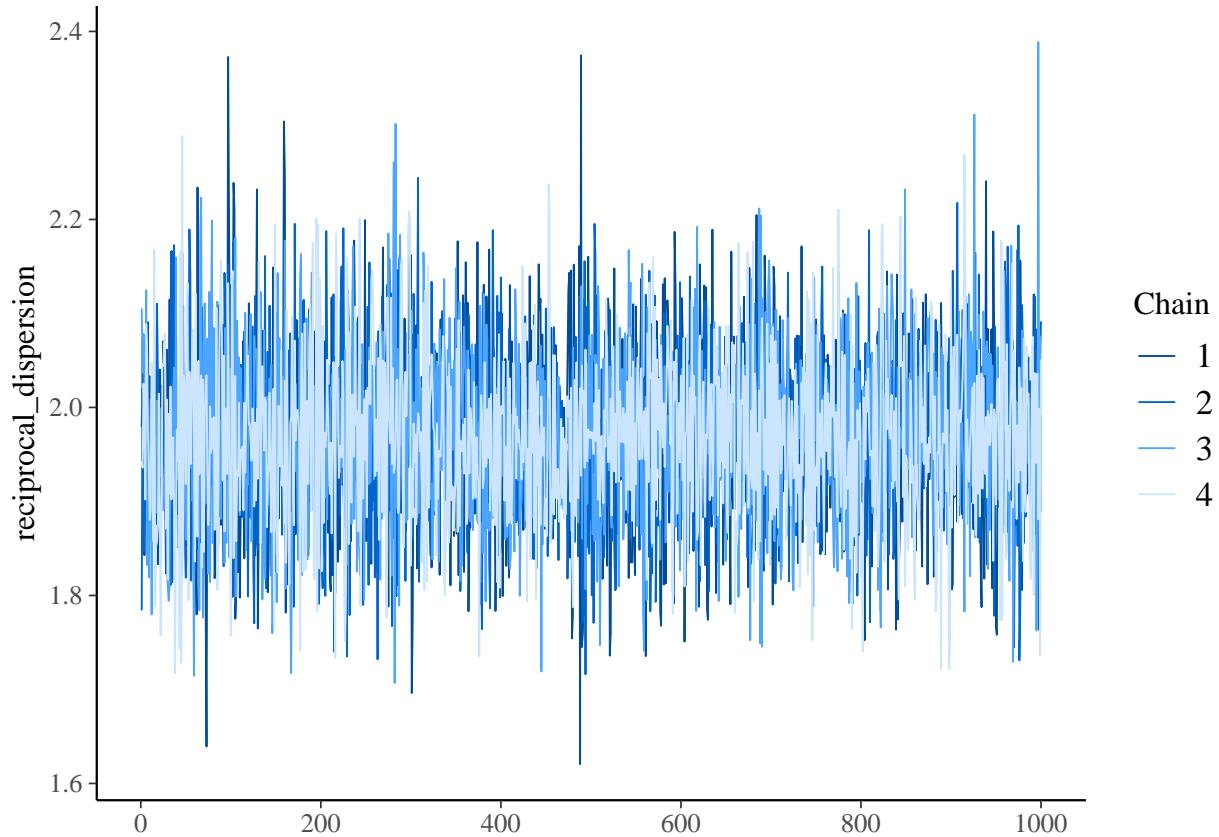
```
plot(model.3.full, "trace", pars = "log(past.arrests):crime3")
```



```
plot(model.3.full, "trace", pars = "log(past.arrests):crime4")
```

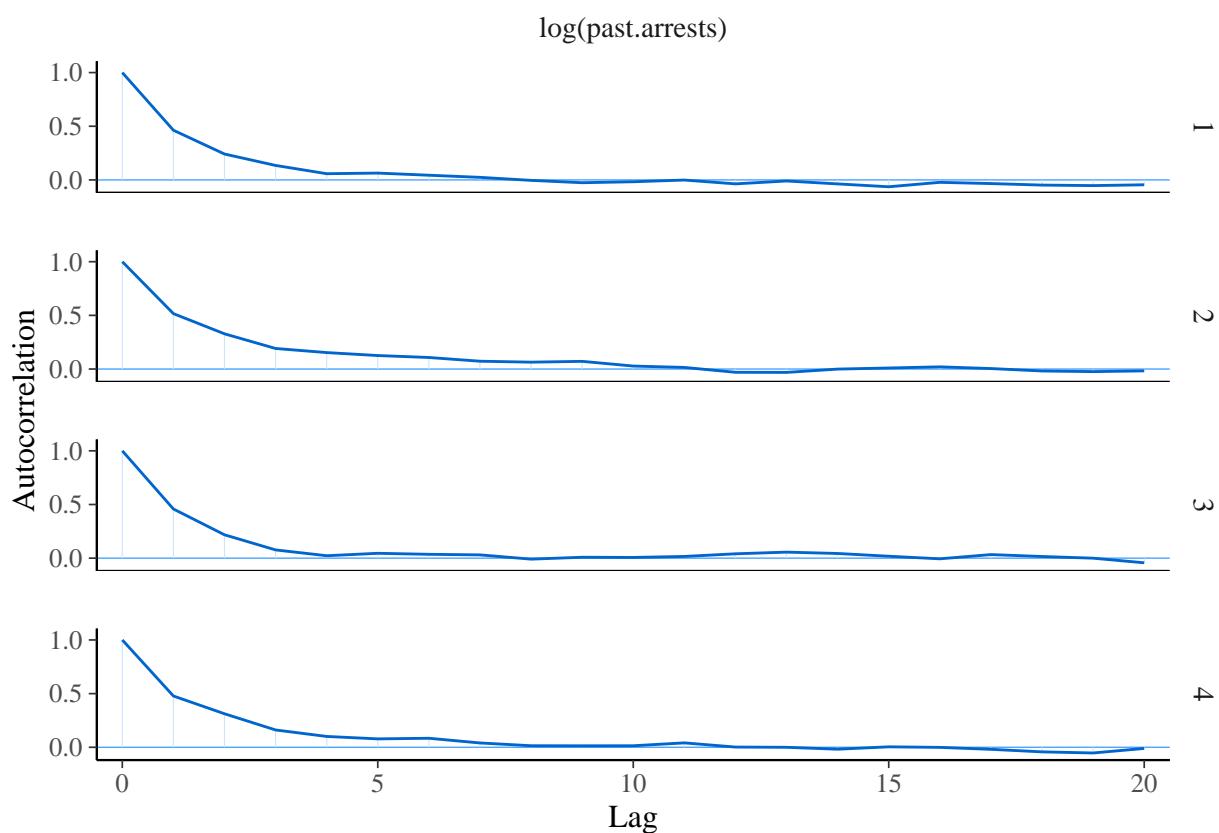


```
plot(model.3.full, "trace", pars = "reciprocal_dispersion")
```

