Bayesian prevalence analysis of autism prevalence in Chile

Adele Tyson

2023-06-09

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

library(nleqslv) # Only needed for robince bayesian prevalence
library(janitor)
library(gridExtra)
library(readxl)
library(viridis)
library(wesanderson)
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)
```

Set up

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))

chile_communes_raw <- read_excel("04_Data/commune_by_health_service.xlsx") %>%
  clean_names()

chile_communes <- chile_communes_raw %>%
  mutate(comuna_upper = toupper(comuna)) %>%
  mutate(school_commune_name = ifelse(comuna_upper == "AISÉN", "AYSÉN",
                                      ifelse(comuna_upper == "LA CALERA", "CALERA",
                                             ifelse(comuna_upper == "COIHAIQUE", "COYHAIQUE",
                                                    ifelse(comuna == "Isla de Pascua", "ISLA DE PASCUA"
                                                           ifelse(comuna_upper == "MAULLÍN" , "MAULLIN"
                                                                  ifelse(comuna == "Pedro Aguirre Cerda
                                                                         ifelse(comuna_upper == "RÁNQUI
                                                                                ifelse(comuna_upper ==
                                                                                       ifelse(comuna_up
))) %>%
  rename("health_service_name" = "servicio_de_salud") %>%
```

```
  select(school_commune_name, health_service_name)
```

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)
         age_cat_name = ifelse(age_cat == 1, "6-8", ifelse(age_cat == 2, "9-11", ifelse(age_cat == 3, "
          # 1 = 6-8, 2 = 9-11, 3 = 12-14, 4 = 15-18
         age_cat_name = factor(age_cat_name, levels = c("6-8", "9-11", "12-14", "15-18")),
         ethnic_2_group = ifelse(ethnic_3_group == "Aymara", "Other ethnic group", ethnic_3_group),
         school_fee_temp = school_fee,
         school_fee = ifelse(school_fee == "", "No information",
                             ifelse(school_fee == "GRATUITO", "Free",
                                    ifelse(school_fee == "$1.000 A $10.000", "$1,000-$10,000",
                                           ifelse(school_fee == "$10.001 A $25.000", "$10,001-$25,000",
                                                  ifelse(school_fee == "$25.001 A $50.000", "$25,001-$5
                                                         ifelse(school_fee == "$50.001 A $100.000", "$5
                                                                ifelse(school_fee == "MAS DE $100.000",
                                                                       ifelse(school_fee == "SIN INFORMA
         school_fee = factor(school_fee, levels = c("Free", "$1,000-$10,000", "$10,001-$25,000", "$25,0
         school_fee_group = ifelse(school_fee == "Free", "Free",
                                   ifelse(school_fee %in% c("$1,000-$10,000", "$10,001-$25,000", "$25,0
                                          ifelse(school_fee == "$100,001+", "High",
                                                 ifelse(school_fee == "No information", "No information
         school_fee_group = factor(school_fee_group, levels = c("Free", "Low", "High", "No information"
  left_join(chile_communes, by = "school_commune_name") %>%

  select(school_region_name_abr,
    sex,
    sex_desc,
    age_june30,
    #edad_alu_2, # equal to age_june30
    age_cat,
    age_cat_name,
    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,
    school_fee_temp,
    ethnicity,
```

```
    mapuche,
    nationality,
    ethnic_3_group,
    ethnic_2_group,
    #asd_chile, # equal to autism
    autism,
    school_commune_name,
    health_service_name
  )

# 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, needs columns called autism, count
  # stdpop = standard population with age and sex counts
  # grouping_vars = variables in x to group by

  n_stdpop <- sum(stdpop$std_pop)

  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_stdpop), # 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
            #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 = ifelse(var == 0, 0, var / (2*adjusted_rate) * qchisq(p = 0.05/2, df = 2*adjusted
            ci_upper = (var + w_M^2) / (2*(adjusted_rate + w_M)) * qchisq(p = 1-0.05/2, df = 2*(adjuste
  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 <
    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 interv
  # 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

  # calcuate 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))))
}
```

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

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

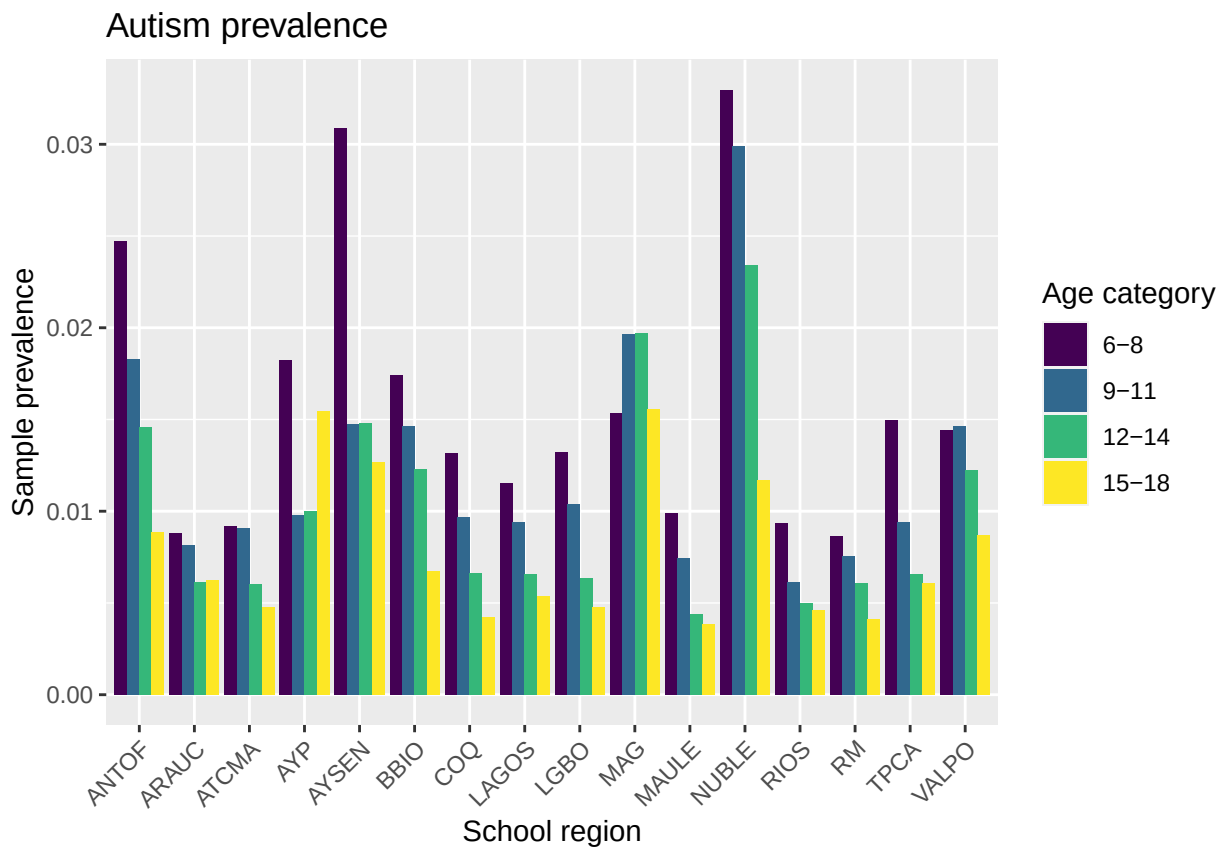
See https://github.com/Dpananos/bayes_multiple_measures/blob/master/analysis/sensitivity_analysis.R for more sensitivity analysis ideas

Random effect on region

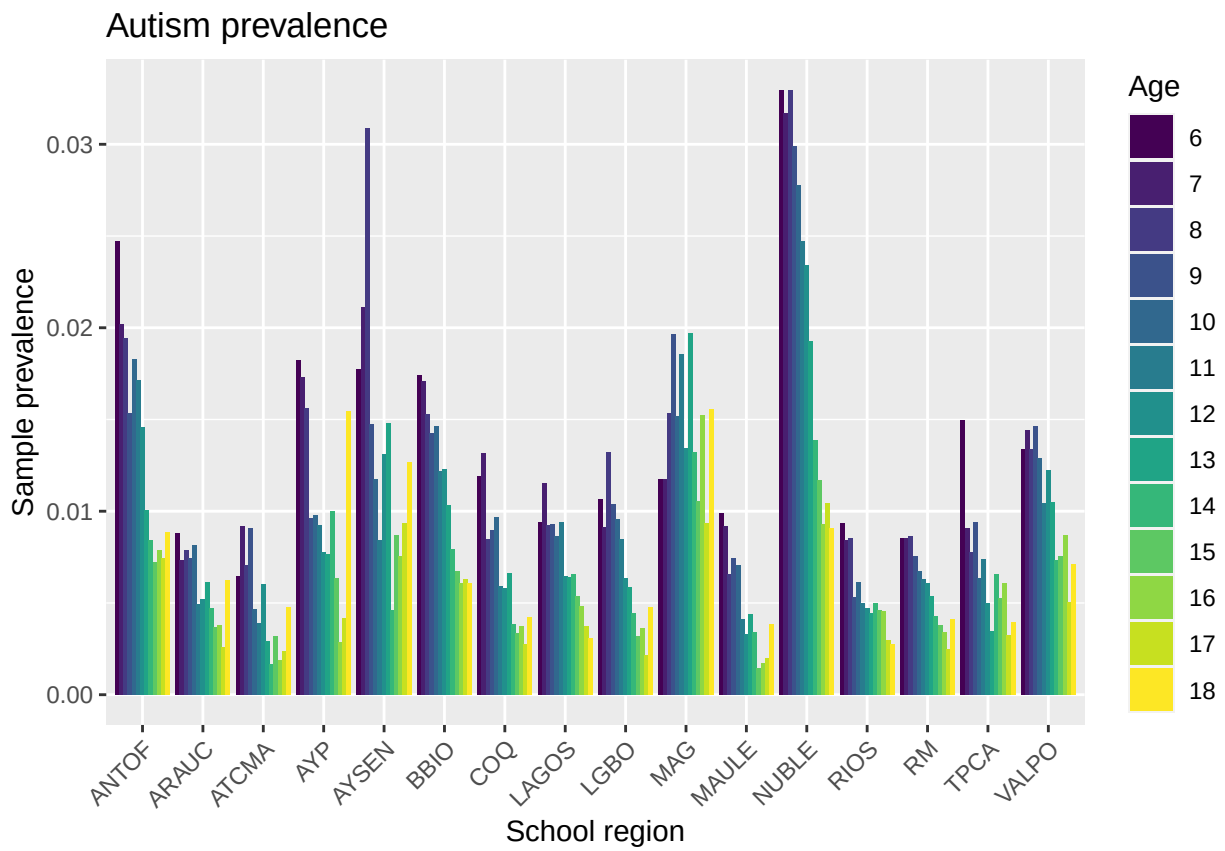
```
aut_prev_region <- get_grouped_prev(x = chile_bayes_aut, stdpop = chile_stdpop,
                                    grouping_vars = c("school_region_name_abr", "age_june30", "age_cat_
```

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

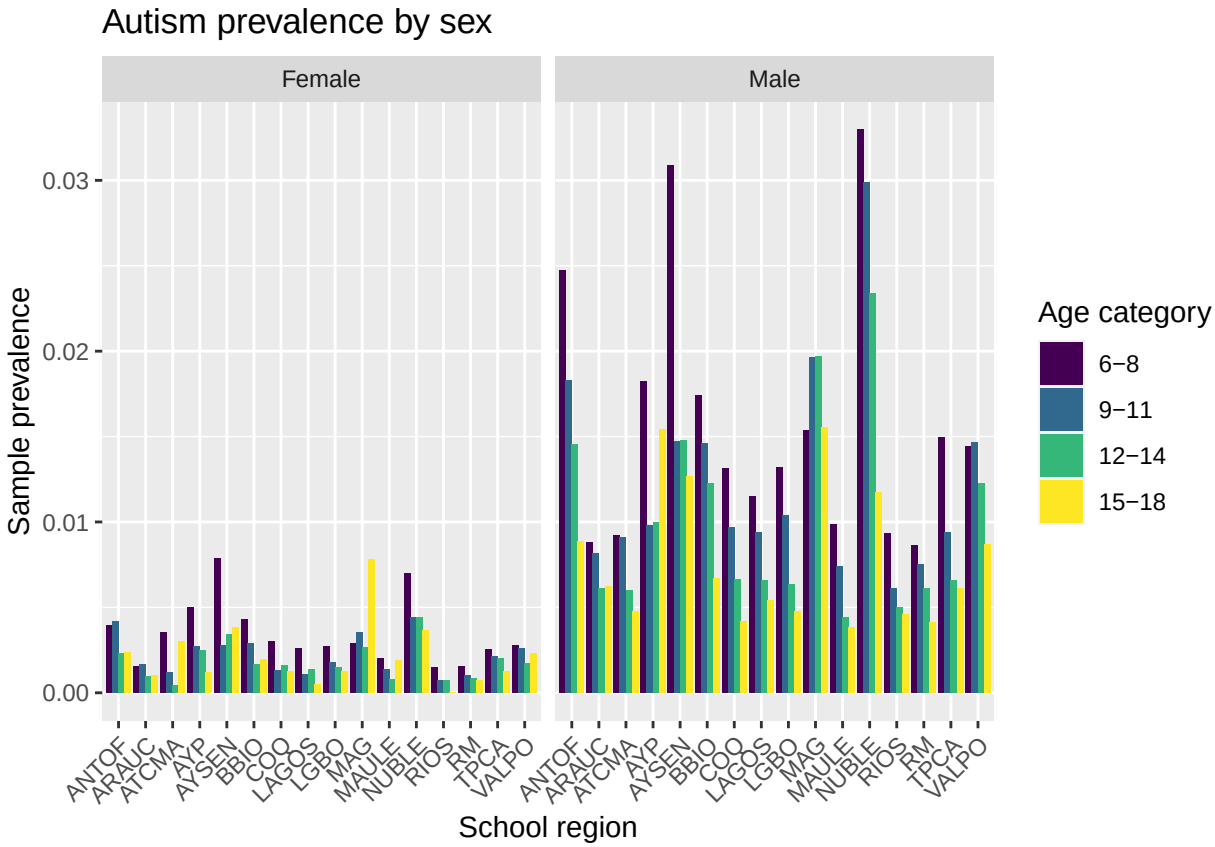
```
ggplot(data = aut_prev_region) +
  geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = age_cat_name, fill = age_cat_
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "School region",
       y = "Sample prevalence",
       fill = "Age category")
```



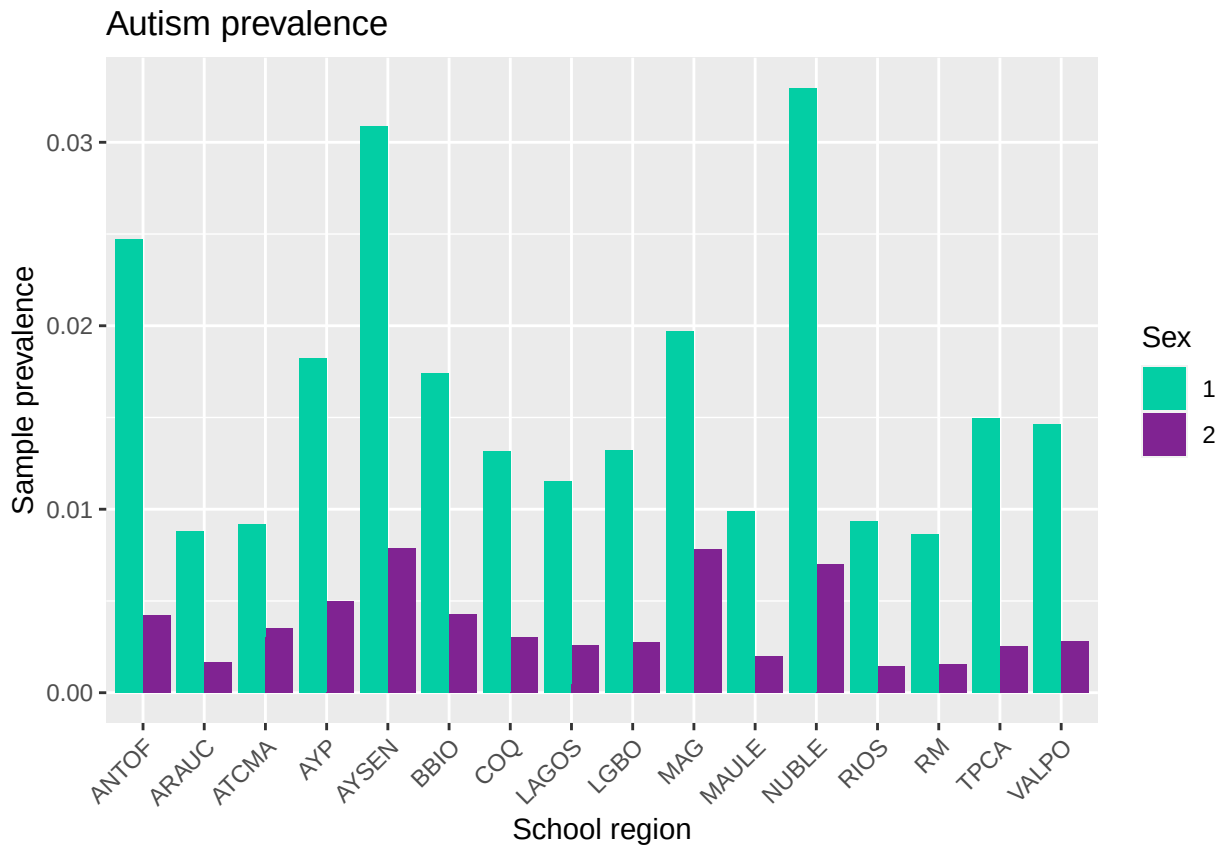

```
ggplot(data = aut_prev_region) +
  geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = age, fill = as.factor(age)), p
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "School region",
       y = "Sample prevalence",
       fill = "Age")
```




```
ggplot(data = aut_prev_region) +
  geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = age_cat_name, fill = age_cat_
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~sex_desc) +
  labs(title = "Autism prevalence by sex",
       x = "School region",
       y = "Sample prevalence",
       fill = "Age category")
```

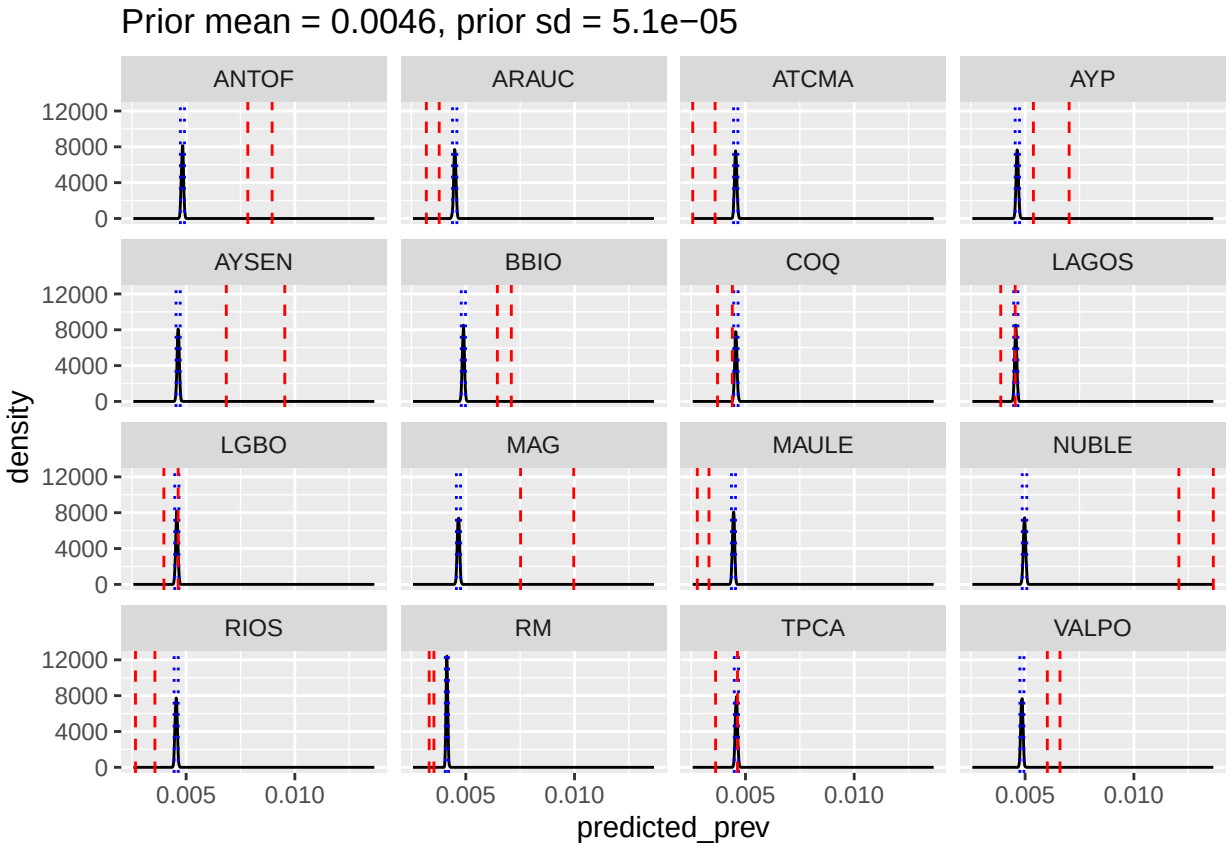



```
ggplot(data = aut_prev_region) +
  geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = sex, fill = as.factor(sex)), p
    # 1 is male, 2 is female
  #scale_fill_manual(values = wes_palette("GrandBudapest1")) +
  scale_fill_manual(values = c("#03CEA4", "#802392")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "School region",
       y = "Sample prevalence",
       fill = "Sex")
```




