# Chile_prev_rmd

Adele Tyson

2023-06-06

```
library(nleqslv) # Only needed for robince bayesian prevalence
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(gridExtra)
library(readxl)
library(psych)
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.3

##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:psych':
##
##     describe

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(poolr)
```

```
## Warning: package 'poolr' was built under R version 4.2.3
```

```
library(epitools)
```

```
##
## Attaching package: 'epitools'

## The following object is masked from 'package:survival':
##
##     ratetable
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(mltools)
```

```
## Warning: package 'mltools' was built under R version 4.2.3
```

```
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.2.3
```

```
library(rjags)
```

```
## Loading required package: coda

## Linked to JAGS 4.3.1

## Loaded modules: basemod,bugs
```

```
library(rstan)
```

```
## Loading required package: StanHeaders

## rstan (Version 2.21.8, 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)

## Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file

##
## Attaching package: 'rstan'

## The following object is masked from 'package:coda':
##
##     traceplot

## The following object is masked from 'package:psych':
##
##     lookup
```

```
library(posterior)
```

```
## This is posterior version 1.3.1
```

```
##
## Attaching package: 'posterior'
## The following objects are masked from 'package:rstan':
##
##     ess_bulk, ess_tail
## The following objects are masked from 'package:stats':
##
##     mad, sd, var
```

```
library(tidybayes)
library(bayesplot)
```

```
## This is bayesplot version 1.10.0
## - Online documentation and vignettes at mc-stan.org/bayesplot
## - bayesplot theme set to bayesplot::theme_default()
##    * Does _not_ affect other ggplot2 plots
##    * See ?bayesplot_theme_set for details on theme setting
##
## Attaching package: 'bayesplot'
## The following object is masked from 'package:posterior':
##
##     rhat
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::%+%()      masks psych::%+%()
## x ggplot2::alpha()    masks psych::alpha()
## x dplyr::combine()    masks gridExtra::combine()
## x tidyr::extract()    masks rstan::extract()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::lift()       masks caret::lift()
## x tidyr::replace_na() masks mltools::replace_na()
## x dplyr::src()        masks Hmisc::src()
## x dplyr::summarize()  masks Hmisc::summarize()
```

# Bayesian prevalence analysis of autism prevalence in Chile

## Load data

```
chile_merged_raw <- read.csv("04_Data/Data_Chile_Merge.csv") %>% clean_names()

chile_merged <- chile_merged_raw %>%
```

```
  rename(sex_desc = sex,
         year = agno,
         school_code = rbd,
         school_check_code = dgv_rbd,
         school_name = nom_rbd,
         school_region_code = cod_reg_rbd,
         school_region_name_abr = nom_reg_rbd_a,
         school_province_code = cod_pro_rbd,
         school_commune_code = cod_com_rbd,
         school_commune_name = nom_com_rbd,
         school_dept_code = cod_deprov_rbd,
         school_dept_name = nom_deprov_rbd,
         school_dependency_code = cod_depe, # has categories 1-6, no1 and no2 here are no1 in grouped
         school_dependency_code_grouped = cod_depe2, # has categories 1-5
         school_rurality_code = rural_rbd,
         school_operation_status = estado_estab,
         teaching_code1 = cod_ense, # min = 10, max = 910, eg preschool, special education hearing impa
         teaching_code2 = cod_ense2, # subject matter coding, 1-8
         teaching_code3 = cod_ense3, # age based coding, 1-7
         grade_code1 = cod_grado, # grade of schooling, 1-10, 21-25, 31-34, nests in teaching_code1
         grade_code2 = cod_grado2, # equivalent grade of schooling for adult special education, 1-8, 99
         grade_letter = let_cur, # refers to the class within the grade, close to start of alphabet is
         course_timing = cod_jor, # time of day, morning, afternoon, both, night, no info
         course_type = cod_tip_cur, # 0 = simple course, 1-4 = combined course, 99 = no info
         course_descr = cod_des_cur, # Description of course (TP secondary education only). 0: Does not
         student_id = mrun,
         sex = gen_alu, # 0 = no info, 1 = male, 2 = female
         dob = fec_nac_alu,
         age_june30 = edad_alu, # age at 30th June 2021
         special_needs_status = int_alu, # integrated student indicator, 0 = no, 1 = yes. Mostly no
         special_needs_code = cod_int_alu, # ADHD, blindness, etc. 0 = none. 105 = autism, 203 = ADHD.
         student_region_code = cod_reg_alu,
         student_commune_code = cod_com_alu,
         student_commune_name = nom_com_alu,
         economic_sector_code = cod_sec,
         economic_specialty_code = cod_espe,
         economic_branch_code = cod_rama,
         economic_profspec_code = cod_men,
         teaching_code_new = ens)

chile_stdpop_raw <- read_excel("04_Data/pop_chile_2021_single_age.xlsx") %>%
  clean_names()

chile_stdpop <- chile_stdpop_raw %>%
  filter(sex != 9) %>%
  rename("std_pop" = "pop_2021") %>%
  mutate(pop_prop = std_pop / sum(std_pop))
```

Try Bayesian analysis of autism prevalence and specificity and sensitivity of school assessment "Bayesian Estimation of Disease Prevalence and the Parameters of Diagnostic Tests in the Absence of a Gold Standard" Lawrence Joseph, Theresa W. Gyorkos, Louis Coupal https://www.cambridge.org/core/journals/epidemiology-and-psychiatric-sciences/article/bayesian-approach-to-estimating-the-population-prevalence-of-mood-and-anxiety-disorders-using-multiple-measures/DB1D2CA6C27C7E8C85C60B62B969BB72

Use sensitivity and specificity of Social Attention and Communication Surveillance–Revised (SACS-R) tool "Diagnostic Accuracy of the Social Attention and Communication Surveillance–Revised With Preschool Tool for Early Autism Detection in Very Young Children" Josephine Barbaro, Nancy Sadka, Melissa Gilbert, et al https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2789926

```
chile_bayes_aut <- chile_merged %>%
  filter(age_june30 >= 6 & age_june30 <= 18,
         #special_needs_status == 1,
         sex != 0) %>%
  mutate(autism = ifelse(special_needs_code == 105, 1, 0),
         age_cat = ifelse(age_june30 <= 8, 1, ifelse(age_june30 <= 11, 2, ifelse(age_june30 <= 14, 3, 4)
            # 1 = 6-8, 2 = 9-11, 3 = 12-14, 4 = 15-18
  select(school_region_name_abr,
    sex,
    sex_desc,
    age_june30,
    #edad_alu_2, # equal to age_june30
    age_cat,
    school_rurality_code,
    #rural_rbd_2, # not quite equal to school_rurality_code as it has NA's
    pago_matricula,
    pago_mensual,
    school_fee,
    ethnicity,
    mapuche,
    nationality,
    ethnic_3_group,
    #asd_chile, # equal to autism
    autism
  )

# Prevalence of autism in Chile dataset
sum(chile_bayes_aut$autism) / nrow(chile_bayes_aut) # 0.00476 = 0.476%, very low
```

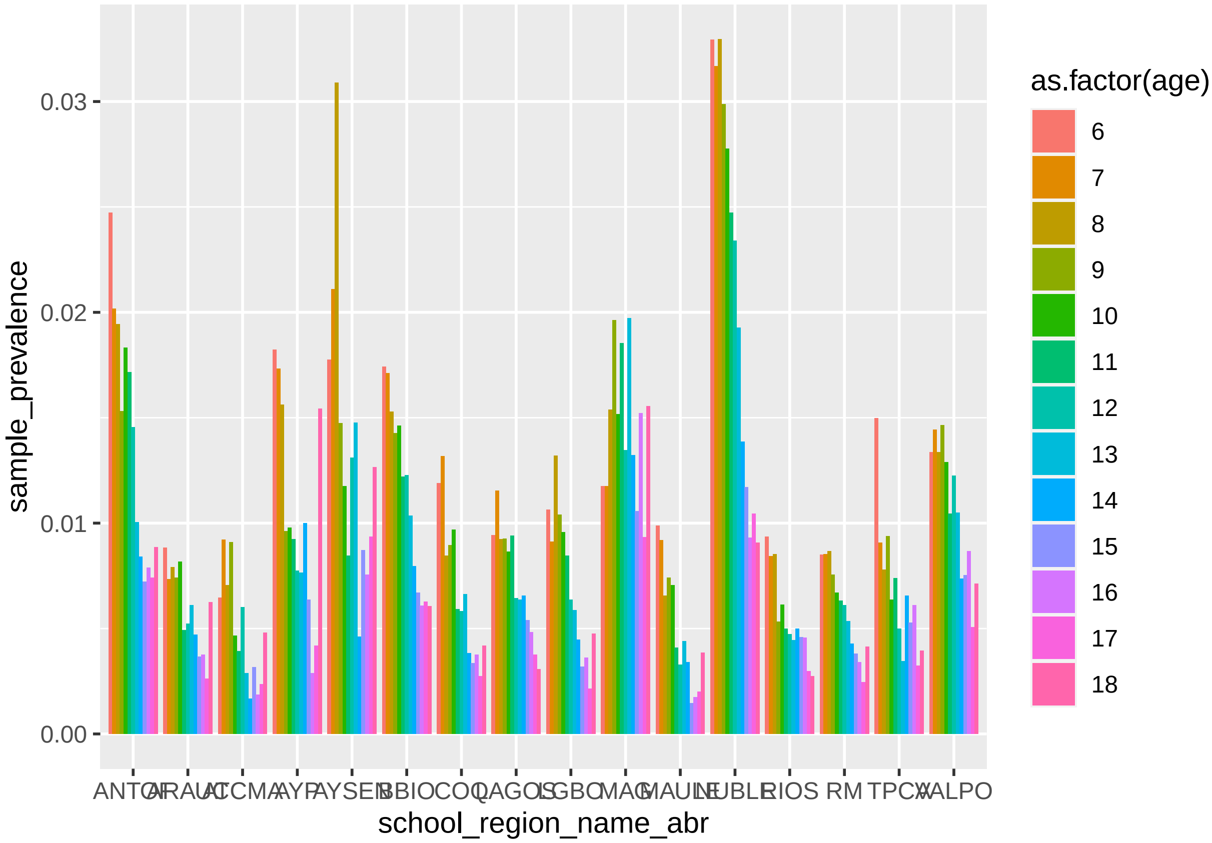
```
## [1] 0.004760322
```

```
# Is prevalence the same across geographic regions, age, sex?
n_std_pop <- sum(chile_stdpop$std_pop)

aut_prev_region <- chile_bayes_aut %>%
  group_by(school_region_name_abr, age_june30, sex, autism) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = autism, values_from = count) %>%
  rename("n_noautism" = "0", "n_autism" = "1", "age" = "age_june30") %>%
  mutate(n_autism = ifelse(is.na(n_autism), 0, n_autism),
         sample_pop_size = n_noautism + n_autism,
         sample_prevalence = n_autism / sample_pop_size) %>%
  left_join(chile_stdpop, by = c("age", "sex")) %>%
  mutate(aut_prev_std = n_autism / sample_pop_size * pop_prop,
         w = std_pop / (sample_pop_size * n_std_pop),
         w2 = pop_prop / sample_pop_size,
         sum_std_pop = sum(std_pop)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'school_region_name_abr', 'age_june30',
```

```
## 'sex'. You can override using the `.groups` argument.
```

```
ggplot(data = aut_prev_region) +
  geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = age, fill = as.factor(age)), p
```




```
  #geom_col(aes(x = school_region_name_abr, y = prevalence, group = sex, fill = as.factor(sex)), positi
    # 1 is male, 2 is female
```

### Bayesian prevalence analysis - common effects model with sample prevalence

```
nObs <- nrow(chile_bayes_aut)
nIter <- 1000
nBurn <- 1000
```

### Bayesian prevalence analysis

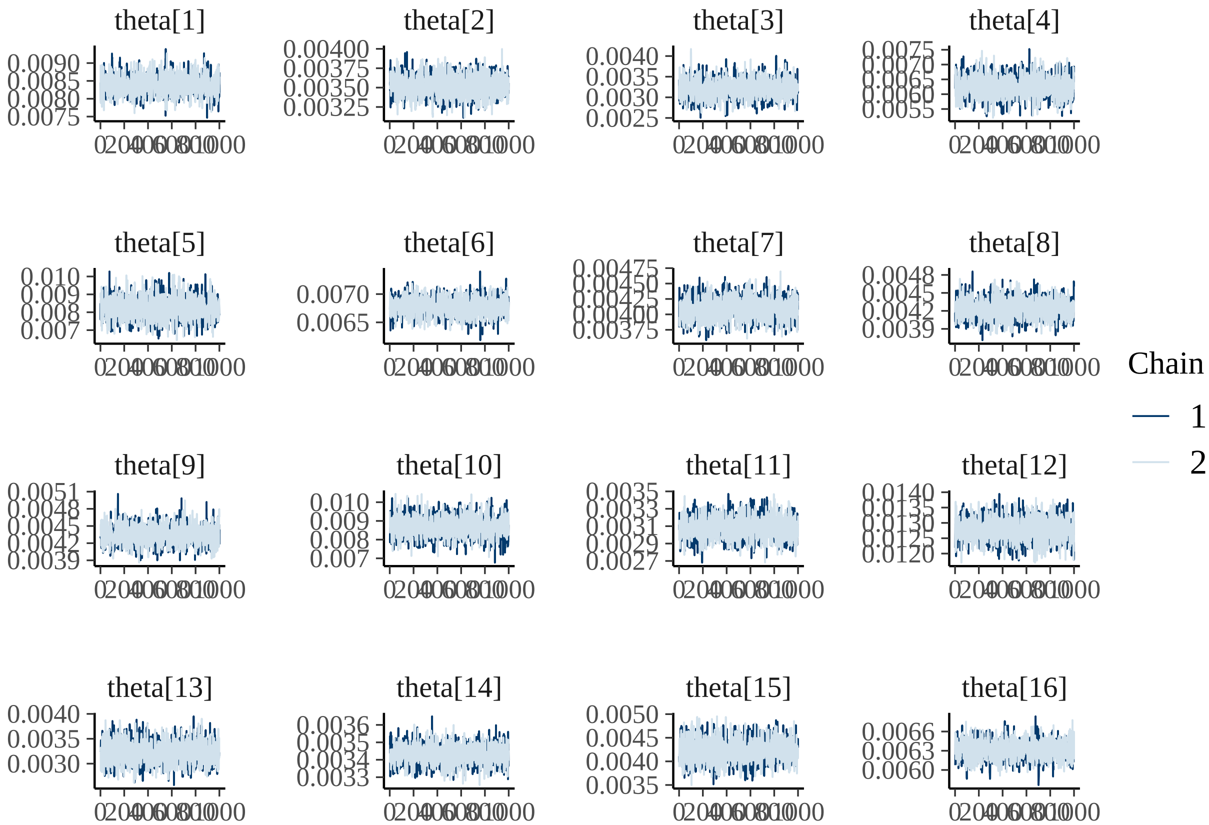
Standardise prevalence by Chile's age and sex based population sizes using https://seer.cancer.gov/seerstat/WebHelp/Rate_Algorithms.htm and https://wonder.cdc.gov/wonder/help/cancer/fayfeuerconfidenceintervals.pdf

See https://github.com/Dpananos/bayes_multiple_measures/blob/master/analysis/sensitivity_analysis.R for more sensitivity analysis ideas

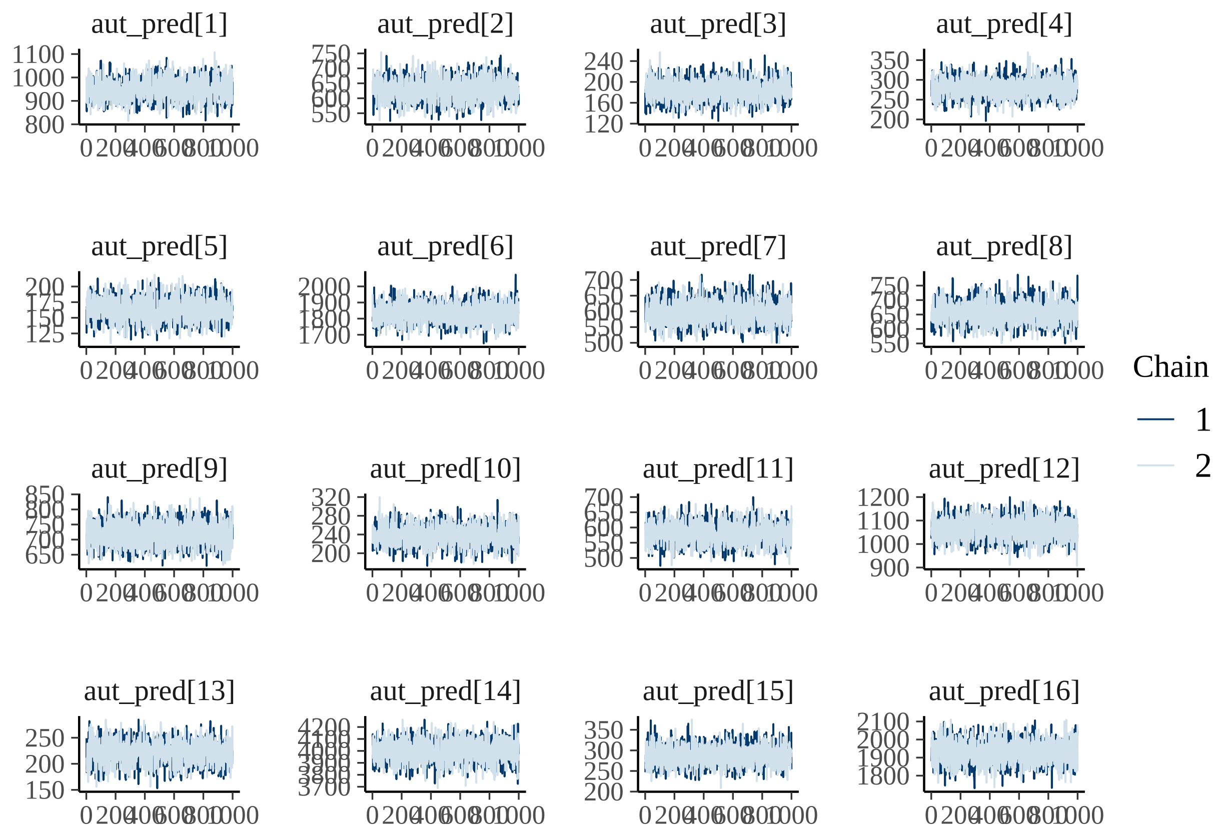
```
aut_prev_region_adj <- aut_prev_region %>%
  group_by(school_region_name_abr) %>%
  summarise(sum_sample_pop_size = sum(sample_pop_size),
            crude_rate = sum(n_autism) / sum(sample_pop_size),
```

```
            crude_count = sum(n_autism),
            adjusted_rate = sum(n_autism / sample_pop_size * pop_prop),
            adjusted_count = round(adjusted_rate * sum_sample_pop_size, 0), # had to fudge this to get 
            #adjusted_count = adjusted_rate * sum_sample_pop_size,
            var = sum(pop_prop^2 * n_autism / sample_pop_size^2),
            #se2 = sqrt(sum((std_pop/sum(std_pop))^2 * n_autism/sample_pop_size^2)),
            w_M = max(w),
            ci_lower = var / (2*adjusted_rate) * qchisq(p = 0.05/2, df = 2*adjusted_rate^2 / var),
            ci_upper = (var + w_M^2) / (2*(adjusted_rate + w_M)) * qchisq(p = 1-0.05/2, df = 2*(adjuste
  arrange(school_region_name_abr)

# Try informative prior
theta_mu <- 0.0046
theta_sigma <- (0.0047-0.0045) / (2*1.96)
theta_mu <- 0.01
theta_sigma <- 0.005 / 1.96 # Allow 0.25% either side
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

nRegion <- length(unique(aut_prev_region$school_region_name_abr))

rand_region_model <- "model {
  for(i in 1:nRegion) { # For each region
    theta[i] ~ dbeta(theta_a, theta_b)
    aut_sample[i] ~ dbin(theta[i], nObs[i])

    aut_pred[i] ~ dbin(theta[i], nObs[i])
  }
}"

rand_region_data <- list(theta_a = theta_a,
                         theta_b = theta_b,
                         #nObs = rep(3056300, 16),
                         #nObs = c(113208, 178136, 57116, 44648, 19858, 270637, 146522, 154827, 165436,
                         nObs = aut_prev_region_adj$sum_sample_pop_size,
                         #nObs = matrix(c(chile_rand_region$nObs), nrow = nRegion),
                         #aut_sample = rep(sum(chile_bayes_aut$autism), 16),
                         aut_sample = aut_prev_region_adj$adjusted_count,
                         #aut_sample = matrix(c(chile_rand_region$aut_sample), nrow = nRegion),
                         nRegion = nRegion)

rand_region_ini <- list(list(theta = rep(0.001, nRegion)), #, spec = 0.5, sens = 0.5),
                        list(theta = rep(0.01, nRegion))) #, spec = 0.9, sens = 0.9))

rand_region_pars <- c("theta_a", "theta_b", "theta", "aut_sample", "aut_pred")

# Run JAGS model and discard burn-in samples
rand_region_jag <- jags.model(textConnection(rand_region_model),
                              data = rand_region_data,
                              inits = rand_region_ini,
                              n.chains = 2,
                              quiet = TRUE)
update(rand_region_jag, n.iter = nBurn)
```

```
rand_region_sam <- coda.samples(model = rand_region_jag,
                                variable.names = rand_region_pars,
                                n.iter = nIter)

# Check for convergence in parameters of interest
#mcmc_trace(rand_region_sam, rand_region_pars)
mcmc_trace(rand_region_sam, paste0("theta[", 1:nRegion, "]")) # Convergence looks fine and rhats <= 1.1
```




```
mcmc_trace(rand_region_sam, paste0("aut_pred[", 1:nRegion, "]"))# Convergence looks fine and rhats <= 1
```



```r
summary(as_draws(rand_region_sam)) %>% print(n = Inf)
```