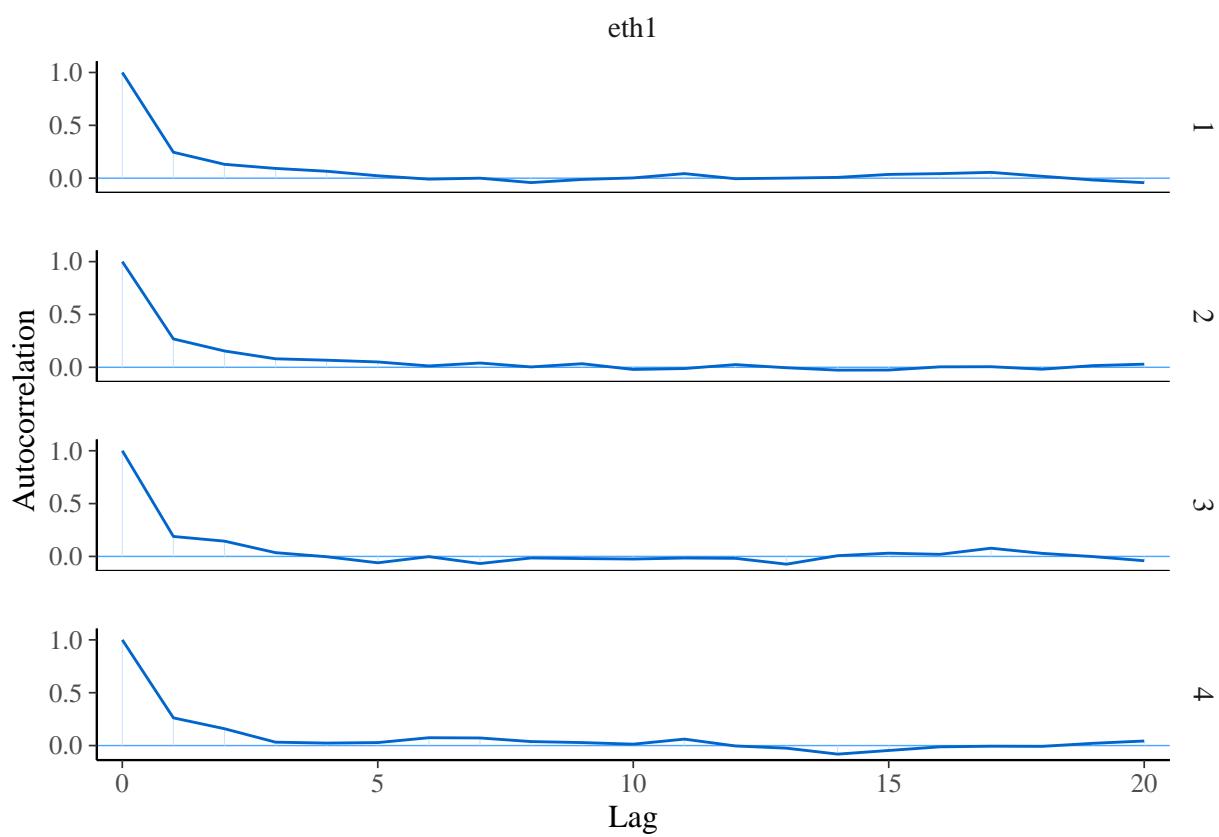


ACF Plots

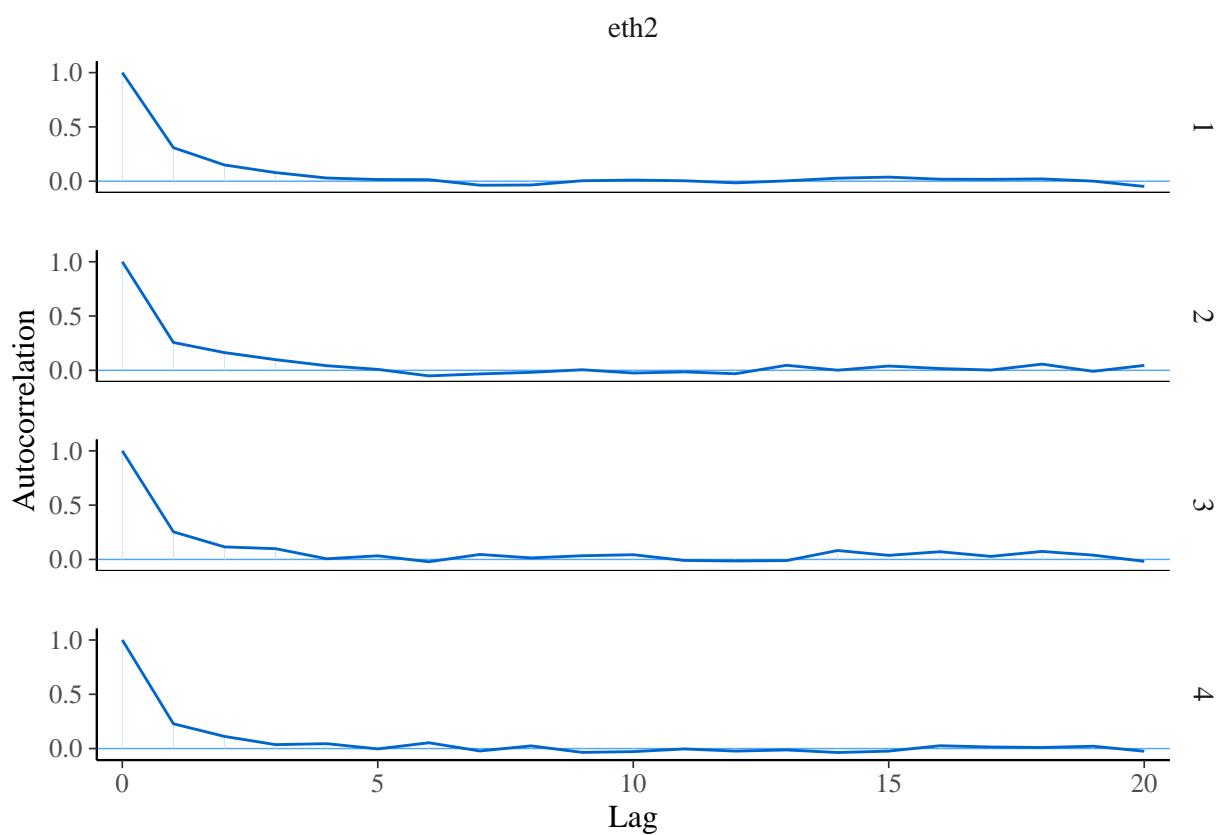
```
plot(model.3.full, "acf", pars = "log(past.arrests)")
```



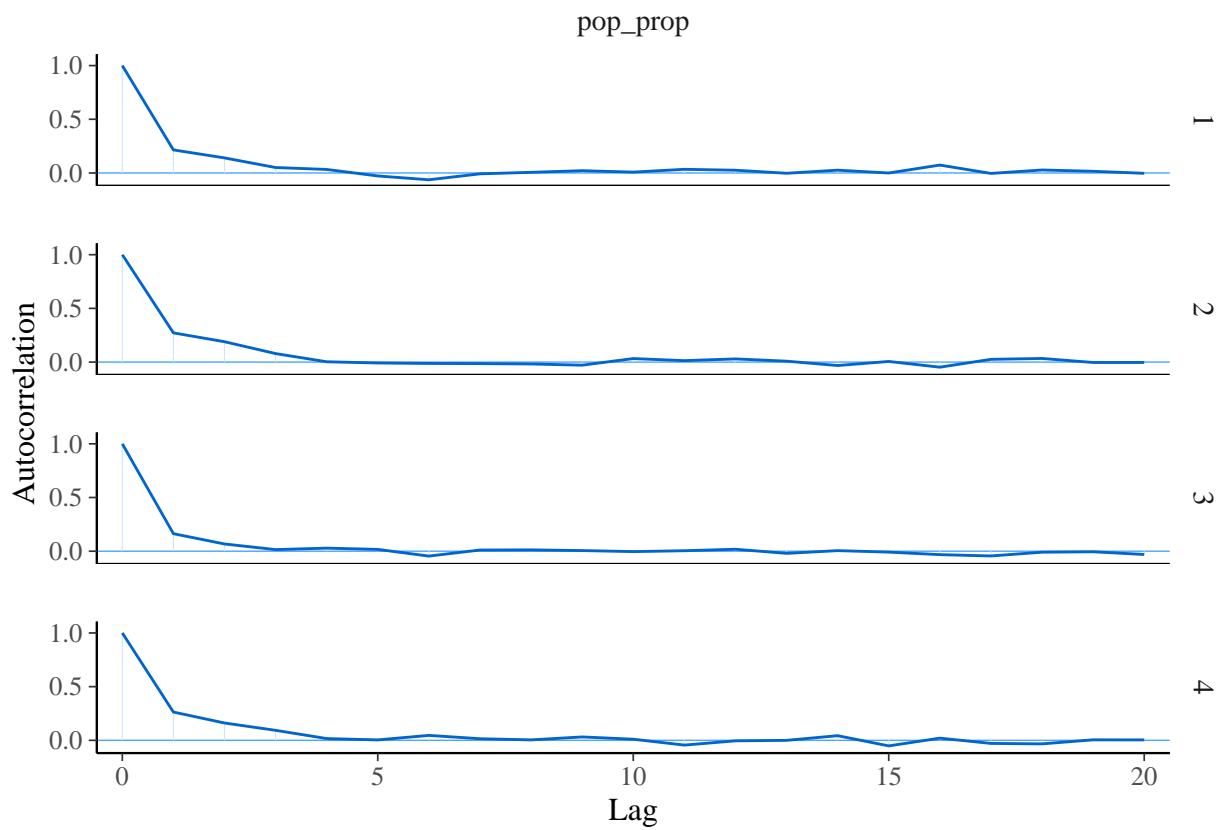
```
plot(model.3.full, "acf", pars = "eth1")
```



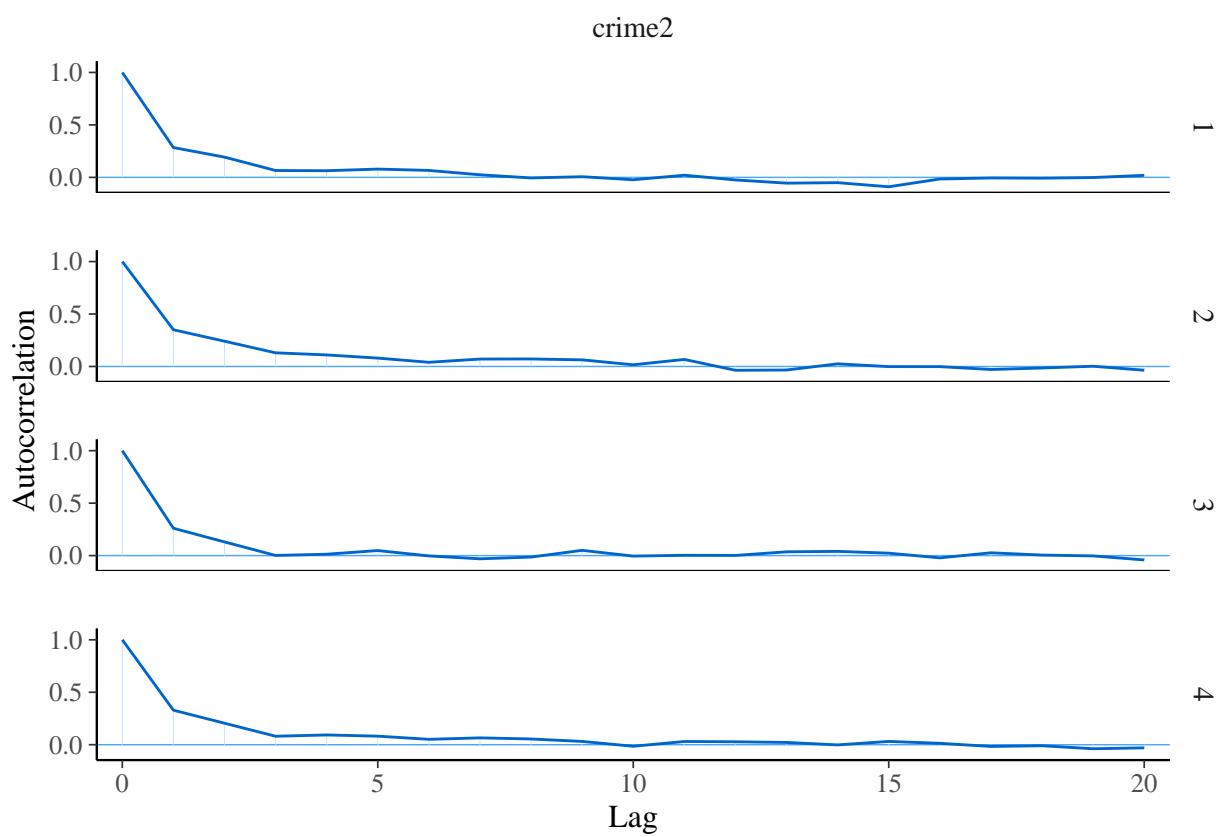
```
plot(model.3.full, "acf", pars = "eth2")
```