```
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 = 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 =
```

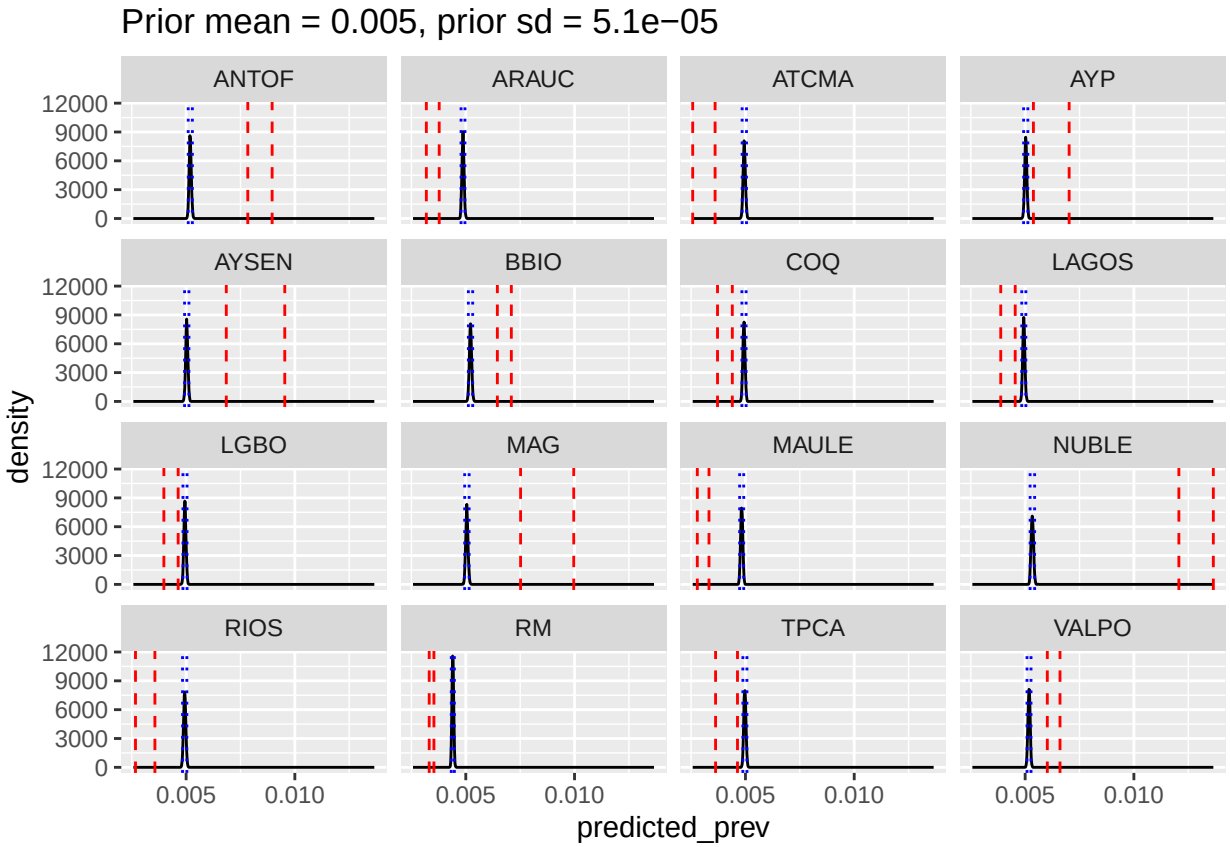
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



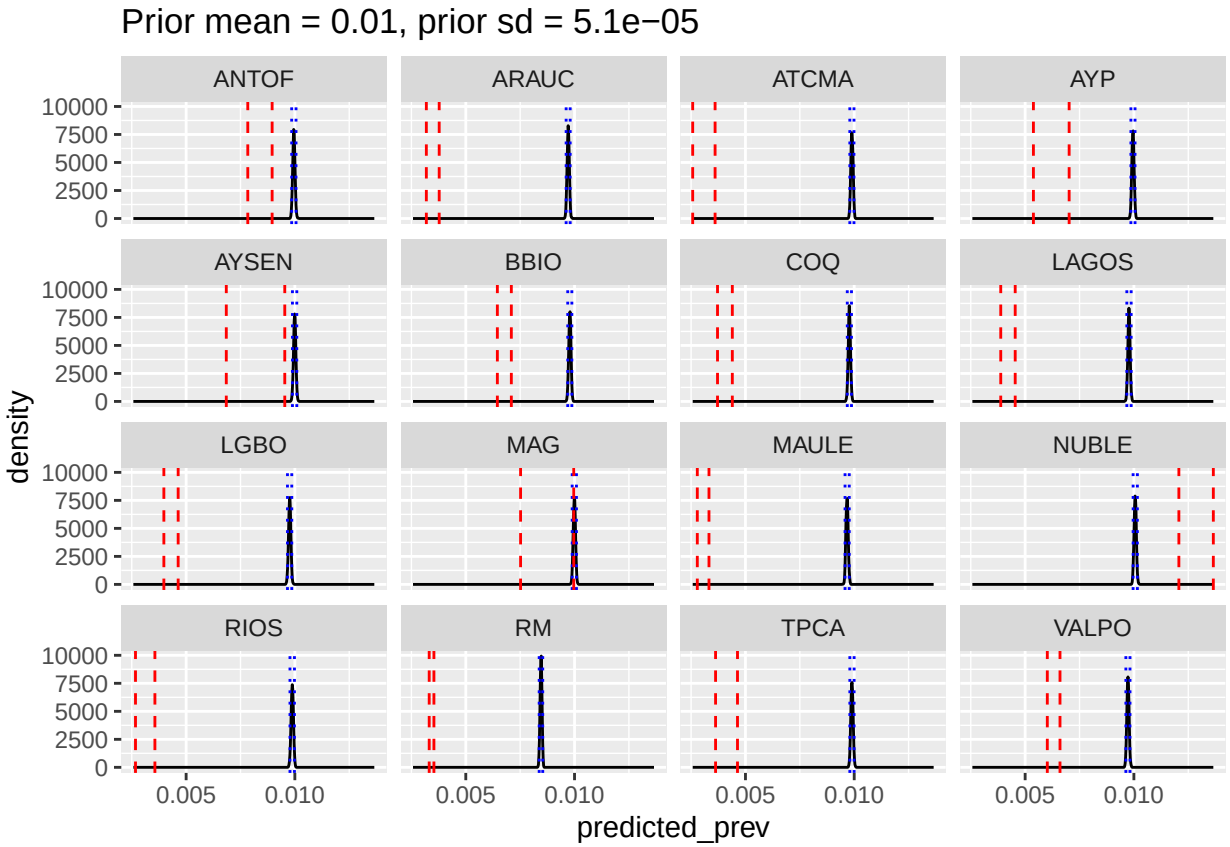
Predictions for higher population prevalence - increase prior mean

```
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])

}
```

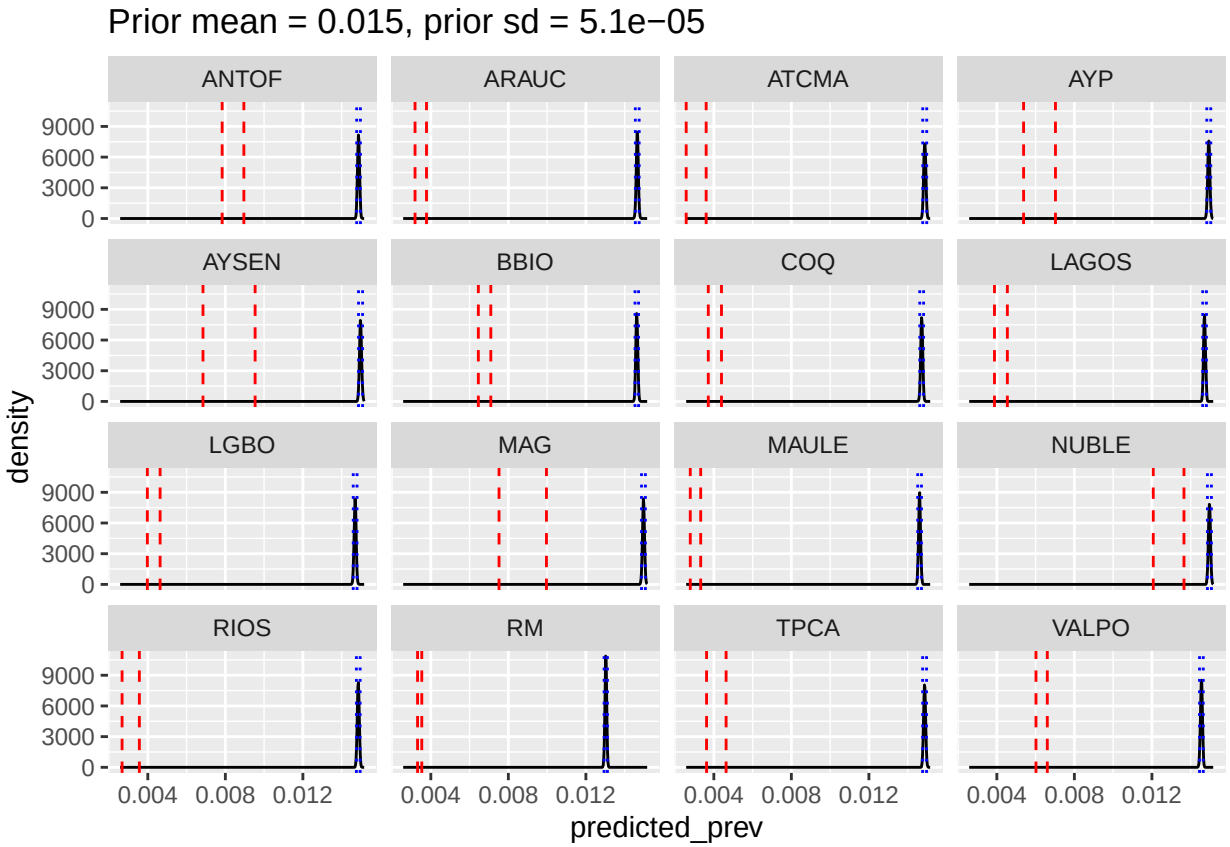
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



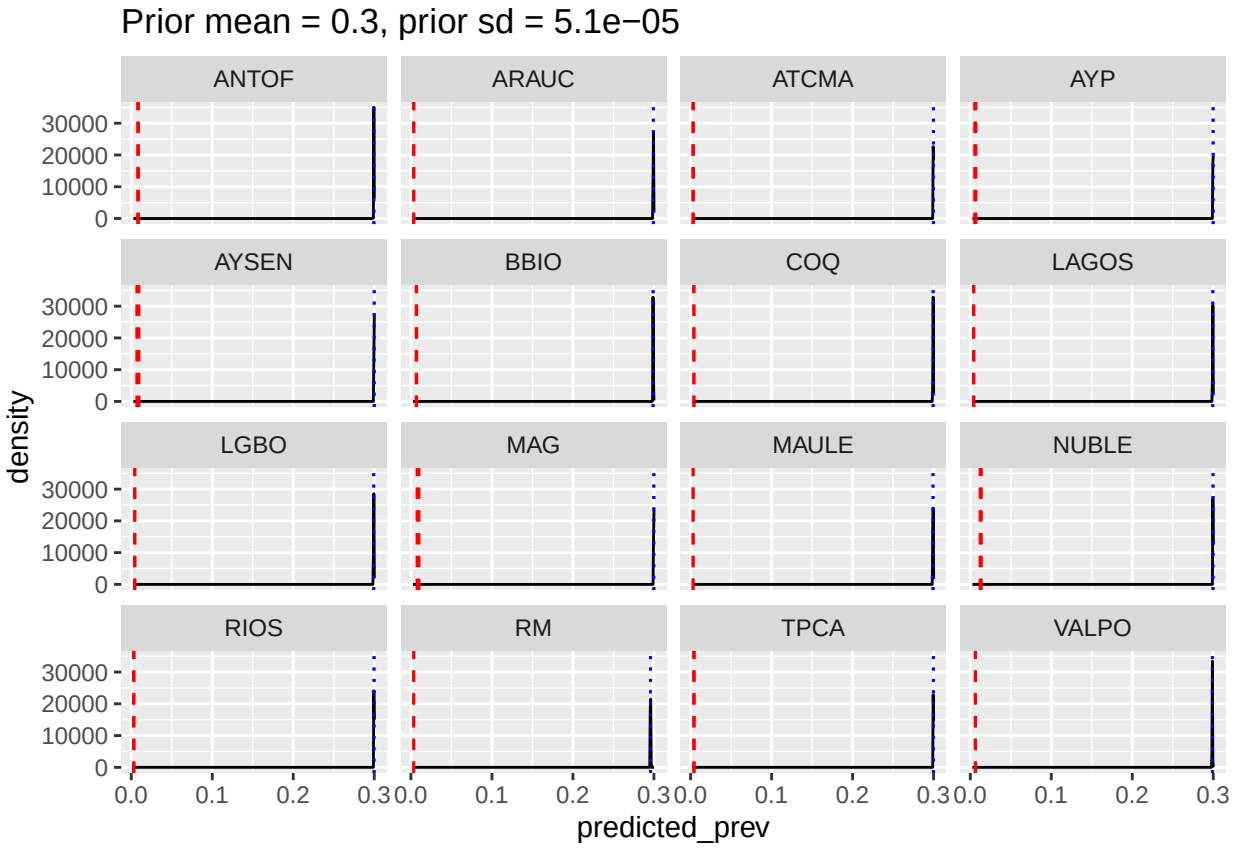

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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



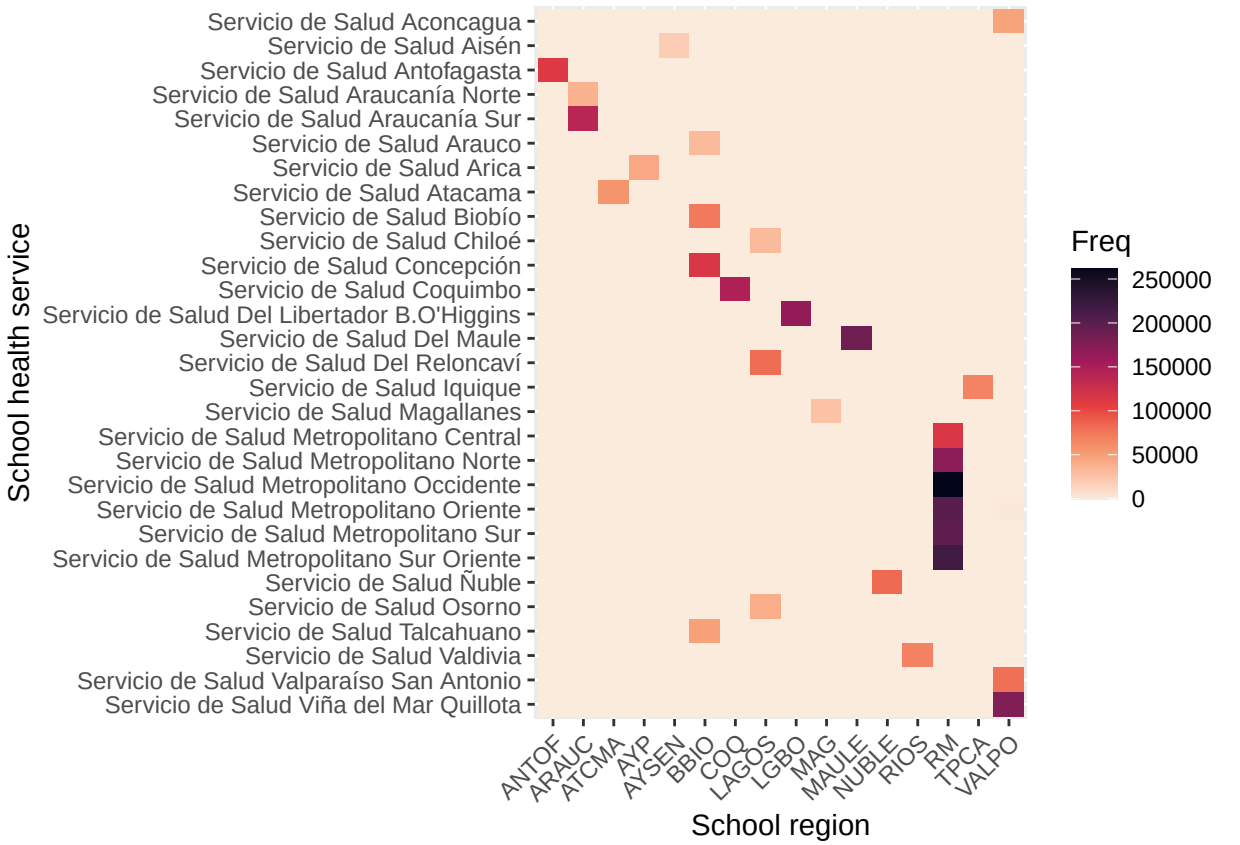

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



Random effect on health service

Show which regions each health service operates in

```
map_tab_df <- as.data.frame(table(chile_bayes_aut$school_region_name_abr, chile_bayes_aut$health_servic
ggplot(map_tab_df, aes(x = Var1, y = Var2, fill = Freq)) +
  geom_tile() +
  #scale_fill_gradient(low = "white", high = "blue") +
  scale_fill_viridis_c(option = "rocket", direction = -1) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_discrete(limits = rev(levels(map_tab_df$Var2))) +
  labs(x = "School region", y = "School health service", fill = "Freq")
```

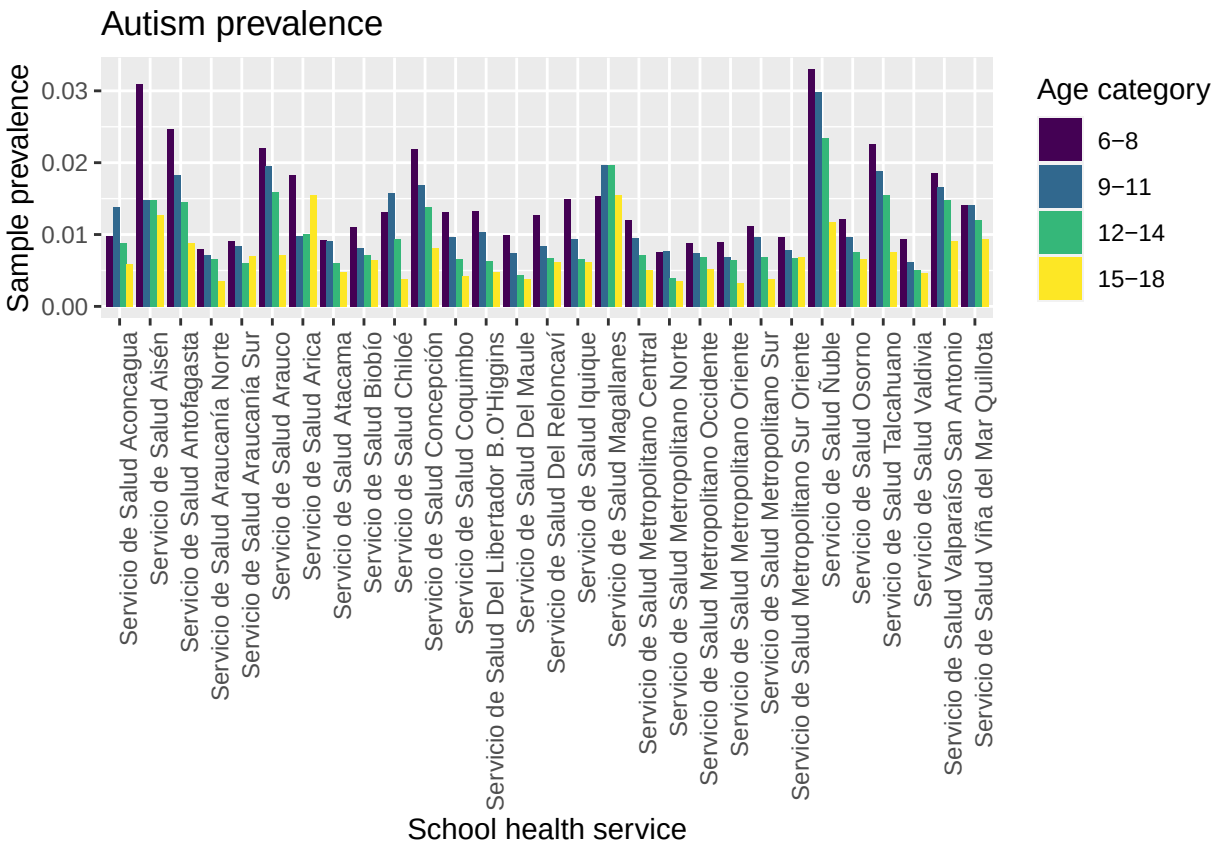


Each health service operates in only one region, some regions (RM, VALPO, LAGOS, BBIO, ARAUC) have multiple health services.

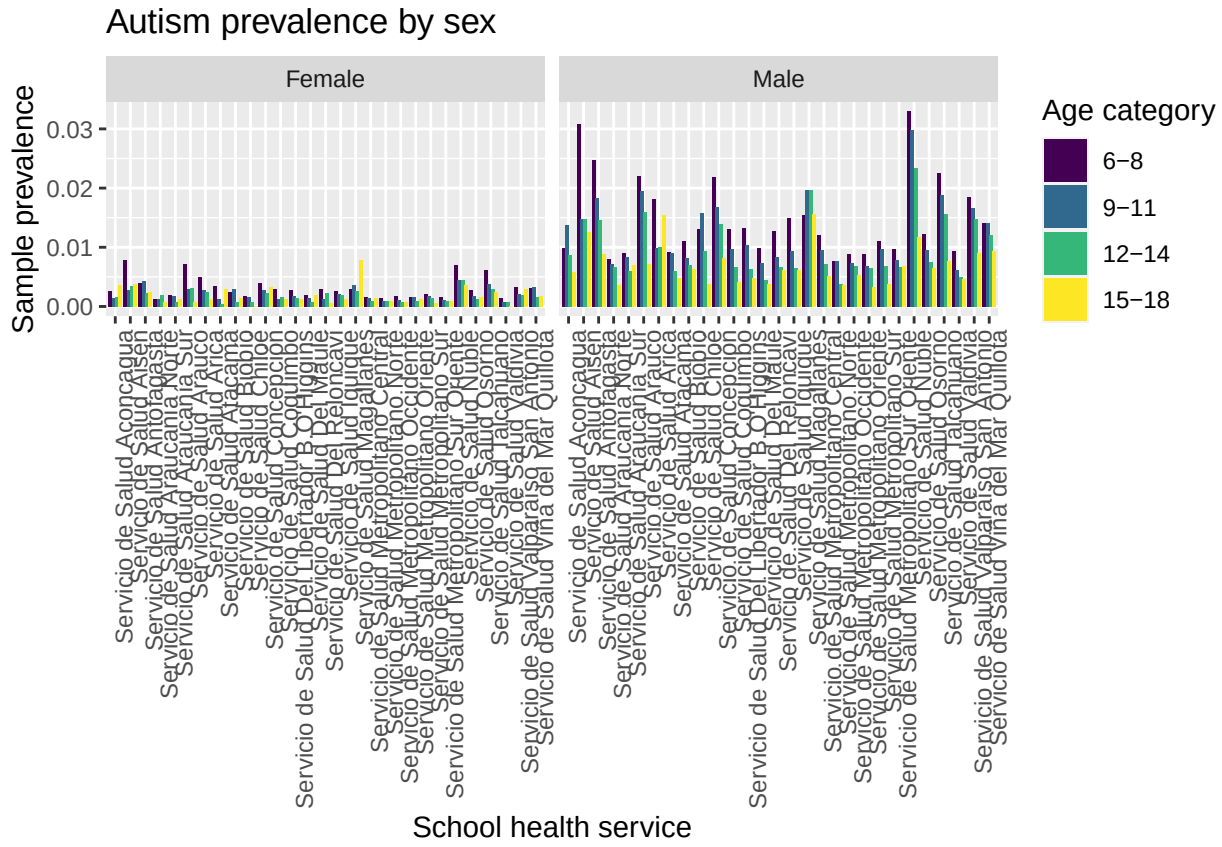
```
aut_prev_health <- get_grouped_prev(x = chile_bayes_aut, stdpop = chile_stdpop,
                                    grouping_vars = c("health_service_name", "age_june30", "age_cat_name
```

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

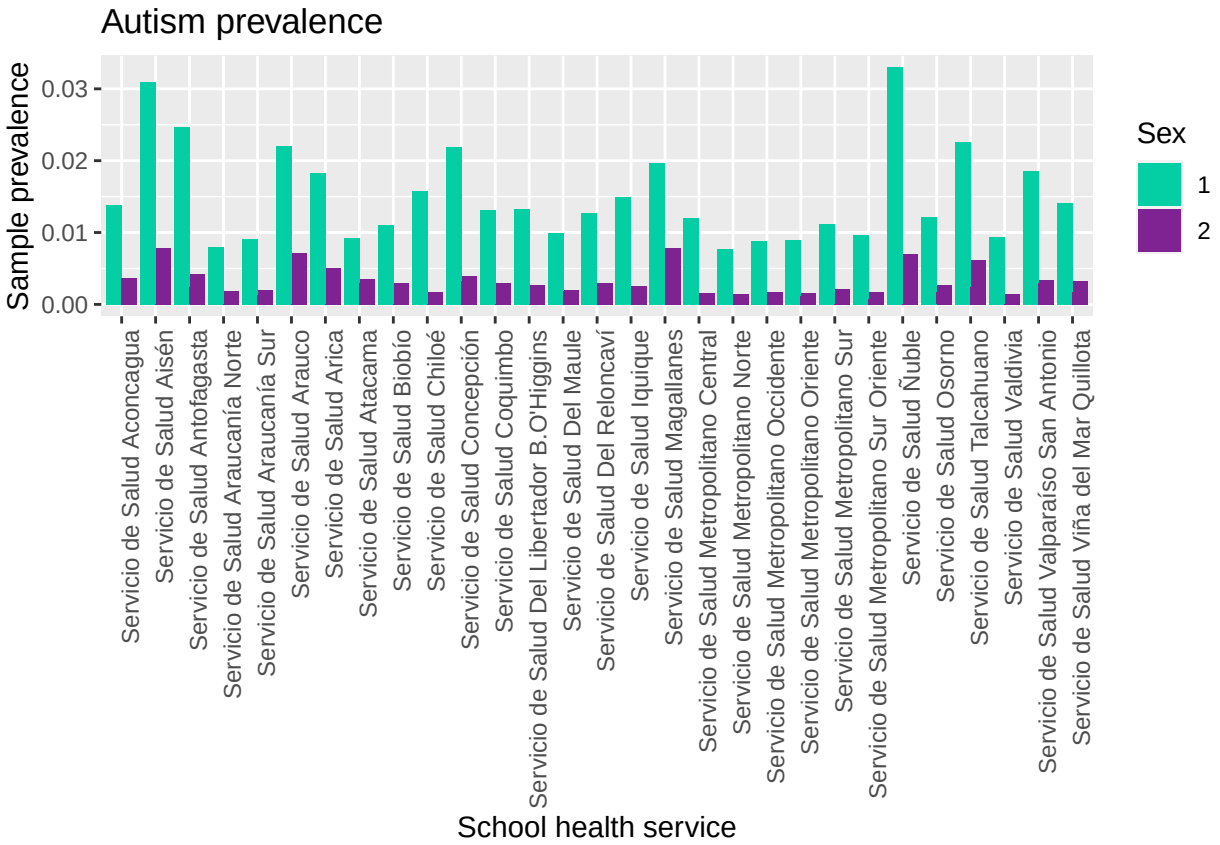
```
ggplot(data = aut_prev_health) +
  geom_col(aes(x = health_service_name, y = sample_prevalence, group = age_cat_name, fill = age_cat_name
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "School health service",
       y = "Sample prevalence",
       fill = "Age category")
```




