

Chile_prev_rmd

Adele Tyson

2023-06-08

```
#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)
library(poolr)
library(epitools)
library(corrplot)
library(caret)
library(mltools)
library(ggrepel)
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.  
                               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_stpop$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 confidence intervals
  # 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 integer counts
            #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(across(all_of(grouping_vars)))
  }

do_jags_rand_model <- function(x, feat, model, theta_mu, theta_sigma, pars, nBurn = 1000, nIter = 1000,
  # x = output from get_adjusted_prev. Needs to have columns sum_sample_pop_size, adjusted_count
  # feat = feature being used as random effect
  # model = JAGS random effects model
  # theta_mu, theta_sigma = mean and sd of beta prior distribution
  # pars = model parameters to report
  # nBurn = number of burn-in samples
  # nIter = number of posterior iterations

  nFeat <- length(unique(x[[feat]]))
  FeatNames <- sort(unique(x[[feat]]))

  # Define beta prior
  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)

  # Initial values for model chains
  rand_ini <- list(list(theta = rep(0.001, nFeat)), #, spec = 0.5, sens = 0.5),
                  list(theta = rep(0.01, nFeat))) #, spec = 0.9, sens = 0.9))

  # Run JAGS model
  rand_data <- list(theta_a = theta_a,
                   theta_b = theta_b,
                   nObs = x$sum_sample_pop_size,
                   aut_sample = x$adjusted_count,
                   nFeat = nFeat)
  rand_jag <- jags.model(textConnection(model),
                        data = rand_data,
                        inits = rand_ini,
                        n.chains = 2,

```

```

      quiet = TRUE)
update(rand_jag, n.iter = nBurn)
rand_sam <- coda.samples(model = rand_jag,
      variable.names = pars,
      n.iter = nIter)

# Convergence checks
if(convergence_checks) {
  print(mcmc_trace(rand_sam, paste0("theta[", 1:nFeat, "]"))) # Convergence looks fine and rhats <= 1
  print(mcmc_trace(rand_sam, paste0("aut_pred[", 1:nFeat, "]"))) # Convergence looks fine and rhats <= 1
  rand_summ <- summary(subset_draws(as_draws(rand_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(rhat))
  print(rand_summ)
}

# Extract posterior density
prev_post <- as_tibble(as_draws_matrix(rand_sam, rownames = "Iteration") %>%
  select(c("Iteration", contains("theta["))) %>%
  pivot_longer(cols = contains("theta["),
    names_to = "Feat",
    values_to = "predicted_prev") %>%
  mutate(Feat_names = factor(Feat, levels = c(paste0("theta[", 1:nFeat, "]")), labels = FeatNames)) %>%
  select(Iteration, Feat_names, predicted_prev)

return(prev_post)
}

plot_post_density <- function(jags_post, sample_data, feat, theta_mu, theta_sigma) {
  # Plots posterior densities and their 95% credible intervals, and sample prevalence confidence intervals
  # jags_post = output from do_jags_rand_model, ie posterior densities
  # sample_data = output from get_adjusted_prev, ie sample prevalences with confidence intervals
  # feat = the same feature used as the random effect in do_jags_rand_model
  # theta_mu, theta_sigma = mean and sd of beta prior distribution used in do_jags_rand_model

  # calculate posterior credible intervals
  post_ci <- jags_post %>%
  group_by(across(all_of(feat))) %>%
  summarise(post_lower = quantile(predicted_prev, 0.025),
    post_upper = quantile(predicted_prev, 0.975))

  print(ggplot() +
    geom_density(data = jags_post, aes(x = predicted_prev)) +
    geom_vline(data = post_ci, aes(xintercept = post_lower), color = "blue", linetype = "dotted") +
    geom_vline(data = post_ci, aes(xintercept = post_upper), color = "blue", linetype = "dotted") +
    geom_vline(data = sample_data, aes(xintercept = ci_lower), color = "red", linetype = "dashed") +
    geom_vline(data = sample_data, aes(xintercept = ci_upper), color = "red", linetype = "dashed") +
    facet_wrap(as.formula(paste0("~", feat))) +
    labs(title = paste0("Prior mean = ", theta_mu, ", prior sd = ", signif(theta_sigma, 3))))
}

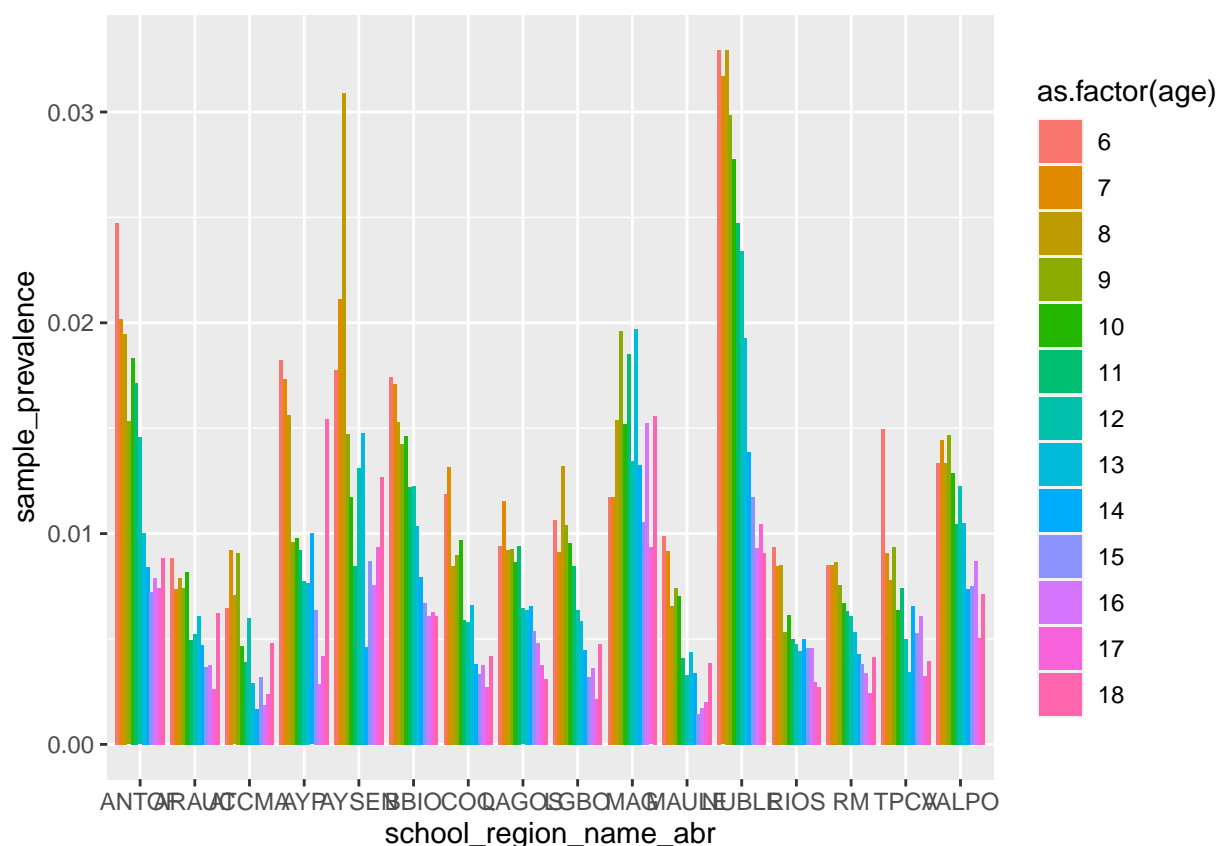
```

Explore data

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

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

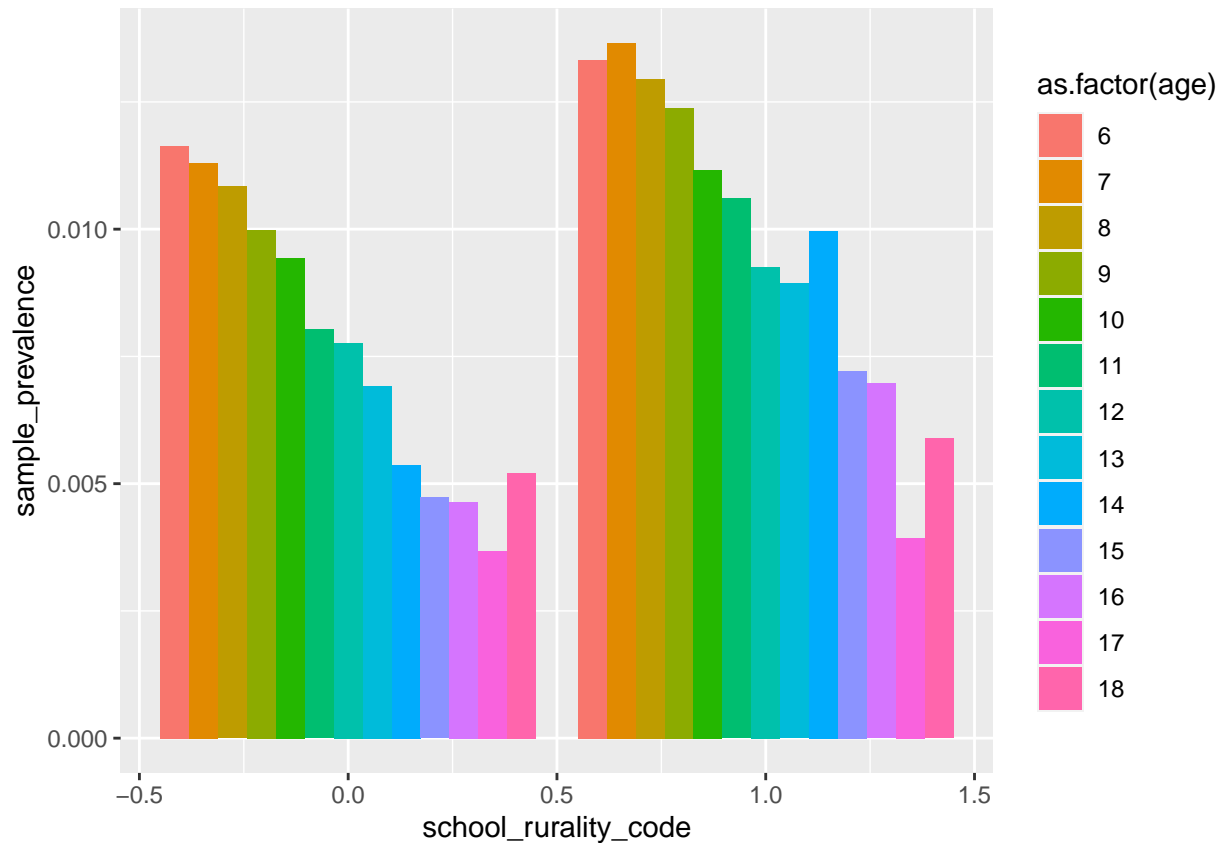


```
#geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = sex, fill = as.factor(sex)),
# 1 is male, 2 is female
```

```
aut_prev_rural <- get_grouped_prev(x = chile_bayes_aut, stdpop = chile_stdpop,
                                   grouping_vars = c("school_rurality_code", "age_june30", "sex", "aut_prev"))
```

```
## `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")
```

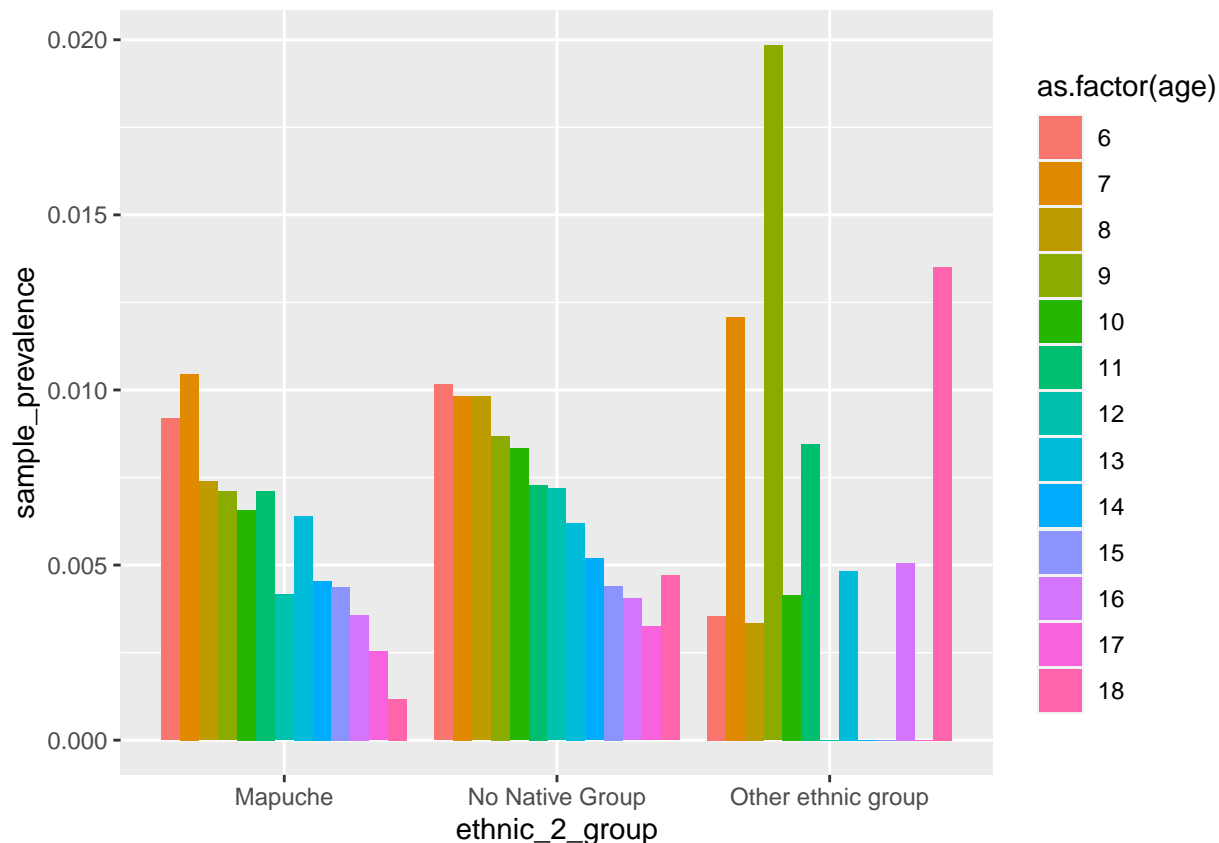


```
#geom_col(aes(x = school_region_name_abr, y = prevalence, group = sex, fill = as.factor(sex)), position = "stack", # 1 is male, 2 is female)
```

```
aut_prev_ethnic <- chile_bayes_aut %>%
  filter(school_region_name_abr %in% c("ARAUC", "BBIO", "LAGOS", "RIOS", "RM")) %>%
  get_grouped_prev(stdpop = chile_stdpop,
    grouping_vars = c("ethnic_2_group", "age_june30", "sex", "autism"))
```

```
## `summarise()` has grouped output by 'ethnic_2_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 = "stack")
```



