

Chile_prev_rmd

Adele Tyson

2023-06-07

```
#source("Chile_prev.R", local = knitr::knit_global())

library(nleqslv) # Only needed for robince bayesian prevalence
library(janitor)
library(gridExtra)
library(readxl)
library(psych)
library(Hmisc)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
library(poolr)
```

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

```
library(epitools)
library(corrplot)
library(caret)
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)
library(rstan)
library(posterior)
library(tidybayes)
library(bayesplot)
library(tidyverse)
```

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
         ethnic_2_group = ifelse(ethnic_3_group == "Aymara", "Other ethnic group", ethnic_3_group),
         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", "$25.001 A $100.000",
                                   ifelse(school_fee == "MAS DE $100.000", "High", "No information"))),

  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,
         school_fee_group,
         ethnicity,
         mapuche,
         nationality,
         ethnic_3_group,
         ethnic_2_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)

```

Define some functions to keep code clean

```

get_grouped_prev <- function(x, stdpop, grouping_vars) {
  # Calculates sample prevalence, age- and sex-standardised prevalence and group weighting for supplied
  # x = chile_bayes_aut
  # stdpop = standard population with age and sex counts
  # grouping_vars = variables in x to group by
  x_grouped <- x %>%
    group_by(across(all_of(grouping_vars))) %>%
    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), # If there are no cases of autism in the gr
      sample_pop_size = n_noautism + n_autism, # Total sample population is autism cases + not cas
      sample_prevalence = n_autism / sample_pop_size) %>% # Prevalence of autism in the group
left_join(stdpop, by = c("age", "sex")) %>%
mutate(aut_prev_std = n_autism / sample_pop_size * pop_prop, # Prevalence of autism in the group, s
      w = std_pop / (sample_pop_size * n_std_pop), # Weight of the group using standard population
      #w2 = pop_prop / sample_pop_size,
      #sum_std_pop = sum(std_pop)
      ) %>%
ungroup()
return(x_grouped)
}

get_adjusted_prev <- function(x, grouping_vars) {
  # Turns grouped prevalences into age- and sex- adjusted prevalences with Fay and Feuer Gamma confiden
  # x = output from get_grouped_prev
  x_adj <- x %>%
  group_by(across(all_of(grouping_vars))) %>%
  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 i
            #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*(adjusted
  arrange(across(all_of(grouping_vars)))
}

```

Set global parameters

```

nObs <- nrow(chile_bayes_aut)
nIter <- 1000
nBurn <- 1000
pars <- c("theta_a", "theta_b", "theta", "aut_sample", "aut_pred")

theta_mu_prior <- 0.0046
theta_sigma_prior <- (0.0047-0.0045) / (2*1.96)
theta_mu_sens <- 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_sens <- 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_mu <- c(theta_mu_prior, theta_mu_sens)
theta_sigma <- c(theta_sigma_prior, theta_sigma_sens)
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)

```

Common effects model with sample prevalence

```

# Uniform prior
theta_a_common <- 1

```

```

theta_b_common <- 1
# This corresponds to a mean of 0.5

# OR Informative prior (global population prevalence)
# Say autism has mean prevalence of 3% and we are 95% confidence that the prevalence is between 2% and 4%
# Then mu = 0.03, sigma = (0.04-0.02) / (2*1.96)
theta_mu_common <- 0.03
theta_sigma_common <- (0.04-0.02) / (2*1.96)
theta_a_common <- theta_mu_common * (theta_mu_common * (1-theta_mu_common) / theta_sigma_common^2 - 1)
theta_b_common <- (1 - theta_mu_common) * (theta_mu_common * (1-theta_mu_common) / theta_sigma_common^2)

common_model <- "model {
  theta ~ dbeta(theta_a, theta_b) # Prior
  aut_sample ~ dbin(theta, nObs) # Prevalence in sample data

  aut_pred ~ dbin(theta, nObs) # Predicted prevalence in new sample of same size

  #spec ~ dnorm(spec_mu, 1/spec_sd) # dnorm requires prevalence not sd or var
  #sens ~ dnorm(sens_mu, 1/sens_sd)
  #aut_post <- aut_sample/nObs * sens + (1 - aut_sample/nObs) * spec
}"

common_data <- list(theta_a = theta_a_common,
  theta_b = theta_b_common,
  nObs = nObs,
  aut_sample = sum(chile_bayes_aut$autism) #,
  #spec_mu = 0.996,
  #spec_sd = (1.00-0.99) / (2*1.96),
  #sens_mu = 0.62,
  #sens_sd = (0.66-0.57) / (2*1.96)
)

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

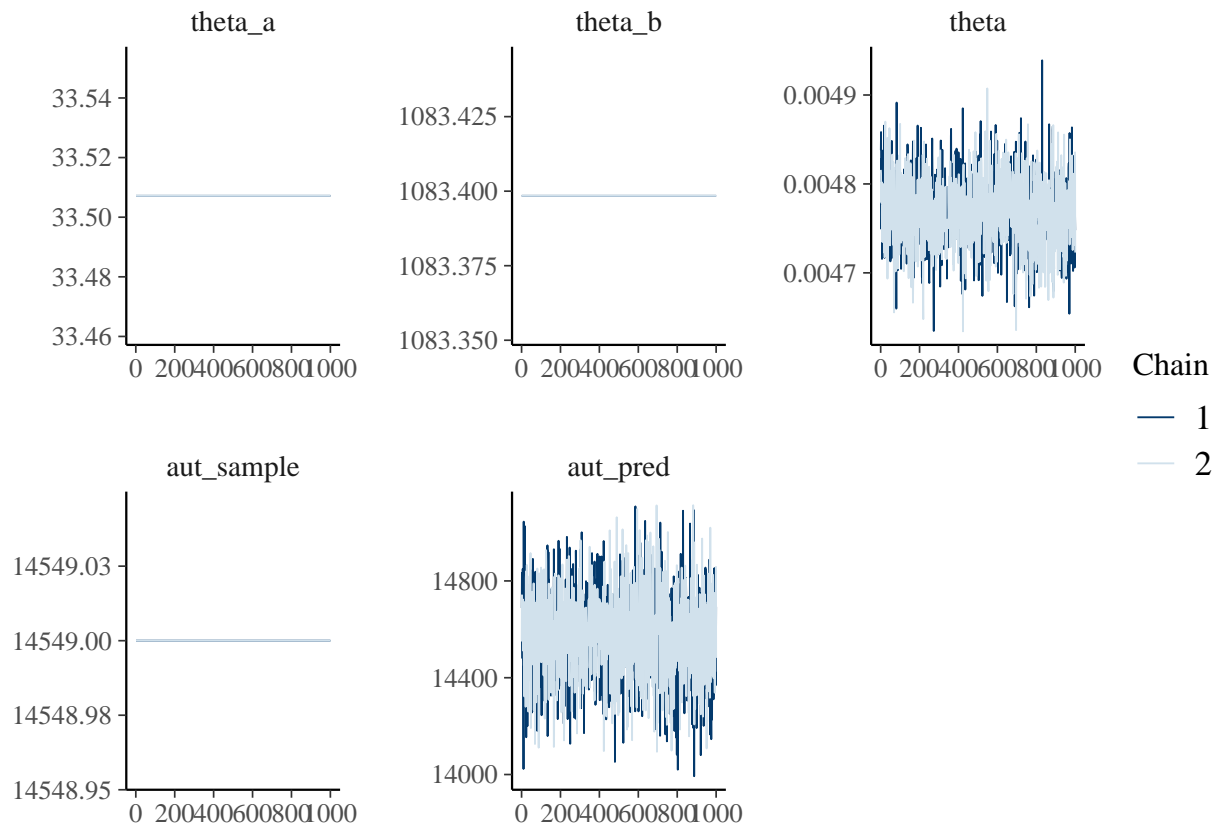
common_pars <- c("theta_a", "theta_b", "theta",
  #"spec", "sens",
  "aut_sample", "aut_pred")

# Run JAGS model and discard burn-in samples
common_jag <- jags.model(textConnection(common_model),
  data = common_data,
  inits = common_ini,
  n.chains = 2,
  quiet = TRUE)
update(common_jag, n.iter = nBurn)
common_sam <- coda.samples(model = common_jag,
  variable.names = common_pars,
  n.iter = nIter)

# Check for convergence in parameters of interest

```

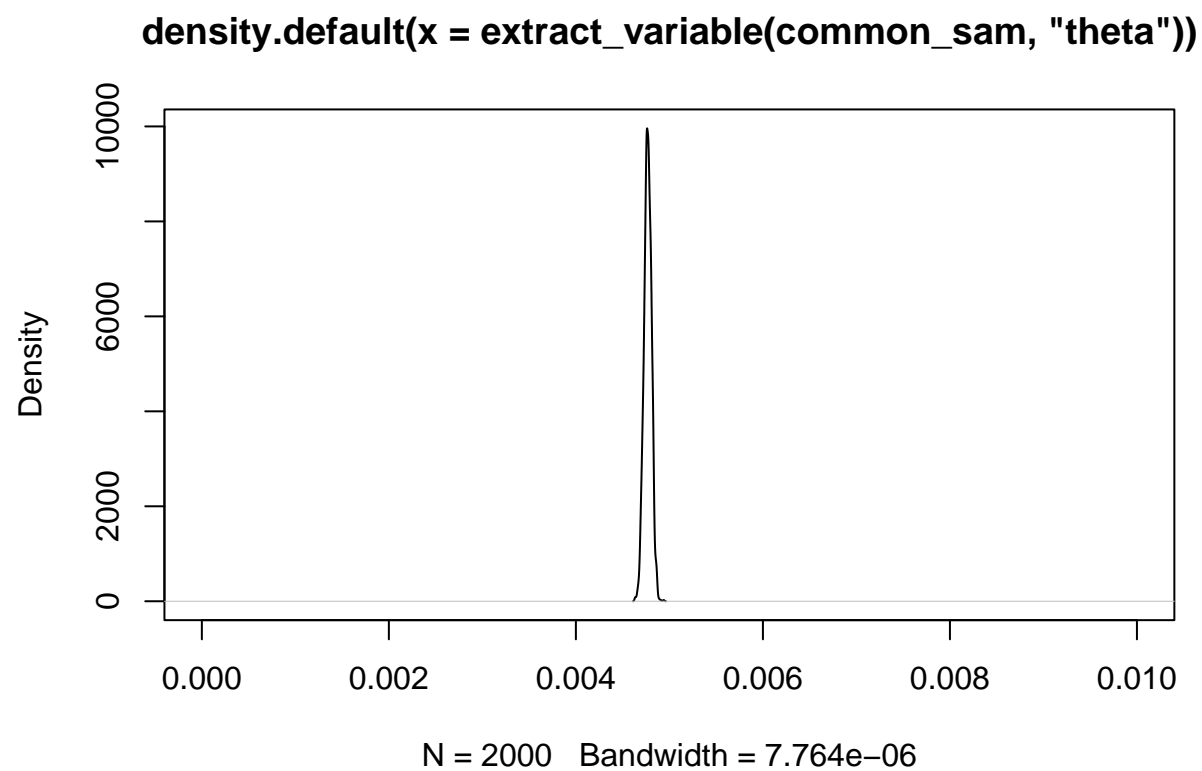
```
mcmc_trace(common_sam, common_pars) # Convergence looks fine and rhats <= 1.1
```



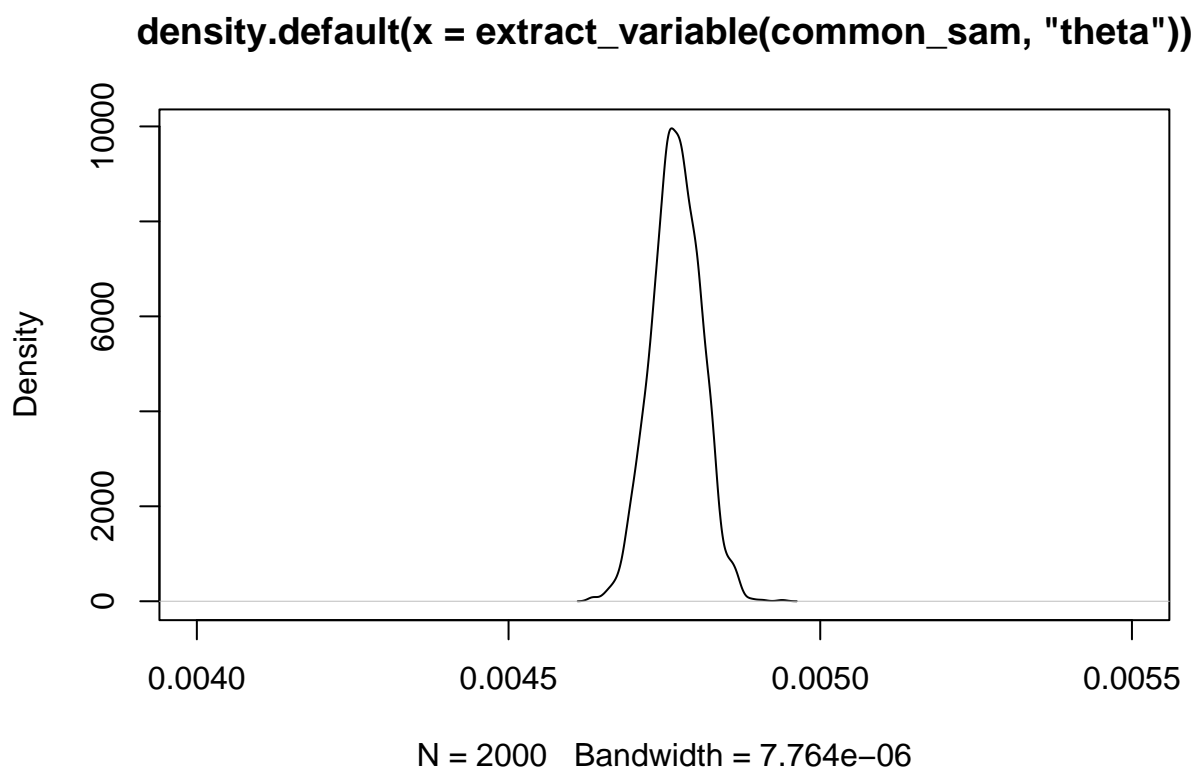
```
summary(as_draws(common_sam)) # mean posterior theta is 0.00477
```

```
## # A tibble: 5 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_pred  1.46e+4  1.46e+4  1.74e+2  1.65e+2  1.43e+4  1.48e+4  1.00  1500.  1761.
## 2 aut_sam~  1.45e+4  1.45e+4  0        0        1.45e+4  1.45e+4  NA    NA    NA
## 3 theta     4.77e-3  4.77e-3  3.94e-5  3.99e-5  4.70e-3  4.83e-3  1.00  1216.  1125.
## 4 theta_a   3.35e+1  3.35e+1  0        0        3.35e+1  3.35e+1  NA    NA    NA
## 5 theta_b   1.08e+3  1.08e+3  0        0        1.08e+3  1.08e+3  NA    NA    NA
## # ... with abbreviated variable names 1: ess_bulk, 2: ess_tail
```

```
plot(density(extract_variable(common_sam, "theta")), xlim = c(0,0.01))
```



```
plot(density(extract_variable(common_sam, "theta")), xlim = c(0.004,0.0055))
```



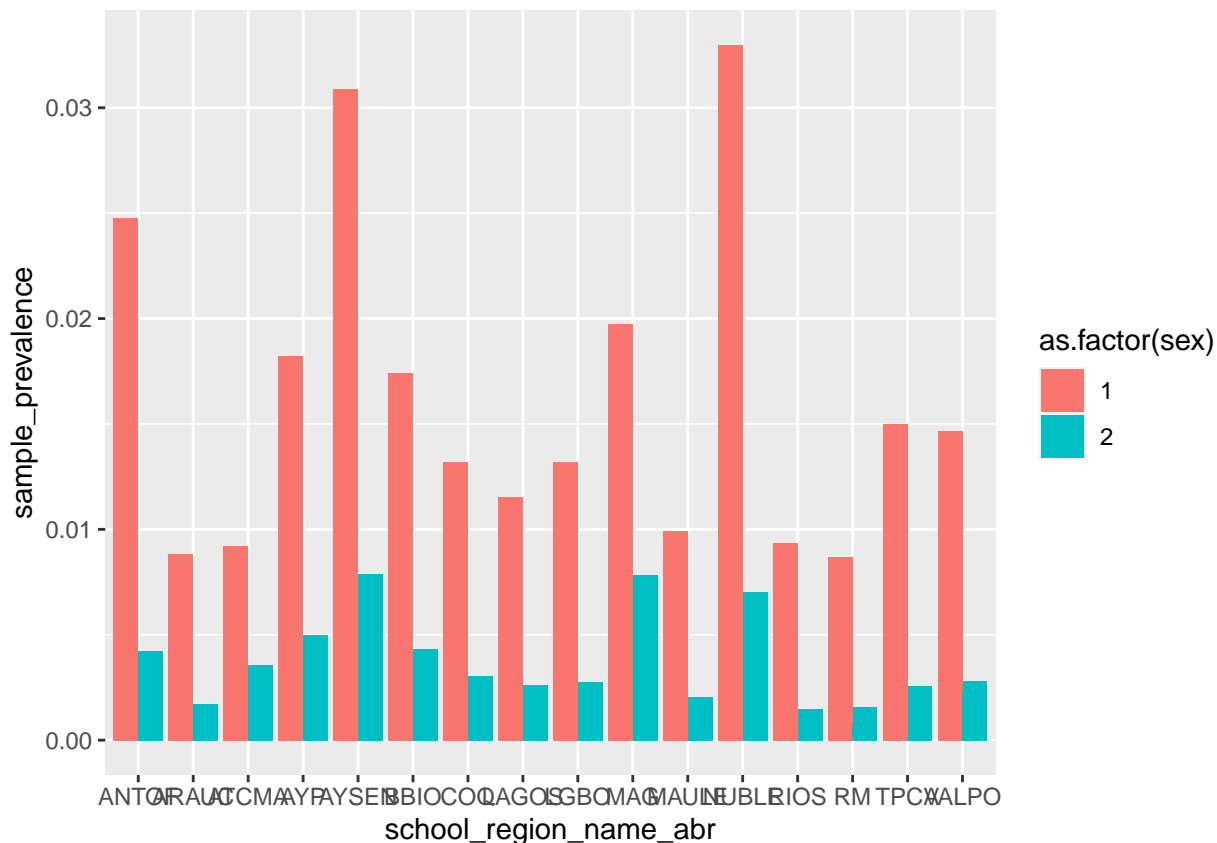
*# Very very narrow posterior distribution centered approx at sample prevalence of 0.00476.
Not that surprising given uniform prior was used.*

Informative prior made no difference to posterior distribution

```
aut_prev_region <- get_grouped_prev(x = chile_bayes_aut, stdpop = chile_stdpop,
                                     grouping_vars = c("school_region_name_abr", "age_june30", "sex", "a
```