```
ggplot(data = aut_prev_health) +
  geom_col(aes(x = health_service_name, y = sample_prevalence, group = age_cat_name, fill = age_cat_name
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_wrap(~sex_desc) +
  labs(title = "Autism prevalence by sex",
       x = "School health service",
       y = "Sample prevalence",
       fill = "Age category")
```

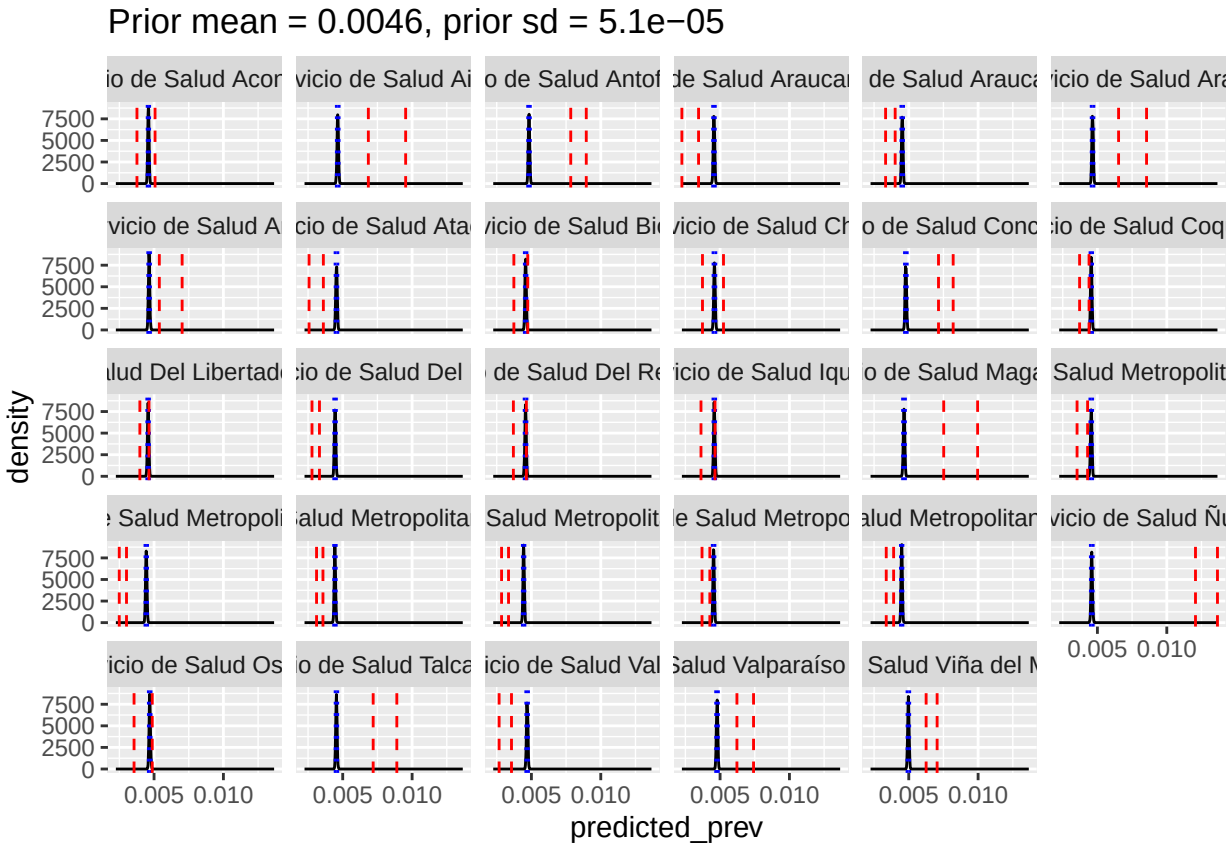



```
ggplot(data = aut_prev_health) +
  geom_col(aes(x = health_service_name, y = sample_prevalence, group = sex, fill = as.factor(sex)), posi
    # 1 is male, 2 is female
  scale_fill_manual(values = c("#03CEA4", "#802392")) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "School health service",
       y = "Sample prevalence",
       fill = "Sex")
```




```
aut_prev_health_adj <- get_adjusted_prev(aut_prev_health, grouping_vars = "health_service_name")

aut_prev_health_post <- do_jags_rand_model(x = aut_prev_health_adj,
                                feat = "health_service_name",
                                model = rand_model,
                                theta_mu = theta_mu_prior,
                                theta_sigma = theta_sigma_prior,
                                pars = pars,
                                convergence_checks = FALSE) %>%
  rename("health_service_name" = "Feat_names")

plot_post_density(aut_prev_health_post, aut_prev_health_adj, feat = "health_service_name", theta_mu = tl
```

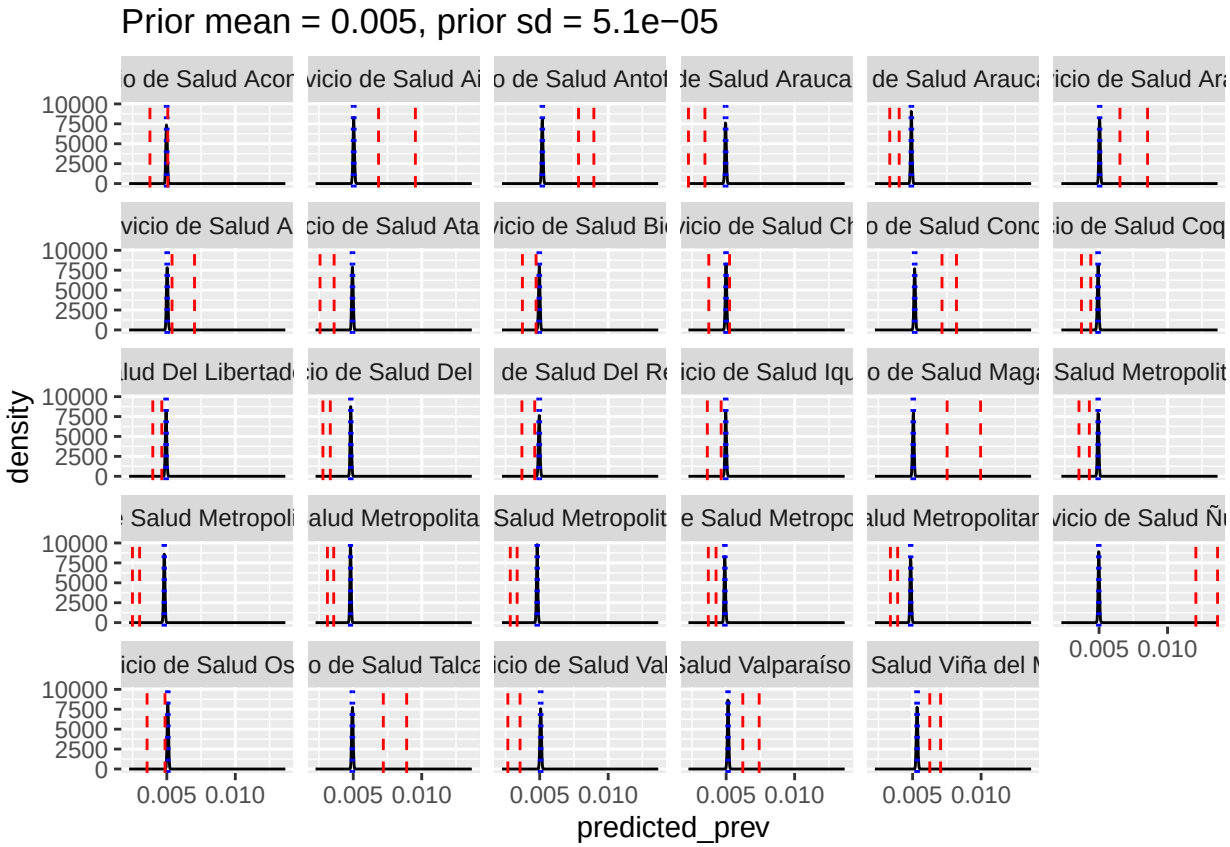
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



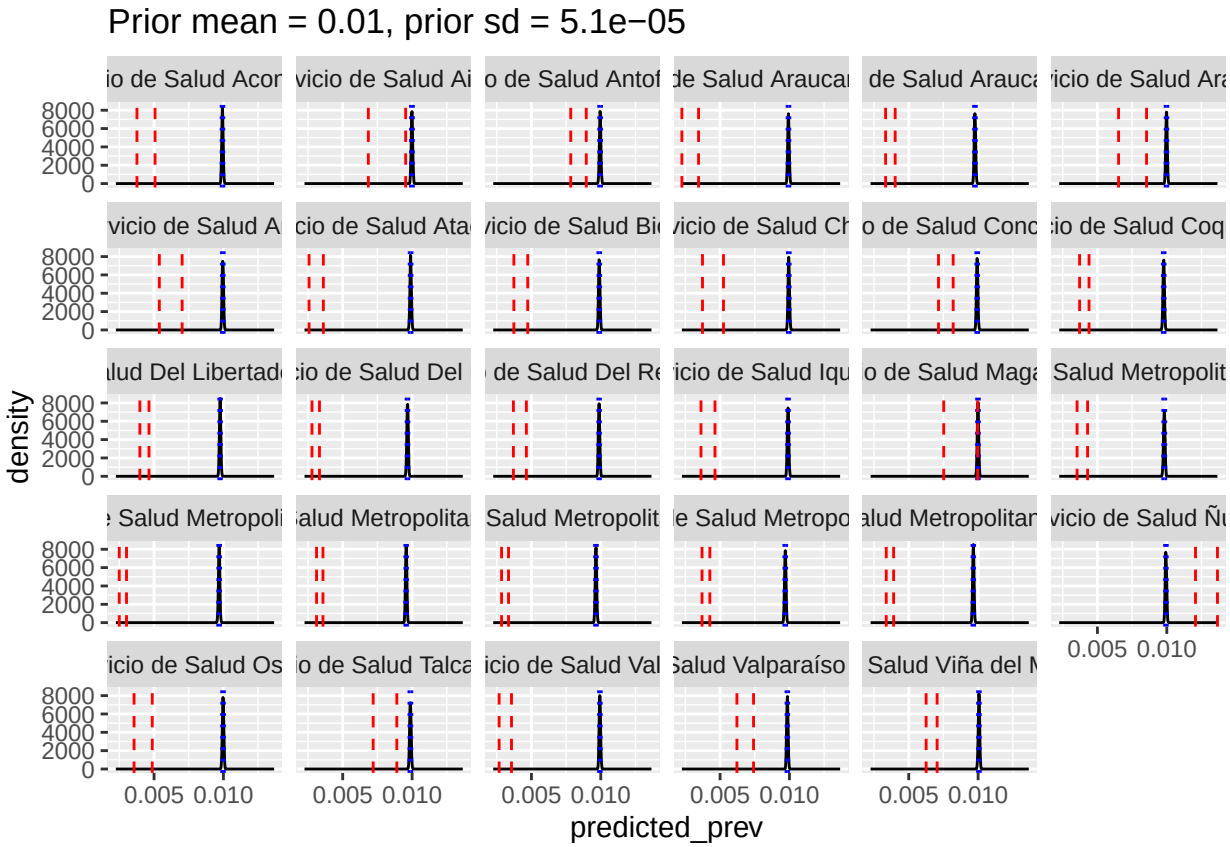
Predictions for higher population prevalence - increase prior mean

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

  plot_post_density(aut_prev_health_post,
                    aut_prev_health_adj,
                    feat = "health_service_name",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])

}
```

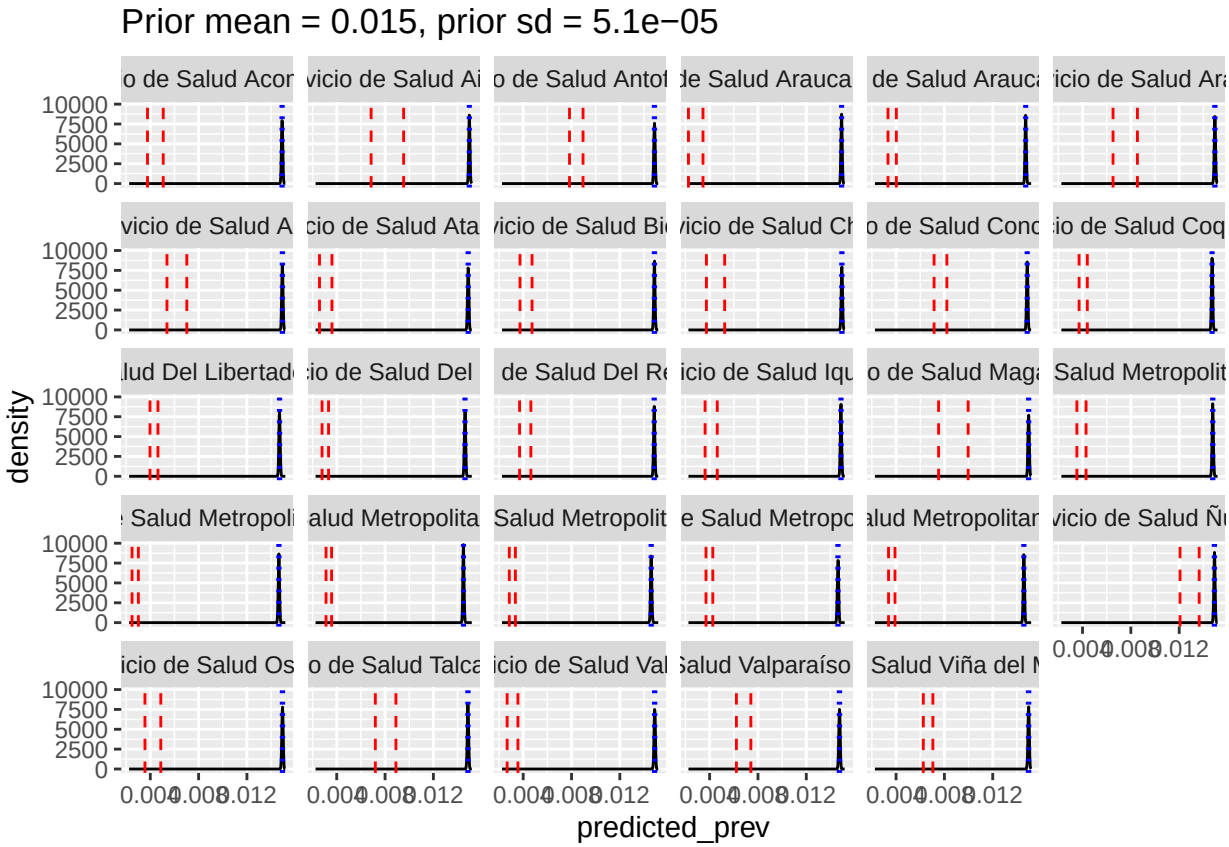
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



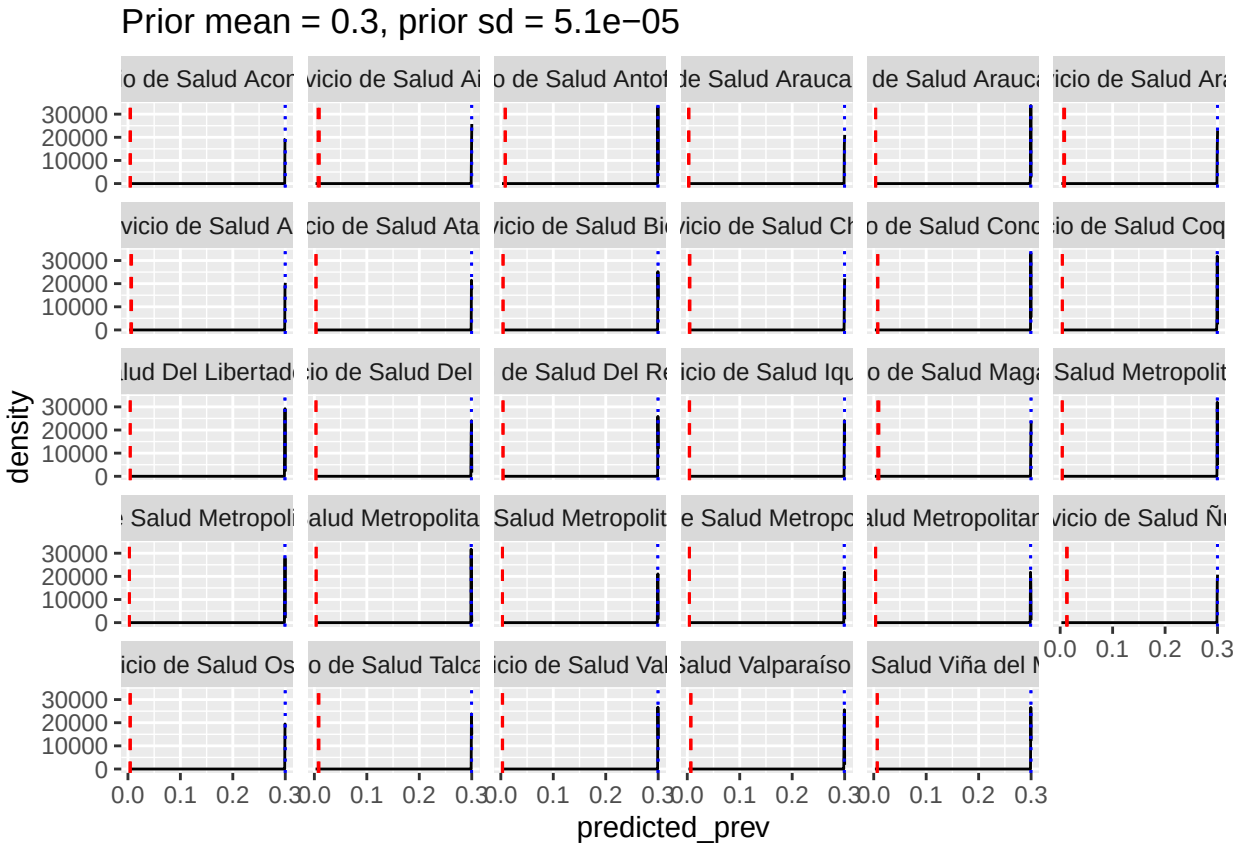

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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