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

Do this plot with 3 year age bands. Do some more plots

Bayesian prevalence analysis

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_extrapolate <- c(0.005, 0.01, 0.015, 0.3) # 0.5%, 1%, 1.5%, 3% prevalence
theta_sigma_extrapolate <- c(rep(0.0001/1.96, 4)) # Same as chosen prior
#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)

rand_model <- "model {
  for(i in 1:nFeat) { # For each category in the feature grouping
    theta[i] ~ dbeta(theta_a, theta_b)
    aut_sample[i] ~ dbin(theta[i], nObs[i])
  }
}
```



```

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

```

Common effects model with unadjusted 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 - 1)

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,

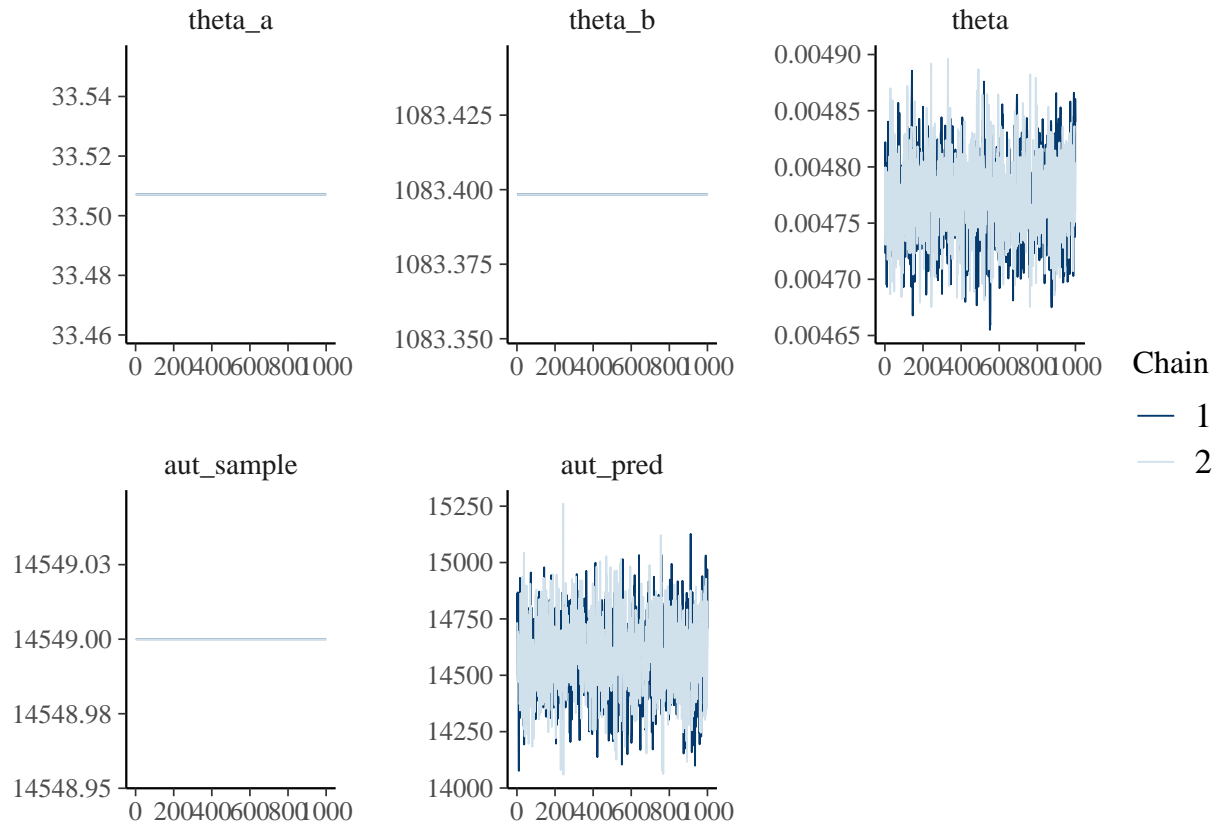
```

```

      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

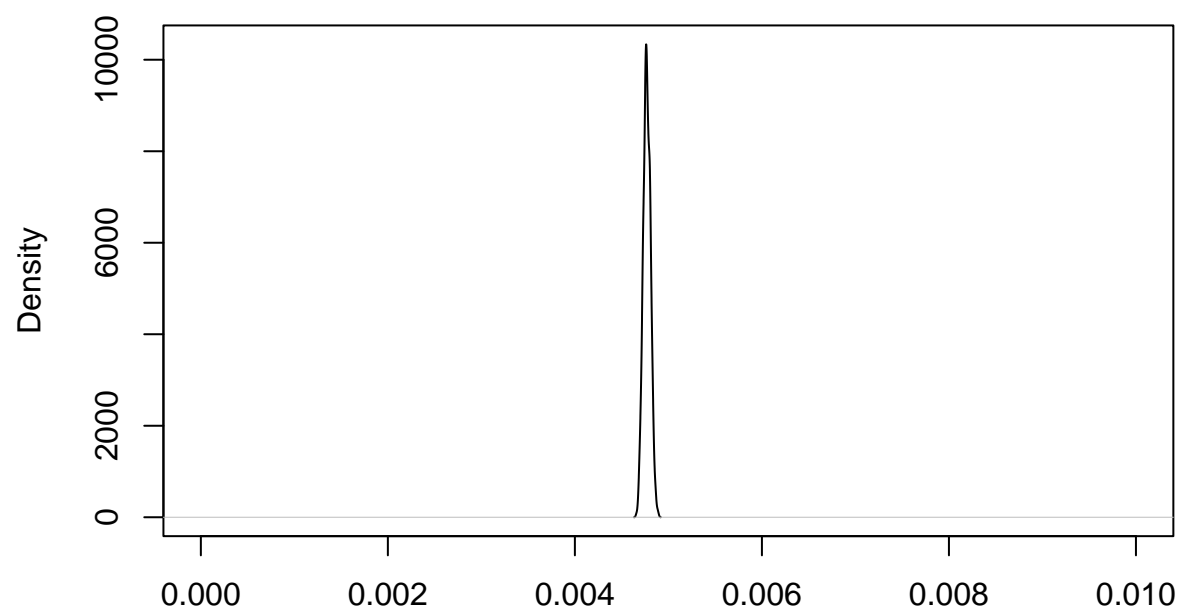
```

```

## # 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 1.71e+2 1.68e+2 1.43e+4 1.49e+4 1.00    1314.    1370.
## 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 3.85e-5 4.02e-5 4.71e-3 4.83e-3 1.00    1176.    1111.
## 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))

```

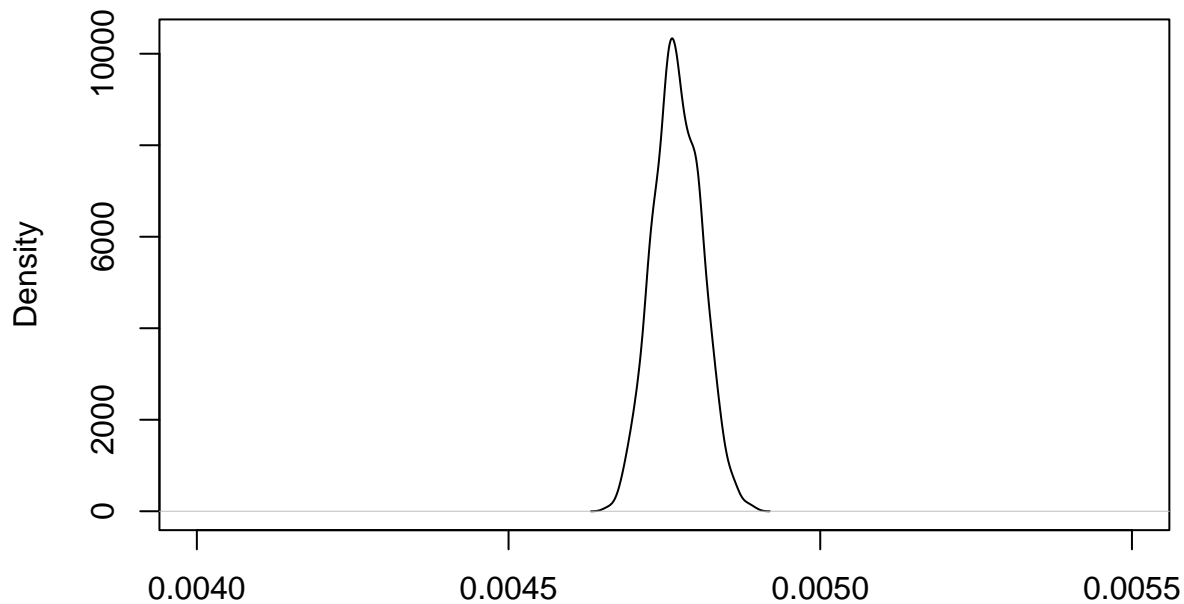
density.default(x = extract_variable(common_sam, "theta"))



N = 2000 Bandwidth = 7.579e-06

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

density.default(x = extract_variable(common_sam, "theta"))



N = 2000 Bandwidth = 7.579e-06

*# 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

Random effects analysis

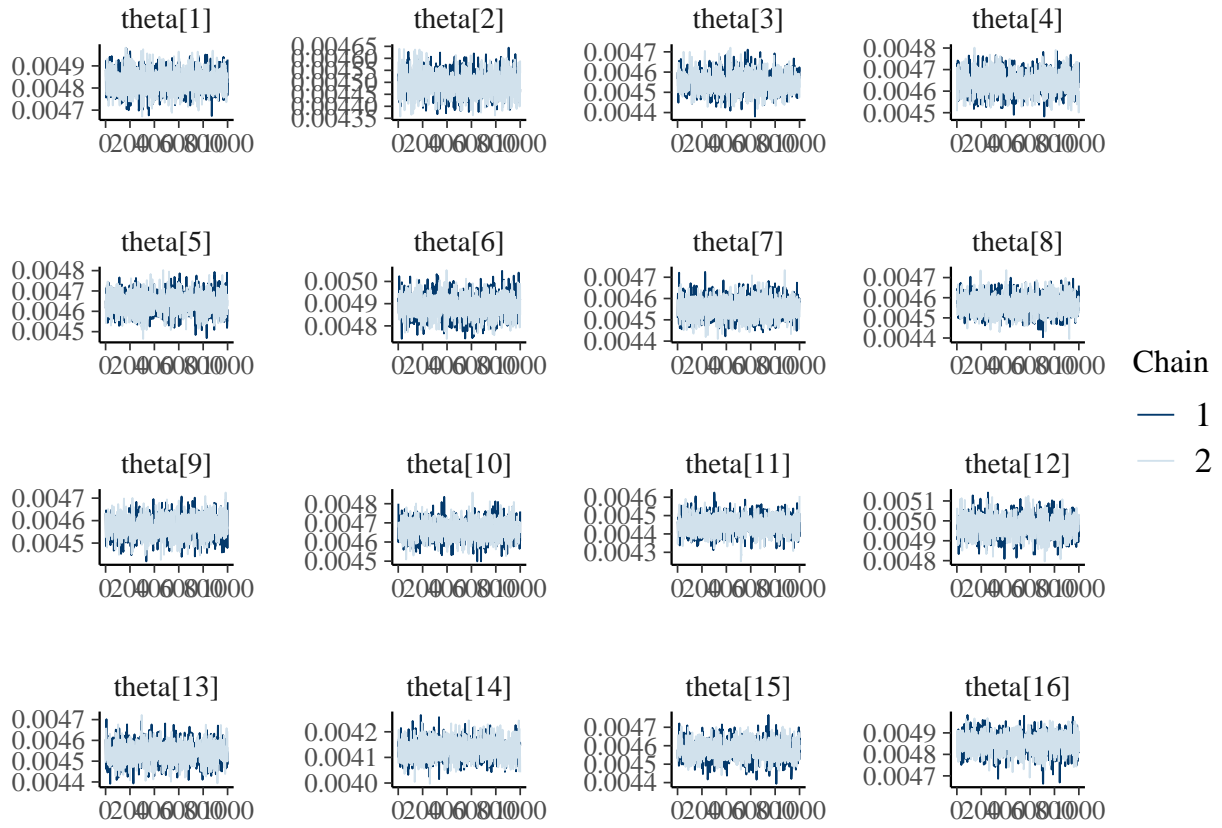
Standardise prevalence by Chile's age and sex based population sizes using https://seer.cancer.gov/seerstat/WebHelp/Rate_Algorithms.htm and <https://wonder.cdc.gov/wonder/help/cancer/fayfeuerconfidenceintervals.pdf>