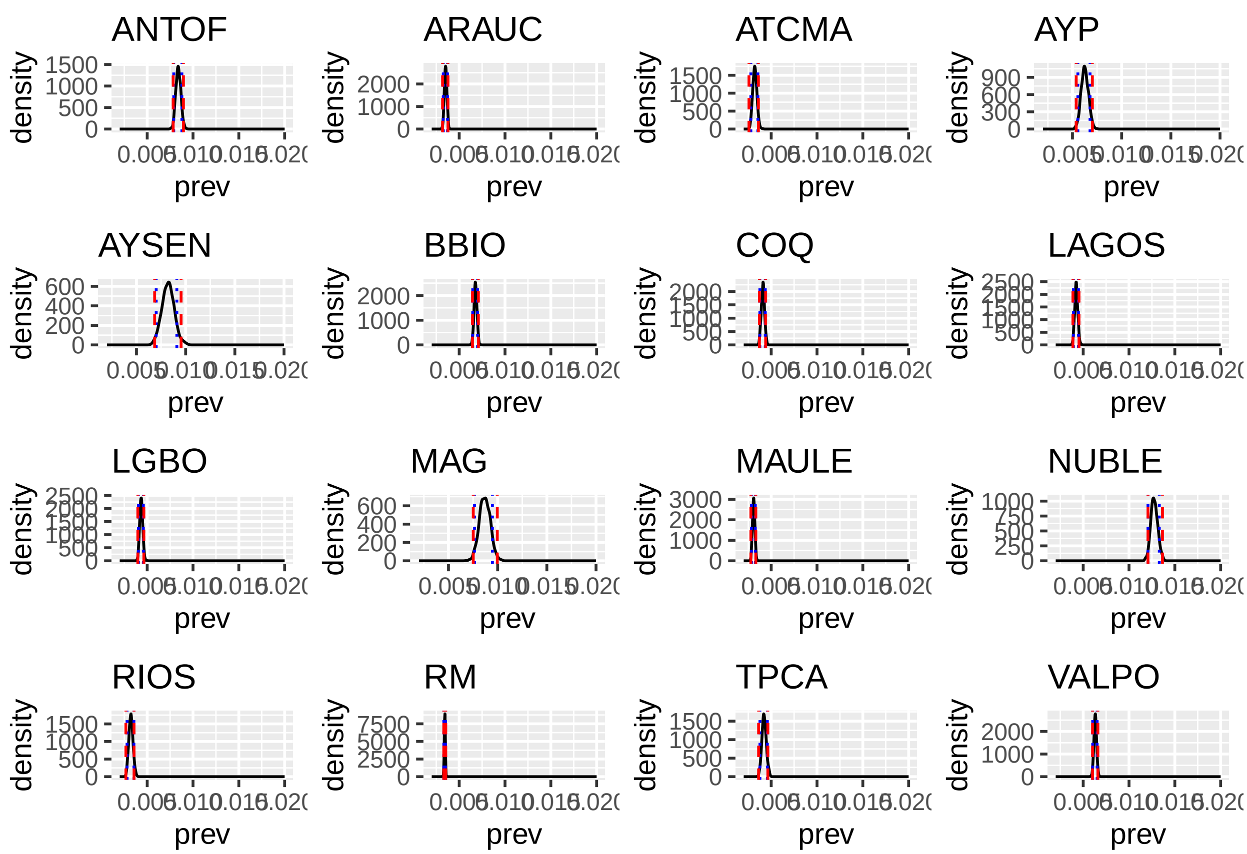
```
## # A tibble: 50 x 10
##    variable          mean  median      sd     mad      q5     q95  rhat ess_b~1
##    <chr>            <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>
##  1 aut_pred[1]    9.50e+2 9.49e+2 4.35e+1 4.30e+1 8.80e+2 1.02e+3  1.00    1427.
##  2 aut_pred[2]    6.27e+2 6.28e+2 3.47e+1 3.41e+1 5.7 e+2 6.84e+2  1.00    1603.
##  3 aut_pred[3]    1.83e+2 1.83e+2 1.89e+1 1.93e+1 1.53e+2 2.15e+2  1.00    1469.
##  4 aut_pred[4]    2.80e+2 2.8 e+2 2.35e+1 2.37e+1 2.41e+2 3.17e+2  1.00    1479.
##  5 aut_pred[5]    1.63e+2 1.63e+2 1.79e+1 1.78e+1 1.34e+2 1.94e+2  1.00    1470.
##  6 aut_pred[6]    1.84e+3 1.84e+3 5.85e+1 5.78e+1 1.74e+3 1.93e+3  1.00    1646.
##  7 aut_pred[7]    6.00e+2 5.99e+2 3.54e+1 3.56e+1 5.42e+2 6.57e+2  1.00    1607.
##  8 aut_pred[8]    6.57e+2 6.55e+2 3.64e+1 3.71e+1 5.99e+2 7.18e+2  1.00    1655.
##  9 aut_pred[9]    7.19e+2 7.18e+2 3.78e+1 4.00e+1 6.56e+2 7.8 e+2  1.00    1644.
## 10 aut_pred[10]   2.37e+2 2.37e+2 2.08e+1 2.08e+1 2.04e+2 2.72e+2  0.999   1589.
## 11 aut_pred[11]   5.81e+2 5.81e+2 3.39e+1 3.41e+1 5.25e+2 6.37e+2  1.00    1643.
## 12 aut_pred[12]   1.06e+3 1.06e+3 4.43e+1 4.45e+1 9.93e+2 1.14e+3  1.00    1585.
## 13 aut_pred[13]   2.21e+2 2.21e+2 2.14e+1 2.08e+1 1.86e+2 2.56e+2  1.00    1305.
## 14 aut_pred[14]   4.00e+3 3.99e+3 8.93e+1 9.04e+1 3.85e+3 4.14e+3  1.00    1568.
## 15 aut_pred[15]   2.87e+2 2.87e+2 2.30e+1 2.37e+1 2.5 e+2 3.25e+2  1.00    1822.
## 16 aut_pred[16]   1.93e+3 1.92e+3 6.31e+1 6.67e+1 1.82e+3 2.03e+3  1.00    1539.
## 17 aut_sample[1]  9.48e+2 9.48e+2 0       0       9.48e+2 9.48e+2 NA         NA
## 18 aut_sample[2]  6.17e+2 6.17e+2 0       0       6.17e+2 6.17e+2 NA         NA
## 19 aut_sample[3]  1.73e+2 1.73e+2 0       0       1.73e+2 1.73e+2 NA         NA
## 20 aut_sample[4]  2.74e+2 2.74e+2 0       0       2.74e+2 2.74e+2 NA         NA
## 21 aut_sample[5]  1.6 e+2 1.6 e+2 0       0       1.6 e+2 1.6 e+2 NA         NA
```

```
## 22 aut_sample[6]  1.83e+3 1.83e+3 0       0       1.83e+3 1.83e+3 NA          NA
## 23 aut_sample[7]  5.91e+2 5.91e+2 0       0       5.91e+2 5.91e+2 NA          NA
## 24 aut_sample[8]  6.49e+2 6.49e+2 0       0       6.49e+2 6.49e+2 NA          NA
## 25 aut_sample[9]  7.09e+2 7.09e+2 0       0       7.09e+2 7.09e+2 NA          NA
## 26 aut_sample[10] 2.35e+2 2.35e+2 0       0       2.35e+2 2.35e+2 NA          NA
## 27 aut_sample[11] 5.72e+2 5.72e+2 0       0       5.72e+2 5.72e+2 NA          NA
## 28 aut_sample[12] 1.07e+3 1.07e+3 0       0       1.07e+3 1.07e+3 NA          NA
## 29 aut_sample[13] 2.11e+2 2.11e+2 0       0       2.11e+2 2.11e+2 NA          NA
## 30 aut_sample[14] 3.99e+3 3.99e+3 0       0       3.99e+3 3.99e+3 NA          NA
## 31 aut_sample[15] 2.78e+2 2.78e+2 0       0       2.78e+2 2.78e+2 NA          NA
## 32 aut_sample[16] 1.92e+3 1.92e+3 0       0       1.92e+3 1.92e+3 NA          NA
## 33 theta[1]       8.40e-3 8.39e-3 2.71e-4 2.80e-4 7.97e-3 8.84e-3  1.00    1108.
## 34 theta[2]       3.52e-3 3.52e-3 1.36e-4 1.36e-4 3.30e-3 3.76e-3  1.00    1209.
## 35 theta[3]       3.21e-3 3.21e-3 2.28e-4 2.31e-4 2.85e-3 3.59e-3  1.00    1191.
## 36 theta[4]       6.25e-3 6.24e-3 3.64e-4 3.66e-4 5.67e-3 6.86e-3  1.00    1223.
## 37 theta[5]       8.22e-3 8.22e-3 6.21e-4 6.07e-4 7.20e-3 9.22e-3  1.00    1190.
## 38 theta[6]       6.78e-3 6.78e-3 1.56e-4 1.56e-4 6.52e-3 7.04e-3  1.00    1281.
## 39 theta[7]       4.10e-3 4.10e-3 1.74e-4 1.76e-4 3.80e-3 4.38e-3  1.00    1336.
## 40 theta[8]       4.24e-3 4.24e-3 1.66e-4 1.61e-4 3.98e-3 4.52e-3  1.00    1289.
## 41 theta[9]       4.35e-3 4.34e-3 1.62e-4 1.61e-4 4.08e-3 4.62e-3  1.00    1246.
## 42 theta[10]      8.70e-3 8.69e-3 5.45e-4 5.66e-4 7.80e-3 9.58e-3  1.00    1163.
## 43 theta[11]      3.09e-3 3.09e-3 1.26e-4 1.27e-4 2.88e-3 3.30e-3  1.00    1362.
## 44 theta[12]      1.28e-2 1.27e-2 3.70e-4 3.71e-4 1.22e-2 1.34e-2  1.00    1251.
## 45 theta[13]      3.22e-3 3.22e-3 2.22e-4 2.23e-4 2.86e-3 3.60e-3  1.00    1210.
## 46 theta[14]      3.43e-3 3.43e-3 5.52e-5 5.61e-5 3.34e-3 3.52e-3  1.00    1320.
## 47 theta[15]      4.23e-3 4.22e-3 2.37e-4 2.39e-4 3.84e-3 4.64e-3  1.00    1379.
## 48 theta[16]      6.31e-3 6.31e-3 1.45e-4 1.41e-4 6.08e-3 6.56e-3  1.00    1342.
## 49 theta_a        1.52e+1 1.52e+1 0       0       1.52e+1 1.52e+1 NA          NA
## 50 theta_b        1.51e+3 1.51e+3 0       0       1.51e+3 1.51e+3 NA          NA
## # ... with 1 more variable: ess_tail <dbl>, and abbreviated variable name
## #   1: ess_bulk
```

```
rand_region_summ <- summary(subset_draws(as_draws(rand_region_sam), rand_region_pars),
                       ~quantile(.x, probs=c(0.025, 0.5, 0.975)),
                       ~mcse_quantile(.x, probs=c(0.025, 0.5, 0.975)),
                       "rhat") %>%
  arrange(desc(mcse_q50))
rand_region_summ
```

```
## # A tibble: 50 x 8
##    variable     `2.5%` `50%` `97.5%` mcse_q2.5 mcse_q50 mcse_q97.5  rhat
##    <chr>         <dbl> <dbl>   <dbl>     <dbl>    <dbl>      <dbl> <dbl>
##  1 aut_pred[14]  3823   3994   4169.       3.5      2.5        6    1.00
##  2 aut_pred[16]  1811   1925   2052.       2.5      2          6    1.00
##  3 aut_pred[1]    865    949   1038        2.5      1.5        2.5  1.00
##  4 aut_pred[6]   1723.  1835   1948.       5        1.5        3.5  1.00
##  5 aut_pred[12]   977   1063   1153        4.5      1.5        3    1.00
##  6 aut_pred[2]    560.   628    694        2.5      1          2.5  1.00
##  7 aut_pred[8]    588    655    730        3.5      1          3.5  1.00
##  8 aut_pred[9]    647    718    791        1.5      1          1    1.00
##  9 aut_pred[11]   515    581    648        1.5      1          2    1.00
## 10 aut_pred[13]   178    221    262.       2        1          1.5  1.00
## # ... with 40 more rows
```

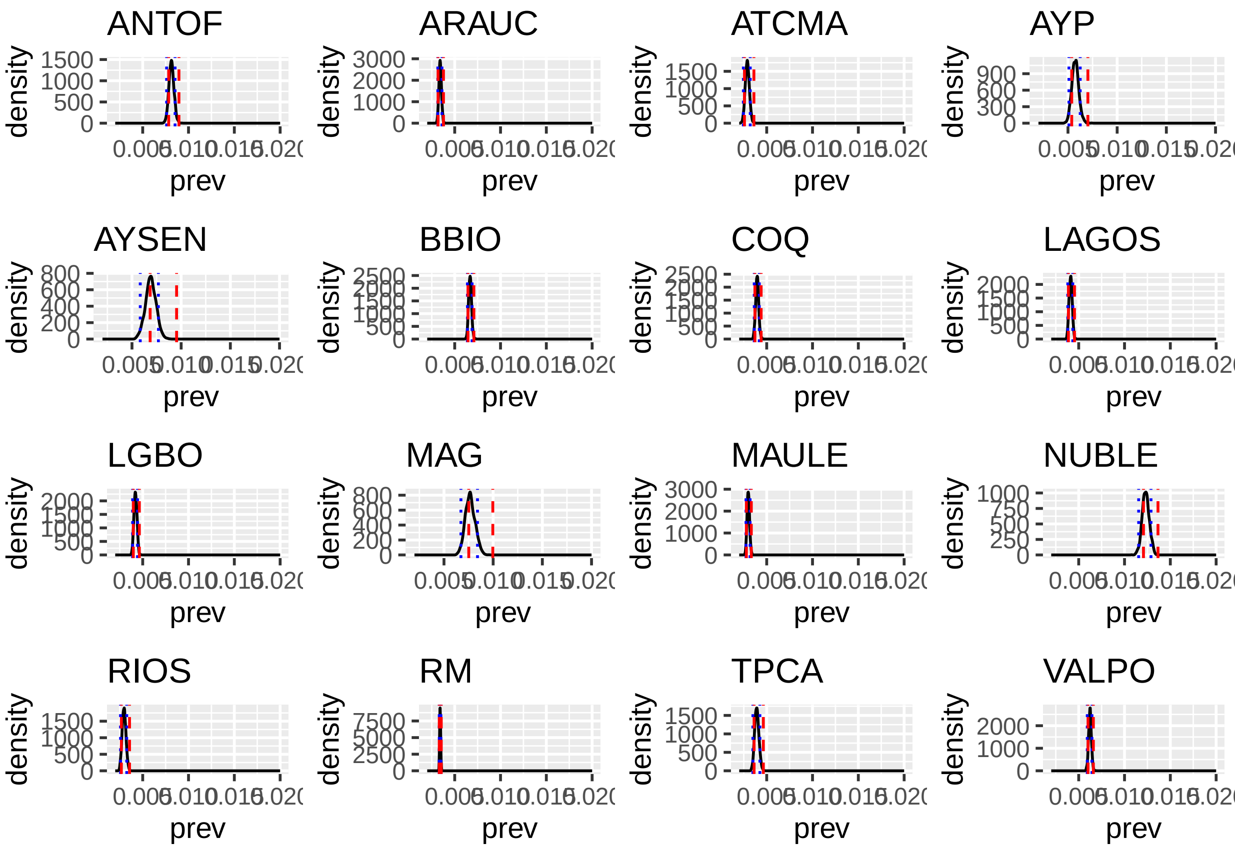
```
aut_prev_region_plots <- list()
region_post_ci_lower <- list()
region_post_ci_upper <- list()

for(i in 1:nRegion) {
  prevs <- data.frame(prev = extract_variable(rand_region_sam, paste0("theta[", i, "]")))
  region_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
  region_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
  density_plot <- ggplot(prevs, aes(x = prev)) +
    geom_density() +
    xlim(c(0.002, 0.02)) +
    geom_vline(xintercept = region_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = region_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = aut_prev_region_adj$ci_lower[i], color = "red", linetype = "dashed") +
    geom_vline(xintercept = aut_prev_region_adj$ci_upper[i], color = "red", linetype = "dashed") +
    labs(title = aut_prev_region_adj$school_region_name_abr[i])
  aut_prev_region_plots[[i]] <- density_plot
}
do.call(grid.arrange, aut_prev_region_plots)
```

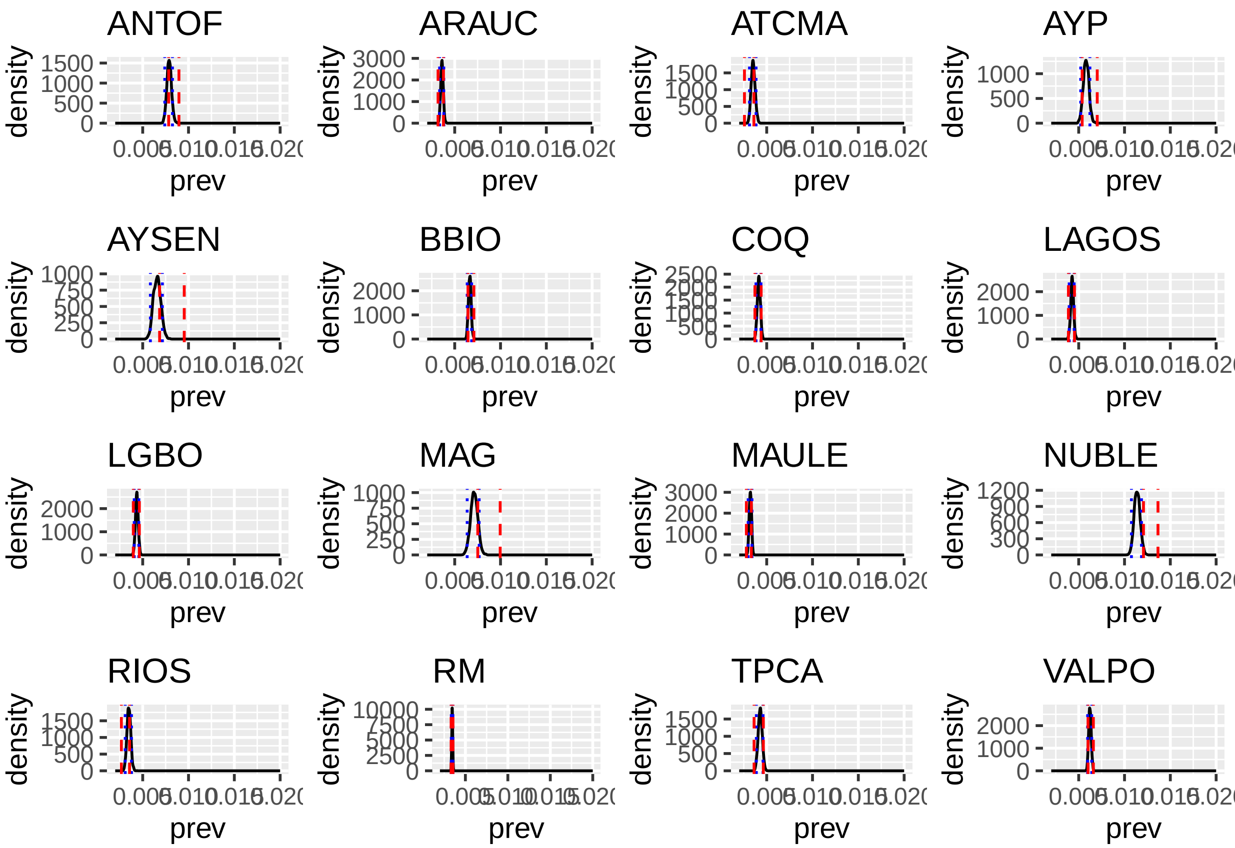


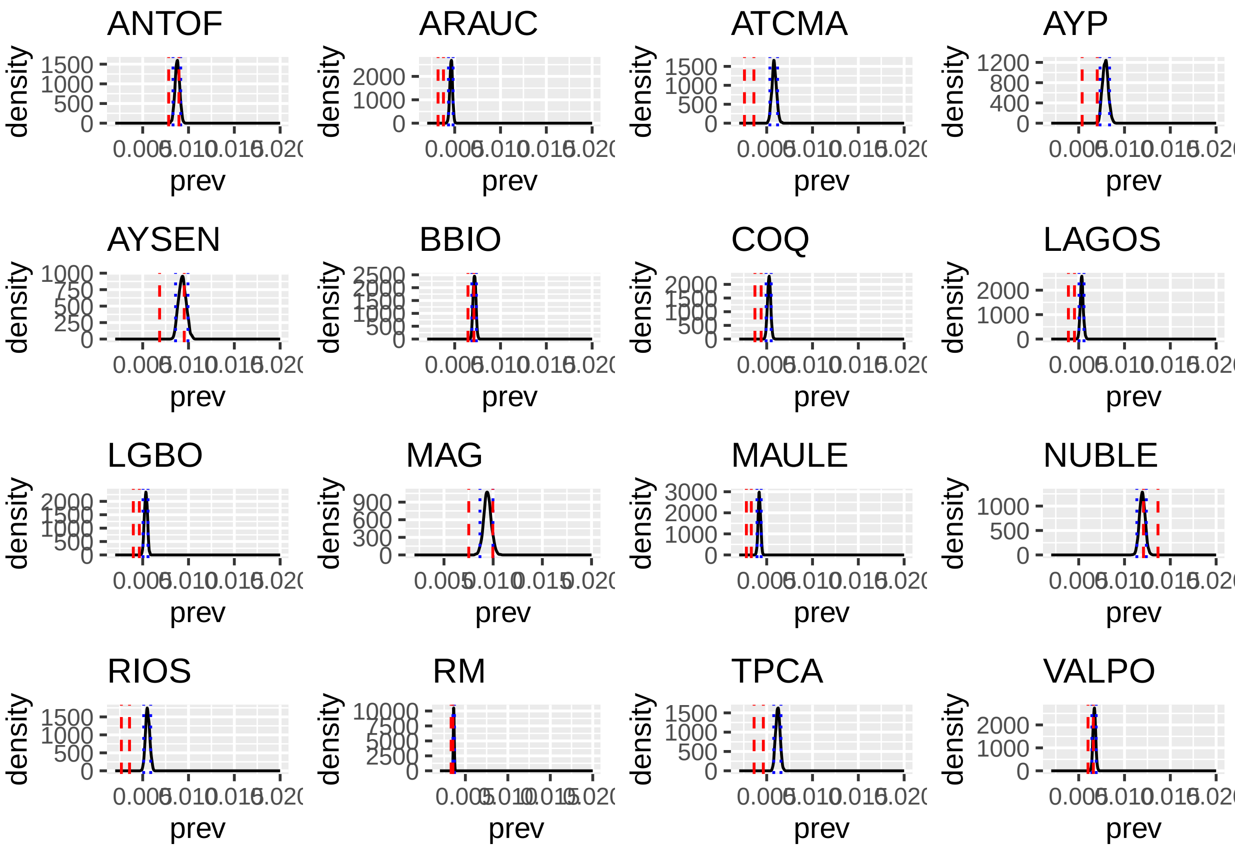

```
#autism_prev_region_plots <- do.call(grid.arrange, aut_prev_region_plots)
#ggsave("autism_prev_region_plots.png", autism_prev_region_plots, height = 10, width = 15)
```

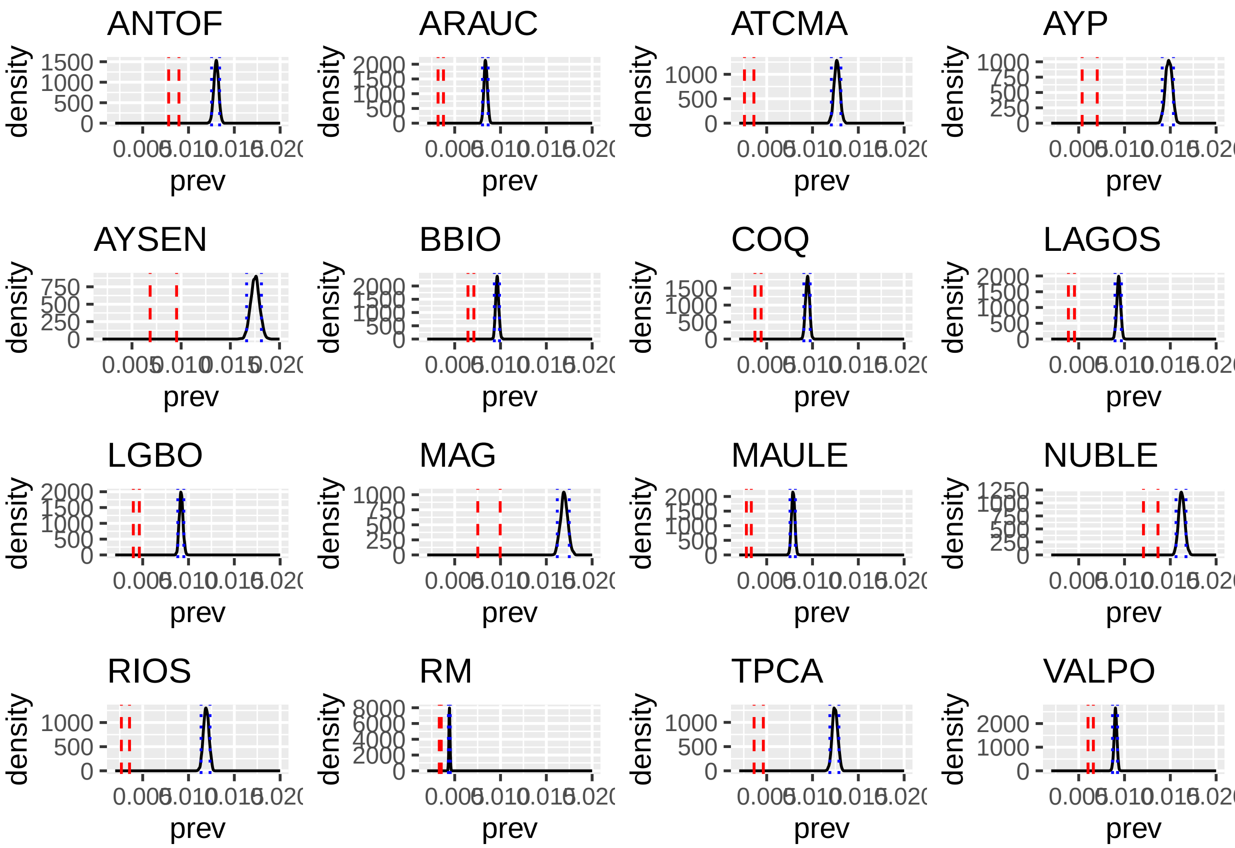
**Sensitivity analysis - alter prior mean and sd**

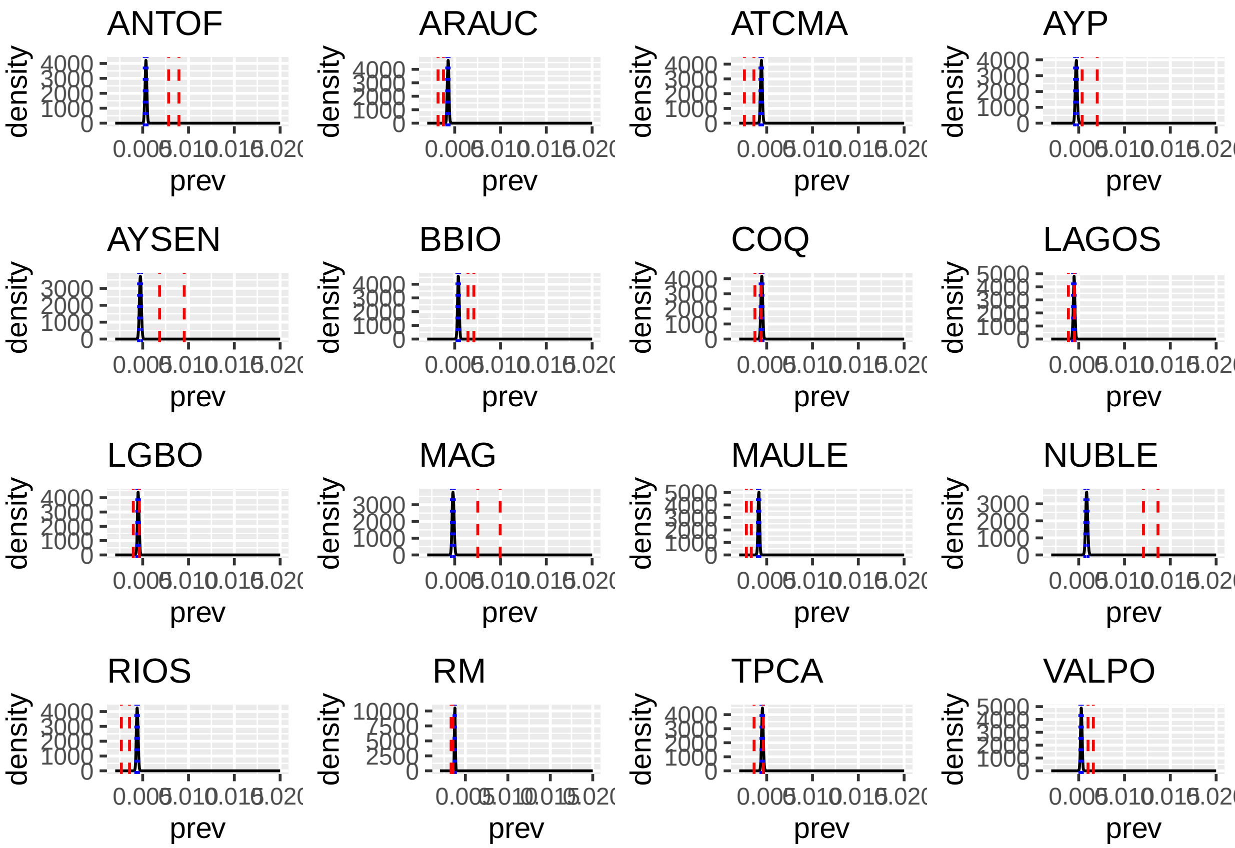
```
theta_mu <- c(0.001, 0.005, 0.01, 0.02, # 0.1%, 0.5%, 1%, 2% prevalence
              rep(0.0046, 4)) # Same as chosen prior
theta_sigma <- c(rep(0.001/1.96, 4), # Same as chosen prior
                 0.0001, 0.001, 0.05, 0.01) # +/- 0.1%, 0.5%, 1%, 5%
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

for(j in 1:length(theta_mu)) {
  #print(j)
  #print(theta_a[j])
  #print(theta_b[j])
  rand_region_data <- list(theta_a = theta_a[j],
                           theta_b = theta_b[j],
                           nObs = aut_prev_region_adj$sum_sample_pop_size,
                           aut_sample = aut_prev_region_adj$adjusted_count,
                           nRegion = nRegion)
  rand_region_jag <- jags.model(textConnection(rand_region_model),
                                data = rand_region_data,
                                inits = rand_region_ini,
                                n.chains = 2,
                                quiet = TRUE)
  update(rand_region_jag, n.iter = nBurn)
  rand_region_sam <- coda.samples(model = rand_region_jag,
                                  variable.names = rand_region_pars,
                                  n.iter = nIter)
  mcmc_trace(rand_region_sam, paste0("theta[", 1:nRegion, "]")) # Convergence looks fine and rhats <= 1
  mcmc_trace(rand_region_sam, paste0("aut_pred[", 1:nRegion, "]"))# Convergence looks fine and rhats <=

  # Plot
  aut_prev_region_plots <- list()
  region_post_ci_lower <- list()
  region_post_ci_upper <- list()

  for(i in 1:nRegion) {
    prevs <- data.frame(prev = extract_variable(rand_region_sam, paste0("theta[", i, "]")))
    region_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
    region_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
    density_plot <- ggplot(prevs, aes(x = prev)) +
      geom_density() +
      xlim(c(0.002, 0.02)) +
      geom_vline(xintercept = region_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = region_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = aut_prev_region_adj$ci_lower[i], color = "red", linetype = "dashed") +
      geom_vline(xintercept = aut_prev_region_adj$ci_upper[i], color = "red", linetype = "dashed") +
      labs(title = aut_prev_region_adj$school_region_name_abr[i])
    aut_prev_region_plots[[i]] <- density_plot
  }
  autism_prev_region_plots <- do.call(grid.arrange, aut_prev_region_plots)
  #ggsave(paste0("autism_prev_region_plots_", j, ".png"), autism_prev_region_plots, height = 10, width
}
```