```
## `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)),
  geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = sex, fill = as.factor(sex)),
```

1 is male, 2 is female

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

By region, with sensitivity analysis (alter prior mean and sd)

```
aut_prev_region_adj <- get_adjusted_prev(aut_prev_region, grouping_vars = "school_region_name_abr")

nRegion <- length(unique(aut_prev_region$school_region_name_abr))
RegionNames <- sort(unique(aut_prev_region_adj$school_region_name_abr))

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_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])
  }
}"

for(j in 1:length(theta_mu)) {
  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 = 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 <= 1
  rand_region_summ <- summary(subset_draws(as_draws(rand_region_sam), 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

# Plot
aut_prev_region_post <- as_tibble(as_draws_matrix(rand_region_sam), rownames = "Iteration") %>%
  select(c("Iteration", contains("theta["))) %>%
  pivot_longer(cols = contains("theta["),
    names_to = "Region",
    values_to = "predicted_prev") %>%
  mutate(school_region_name_abr = factor(Region,
    levels = c(paste0("theta[", 1:nRegion, "]")),
    labels = RegionNames))

aut_prev_region_ci <- aut_prev_region_post %>%
  group_by(school_region_name_abr) %>%
  summarise(post_lower = quantile(predicted_prev, 0.025),
    post_upper = quantile(predicted_prev, 0.975))

(ggplot() +
  geom_density(data = aut_prev_region_post, aes(x = predicted_prev)) +
  geom_vline(data = aut_prev_region_ci, aes(xintercept = post_lower), color = "blue", linetype = "dotted") +
  geom_vline(data = aut_prev_region_ci, aes(xintercept = post_upper), color = "blue", linetype = "dotted") +
  geom_vline(data = aut_prev_region_adj, aes(xintercept = ci_lower), color = "red", linetype = "dashed") +
  geom_vline(data = aut_prev_region_adj, aes(xintercept = ci_upper), color = "red", linetype = "dashed") +
  facet_wrap(~school_region_name_abr))
}

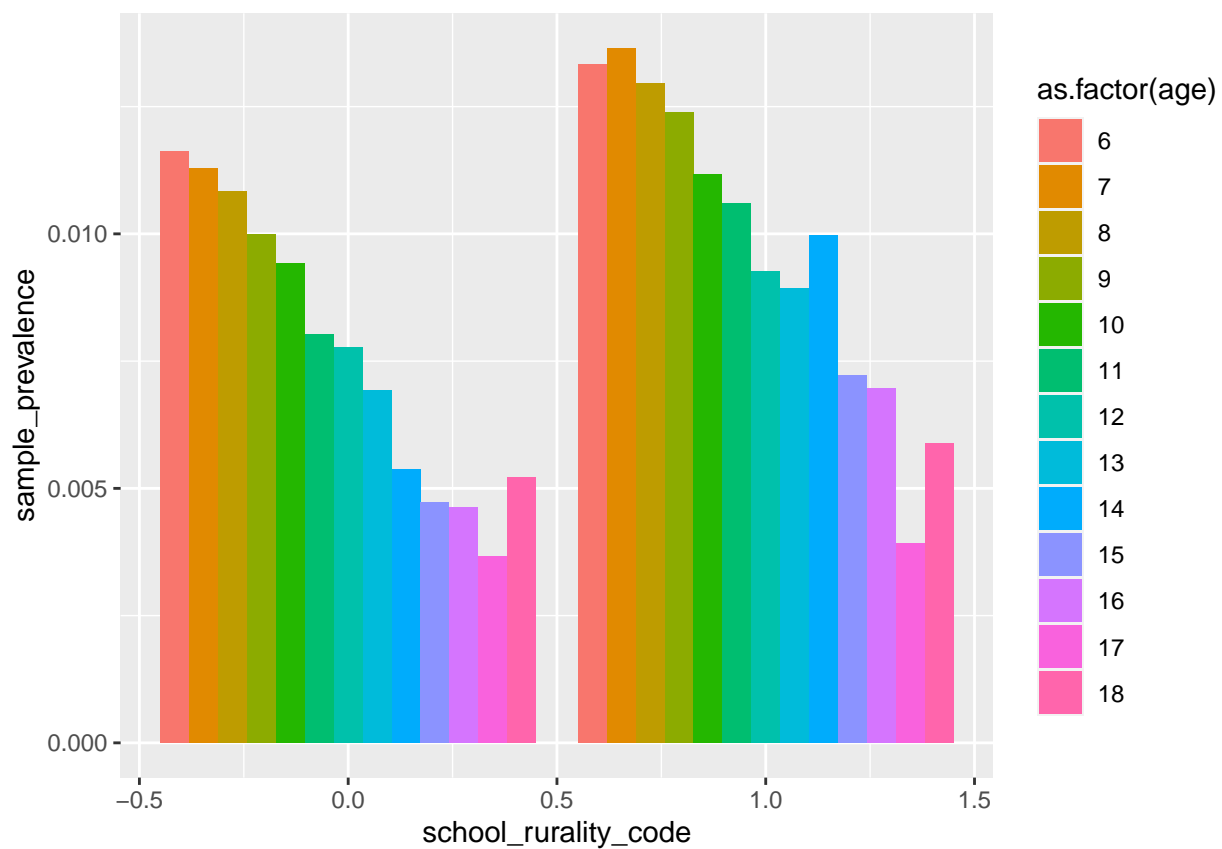
```

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)), position = "dodge")
```



```
#geom_col(aes(x = school_region_name_abr, y = prevalence, group = sex, fill = as.factor(sex)), position = "dodge")
# 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 1
            #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*(adjusted_rate^2 / var)),
            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))

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])
  }
}"

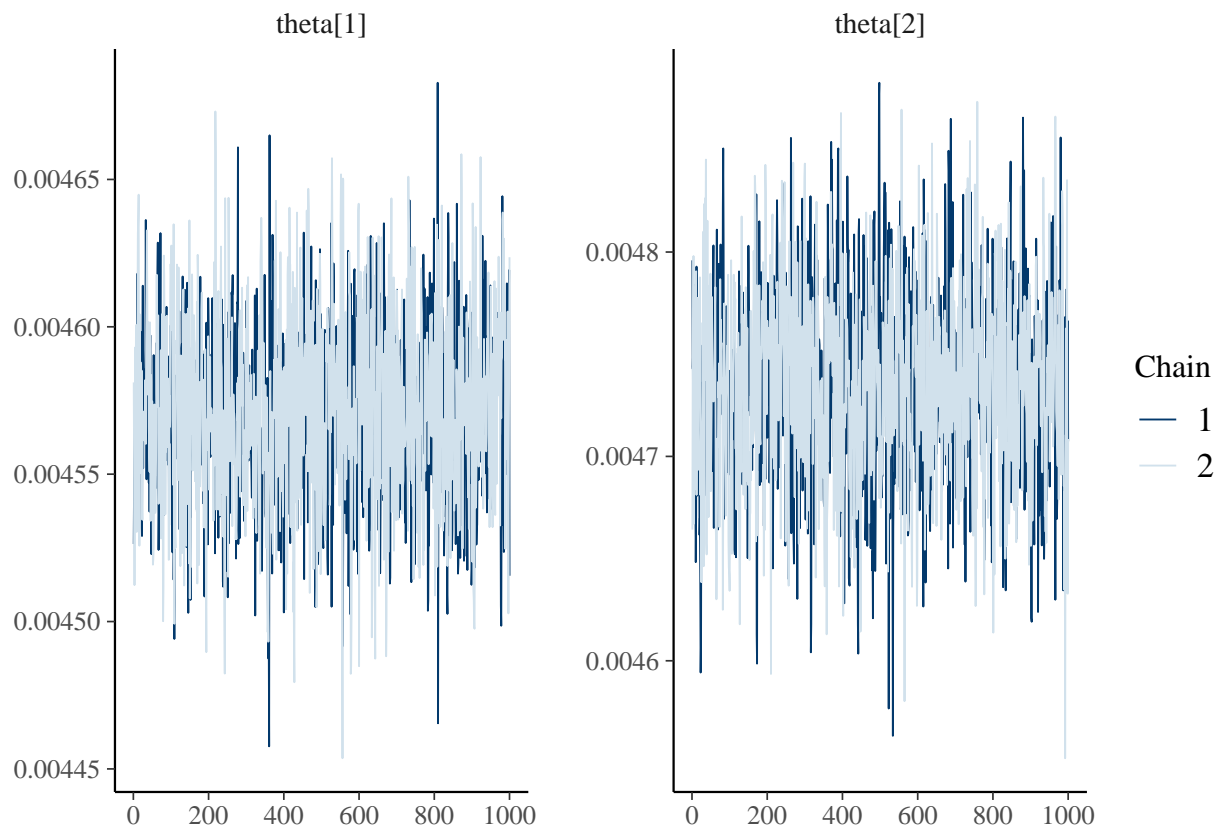
rand_rural_data <- list(theta_a = theta_a,
                       theta_b = theta_b,
                       nObs = aut_prev_rural_adj$sum_sample_pop_size,
                       aut_sample = aut_prev_rural_adj$adjusted_count,
                       nRural = nRural)

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

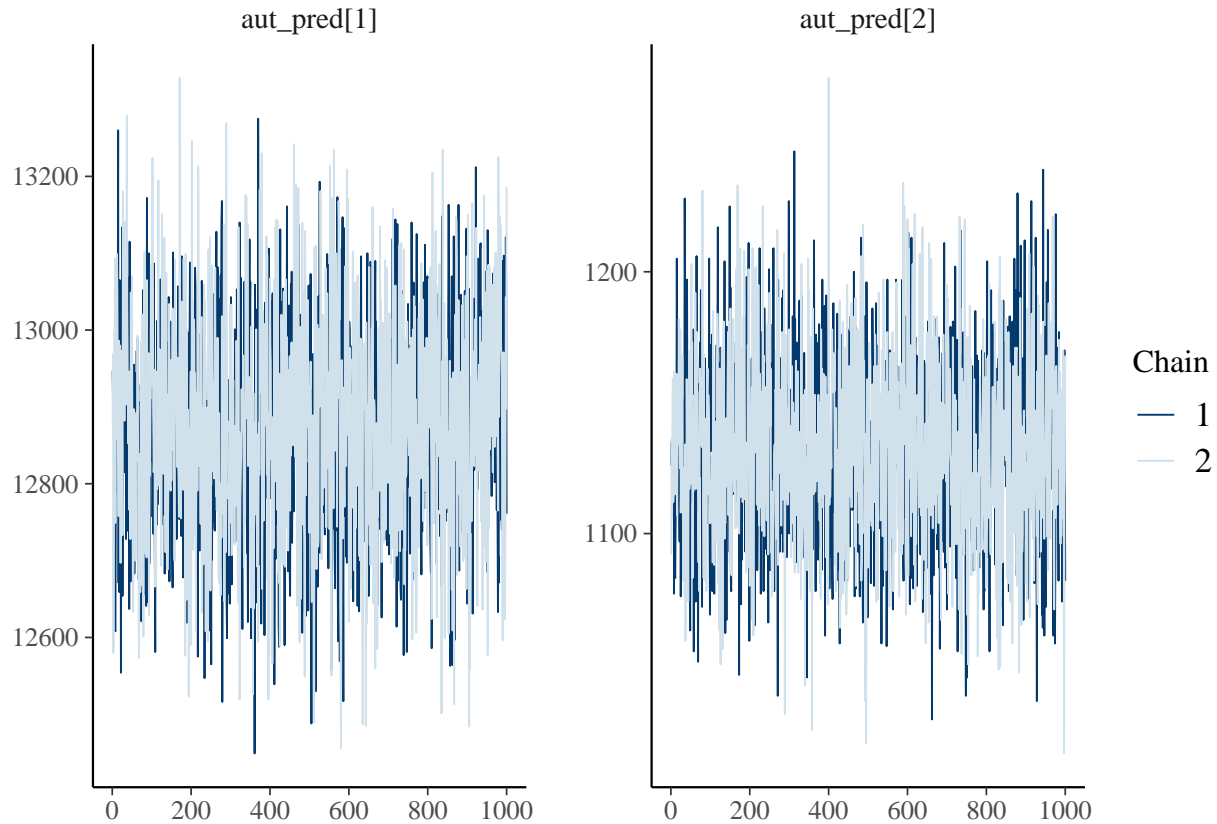
# Run JAGS model and discard burn-in samples
rand_rural_jag <- jags.model(textConnection(rand_rural_model),
                             data = rand_rural_data,
                             #inits = rand_region_ini,
                             n.chains = 2,
                             quiet = TRUE)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

```

```
# Check for convergence in parameters of interest
#mcmc_trace(rand_region_sam, 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
##   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~ 1.29e+4 1.29e+4 1.41e+2 1.42e+2 1.26e+4 1.31e+4 1.00   1487.   1626.
## 2 aut_pr~ 1.13e+3 1.13e+3 3.53e+1 3.56e+1 1.08e+3 1.19e+3 0.999  2026.   1818.
## 3 aut_sa~ 1.28e+4 1.28e+4 0         0      1.28e+4 1.28e+4 NA      NA      NA
## 4 aut_sa~ 1.37e+3 1.37e+3 0         0      1.37e+3 1.37e+3 NA      NA      NA
## 5 theta[~ 4.57e-3 4.57e-3 3.14e-5 3.14e-5 4.52e-3 4.62e-3 1.01   1381.   1246.
## 6 theta[~ 4.73e-3 4.74e-3 4.81e-5 4.74e-5 4.66e-3 4.81e-3 1.00   1219.   1267.
## 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: variable, 2: ess_bulk, 3: ess_tail
```

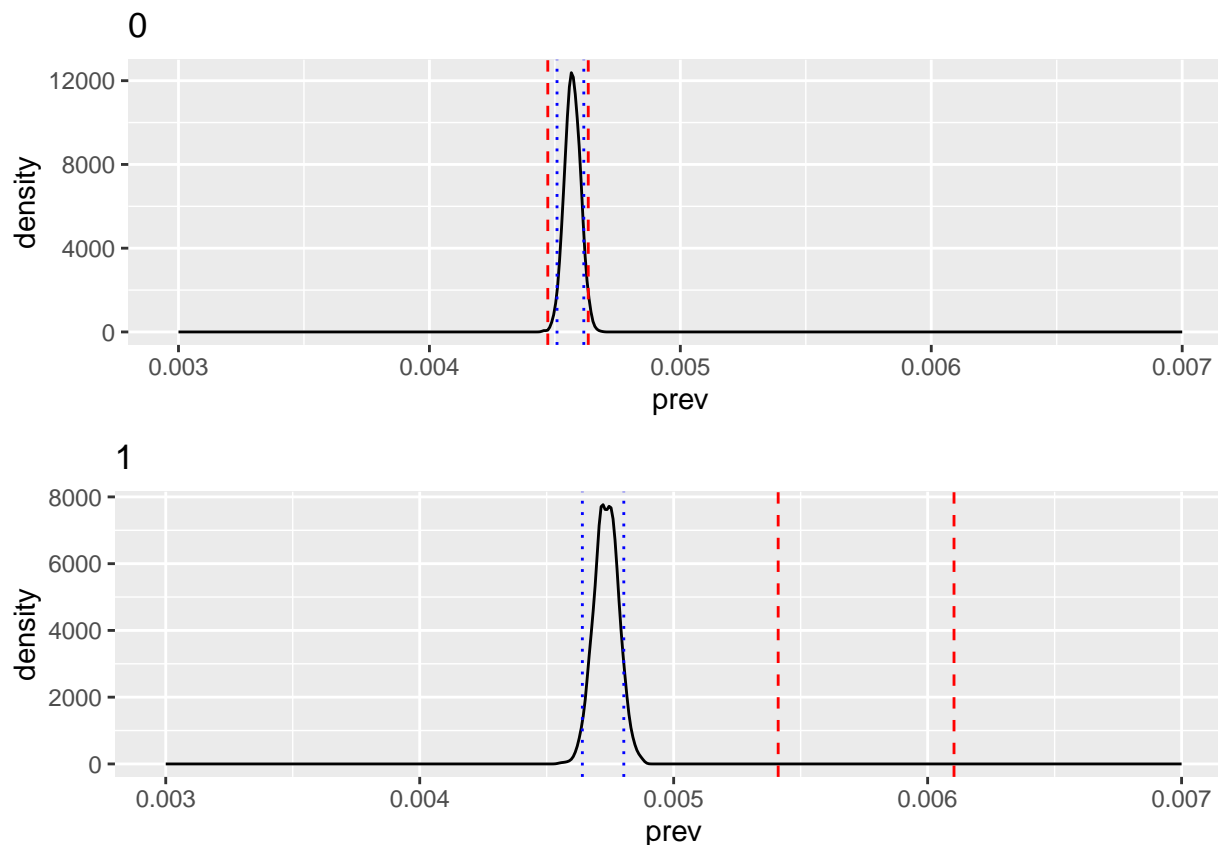
```
rand_rural_summ <- summary(subset_draws(as_draws(rand_rural_sam), pars),
  ~quantile(.x, probs=c(0.025, 0.5, 0.975)),
  ~mcse_quantile(.x, probs=c(0.025, 0.5, 0.975)),
  "rhat") %>%
  arrange(desc(mcse_q50))
rand_rural_summ
```

```
## # 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]  12597.    1.29e+4 1.31e+4  5.5 e+0  3.5 e+0  6 e+0  1.00
## 2 aut_pred[2]   1066    1.13e+3 1.20e+3  2.5 e+0  1.5 e+0  2 e+0  0.999
```

```
## 3 theta[2]          0.00464  4.74e-3 4.83e-3  3.59e-6  1.79e-6  4.38e-6  1.00
## 4 theta[1]          0.00451  4.57e-3 4.63e-3  3.52e-6  9.47e-7  2.40e-6  1.01
## 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)
```