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

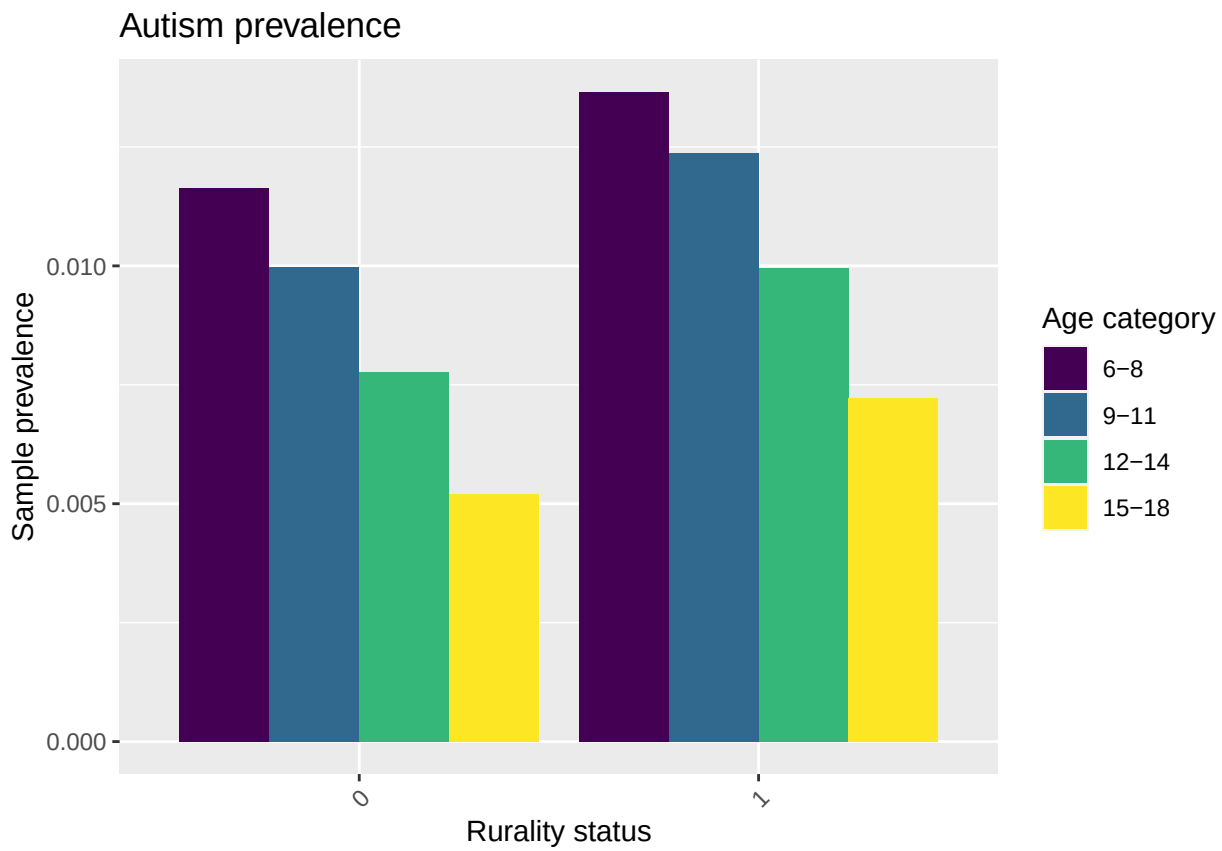


Random effect on rurality

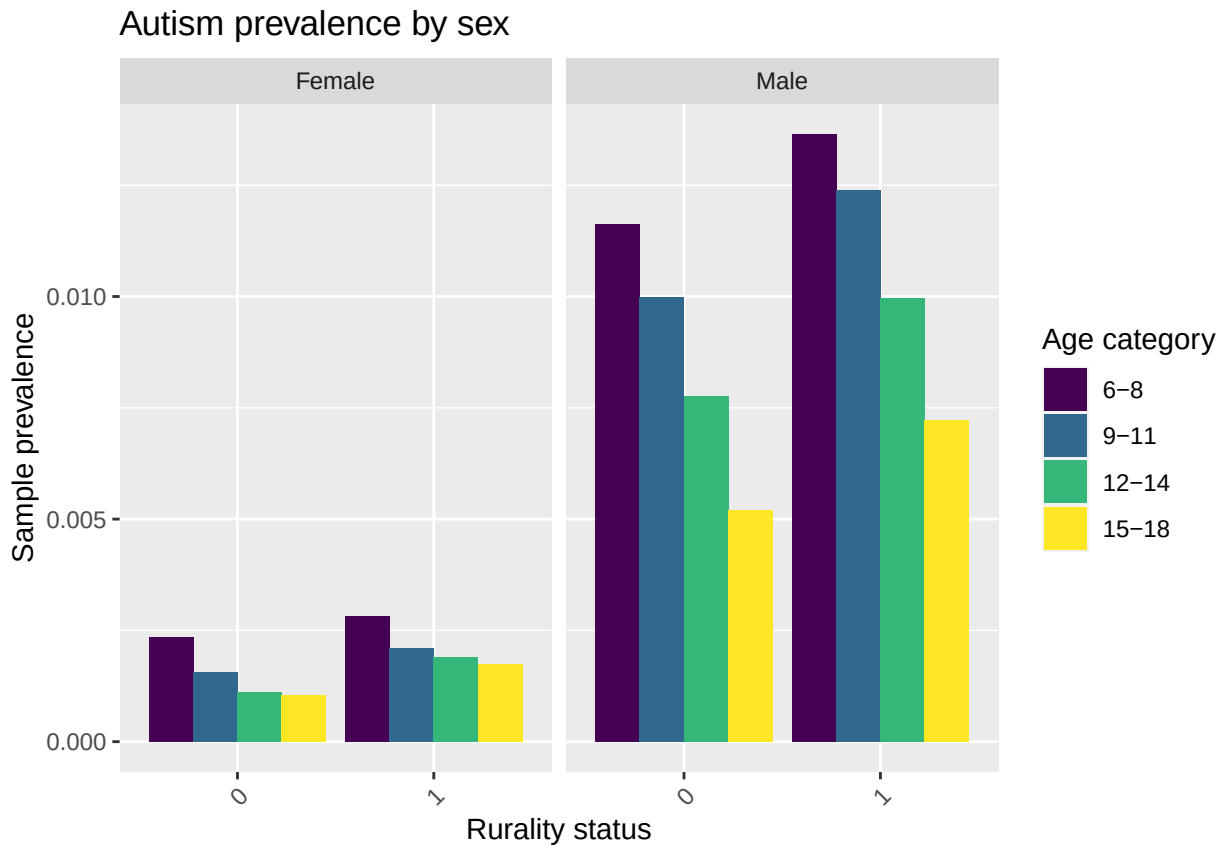
```
aut_prev_rural <- get_grouped_prev(x = chile_bayes_aut, stdpop = chile_stdpop,
                                   grouping_vars = c("school_rurality_code", "age_june30", "age_cat_nam
```

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

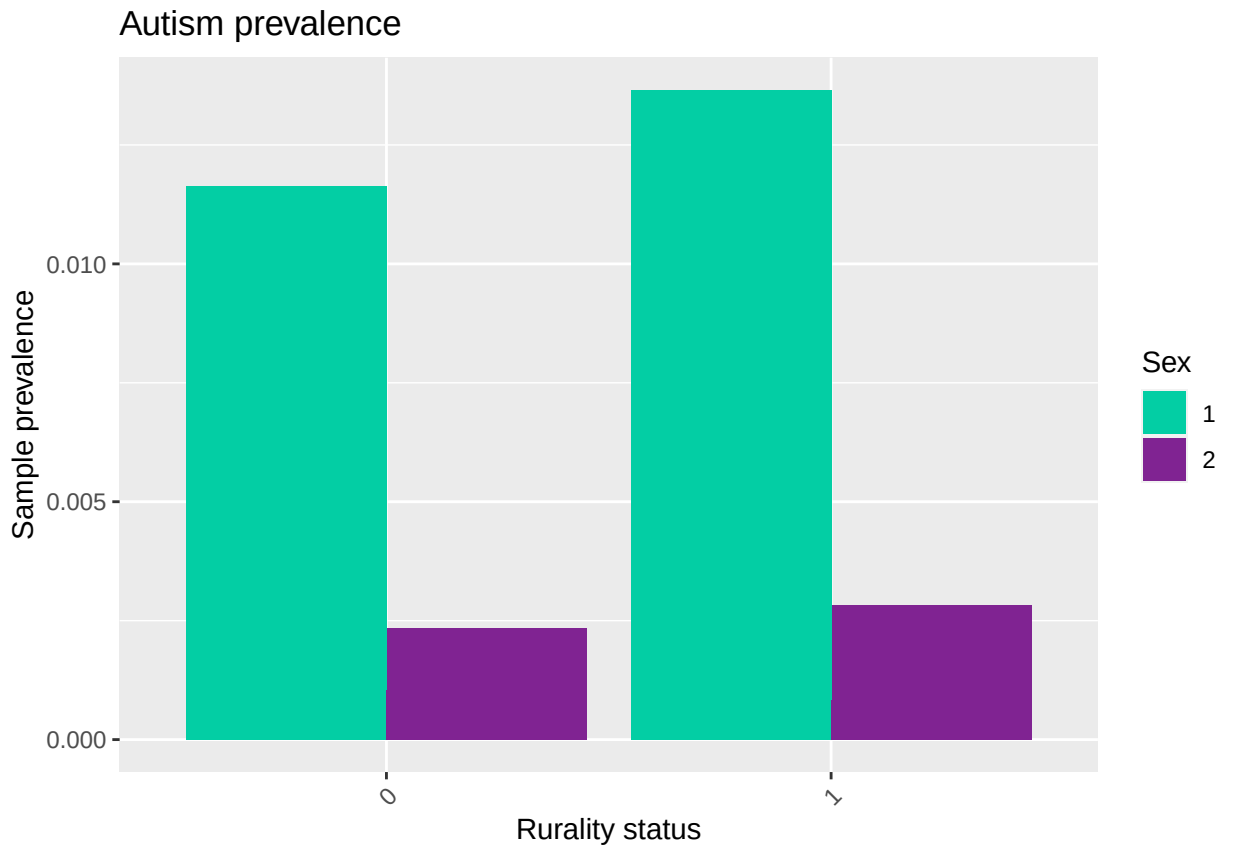
```
ggplot(data = aut_prev_rural) +
  geom_col(aes(x = as.factor(school_rurality_code), y = sample_prevalence, group = age_cat_name, fill =
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Rurality status",
       y = "Sample prevalence",
       fill = "Age category")
```




```
ggplot(data = aut_prev_rural) +
  geom_col(aes(x = as.factor(school_rurality_code), y = sample_prevalence, group = age_cat_name, fill =
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~sex_desc) +
  labs(title = "Autism prevalence by sex",
       x = "Rurality status",
       y = "Sample prevalence",
       fill = "Age category")
```




```
ggplot(data = aut_prev_rural) +
  geom_col(aes(x = as.factor(school_rurality_code), y = sample_prevalence, group = sex, fill = as.factor
  scale_fill_manual(values = c("#03CEA4", "#802392")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Rurality status",
       y = "Sample prevalence",
       fill = "Sex")
```



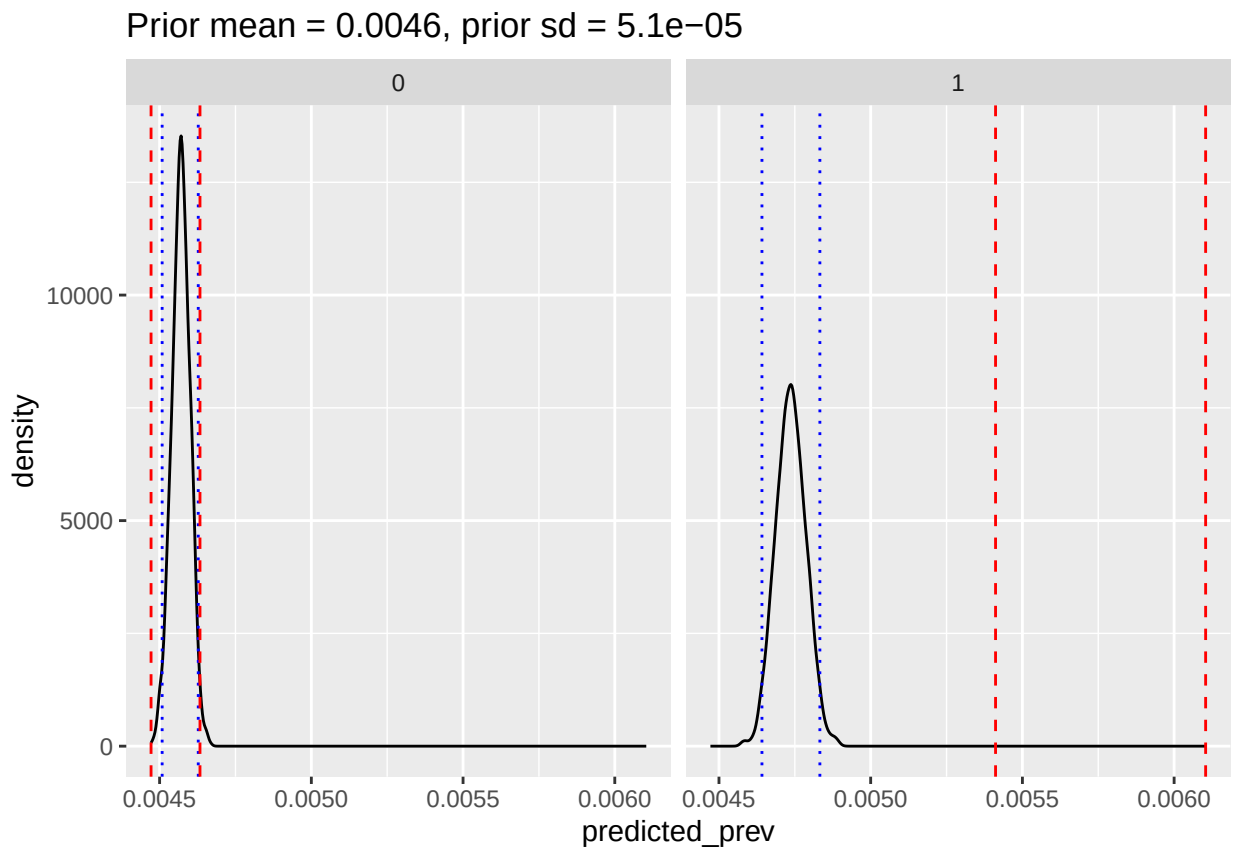

```
aut_prev_rural_adj <- get_adjusted_prev(aut_prev_rural, grouping_vars = "school_rurality_code")

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 = FALSE) %>%
  rename("school_rurality_code" = "Feat_names")

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.
```

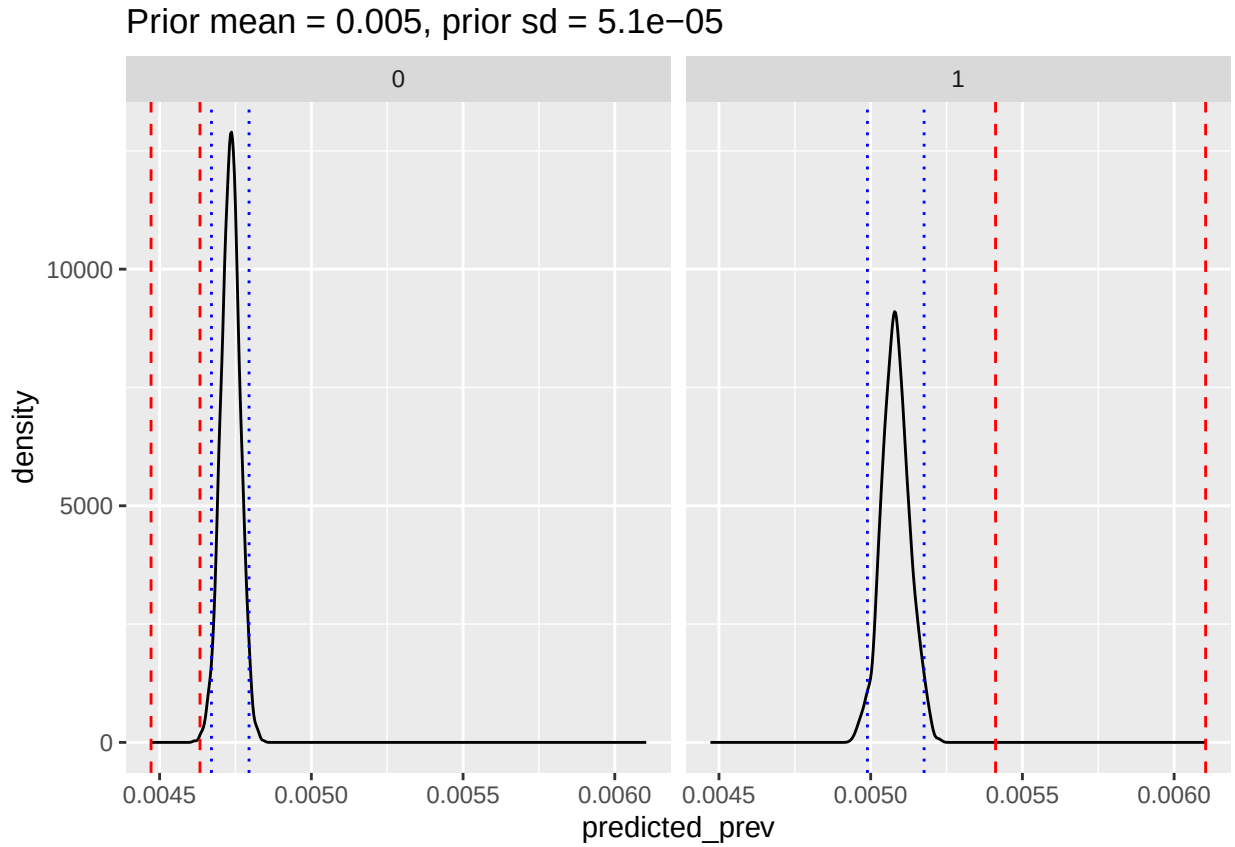


Assuming 0 = city, 1 = rural. Narrower sample CI for city because sample size is bigger

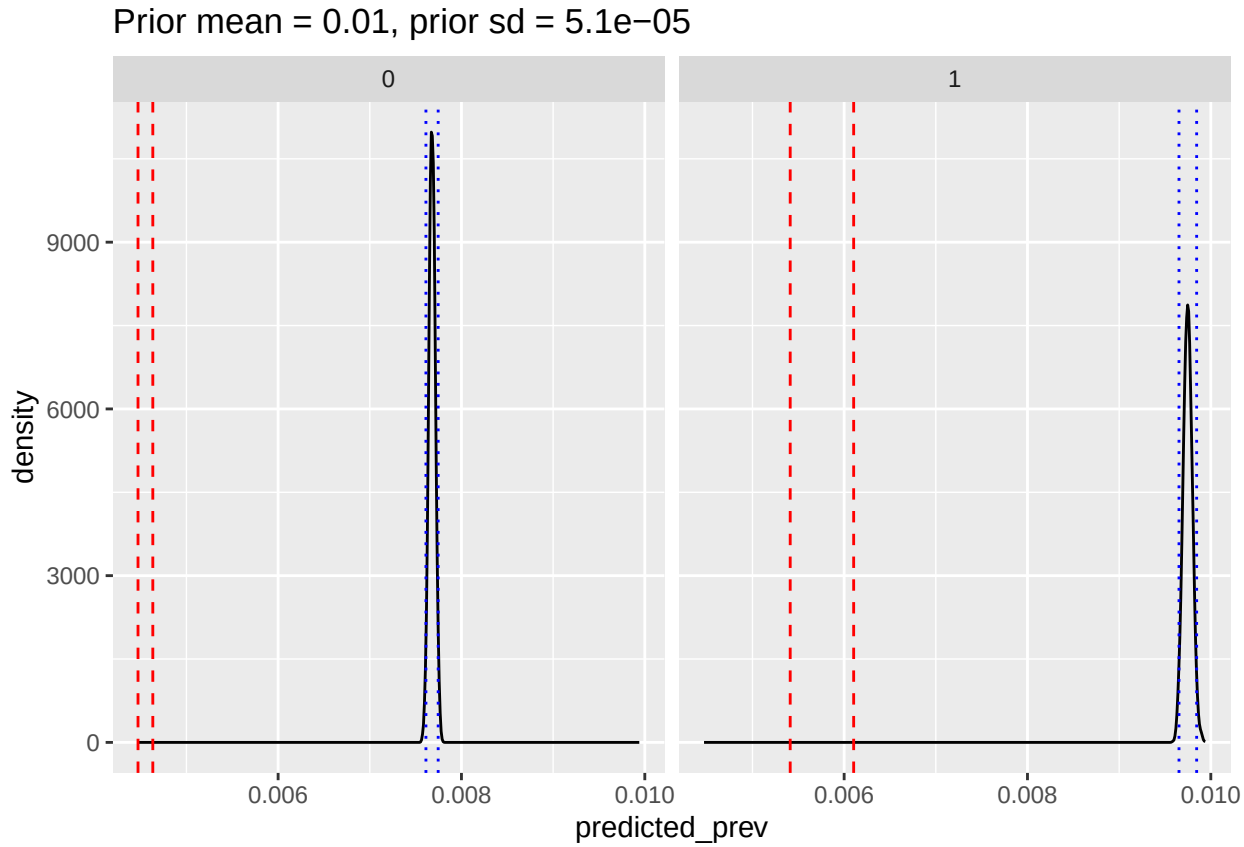
Predictions for higher population prevalence - increase prior mean

```
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])
}
```

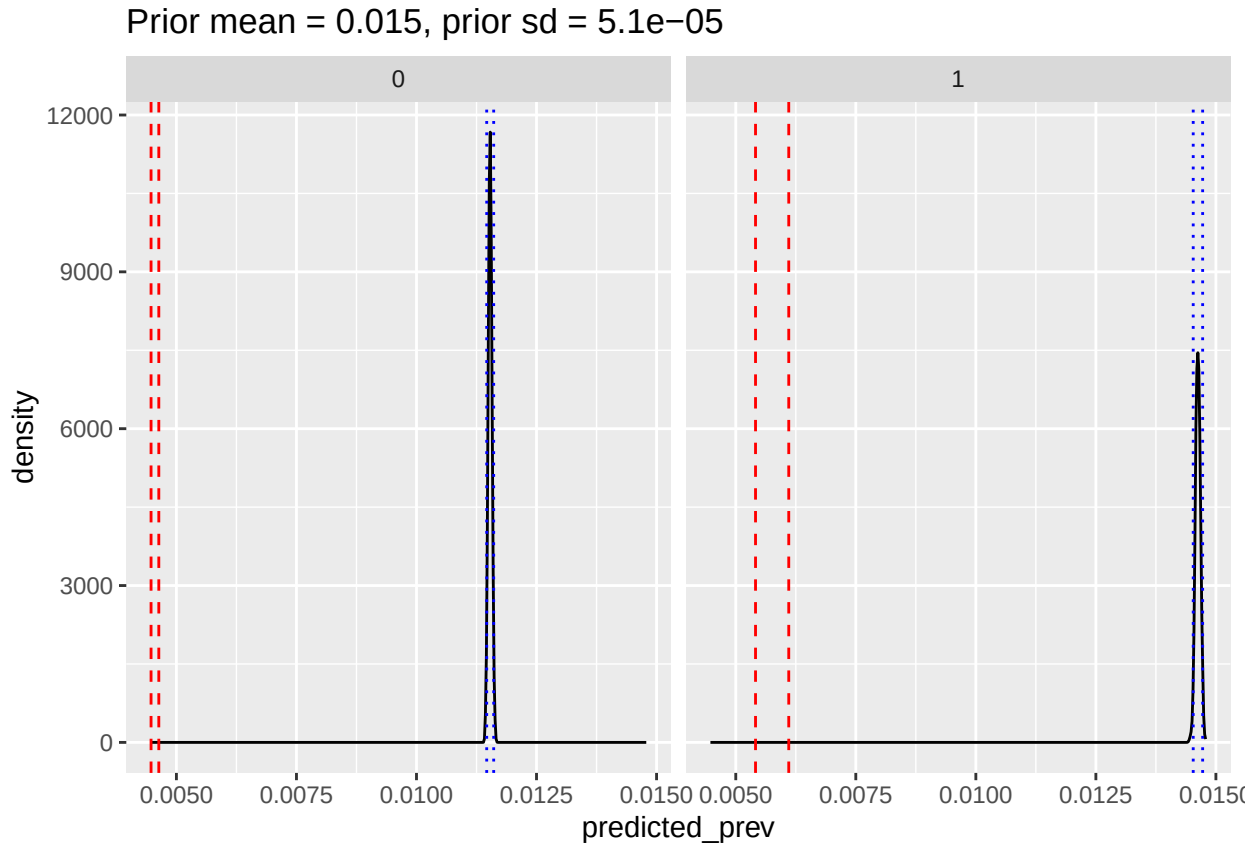
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