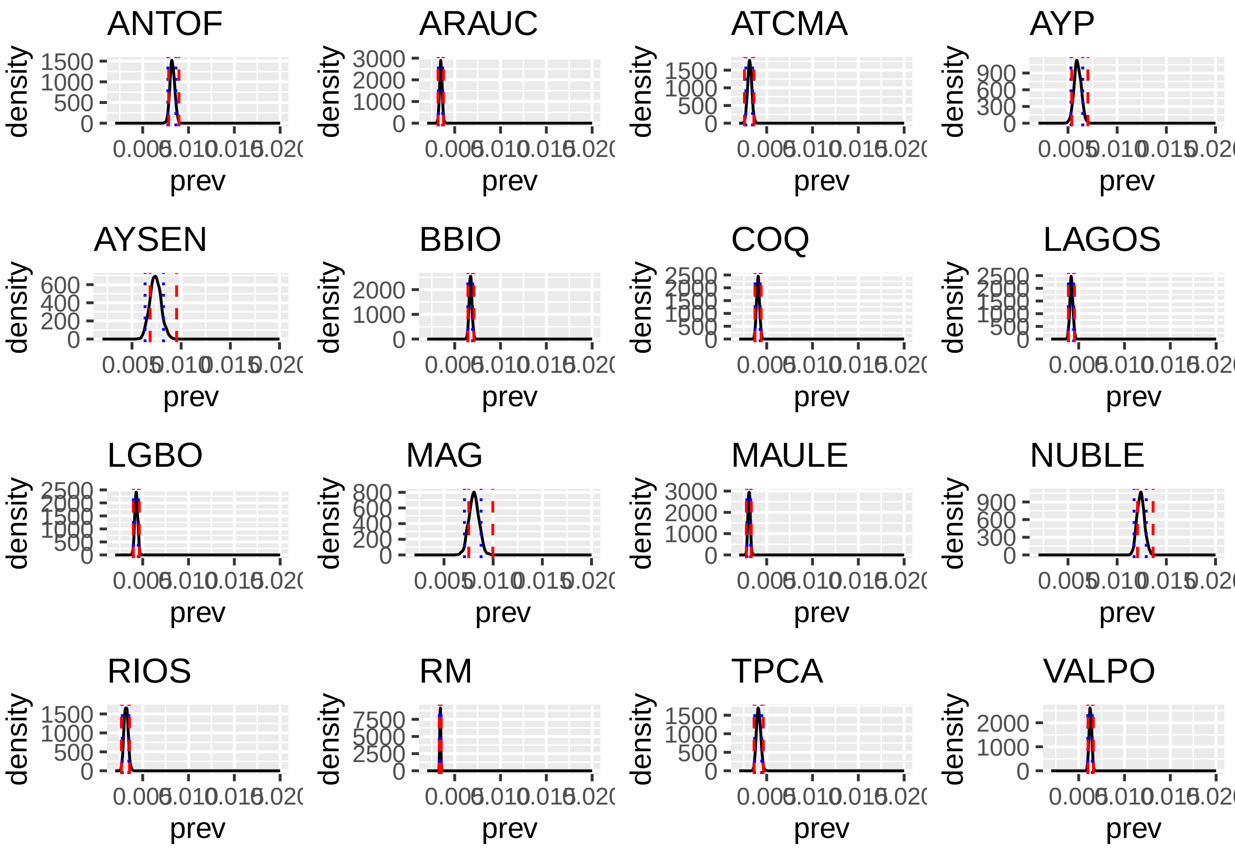

ANTOF
ARAUC
ATCMA
AYP
AYSEN
BBIO
COQ
LAGOS
LGBO
MAG
MAULE
NUBLE
RIOS
RM
TPCA
VALPO
density
prev

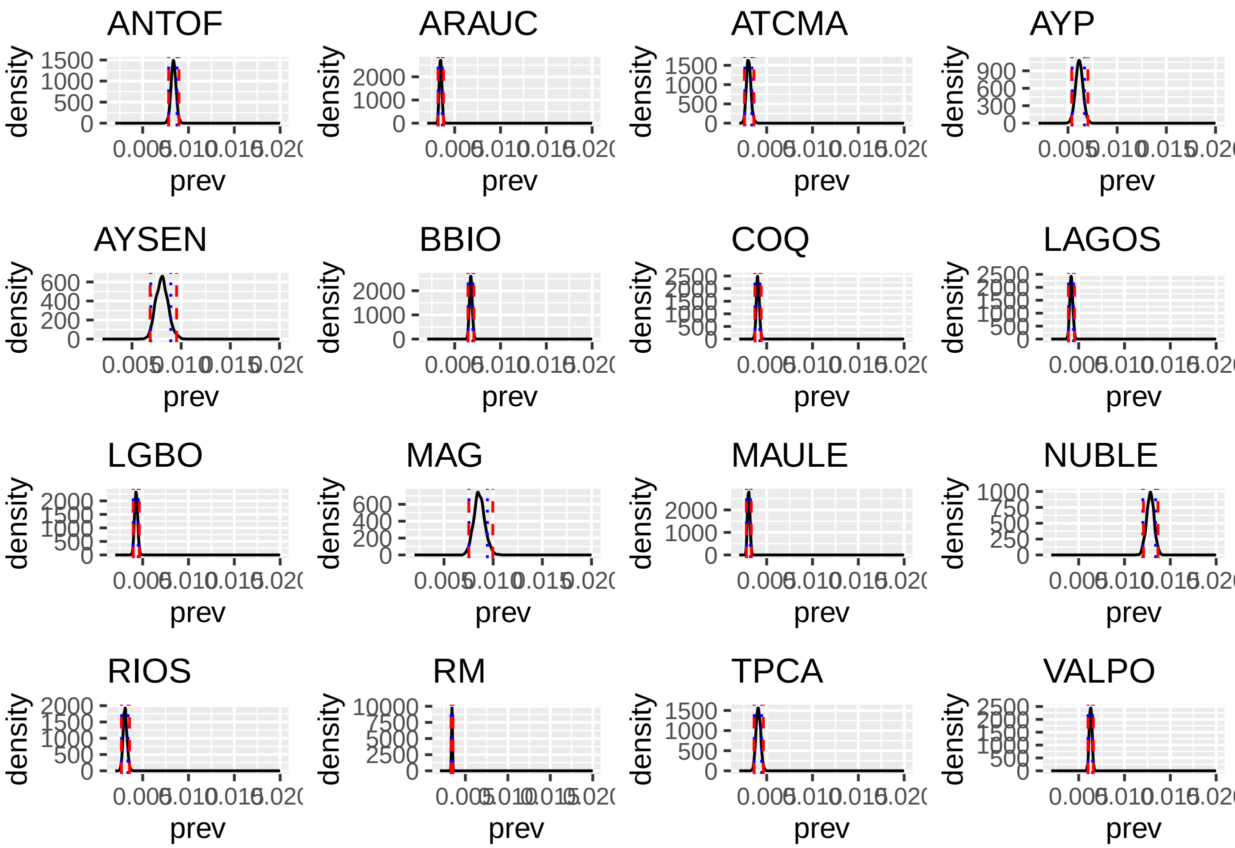

ANTOF
ARAUC
ATCMA
AYP
AYSEN
BBIO
COQ
LAGOS
LGBO
MAG
MAULE
NUBLE
RIOS
RM
TPCA
VALPO
density
prev

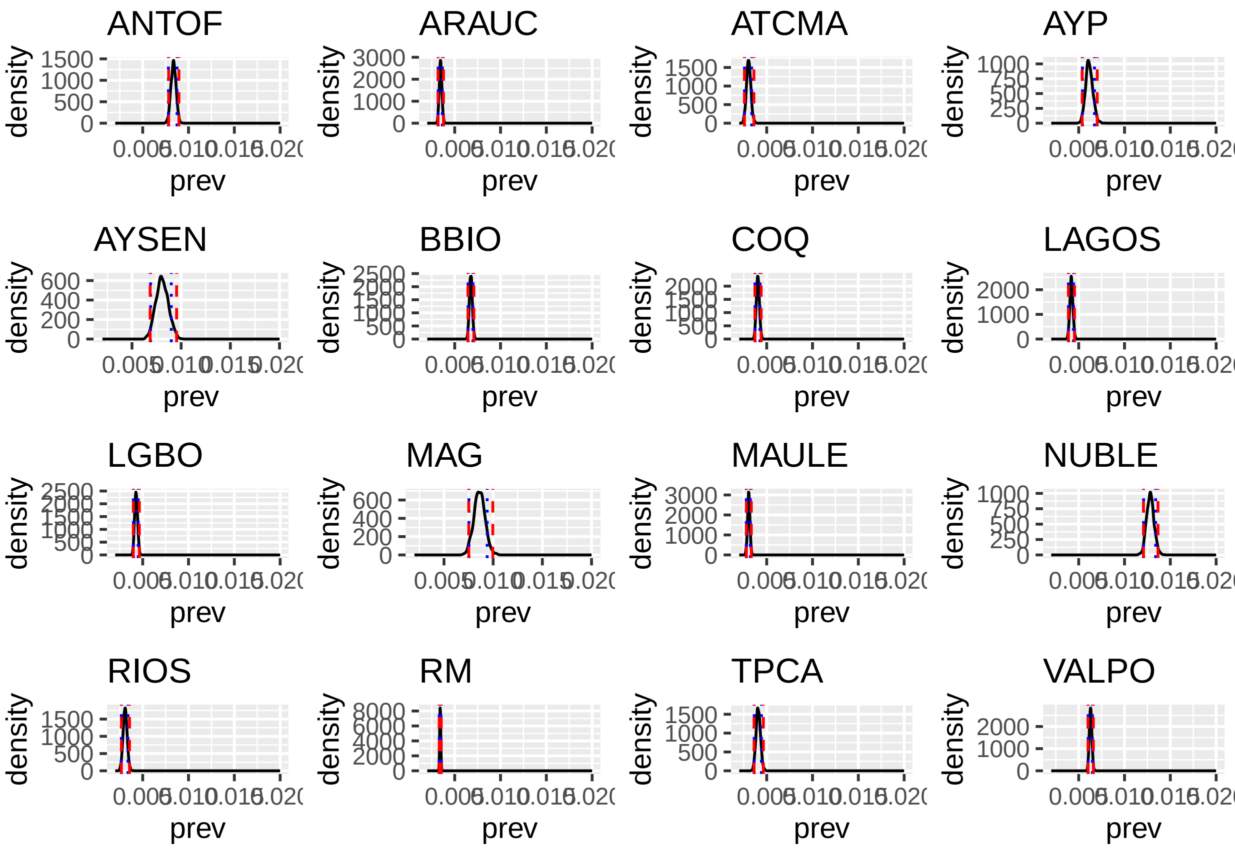

ANTOF
ARAUC
ATCMA
AYP
AYSEN
BBIO
COQ
LAGOS
LGBO
MAG
MAULE
NUBLE
RIOS
RM
TPCA
VALPO
density
prev


ANTOF
ARAUC
ATCMA
AYP
AYSEN
BBIO
COQ
LAGOS
LGBO
MAG
MAULE
NUBLE
RIOS
RM
TPCA
VALPO
density
prev


ANTOF
ARAUC
ATCMA
AYP
AYSEN
BBIO
COQ
LAGOS
LGBO
MAG
MAULE
NUBLE
RIOS
RM
TPCA
VALPO
density
prev


ANTOF
ARAUC
ATCMA
AYP
AYSEN
BBIO
COQ
LAGOS
LGBO
MAG
MAULE
NUBLE
RIOS
RM
TPCA
VALPO
density
prev


ANTOF
ARAUC
ATCMA
AYP
AYSEN
BBIO
COQ
LAGOS
LGBO
MAG
MAULE
NUBLE
RIOS
RM
TPCA
VALPO
density
prev

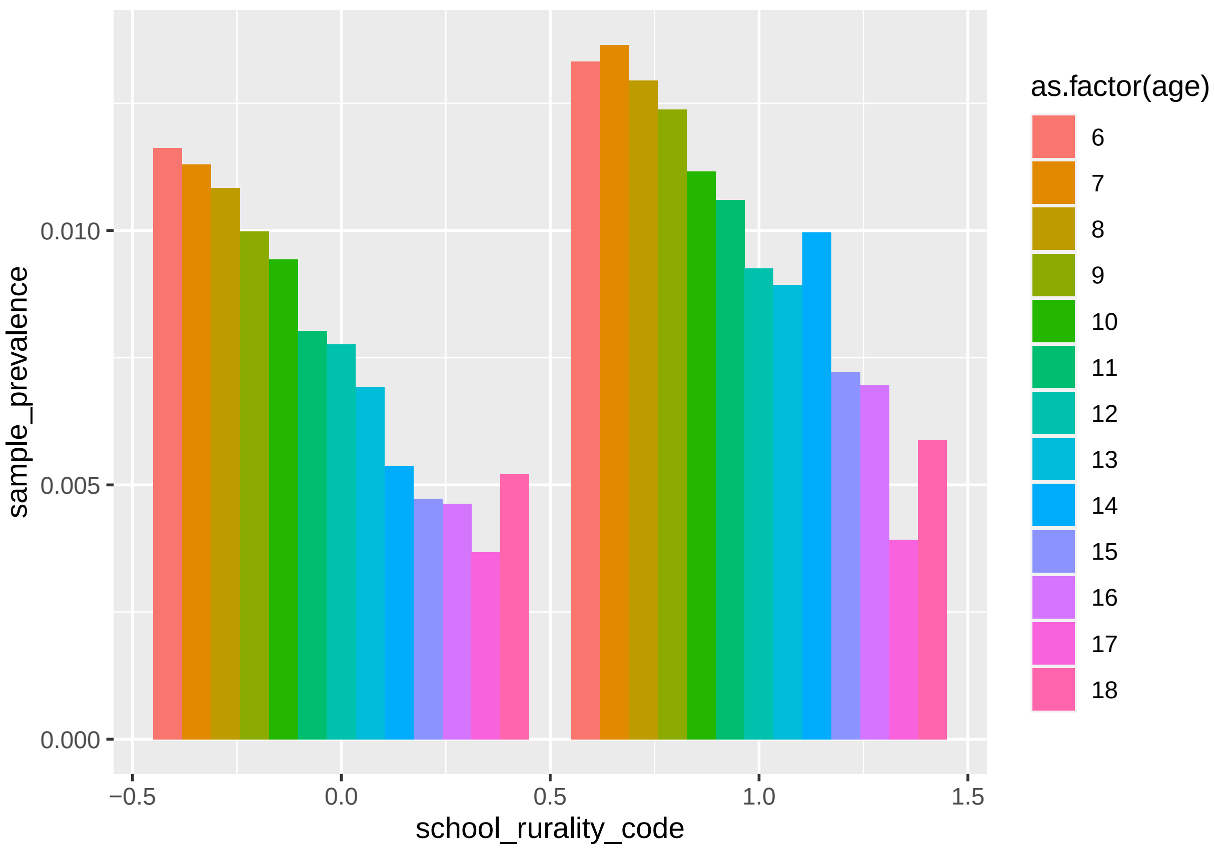



### Bayesian prevalence by rurality

```
aut_prev_rural <- chile_bayes_aut %>%
  group_by(school_rurality_code, age_june30, sex, autism) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = autism, values_from = count) %>%
  rename("n_noautism" = "0", "n_autism" = "1", "age" = "age_june30") %>%
  mutate(n_autism = ifelse(is.na(n_autism), 0, n_autism),
         sample_pop_size = n_noautism + n_autism,
         sample_prevalence = n_autism / sample_pop_size) %>%
  left_join(chile_stdpop, by = c("age", "sex")) %>%
  mutate(aut_prev_std = n_autism / sample_pop_size * pop_prop,
         w = std_pop / (sample_pop_size * n_std_pop),
         w2 = pop_prop / sample_pop_size,
         #sum_std_pop = sum(std_pop)
         ) %>%
  ungroup()
```

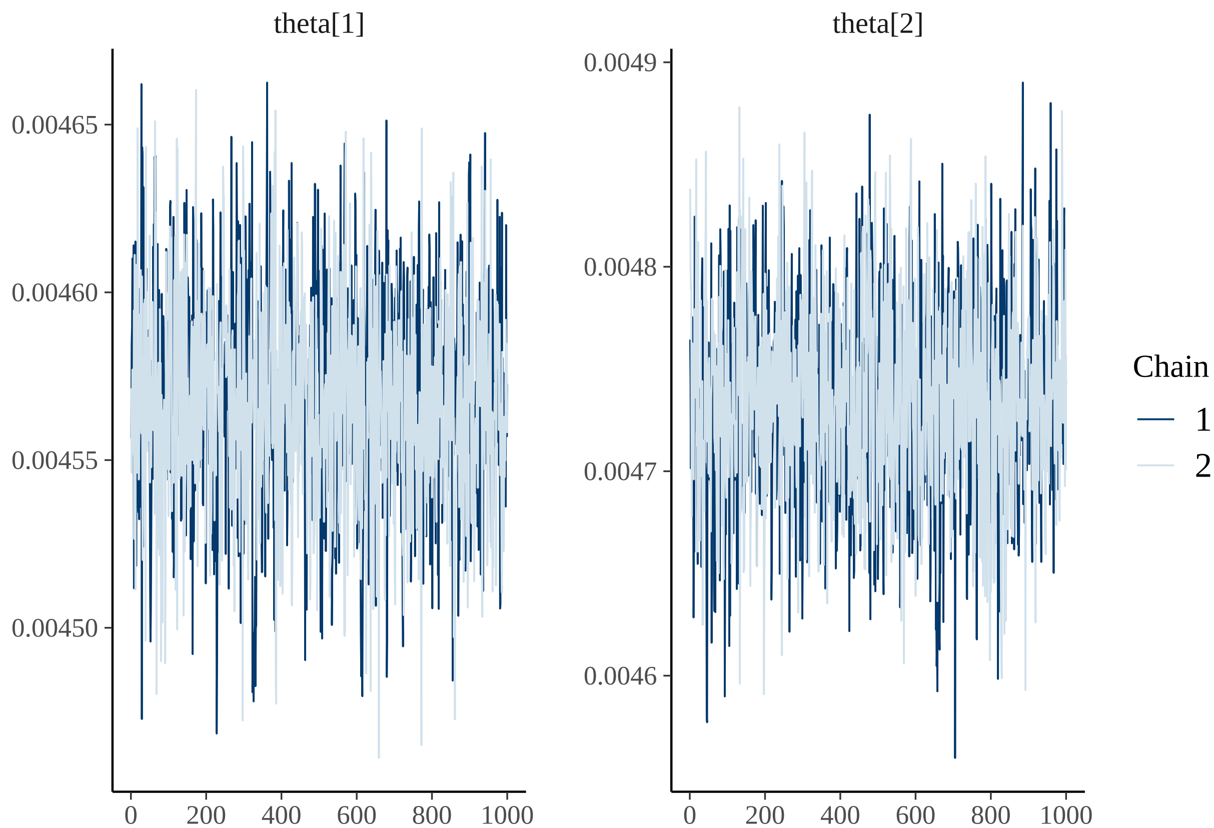
```
## `summarise()` has grouped output by 'school_rurality_code', 'age_june30',
## 'sex'. You can override using the `.groups` argument.
```

```
ggplot(data = aut_prev_rural) +
  geom_col(aes(x = school_rurality_code, y = sample_prevalence, group = age, fill = as.factor(age)), po
```

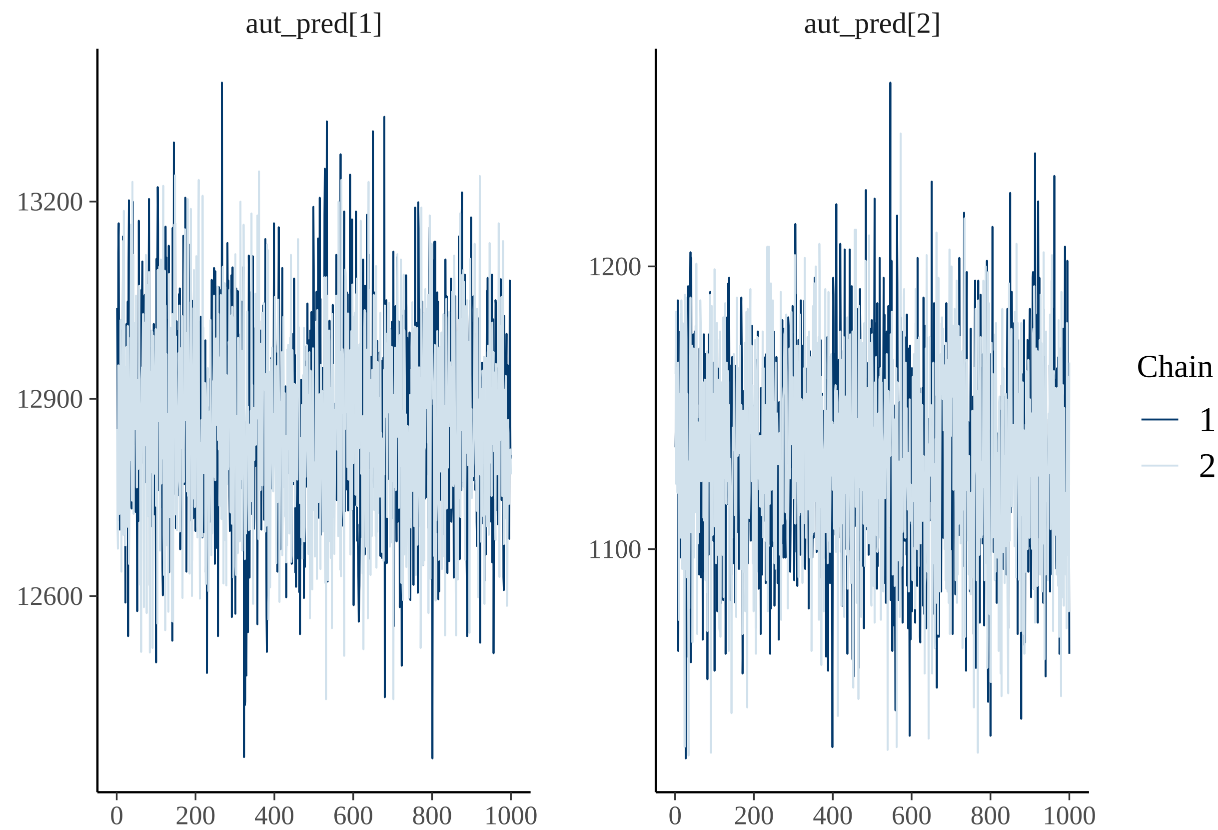



```
#geom_col(aes(x = school_region_name_abr, y = prevalence, group = sex, fill = as.factor(sex)), position
# 1 is male, 2 is female

aut_prev_rural_adj <- aut_prev_rural %>%
  group_by(school_rurality_code) %>%
  summarise(sum_sample_pop_size = sum(sample_pop_size),
            crude_rate = sum(n_autism) / sum(sample_pop_size),
            crude_count = sum(n_autism),
            adjusted_rate = sum(n_autism / sample_pop_size * pop_prop),
            adjusted_count = round(adjusted_rate * sum_sample_pop_size, 0), # had to fudge this to get 
            #adjusted_count = adjusted_rate * sum_sample_pop_size,
            var = sum(pop_prop^2 * n_autism / sample_pop_size^2),
            #se2 = sqrt(sum((std_pop/sum(std_pop))^2 * n_autism/sample_pop_size^2)),
            w_M = max(w),
            ci_lower = var / (2*adjusted_rate) * qchisq(p = 0.05/2, df = 2*adjusted_rate^2 / var),
            ci_upper = (var + w_M^2) / (2*(adjusted_rate + w_M)) * qchisq(p = 1-0.05/2, df = 2*(adjuste
  arrange(school_rurality_code)

# Prior: age and sex standardised prevalence in the whole Chile dataset
theta_mu <- 0.0046
theta_sigma <- (0.0047-0.0045) / (2*1.96)
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

nRural <- length(unique(aut_prev_rural$school_rurality_code))
```

```r
rand_rural_model <- "model {
  for(i in 1:nRural) { # For each rurality
    theta[i] ~ dbeta(theta_a, theta_b)
    aut_sample[i] ~ dbin(theta[i], nObs[i])

    aut_pred[i] ~ dbin(theta[i], nObs[i])
  }
}"

rand_rural_data <- list(theta_a = theta_a,
                         theta_b = theta_b,
                         nObs = aut_prev_rural_adj$sum_sample_pop_size,
                         aut_sample = aut_prev_rural_adj$adjusted_count,
                         nRural = nRural)

#rand_rural_ini <- list(list(theta = 0.001), #, spec = 0.5, sens = 0.5),
#                   list(theta = 0.01)) #, spec = 0.9, sens = 0.9))

rand_rural_pars <- c("theta_a", "theta_b", "theta", "aut_sample", "aut_pred")

# Run JAGS model and discard burn-in samples
rand_rural_jag <- jags.model(textConnection(rand_rural_model),
                              data = rand_rural_data,
                              #inits = rand_region_ini,
                              n.chains = 2,
                              quiet = TRUE)
update(rand_rural_jag, n.iter = nBurn)
rand_rural_sam <- coda.samples(model = rand_rural_jag,
                                variable.names = rand_rural_pars,
                                n.iter = nIter)

# Check for convergence in parameters of interest
#mcmc_trace(rand_region_sam, rand_region_pars)
mcmc_trace(rand_rural_sam, paste0("theta[", 1:nRural, "]")) # Convergence looks fine and rhats <= 1.1
```




```
mcmc_trace(rand_rural_sam, paste0("aut_pred[", 1:nRural, "]"))# Convergence looks fine and rhats <= 1.1
```




```
summary(as_draws(rand_rural_sam)) %>% print(n = Inf)
```

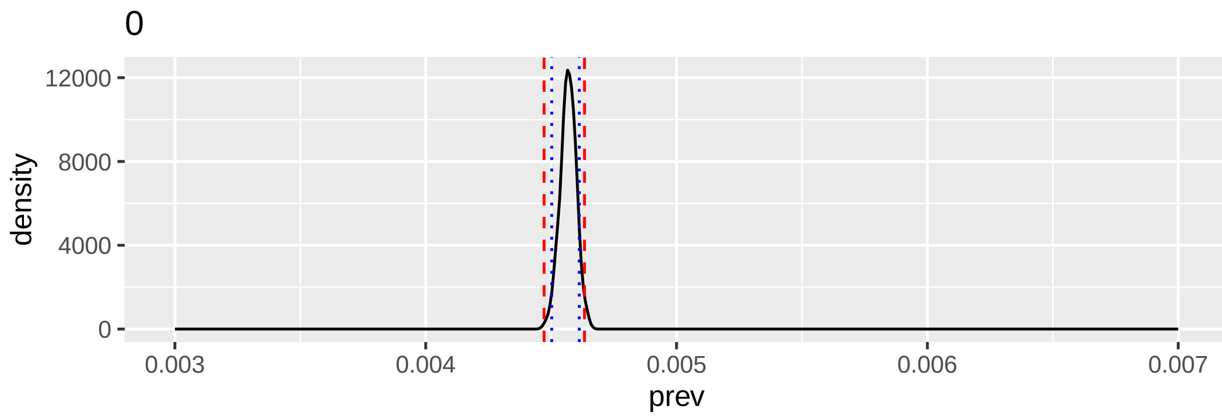
```
## # A tibble: 8 x 10
##   variable    mean  median      sd     mad      q5     q95  rhat ess_b~1 ess_t~2
##   <chr>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
## 1 aut_pre~ 1.29e+4 1.29e+4 1.47e+2 1.50e+2 1.26e+4 1.31e+4  1.00   1599.   1712.
## 2 aut_pre~ 1.13e+3 1.13e+3 3.50e+1 3.56e+1 1.07e+3 1.19e+3  1.00   2069.   1884.
## 3 aut_sam~ 1.28e+4 1.28e+4 0       0       1.28e+4 1.28e+4 NA        NA      NA
## 4 aut_sam~ 1.37e+3 1.37e+3 0       0       1.37e+3 1.37e+3 NA        NA      NA
## 5 theta[1] 4.57e-3 4.57e-3 3.19e-5 3.14e-5 4.51e-3 4.62e-3  1.00   1188.    914.
## 6 theta[2] 4.73e-3 4.74e-3 4.86e-5 4.74e-5 4.65e-3 4.82e-3  1.00   1338.    858.
## 7 theta_a  8.09e+3 8.09e+3 0       0       8.09e+3 8.09e+3 NA        NA      NA
## 8 theta_b  1.75e+6 1.75e+6 0       0       1.75e+6 1.75e+6 NA        NA      NA
## # ... with abbreviated variable names 1: ess_bulk, 2: ess_tail
```