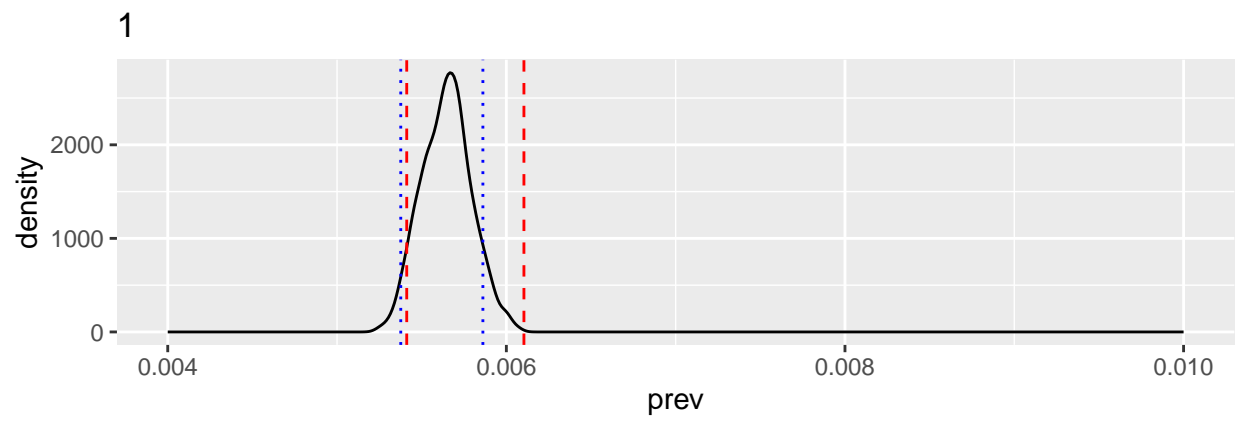
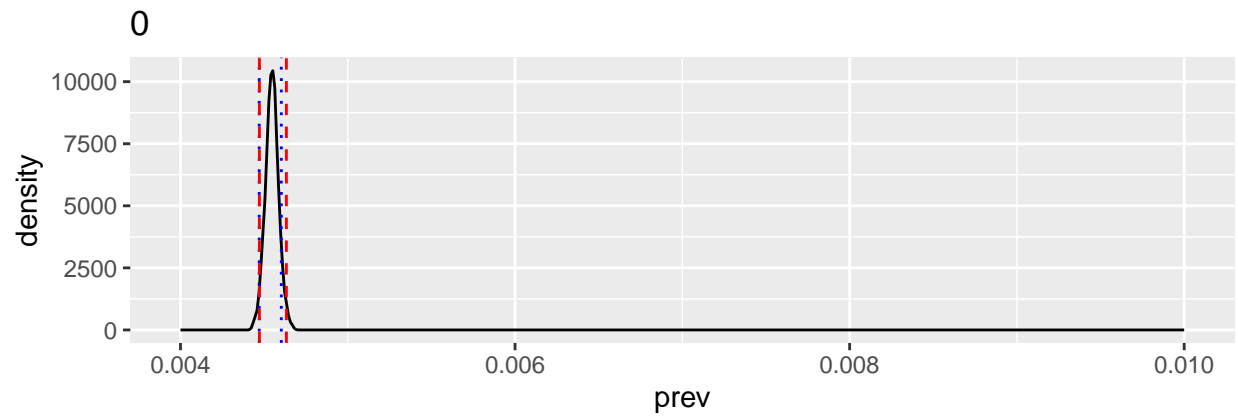
```

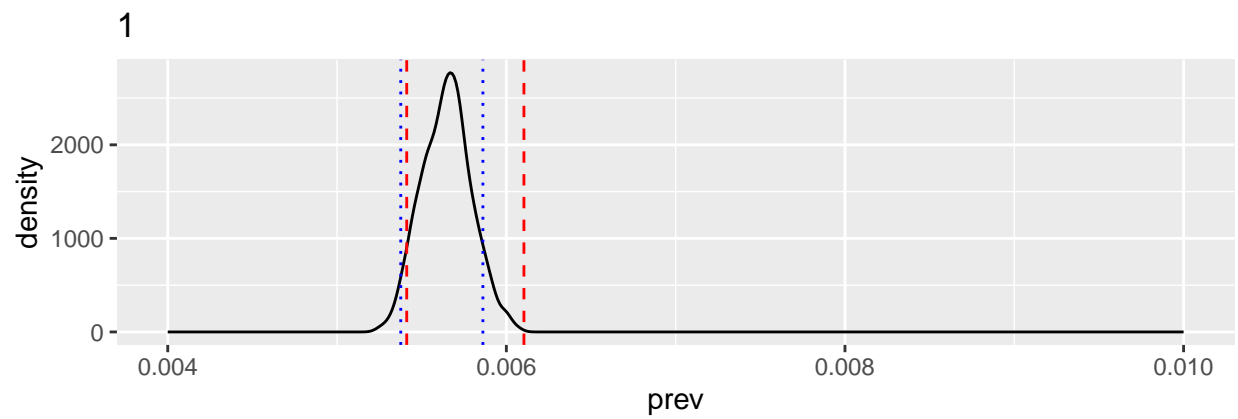
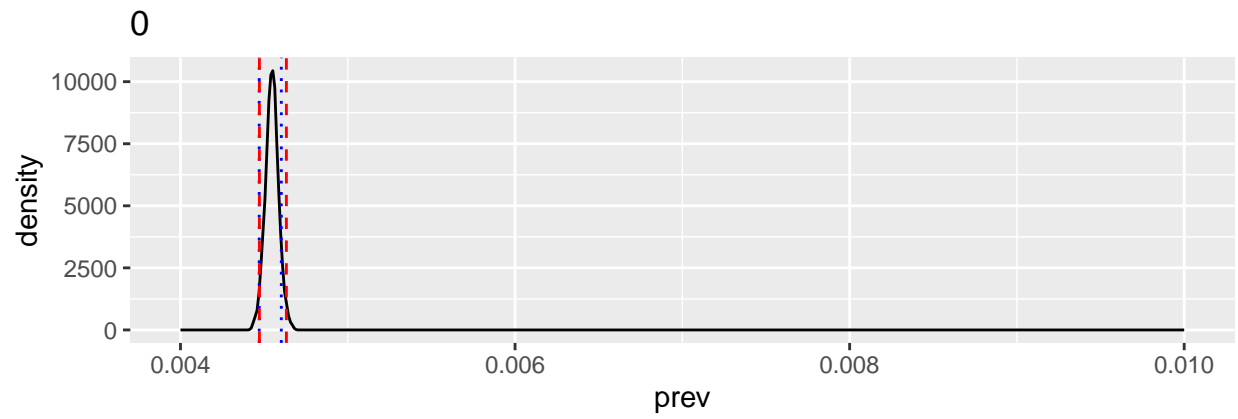
rand_rural_sam <- coda.samples(model = rand_rural_jag,
                              variable.names = pars,
                              n.iter = nIter)

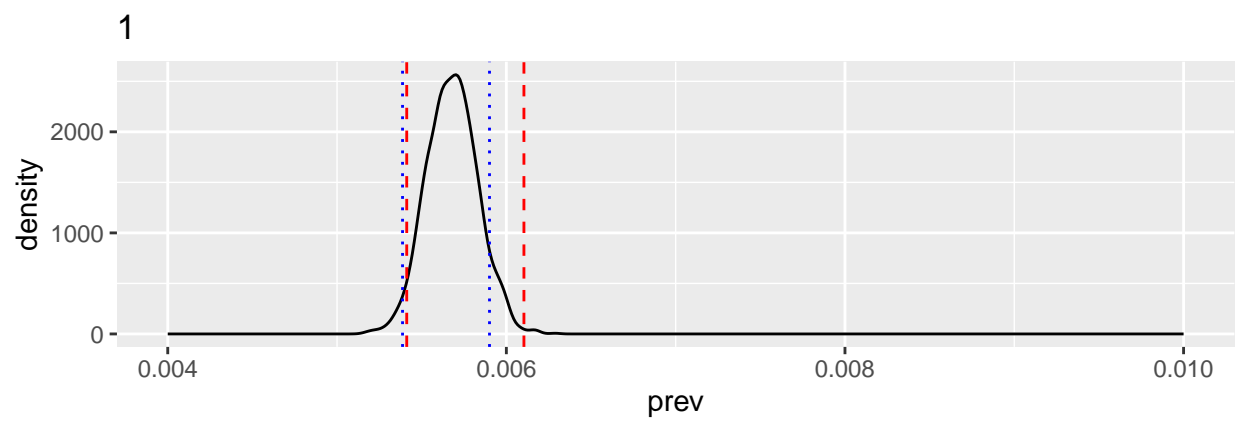
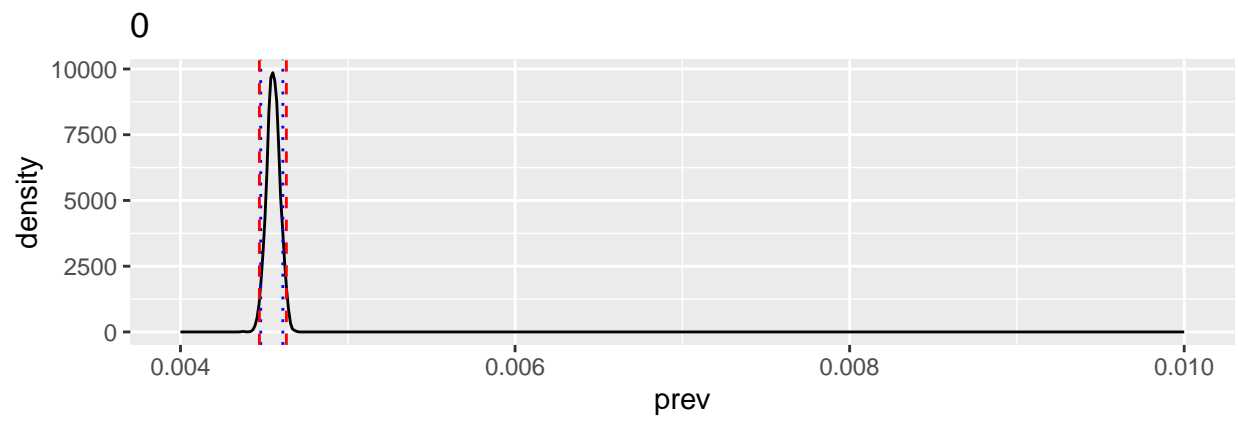
# Plots
aut_prev_rural_plots <- list()
rural_post_ci_lower <- list()
rural_post_ci_upper <- list()

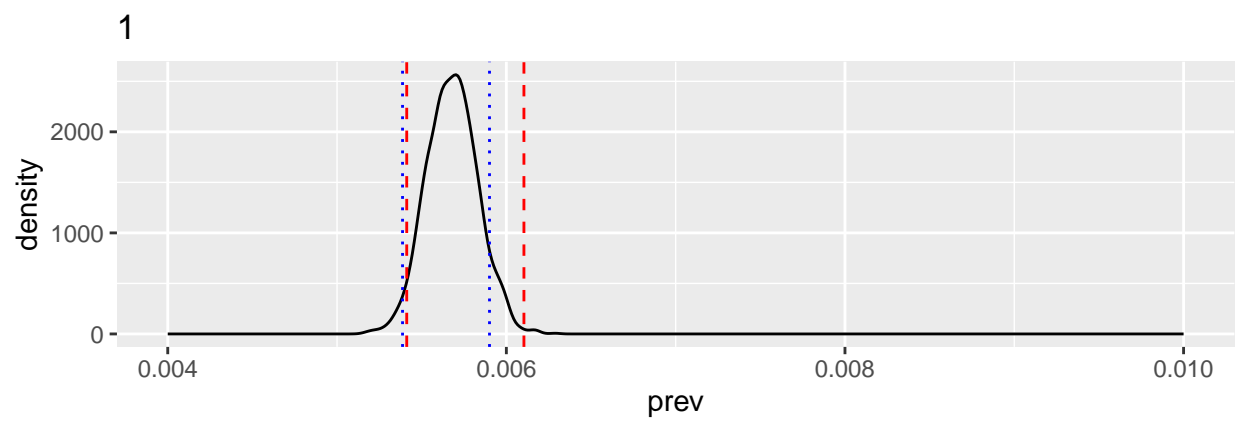
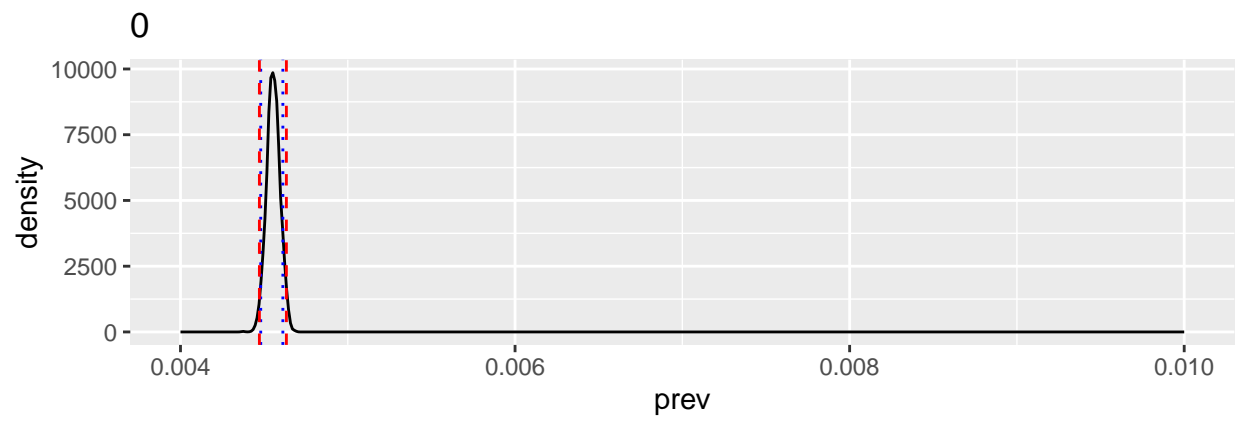
for(i in 1:nRural) {
  prevs <- data.frame(prev = extract_variable(rand_rural_sam, paste0("theta[", i, "]")))
  rural_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
  rural_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
  density_plot <- ggplot(prevs, aes(x = prev), color = "blue") +
    geom_density() +
    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 = 10)
}

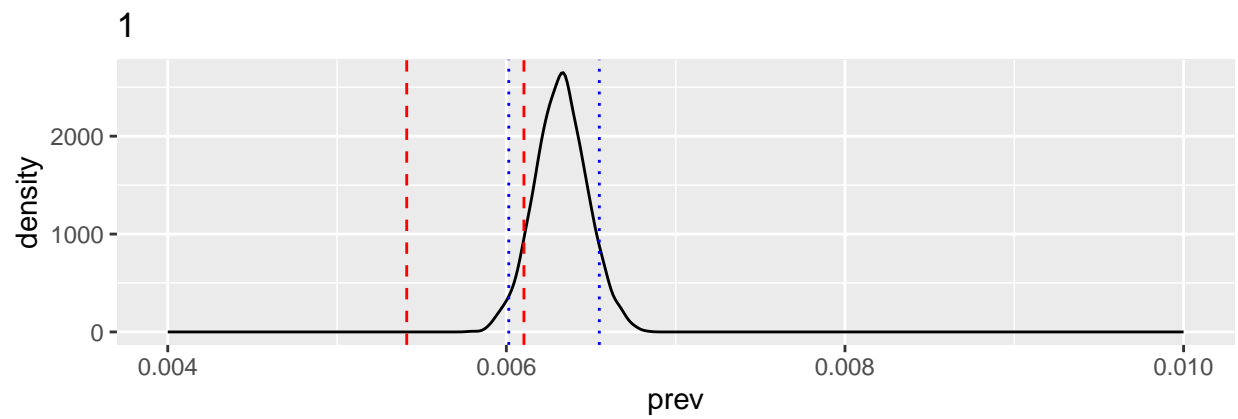
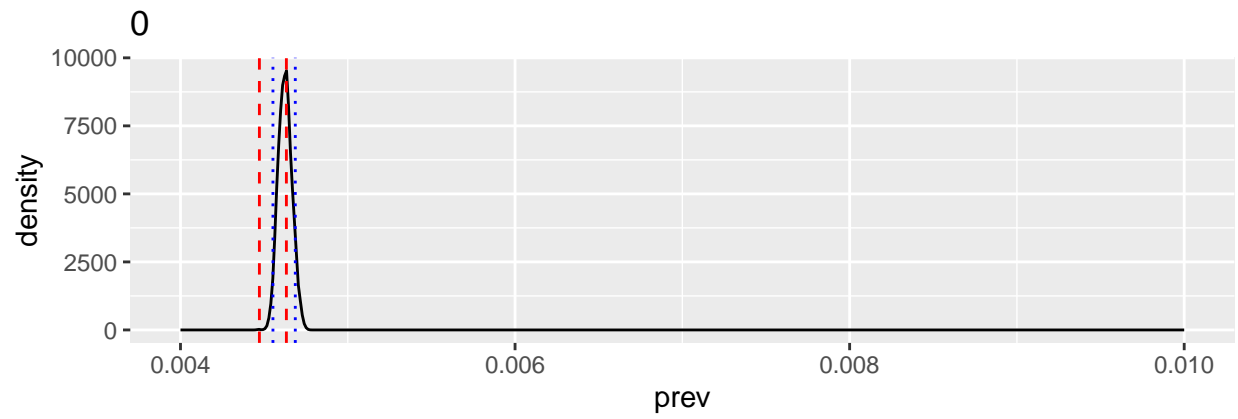
```

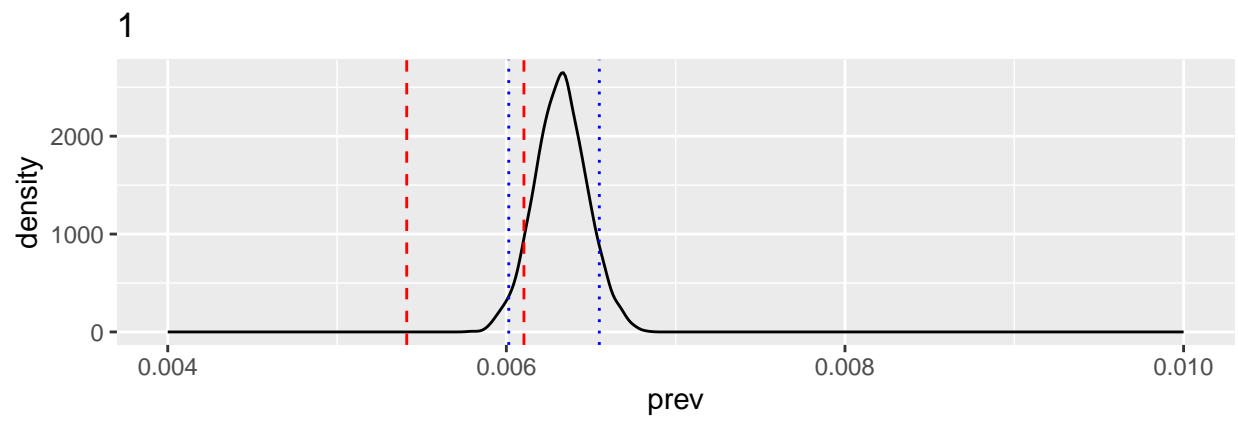
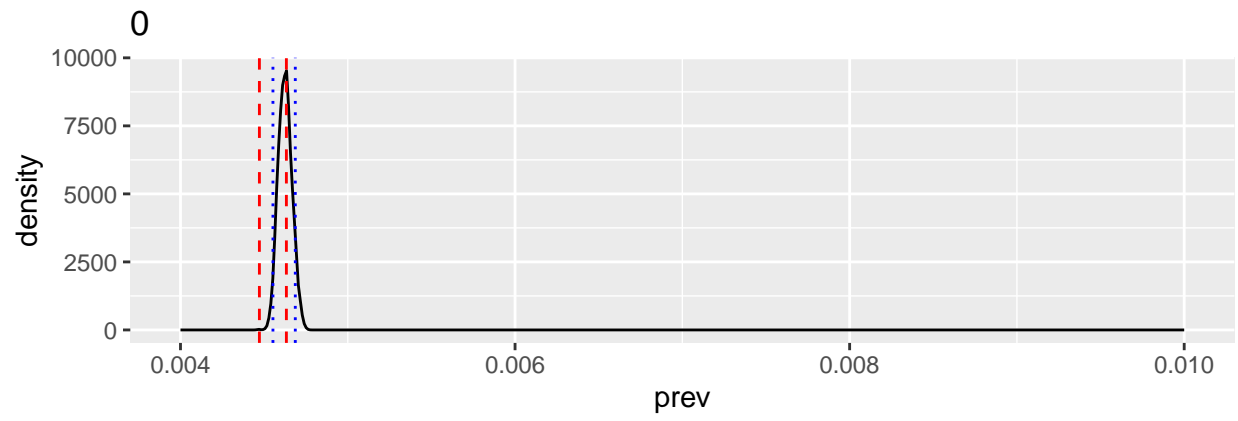