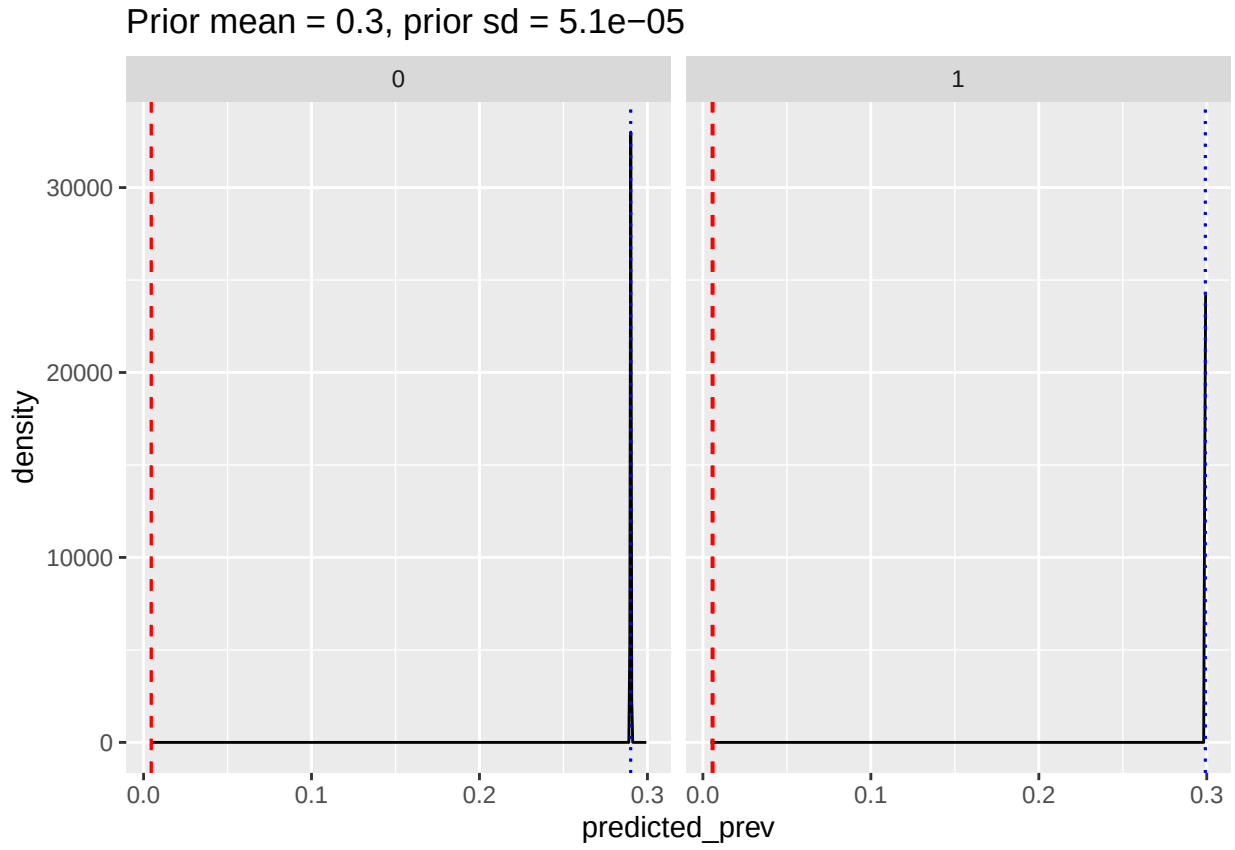

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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




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



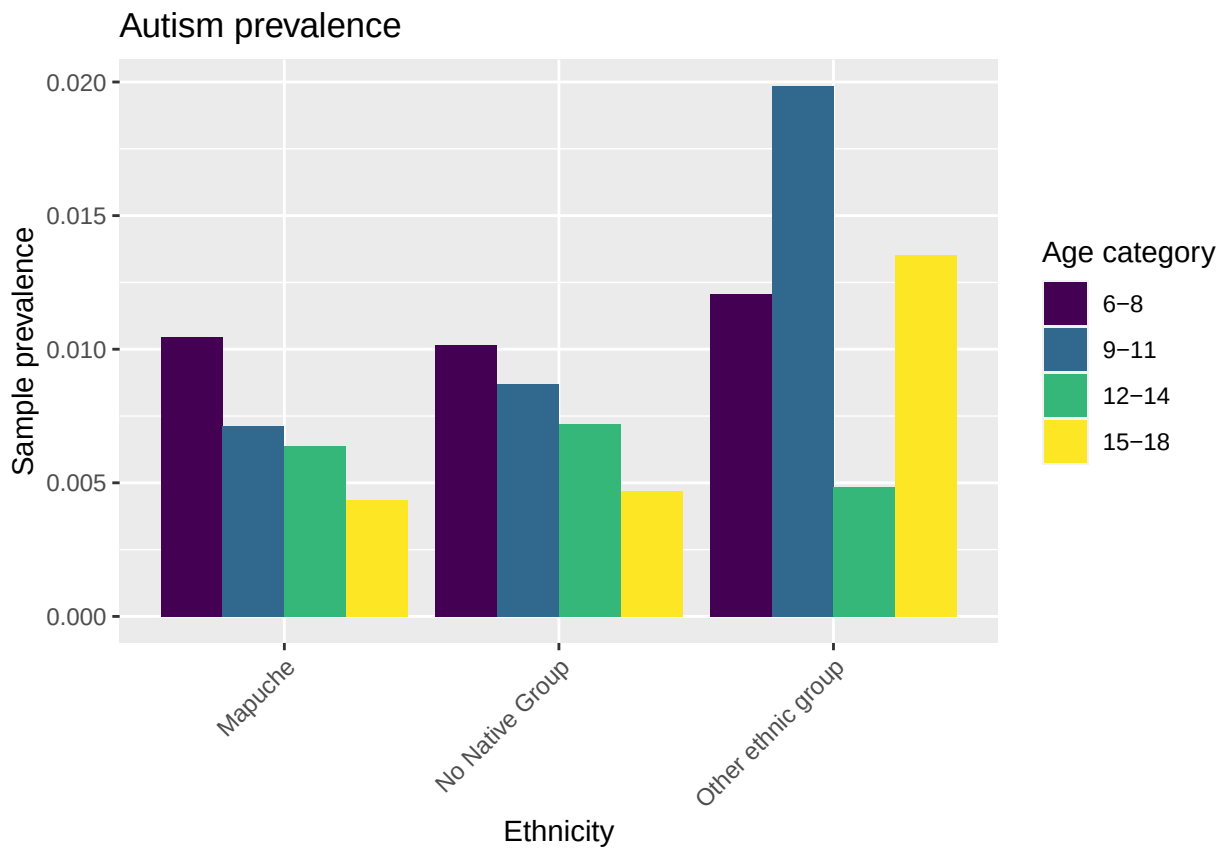
Random effect on ethnicity

Only use regions with large Mapuche populations

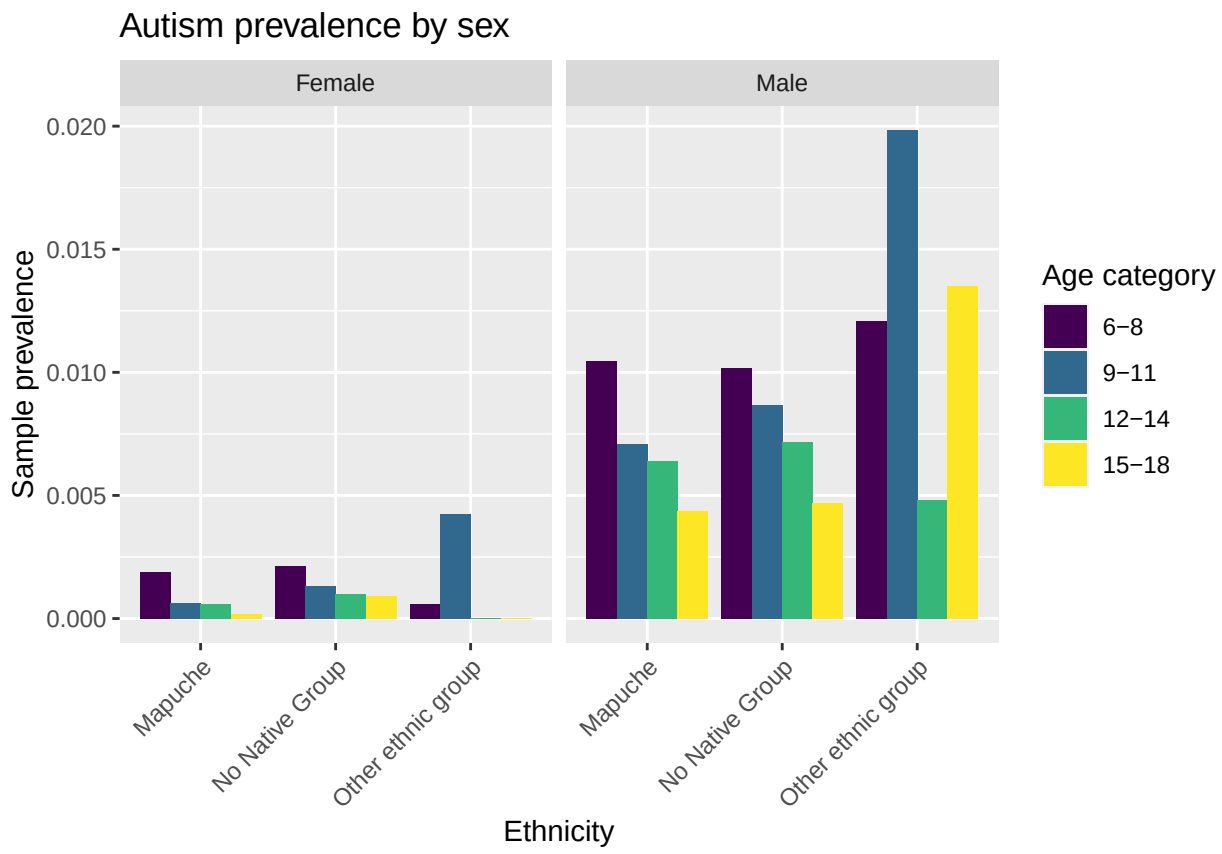
```
aut_prev_ethnic <- chile_bayes_aut %>%
  filter(school_region_name_abr %in% c("ARAUC", "BBIO", "LAGOS", "RIOS", "RM")) %>%
  get_grouped_prev(stdpop = chile_stdpop,
                   grouping_vars = c("ethnic_2_group", "age_june30", "age_cat_name", "sex", "sex_desc",
```

```
## `summarise()` has grouped output by 'ethnic_2_group', 'age_june30',
## 'age_cat_name', 'sex', 'sex_desc'. You can override using the `.groups`
## argument.
```

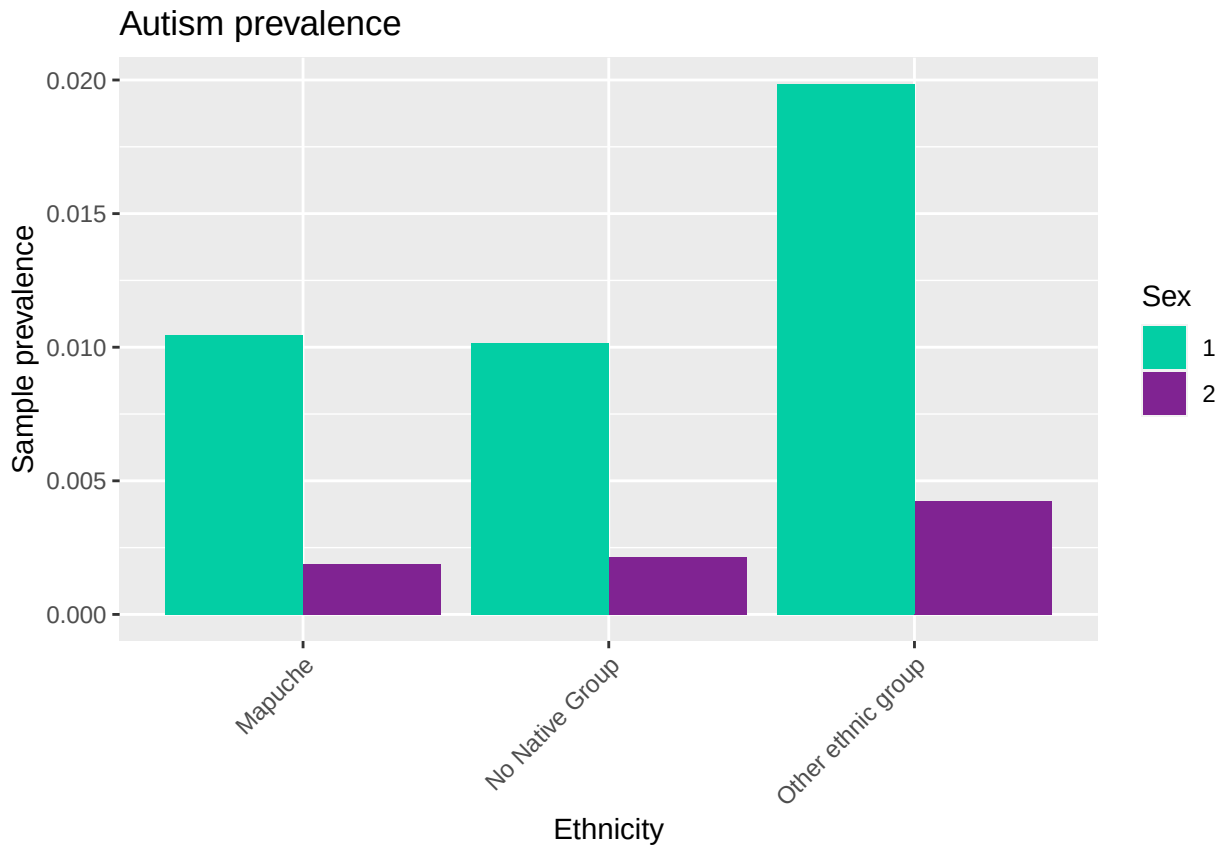
```
ggplot(data = aut_prev_ethnic) +
  geom_col(aes(x = as.factor(ethnic_2_group), y = sample_prevalence, group = age_cat_name, fill = age_ca
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Ethnicity",
       y = "Sample prevalence",
       fill = "Age category")
```




```
ggplot(data = aut_prev_ethnic) +
  geom_col(aes(x = as.factor(ethnic_2_group), y = sample_prevalence, group = age_cat_name, fill = age_ca
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~sex_desc) +
  labs(title = "Autism prevalence by sex",
       x = "Ethnicity",
       y = "Sample prevalence",
       fill = "Age category")
```

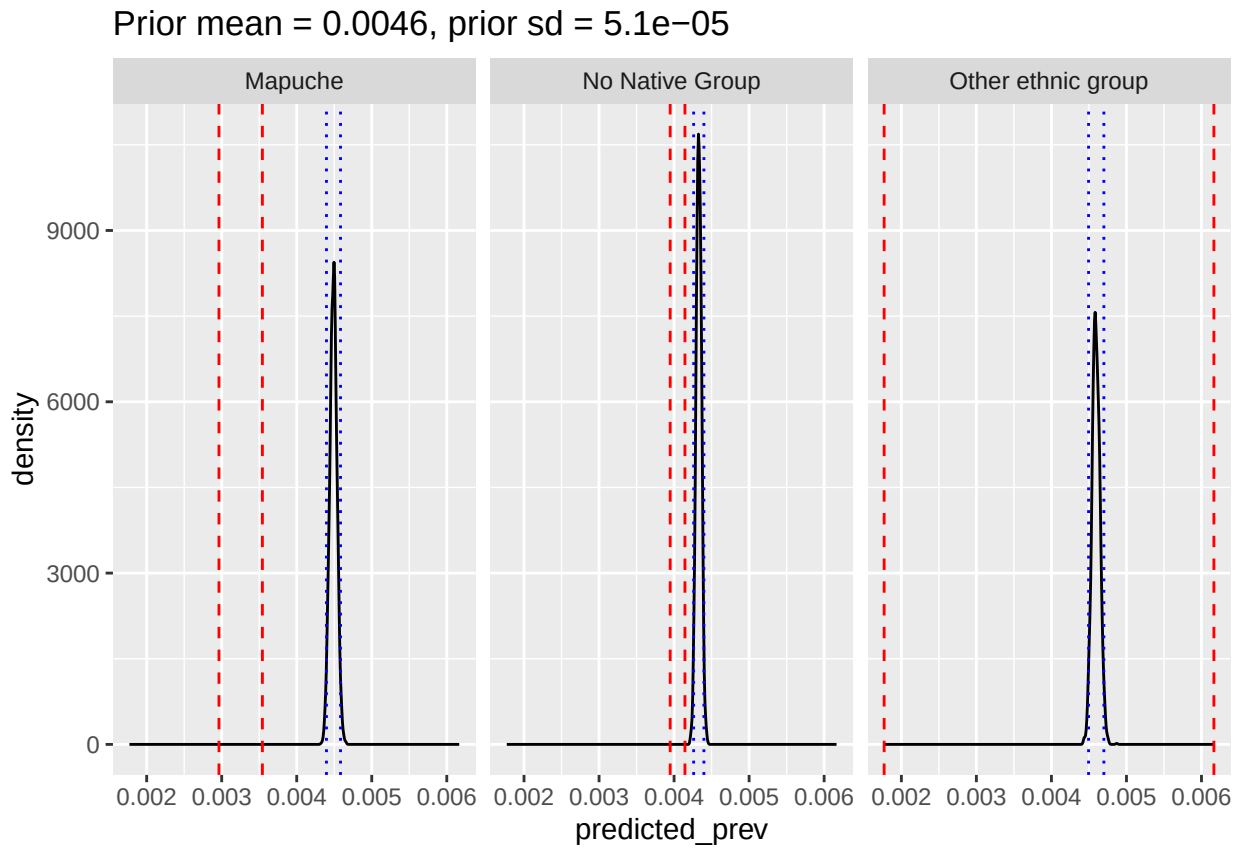



```
ggplot(data = aut_prev_ethnic) +
  geom_col(aes(x = as.factor(ethnic_2_group), y = sample_prevalence, group = sex, fill = as.factor(sex)),
  scale_fill_manual(values = c("#03CEA4", "#802392")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Ethnicity",
       y = "Sample prevalence",
       fill = "Sex")
```




```
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 = 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_
```

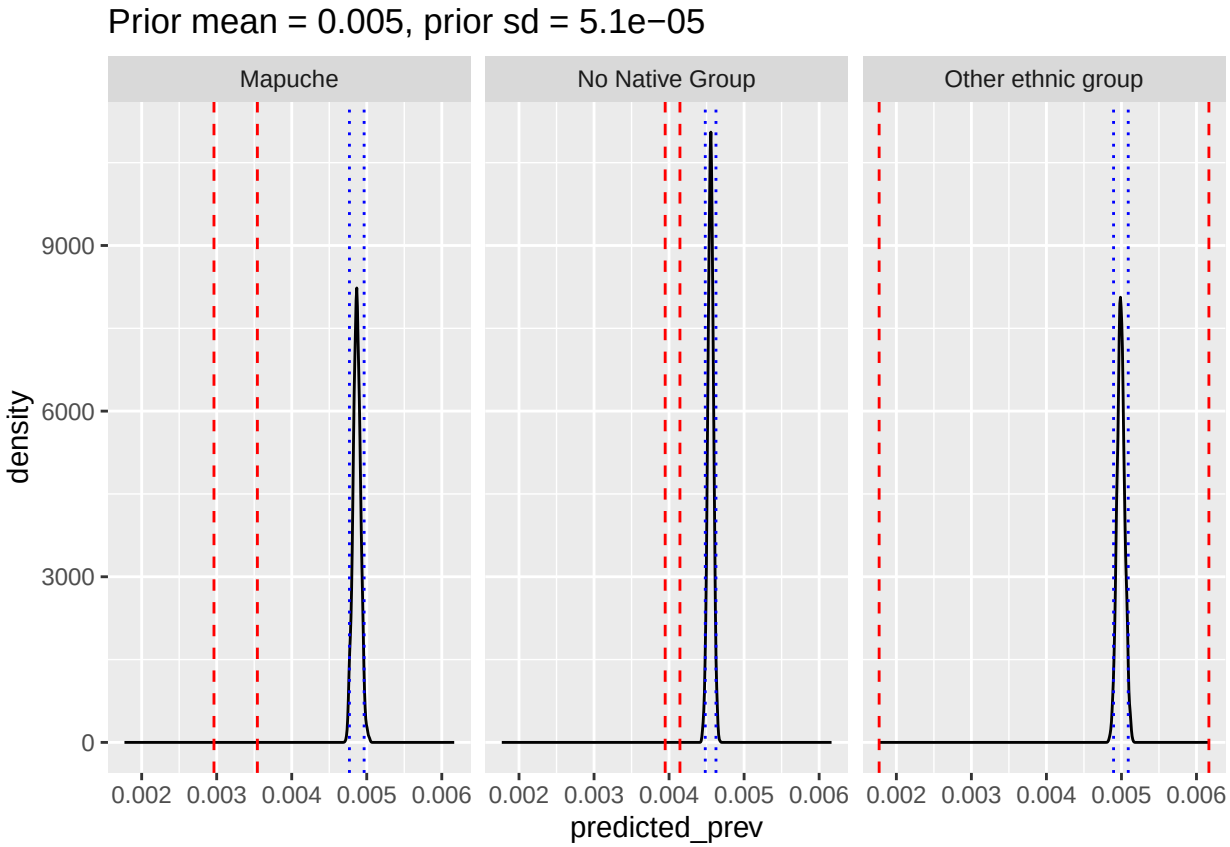
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