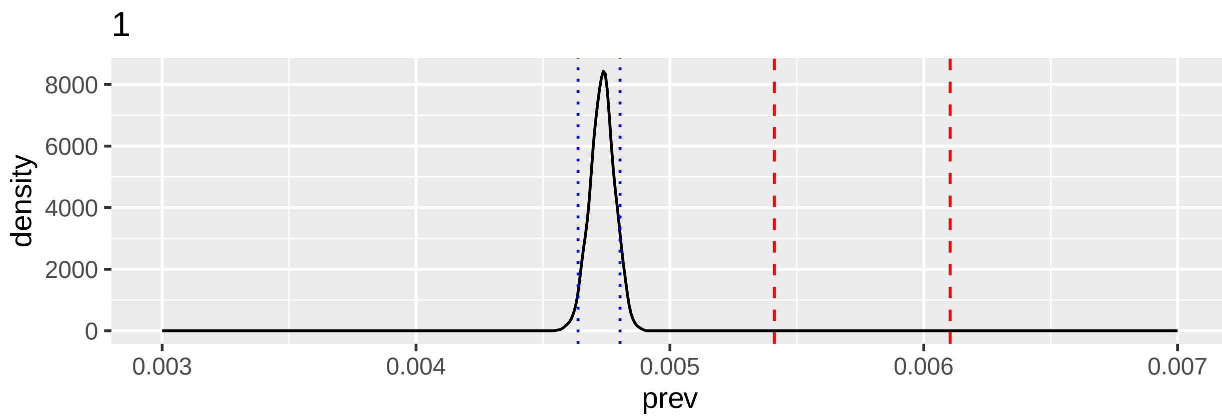
```
rand_rural_summ <- summary(subset_draws(as_draws(rand_rural_sam), rand_rural_pars),
                           ~quantile(.x, probs=c(0.025, 0.5, 0.975)),
                           ~mcse_quantile(.x, probs=c(0.025, 0.5, 0.975)),
                           "rhat") %>%
  arrange(desc(mcse_q50))
rand_rural_summ
```

```
## # A tibble: 8 x 8
##   variable           `2.5%`     `50%` `97.5%` mcse_q~1 mcse_q50 mcse_q~2  rhat
##   <chr>               <dbl>     <dbl>   <dbl>    <dbl>    <dbl>    <dbl> <dbl>
## 1 aut_pred[1]      12586.     1.29e+4 1.32e+4  1.05e+1  4    e+0  7.5 e+0  1.00
## 2 aut_pred[2]       1063      1.13e+3 1.20e+3  1.5 e+0  1    e+0  2.5 e+0  1.00
```

```
## 3 theta[2]           0.00464   4.74e-3 4.83e-3  5.29e-6  1.66e-6  2.42e-6  1.00
## 4 theta[1]           0.00450   4.57e-3 4.63e-3  4.03e-6  1.10e-6  2.56e-6  1.00
## 5 theta_a         8091.        8.09e+3 8.09e+3 NA       NA       NA       NA
## 6 theta_b      1750915.        1.75e+6 1.75e+6 NA       NA       NA       NA
## 7 aut_sample[1]   12823        1.28e+4 1.28e+4 NA       NA       NA       NA
## 8 aut_sample[2]    1370        1.37e+3 1.37e+3 NA       NA       NA       NA
## # ... with abbreviated variable names 1: mcse_q2.5, 2: mcse_q97.5
```

```
aut_prev_rural_plots <- list()
rural_post_ci_lower <- list()
rural_post_ci_upper <- list()

for(i in 1:nRural) {
  prevs <- data.frame(prev = extract_variable(rand_rural_sam, paste0("theta[", i, "]")))
  rural_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
  rural_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
  density_plot <- ggplot(prevs, aes(x = prev), color =  "blue") +
    geom_density() +
    xlim(c(0.003, 0.007)) +
    geom_vline(xintercept = rural_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = rural_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = aut_prev_rural_adj$ci_lower[i], color = "red", linetype = "dashed") +
    geom_vline(xintercept = aut_prev_rural_adj$ci_upper[i], color = "red", linetype = "dashed") +
    labs(title = aut_prev_rural_adj$school_rurality_code[i])
  aut_prev_rural_plots[[i]] <- density_plot
}
do.call(grid.arrange, aut_prev_rural_plots)
```



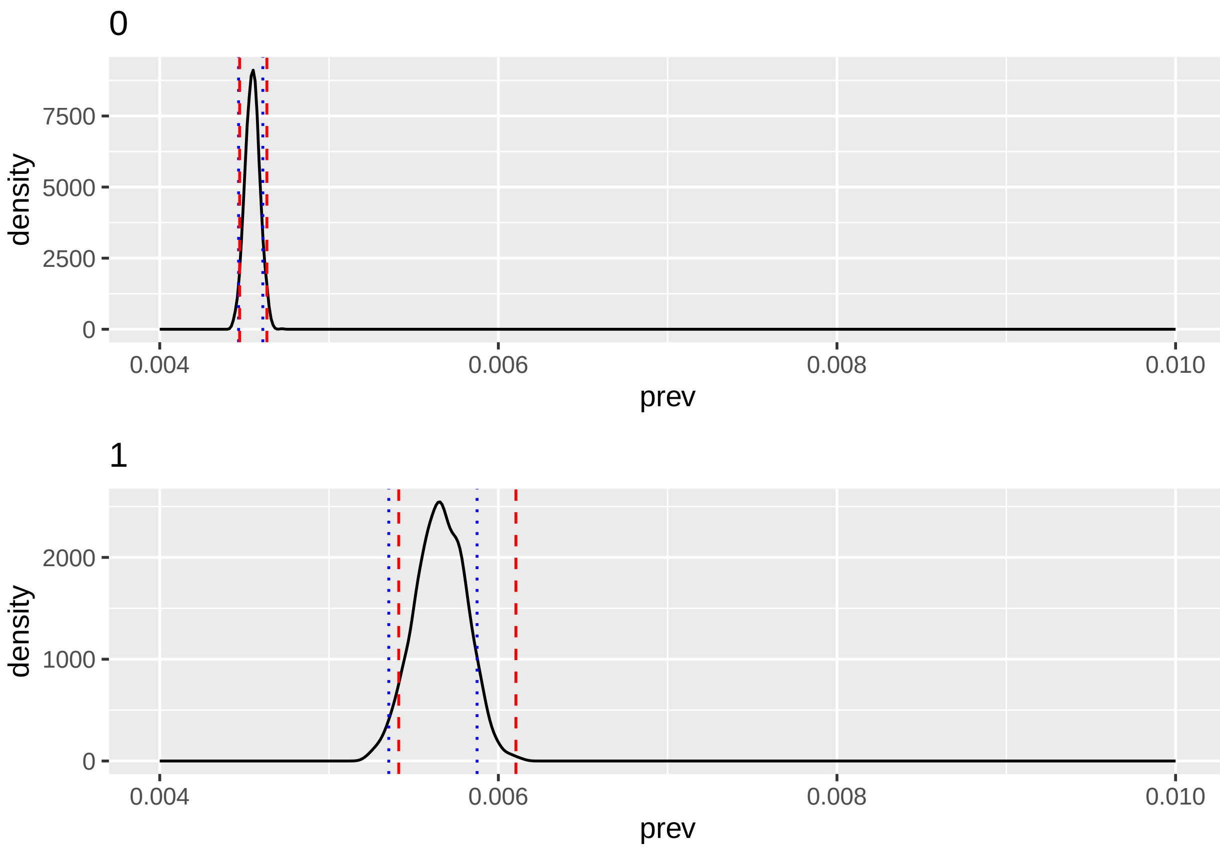

```
#autism_prev_rural_plots <- do.call(grid.arrange, aut_prev_rural_plots)
#ggsave("autism_prev_rural_plots.png", autism_prev_rural_plots, height = 10, width = 15)
```

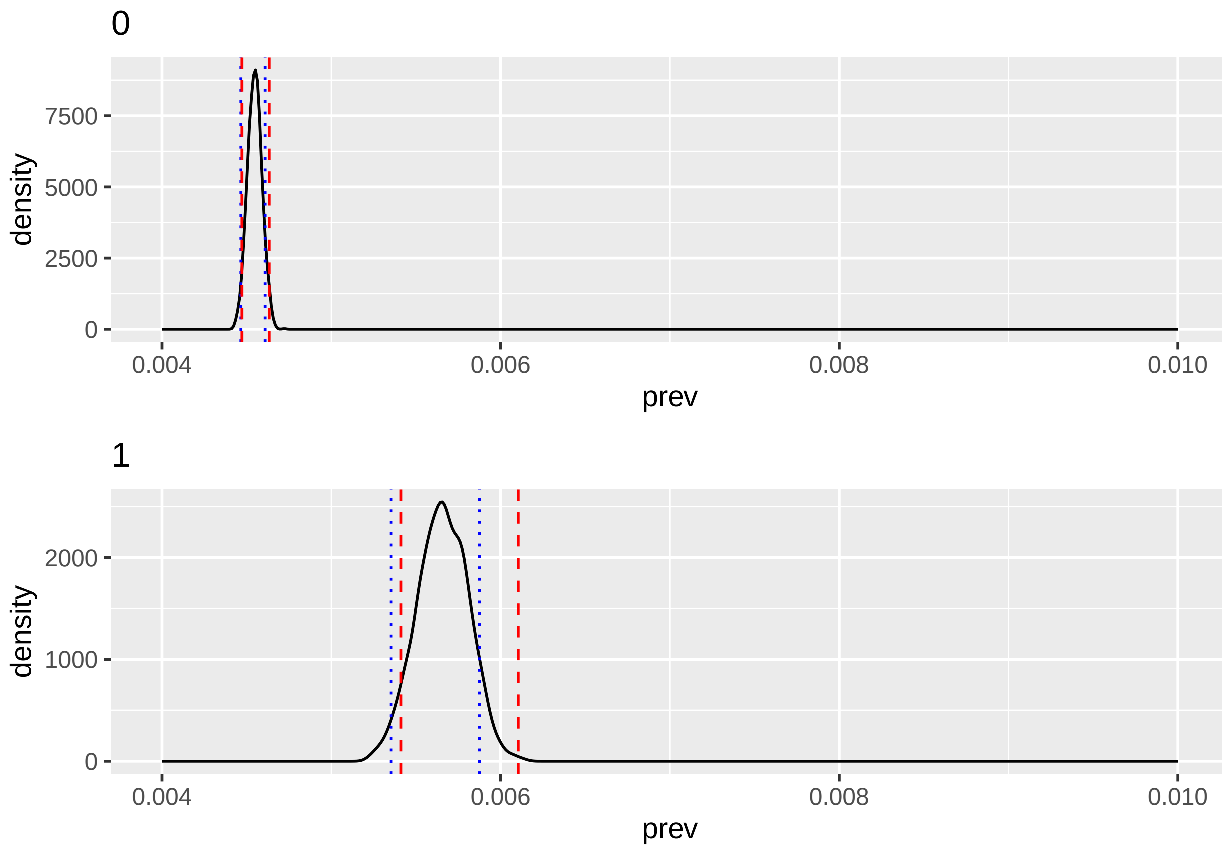
Assuming 0 = city, 1 = rural. Narrower CI for city because sample size is bigger

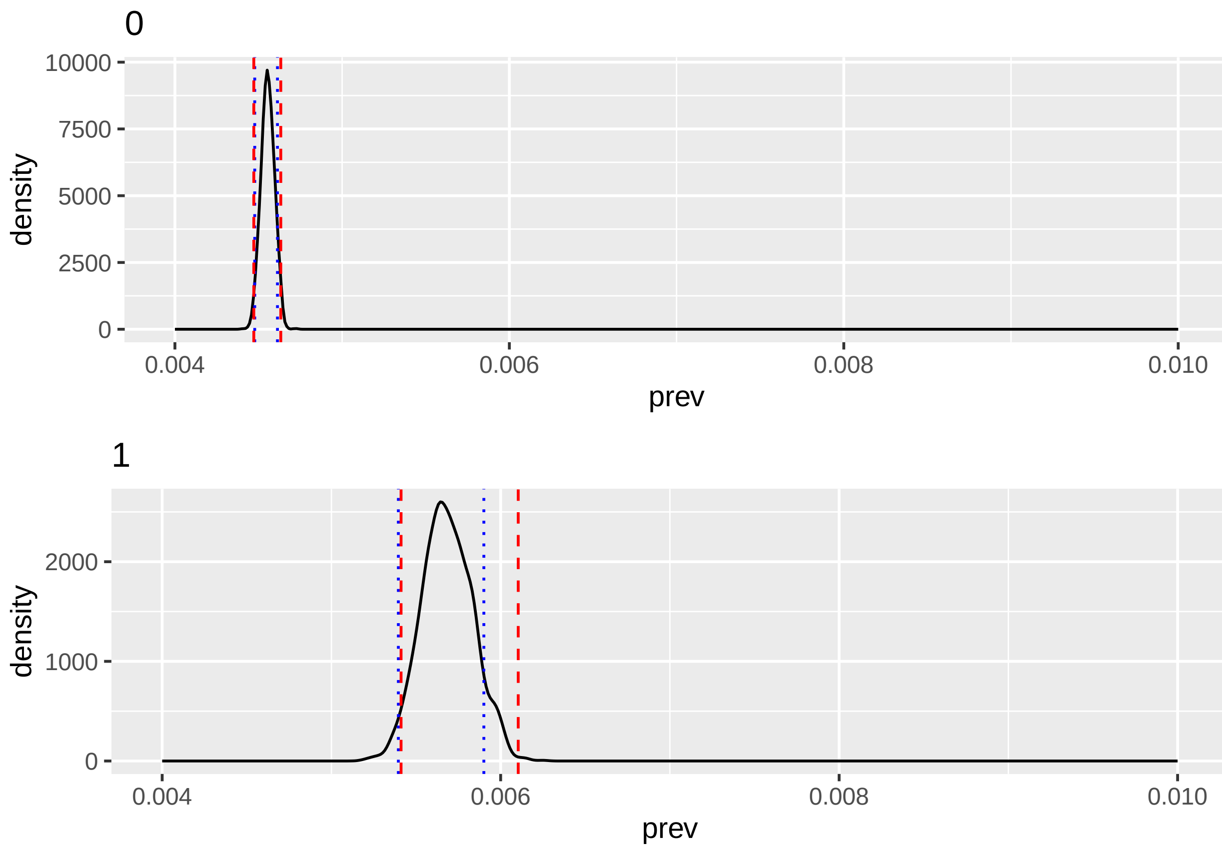
#### Sensitivity analysis - alter prior mean and sd

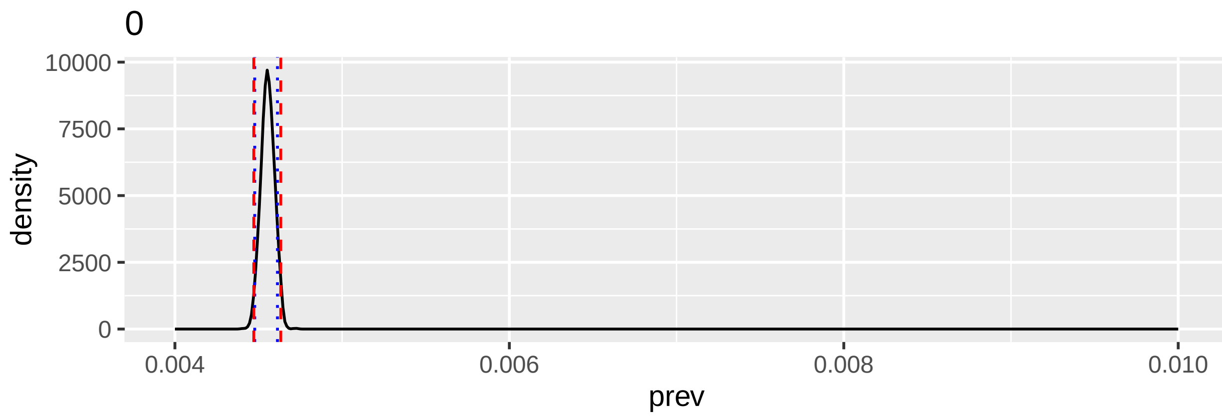
```
theta_mu <- c(0.001, 0.005, 0.01, 0.02, # 0.1%, 0.5%, 1%, 2% prevalence
              rep(0.0046, 4)) # Same as chosen prior
theta_sigma <- c(rep(0.001/1.96, 4), # Same as chosen prior
                 0.0001, 0.001, 0.05, 0.01) # +/- 0.1%, 0.5%, 1%, 5%
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

for(j in 1:length(theta_mu)) {
  rand_rural_data <- list(theta_a = theta_a[j],
                          theta_b = theta_b[j],
                          nObs = aut_prev_rural_adj$sum_sample_pop_size,
                          aut_sample = aut_prev_rural_adj$adjusted_count,
                          nRural = nRural)
  rand_rural_jag <- jags.model(textConnection(rand_rural_model),
                               data = rand_rural_data,
                               #inits = rand_region_ini,
                               n.chains = 2,
                               quiet = TRUE)
  update(rand_rural_jag, n.iter = nBurn)
```

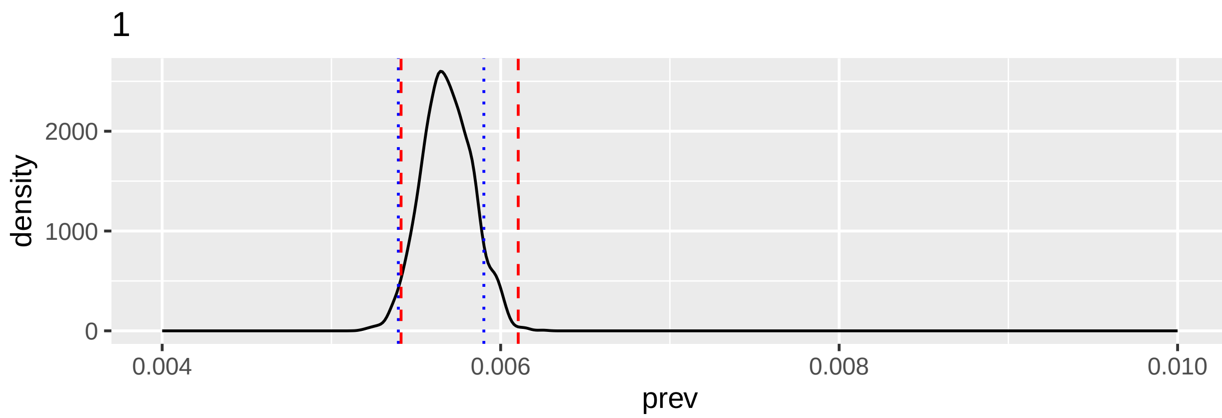
```
  rand_rural_sam <- coda.samples(model = rand_rural_jag,
                                 variable.names = rand_rural_pars,
                                 n.iter = nIter)

  # Plots
  aut_prev_rural_plots <- list()
  rural_post_ci_lower <- list()
  rural_post_ci_upper <- list()

  for(i in 1:nRural) {
    prevs <- data.frame(prev = extract_variable(rand_rural_sam, paste0("theta[", i, "]")))
    rural_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
    rural_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
    density_plot <- ggplot(prevs, aes(x = prev), color =  "blue") +
      geom_density() +
      xlim(c(0.004, 0.01)) +
      geom_vline(xintercept = rural_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = rural_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = aut_prev_rural_adj$ci_lower[i], color = "red", linetype = "dashed") +
      geom_vline(xintercept = aut_prev_rural_adj$ci_upper[i], color = "red", linetype = "dashed") +
      labs(title = aut_prev_rural_adj$school_rurality_code[i])
    aut_prev_rural_plots[[i]] <- density_plot
  }
  do.call(grid.arrange, aut_prev_rural_plots)
  autism_prev_rural_plots <- do.call(grid.arrange, aut_prev_rural_plots)
  ggsave(paste0("autism_prev_rural_plots_", j, ".png"), autism_prev_rural_plots, height = 10, width = 15
}
```

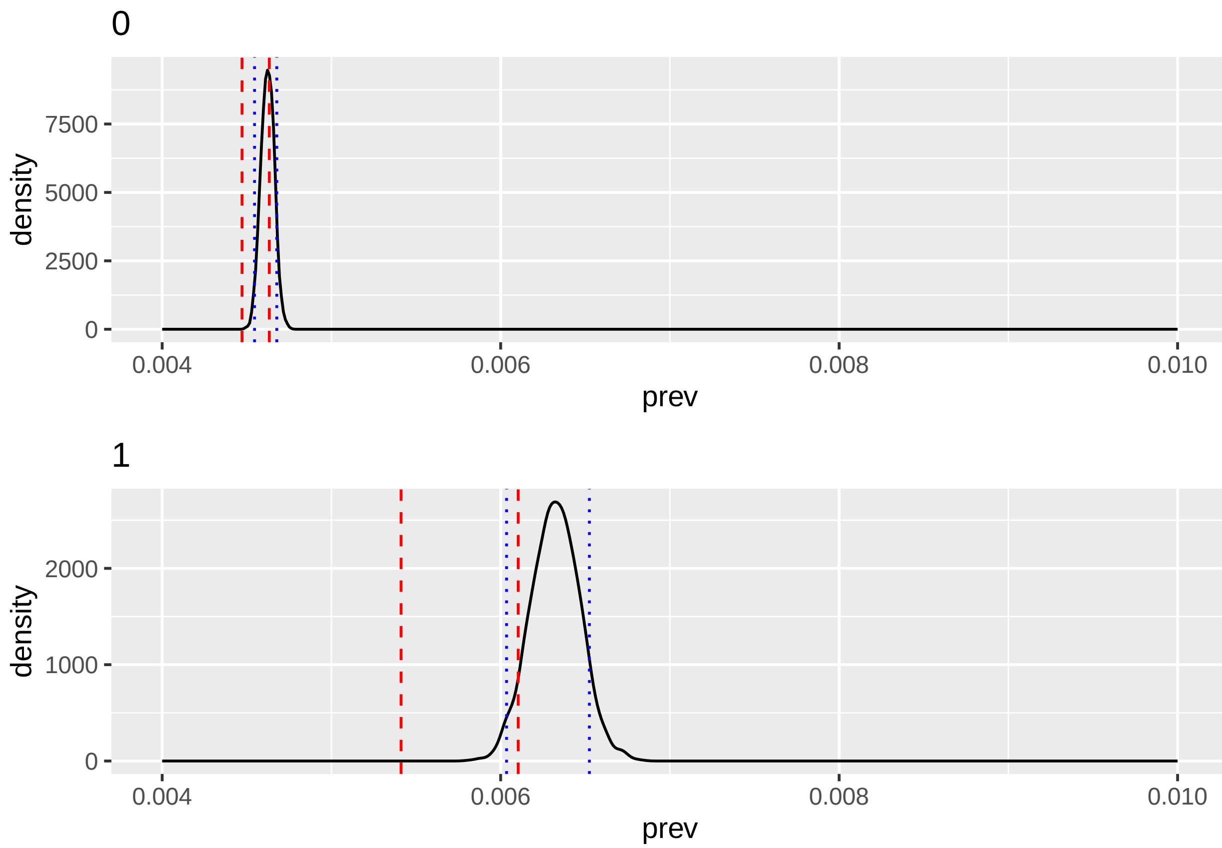

0
7500
5000
2500
0
density
0.004
0.006
0.008
0.010
prev
1
2000
1000
0
density
0.004
0.006
0.008
0.010
prev


0
density
7500
5000
2500
0
0.004
0.006
0.008
0.010
prev
1
density
2000
1000
0
0.004
0.006
0.008
0.010
prev

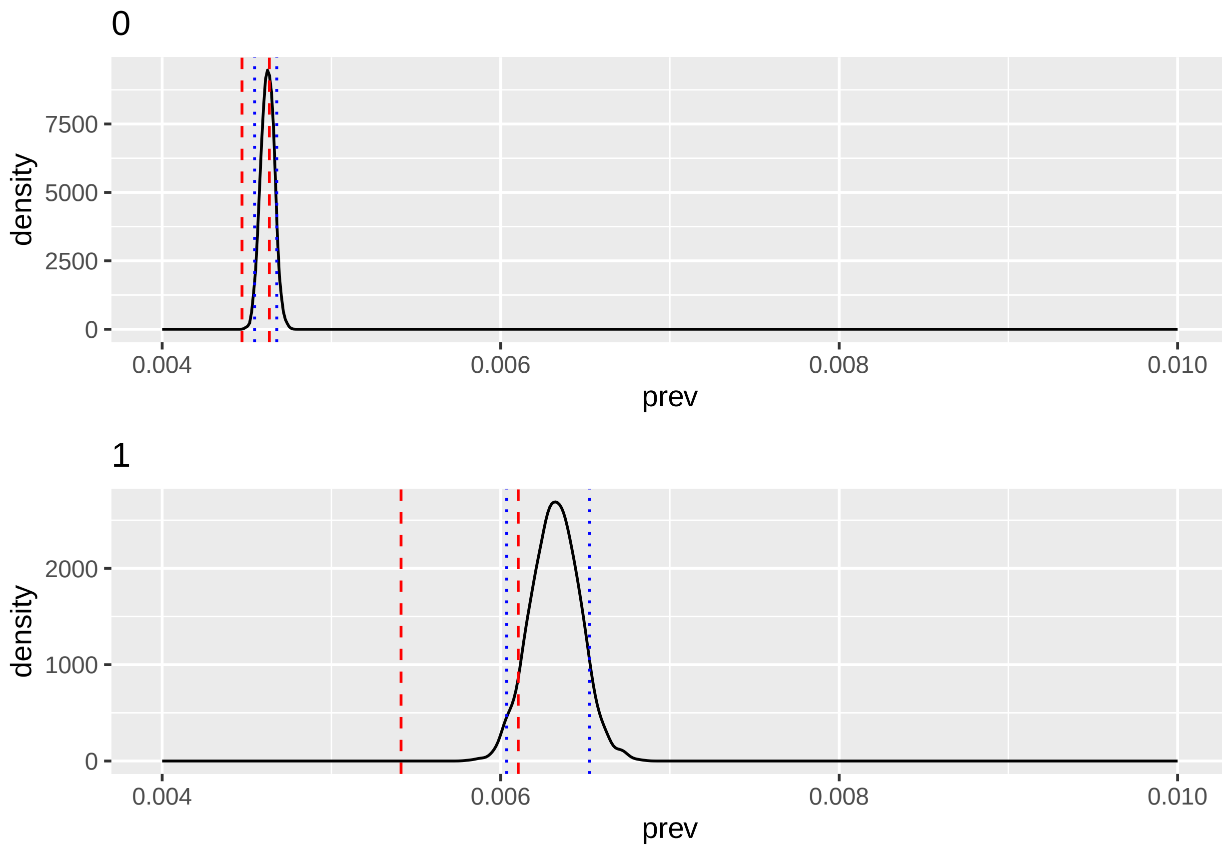

0
10000
7500
5000
2500
0
density
0.004
0.006
0.008
0.010
prev
1
2000
1000
0
density
0.004
0.006
0.008
0.010
prev