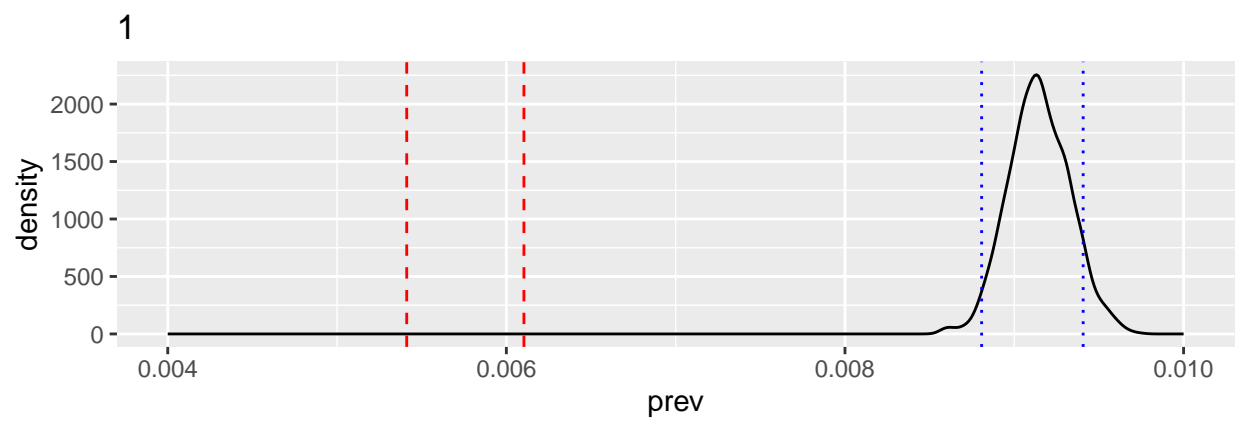
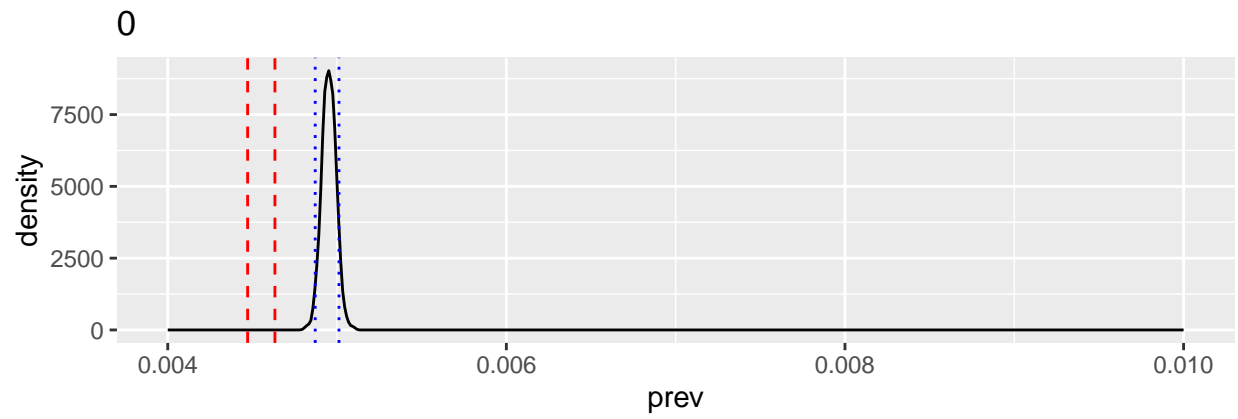


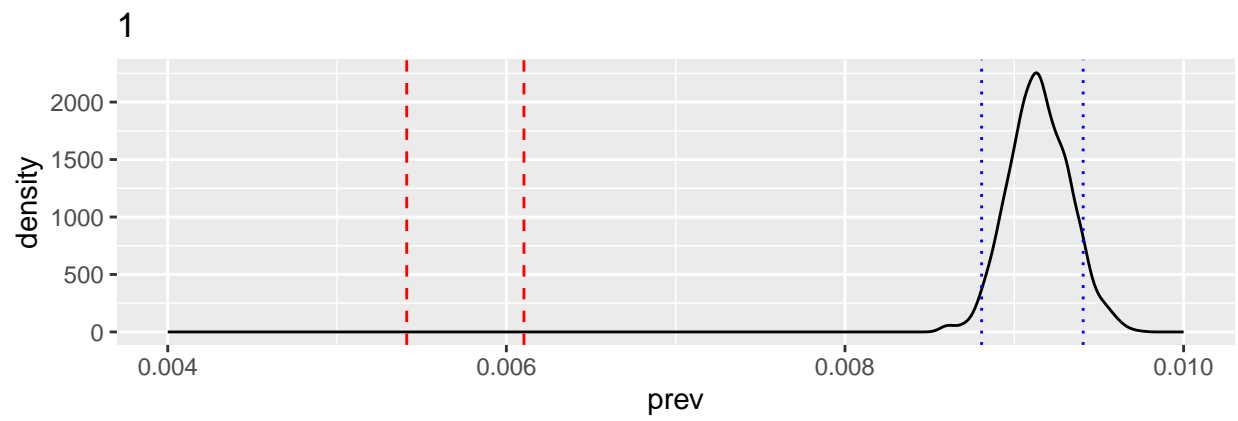
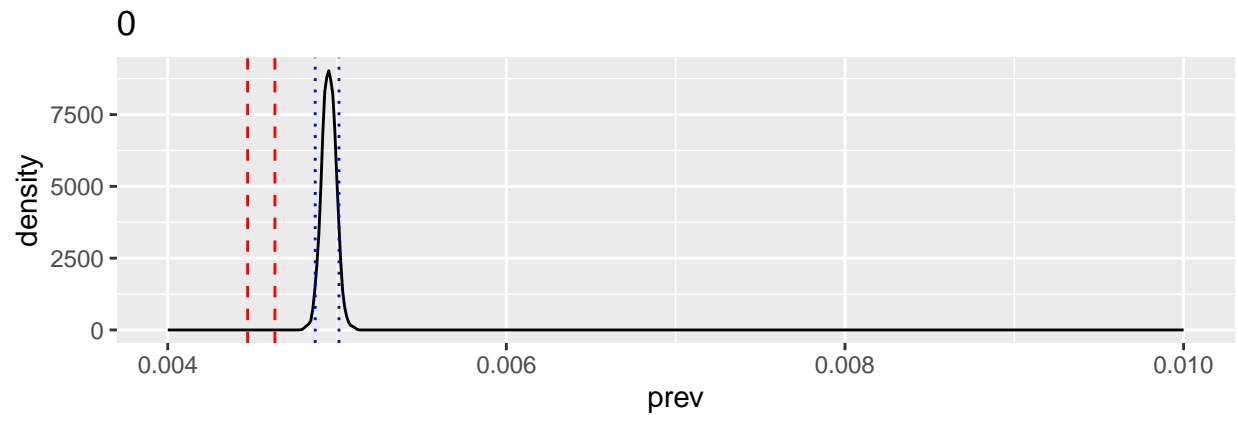


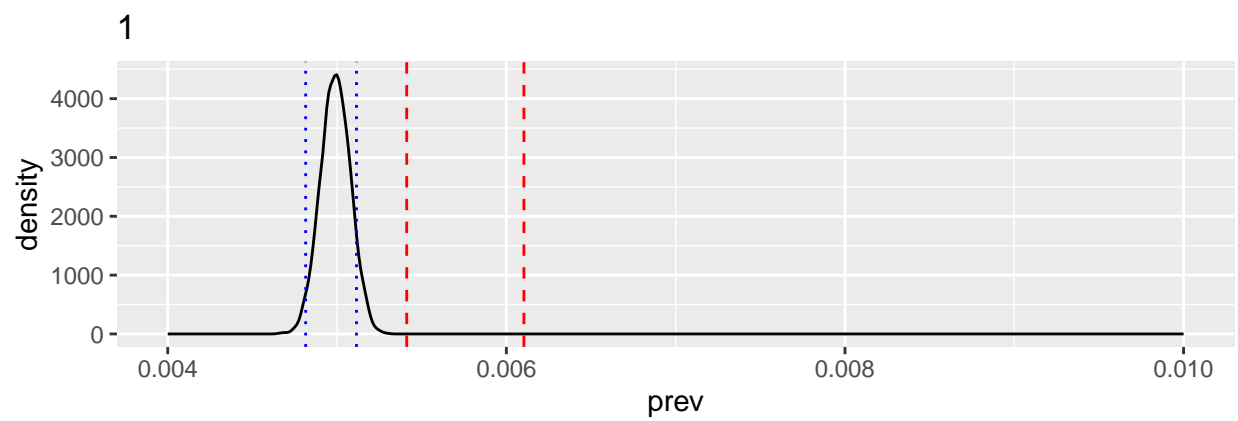
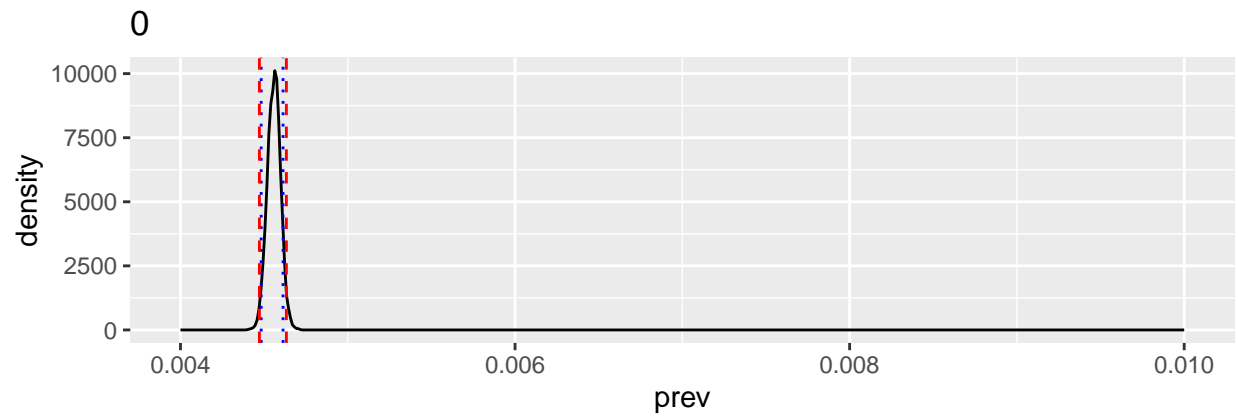


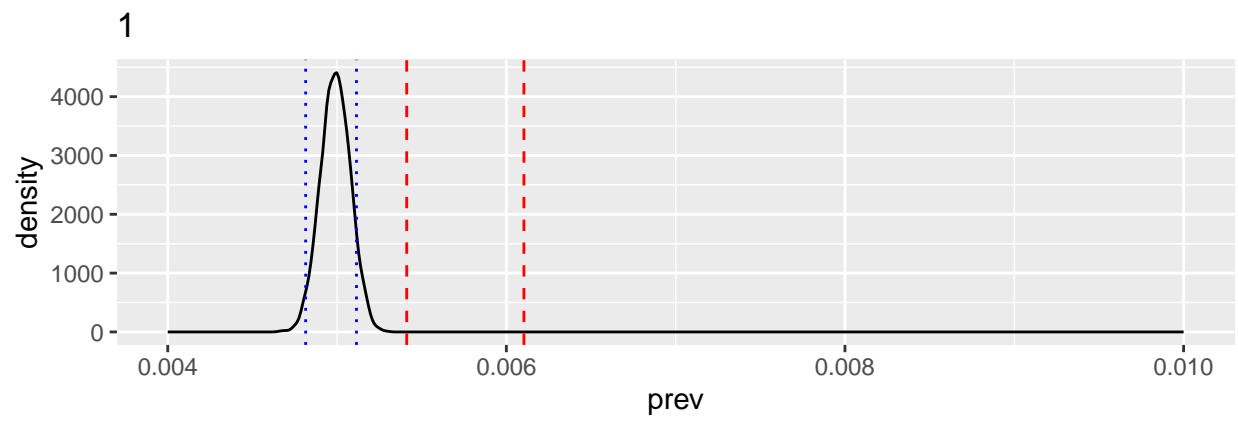
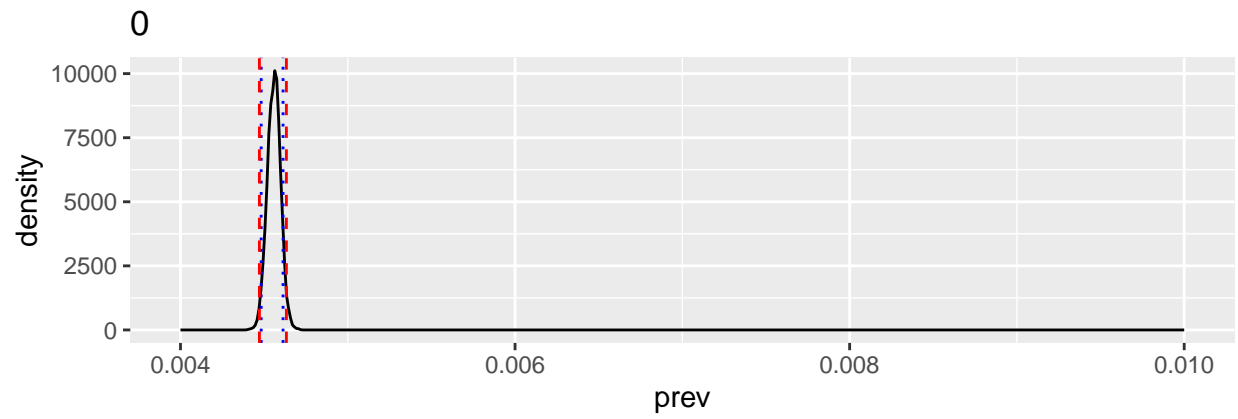


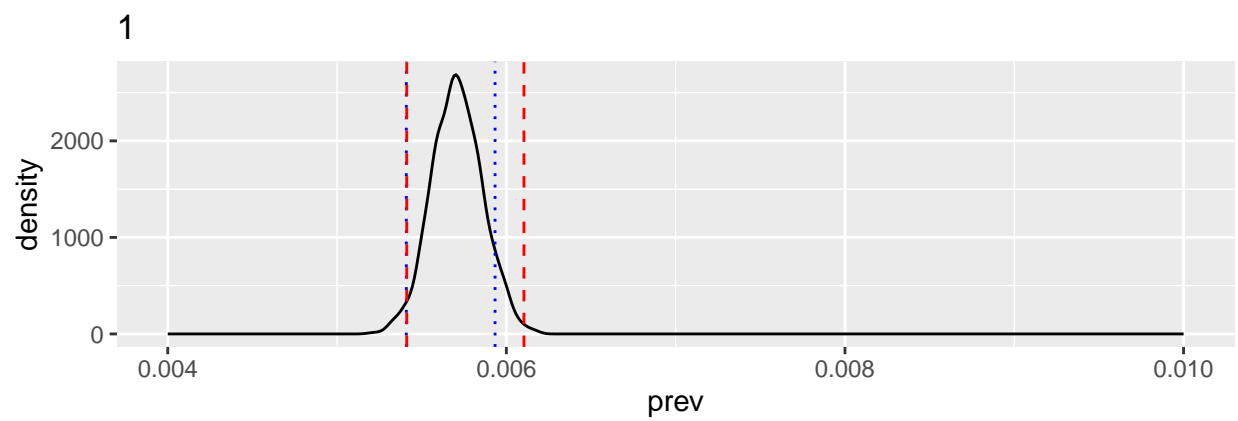
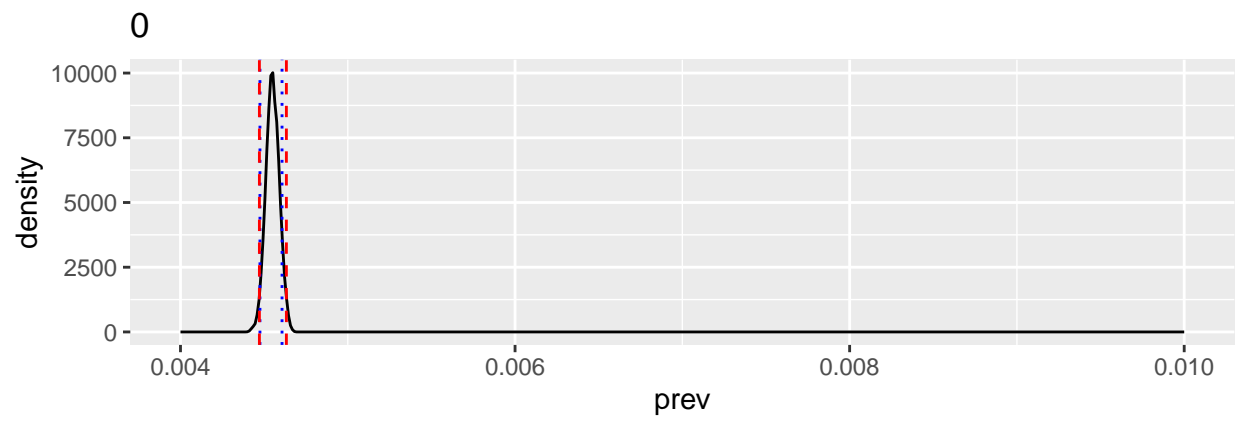