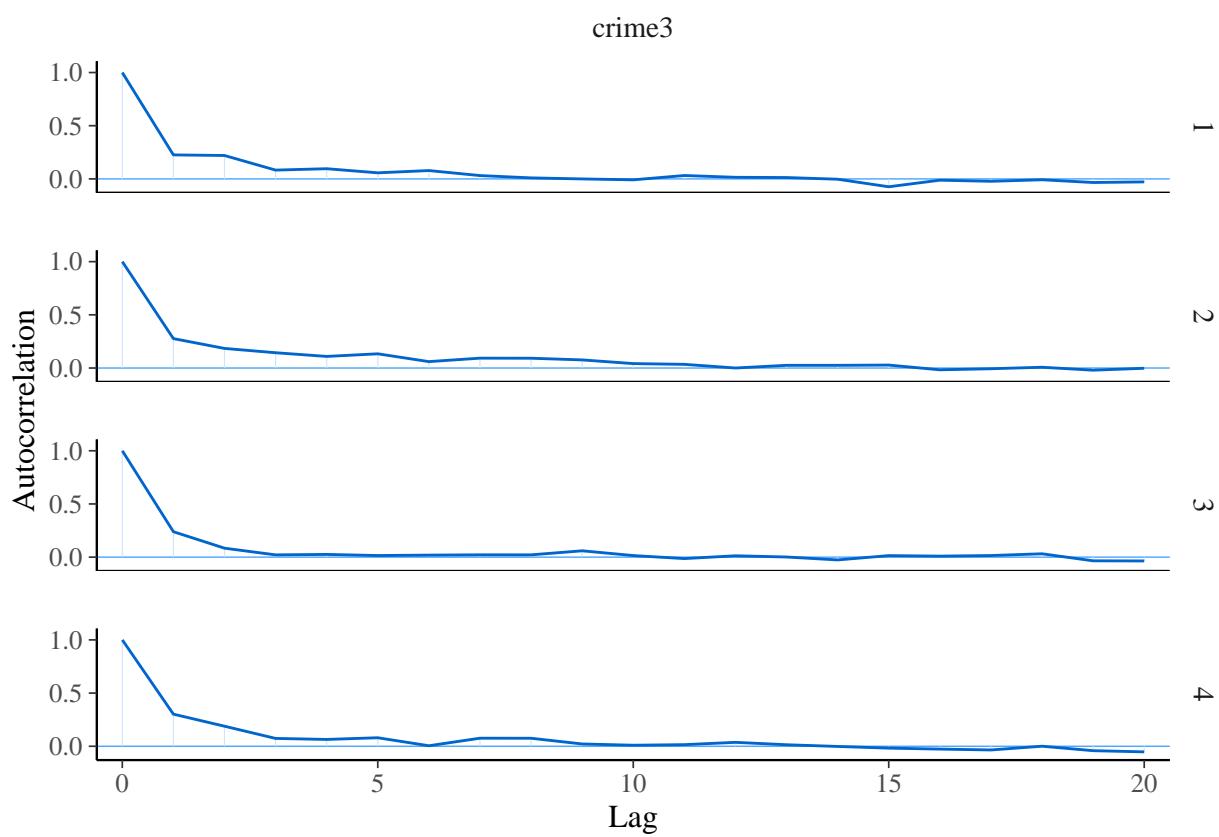
```
plot(model.3.full, "acf", pars = "pop_prop")
```



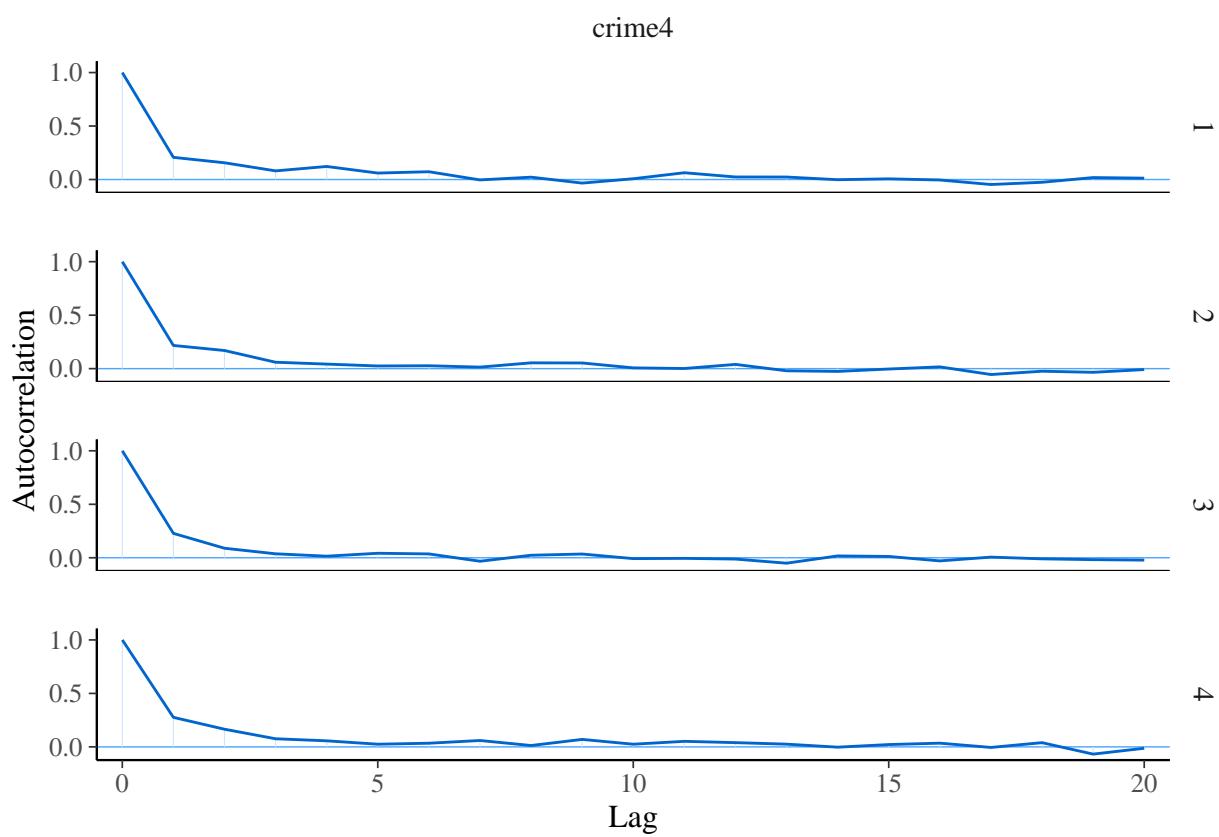
```
plot(model.3.full, "acf", pars = "crime2")
```



```
plot(model.3.full, "acf", pars = "crime3")
```

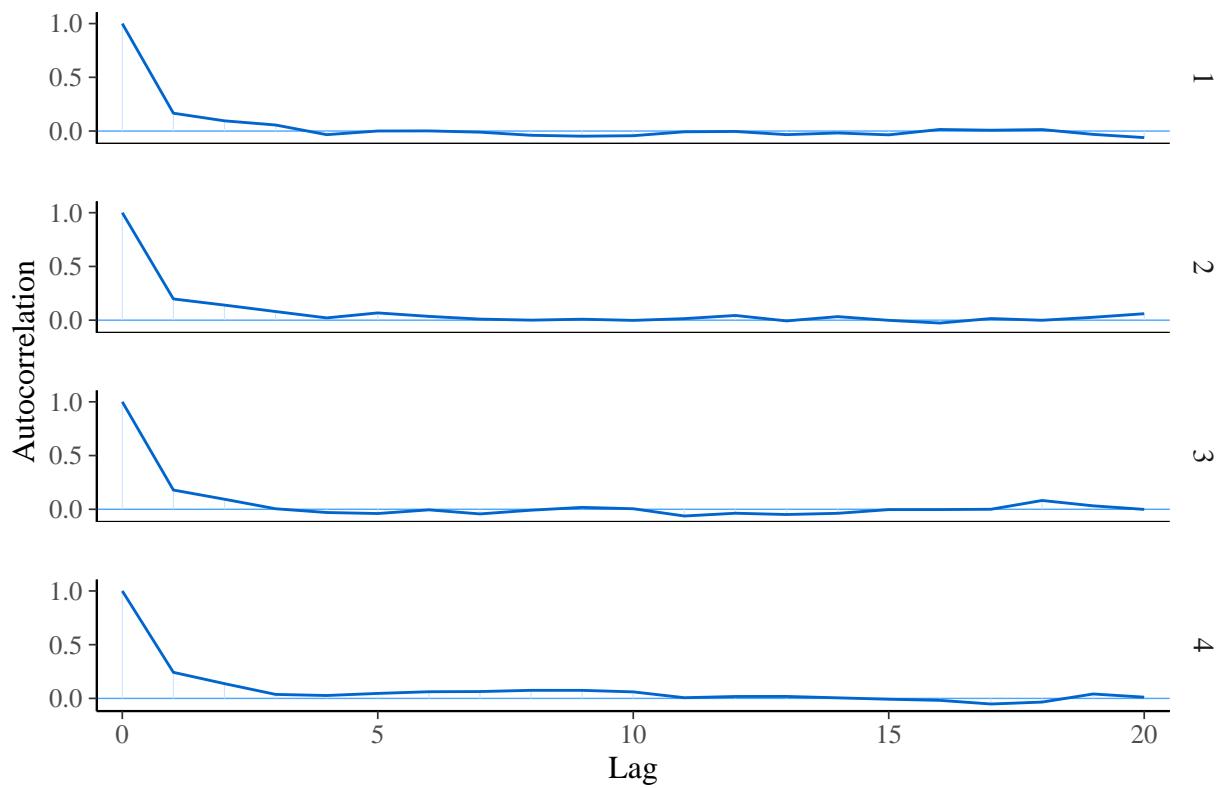


```
plot(model.3.full, "acf", pars = "crime4")
```