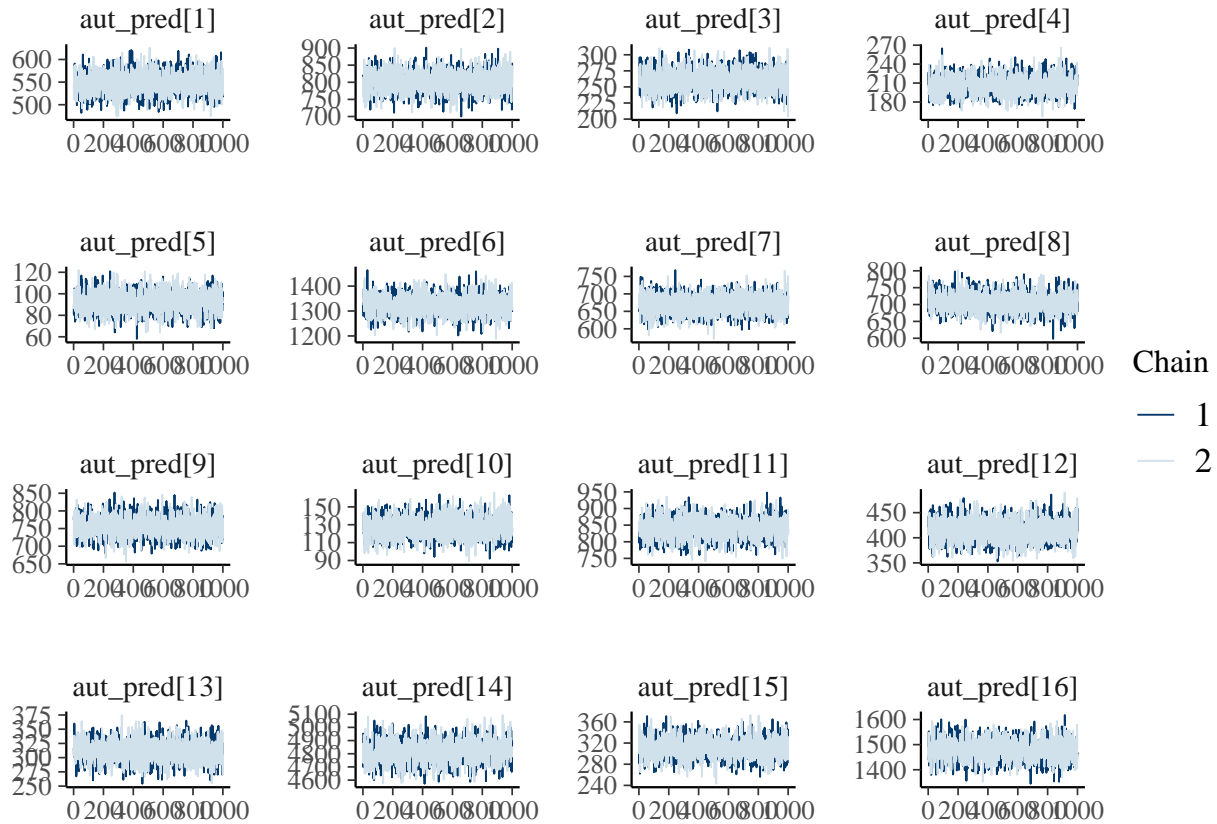
See https://github.com/Dpananos/bayes_multiple_measures/blob/master/analysis/sensitivity_analysis.R for more sensitivity analysis ideas

Random effect on region

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

aut_prev_region_post <- do_jags_rand_model(x = aut_prev_region_adj,
  feat = "school_region_name_abr",
  model = rand_model,
  theta_mu = theta_mu_prior,
  theta_sigma = theta_sigma_prior,
  pars = pars,
  convergence_checks = TRUE) %>%
  rename("school_region_name_abr" = "Feat_names")
```



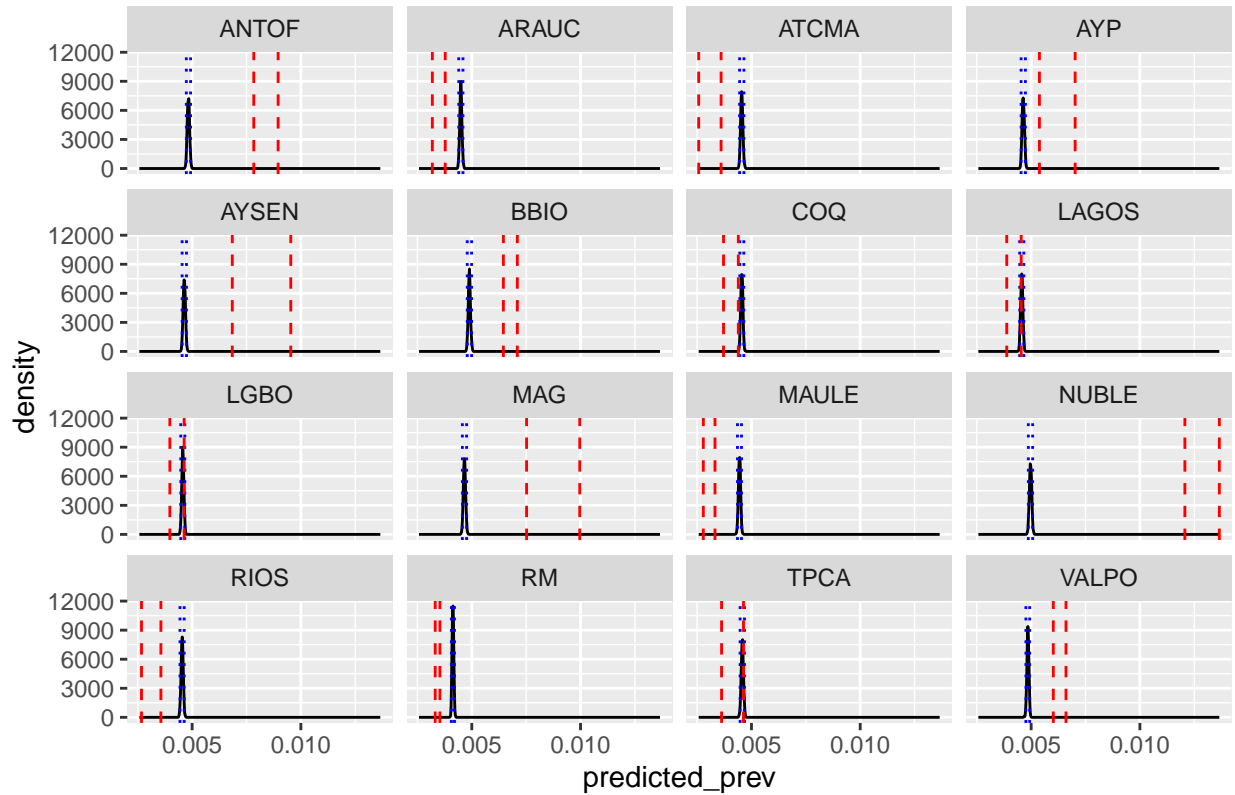


```
## # A tibble: 50 x 8
##   variable      `2.5%`      `50%`      `97.5%`  mcse_q2.5  mcse_q50  mcse_~1  rhat
##   <chr>         <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl> <dbl>
## 1 theta[11]    0.00436    0.00445    0.00454  0.00000420 0.00000169 3.91e-6 1.01
## 2 theta[13]    0.00444    0.00454    0.00463  0.00000467 0.00000171 3.04e-6 1.00
## 3 theta[15]    0.00448    0.00458    0.00468  0.00000666 0.00000124 3.55e-6 1.00
## 4 theta[16]    0.00476    0.00485    0.00493  0.00000430 0.00000177 1.83e-6 1.00
## 5 theta[10]    0.00457    0.00466    0.00476  0.00000535 0.00000178 2.87e-6 1.00
## 6 theta[6]     0.00479    0.00489    0.00499  0.00000389 0.00000132 4.34e-6 1.00
## 7 theta[2]     0.00440    0.00450    0.00459  0.00000372 0.00000147 3.76e-6 1.00
## 8 theta[9]     0.00448    0.00457    0.00466  0.00000483 0.00000144 3.12e-6 1.00
## 9 theta[3]     0.00445    0.00455    0.00465  0.00000340 0.00000185 5.28e-6 1.00
## 10 aut_pred[8] 656      707      764      1.5        1        3      e+0 1.00
## # ... with 40 more rows, and abbreviated variable name 1: mcse_q97.5
```

```
plot_post_density(aut_prev_region_post, aut_prev_region_adj, feat = "school_region_name_abr", theta_mu =
```

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.0046, prior sd = 5.1e-05



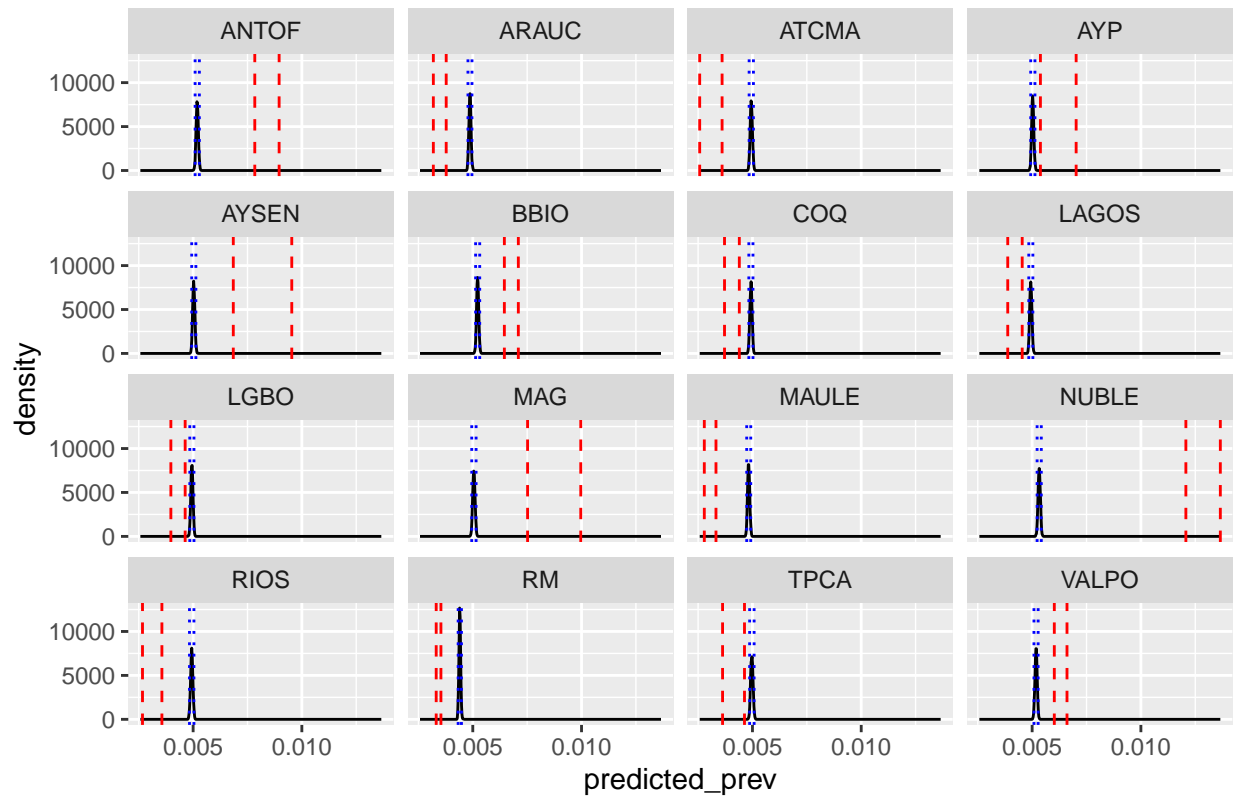
```
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_region_post <- do_jags_rand_model(x = aut_prev_region_adj,
                                             feat = "school_region_name_abr",
                                             model = rand_model,
                                             theta_mu = theta_mu_extrapolate[j],
                                             theta_sigma = theta_sigma_extrapolate[j],
                                             pars = pars,
                                             convergence_checks = FALSE) %>%
    rename("school_region_name_abr" = "Feat_names")