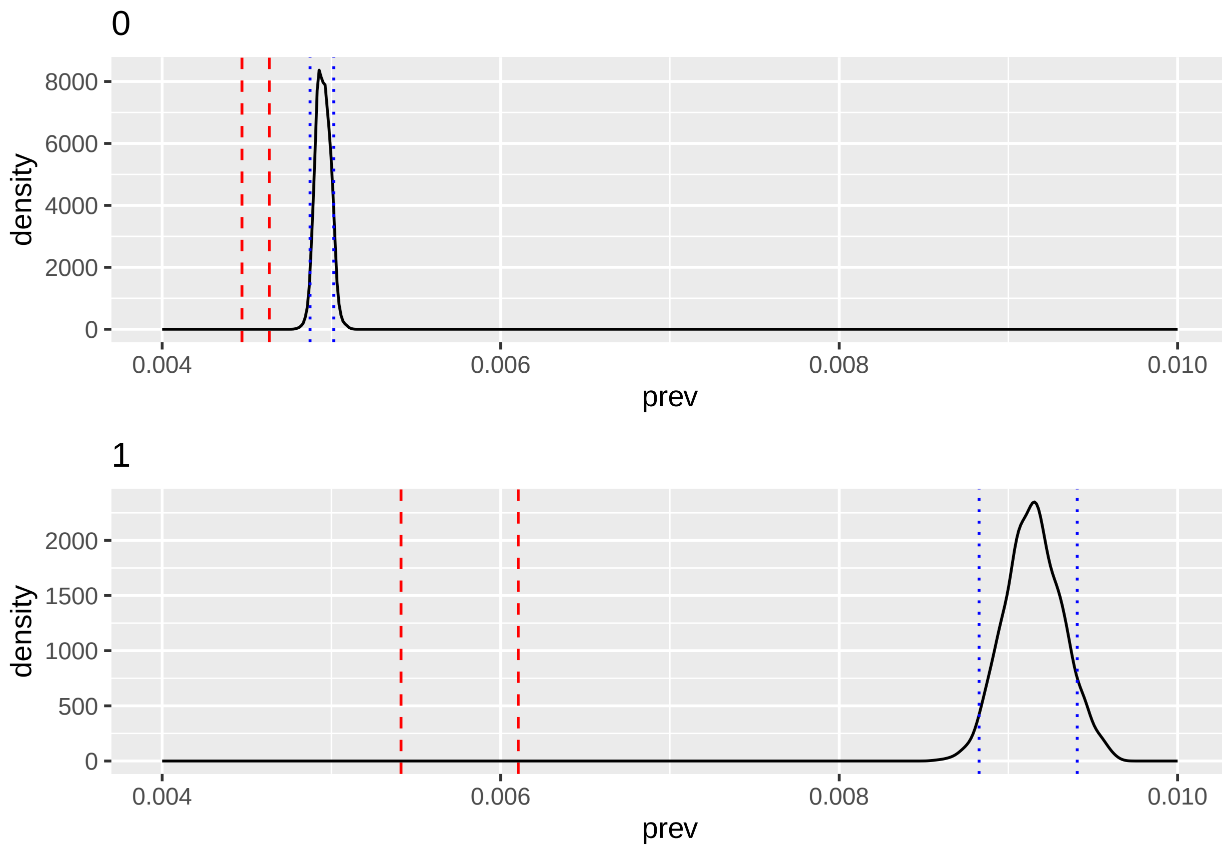

0
10000
7500
5000
2500
0
density
0.004
0.006
0.008
0.010
prev

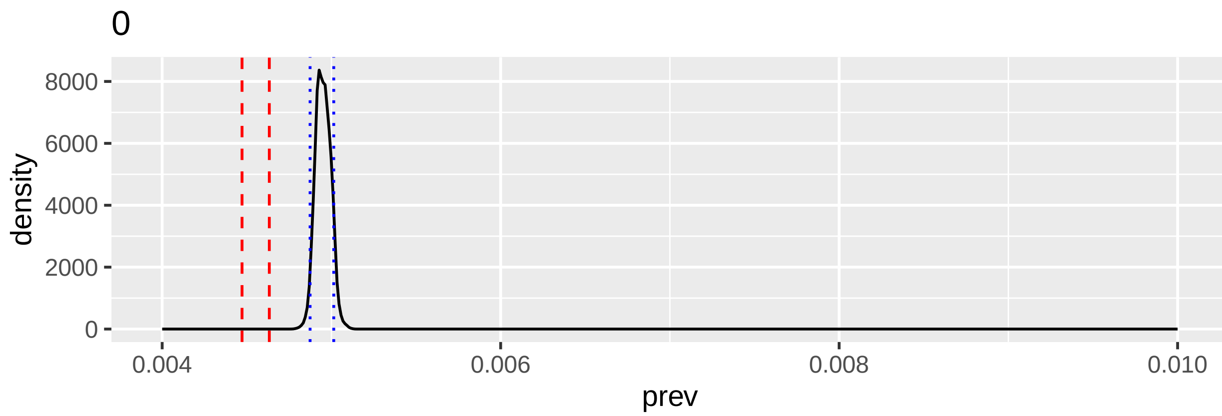

1
2000
1000
0
density
0.004
0.006
0.008
0.010
prev

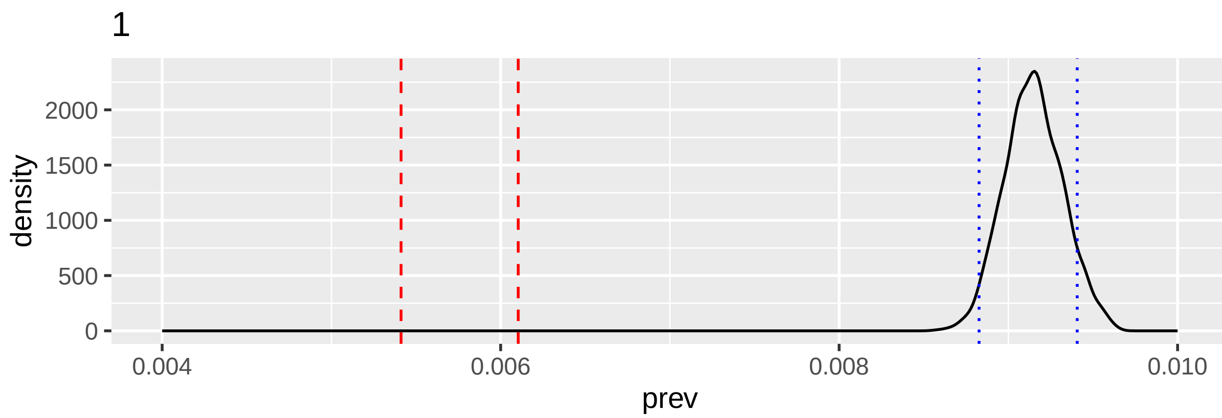


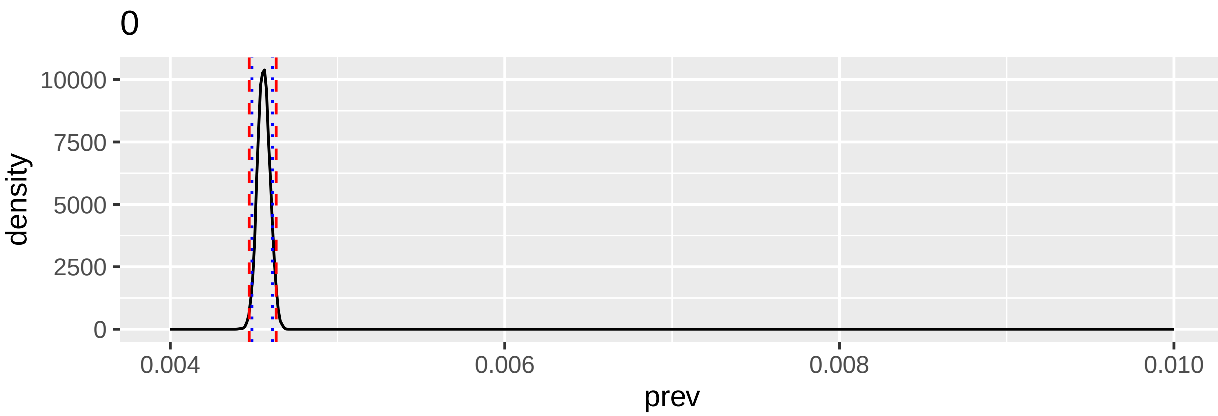
0
density
7500
5000
2500
0
0.004
0.006
0.008
0.010
prev
1
density
2000
1000
0
0.004
0.006
0.008
0.010
prev

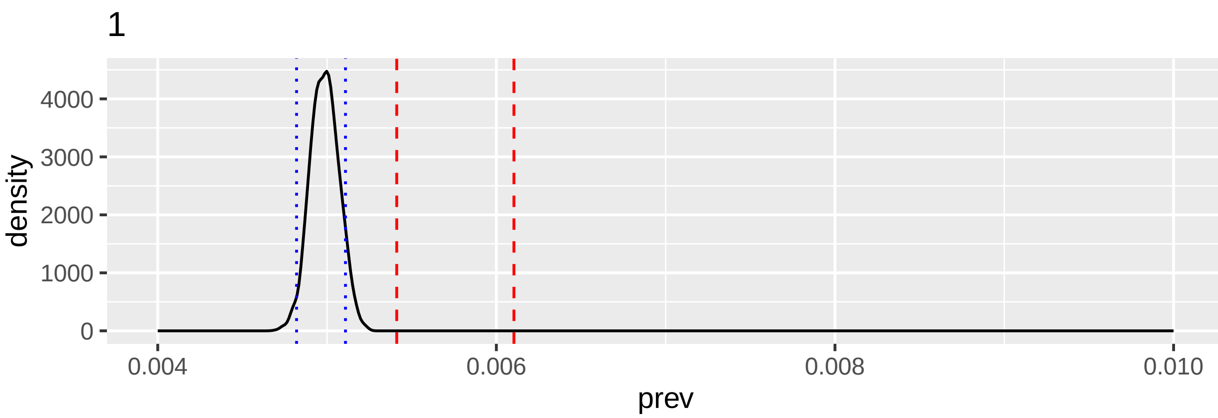

0
density
7500
5000
2500
0
0.004
0.006
0.008
0.010
prev
1
density
2000
1000
0
0.004
0.006
0.008
0.010
prev

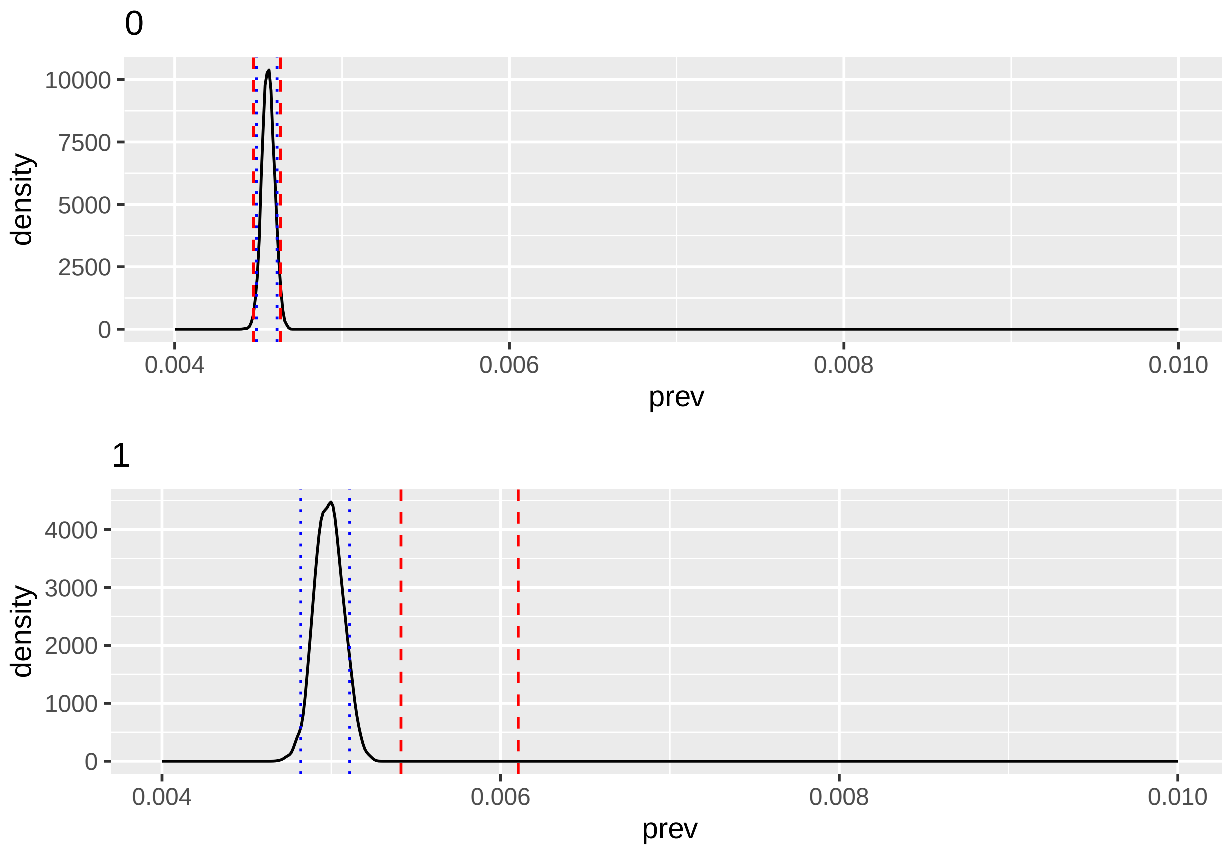

0
8000
6000
4000
2000
0
density
0.004
0.006
0.008
0.010
prev
1
2000
1500
1000
500
0
density
0.004
0.006
0.008
0.010
prev

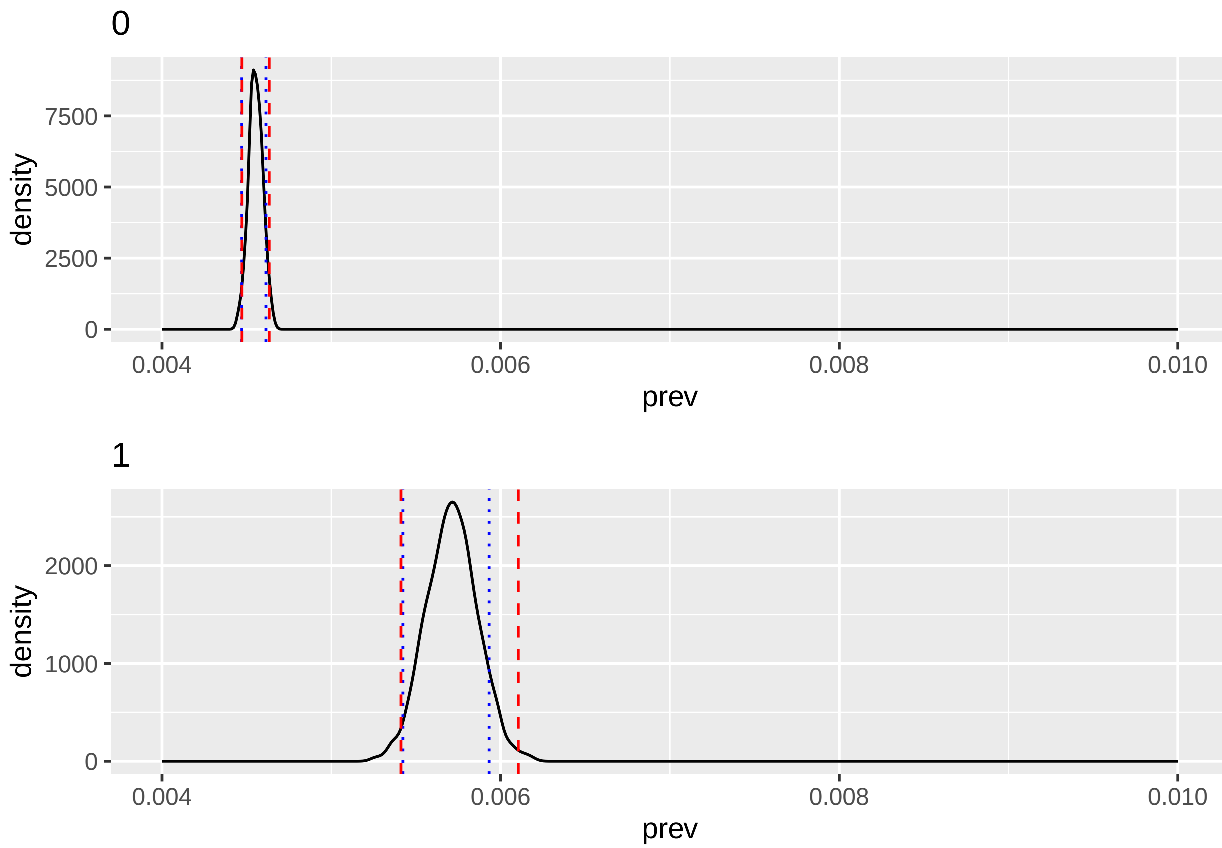

0
density
8000
6000
4000
2000
0
0.004
0.006
0.008
0.010
prev


1
density
2000
1500
1000
500
0
0.004
0.006
0.008
0.010
prev

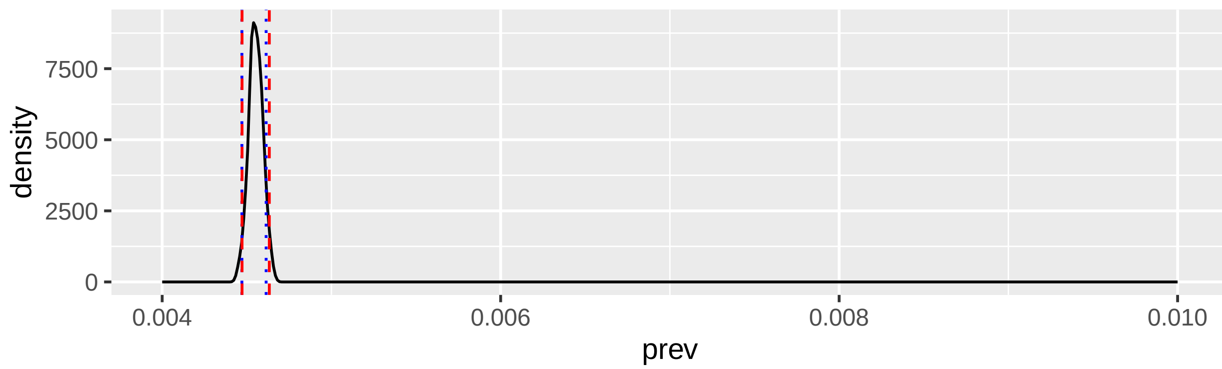

0
10000
7500
5000
2500
0
density
0.004
0.006
0.008
0.010
prev


1
4000
3000
2000
1000
0
density
0.004
0.006
0.008
0.010
prev

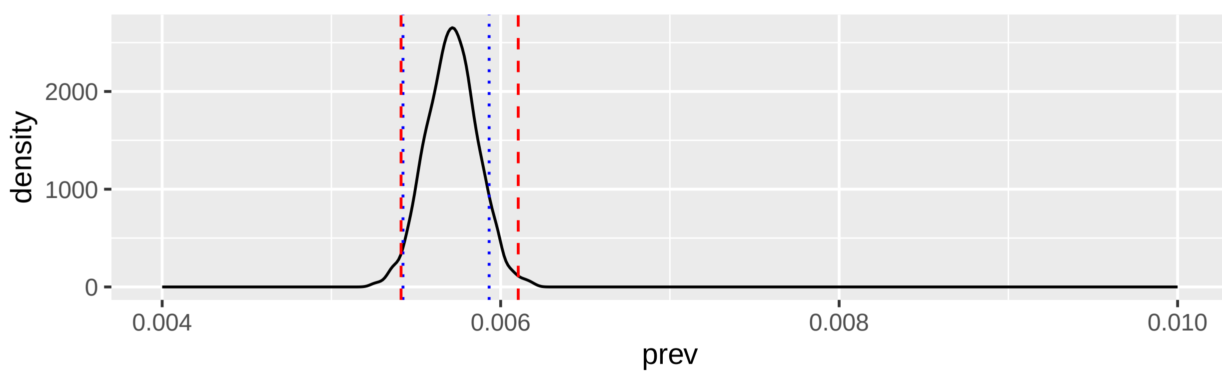

0
density
10000
7500
5000
2500
0
0.004
0.006
0.008
0.010
prev
1
density
4000
3000
2000
1000
0
0.004
0.006
0.008
0.010
prev

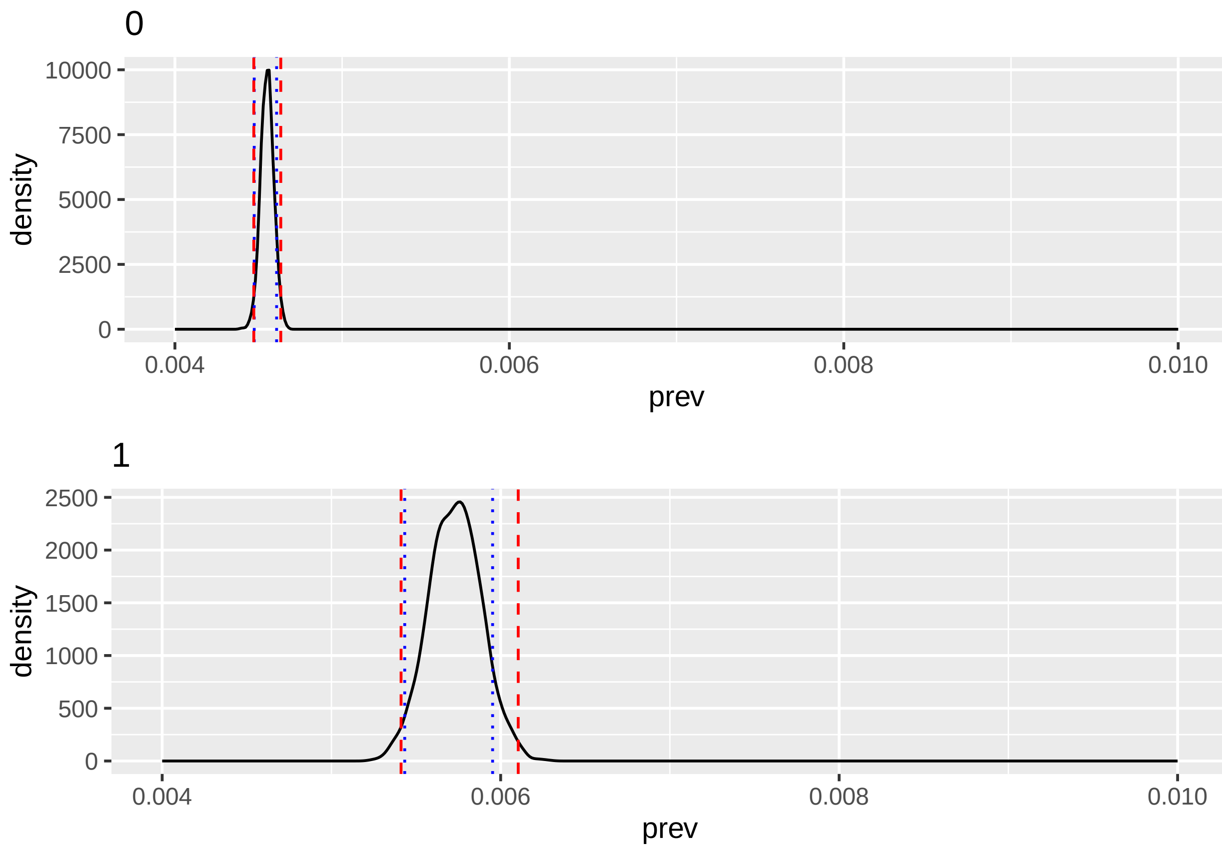

0
density
7500
5000
2500
0
0.004
0.006
0.008
0.010
prev
1
density
2000
1000
0
0.004
0.006
0.008
0.010
prev