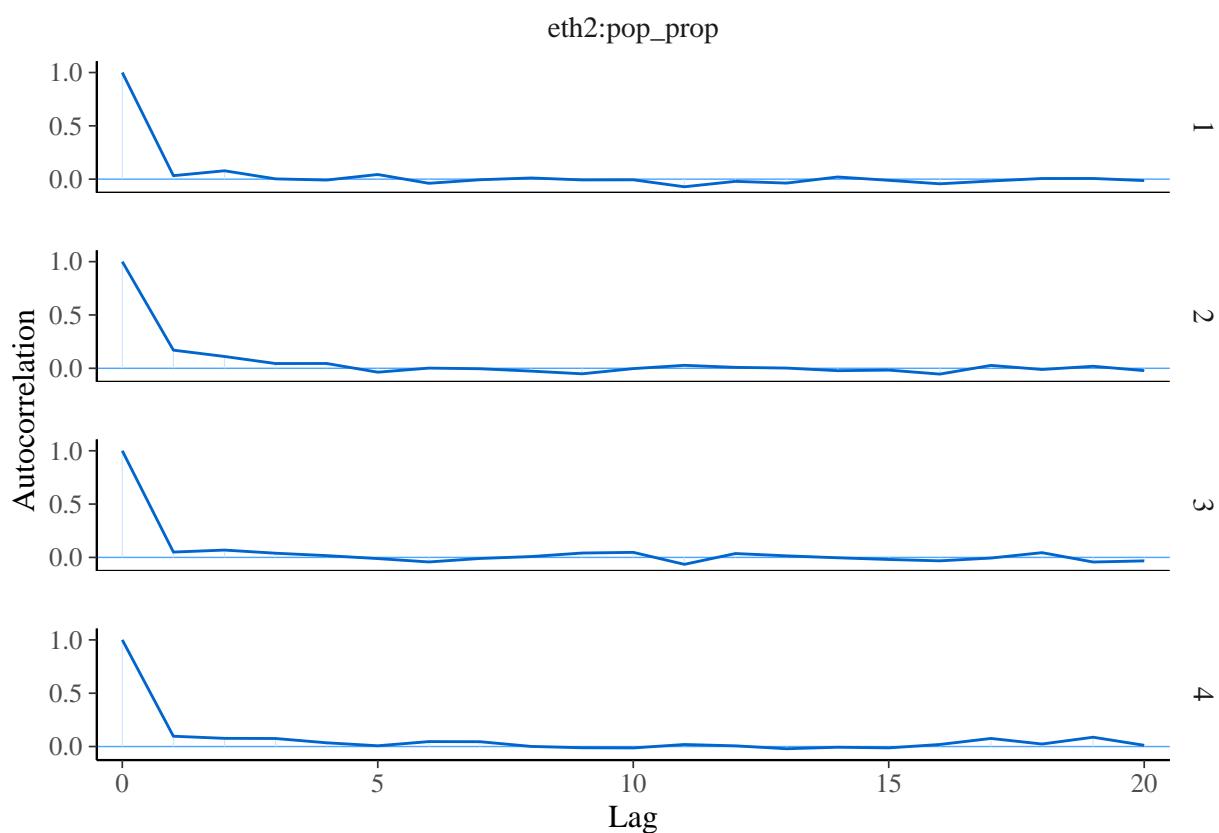


```
plot(model.3.full, "acf", pars = "eth1:pop_prop")
```

eth1:pop_prop

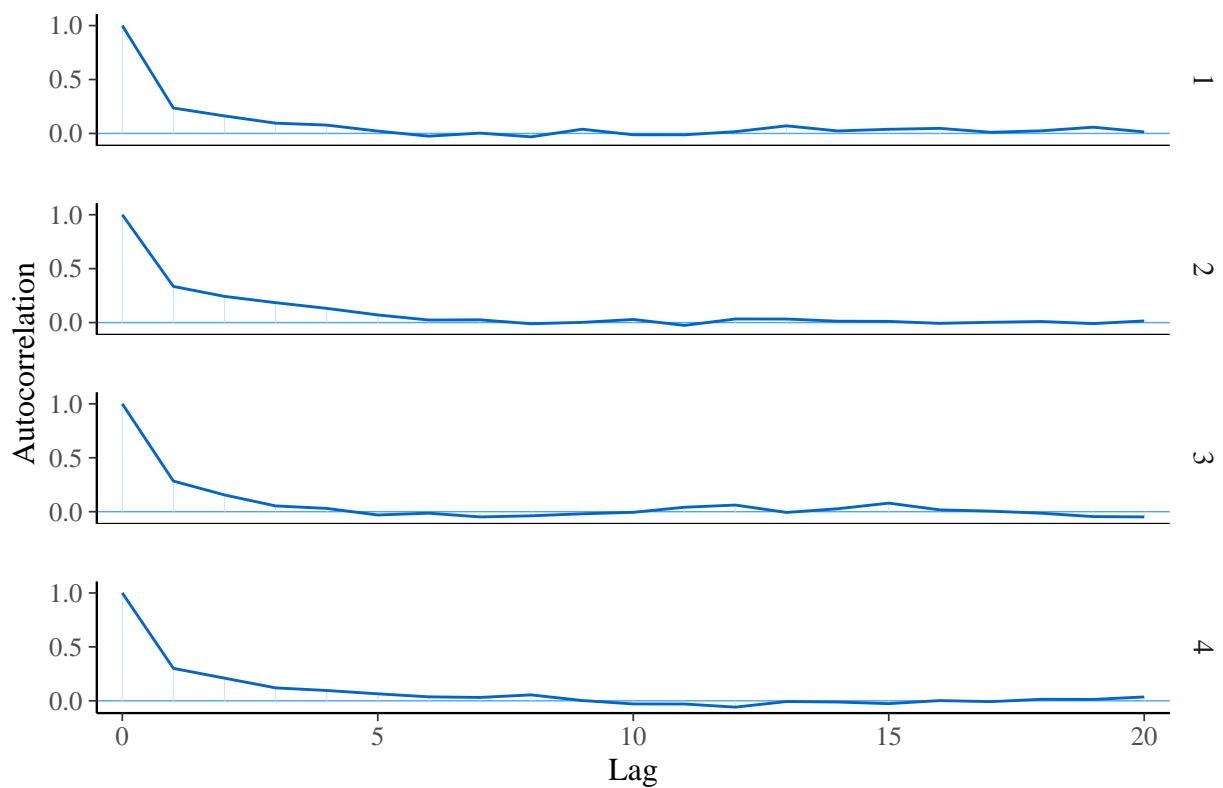


```
plot(model.3.full, "acf", pars = "eth2:pop_prop")
```



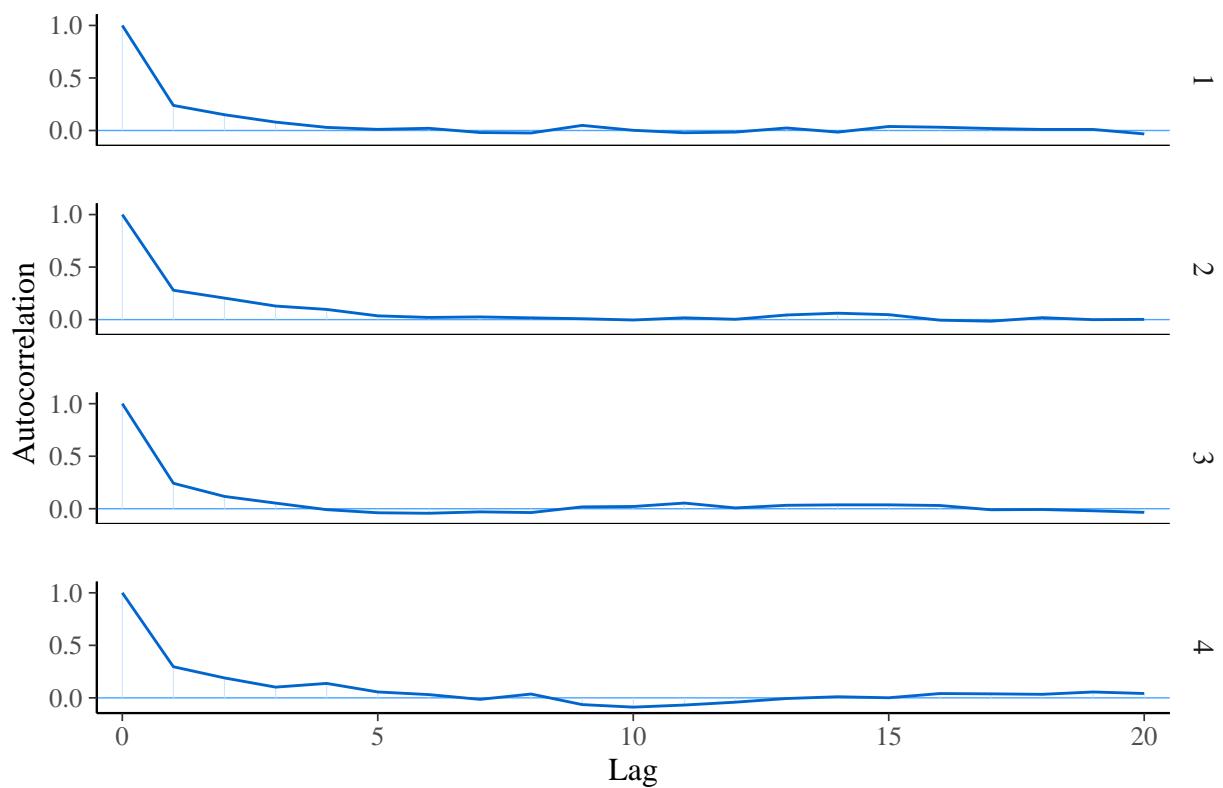
```
plot(model.3.full, "acf", pars = "eth1:crime2")
```

eth1:crime2



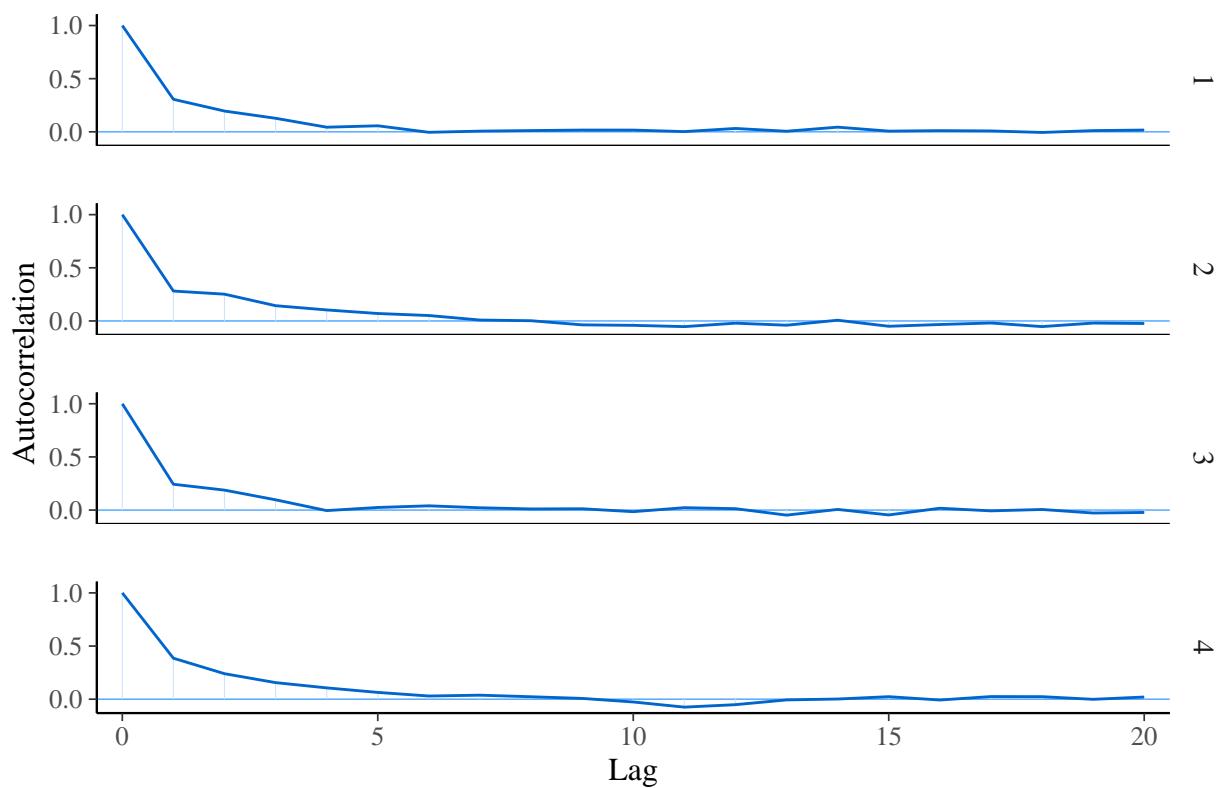
```
plot(model.3.full, "acf", pars = "eth2:crime2")
```