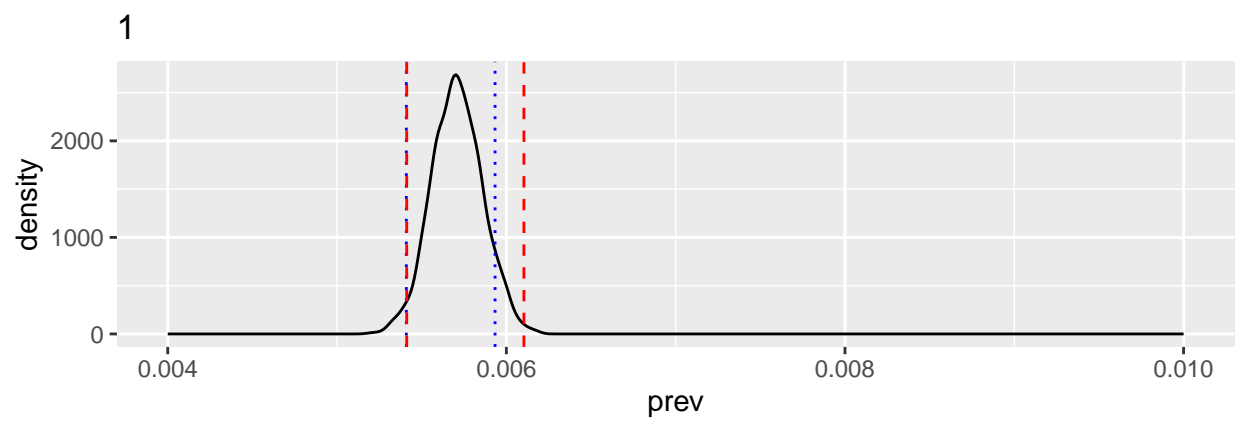
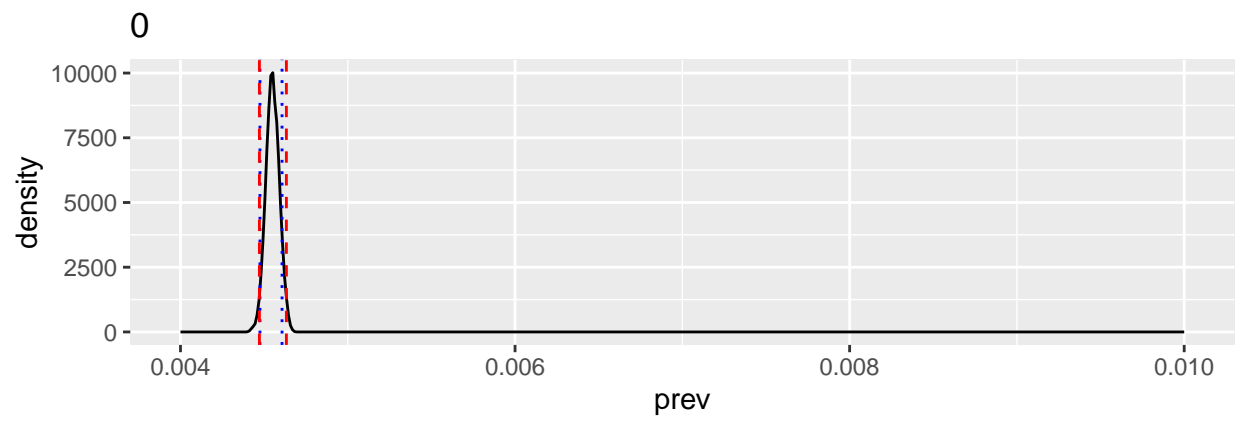


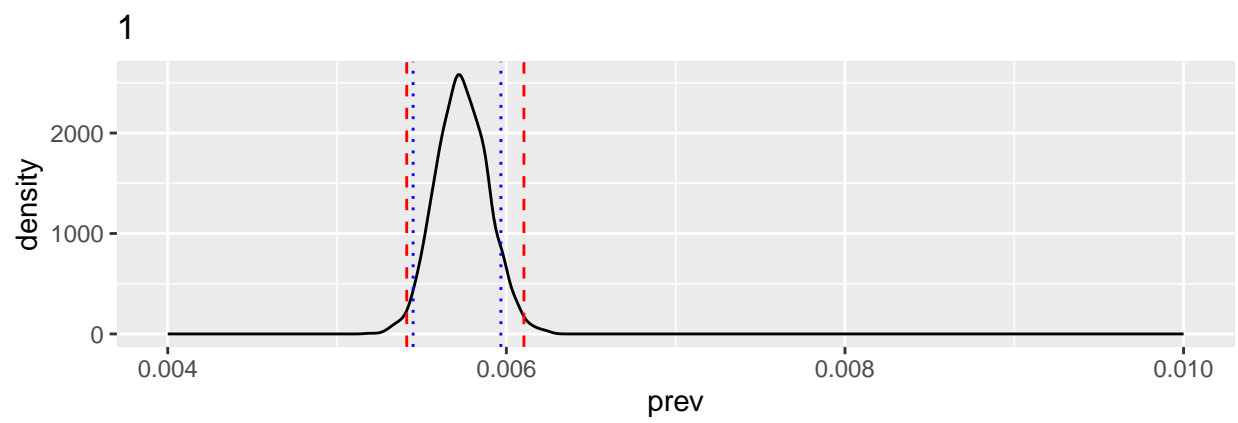
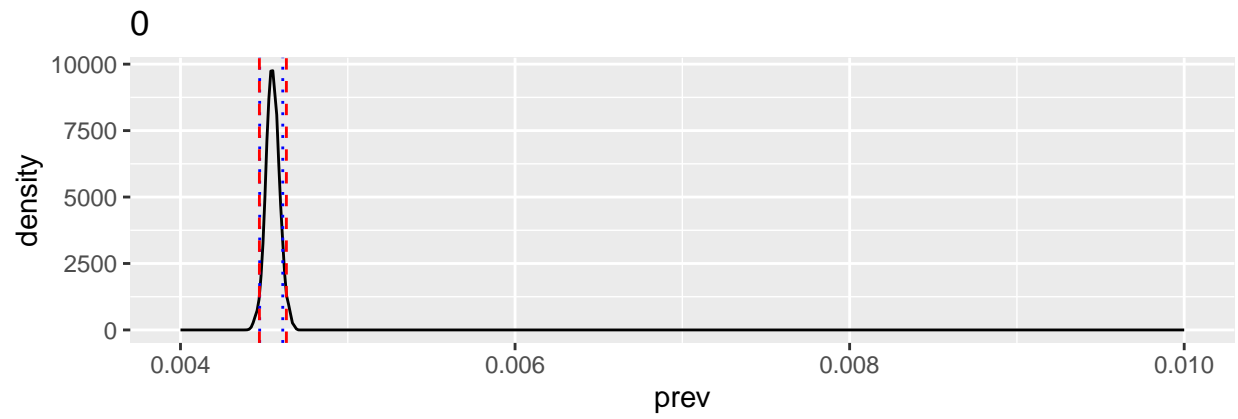


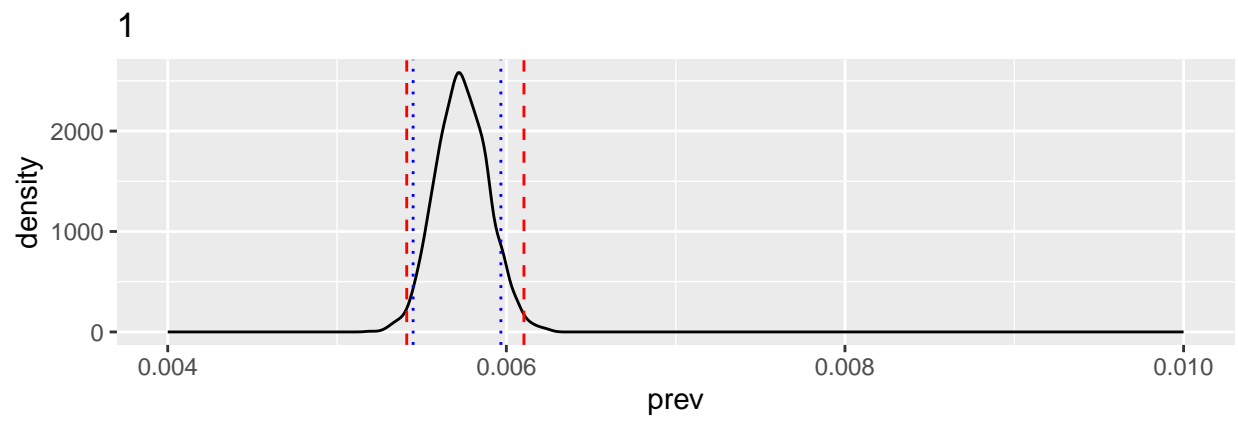
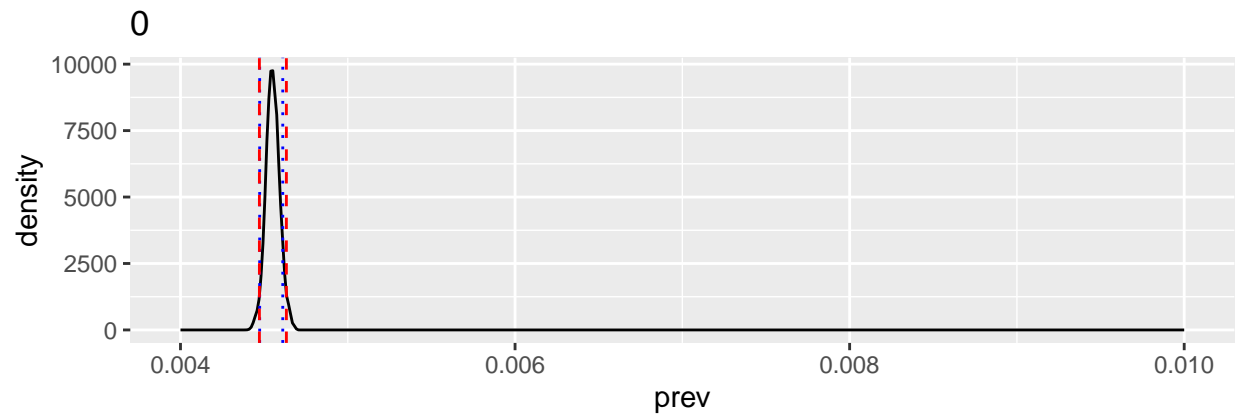


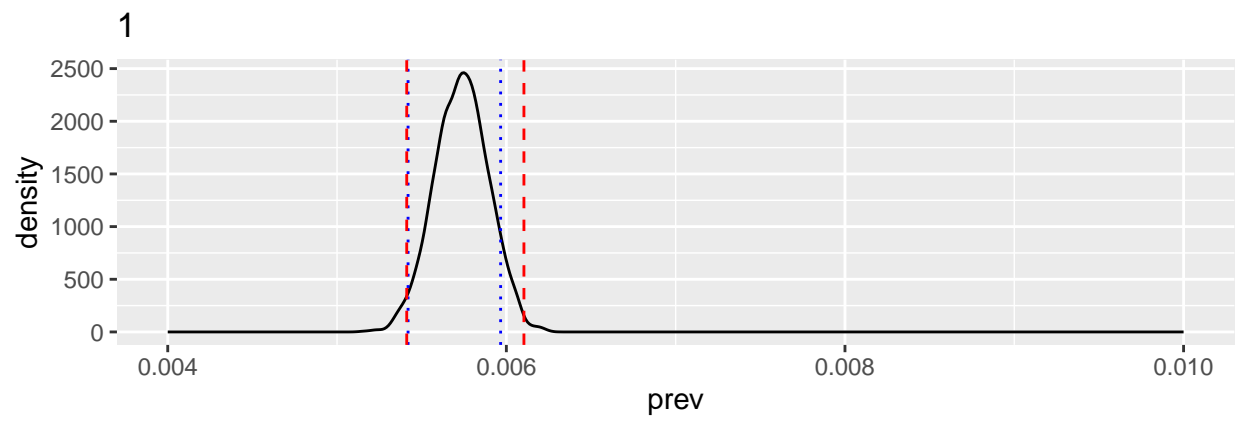
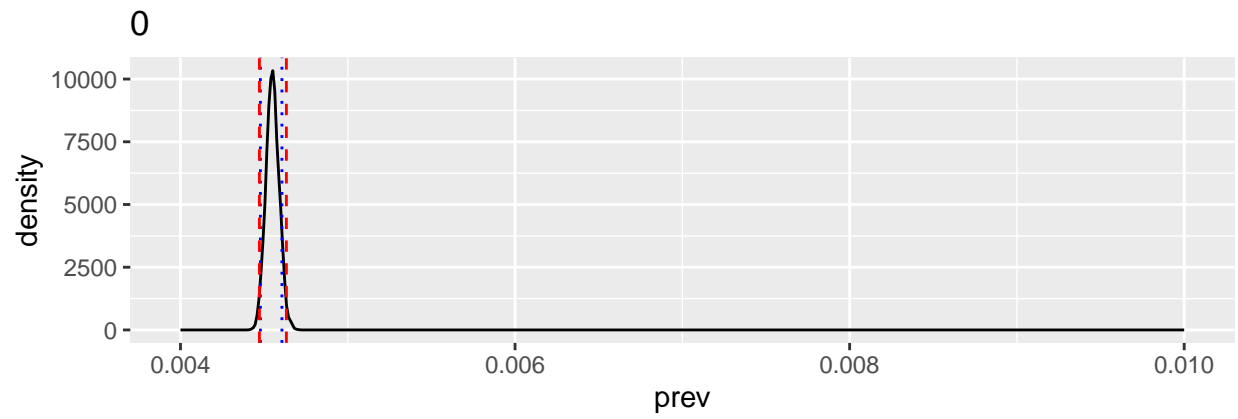


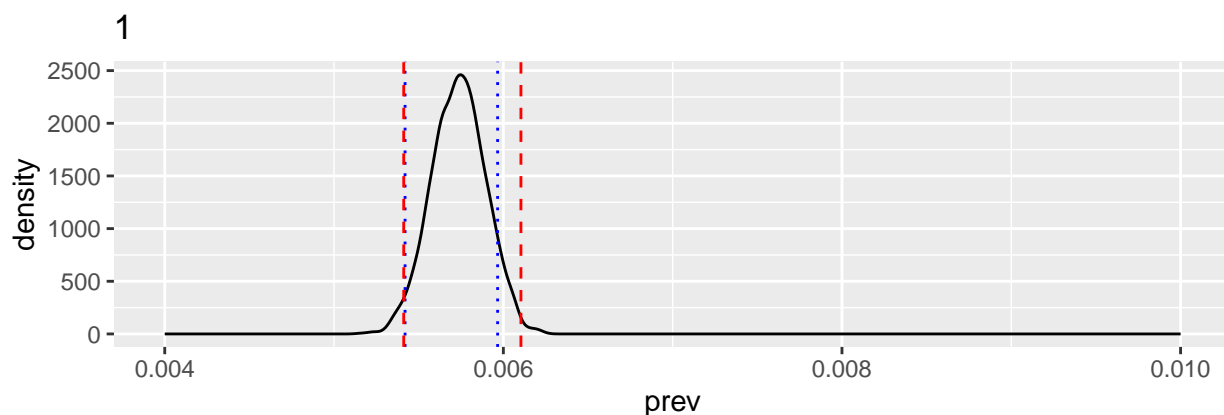
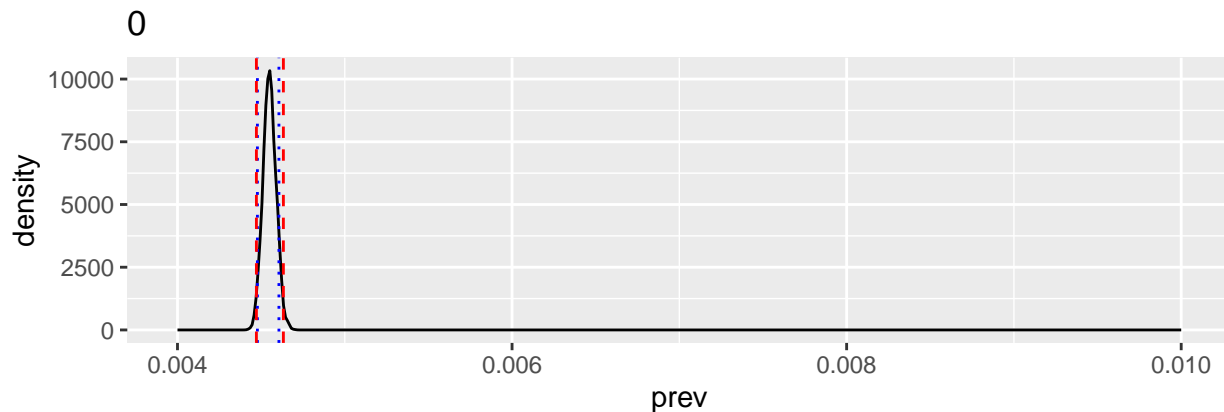












```
#####
```