  plot_post_density(aut_prev_region_post,
                    aut_prev_region_adj,
                    feat = "school_region_name_abr",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}
```

Predictions for higher population prevalence - increase prior mean

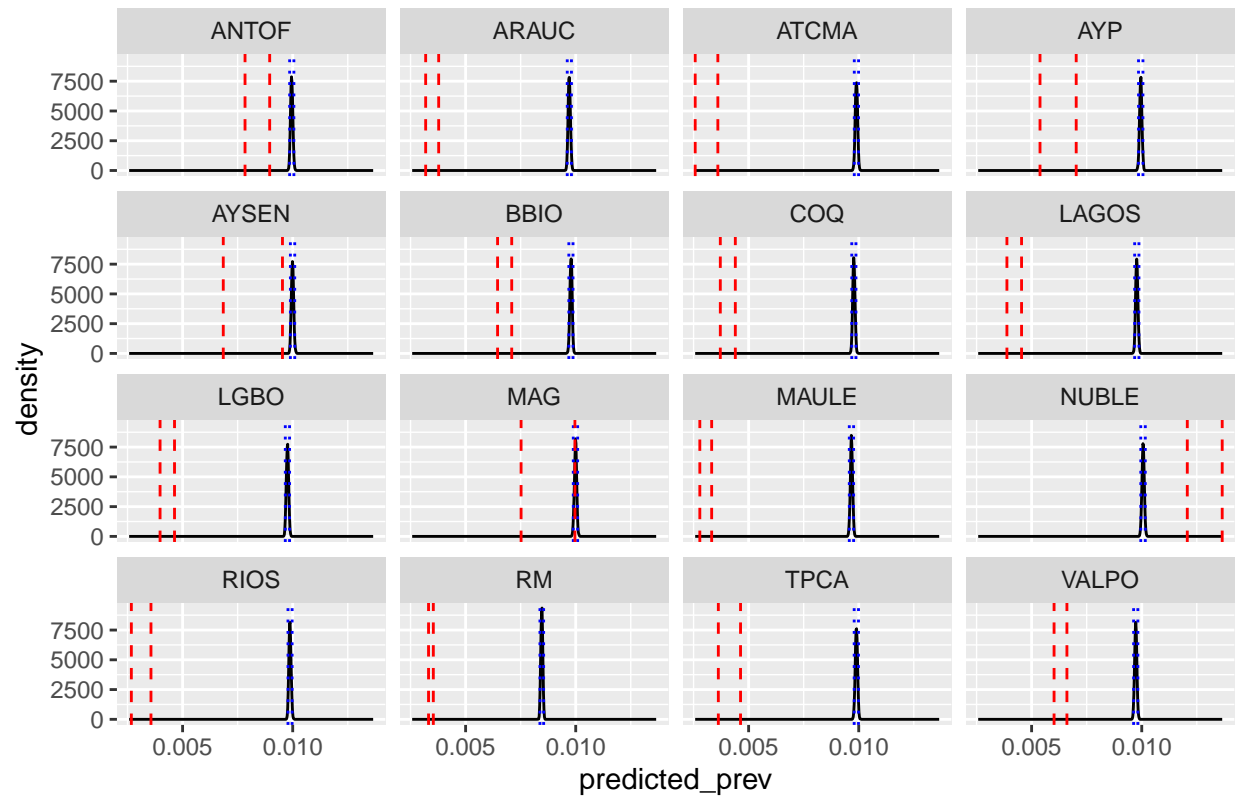
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.005, prior sd = 5.1e-05



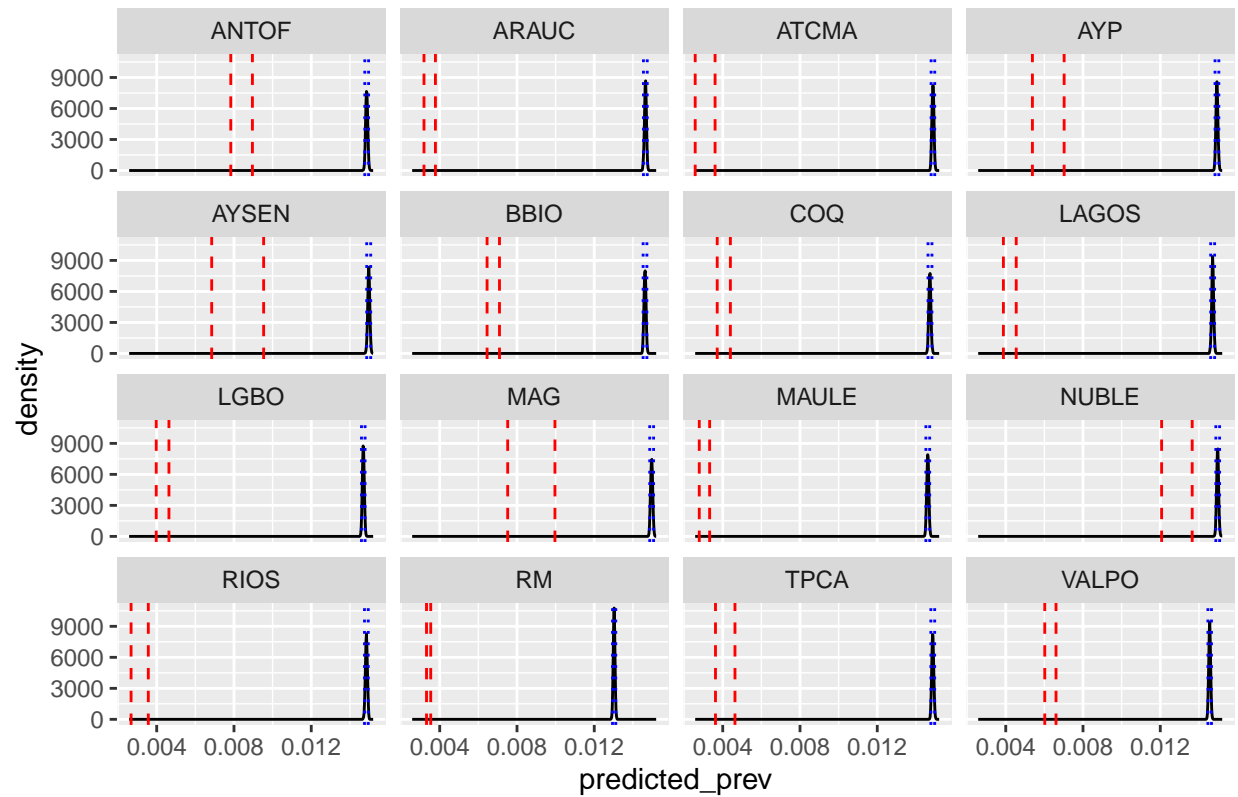
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```


Prior mean = 0.01, prior sd = 5.1e-05



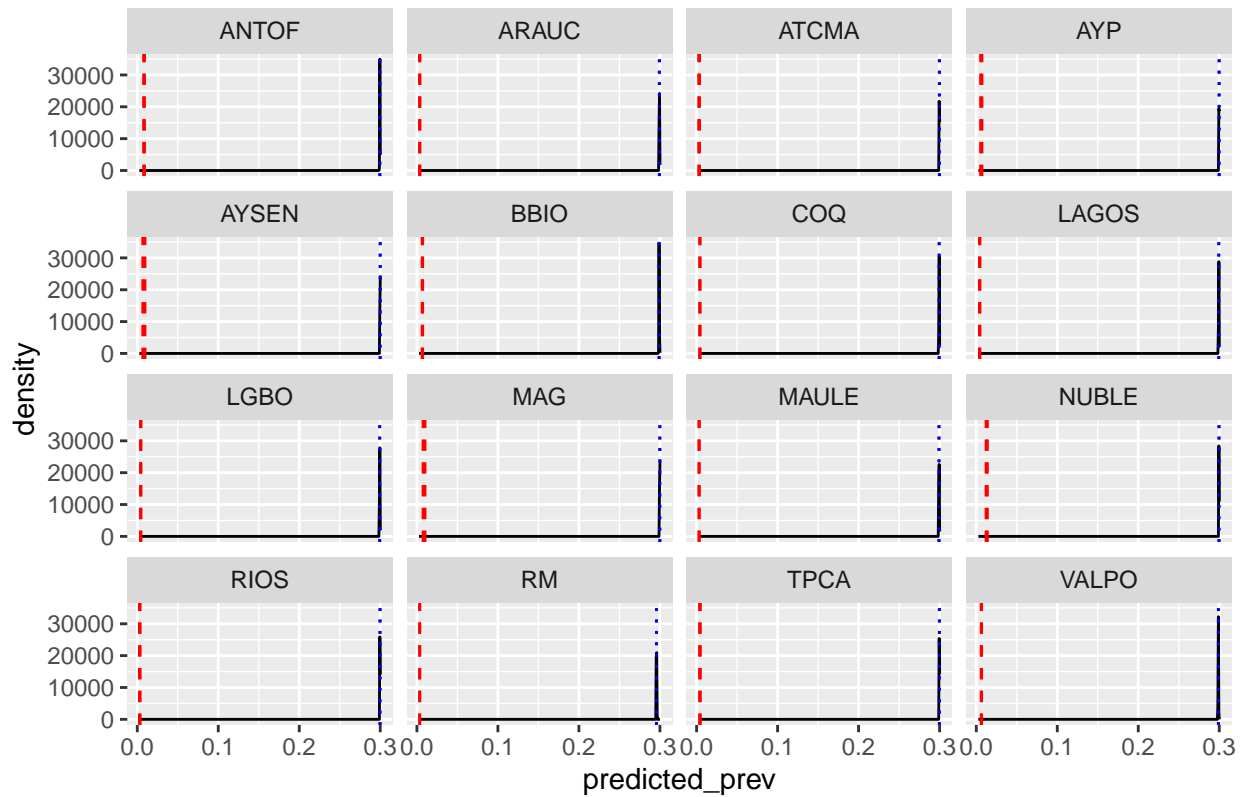
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.015, prior sd = 5.1e-05



```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

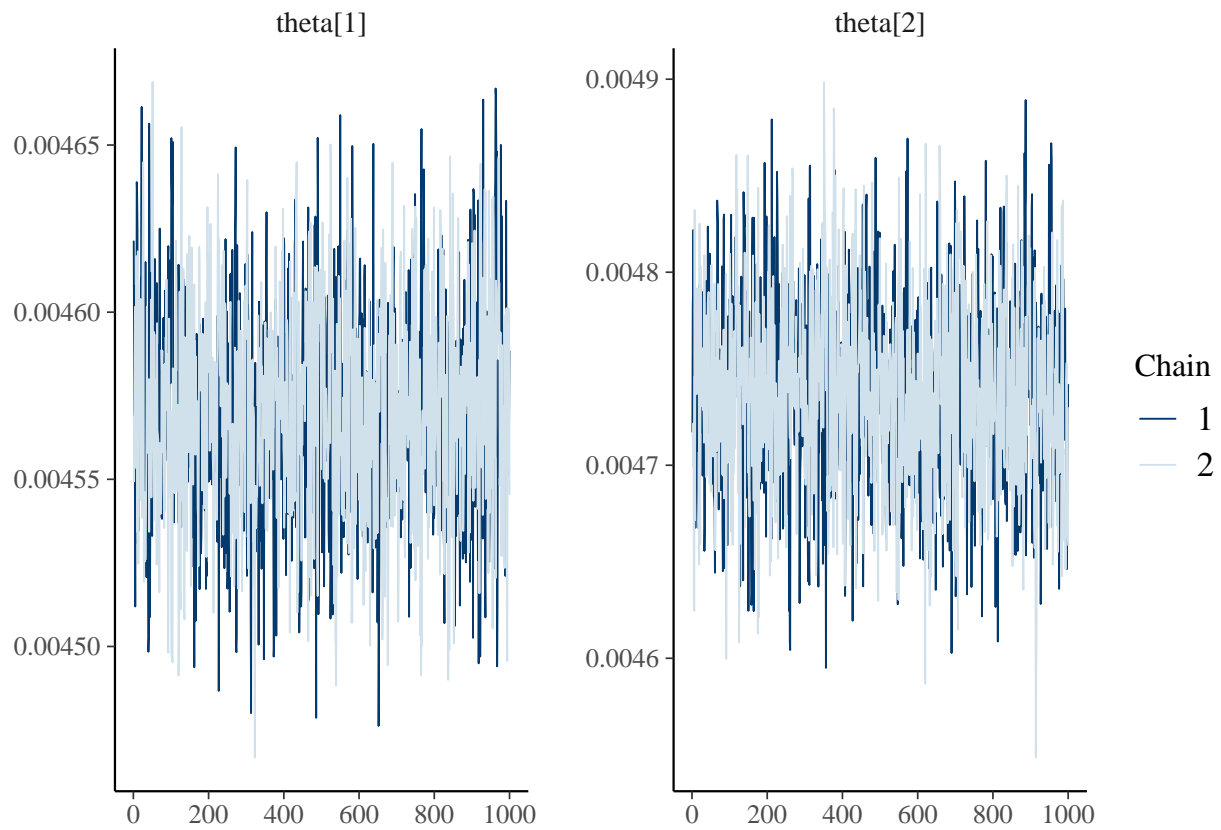
Prior mean = 0.3, prior sd = 5.1e-05

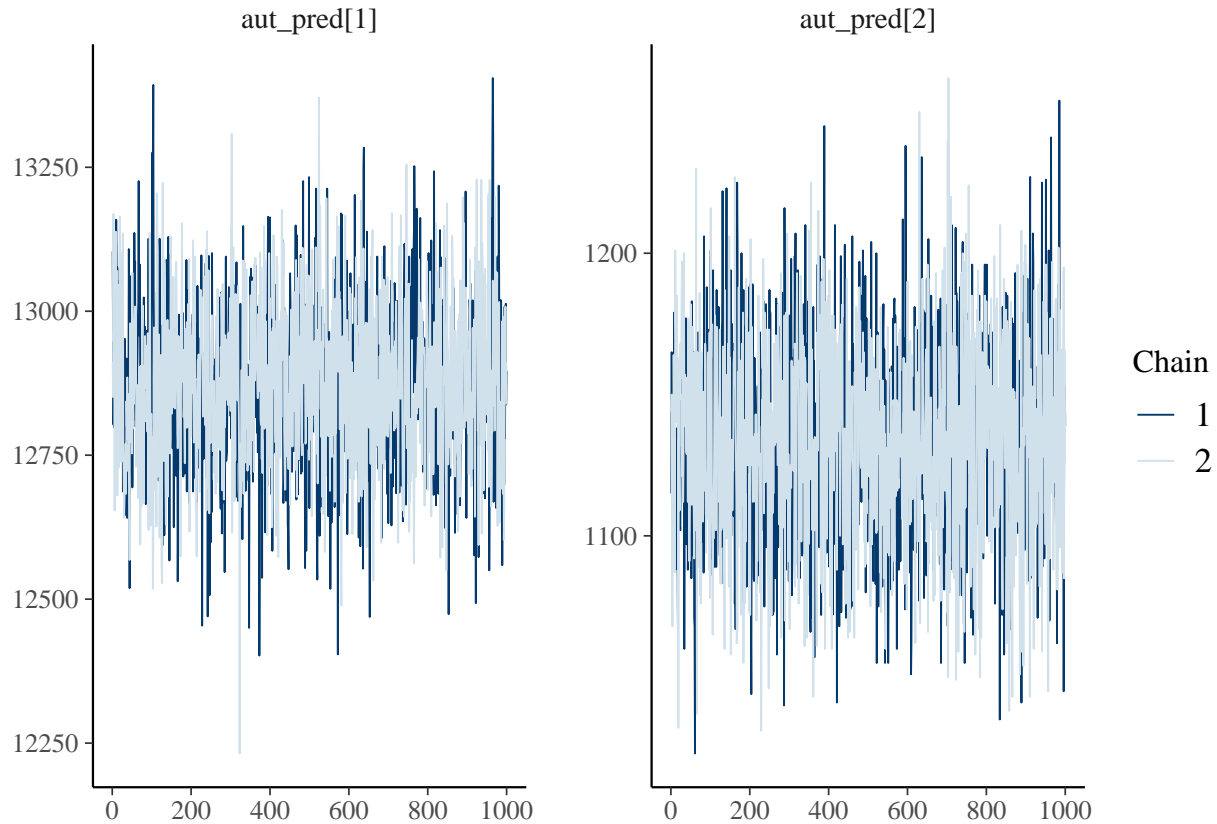


Random effect on rurality

```
aut_prev_rural_adj <- get_adjusted_prev(aut_prev_rural, grouping_vars = "school_rurality_code")

aut_prev_rural_post <- do_jags_rand_model(x = aut_prev_rural_adj,
  feat = "school_rurality_code",
  model = rand_model,
  theta_mu = theta_mu_prior,
  theta_sigma = theta_sigma_prior,
  pars = pars,
  convergence_checks = TRUE) %>%
  rename("school_rurality_code" = "Feat_names")
```





```
## # 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 theta[2]      0.00464      4.74e-3      4.84e-3      4.49e-6   1.46e-6   3.94e-6 1.00
## 2 theta[1]      0.00451      4.57e-3      4.63e-3      2.82e-6   1.06e-6   2.18e-6 1.00
## 3 aut_pred[2]    1062         1.13e+3      1.20e+3        2 e+0     5 e-1     2.5 e+0 1.00
## 4 aut_pred[1]   12598         1.29e+4      1.32e+4      1.15e+1    4.5 e+0    1.1 e+1 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
```

```
plot_post_density(aut_prev_rural_post, aut_prev_rural_adj, feat = "school_rurality_code", theta_mu = th
```

```
## Warning: Combining variables of class <factor> and <integer> was deprecated in ggplot2
## 3.4.0.
```

```
## i Please ensure your variables are compatible before plotting (location:
## `combine_vars()`)
```

```
## This warning is displayed once every 8 hours.
```