eth2:crime2



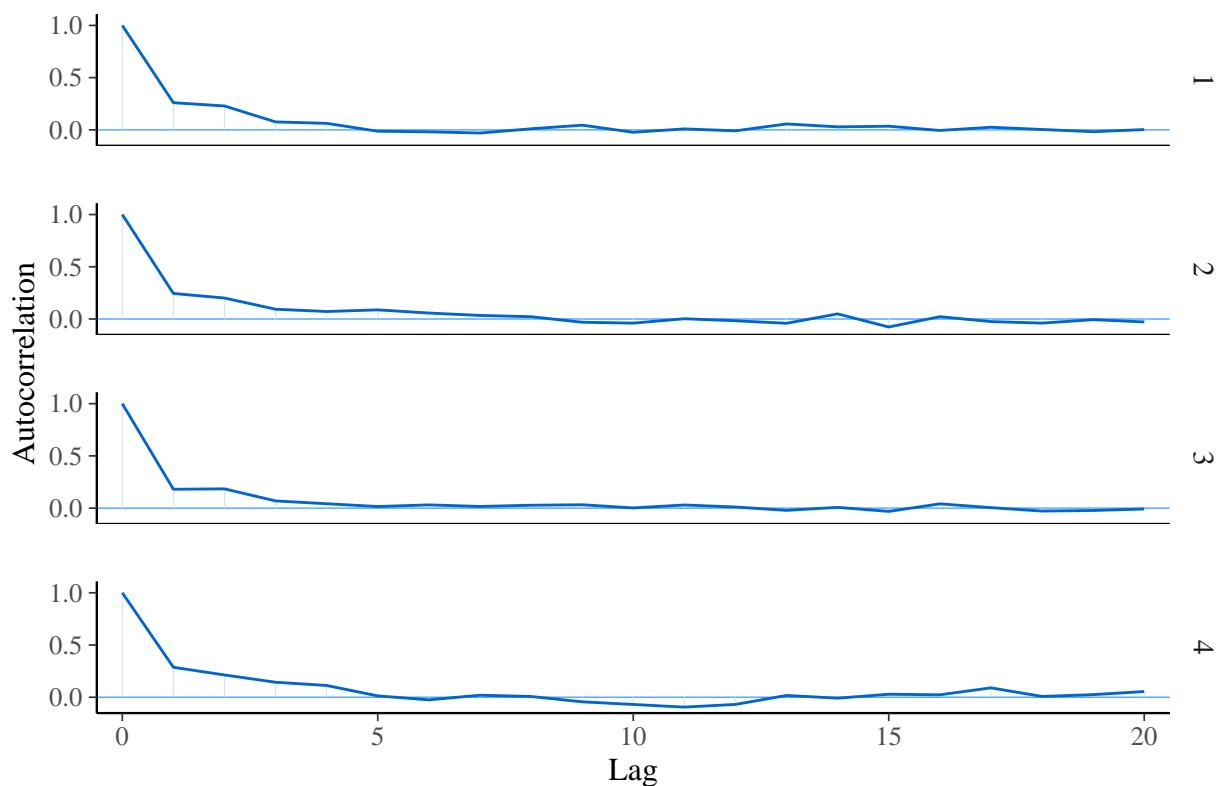
```
plot(model.3.full, "acf", pars = "eth1:crime3")
```

eth1:crime3



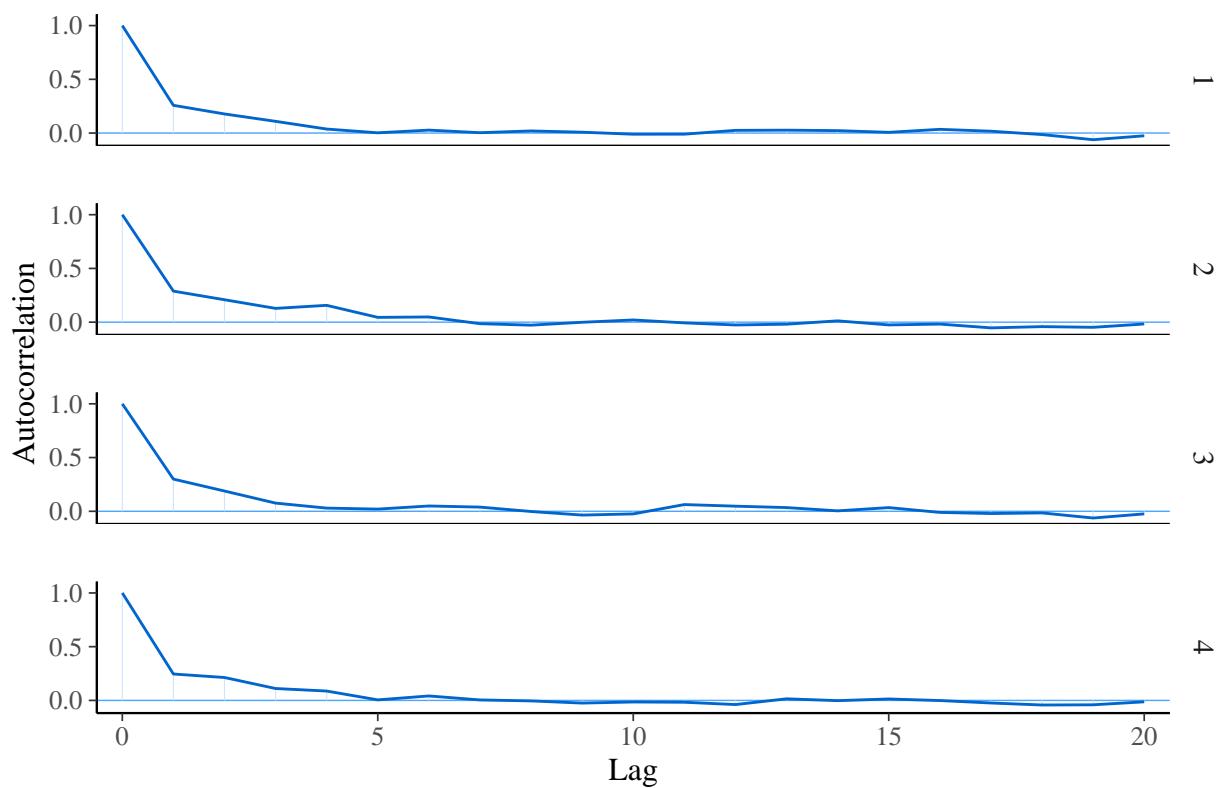
```
plot(model.3.full, "acf", pars = "eth2:crime3")
```

eth2:crime3



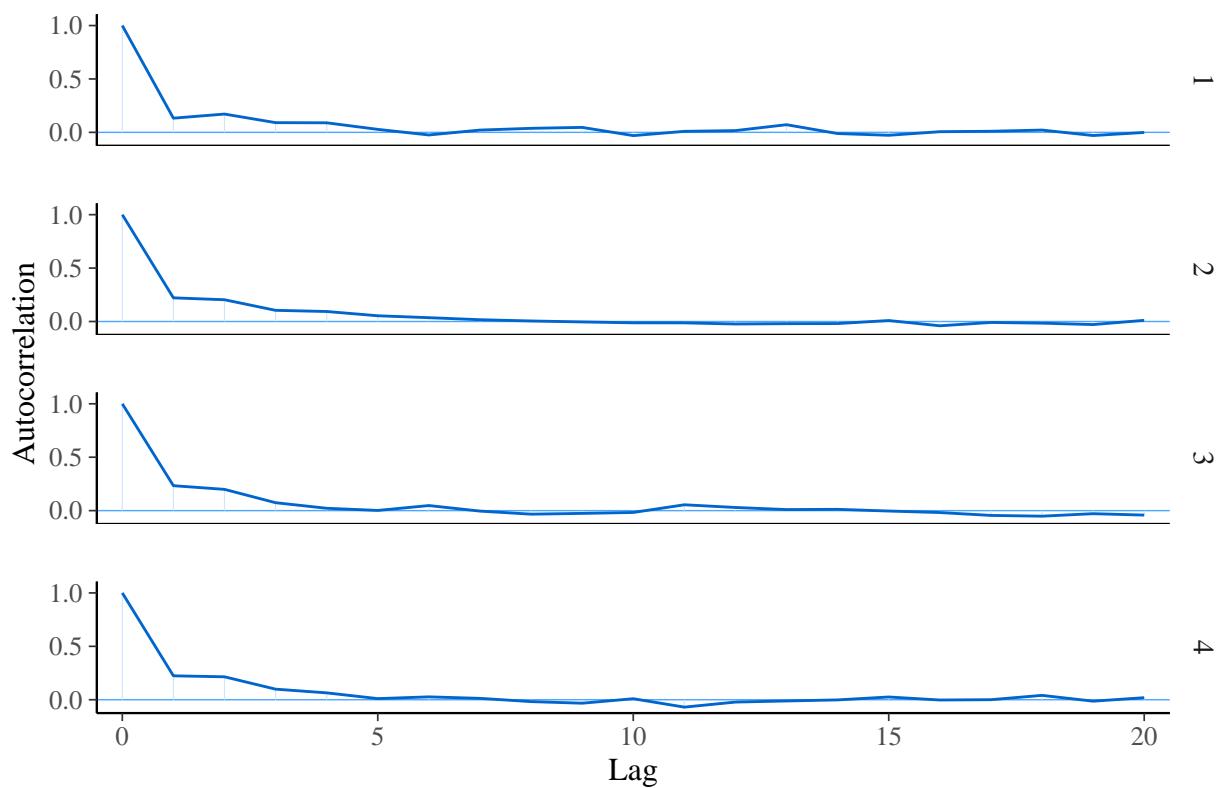
```
plot(model.3.full, "acf", pars = "eth1:crime4")
```