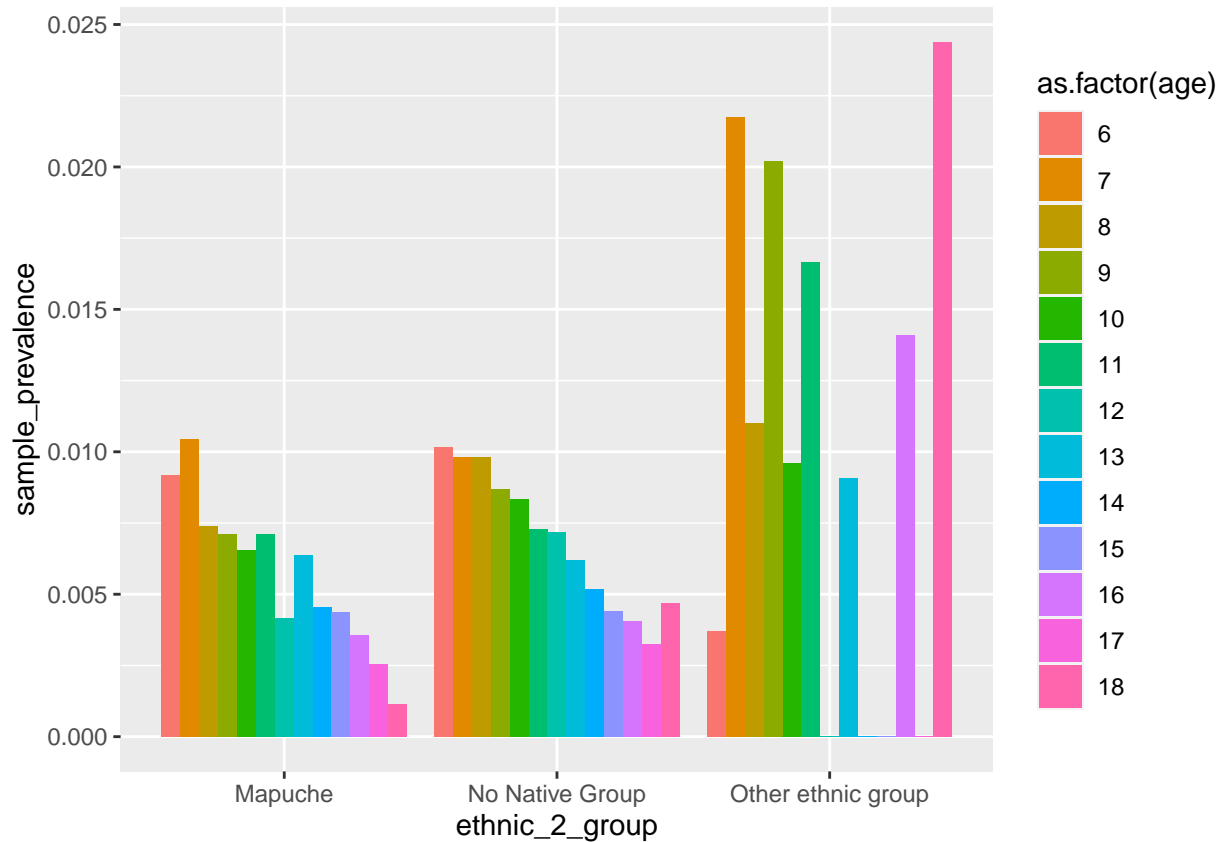
```
### Bayesian prevalence by ethnicity
```

```
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)), position = "stack")
```

```
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)), position
# 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 1
            #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*(adjusted_rate^2 + w_M^2)),
            #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)
```

```

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])
  }
}"

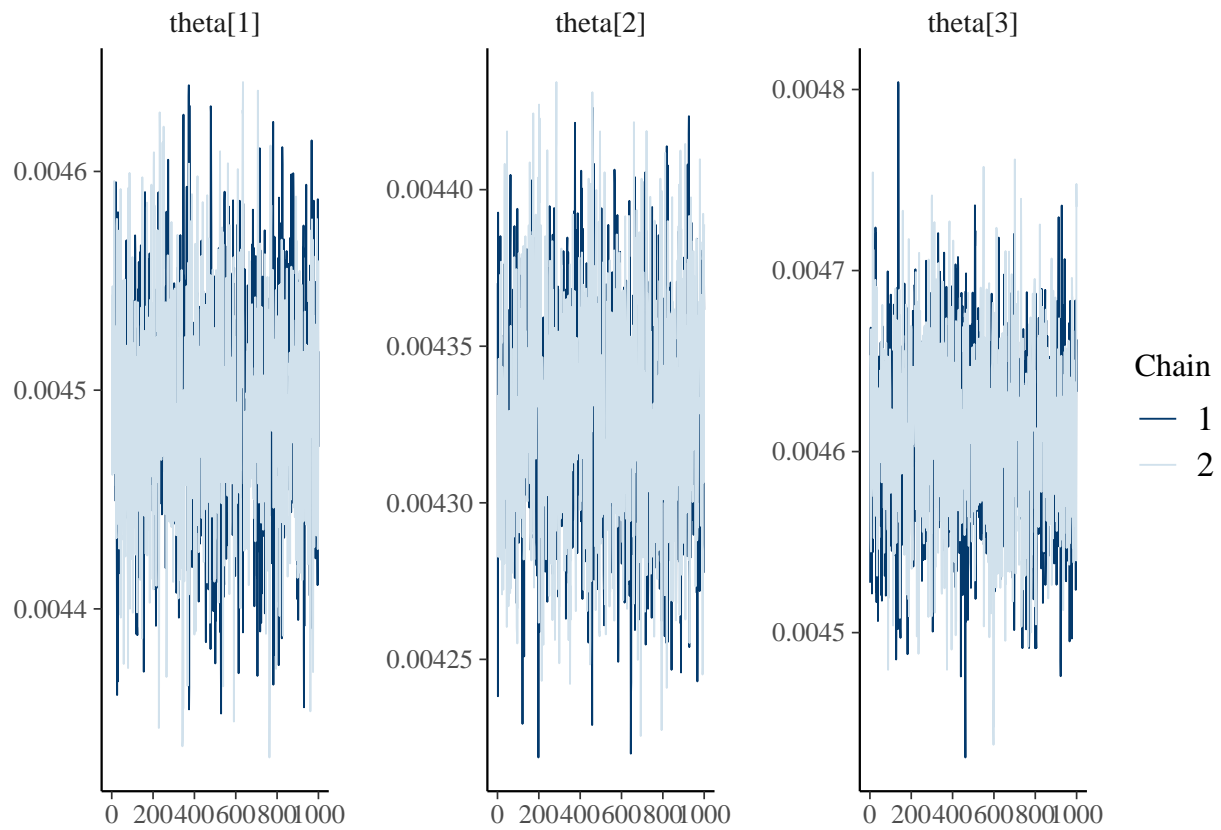
rand_ethnic_data <- list(theta_a = theta_a,
                        theta_b = theta_b,
                        nObs = aut_prev_ethnic_adj$sum_sample_pop_size,
                        aut_sample = aut_prev_ethnic_adj$adjusted_count,
                        nEthnic = nEthnic)

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

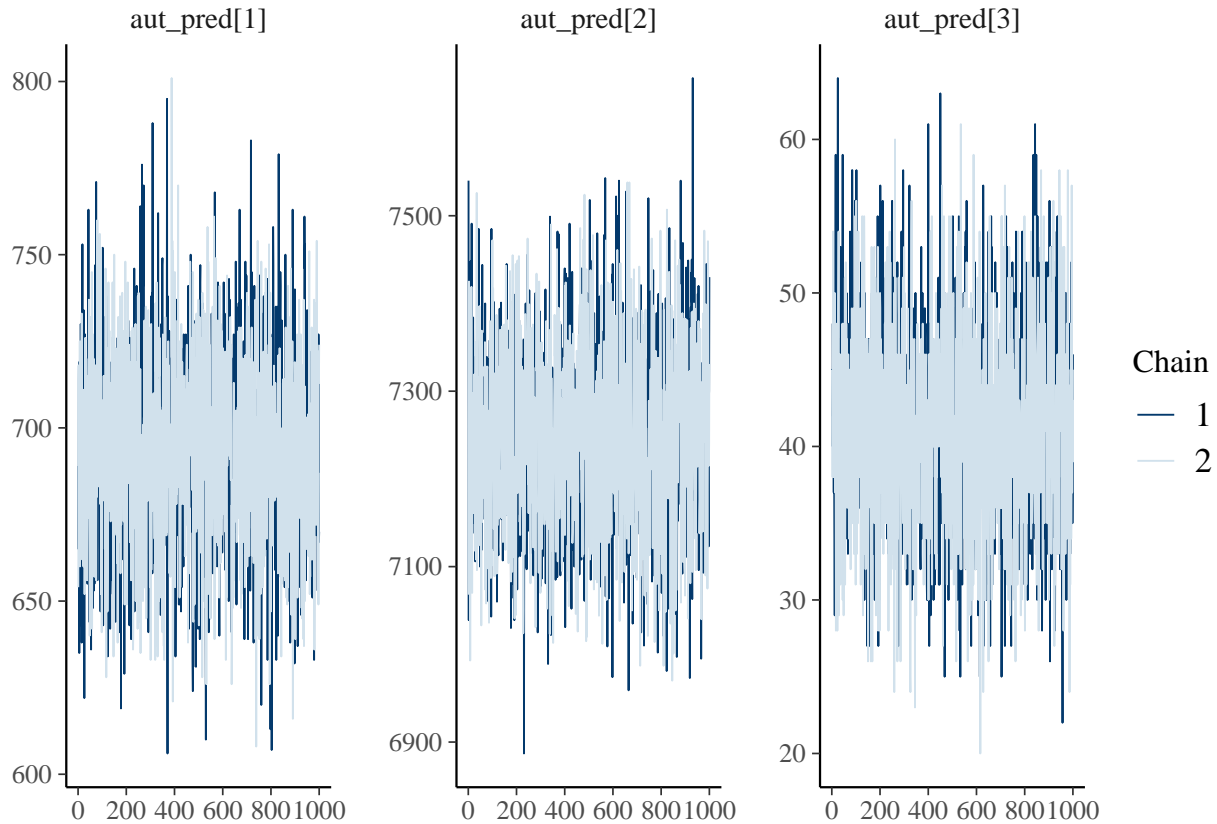
# Run JAGS model and discard burn-in samples
rand_ethnic_jag <- jags.model(textConnection(rand_ethnic_model),
                             data = rand_ethnic_data,
                             #inits = rand_region_ini,
                             n.chains = 2,
                             quiet = TRUE)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

# Check for convergence in parameters of interest
#mcmc_trace(rand_region_sam, 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
```



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

```
## # A tibble: 11 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] 6.93e+2 6.93e+2 2.77e+1 2.82e+1 6.48e+2 7.38e+2 1.00 1863.
## 2 aut_pred[2] 7.25e+3 7.25e+3 1.04e+2 1.07e+2 7.08e+3 7.42e+3 0.999 1653.
## 3 aut_pred[3] 4.12e+1 4.1 e+1 6.61e+0 7.41e+0 3.1 e+1 5.3 e+1 1.00 1849.
## 4 aut_sample[1] 4.99e+2 4.99e+2 0      0      4.99e+2 4.99e+2 NA      NA
## 5 aut_sample[2] 6.77e+3 6.77e+3 0      0      6.77e+3 6.77e+3 NA      NA
## 6 aut_sample[3] 5.9 e+1 5.9 e+1 0      0      5.9 e+1 5.9 e+1 NA      NA
## 7 theta[1] 4.49e-3 4.49e-3 5.00e-5 4.90e-5 4.41e-3 4.57e-3 1.00 1252.
## 8 theta[2] 4.33e-3 4.33e-3 3.60e-5 3.72e-5 4.27e-3 4.39e-3 1.00 1106.
## 9 theta[3] 4.61e-3 4.61e-3 4.93e-5 5.09e-5 4.53e-3 4.69e-3 1.00 1320.
## 10 theta_a 8.09e+3 8.09e+3 0      0      8.09e+3 8.09e+3 NA      NA
## 11 theta_b 1.75e+6 1.75e+6 0      0      1.75e+6 1.75e+6 NA      NA
## # ... with 1 more variable: ess_tail <dbl>, and abbreviated variable name
## #   1: ess_bulk
```

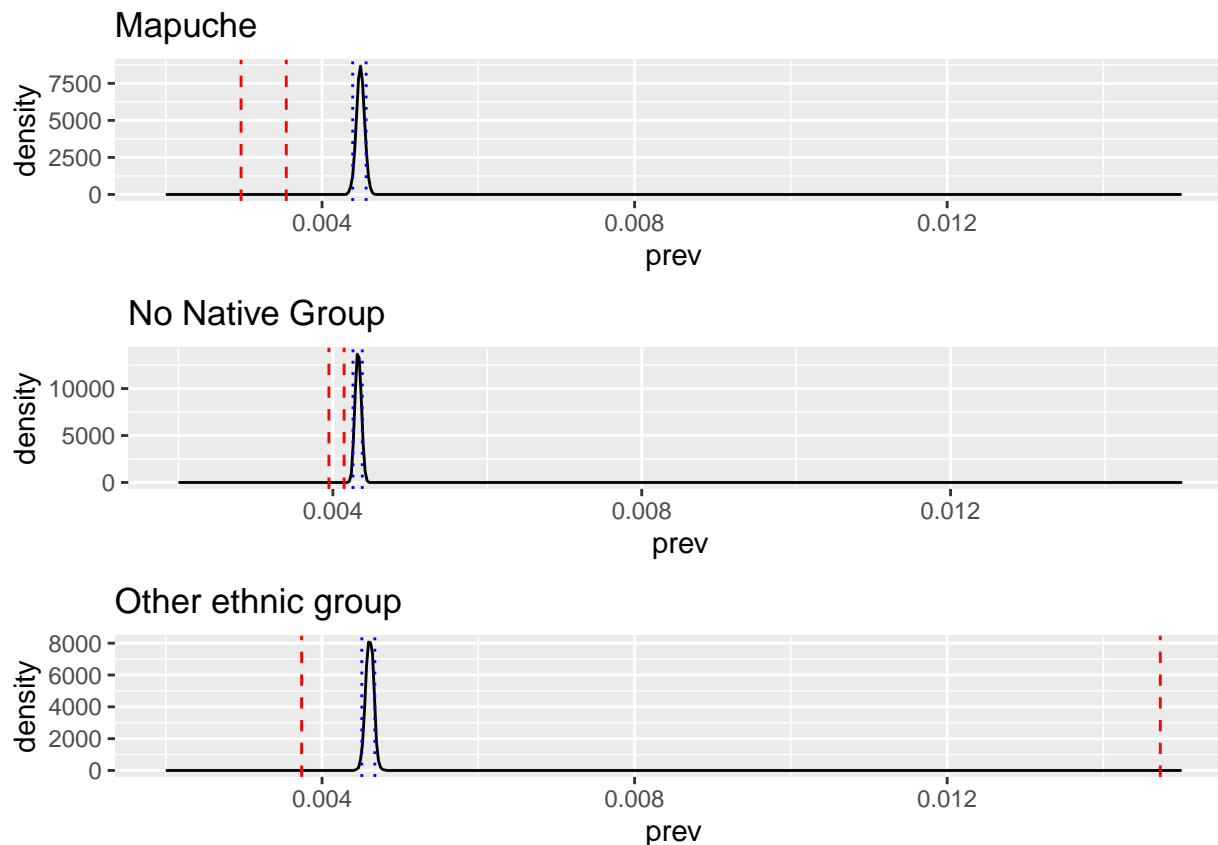
```
rand_ethnic_summ <- summary(subset_draws(as_draws(rand_ethnic_sam), pars),
  ~quantile(.x, probs=c(0.025, 0.5, 0.975)),
  ~mcse_quantile(.x, probs=c(0.025, 0.5, 0.975)),
  "rhat") %>%
  arrange(desc(mcse_q50))
rand_ethnic_summ
```

```
## # 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]      7050.        7.25e+3 7.45e+3 4      e+0 3.5 e+0 8.5 e+0 0.999
## 2 aut_pred[1]       639         6.93e+2 7.46e+2 1.5 e+0 5      e-1 2      e+0 1.00
## 3 theta[1]          0.00439 4.49e-3 4.59e-3 4.28e-6 1.86e-6 3.19e-6 1.00
## 4 theta[2]          0.00426 4.33e-3 4.40e-3 2.33e-6 1.52e-6 3.99e-6 1.00
## 5 theta[3]          0.00451 4.61e-3 4.70e-3 6.32e-6 1.44e-6 5.14e-6 1.00
## 6 aut_pred[3]       29          4.1 e+1 5.5 e+1 1      e+0 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 = pars,
```