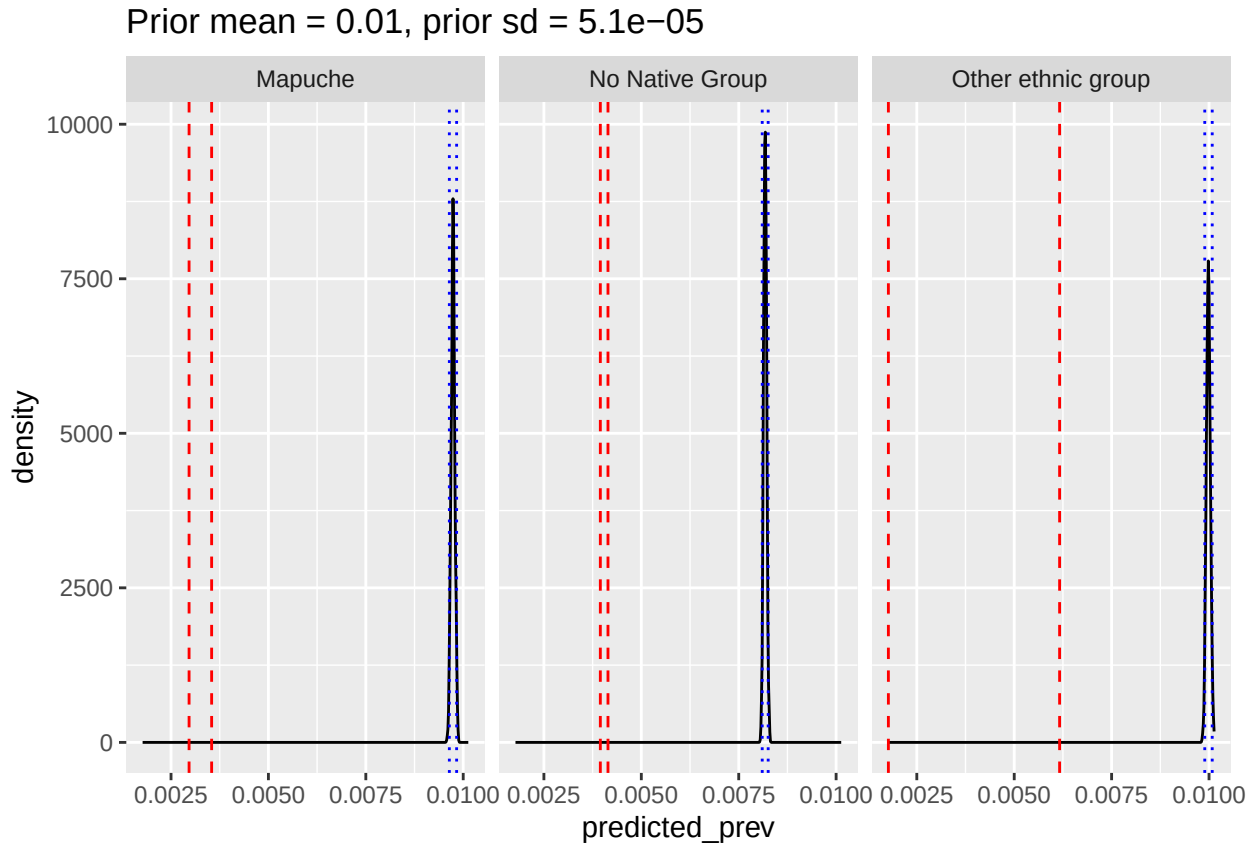
Predictions for higher population prevalence - increase prior mean

```
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])
}
```

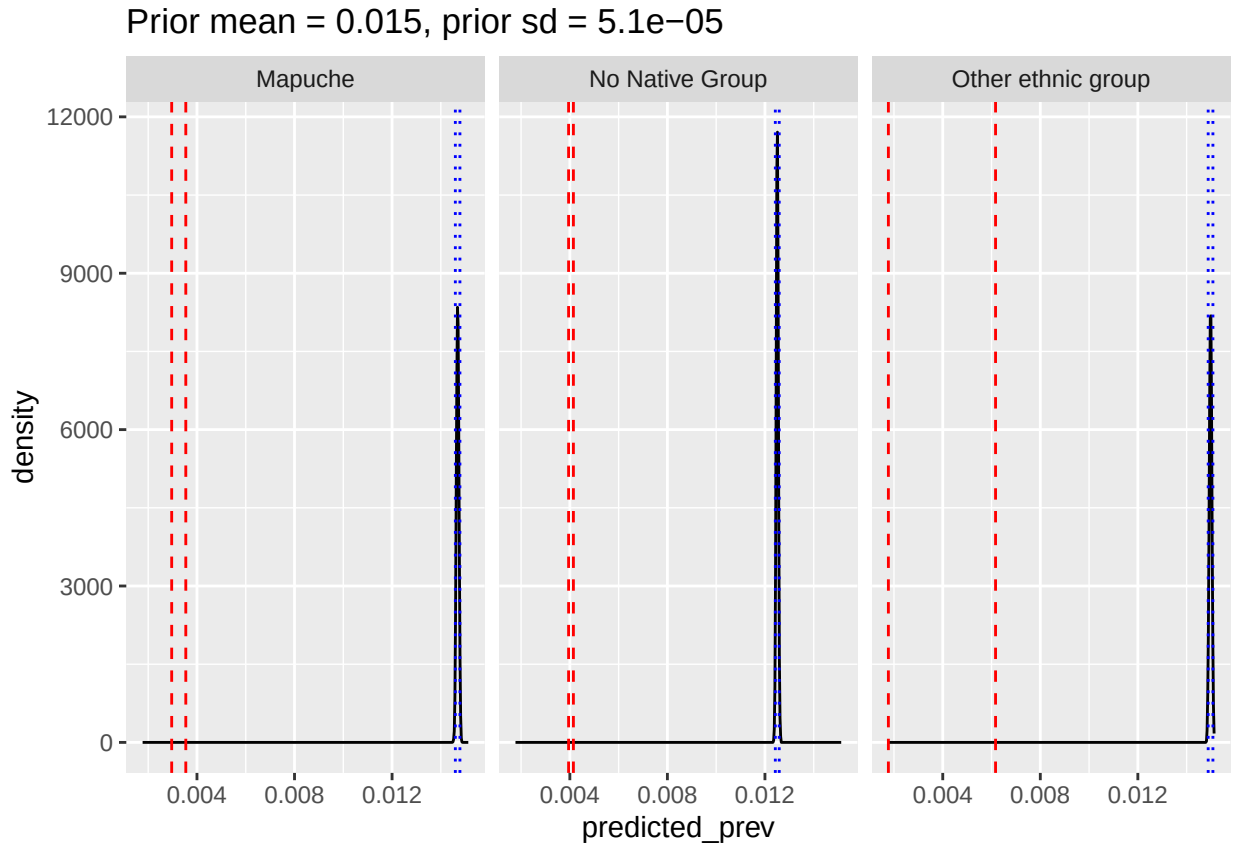
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



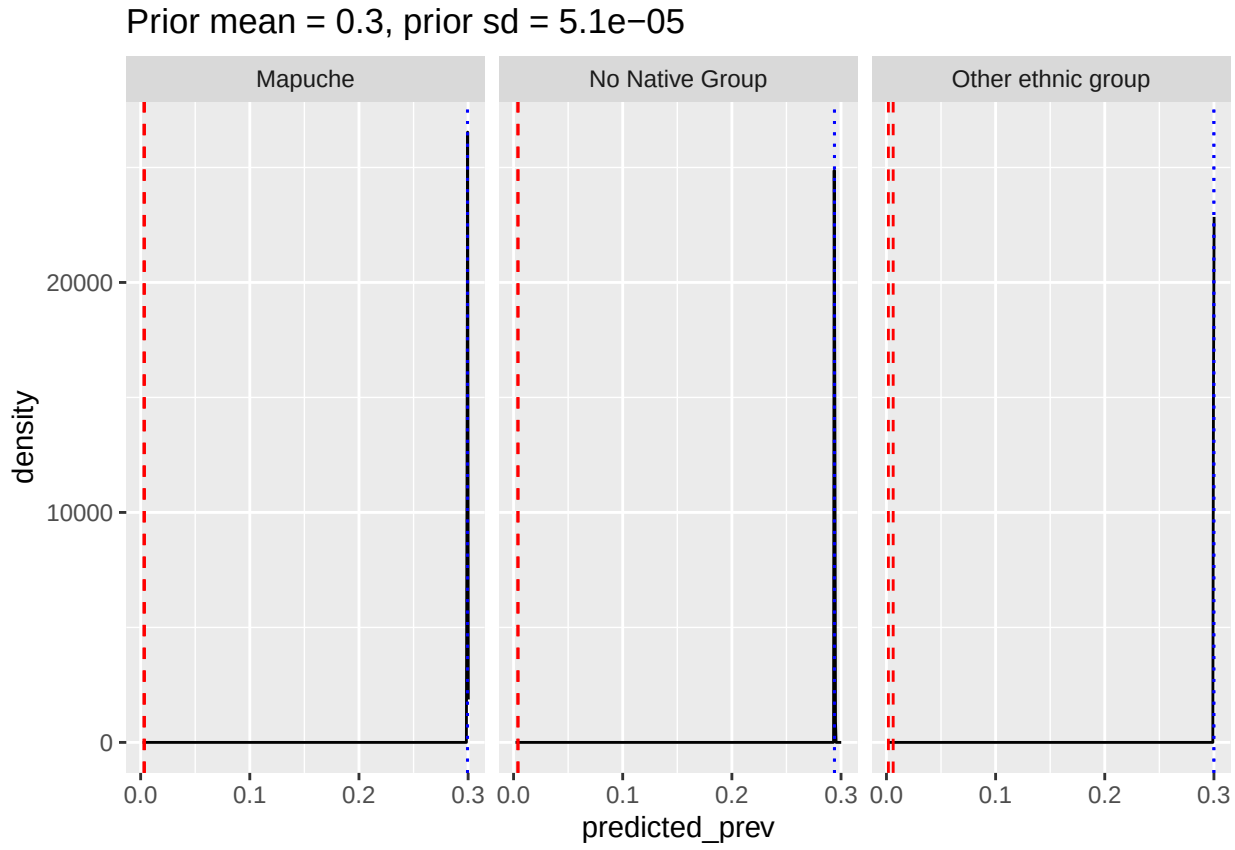

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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




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

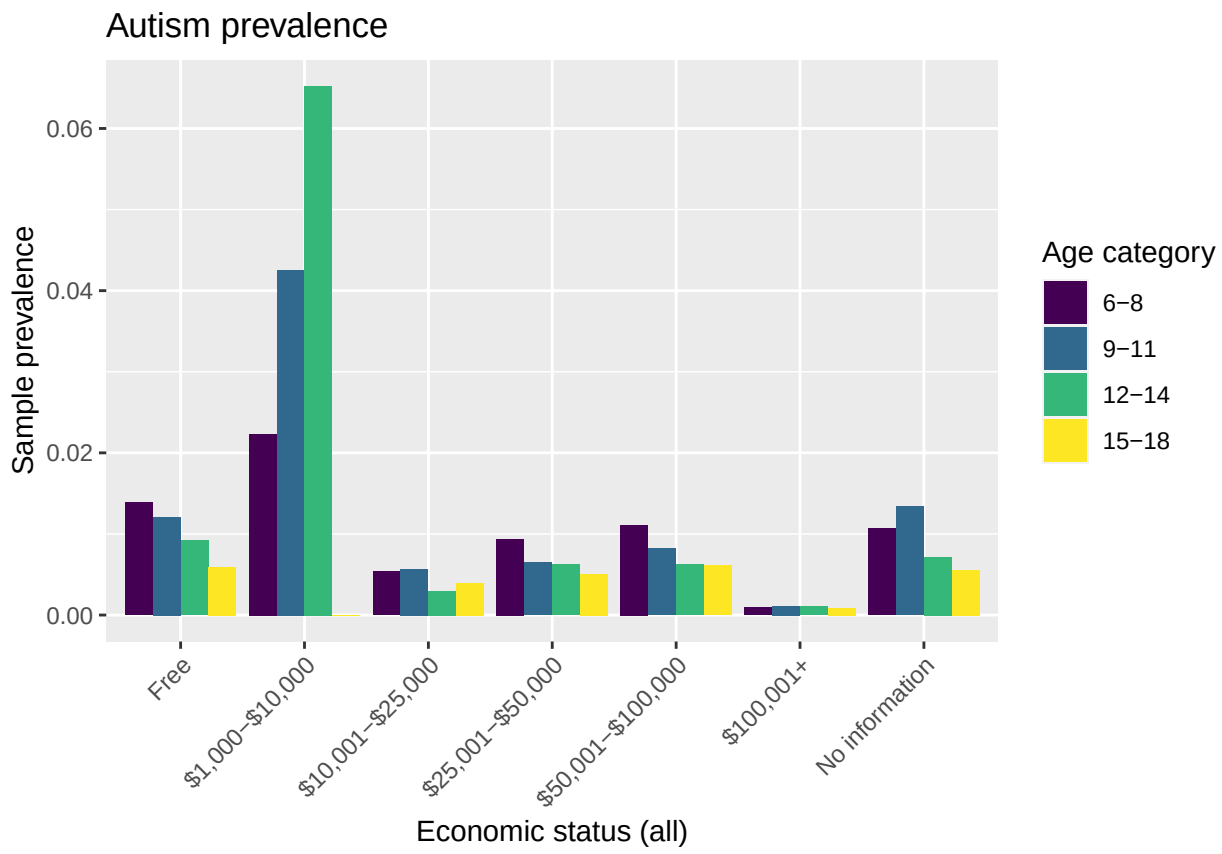


Random effect on economic status - all economic groups

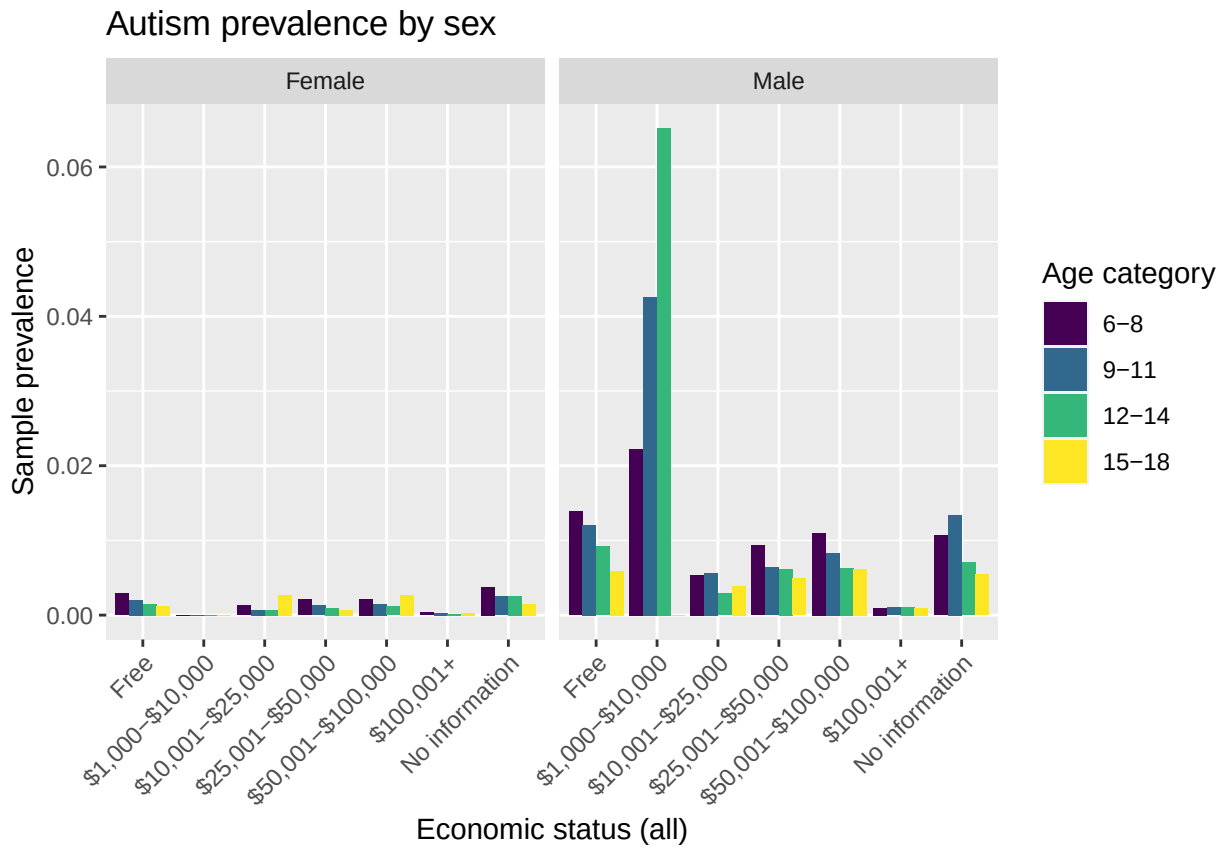
```
aut_prev_econ <- get_grouped_prev(x = chile_bayes_aut, stdpop = chile_stdpop,
                                  grouping_vars = c("school_fee", "school_fee_group", "age_june30", "a
```

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

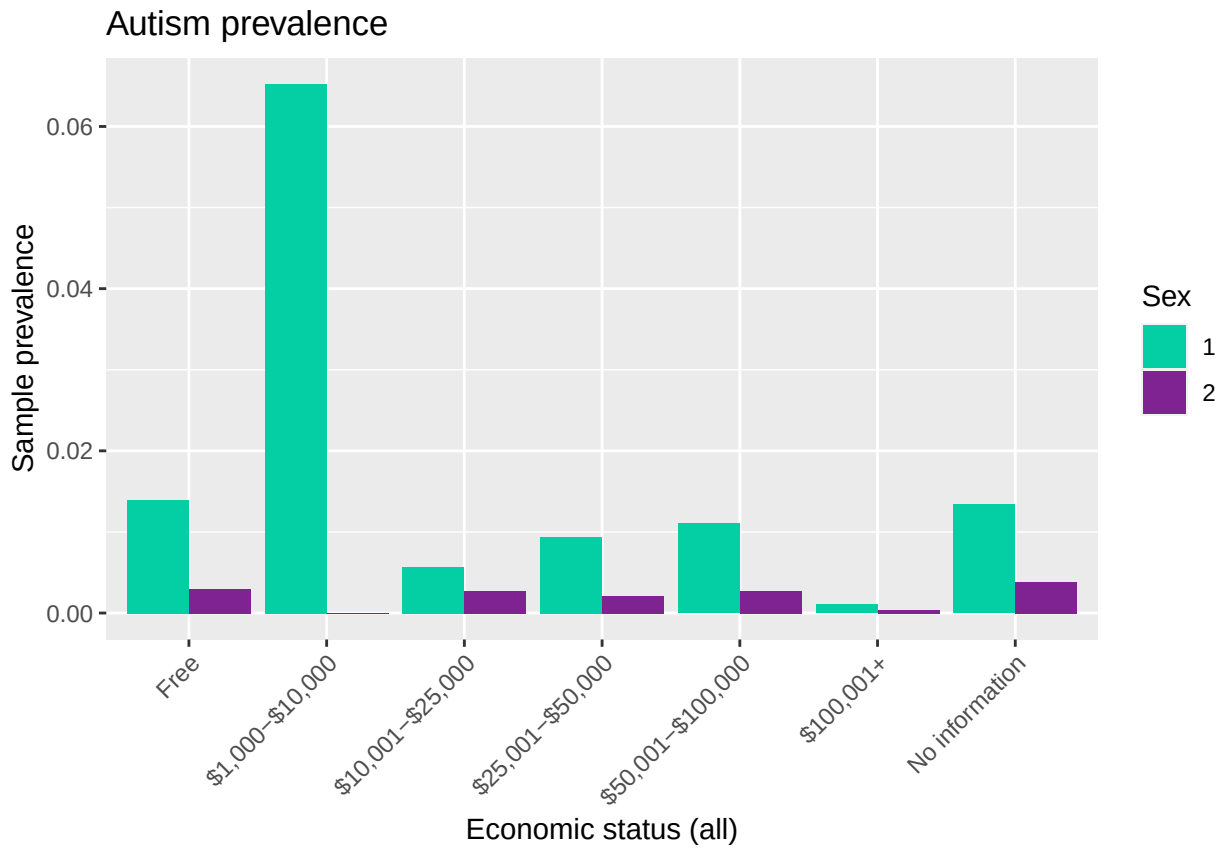
```
ggplot(data = aut_prev_econ) +
  geom_col(aes(x = as.factor(school_fee), y = sample_prevalence, group = age_cat_name, fill = age_cat_na
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Economic status (all)",
       y = "Sample prevalence",
       fill = "Age category")
```




```
ggplot(data = aut_prev_econ) +
  geom_col(aes(x = as.factor(school_fee), y = sample_prevalence, group = age_cat_name, fill = age_cat_na
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~sex_desc) +
  labs(title = "Autism prevalence by sex",
       x = "Economic status (all)",
       y = "Sample prevalence",
       fill = "Age category")
```