eth1:crime4

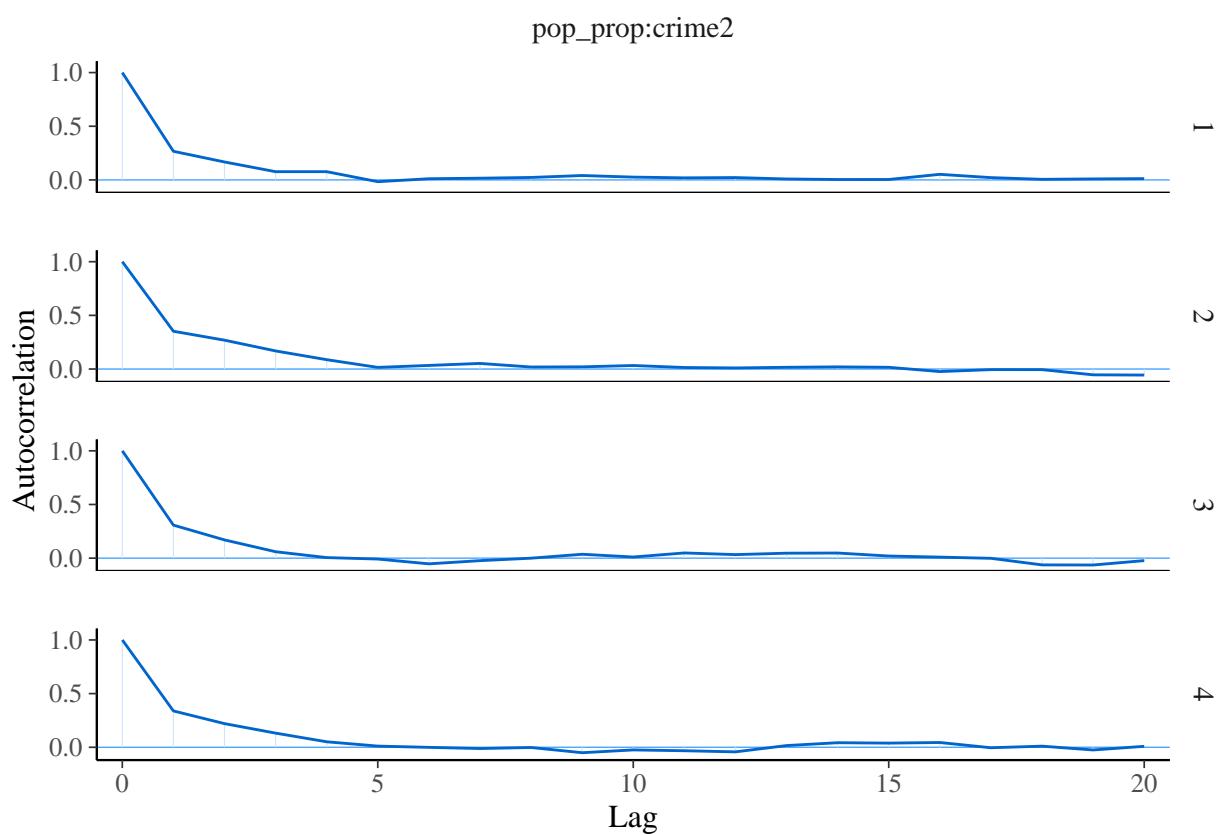


```
plot(model.3.full, "acf", pars = "eth2:crime4")
```

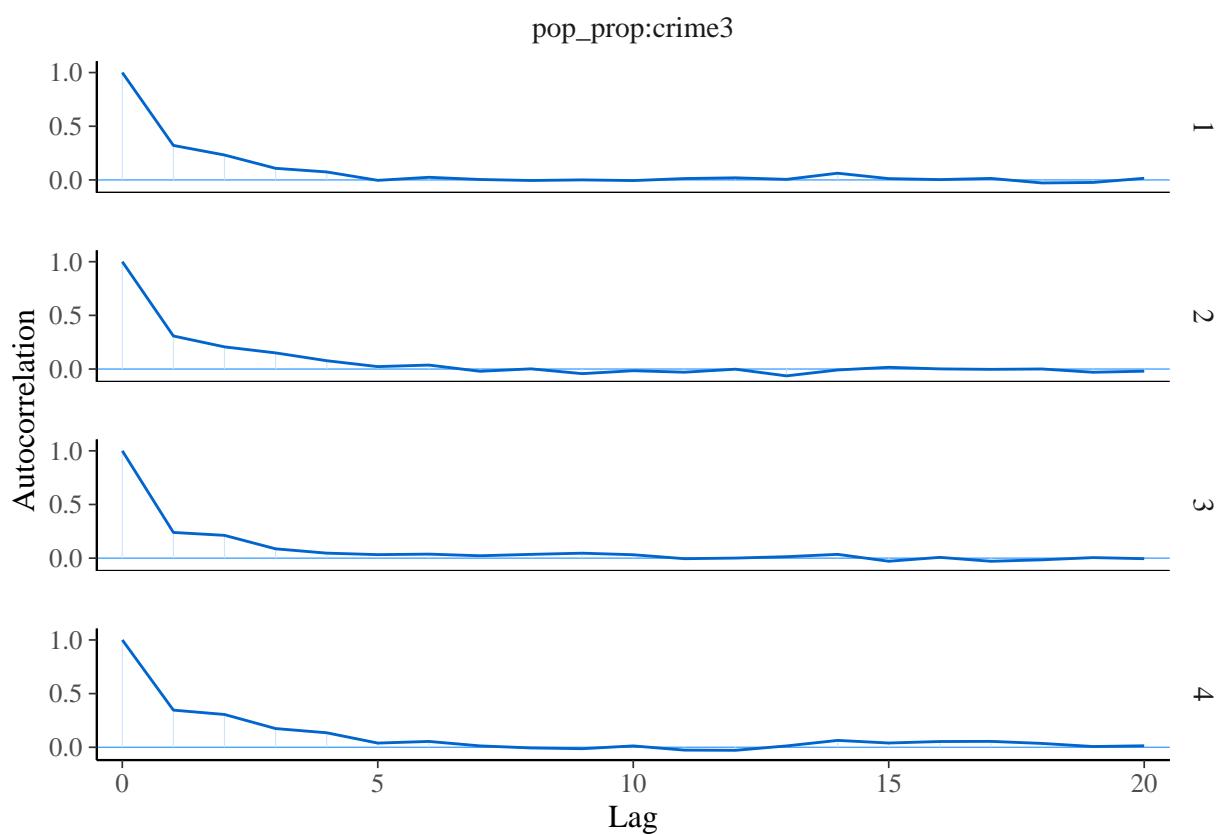
eth2:crime4



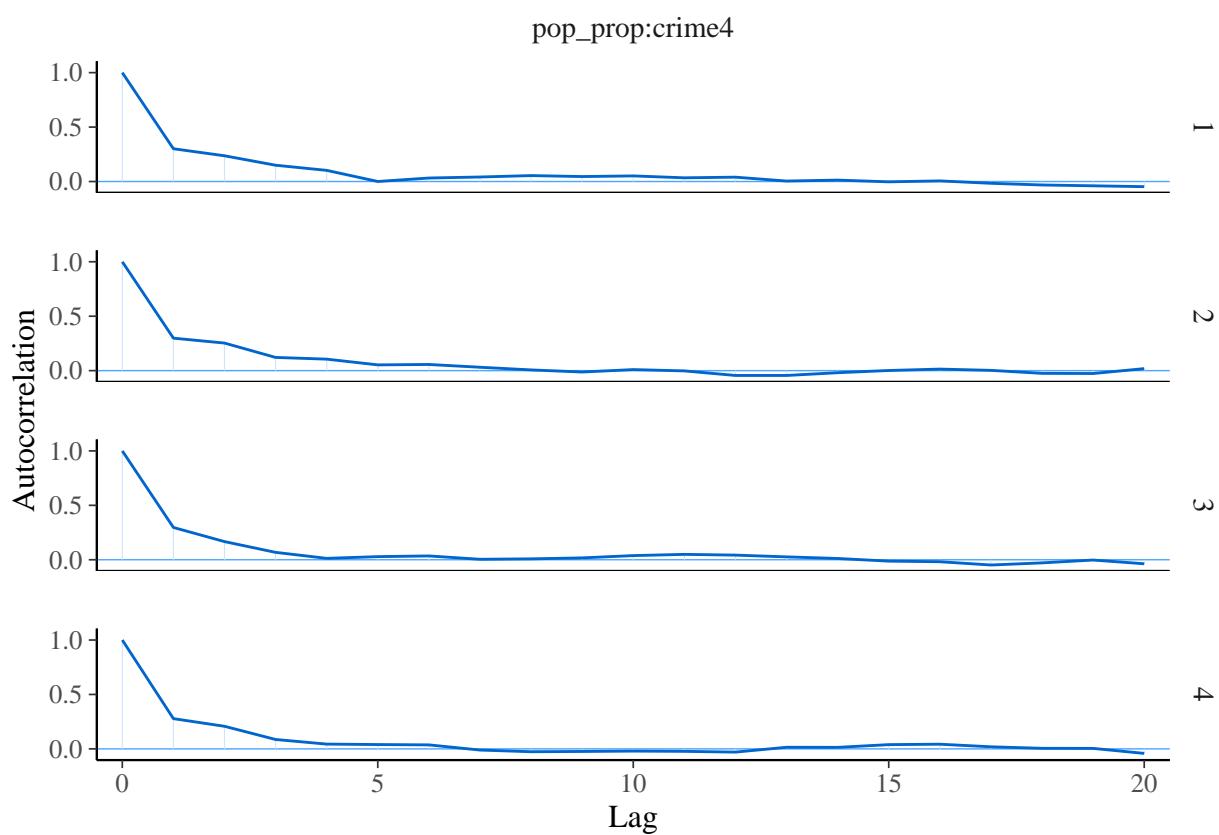
```
plot(model.3.full, "acf", pars = "pop_prop:crime2")
```