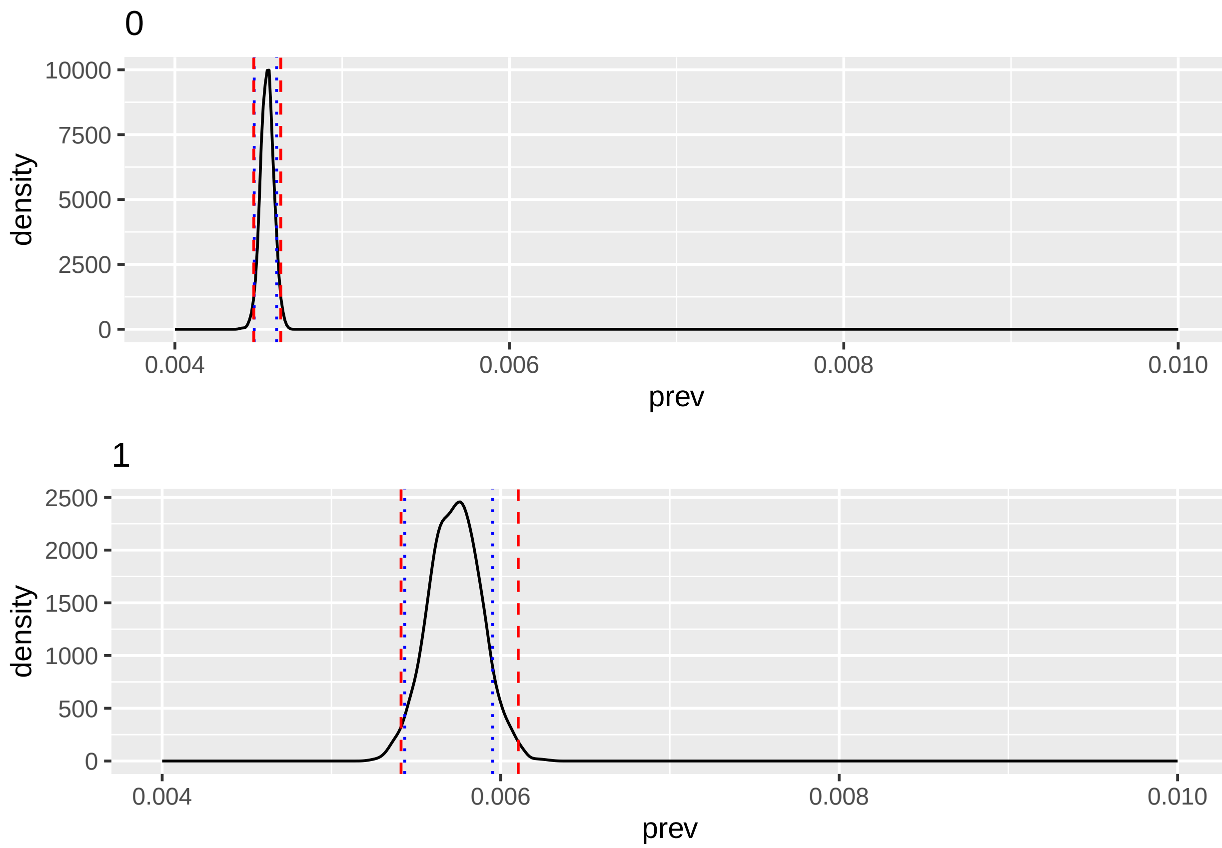
0

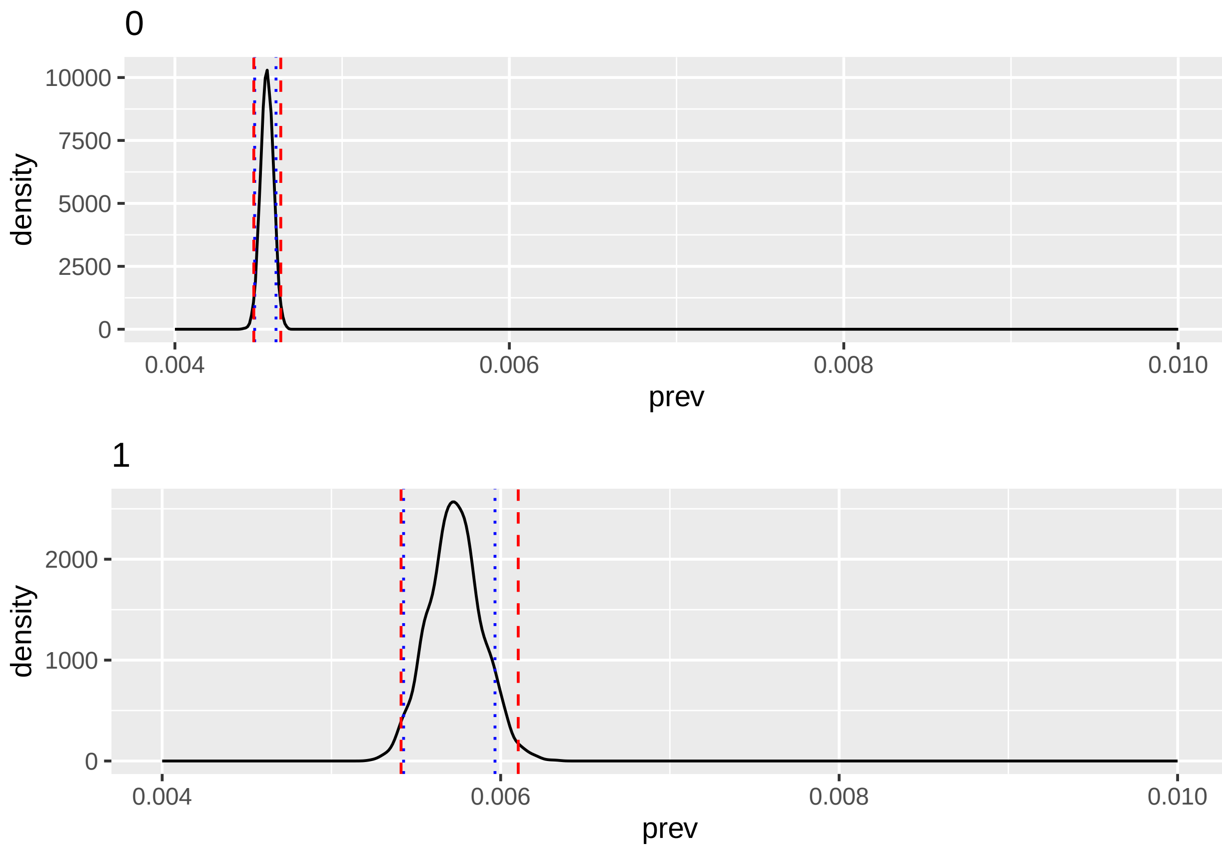

density
7500
5000
2500
0
0.004
0.006
0.008
0.010
prev

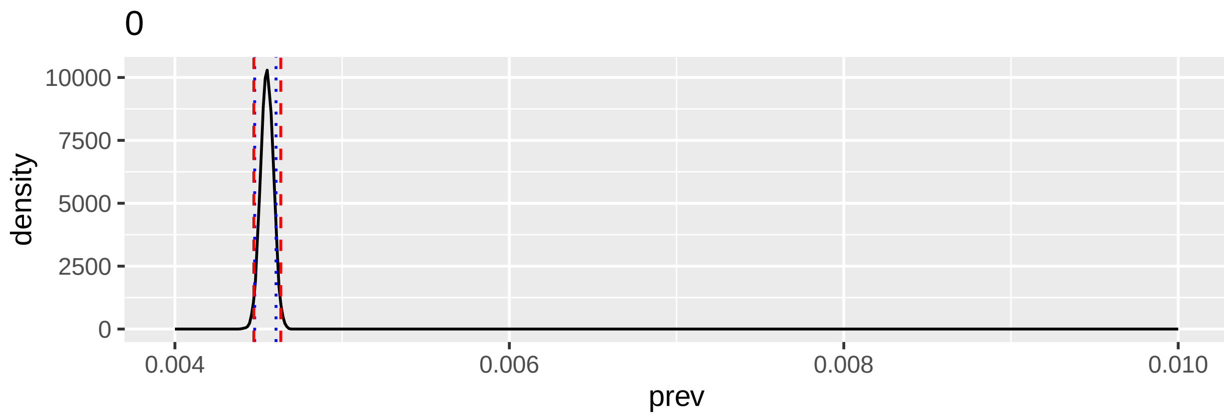
1

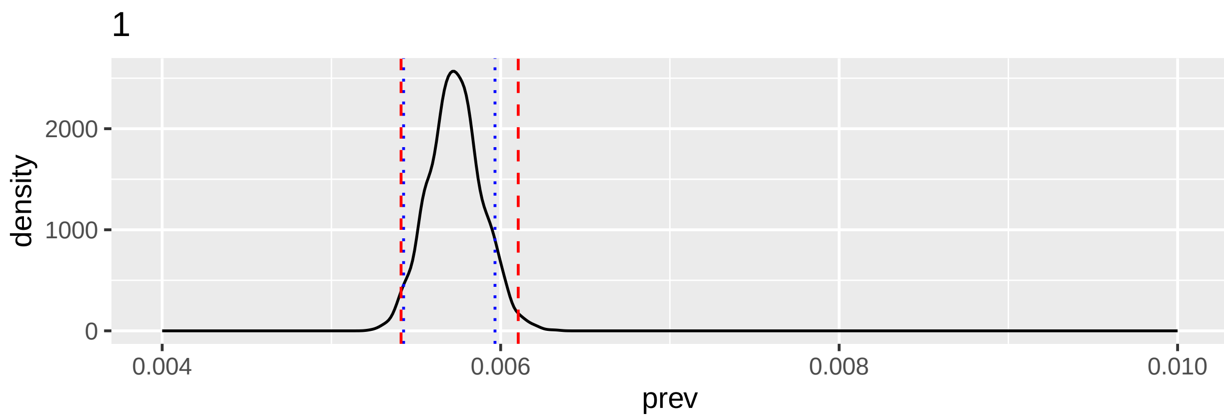

density
2000
1000
0
0.004
0.006
0.008
0.010
prev


0
density
10000
7500
5000
2500
0
0.004
0.006
0.008
0.010
prev
1
density
2500
2000
1500
1000
500
0
0.004
0.006
0.008
0.010
prev


0
10000
7500
5000
2500
0
density
0.004
0.006
0.008
0.010
prev
1
2500
2000
1500
1000
500
0
density
0.004
0.006
0.008
0.010
prev


0
10000
7500
5000
2500
0
density
0.004
0.006
0.008
0.010
prev
1
2000
1000
0
density
0.004
0.006
0.008
0.010
prev



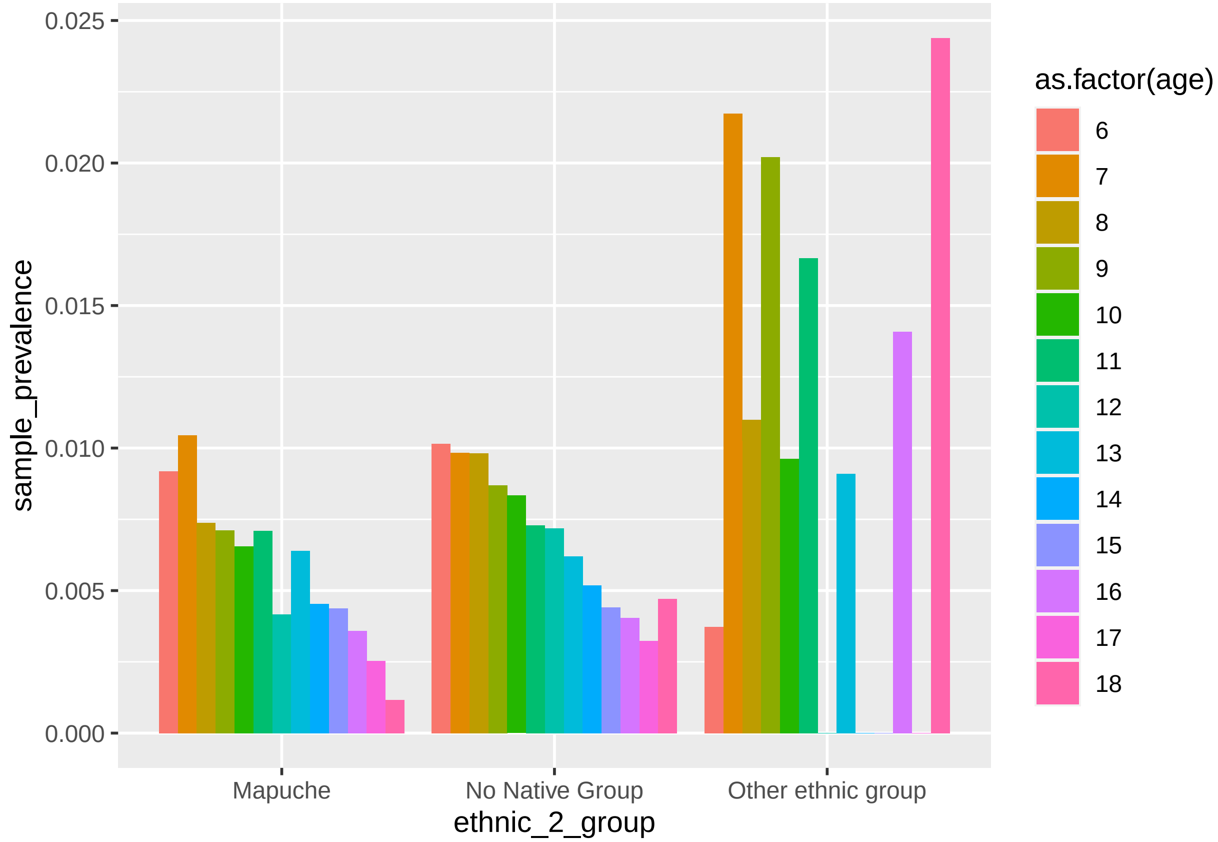




```
################################################################################

### Bayesian prevalence by ethnicity

aut_prev_ethnic <- chile_bayes_aut %>%
  filter(school_region_name_abr %in% c("ARAUC", "BBIO", "LAGOS", "RIOS", "RM")) %>%
  group_by(ethnic_3_group, age_june30, sex, autism) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = autism, values_from = count) %>%
  rename("n_noautism" = "0", "n_autism" = "1", "age" = "age_june30") %>%
  mutate(ethnic_2_group = ifelse(ethnic_3_group == "Aymara", "Other ethnic group", ethnic_3_group),
         n_autism = ifelse(is.na(n_autism), 0, n_autism),
         sample_pop_size = n_noautism + n_autism,
         sample_prevalence = n_autism / sample_pop_size) %>%
  left_join(chile_stdpop, by = c("age", "sex")) %>%
  mutate(aut_prev_std = n_autism / sample_pop_size * pop_prop,
         w = std_pop / (sample_pop_size * n_std_pop),
         w2 = pop_prop / sample_pop_size,
         #sum_std_pop = sum(std_pop)
  ) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'ethnic_3_group', 'age_june30', 'sex'. You
## can override using the `.groups` argument.
```

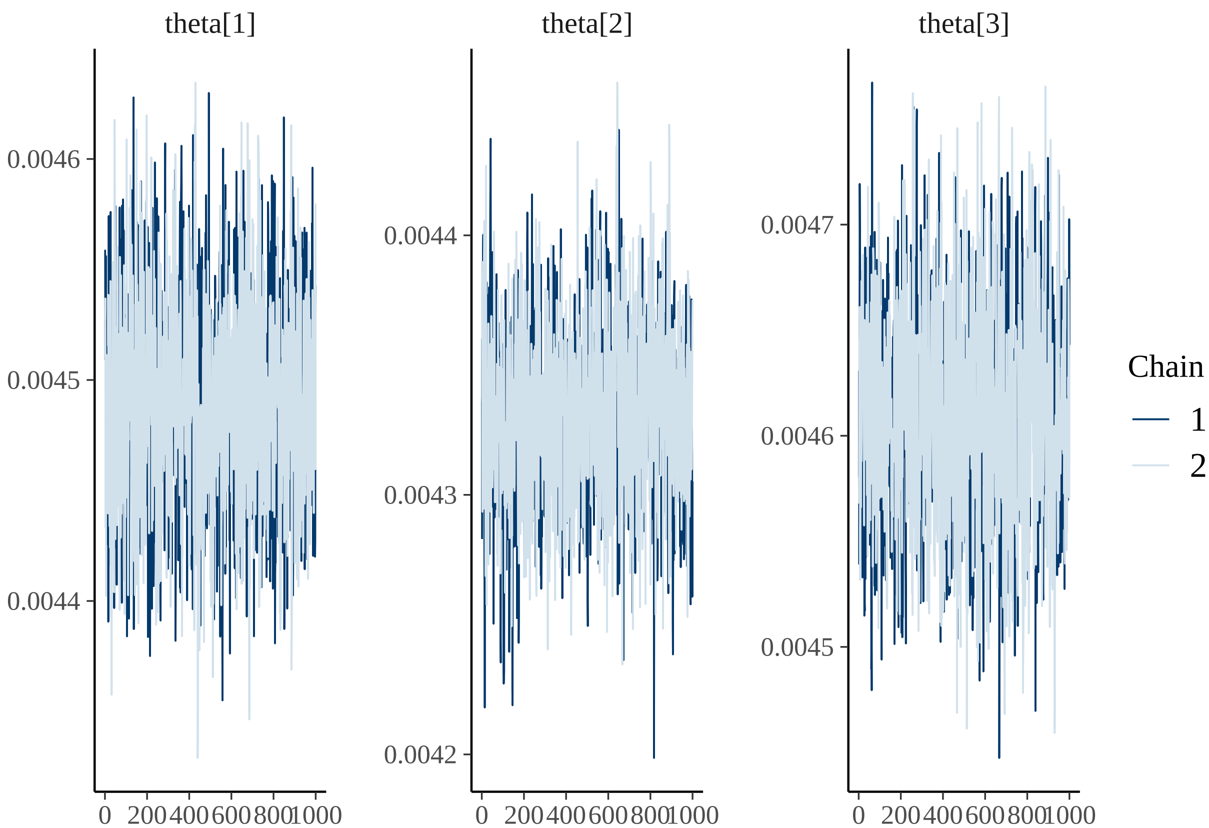
```
ggplot(data = aut_prev_ethnic) +
  #geom_col(aes(x = ethnic_3_group, y = sample_prevalence, group = age, fill = as.factor(age)), positio
```

```
  geom_col(aes(x = ethnic_2_group, y = sample_prevalence, group = age, fill = as.factor(age)), position
```

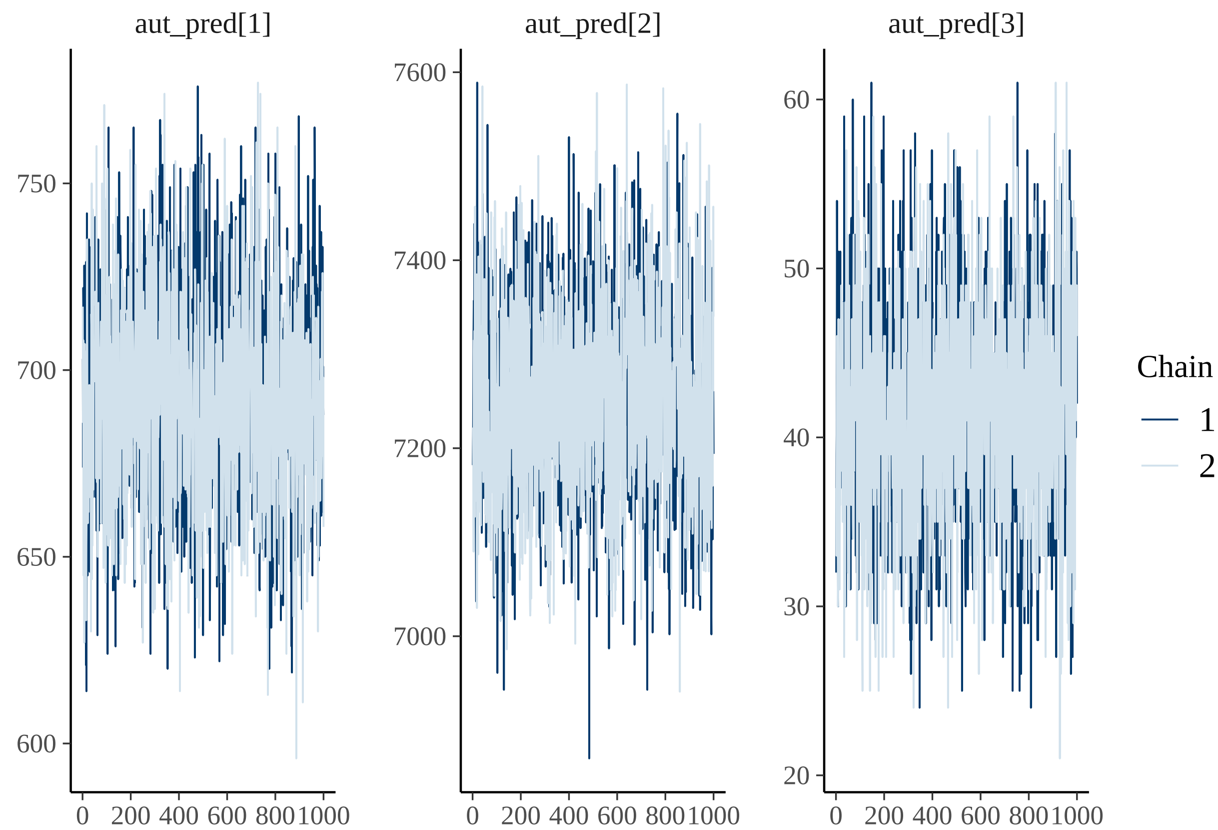



```
  #geom_col(aes(x = ethnic_3_group, y = sample_prevalence, group = sex, fill = as.factor(sex)), positio
# 1 is male, 2 is female

aut_prev_ethnic_adj <- aut_prev_ethnic %>%
  #group_by(ethnic_3_group) %>%
  group_by(ethnic_2_group) %>%
  summarise(sum_sample_pop_size = sum(sample_pop_size),
            crude_rate = sum(n_autism) / sum(sample_pop_size),
            crude_count = sum(n_autism),
            adjusted_rate = sum(n_autism / sample_pop_size * pop_prop),
            adjusted_count = round(adjusted_rate * sum_sample_pop_size, 0), # had to fudge this to get
            #adjusted_count = adjusted_rate * sum_sample_pop_size,
            var = sum(pop_prop^2 * n_autism / sample_pop_size^2),
            #se2 = sqrt(sum((std_pop/sum(std_pop))^2 * n_autism/sample_pop_size^2)),
            w_M = max(w),
            ci_lower = var / (2*adjusted_rate) * qchisq(p = 0.05/2, df = 2*adjusted_rate^2 / var),
            ci_upper = (var + w_M^2) / (2*(adjusted_rate + w_M)) * qchisq(p = 1-0.05/2, df = 2*(adjuste
  #arrange(ethnic_3_group)
  arrange(ethnic_2_group)

# Prior: age and sex standardised prevalence in the whole Chile dataset
theta_mu <- 0.0046
theta_sigma <- (0.0047-0.0045) / (2*1.96)
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
```

```r
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

#nEthnic <- length(unique(aut_prev_ethnic$ethnic_3_group))
nEthnic <- length(unique(aut_prev_ethnic$ethnic_2_group))

rand_ethnic_model <- "model {
  for(i in 1:nEthnic) { # For each ethnic group
    theta[i] ~ dbeta(theta_a, theta_b)
    aut_sample[i] ~ dbin(theta[i], nObs[i])

    aut_pred[i] ~ dbin(theta[i], nObs[i])
  }
}"

rand_ethnic_data <- list(theta_a = theta_a,
                         theta_b = theta_b,
                         nObs = aut_prev_ethnic_adj$sum_sample_pop_size,
                         aut_sample = aut_prev_ethnic_adj$adjusted_count,
                         nEthnic = nEthnic)

#rand_rural_ini <- list(list(theta = 0.001), #, spec = 0.5, sens = 0.5),
#                    list(theta = 0.01)) #, spec = 0.9, sens = 0.9))

rand_ethnic_pars <- c("theta_a", "theta_b", "theta", "aut_sample", "aut_pred")

# Run JAGS model and discard burn-in samples
rand_ethnic_jag <- jags.model(textConnection(rand_ethnic_model),
                              data = rand_ethnic_data,
                              #inits = rand_region_ini,
                              n.chains = 2,
                              quiet = TRUE)
update(rand_ethnic_jag, n.iter = nBurn)
rand_ethnic_sam <- coda.samples(model = rand_ethnic_jag,
                                variable.names = rand_ethnic_pars,
                                n.iter = nIter)

# Check for convergence in parameters of interest
#mcmc_trace(rand_region_sam, rand_region_pars)
mcmc_trace(rand_ethnic_sam, paste0("theta[", 1:nEthnic, "]")) # Convergence looks fine and rhats <= 1.1
```




```
mcmc_trace(rand_ethnic_sam, paste0("aut_pred[", 1:nEthnic, "]"))# Convergence looks fine and rhats <= 1
```




```r
summary(as_draws(rand_ethnic_sam)) %>% print(n = Inf)
```

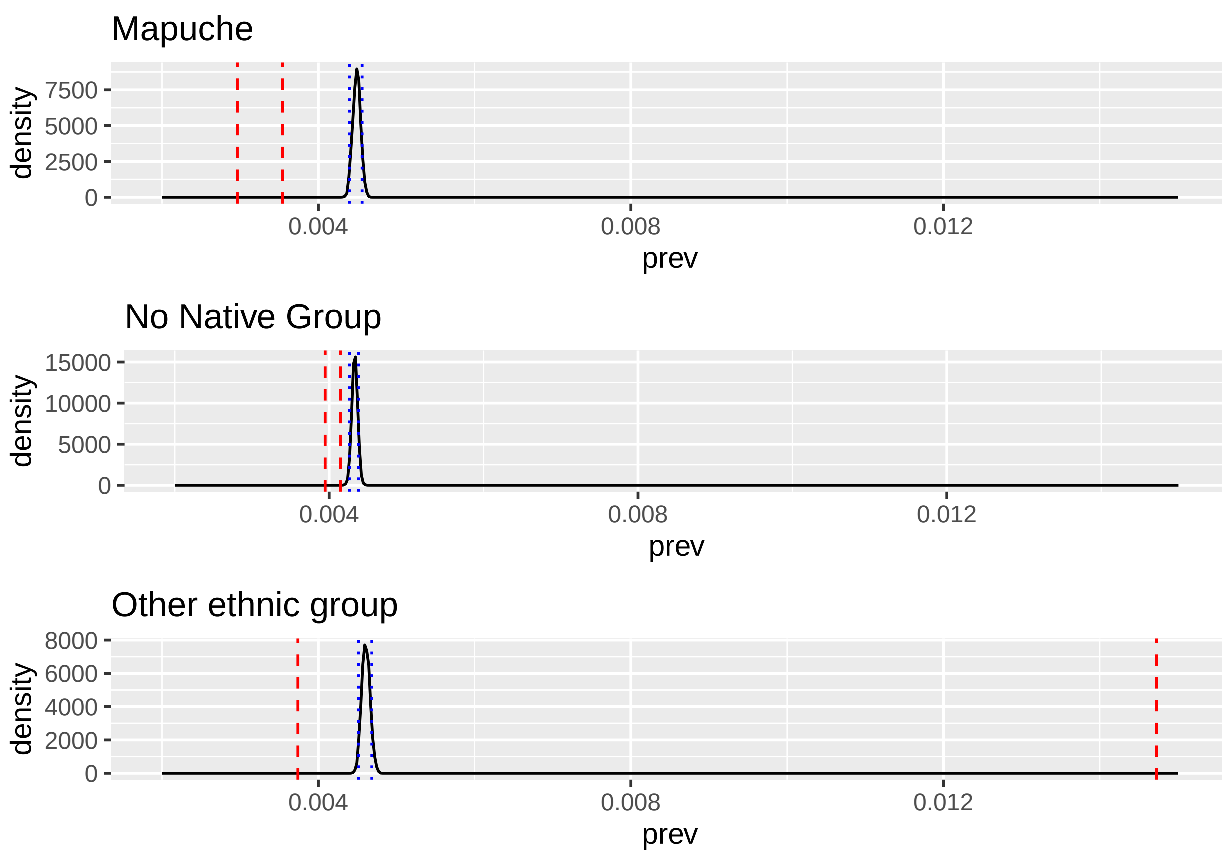
```
## # A tibble: 11 x 10
##    varia~1    mean  median      sd     mad      q5     q95  rhat ess_b~2 ess_t~3
##    <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
##  1 aut_pr~ 6.92e+2 6.92e+2 2.75e+1 2.67e+1 6.48e+2 7.39e+2  1.00   1881.   1931.
##  2 aut_pr~ 7.25e+3 7.25e+3 1.04e+2 1.02e+2 7.09e+3 7.42e+3  1.00   1646.   1703.
##  3 aut_pr~ 4.15e+1 4.1 e+1 6.34e+0 5.93e+0 3.1 e+1 5.2 e+1  1.00   1890.   1729.
##  4 aut_sa~ 4.99e+2 4.99e+2 0       0       4.99e+2 4.99e+2 NA        NA      NA
##  5 aut_sa~ 6.77e+3 6.77e+3 0       0       6.77e+3 6.77e+3 NA        NA      NA
##  6 aut_sa~ 5.9 e+1 5.9 e+1 0       0       5.9 e+1 5.9 e+1 NA        NA      NA
##  7 theta[~ 4.49e-3 4.49e-3 4.81e-5 4.87e-5 4.41e-3 4.57e-3  1.00   1232.   1178.
##  8 theta[~ 4.33e-3 4.33e-3 3.46e-5 3.37e-5 4.28e-3 4.39e-3  1.00   1201.   1099.
##  9 theta[~ 4.61e-3 4.61e-3 5.19e-5 5.46e-5 4.53e-3 4.70e-3  1.00   1306.   1130.
## 10 theta_a 8.09e+3 8.09e+3 0       0       8.09e+3 8.09e+3 NA        NA      NA
## 11 theta_b 1.75e+6 1.75e+6 0       0       1.75e+6 1.75e+6 NA        NA      NA
## # ... with abbreviated variable names 1: variable, 2: ess_bulk, 3: ess_tail
```

```r
rand_ethnic_summ <- summary(subset_draws(as_draws(rand_ethnic_sam), rand_ethnic_pars),
                           ~quantile(.x, probs=c(0.025, 0.5, 0.975)),
                           ~mcse_quantile(.x, probs=c(0.025, 0.5, 0.975)),
                           "rhat") %>%
  arrange(desc(mcse_q50))
rand_ethnic_summ
```

```
## # A tibble: 11 x 8
##    variable             `2.5%`    `50%` `97.5%` mcse_q~1 mcse_q50 mcse_q~2  rhat
```

```
##    <chr>                  <dbl>    <dbl>   <dbl>    <dbl>    <dbl>    <dbl> <dbl>
##  1 aut_pred[2]      7049.        7.25e+3 7.46e+3  8    e+0  3    e+0  5.5 e+0  1.00
##  2 aut_pred[1]       640         6.92e+2 7.49e+2  2.5 e+0  1    e+0  1.5 e+0  1.00
##  3 theta[1]            0.00440   4.49e-3 4.58e-3  3.22e-6  1.88e-6  3.90e-6  1.00
##  4 theta[3]            0.00451   4.61e-3 4.72e-3  3.10e-6  1.79e-6  4.36e-6  1.00
##  5 theta[2]            0.00426   4.33e-3 4.40e-3  3.65e-6  8.99e-7  2.52e-6  1.00
##  6 aut_pred[3]        29         4.1 e+1 5.4 e+1  5    e-1  0         5    e-1  1.00
##  7 theta_a          8091.        8.09e+3 8.09e+3 NA       NA       NA       NA
##  8 theta_b       1750915.        1.75e+6 1.75e+6 NA       NA       NA       NA
##  9 aut_sample[1]     499         4.99e+2 4.99e+2 NA       NA       NA       NA
## 10 aut_sample[2]    6772         6.77e+3 6.77e+3 NA       NA       NA       NA
## 11 aut_sample[3]      59         5.9 e+1 5.9 e+1 NA       NA       NA       NA
## # ... with abbreviated variable names 1: mcse_q2.5, 2: mcse_q97.5
```

```
aut_prev_ethnic_plots <- list()
ethnic_post_ci_lower <- list()
ethnic_post_ci_upper <- list()

for(i in 1:nEthnic) {
  prevs <- data.frame(prev = extract_variable(rand_ethnic_sam, paste0("theta[", i, "]")))
  ethnic_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
  ethnic_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
  density_plot <- ggplot(prevs, aes(x = prev)) +
    geom_density() +
    xlim(c(0.002, 0.015)) +
    geom_vline(xintercept = ethnic_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = ethnic_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = aut_prev_ethnic_adj$ci_lower[i], color = "red", linetype = "dashed") +
    geom_vline(xintercept = aut_prev_ethnic_adj$ci_upper[i], color = "red", linetype = "dashed") +
    #labs(title = aut_prev_ethnic_adj$ethnic_3_group[i])
    labs(title = aut_prev_ethnic_adj$ethnic_2_group[i])
  aut_prev_ethnic_plots[[i]] <- density_plot
}
do.call(grid.arrange, aut_prev_ethnic_plots)
```