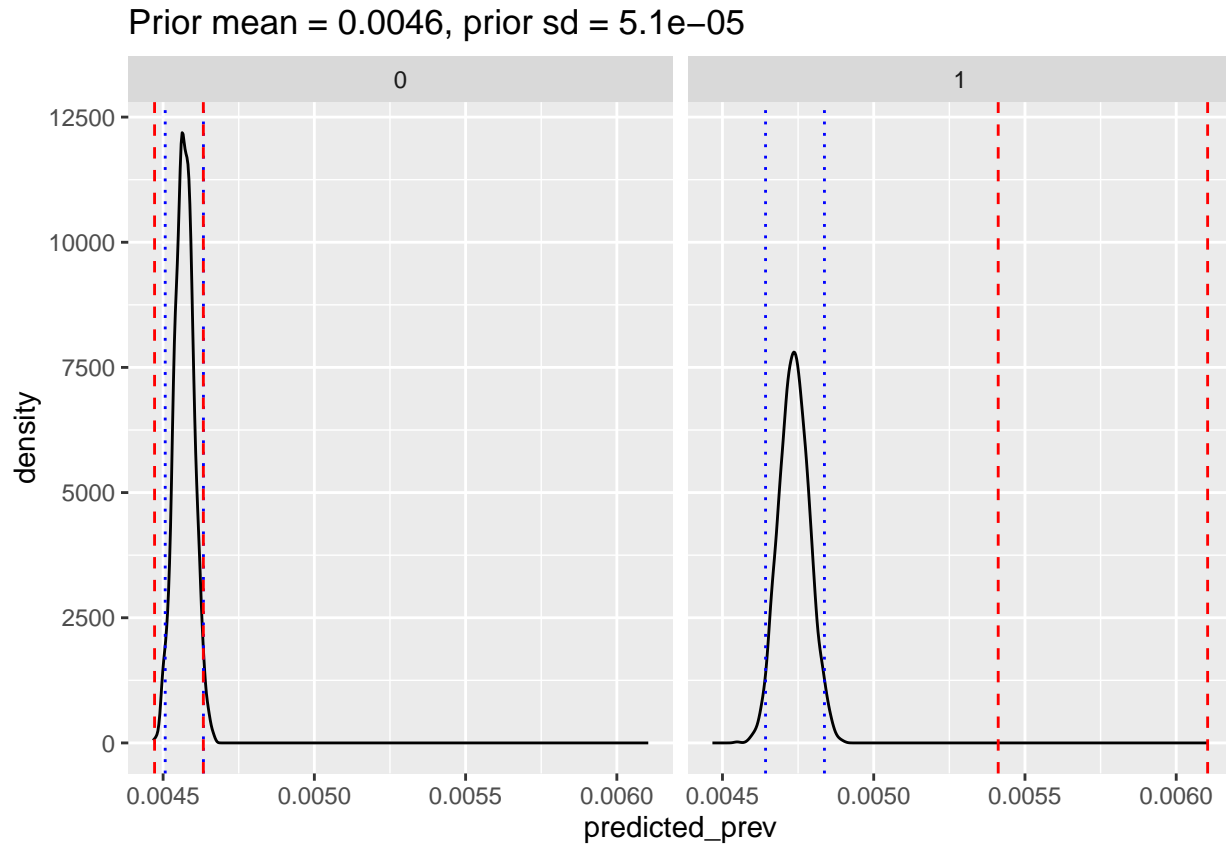
```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Combining variables of class <integer> and <factor> was deprecated in ggplot2
## 3.4.0.
```

```
## i Please ensure your variables are compatible before plotting (location:
## `combine_vars()`)
```

```
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



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

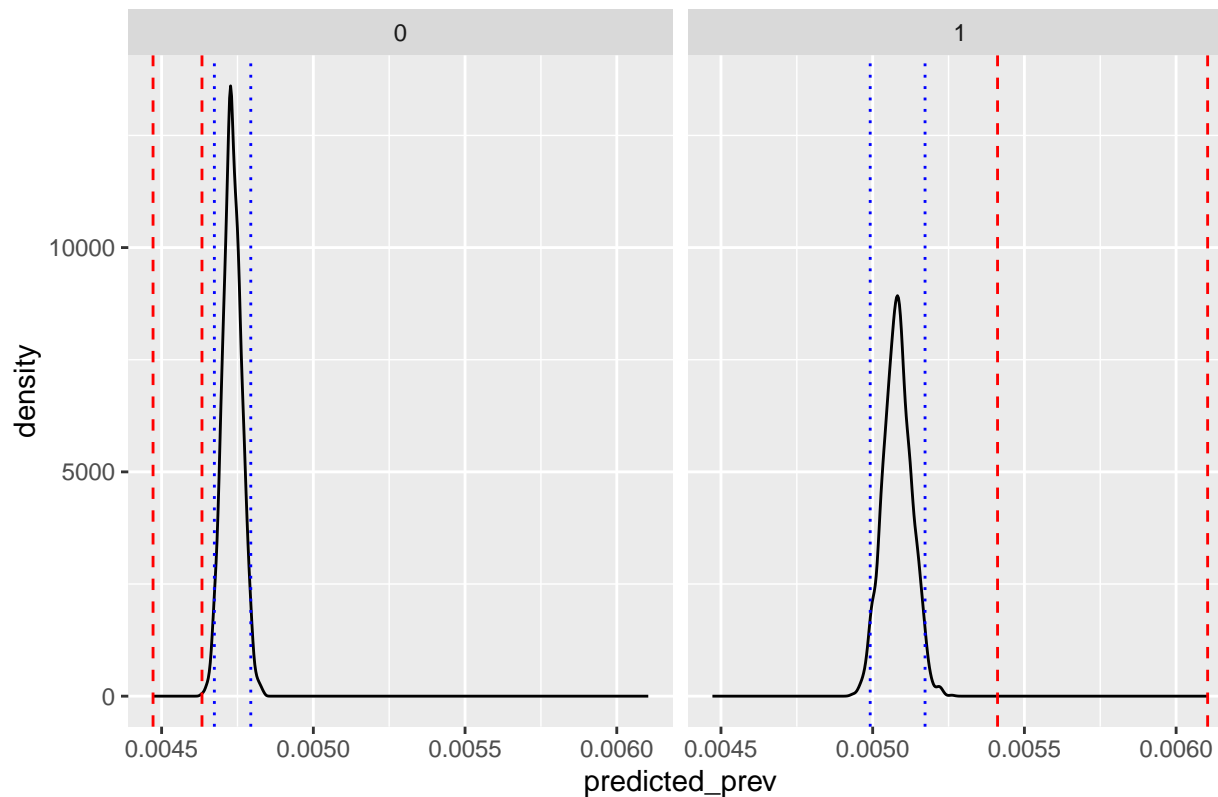
```
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_rural_post <- do_jags_rand_model(x = aut_prev_rural_adj,
                                           feat = "school_rurality_code",
                                           model = rand_model,
                                           theta_mu = theta_mu_extrapolate[j],
                                           theta_sigma = theta_sigma_extrapolate[j],
                                           pars = pars,
                                           convergence_checks = FALSE) %>%
  rename("school_rurality_code" = "Feat_names")

  plot_post_density(aut_prev_rural_post,
                    aut_prev_rural_adj,
                    feat = "school_rurality_code",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}
```

Predictions for higher population prevalence - increase prior mean

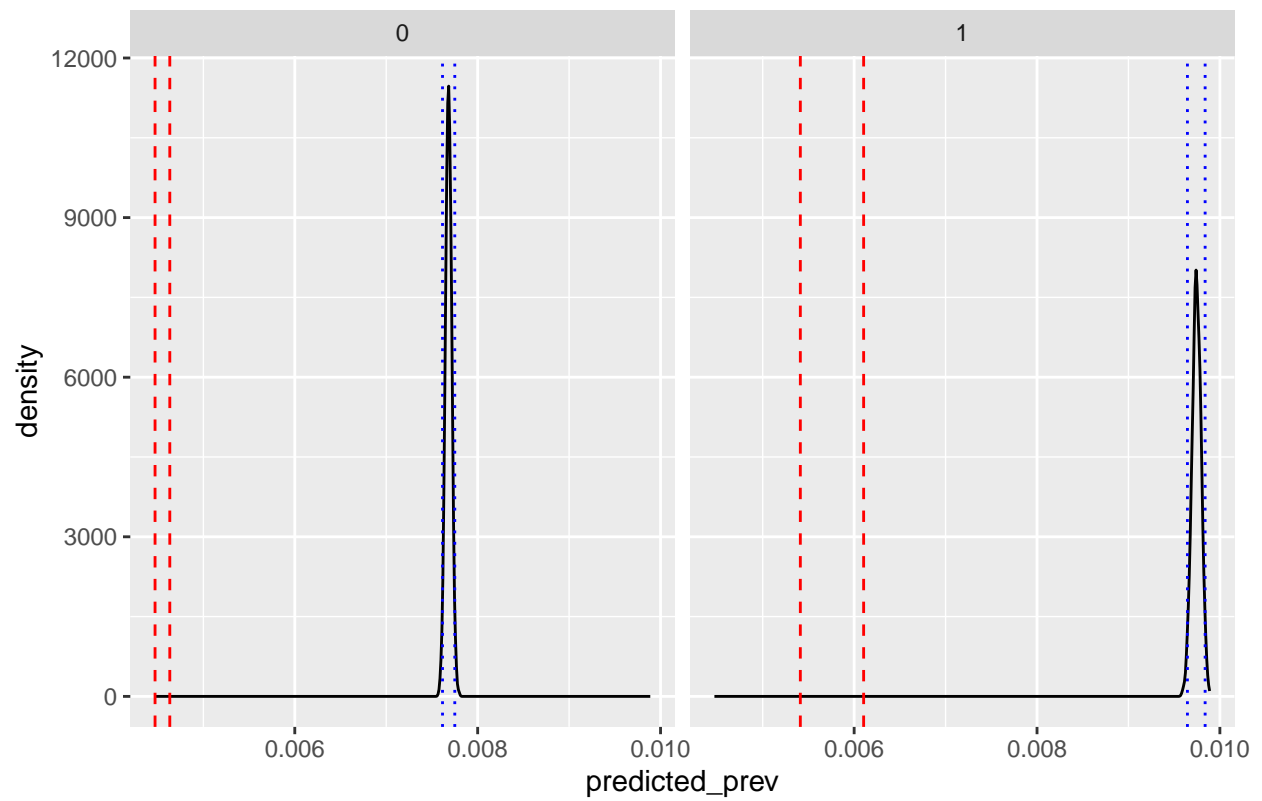
```
## Don't know how to automatically pick scale for object of type  
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.005, prior sd = 5.1e-05