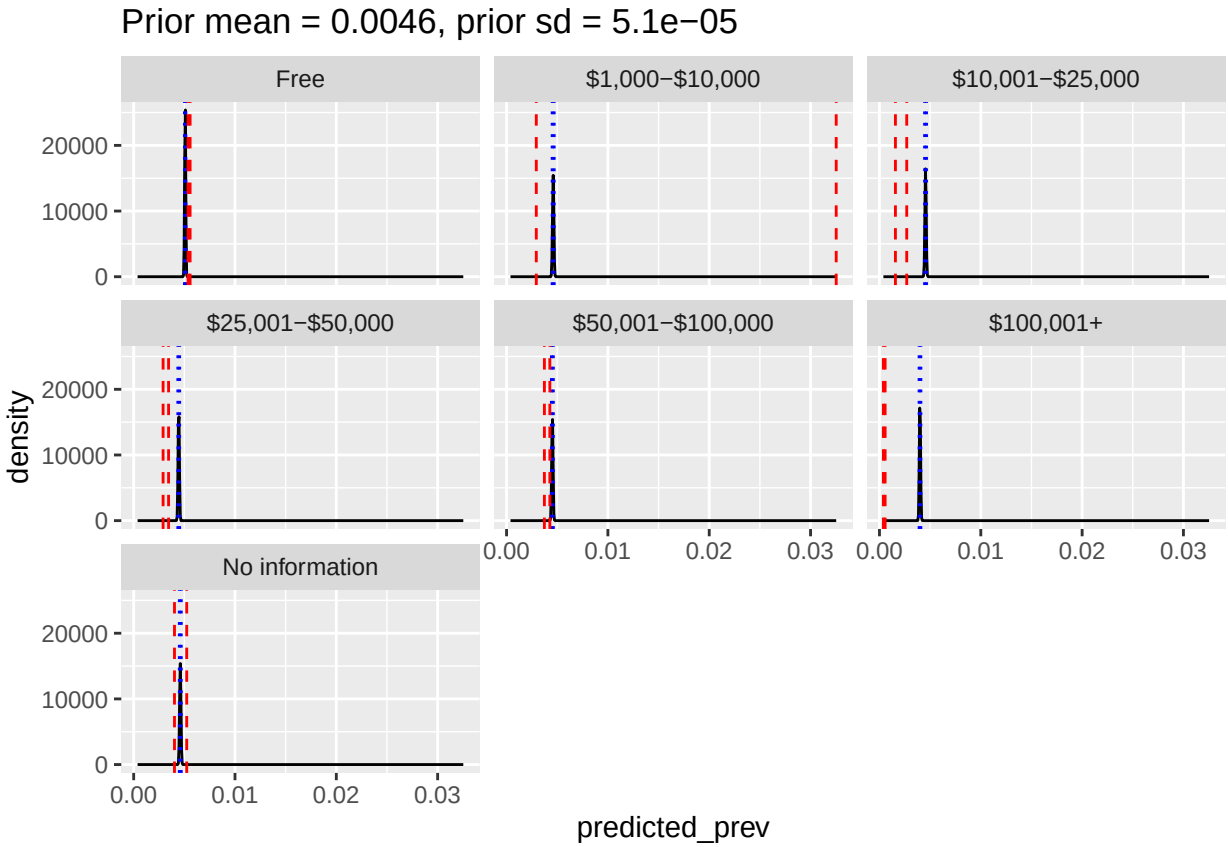



```
ggplot(data = aut_prev_econ) +
  geom_col(aes(x = as.factor(school_fee), y = sample_prevalence, group = sex, fill = as.factor(sex)), p
  scale_fill_manual(values = c("#03CEA4", "#802392")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Economic status (all)",
       y = "Sample prevalence",
       fill = "Sex")
```




```
#aut_prev_econ <- get_grouped_prev(x = chile_bayes_aut, stdpop = chile_stdpop,
#                                   grouping_vars = c("school_fee", "age_june30", "sex", "autism"))

aut_prev_econ_adj <- get_adjusted_prev(aut_prev_econ, grouping_vars = "school_fee")

aut_prev_econ_post <- do_jags_rand_model(x = aut_prev_econ_adj,
                              feat = "school_fee",
                              model = rand_model,
                              theta_mu = theta_mu_prior,
                              theta_sigma = theta_sigma_prior,
                              pars = pars,
                              convergence_checks = FALSE) %>%
  rename("school_fee" = "Feat_names")

plot_post_density(aut_prev_econ_post, aut_prev_econ_adj, feat = "school_fee", theta_mu = theta_mu_prior
```

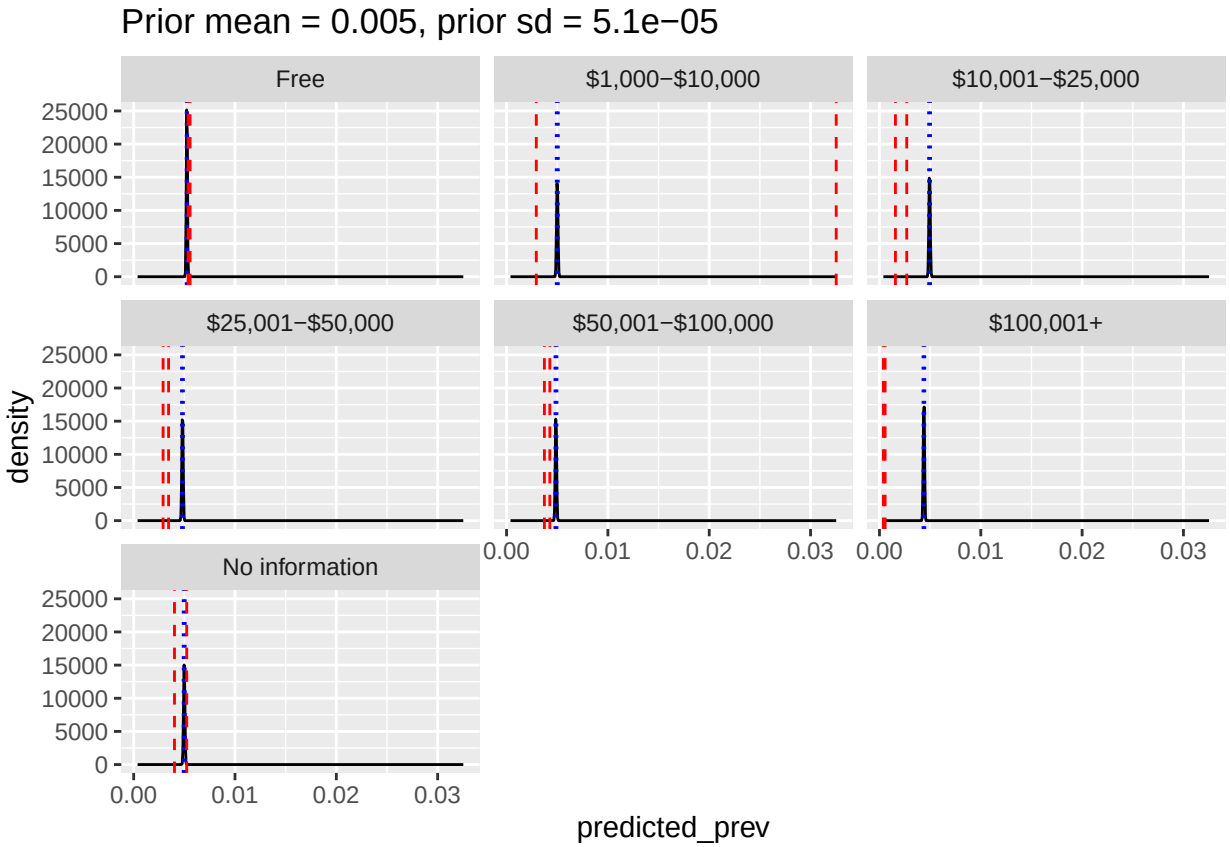
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



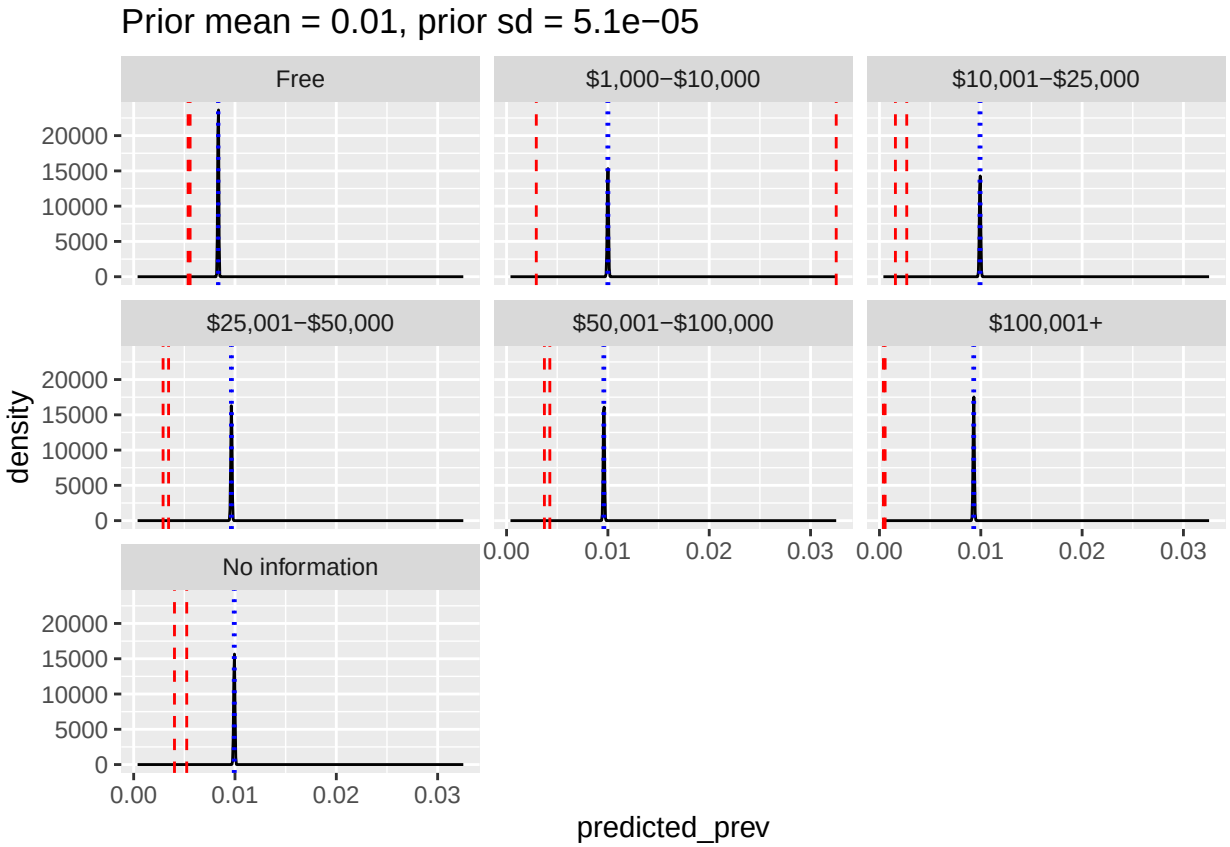
Predictions for higher population prevalence - increase prior mean

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

  plot_post_density(aut_prev_econ_post,
                    aut_prev_econ_adj,
                    feat = "school_fee",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}
```

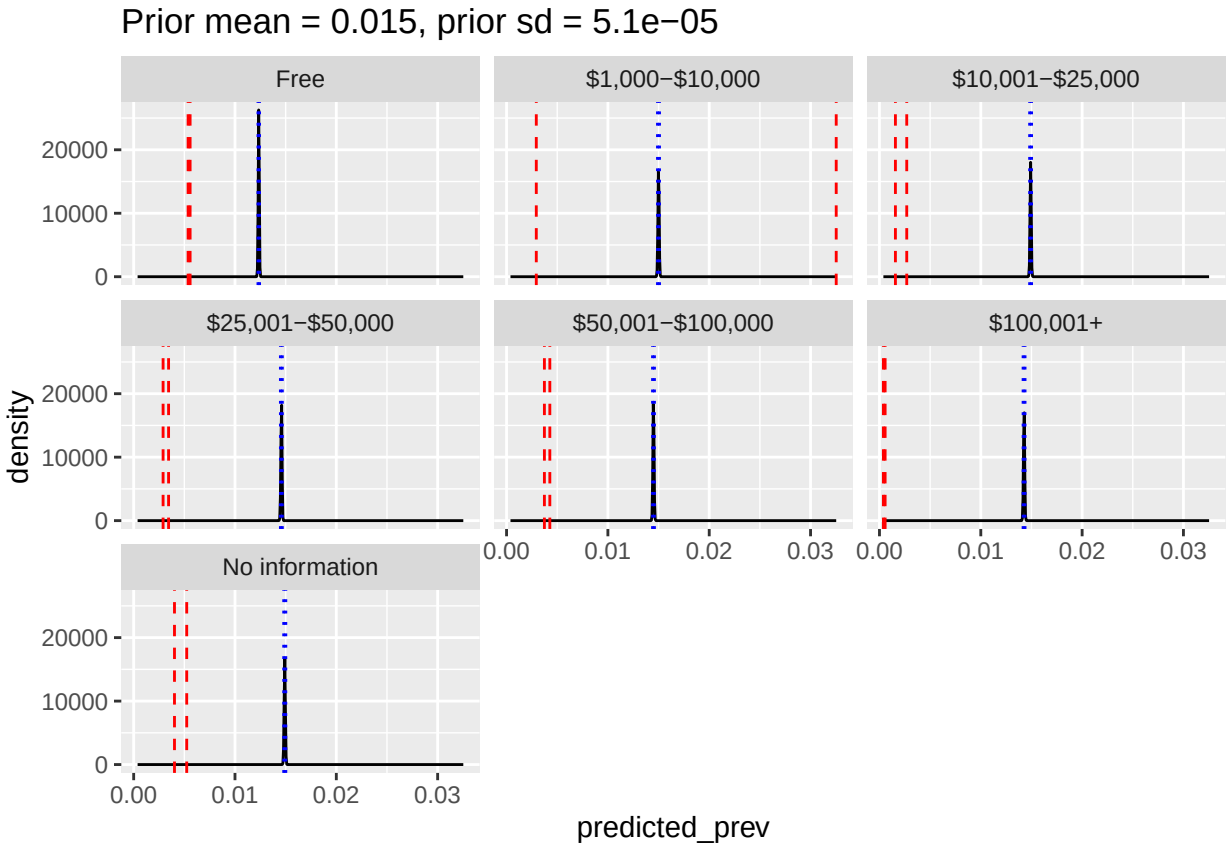
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



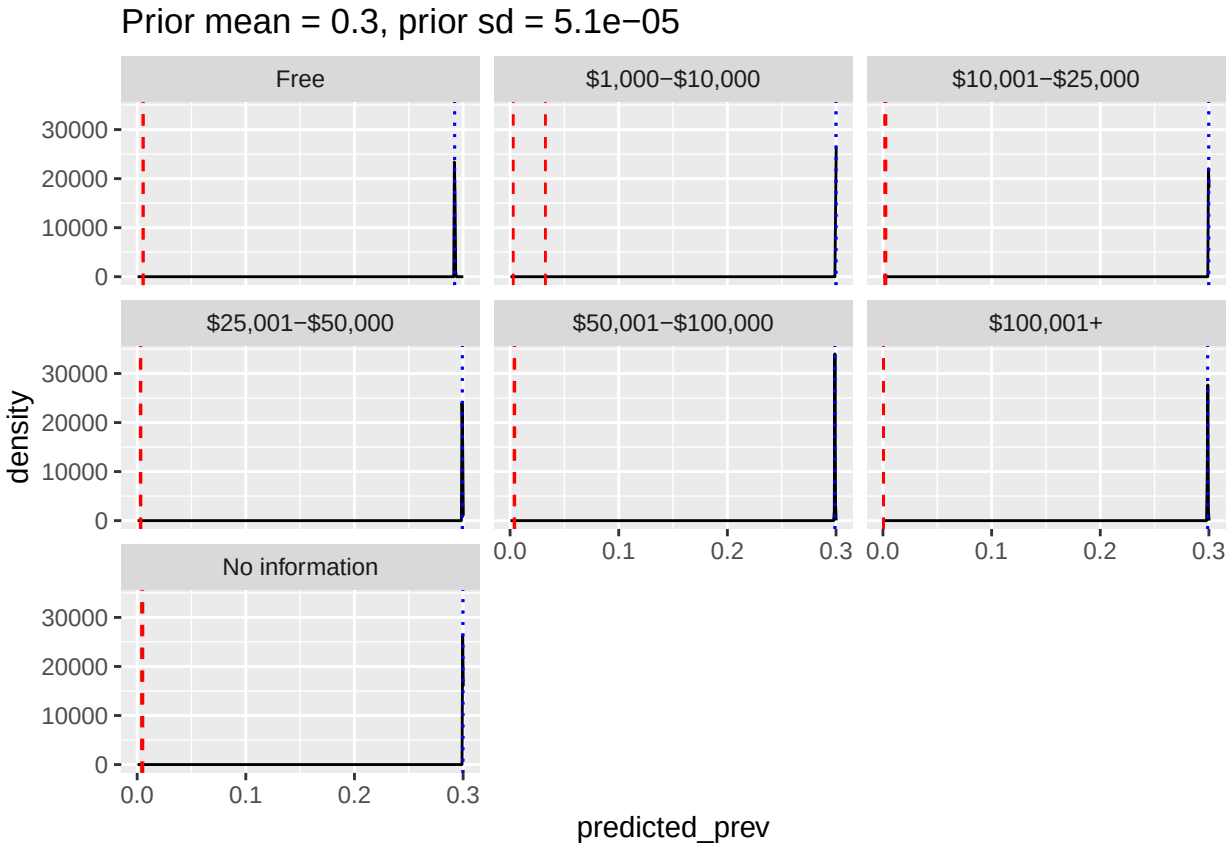

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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

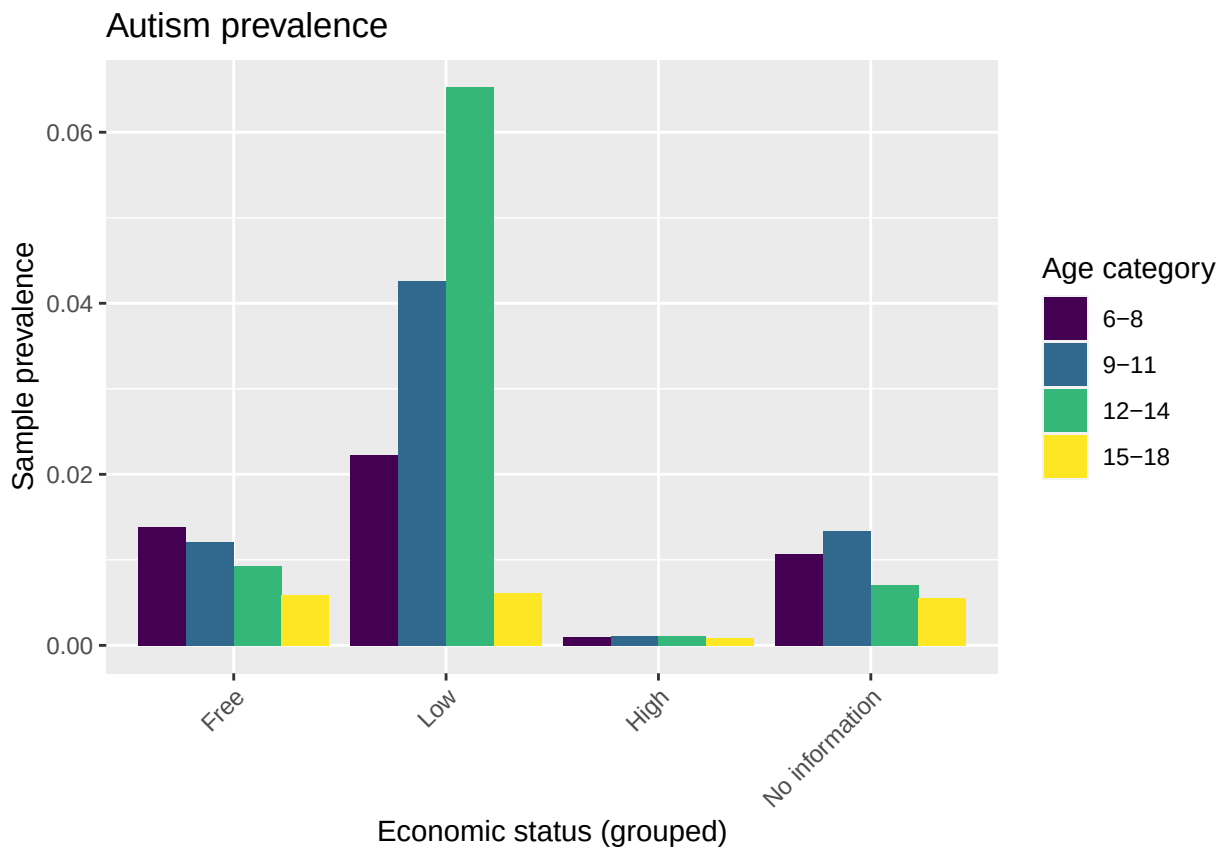



```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

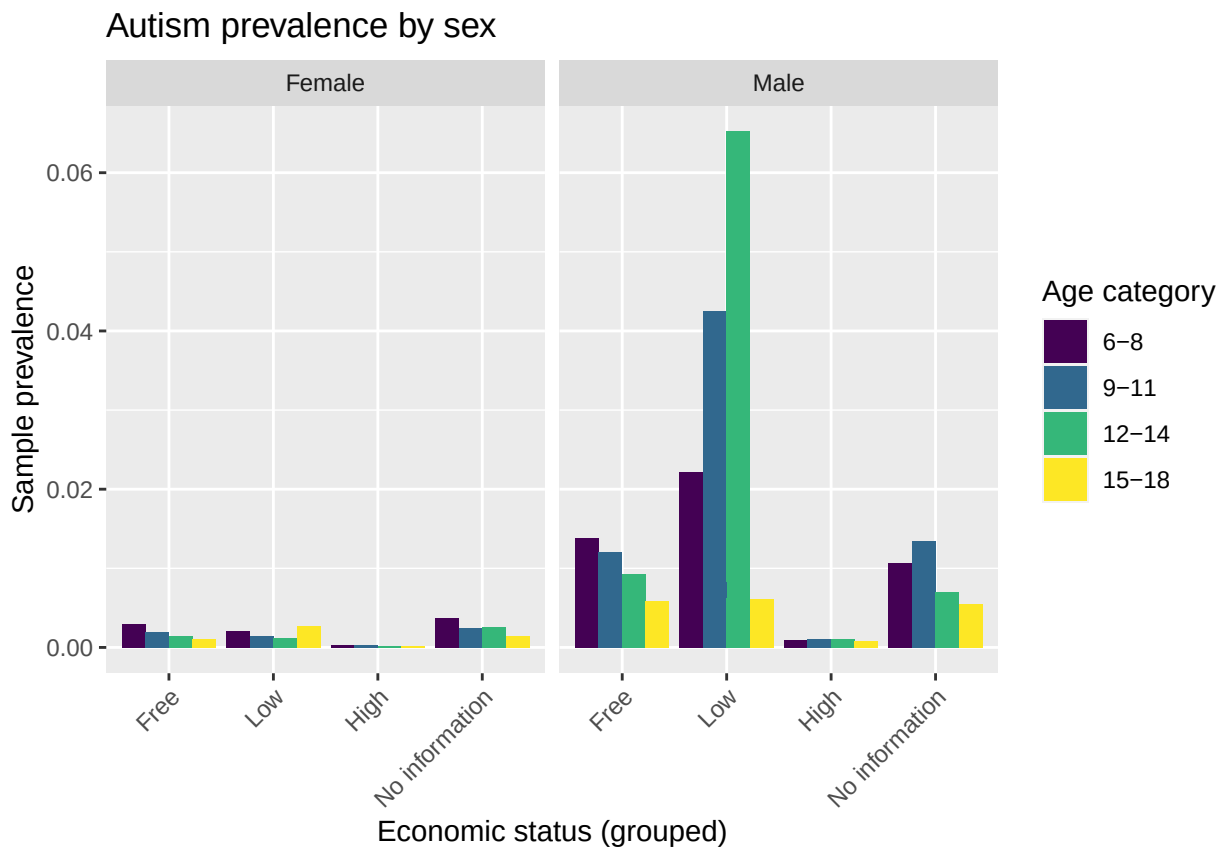


Random effect on economic status - free education, low status, high status and no information