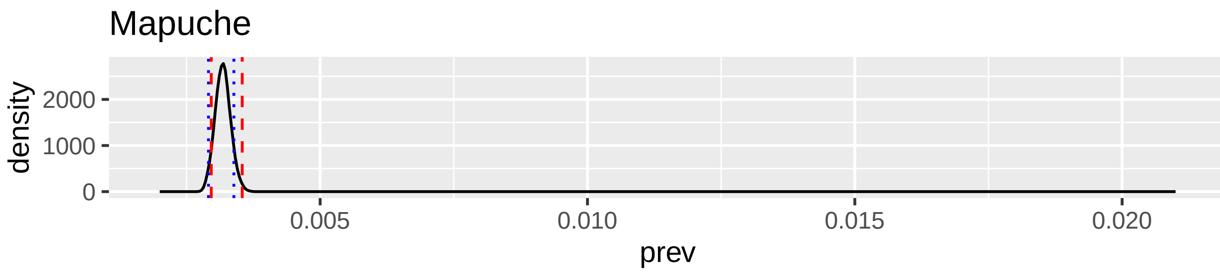


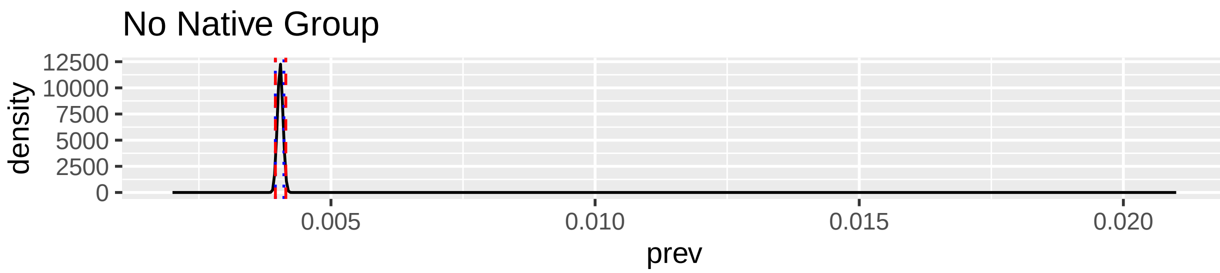

```
#autism_prev_ethnic_plots <- do.call(grid.arrange, aut_prev_ethnic_plots)
#ggsave("autism_prev_ethnicity_plots.png", autism_prev_ethnic_plots, height = 10, width = 15)
```

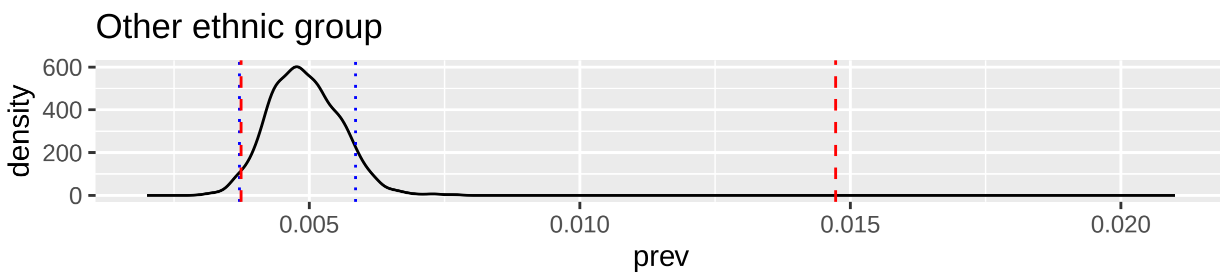
**Sensitivity analysis - alter prior mean and sd**

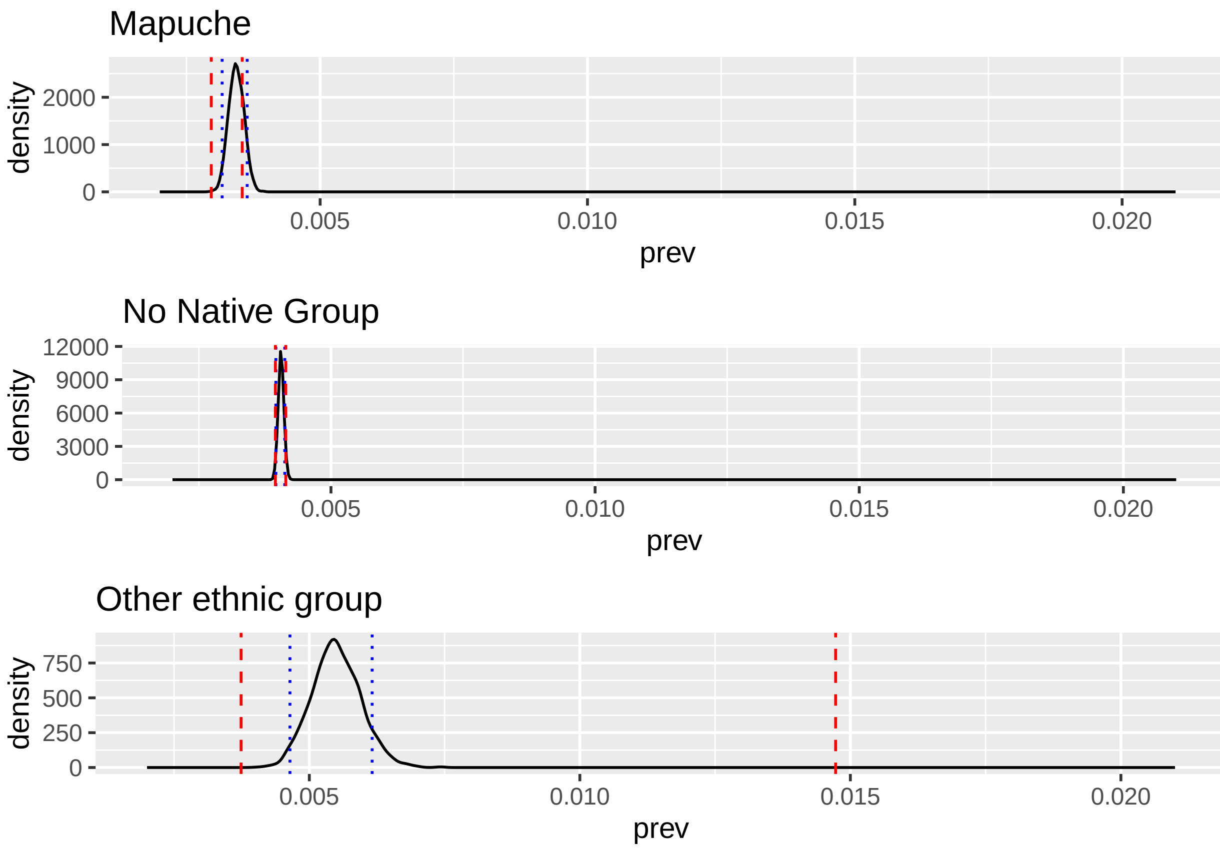
```
theta_mu <- c(0.001, 0.005, 0.01, 0.02, # 0.1%, 0.5%, 1%, 2% prevalence
              rep(0.0046, 4)) # Same as chosen prior
theta_sigma <- c(rep(0.001/1.96, 4), # Same as chosen prior
                 0.0001, 0.001, 0.05, 0.01) # +/- 0.1%, 0.5%, 1%, 5%
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

for(j in 1:length(theta_mu)) {
  rand_ethnic_data <- list(theta_a = theta_a[j],
                           theta_b = theta_b[j],
                           nObs = aut_prev_ethnic_adj$sum_sample_pop_size,
                           aut_sample = aut_prev_ethnic_adj$adjusted_count,
                           nEthnic = nEthnic)
  rand_ethnic_jag <- jags.model(textConnection(rand_ethnic_model),
                                data = rand_ethnic_data,
                                #inits = rand_region_ini,
                                n.chains = 2,
                                quiet = TRUE)
  update(rand_ethnic_jag, n.iter = nBurn)
  rand_ethnic_sam <- coda.samples(model = rand_ethnic_jag,
                                  variable.names = rand_ethnic_pars,
```

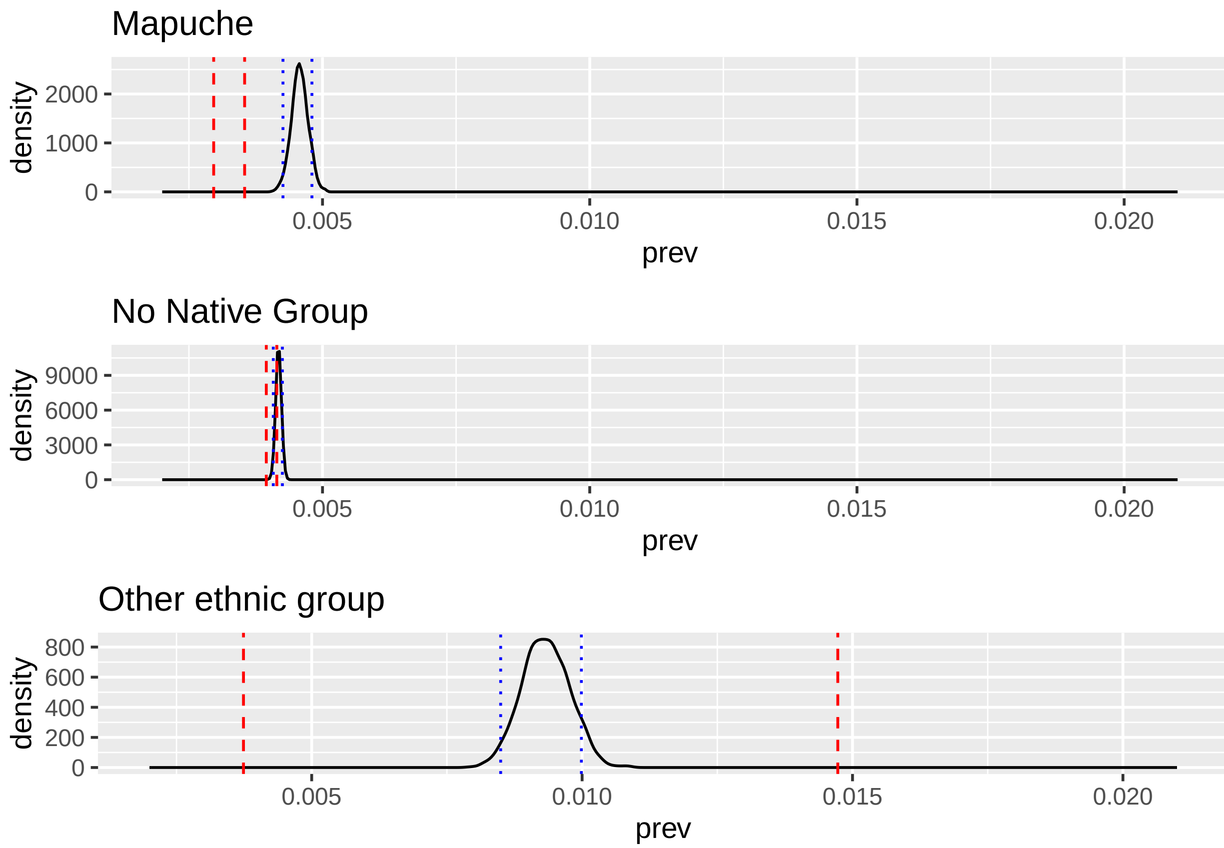
```r
                                  n.iter = nIter)

  # Plots
  aut_prev_ethnic_plots <- list()
  ethnic_post_ci_lower <- list()
  ethnic_post_ci_upper <- list()

  for(i in 1:nEthnic) {
    prevs <- data.frame(prev = extract_variable(rand_ethnic_sam, paste0("theta[", i, "]")))
    ethnic_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
    ethnic_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
    density_plot <- ggplot(prevs, aes(x = prev)) +
      geom_density() +
      xlim(c(0.002, 0.021)) +
      geom_vline(xintercept = ethnic_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = ethnic_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = aut_prev_ethnic_adj$ci_lower[i], color = "red", linetype = "dashed") +
      geom_vline(xintercept = aut_prev_ethnic_adj$ci_upper[i], color = "red", linetype = "dashed") +
      #labs(title = aut_prev_ethnic_adj$ethnic_3_group[i])
      labs(title = aut_prev_ethnic_adj$ethnic_2_group[i])
    aut_prev_ethnic_plots[[i]] <- density_plot
  }
  do.call(grid.arrange, aut_prev_ethnic_plots)
  #autism_prev_ethnic_plots <- do.call(grid.arrange, aut_prev_ethnic_plots)
  #ggsave(paste0("autism_prev_ethnicity_plots_", j, ".png"), autism_prev_ethnic_plots, height = 10, wid

}
```

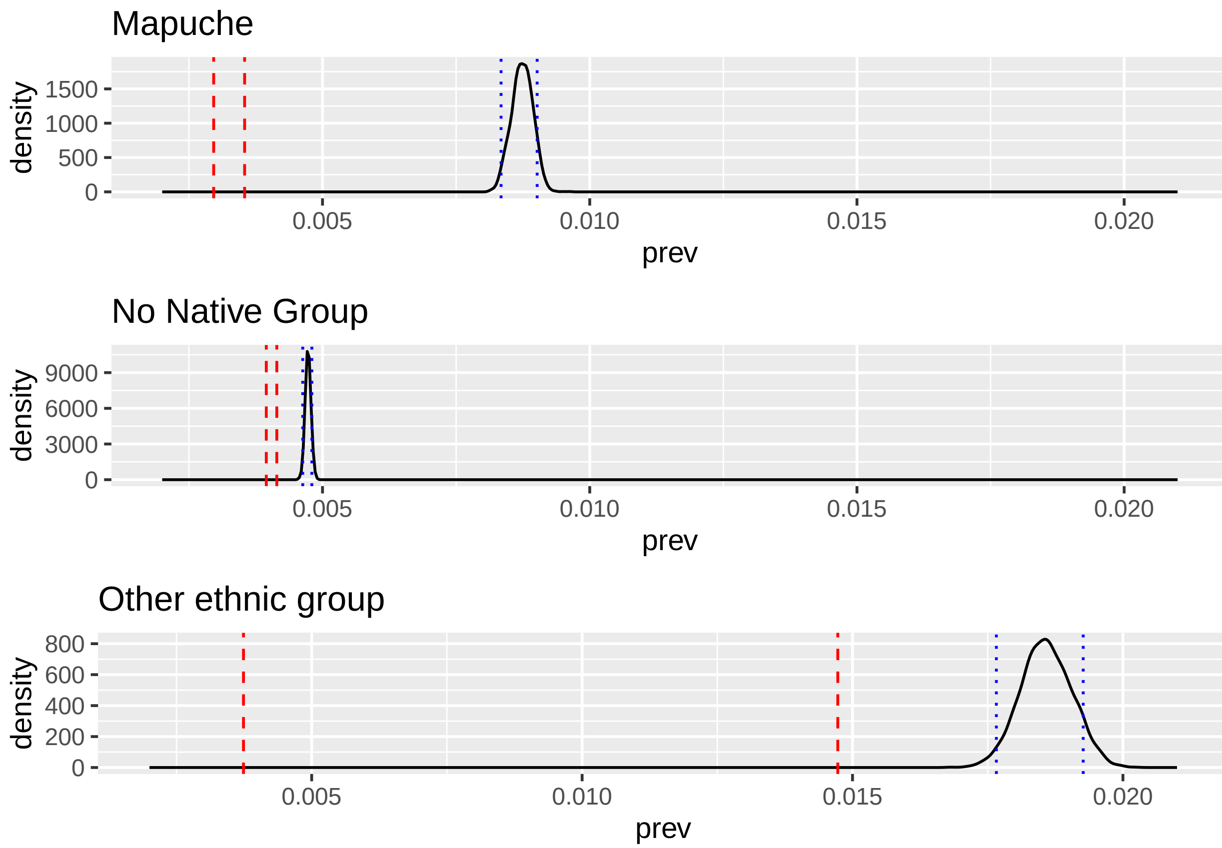

Mapuche
density
2000
1000
0
0.005
0.010
0.015
0.020
prev


No Native Group
density
12500
10000
7500
5000
2500
0
0.005
0.010
0.015
0.020
prev

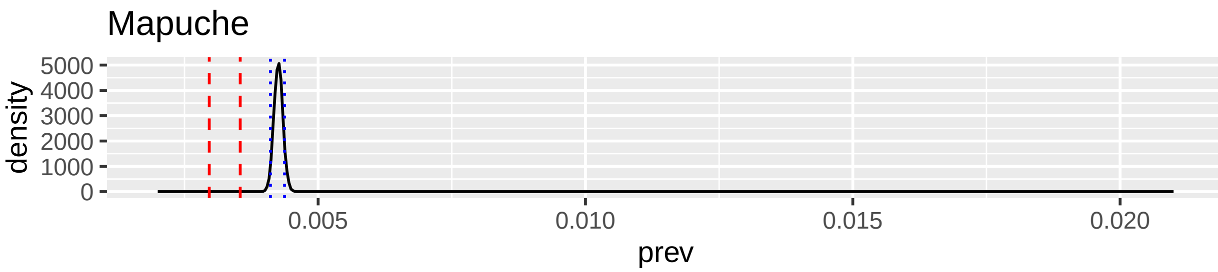

Other ethnic group
density
600
400
200
0
0.005
0.010
0.015
0.020
prev

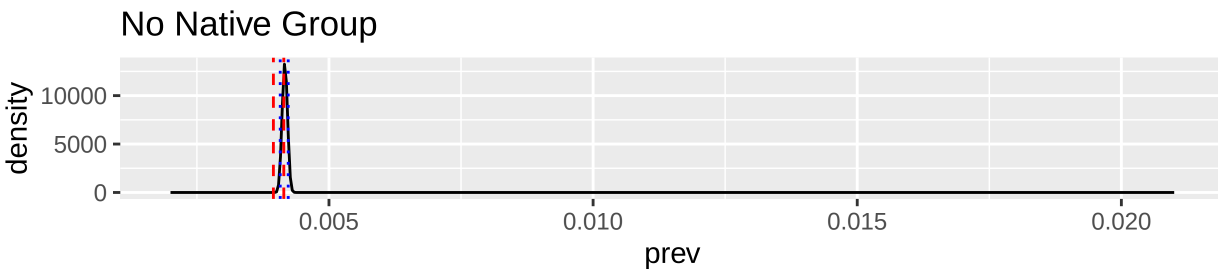

Mapuche
density
2000
1000
0
0.005
0.010
0.015
0.020
prev
No Native Group
density
12000
9000
6000
3000
0
0.005
0.010
0.015
0.020
prev
Other ethnic group
density
750
500
250
0
0.005
0.010
0.015
0.020
prev

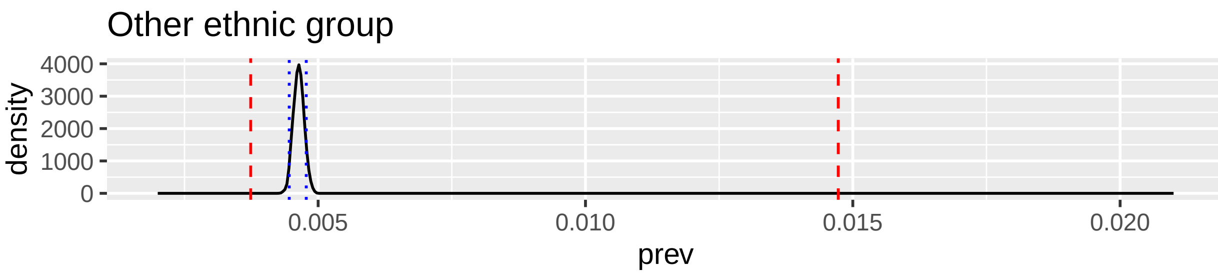

Mapuche
density
2000
1000
0
0.005
0.010
0.015
0.020
prev
No Native Group
density
9000
6000
3000
0
0.005
0.010
0.015
0.020
prev
Other ethnic group
density
800
600
400
200
0
0.005
0.010
0.015
0.020
prev

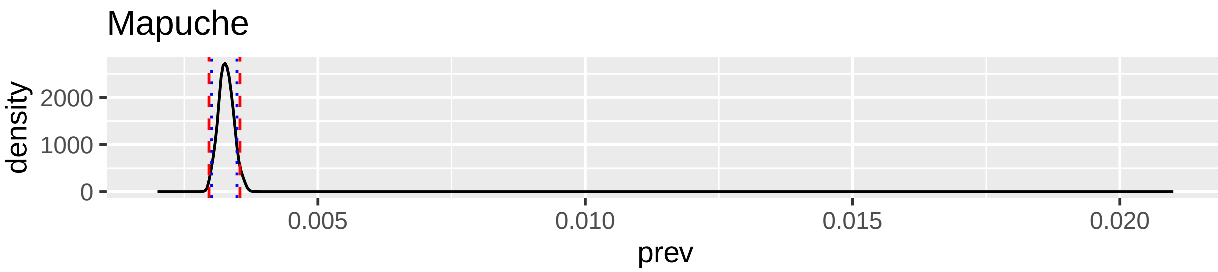


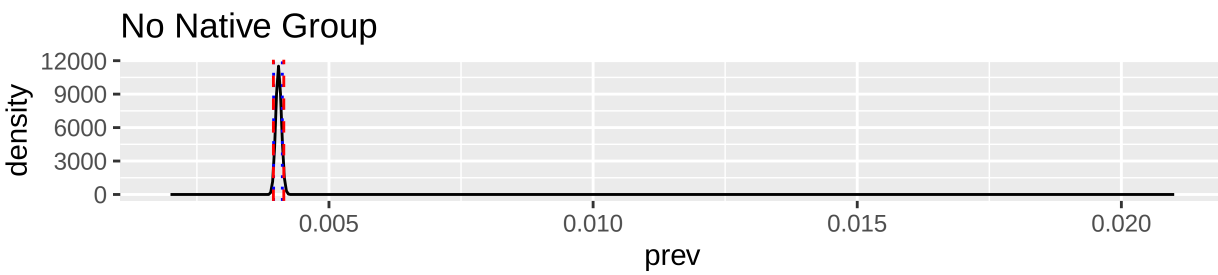
Mapuche
density
1500
1000
500
0
0.005
0.010
0.015
0.020
prev
No Native Group
density
9000
6000
3000
0
0.005
0.010
0.015
0.020
prev
Other ethnic group
density
800
600
400
200
0
0.005
0.010
0.015
0.020
prev