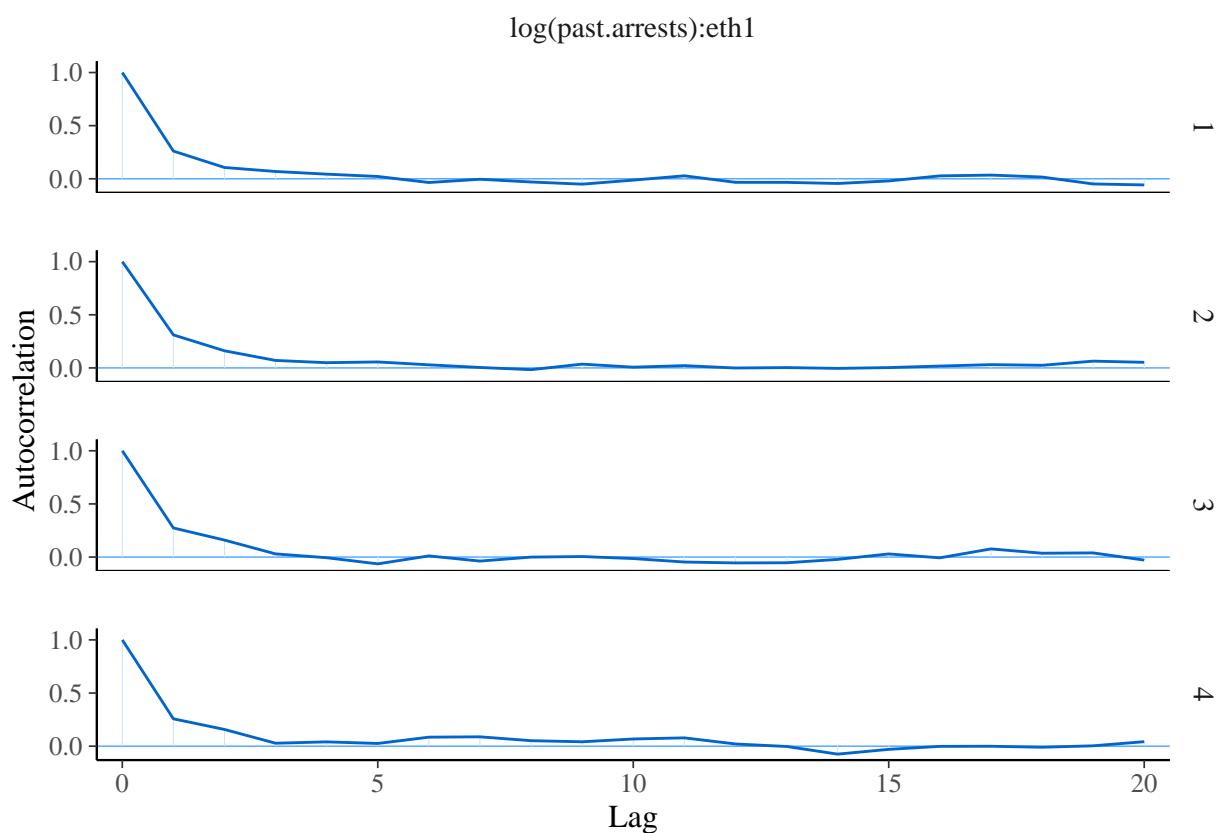
```
plot(model.3.full, "acf", pars = "pop_prop:crime3")
```



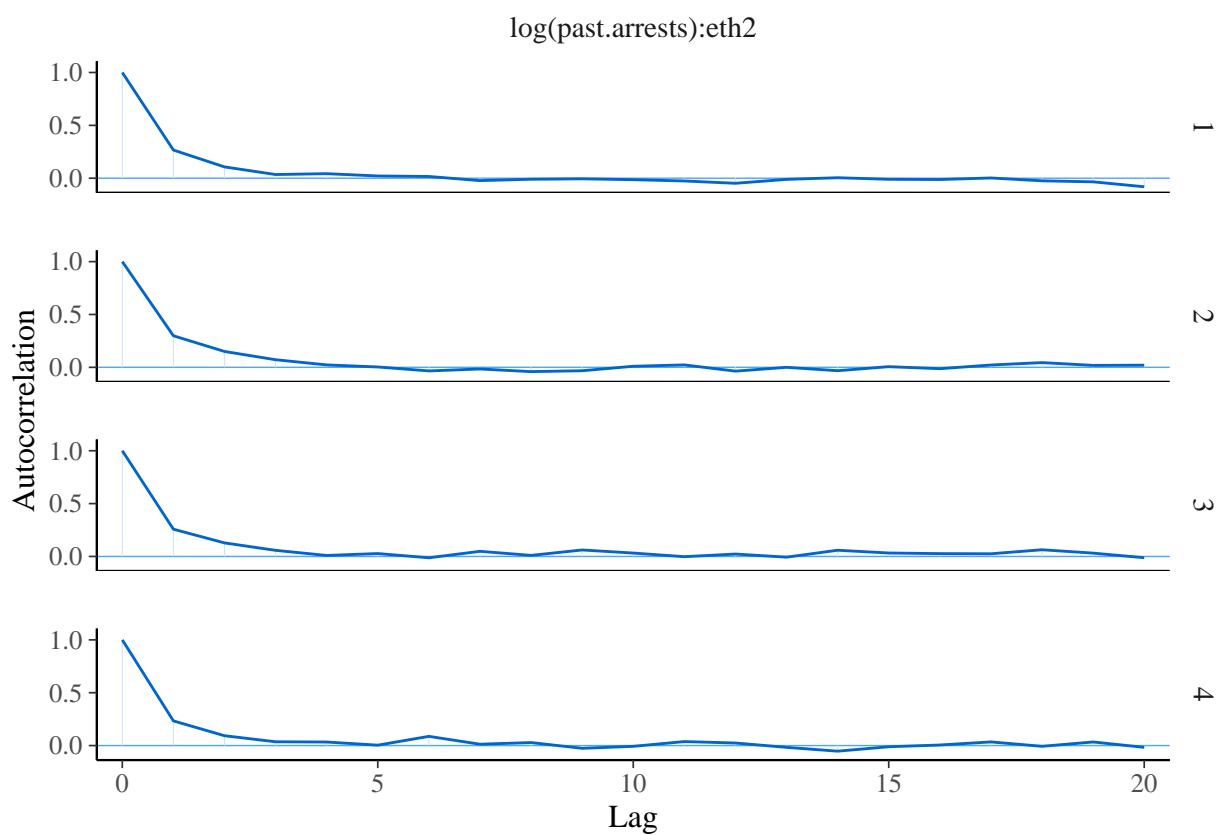
```
plot(model.3.full, "acf", pars = "pop_prop:crime4")
```



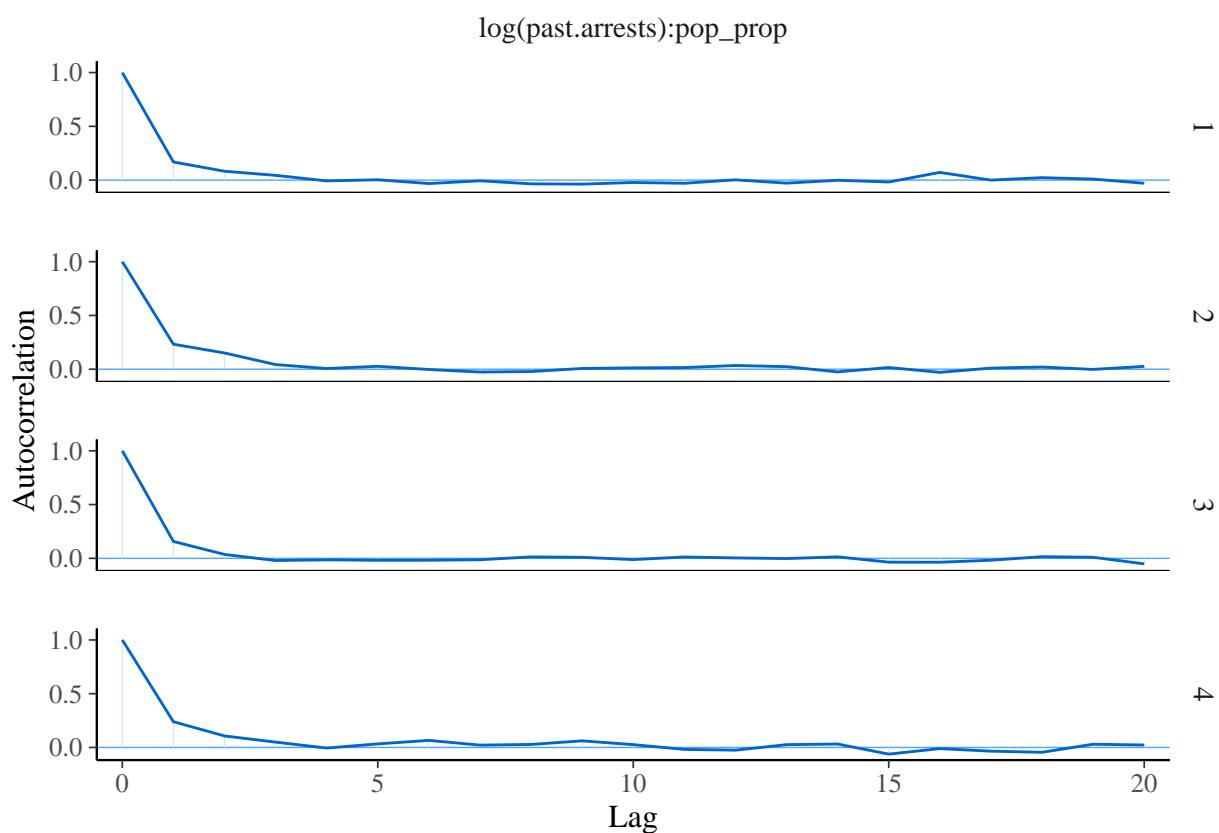
```
plot(model.3.full, "acf", pars = "log(past.arrests):eth1")
```