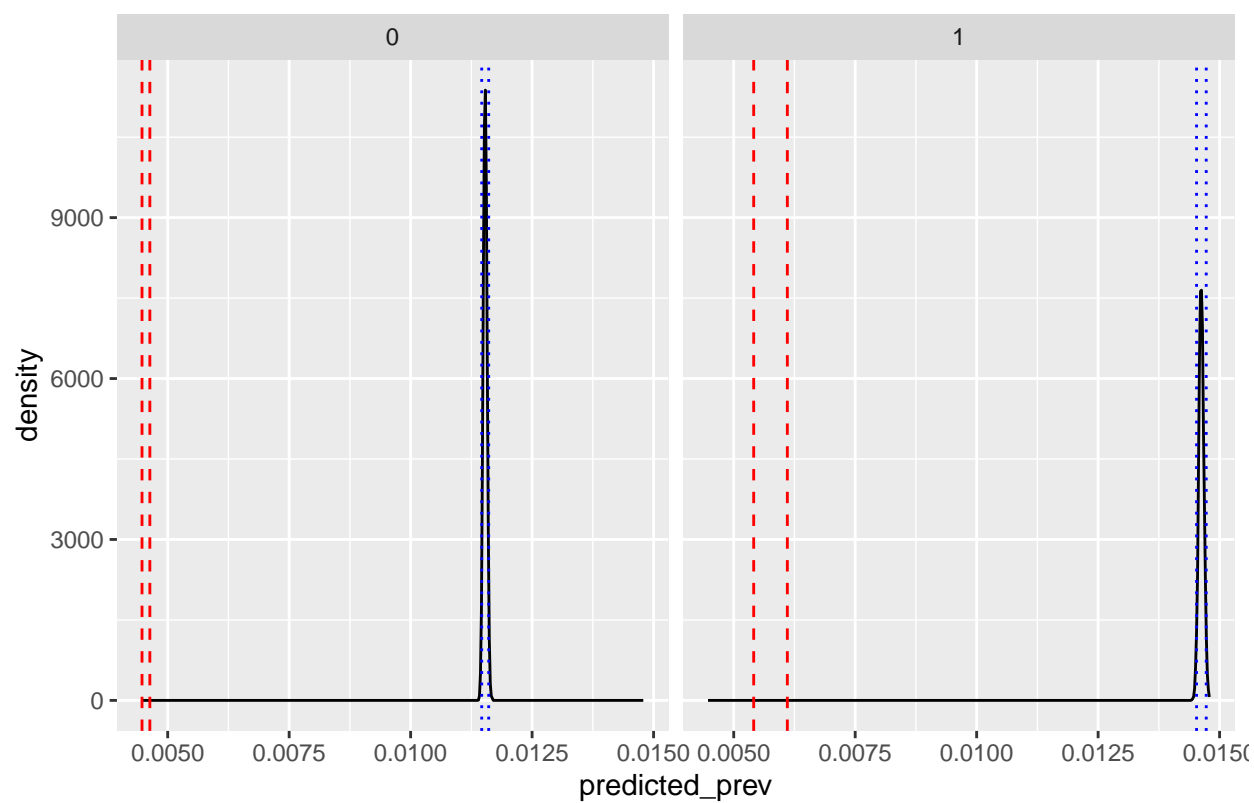
```
## Don't know how to automatically pick scale for object of type  
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.01, prior sd = 5.1e-05

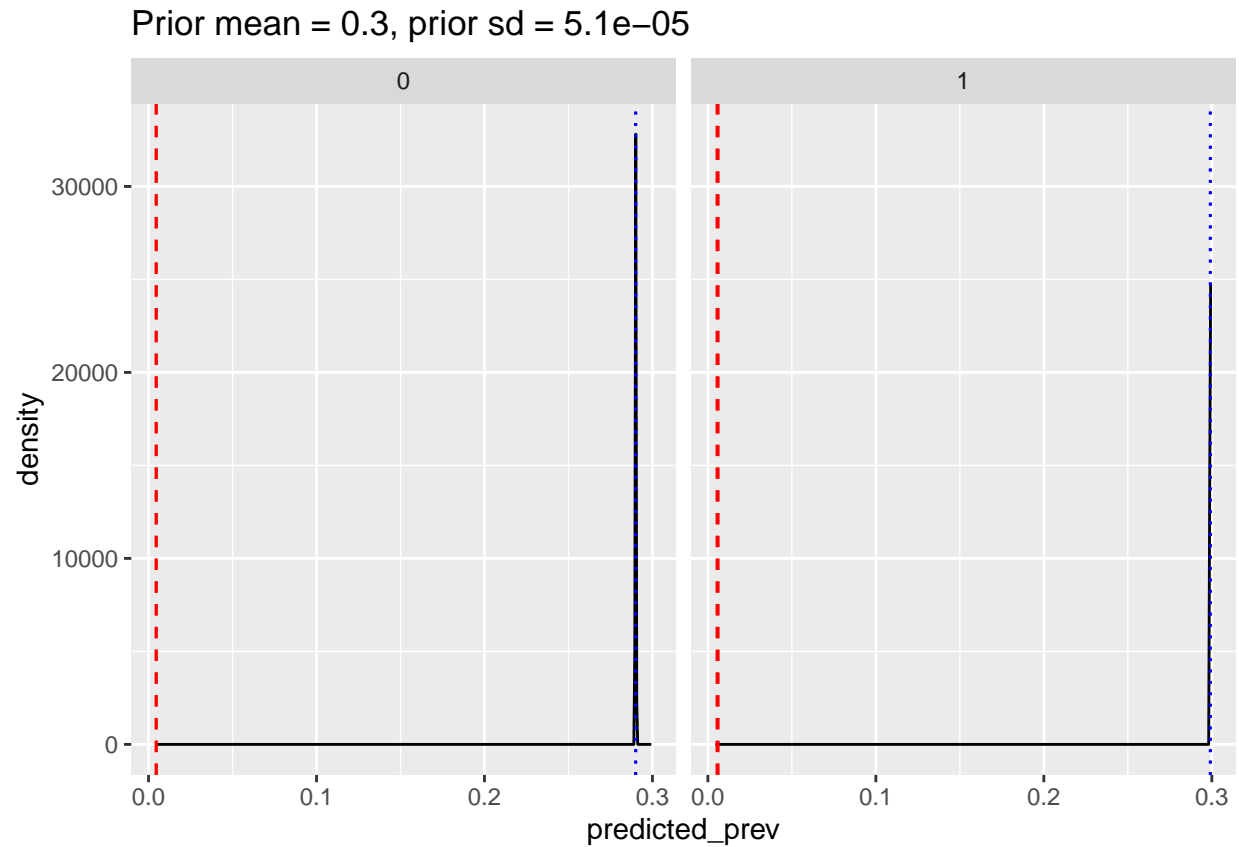


```
## Don't know how to automatically pick scale for object of type  
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```


Prior mean = 0.015, prior sd = 5.1e-05



```
## Don't know how to automatically pick scale for object of type  
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

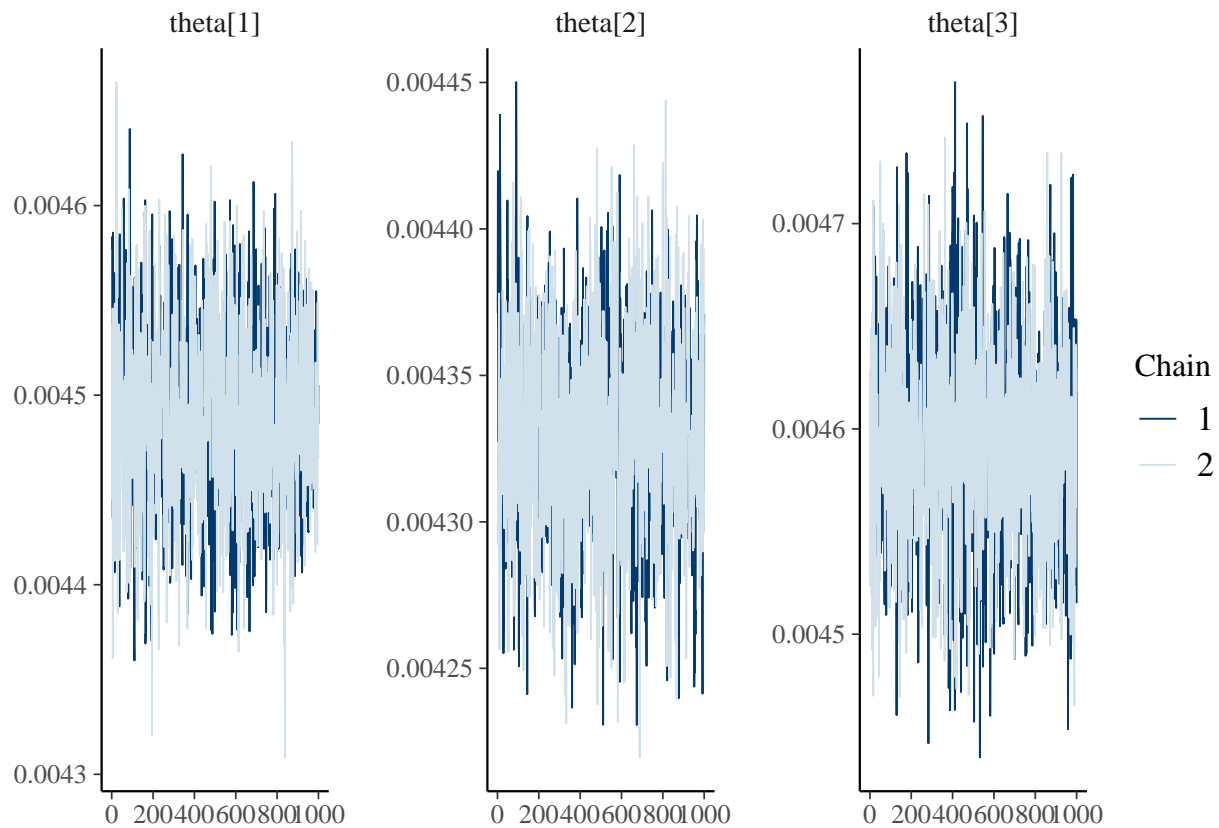


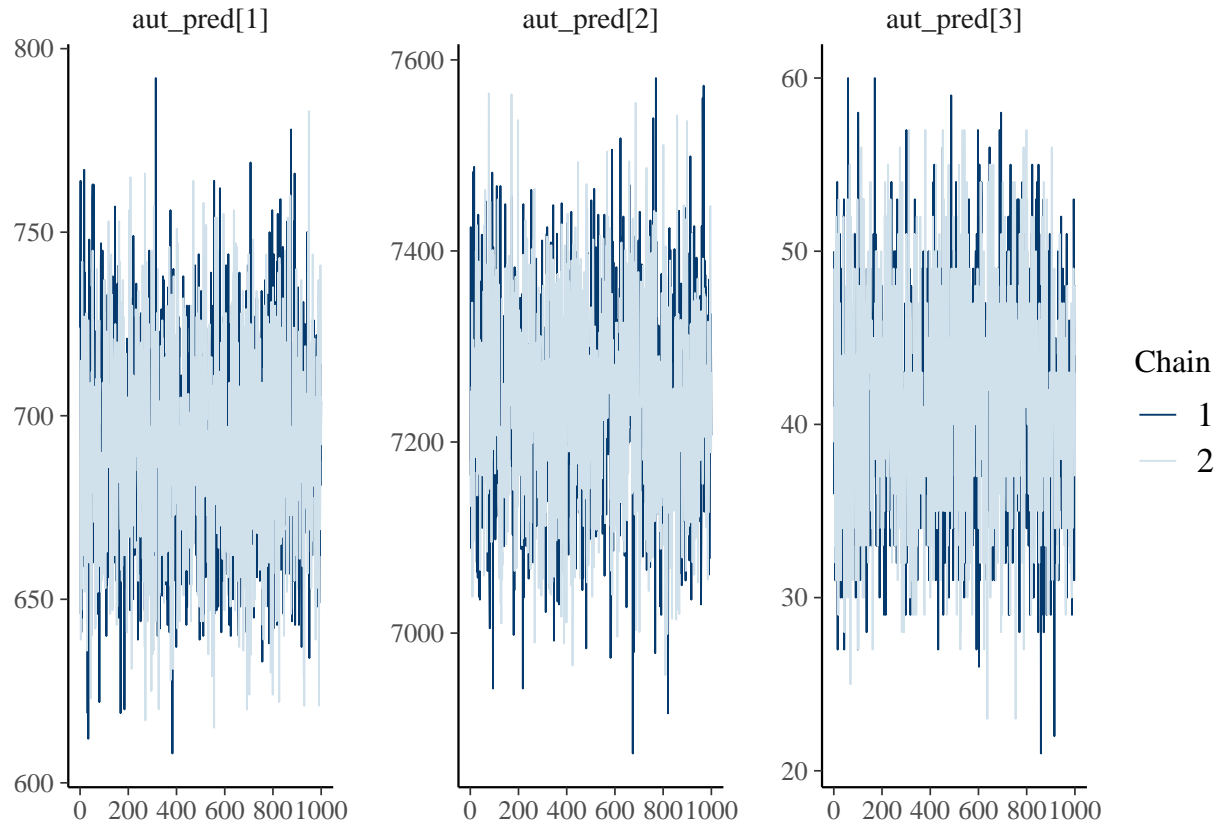
Random effect on ethnicity

Only use regions with large Mapuche populations

```
aut_prev_ethnic_adj <- get_adjusted_prev(aut_prev_ethnic, grouping_vars = "ethnic_2_group")

aut_prev_ethnic_post <- do_jags_rand_model(x = aut_prev_ethnic_adj,
  feat = "ethnic_2_group",
  model = rand_model,
  theta_mu = theta_mu_prior,
  theta_sigma = theta_sigma_prior,
  pars = pars,
  convergence_checks = TRUE) %>%
  rename("ethnic_2_group" = "Feat_names")
```



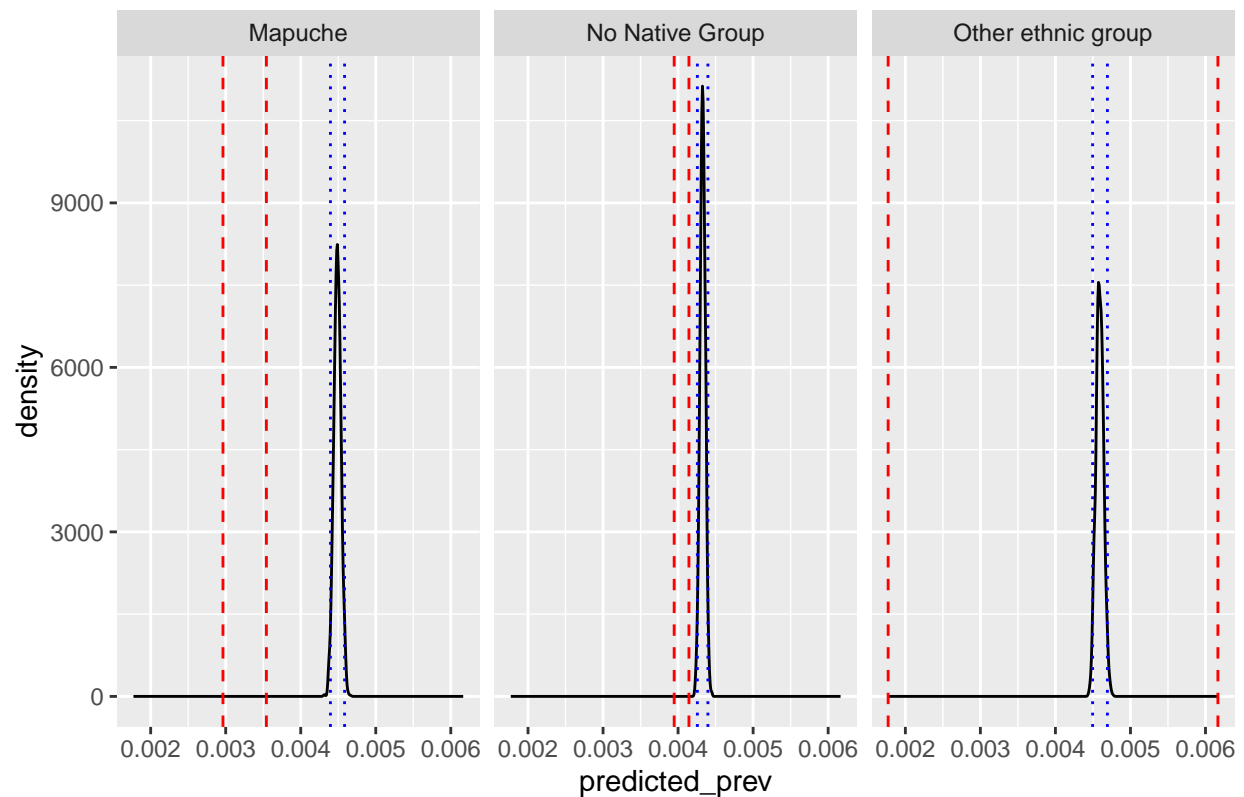