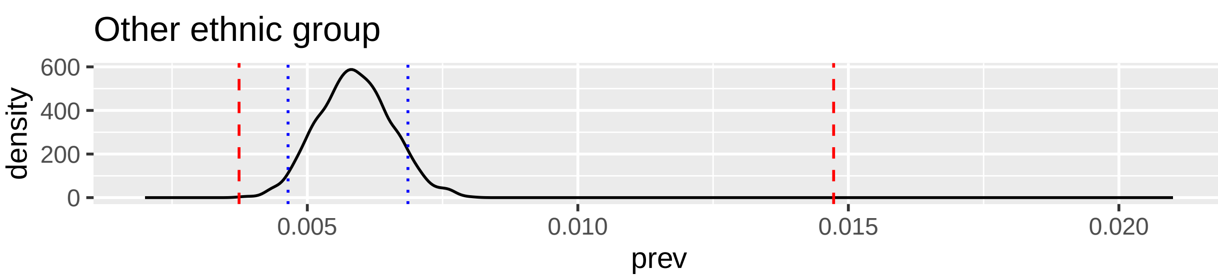

Mapuche
density
5000
4000
3000
2000
1000
0
0.005
0.010
0.015
0.020
prev

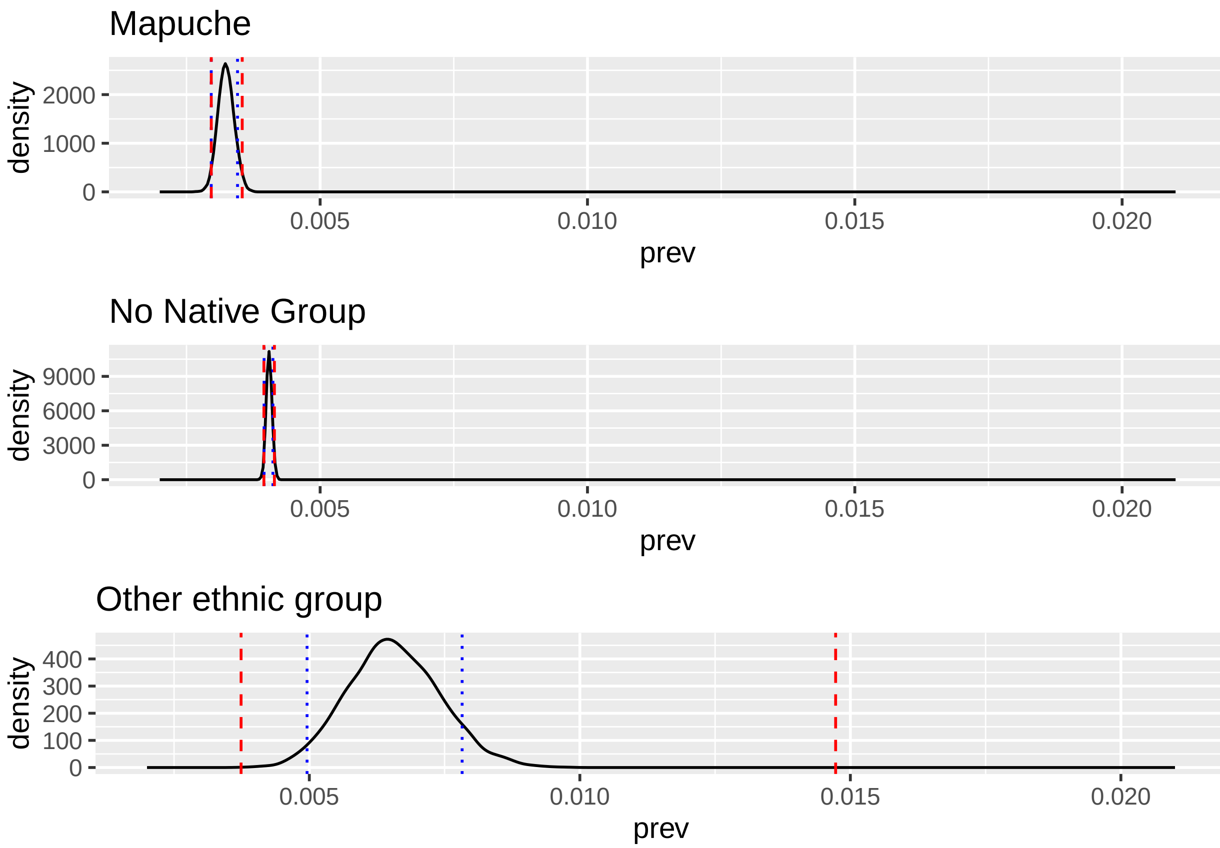

No Native Group
density
10000
5000
0
0.005
0.010
0.015
0.020
prev

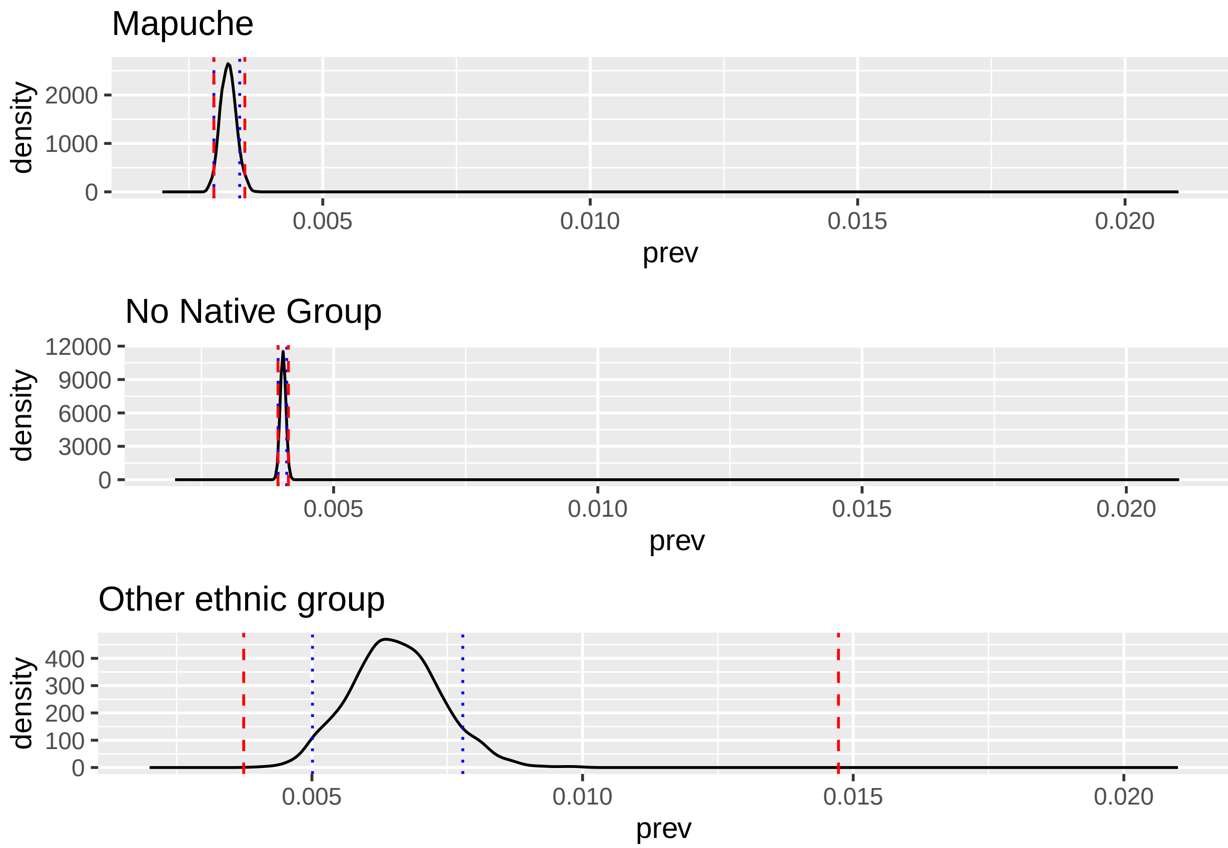

Other ethnic group
density
4000
3000
2000
1000
0
0.005
0.010
0.015
0.020
prev


Mapuche
density
2000
1000
0
0.005
0.010
0.015
0.020
prev


No Native Group
density
12000
9000
6000
3000
0
0.005
0.010
0.015
0.020
prev


Other ethnic group
density
600
400
200
0
0.005
0.010
0.015
0.020
prev


Mapuche
density
2000
1000
0
0.005
0.010
0.015
0.020
prev
No Native Group
density
9000
6000
3000
0
0.005
0.010
0.015
0.020
prev
Other ethnic group
density
400
300
200
100
0
0.005
0.010
0.015
0.020
prev




### Bayesian prevalence by economic status

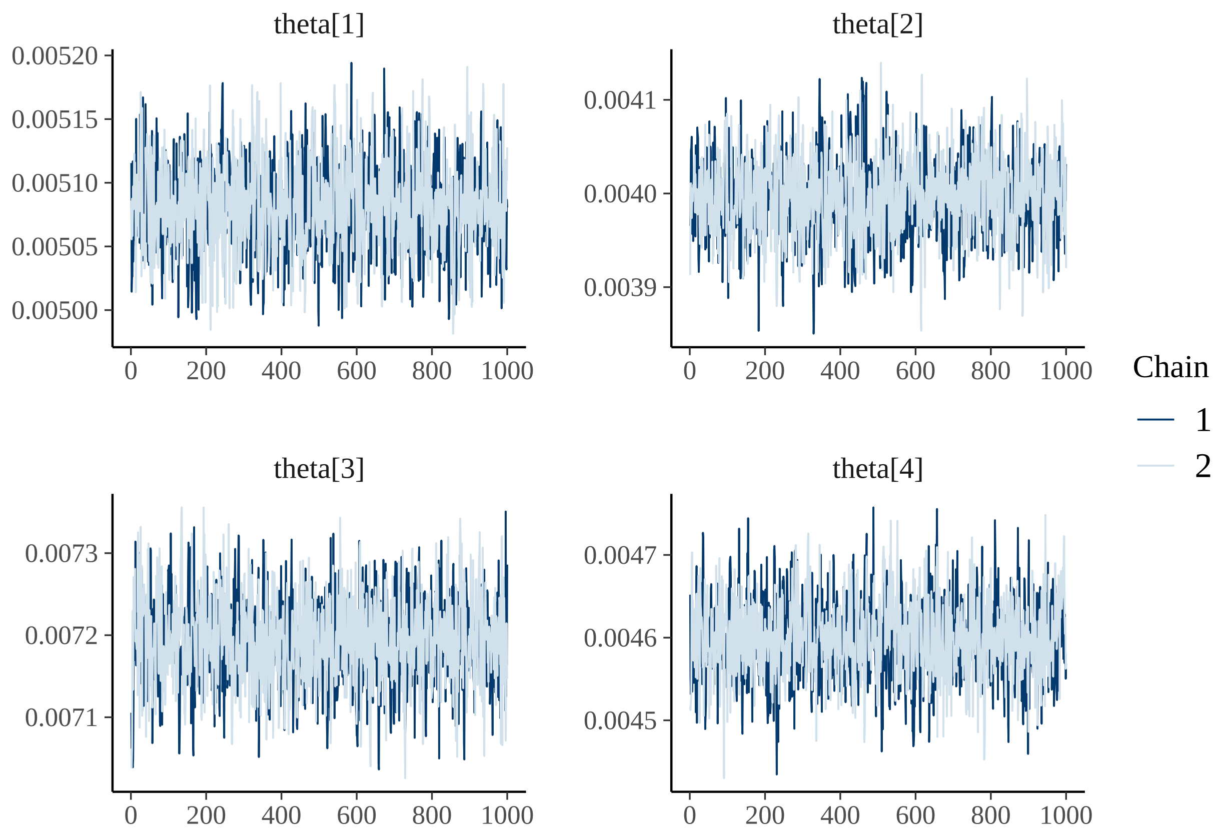
```
aut_prev_econ <- chile_bayes_aut %>%
  mutate(school_fee = ifelse(school_fee == "", "SIN INFORMACION", school_fee),
         school_fee_group = ifelse(school_fee == "GRATUITO", "Free",
                                   ifelse(school_fee %in% c("$1.000 A $10.000", "$10.001 A $25.000", "$25
                                          ifelse(school_fee == "MAS DE $100.000", "High", "No informatio
  group_by(school_fee, school_fee_group, age_june30, sex, autism) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = autism, values_from = count) %>%
  rename("n_noautism" = "0", "n_autism" = "1", "age" = "age_june30") %>%
  mutate(n_autism = ifelse(is.na(n_autism), 0, n_autism),
         sample_pop_size = n_noautism + n_autism,
         sample_prevalence = n_autism / sample_pop_size) %>%
  left_join(chile_stdpop, by = c("age", "sex")) %>%
  mutate(aut_prev_std = n_autism / sample_pop_size * pop_prop,
         w = std_pop / (sample_pop_size * n_std_pop),
         w2 = pop_prop / sample_pop_size,
         sum_std_pop = sum(std_pop)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'school_fee', 'school_fee_group',
## 'age_june30', 'sex'. You can override using the `.groups` argument.
```

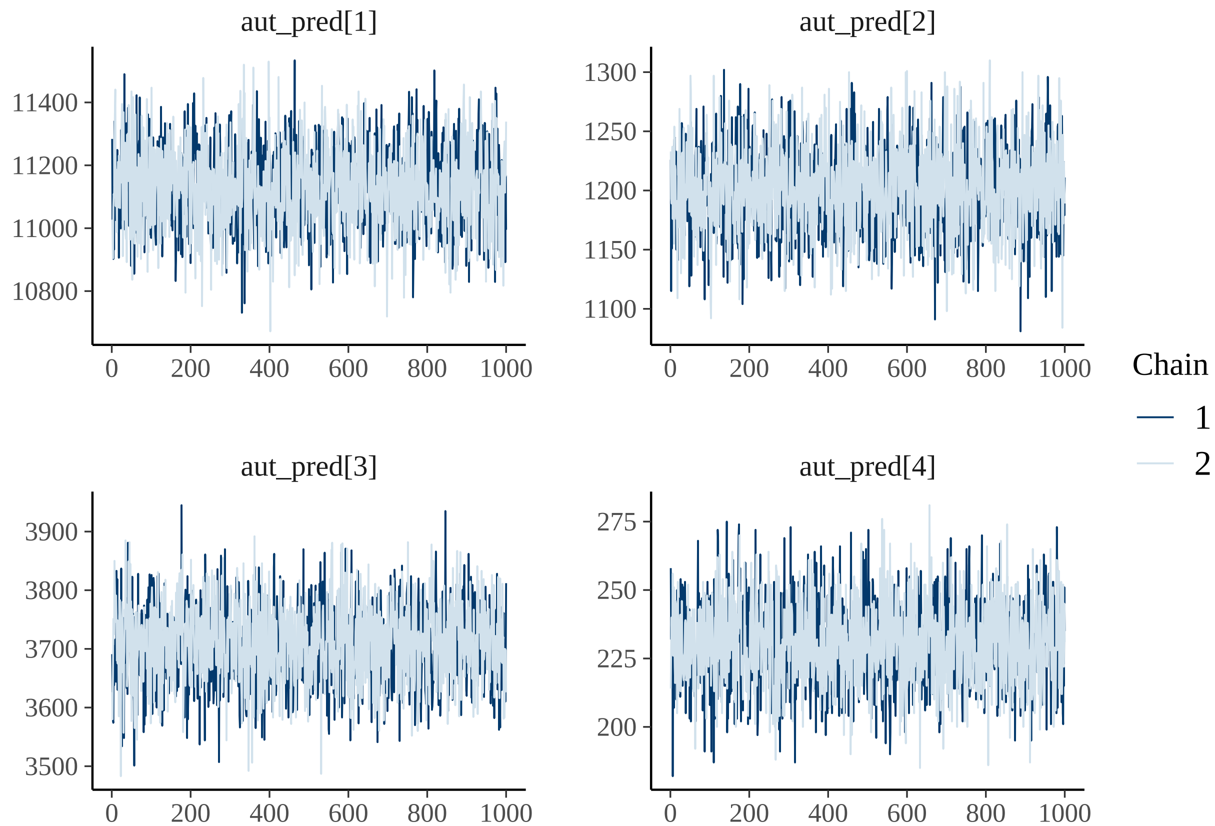
```
aut_prev_econ_adj <- aut_prev_econ %>%
  #group_by(school_fee) %>%
```

```
  group_by(school_fee_group) %>%
  summarise(sum_sample_pop_size = sum(sample_pop_size),
            crude_rate = sum(n_autism) / sum(sample_pop_size),
            crude_count = sum(n_autism),
            adjusted_rate = sum(n_autism / sample_pop_size * pop_prop),
            adjusted_count = round(adjusted_rate * sum_sample_pop_size, 0), # had to fudge this to get 
            #adjusted_count = adjusted_rate * sum_sample_pop_size,
            var = sum(pop_prop^2 * n_autism / sample_pop_size^2),
            #se2 = sqrt(sum((std_pop/sum(std_pop))^2 * n_autism/sample_pop_size^2)),
            w_M = max(w),
            ci_lower = var / (2*adjusted_rate) * qchisq(p = 0.05/2, df = 2*adjusted_rate^2 / var),
            ci_upper = (var + w_M^2) / (2*(adjusted_rate + w_M)) * qchisq(p = 1-0.05/2, df = 2*(adjuste
  #arrange(school_fee)
  arrange(school_fee_group)

# Try informative prior
theta_mu <- 0.0046
theta_sigma <- (0.0047-0.0045) / (2*1.96)
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

nEcon <- length(unique(aut_prev_econ$school_fee))
nEcon <- length(unique(aut_prev_econ$school_fee_group))

rand_econ_model <- "model {
  for(i in 1:nEcon) { # For each economic status level
    theta[i] ~ dbeta(theta_a, theta_b)
    aut_sample[i] ~ dbin(theta[i], nObs[i])

    aut_pred[i] ~ dbin(theta[i], nObs[i])
  }
}"

rand_econ_data <- list(theta_a = theta_a,
                       theta_b = theta_b,
                       nObs = aut_prev_econ_adj$sum_sample_pop_size,
                       aut_sample = aut_prev_econ_adj$adjusted_count,
                       nEcon = nEcon)

rand_econ_ini <- list(list(theta = rep(0.001, nEcon)), #, spec = 0.5, sens = 0.5),
                      list(theta = rep(0.01, nEcon))) #, spec = 0.9, sens = 0.9))

rand_econ_pars <- c("theta_a", "theta_b", "theta", "aut_sample", "aut_pred")

# Run JAGS model and discard burn-in samples
rand_econ_jag <- jags.model(textConnection(rand_econ_model),
                            data = rand_econ_data,
                            inits = rand_econ_ini,
                            n.chains = 2,
                            quiet = TRUE)
update(rand_econ_jag, n.iter = nBurn)
rand_econ_sam <- coda.samples(model = rand_econ_jag,
                              variable.names = rand_econ_pars,
```

```
                                 n.iter = nIter)

# Check for convergence in parameters of interest
#mcmc_trace(rand_region_sam, rand_region_pars)
mcmc_trace(rand_econ_sam, paste0("theta[", 1:nEcon, "]")) # Convergence looks fine and rhats <= 1.1
```




```
mcmc_trace(rand_econ_sam, paste0("aut_pred[", 1:nEcon, "]"))# Convergence looks fine and rhats <= 1.1
```



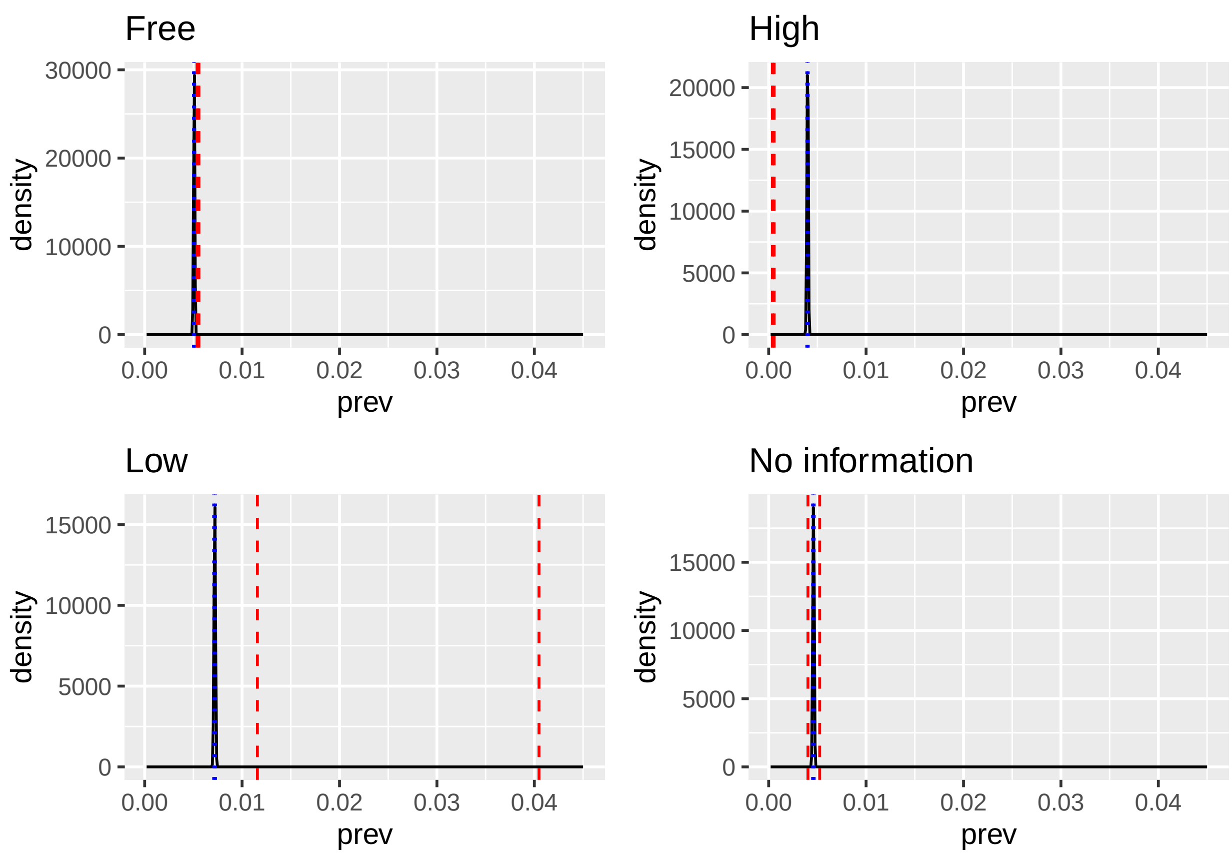

```r
summary(as_draws(rand_econ_sam)) %>% print(n = Inf)
```

```
## # A tibble: 14 x 10
##    varia~1    mean  median      sd     mad      q5     q95  rhat ess_b~2 ess_t~3
##    <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
##  1 aut_pr~ 1.11e+4 1.11e+4 1.31e+2 1.33e+2 1.09e+4 1.13e+4  1.00   1758.   1762.
##  2 aut_pr~ 1.20e+3 1.20e+3 3.67e+1 3.71e+1 1.14e+3 1.26e+3  1.00   1870.   1915.
##  3 aut_pr~ 3.71e+3 3.70e+3 6.71e+1 6.82e+1 3.60e+3 3.82e+3  1.00   1564.   1879.
##  4 aut_pr~ 2.30e+2 2.29e+2 1.55e+1 1.63e+1 2.05e+2 2.55e+2  1.00   1935.   1983.
##  5 aut_sa~ 1.20e+4 1.20e+4 0       0       1.20e+4 1.20e+4 NA        NA      NA
##  6 aut_sa~ 1.39e+2 1.39e+2 0       0       1.39e+2 1.39e+2 NA        NA      NA
##  7 aut_sa~ 8.27e+3 8.27e+3 0       0       8.27e+3 8.27e+3 NA        NA      NA
##  8 aut_sa~ 2.29e+2 2.29e+2 0       0       2.29e+2 2.29e+2 NA        NA      NA
##  9 theta[~ 5.08e-3 5.08e-3 3.65e-5 3.63e-5 5.02e-3 5.14e-3  1.00   1256.   1086.
## 10 theta[~ 4.00e-3 4.00e-3 4.30e-5 4.15e-5 3.93e-3 4.07e-3  1.00   1196.   1185.
## 11 theta[~ 7.19e-3 7.19e-3 5.51e-5 5.51e-5 7.10e-3 7.28e-3  1.00   1187.   1152.
## 12 theta[~ 4.60e-3 4.60e-3 4.86e-5 4.71e-5 4.52e-3 4.68e-3  1.00   1213.   1193.
## 13 theta_a 8.09e+3 8.09e+3 0       0       8.09e+3 8.09e+3 NA        NA      NA
## 14 theta_b 1.75e+6 1.75e+6 0       0       1.75e+6 1.75e+6 NA        NA      NA
## # ... with abbreviated variable names 1: variable, 2: ess_bulk, 3: ess_tail
```

```r
rand_econ_summ <- summary(subset_draws(as_draws(rand_econ_sam), rand_econ_pars),
                          ~quantile(.x, probs=c(0.025, 0.5, 0.975)),
                          ~mcse_quantile(.x, probs=c(0.025, 0.5, 0.975)),
                          "rhat") %>%
  arrange(desc(mcse_q50))
rand_econ_summ
```

```
## # A tibble: 14 x 8
##    variable             `2.5%`    `50%` `97.5%` mcse_q~1 mcse_q50 mcse_q~2  rhat
##    <chr>                 <dbl>    <dbl>   <dbl>    <dbl>    <dbl>    <dbl> <dbl>
##  1 aut_pred[1]     10882.       1.11e+4 1.14e+4  6.5 e+0  3.5 e+0  9.5 e+0  1.00
##  2 aut_pred[3]      3578.       3.70e+3 3.84e+3  3.5 e+0  2   e+0  5   e+0  1.00
##  3 aut_pred[2]      1128        1.20e+3 1.27e+3  2.5 e+0  1   e+0  3   e+0  1.00
##  4 aut_pred[4]       201        2.29e+2 2.61e+2  1   e+0  5   e-1  1.5 e+0  1.00
##  5 theta[4]            0.00450  4.60e-3 4.70e-3  3.84e-6  1.75e-6  3.43e-6  1.00
##  6 theta[3]            0.00708  7.19e-3 7.30e-3  6.69e-6  1.66e-6  4.98e-6  1.00
##  7 theta[1]            0.00501  5.08e-3 5.15e-3  2.15e-6  1.57e-6  1.73e-6  1.00
##  8 theta[2]            0.00391  4.00e-3 4.08e-3  3.69e-6  1.11e-6  4.16e-6  1.00
##  9 theta_a          8091.       8.09e+3 8.09e+3 NA       NA       NA       NA
## 10 theta_b       1750915.       1.75e+6 1.75e+6 NA       NA       NA       NA
## 11 aut_sample[1]   11980        1.20e+4 1.20e+4 NA       NA       NA       NA
## 12 aut_sample[2]     139        1.39e+2 1.39e+2 NA       NA       NA       NA
## 13 aut_sample[3]    8267        8.27e+3 8.27e+3 NA       NA       NA       NA
## 14 aut_sample[4]     229        2.29e+2 2.29e+2 NA       NA       NA       NA
## # ... with abbreviated variable names 1: mcse_q2.5, 2: mcse_q97.5
```

```
aut_prev_econ_plots <- list()
econ_post_ci_lower <- list()
econ_post_ci_upper <- list()

for(i in 1:nEcon) {
  prevs <- data.frame(prev = extract_variable(rand_econ_sam, paste0("theta[", i, "]")))
  econ_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
  econ_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
  density_plot <- ggplot(prevs, aes(x = prev)) +
    geom_density() +
    xlim(c(0.0002, 0.045)) +
    geom_vline(xintercept = econ_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = econ_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
    geom_vline(xintercept = aut_prev_econ_adj$ci_lower[i], color = "red", linetype = "dashed") +
    geom_vline(xintercept = aut_prev_econ_adj$ci_upper[i], color = "red", linetype = "dashed") +
    #labs(title = aut_prev_econ_adj$school_fee[i])
    labs(title = aut_prev_econ_adj$school_fee_group[i])
  aut_prev_econ_plots[[i]] <- density_plot
}
do.call(grid.arrange, aut_prev_econ_plots)
```




```
#autism_prev_econ_plots <- do.call(grid.arrange, aut_prev_econ_plots)
#ggsave("autism_prev_econ_plots.png", autism_prev_econ_plots, height = 10, width = 15)
```

**Sensitivity analysis - alter prior mean and sd**

```
theta_mu <- c(0.001, 0.005, 0.01, 0.02, # 0.1%, 0.5%, 1%, 2% prevalence
              rep(0.0046, 4)) # Same as chosen prior
theta_sigma <- c(rep(0.001/1.96, 4), # Same as chosen prior
                 0.0001, 0.001, 0.05, 0.01) # +/- 0.1%, 0.5%, 1%, 5%
theta_a <- theta_mu * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)
theta_b <- (1 - theta_mu) * (theta_mu * (1-theta_mu) / theta_sigma^2 - 1)

for(j in 1:length(theta_mu)) {
  #print(j)
  #print(theta_a[j])
  #print(theta_b[j])
  rand_econ_data <- list(theta_a = theta_a[j],
                         theta_b = theta_b[j],
                         nObs = aut_prev_econ_adj$sum_sample_pop_size,
                         aut_sample = aut_prev_econ_adj$adjusted_count,
                         nEcon = nEcon)
  rand_econ_jag <- jags.model(textConnection(rand_econ_model),
                              data = rand_econ_data,
                              inits = rand_econ_ini,
                              n.chains = 2,
                              quiet = TRUE)
```

```r
  update(rand_econ_jag, n.iter = nBurn)
  rand_econ_sam <- coda.samples(model = rand_econ_jag,
                                variable.names = rand_econ_pars,
                                n.iter = nIter)
  mcmc_trace(rand_econ_sam, paste0("theta[", 1:nEcon, "]")) # Convergence looks fine and rhats <= 1.1
  mcmc_trace(rand_econ_sam, paste0("aut_pred[", 1:nEcon, "]"))# Convergence looks fine and rhats <= 1.1

  # Plot
  aut_prev_econ_plots <- list()
  econ_post_ci_lower <- list()
  econ_post_ci_upper <- list()

  for(i in 1:nEcon) {
    prevs <- data.frame(prev = extract_variable(rand_econ_sam, paste0("theta[", i, "]")))
    econ_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
    econ_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
    density_plot <- ggplot(prevs, aes(x = prev)) +
      geom_density() +
      xlim(c(0.0002, 0.05)) +
      geom_vline(xintercept = econ_post_ci_lower[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = econ_post_ci_upper[[i]], color = "blue", linetype = "dotted") +
      geom_vline(xintercept = aut_prev_econ_adj$ci_lower[i], color = "red", linetype = "dashed") +
      geom_vline(xintercept = aut_prev_econ_adj$ci_upper[i], color = "red", linetype = "dashed") +
      #labs(title = aut_prev_econ_adj$school_fee[i])
      labs(title = aut_prev_econ_adj$school_fee_group[i])
    aut_prev_econ_plots[[i]] <- density_plot
  }
  #autism_prev_econ_plots <- do.call(grid.arrange, aut_prev_econ_plots)
  #ggsave(paste0("autism_prev_econ_plots_", j, ".png"), autism_prev_econ_plots, height = 10, width = 15
}
```