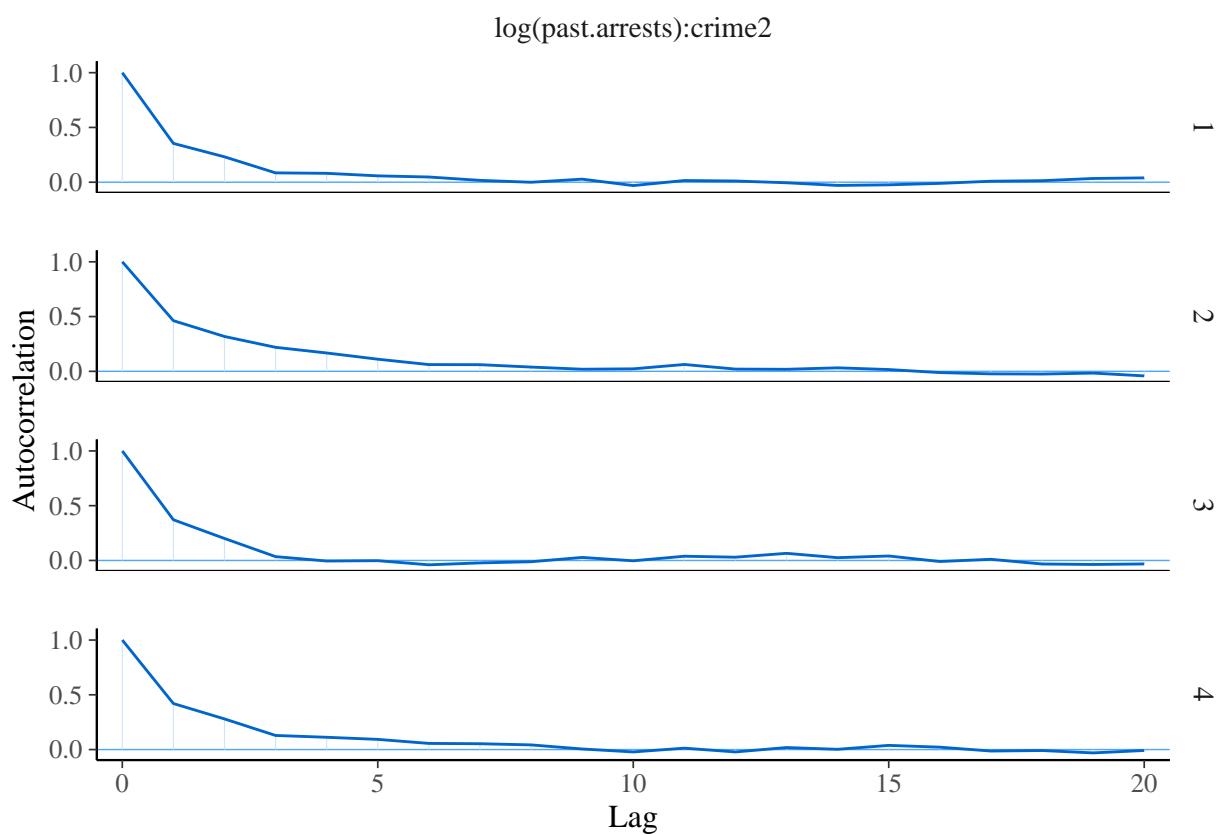
```
plot(model.3.full, "acf", pars = "log(past.arrests):eth2")
```



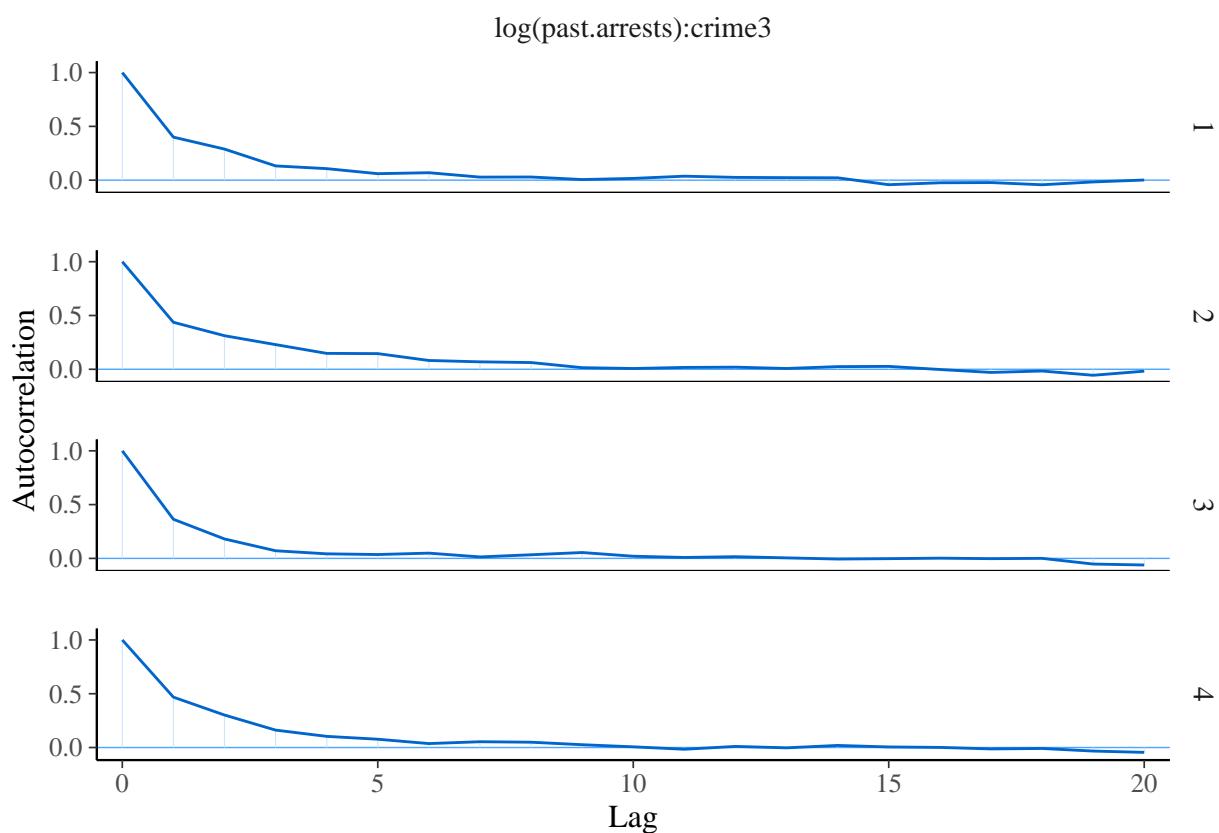
```
plot(model.3.full, "acf", pars = "log(past.arrests):pop_prop")
```



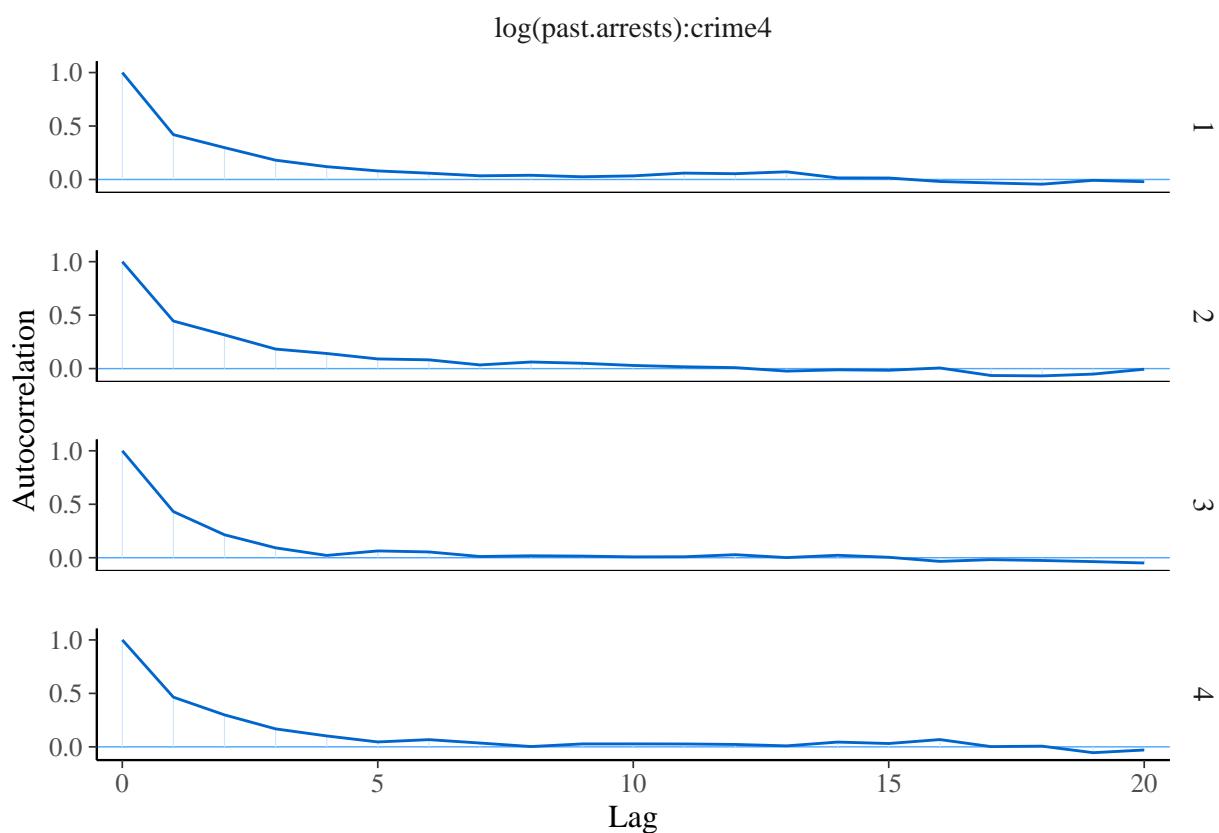
```
plot(model.3.full, "acf", pars = "log(past.arrests):crime2")
```



```
plot(model.3.full, "acf", pars = "log(past.arrests):crime3")
```



```
plot(model.3.full, "acf", pars = "log(past.arrests):crime4")
```



```
plot(model.3.full, "acf", pars = "reciprocal_dispersion")
```

reciprocal_dispersion