```
## # A tibble: 11 x 8
##   variable      `2.5%`    `50%`    `97.5%` mcse_q~1 mcse_q50 mcse_q~2 rhat
##   <chr>          <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <dbl>
## 1 theta[1]      0.00440  4.49e-3  4.58e-3  4.95e-6  1.46e-6  3.26e-6 1.01
## 2 theta[2]      0.00426  4.33e-3  4.40e-3  2.80e-6  1.36e-6  3.23e-6 1.00
## 3 theta[3]      0.00450  4.59e-3  4.69e-3  3.03e-6  1.96e-6  3.33e-6 1.00
## 4 aut_pred[2]   7045.    7.24e+3  7.44e+3    7    e+0  3.5    e+0  5.5    e+0 1.00
## 5 aut_pred[1]    641.    6.91e+2  7.45e+2    1    e+0    1    e+0  1.5    e+0 1.00
## 6 aut_pred[3]    30.0    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]   28    2.8    e+1  2.8    e+1   NA      NA      NA      NA
## # ... with abbreviated variable names 1: mcse_q2.5, 2: mcse_q97.5
```

```
plot_post_density(aut_prev_ethnic_post, aut_prev_ethnic_adj, feat = "ethnic_2_group", theta_mu = theta_mu)
```

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.0046, prior sd = 5.1e-05



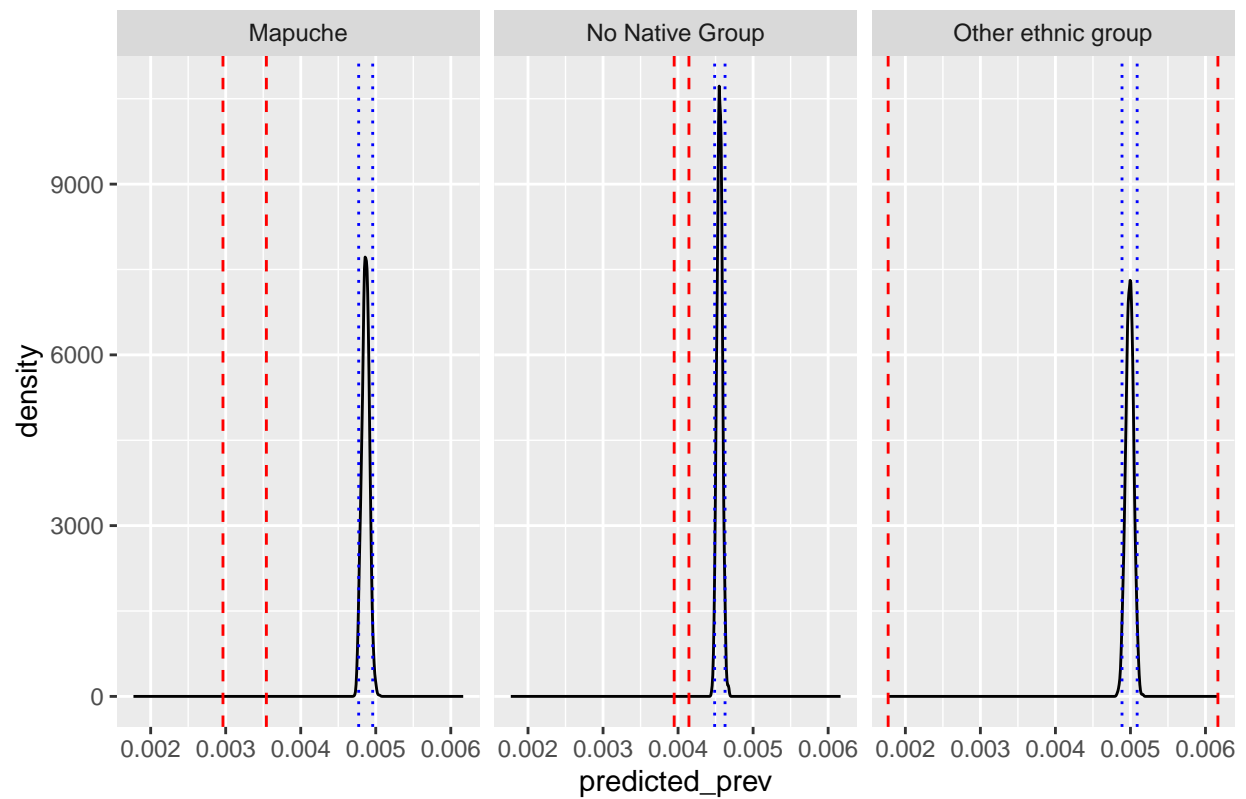
```
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_ethnic_post <- do_jags_rand_model(x = aut_prev_ethnic_adj,
                                             feat = "ethnic_2_group",
                                             model = rand_model,
                                             theta_mu = theta_mu_extrapolate[j],
                                             theta_sigma = theta_sigma_extrapolate[j],
                                             pars = pars,
                                             convergence_checks = FALSE) %>%
    rename("ethnic_2_group" = "Feat_names")

  plot_post_density(aut_prev_ethnic_post,
                    aut_prev_ethnic_adj,
                    feat = "ethnic_2_group",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}
```

Predictions for higher population prevalence - increase prior mean

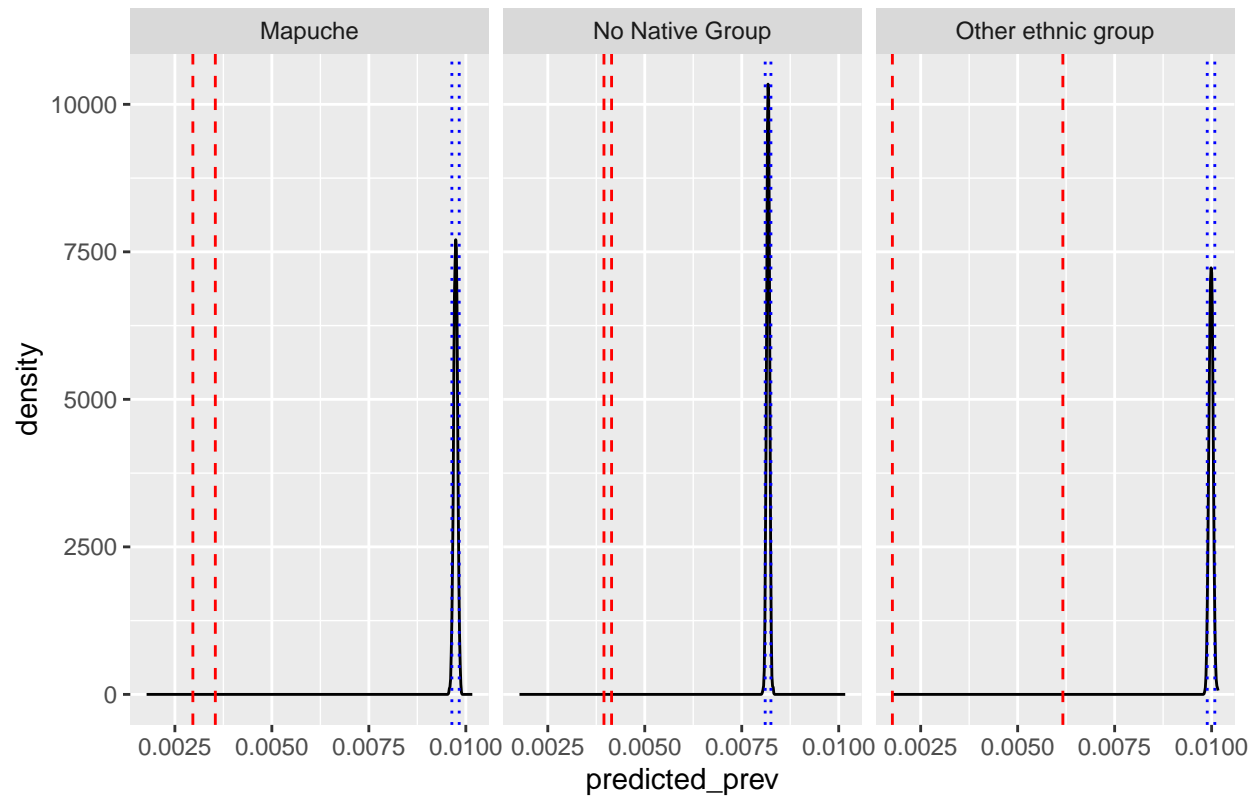
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.005, prior sd = 5.1e-05



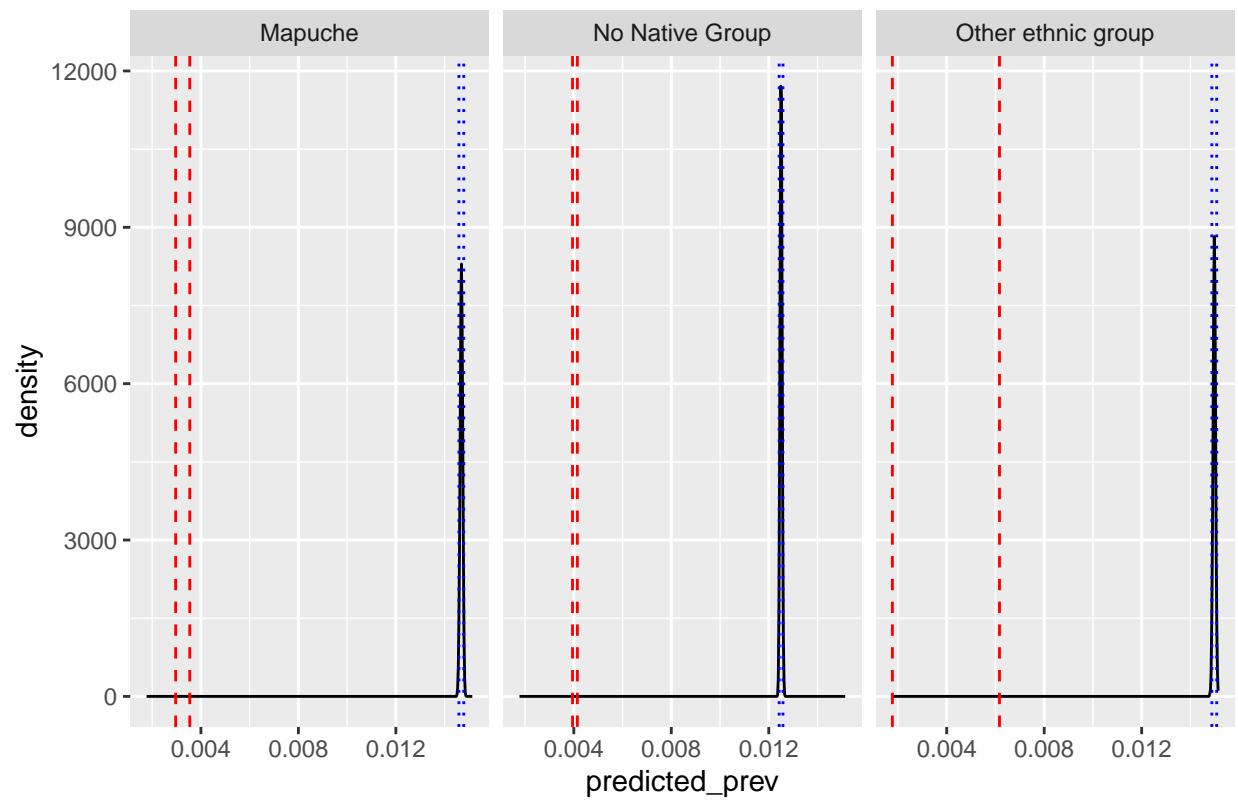
```
## Don't know how to automatically pick scale for object of type  
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.01, prior sd = 5.1e-05



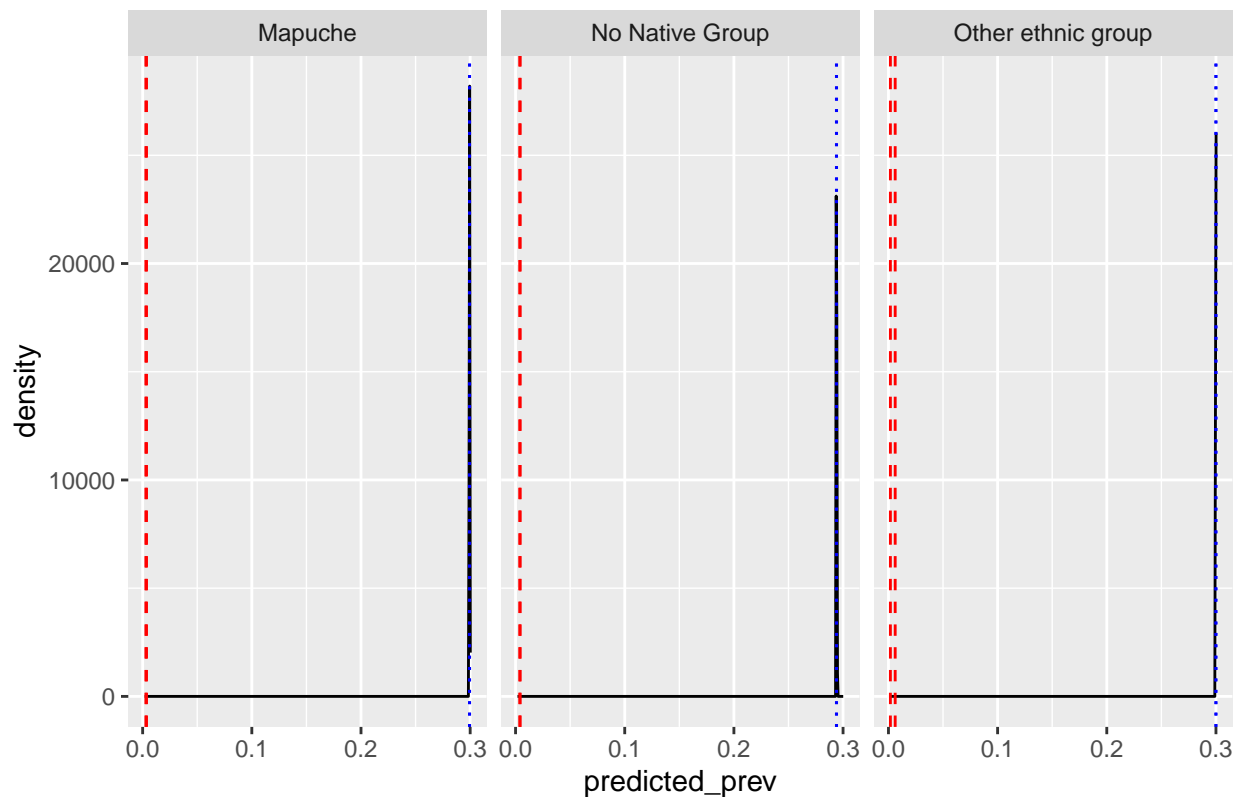
```
## Don't know how to automatically pick scale for object of type  
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.015, prior sd = 5.1e-05