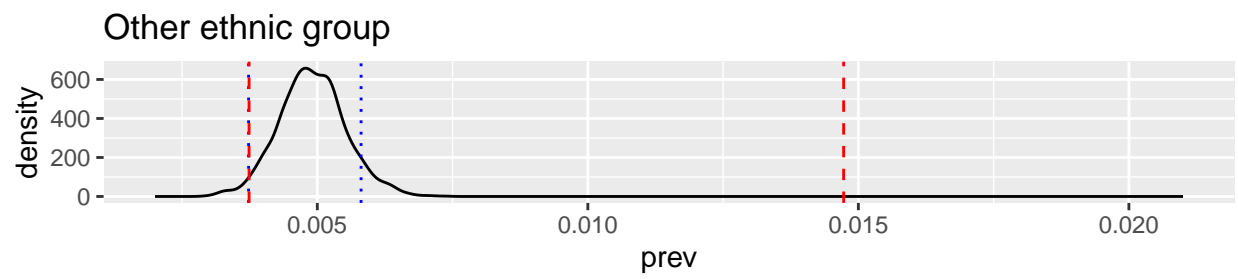
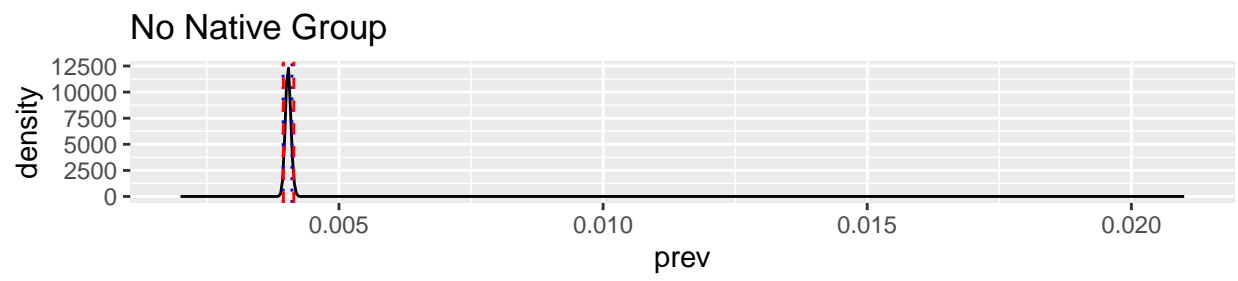
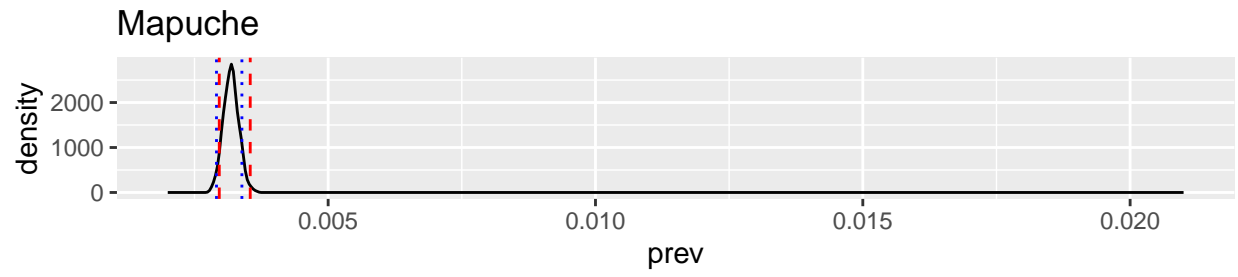
```

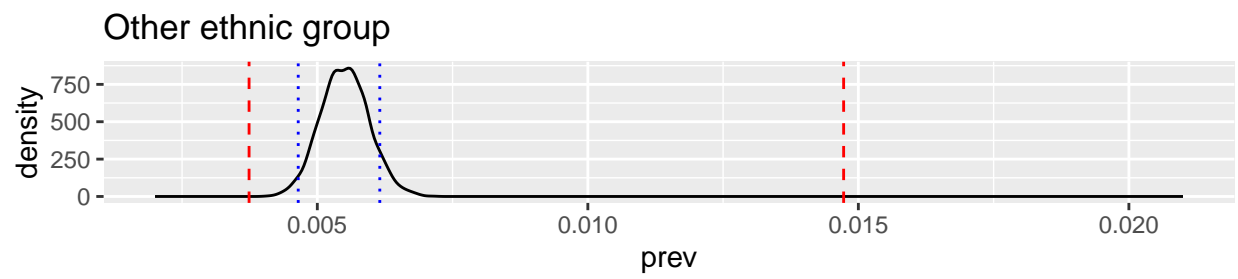
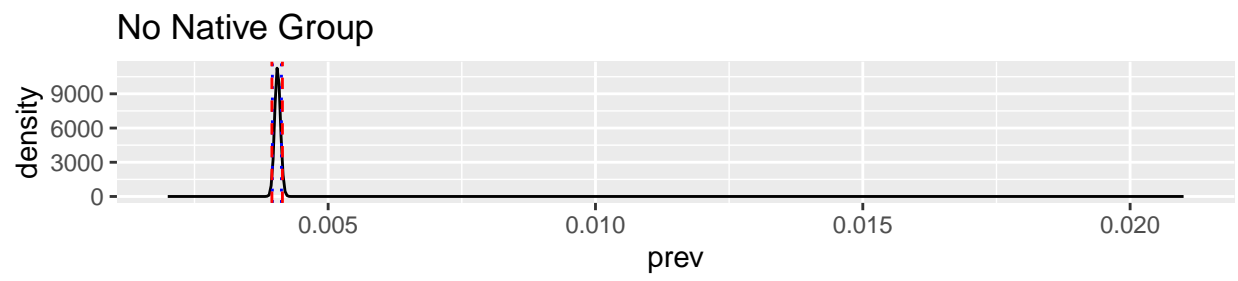
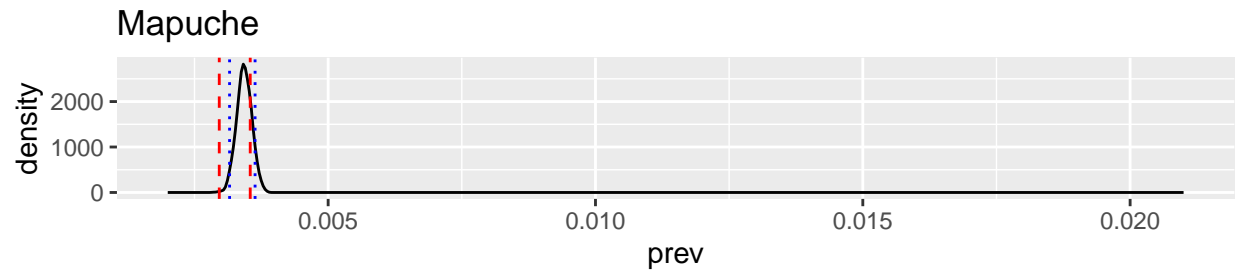
n.iter = nIter)

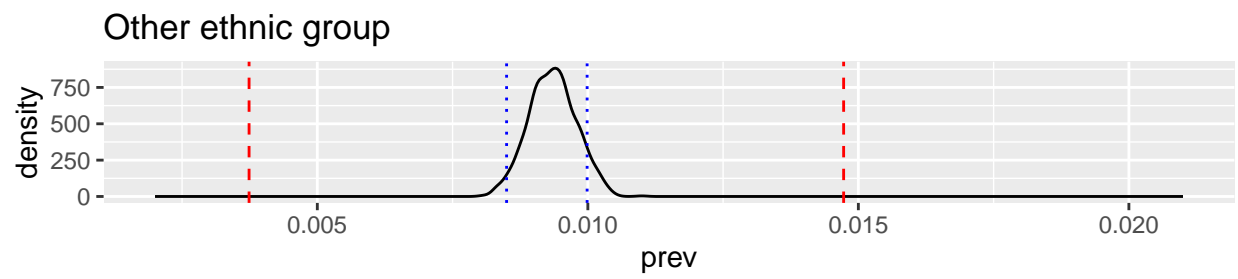
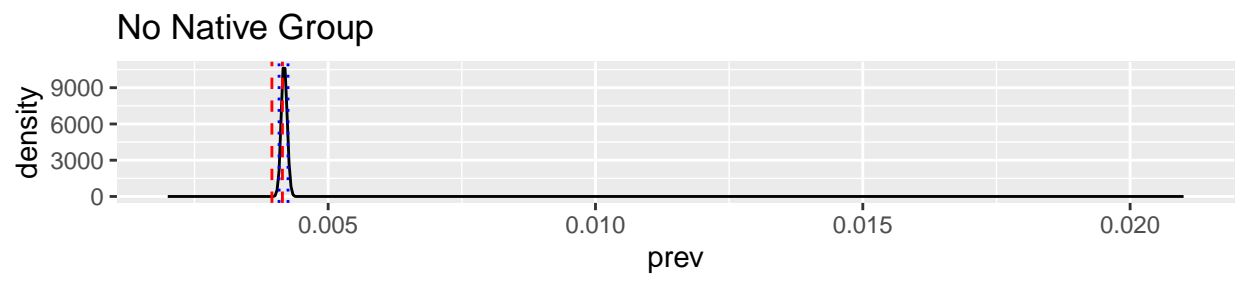
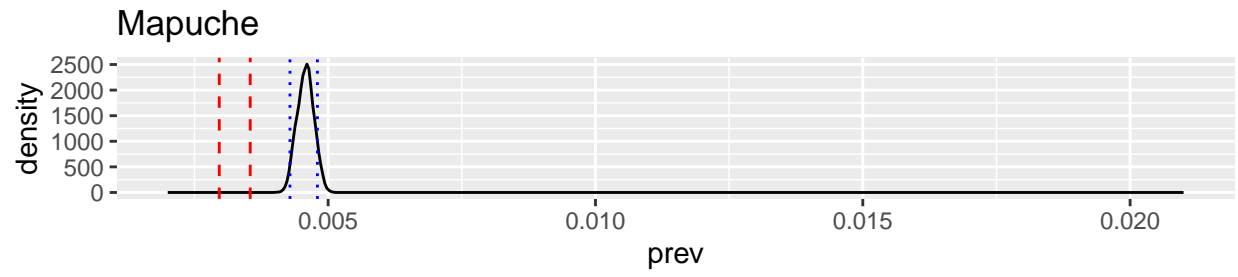
# Plots
aut_prev_ethnic_plots <- list()
ethnic_post_ci_lower <- list()
ethnic_post_ci_upper <- list()

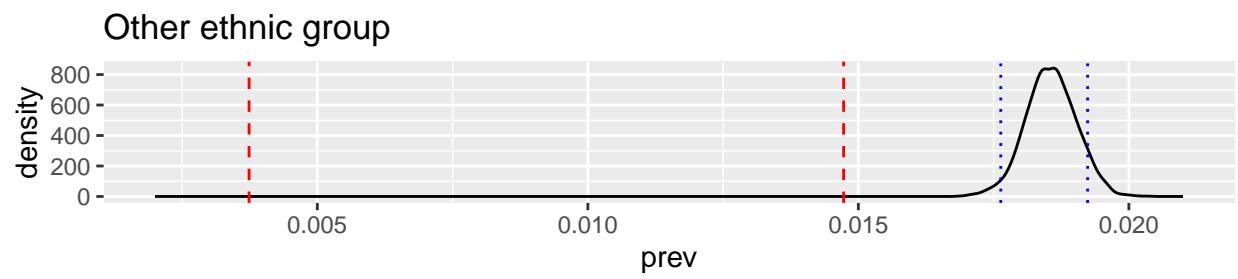
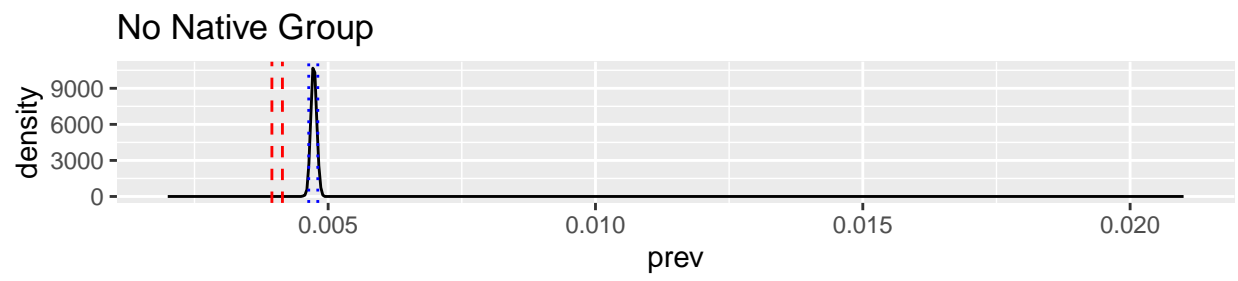
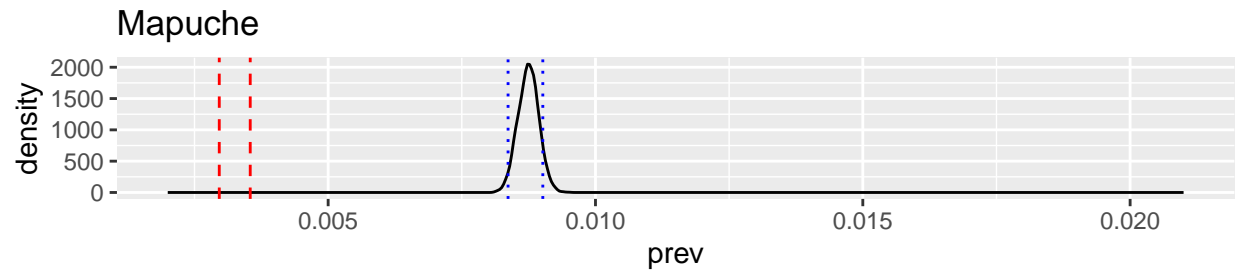
for(i in 1:nEthnic) {
  prevs <- data.frame(prev = extract_variable(rand_ethnic_sam, paste0("theta[", i, "]")))
  ethnic_post_ci_lower[[i]] <- quantile(prevs$prev, 0.025)
  ethnic_post_ci_upper[[i]] <- quantile(prevs$prev, 0.925)
  density_plot <- ggplot(prevs, aes(x = prev)) +
    geom_density() +
    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
}

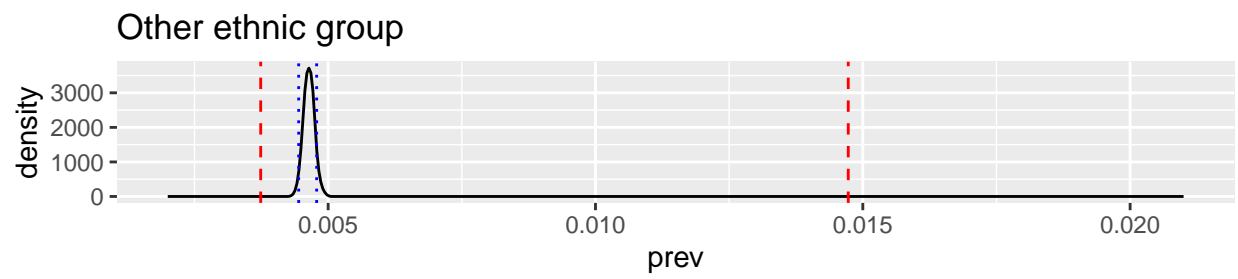
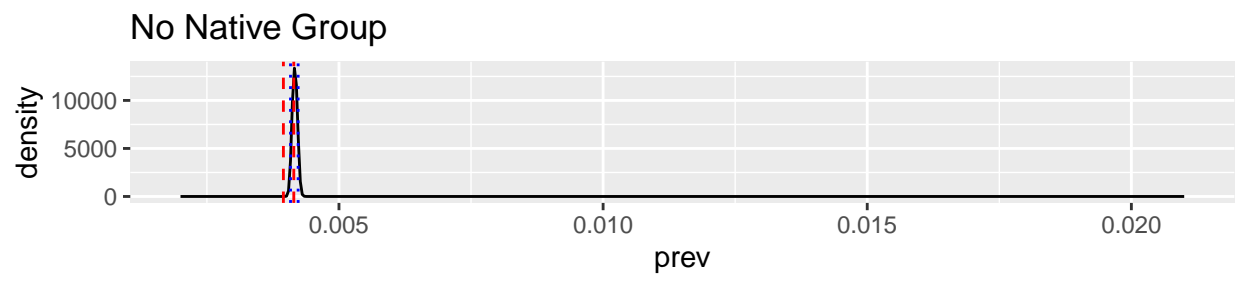
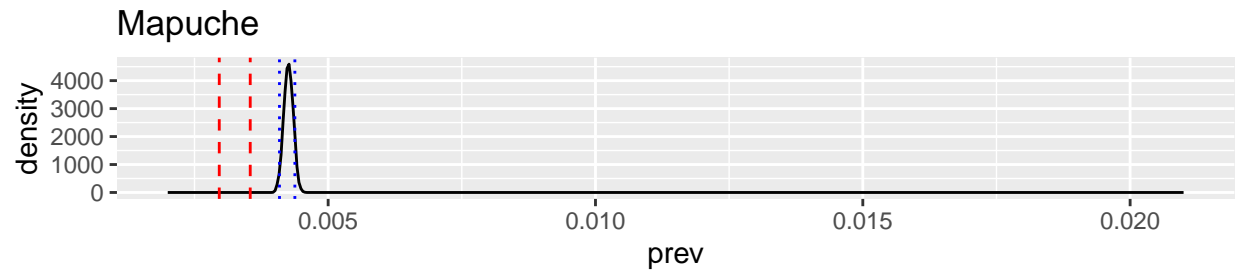
```

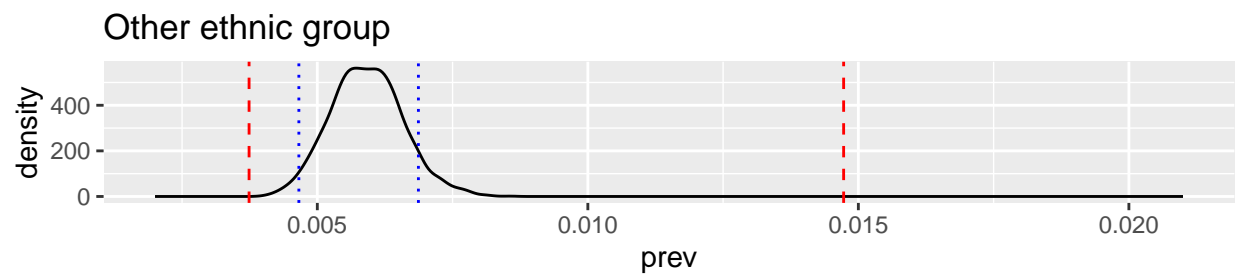
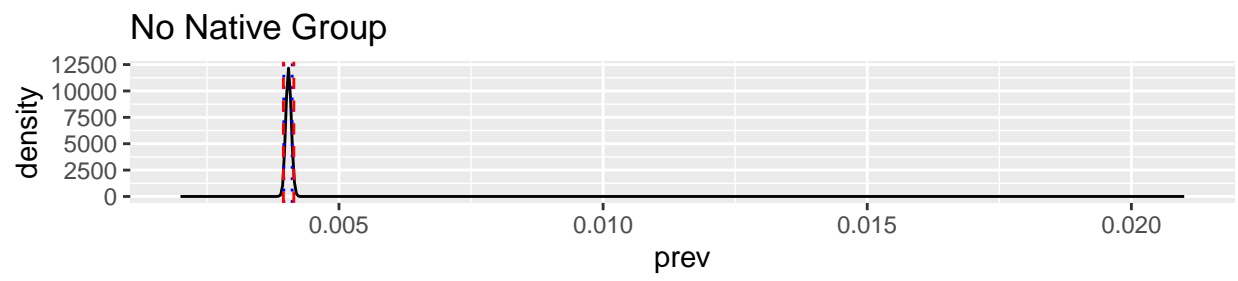
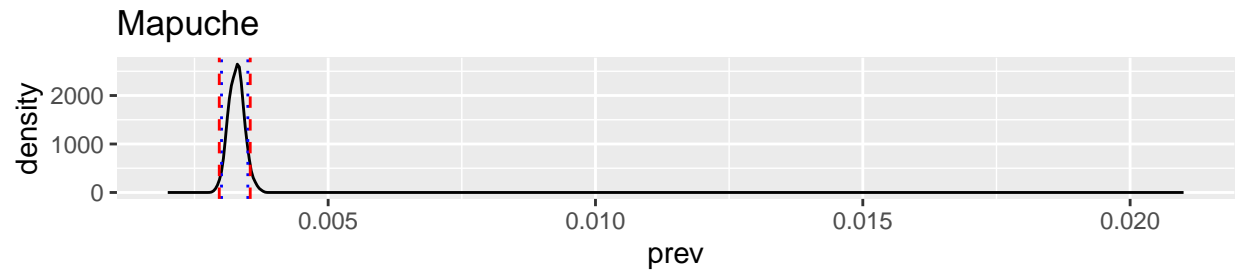



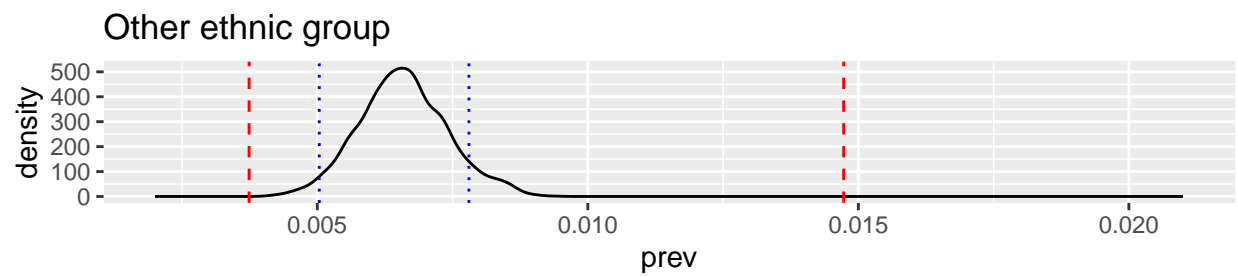
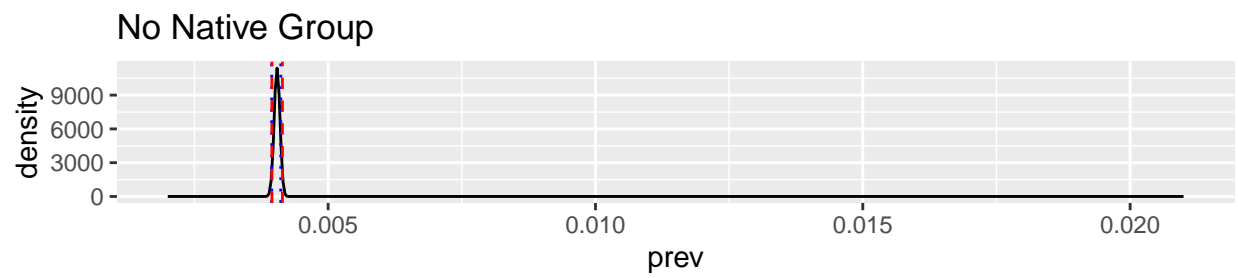
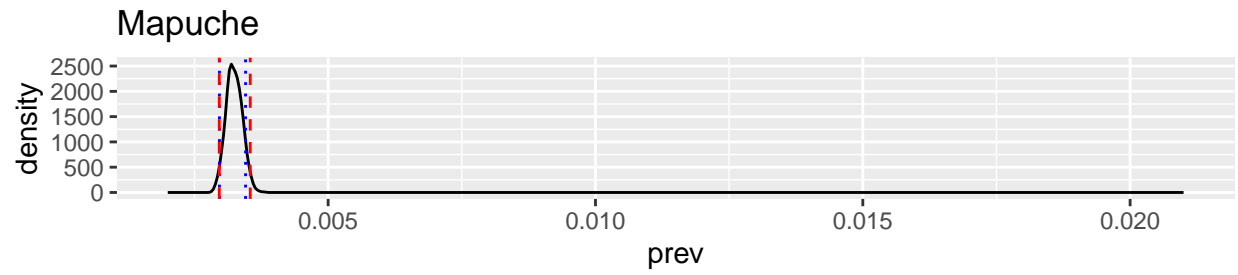