```
## Don't know how to automatically pick scale for object of type  
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```


Prior mean = 0.3, prior sd = 5.1e-05



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",
                                                            "MAS DE $100.000", "High", "No information"),
                                           "Other", "Low")),
         age_june30 = ifelse(is.na(age_june30), 0, age_june30),
         sex = ifelse(is.na(sex), "Other", sex),
         autism = ifelse(is.na(autism), 0, 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)

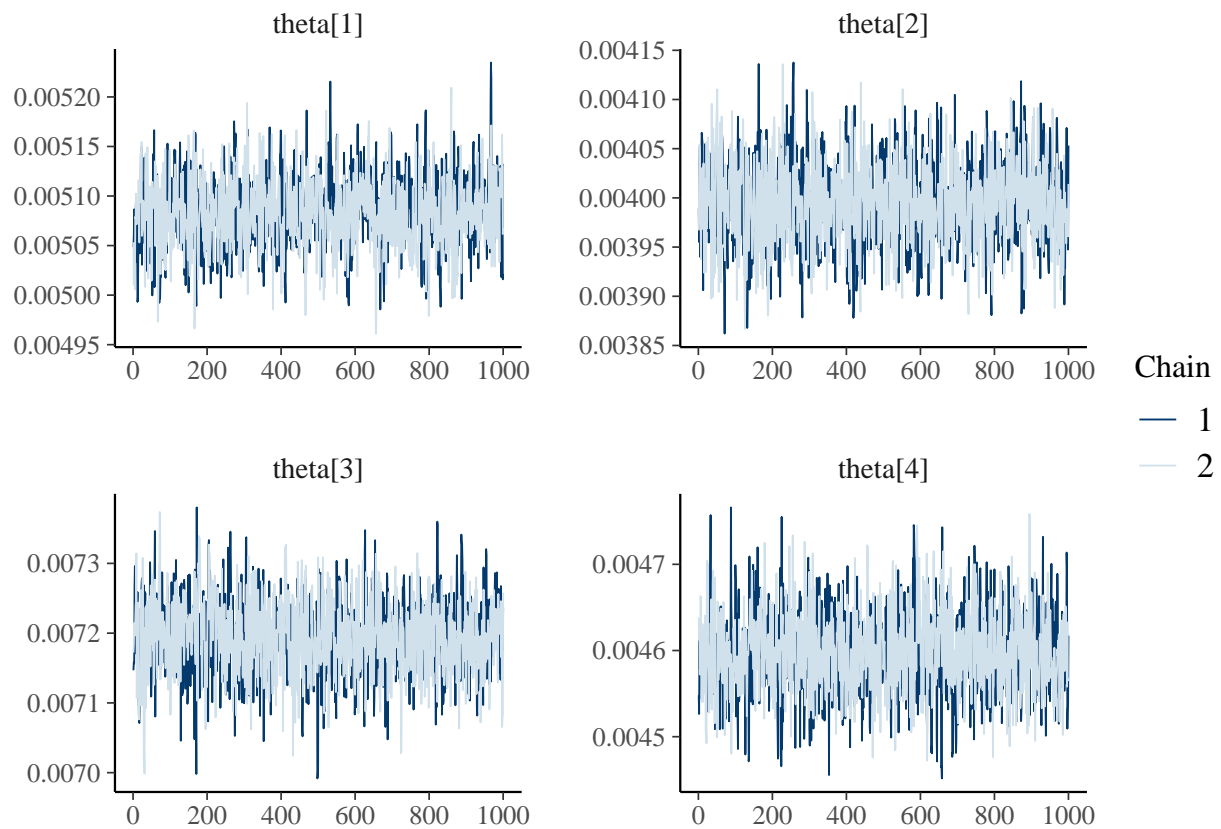
rand_econ_ini <- list(list(theta = rep(0.001, nEcon)), #, spec = 0.5, sens = 0.5),
                     list(theta = rep(0.01, nEcon))) #, spec = 0.9, sens = 0.9))

# 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)

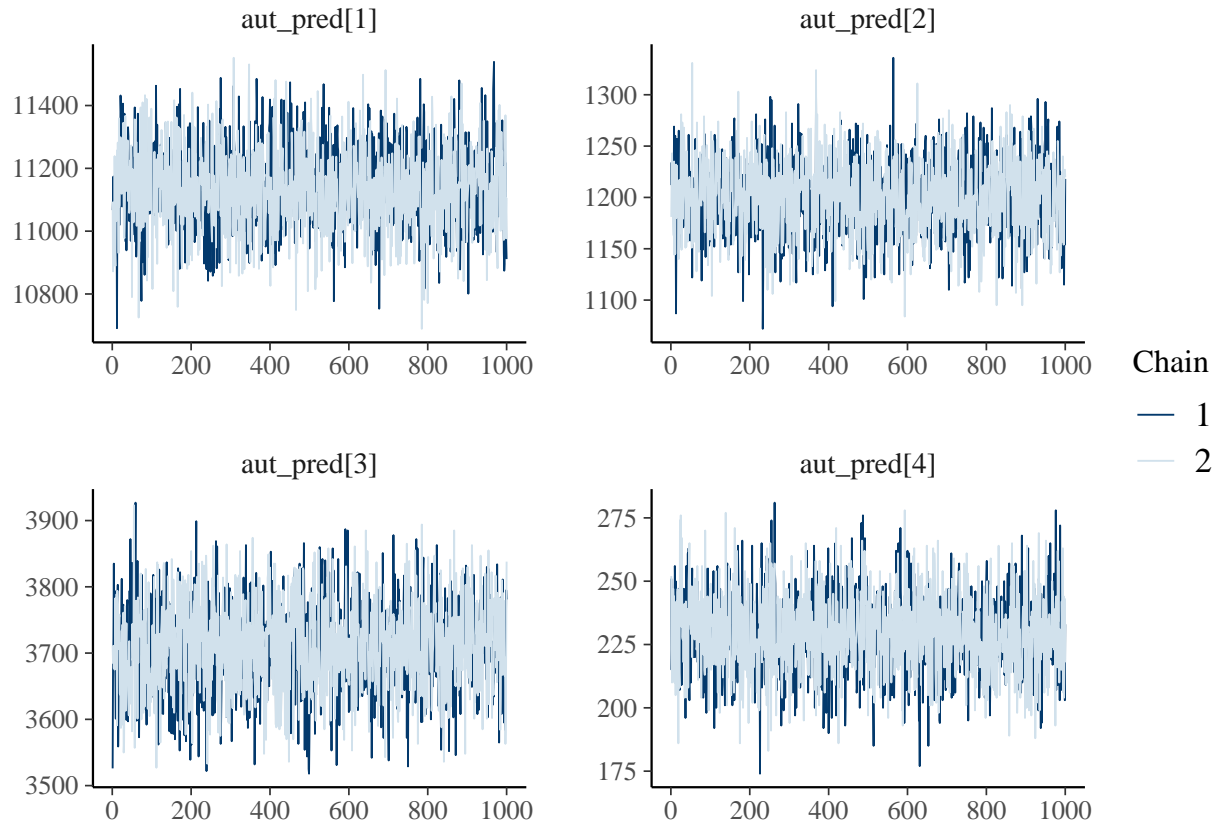
update(rand_econ_jag, n.iter = nBurn)
rand_econ_sam <- coda.samples(model = rand_econ_jag,
                             variable.names = pars,
                             n.iter = nIter)

```

```
# 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)
```

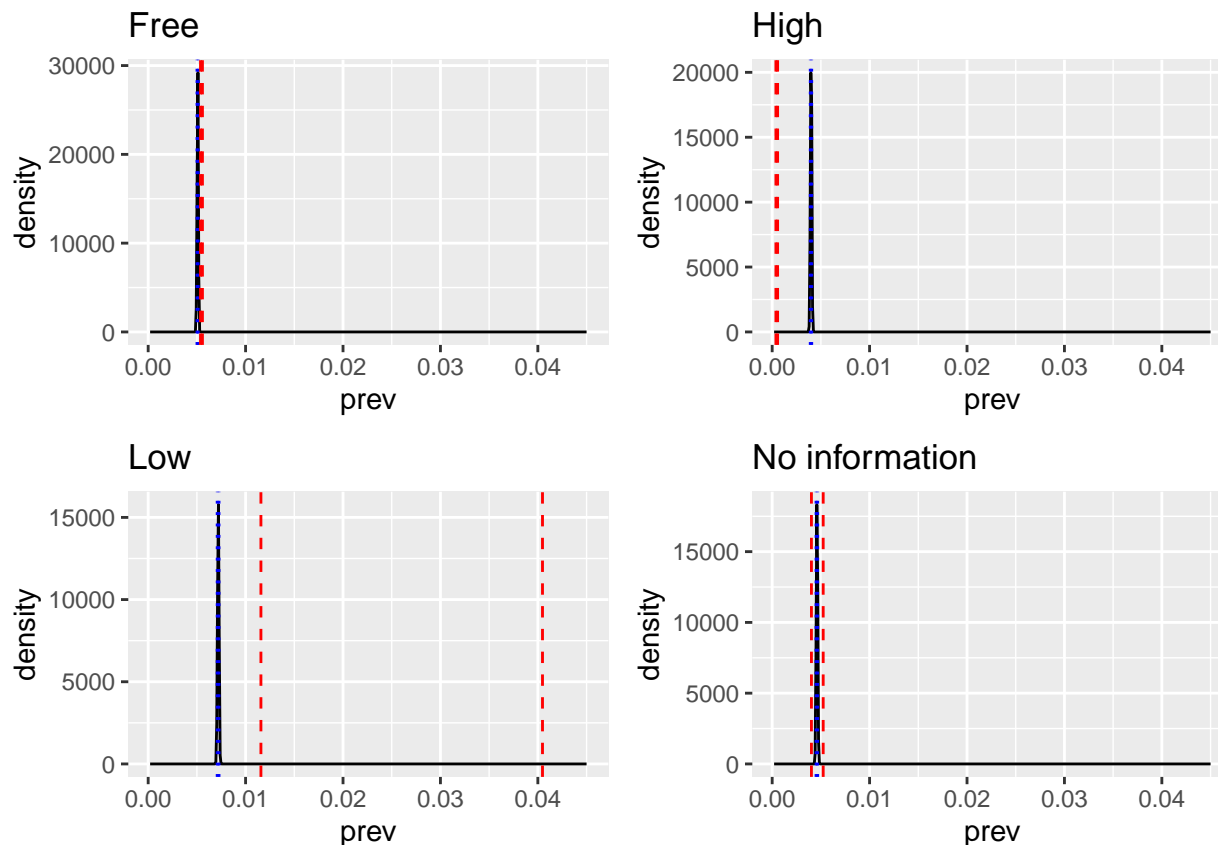
```
## # A tibble: 14 x 10
##   varia~1 mean median sd mad q5 q95 rhat ess_b~2 ess_t~3
##   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 aut_pr~ 1.11e+4 1.11e+4 1.32e+2 1.29e+2 1.09e+4 1.14e+4 1.00 1491. 1615.
## 2 aut_pr~ 1.20e+3 1.2 e+3 3.65e+1 3.71e+1 1.14e+3 1.26e+3 1.00 2001. 1770.
## 3 aut_pr~ 3.71e+3 3.71e+3 6.75e+1 6.52e+1 3.60e+3 3.82e+3 1.00 1781. 1592.
## 4 aut_pr~ 2.29e+2 2.3 e+2 1.56e+1 1.48e+1 2.04e+2 2.56e+2 1.00 2009. 1777.
## 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 3.70e-5 3.77e-5 5.02e-3 5.14e-3 1.00 977. 1079.
## 10 theta[~ 4.00e-3 4.00e-3 4.39e-5 4.40e-5 3.92e-3 4.07e-3 1.00 1365. 1074.
## 11 theta[~ 7.20e-3 7.20e-3 5.58e-5 5.70e-5 7.10e-3 7.28e-3 1.00 1270. 1061.
## 12 theta[~ 4.60e-3 4.60e-3 4.91e-5 4.97e-5 4.52e-3 4.68e-3 1.00 1330. 1322.
## 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
```

```
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]    10882.    1.11e+4 1.14e+4 8    e+0 3.5  e+0 7.5  e+0 1.00
## 2 aut_pred[3]    3575.    3.71e+3 3.84e+3 5    e+0 2    e+0 3.5  e+0 1.00
## 3 aut_pred[2]    1127     1.2  e+3 1.27e+3 1.5  e+0 1.5  e+0 2    e+0 1.00
## 4 aut_pred[4]     199     2.3  e+2 2.62e+2 1    e+0 5    e-1 1.5  e+0 1.00
## 5 theta[3]        0.00709 7.20e-3 7.30e-3 5.46e-6 2.07e-6 6.57e-6 1.00
## 6 theta[4]        0.00450 4.60e-3 4.70e-3 4.76e-6 1.66e-6 3.63e-6 1.00
## 7 theta[1]        0.00501 5.08e-3 5.16e-3 3.17e-6 1.37e-6 3.37e-6 1.00
## 8 theta[2]        0.00391 4.00e-3 4.08e-3 3.16e-6 1.33e-6 1.98e-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)
}

```