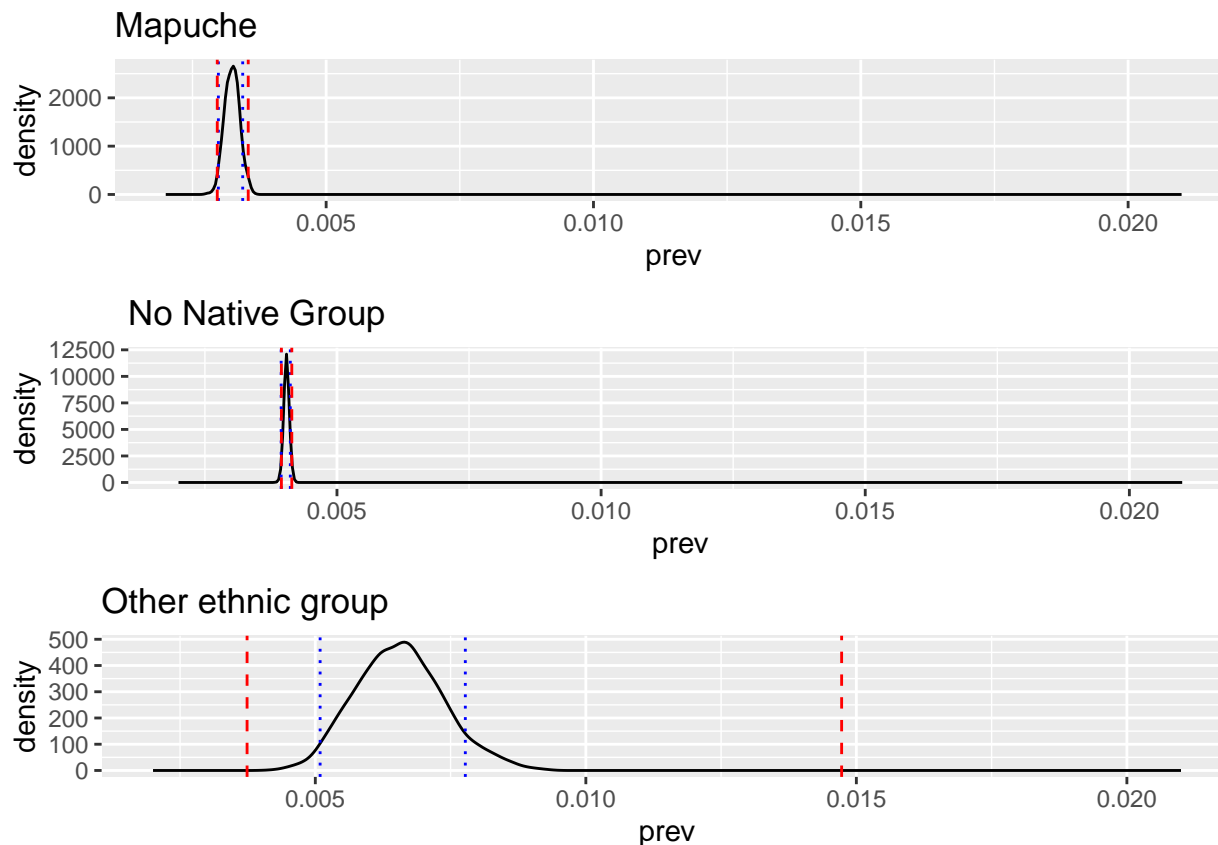












Bayesian prevalence by economic status

```
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", "$25.001 A $100.000",
                                                            ifelse(school_fee == "MAS DE $100.000", "High", "No information"))),
                                   "Other"))
  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 1
  #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*(adjusted_rate + w_M)),
  #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])
  }
}"

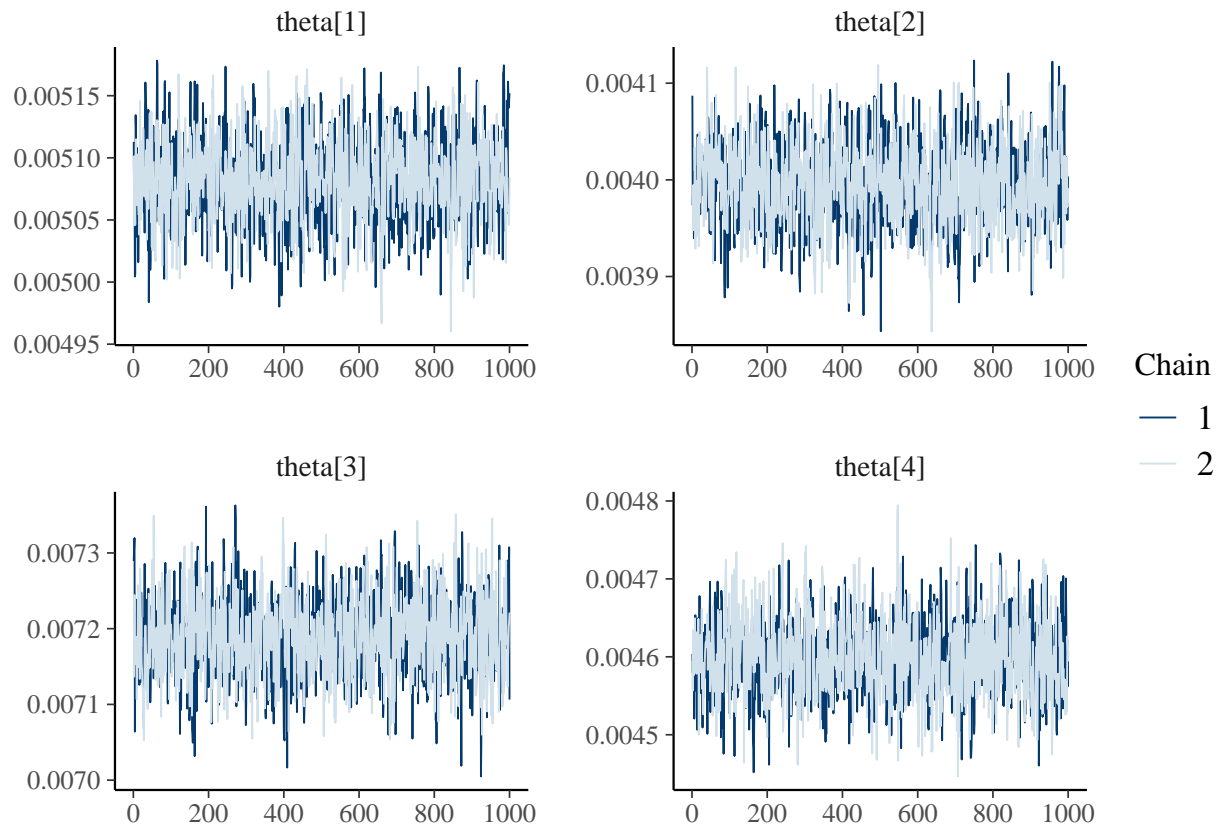
rand_econ_data <- list(theta_a = theta_a,
  theta_b = theta_b,
  nObs = aut_prev_econ_adj$sum_sample_pop_size,
  aut_sample = aut_prev_econ_adj$adjusted_count,
  nEcon = nEcon)

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))

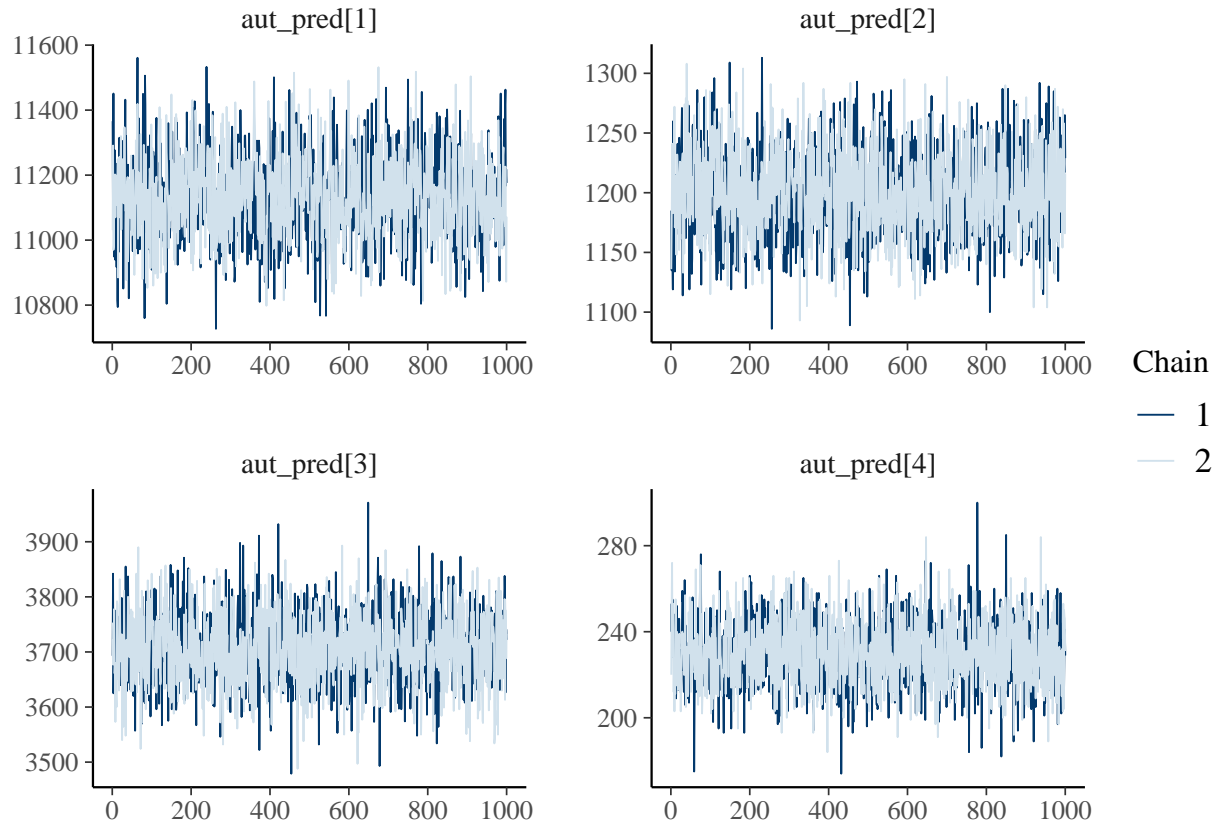
# Run JAGS model and discard burn-in samples
rand_econ_jag <- jags.model(textConnection(rand_econ_model),
  data = rand_econ_data,
  inits = rand_econ_ini,
  n.chains = 2,
  quiet = TRUE)
update(rand_econ_jag, n.iter = nBurn)
rand_econ_sam <- coda.samples(model = rand_econ_jag,
  variable.names = pars,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

```

```
# Check for convergence in parameters of interest
#mcmc_trace(rand_region_sam, 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
```



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

```
## # A tibble: 14 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] 1.11e+4 1.11e+4 1.28e+2 1.32e+2 1.09e+4 1.14e+4 1.00 1391.
## 2 aut_pred[2] 1.20e+3 1.20e+3 3.58e+1 3.56e+1 1.14e+3 1.26e+3 1.00 1812.
## 3 aut_pred[3] 3.71e+3 3.70e+3 6.51e+1 6.38e+1 3.6 e+3 3.82e+3 1.00 1849.
## 4 aut_pred[4] 2.30e+2 2.3 e+2 1.54e+1 1.63e+1 2.06e+2 2.55e+2 0.999 2096.
## 5 aut_sample[1] 1.20e+4 1.20e+4 0      0      1.20e+4 1.20e+4 NA      NA
## 6 aut_sample[2] 1.39e+2 1.39e+2 0      0      1.39e+2 1.39e+2 NA      NA
## 7 aut_sample[3] 8.27e+3 8.27e+3 0      0      8.27e+3 8.27e+3 NA      NA
## 8 aut_sample[4] 2.29e+2 2.29e+2 0      0      2.29e+2 2.29e+2 NA      NA
## 9 theta[1]      5.08e-3 5.08e-3 3.53e-5 3.50e-5 5.02e-3 5.14e-3 1.00 1186.
## 10 theta[2]     4.00e-3 4.00e-3 4.35e-5 4.33e-5 3.92e-3 4.07e-3 1.00 1149.
## 11 theta[3]     7.19e-3 7.19e-3 5.44e-5 5.42e-5 7.10e-3 7.28e-3 1.00 1282.
## 12 theta[4]     4.60e-3 4.60e-3 5.03e-5 4.95e-5 4.52e-3 4.69e-3 1.00 1116.
## 13 theta_a      8.09e+3 8.09e+3 0      0      8.09e+3 8.09e+3 NA      NA
## 14 theta_b      1.75e+6 1.75e+6 0      0      1.75e+6 1.75e+6 NA      NA
## # ... with 1 more variable: ess_tail <dbl>, and abbreviated variable name
## #   1: ess_bulk
```

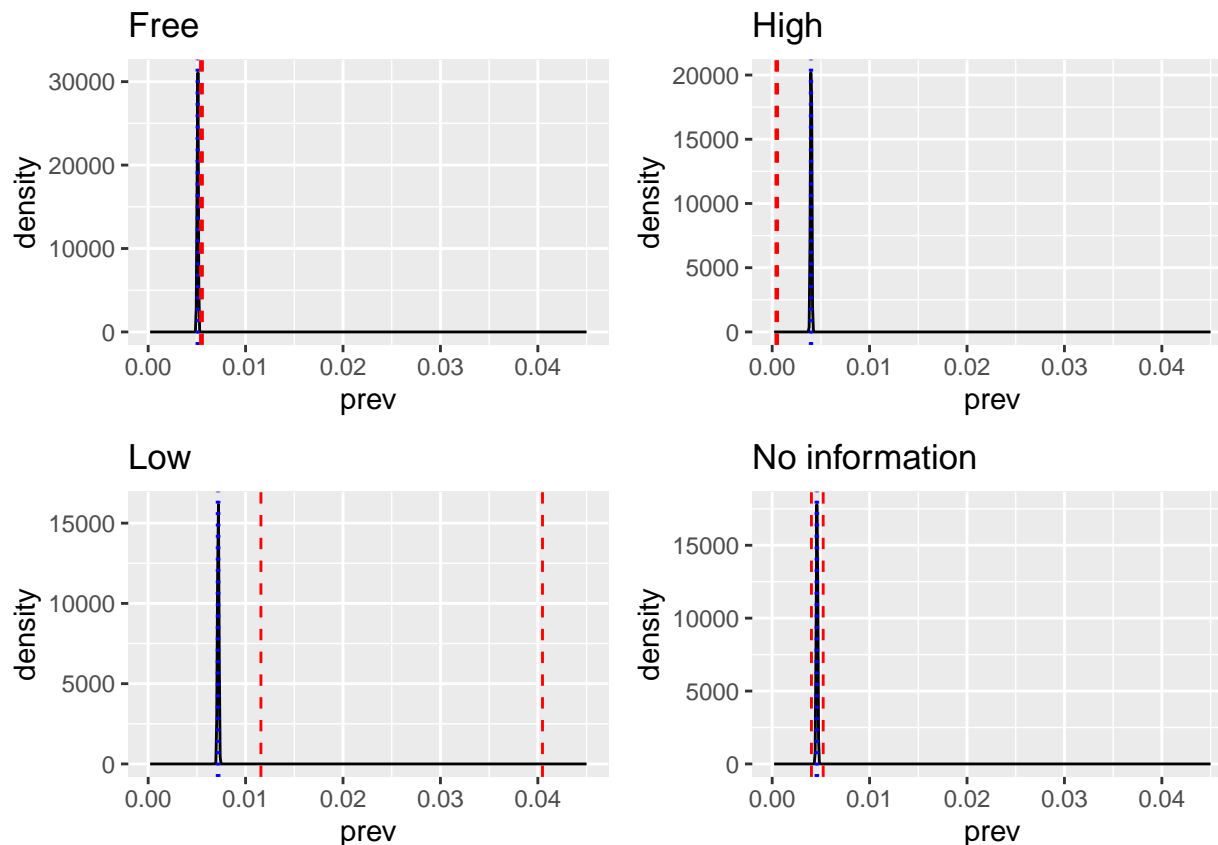
```
rand_econ_summ <- summary(subset_draws(as_draws(rand_econ_sam), pars),
  ~quantile(.x, probs=c(0.025, 0.5, 0.975)),
  ~mcse_quantile(.x, probs=c(0.025, 0.5, 0.975)),
  "rhat") %>%
  arrange(desc(mcse_q50))
```

```
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]    10883.    1.11e+4  1.14e+4    9 e+0    4 e+0    8.5 e+0  1.00
## 2 aut_pred[3]     3577.    3.70e+3  3.83e+3    5 e+0    2 e+0    3.5 e+0  1.00
## 3 aut_pred[2]     1133.    1.20e+3  1.27e+3    1.5 e+0  1 e+0    1.5 e+0  1.00
## 4 aut_pred[4]      201      2.3 e+2  2.6 e+2    1 e+0    5 e-1    1 e+0    0.999
## 5 theta[3]         0.00708  7.19e-3  7.30e-3   4.07e-6  1.72e-6  6.40e-6  1.00
## 6 theta[4]         0.00450  4.60e-3  4.70e-3   3.61e-6  1.70e-6  4.34e-6  1.00
## 7 theta[2]         0.00391  4.00e-3  4.08e-3   3.69e-6  1.32e-6  3.74e-6  1.00
## 8 theta[1]         0.00501  5.08e-3  5.15e-3   2.04e-6  1.07e-6  2.30e-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)
```

```

update(rand_econ_jag, n.iter = nBurn)
rand_econ_sam <- coda.samples(model = rand_econ_jag,
                             variable.names = 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)
}

```