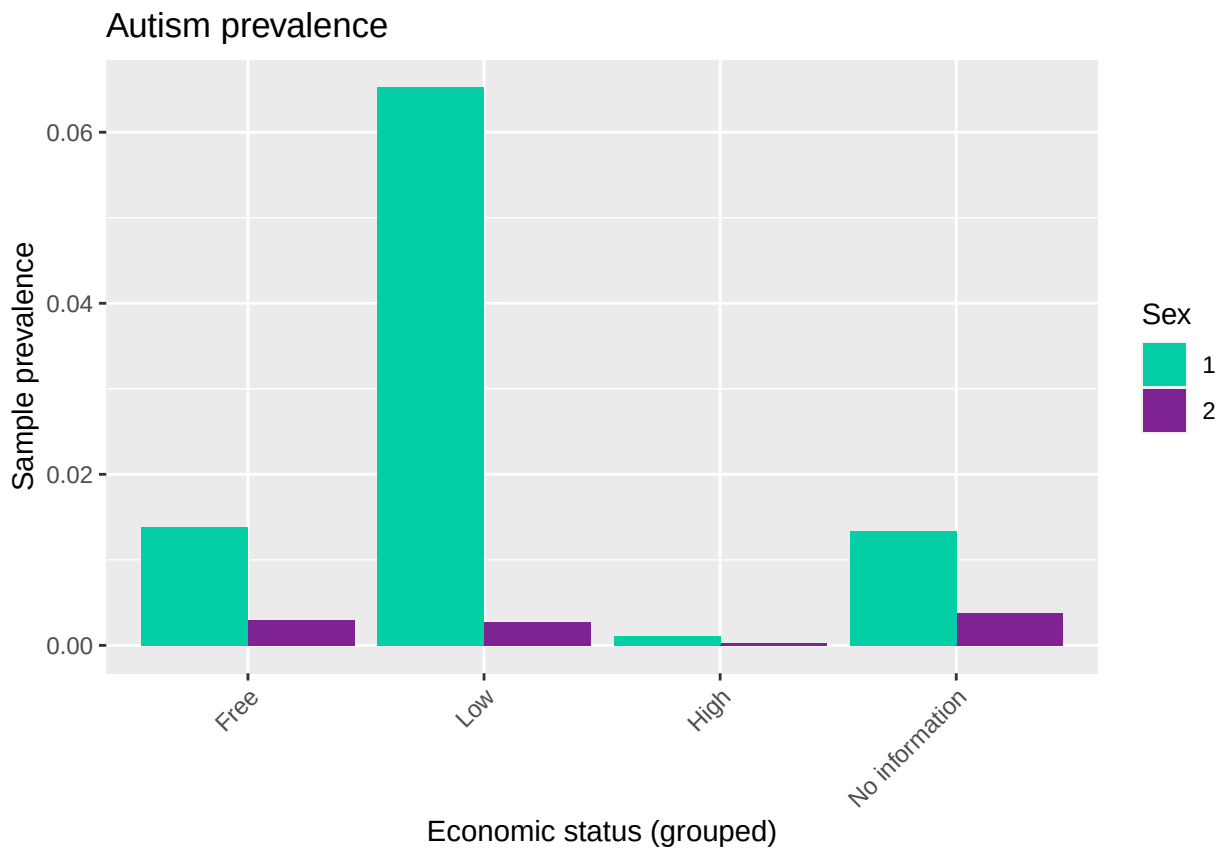
```
ggplot(data = aut_prev_econ) +
  geom_col(aes(x = as.factor(school_fee_group), y = sample_prevalence, group = age_cat_name, fill = age_
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Economic status (grouped)",
       y = "Sample prevalence",
       fill = "Age category")
```




```
ggplot(data = aut_prev_econ) +
  geom_col(aes(x = as.factor(school_fee_group), y = sample_prevalence, group = age_cat_name, fill = age_
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~sex_desc) +
  labs(title = "Autism prevalence by sex",
       x = "Economic status (grouped)",
       y = "Sample prevalence",
       fill = "Age category")
```

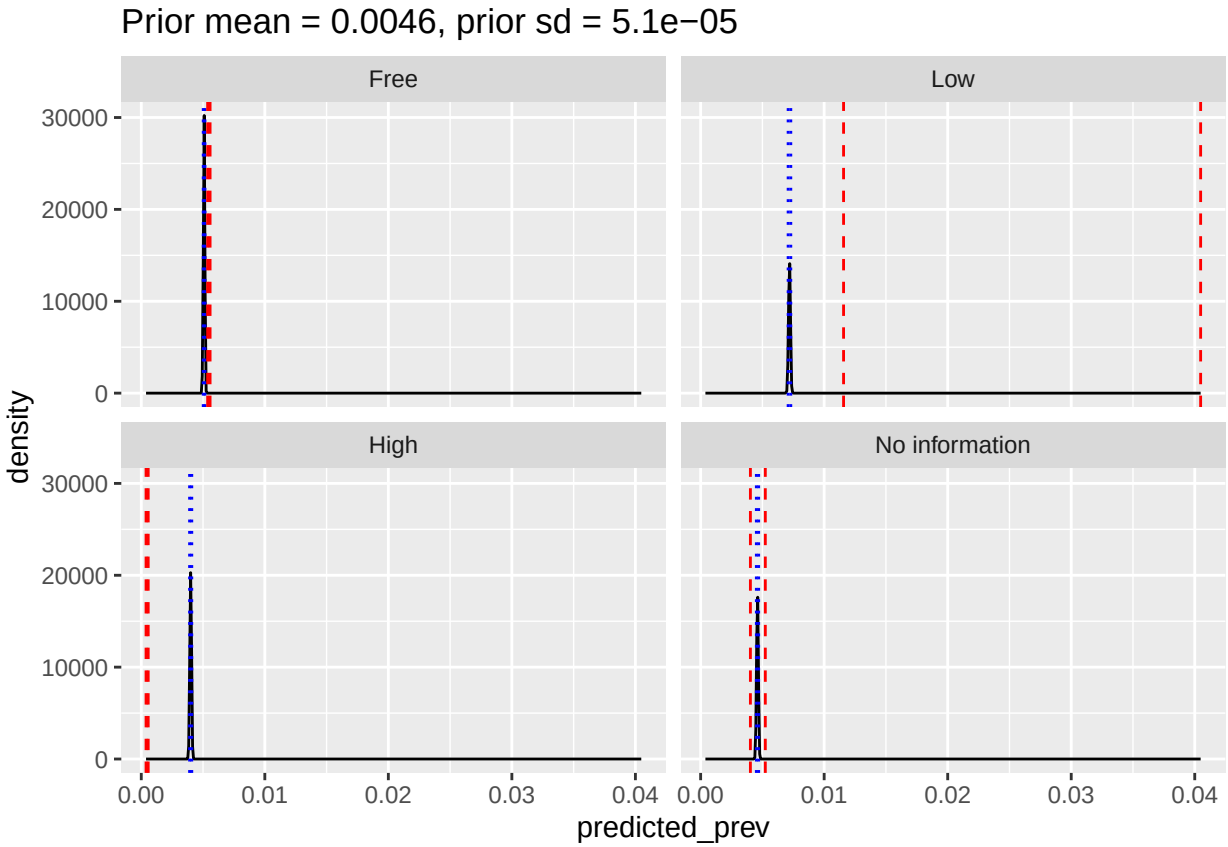



```
ggplot(data = aut_prev_econ) +
  geom_col(aes(x = as.factor(school_fee_group), y = sample_prevalence, group = sex, fill = as.factor(se
  scale_fill_manual(values = c("#03CEA4", "#802392")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Autism prevalence",
       x = "Economic status (grouped)",
       y = "Sample prevalence",
       fill = "Sex")
```




```
aut_prev_econG_adj <- get_adjusted_prev(aut_prev_econ, grouping_vars = "school_fee_group")

aut_prev_econG_post <- do_jags_rand_model(x = aut_prev_econG_adj,
                                feat = "school_fee_group",
                                model = rand_model,
                                theta_mu = theta_mu_prior,
                                theta_sigma = theta_sigma_prior,
                                pars = pars,
                                convergence_checks = FALSE) %>%
  rename("school_fee_group" = "Feat_names")

plot_post_density(aut_prev_econG_post, aut_prev_econG_adj, feat = "school_fee_group", theta_mu = theta_
```

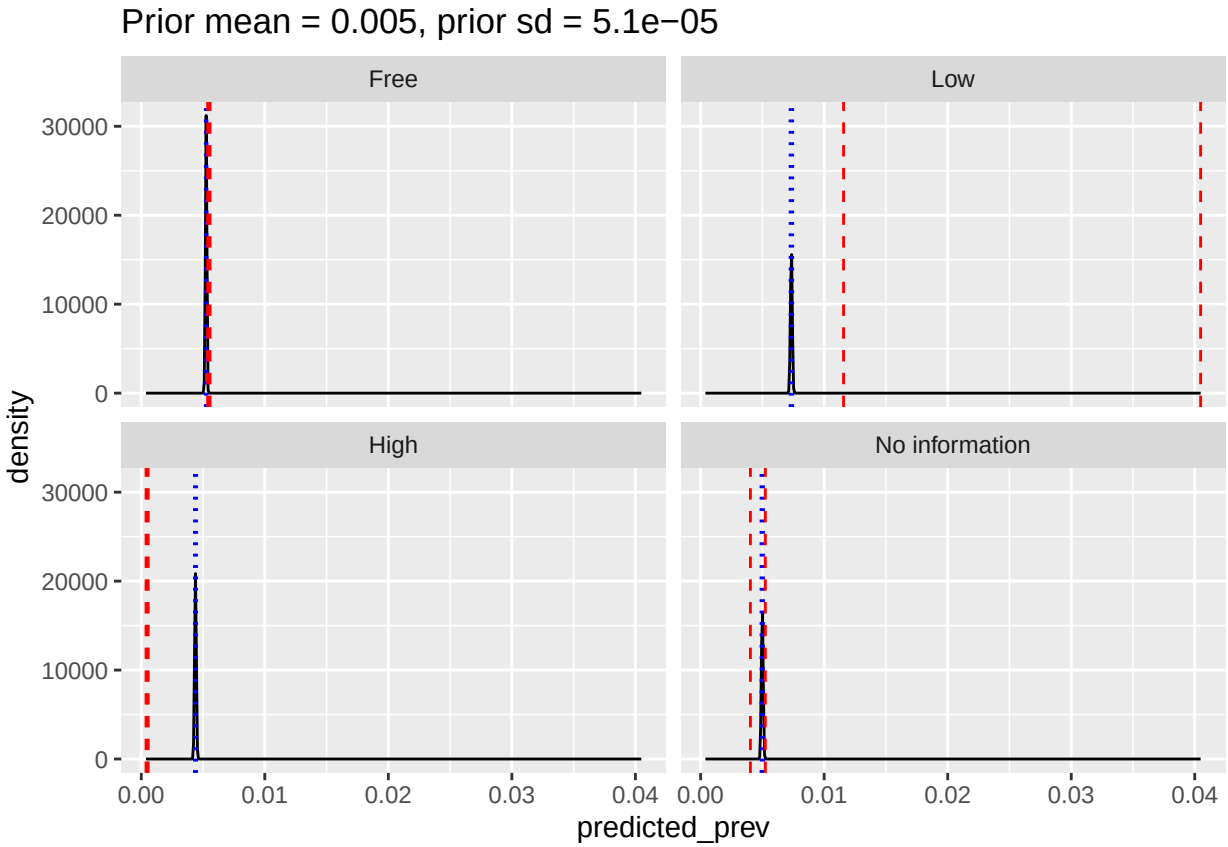
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



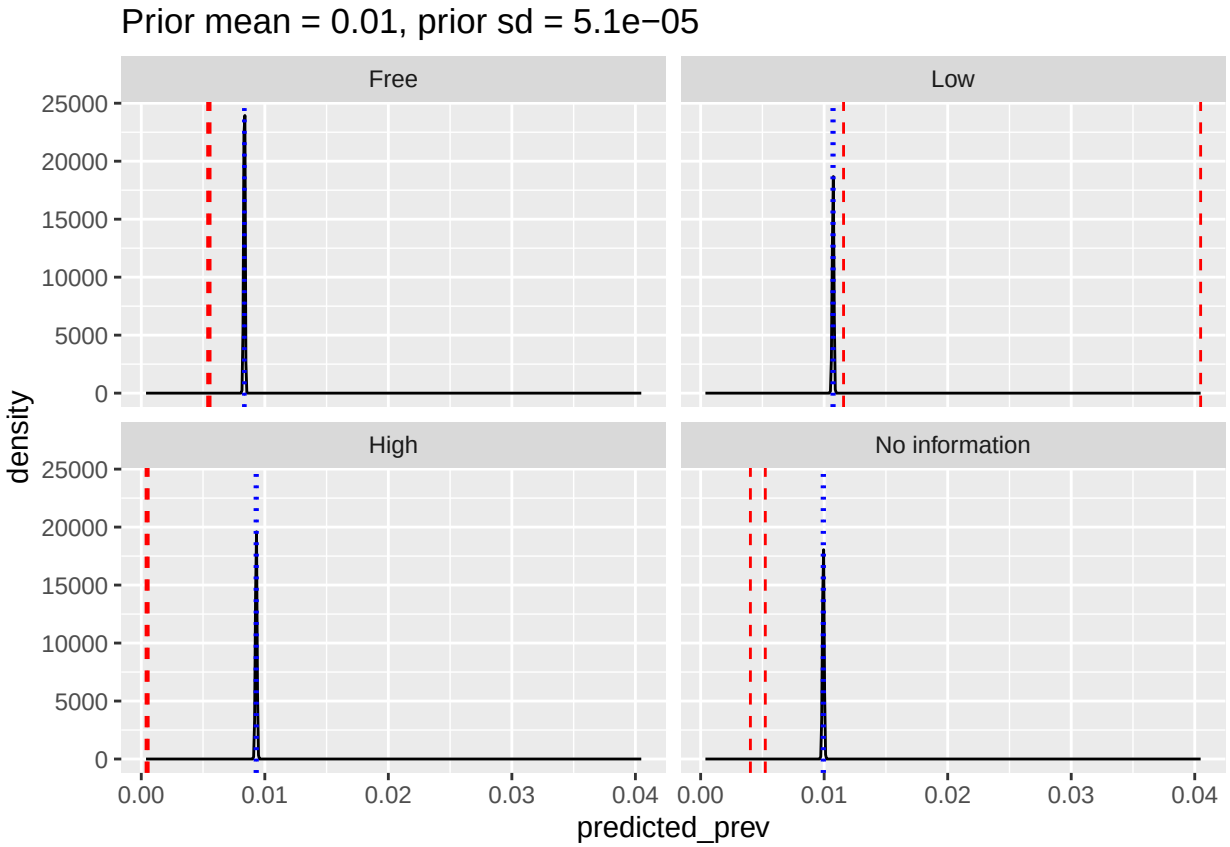
Predictions for higher population prevalence - increase prior mean

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

  plot_post_density(aut_prev_econG_post,
                    aut_prev_econG_adj,
                    feat = "school_fee_group",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}
```

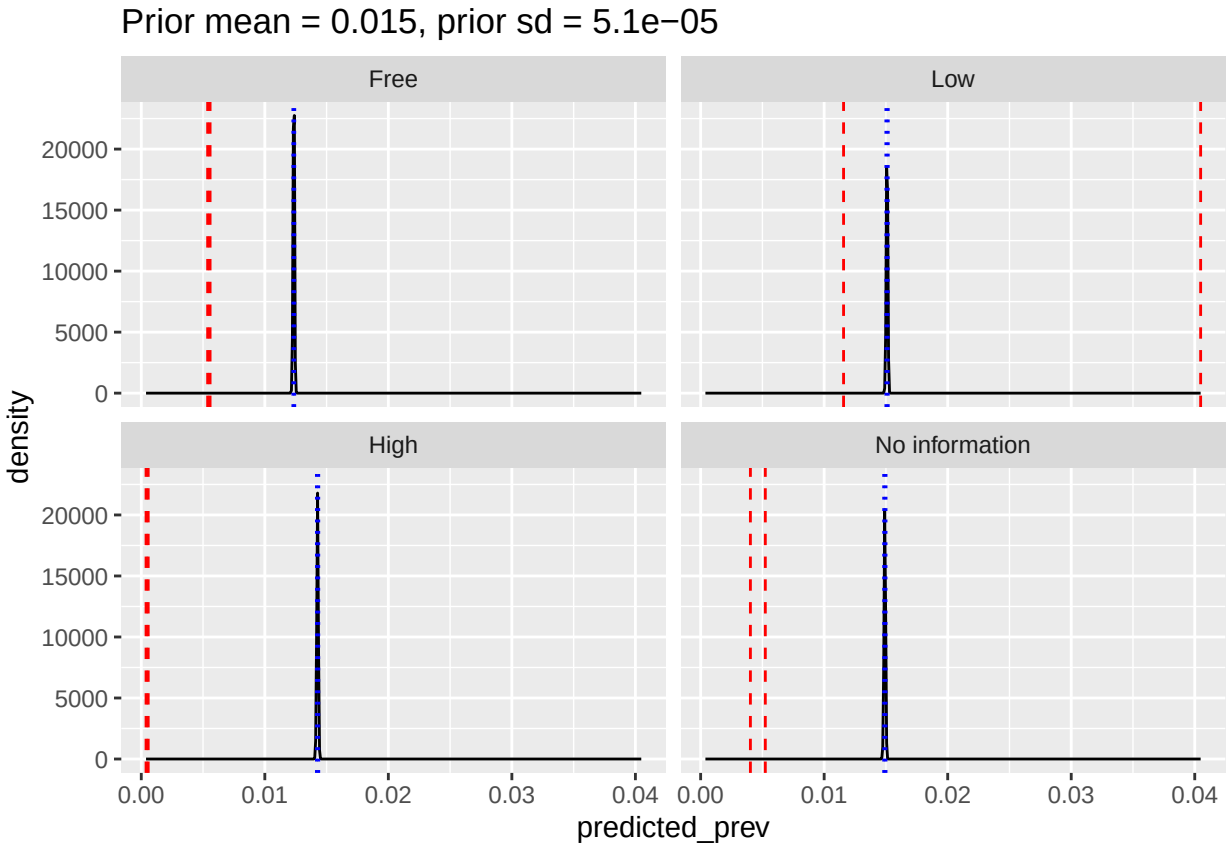
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



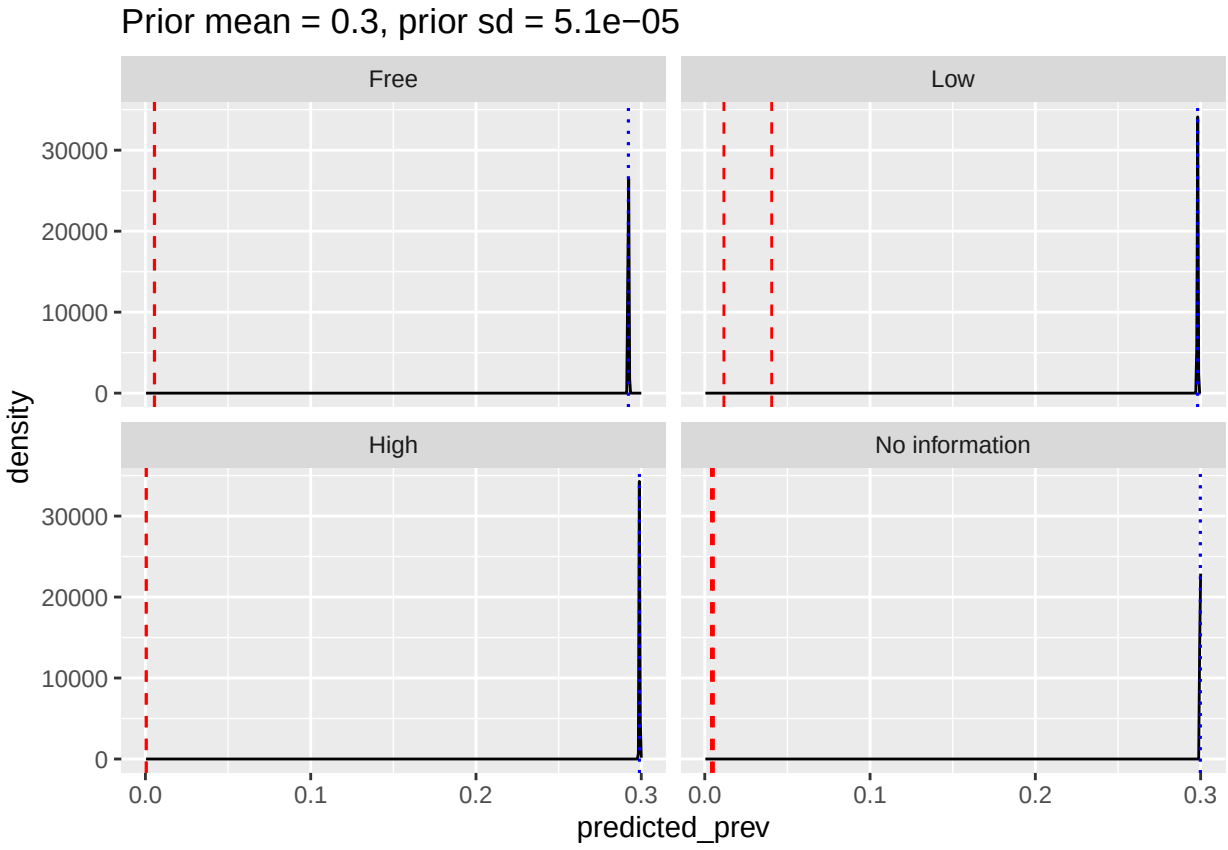

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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




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



Random effect on sex

```
chile_stdpop_f <- chile_stdpop %>%
  filter(sex == 2) %>%
  mutate(pop_prop = std_pop / sum(std_pop))
chile_stdpop_m <- chile_stdpop %>%
  filter(sex == 1) %>%
  mutate(pop_prop = std_pop / sum(std_pop))

aut_prev_f <- chile_bayes_aut %>%
  filter(sex == 2) %>%
  get_grouped_prev(stdpop = chile_stdpop_f, grouping_vars = c("age_june30", "sex", "autism"))
```

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

```
aut_prev_adj_f <- get_adjusted_prev(aut_prev_f, grouping_vars = c()) %>% mutate(sex = 2)

aut_prev_m <- chile_bayes_aut %>%
  filter(sex == 1) %>%
  get_grouped_prev(stdpop = chile_stdpop_m, grouping_vars = c("age_june30", "sex", "autism"))
```

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

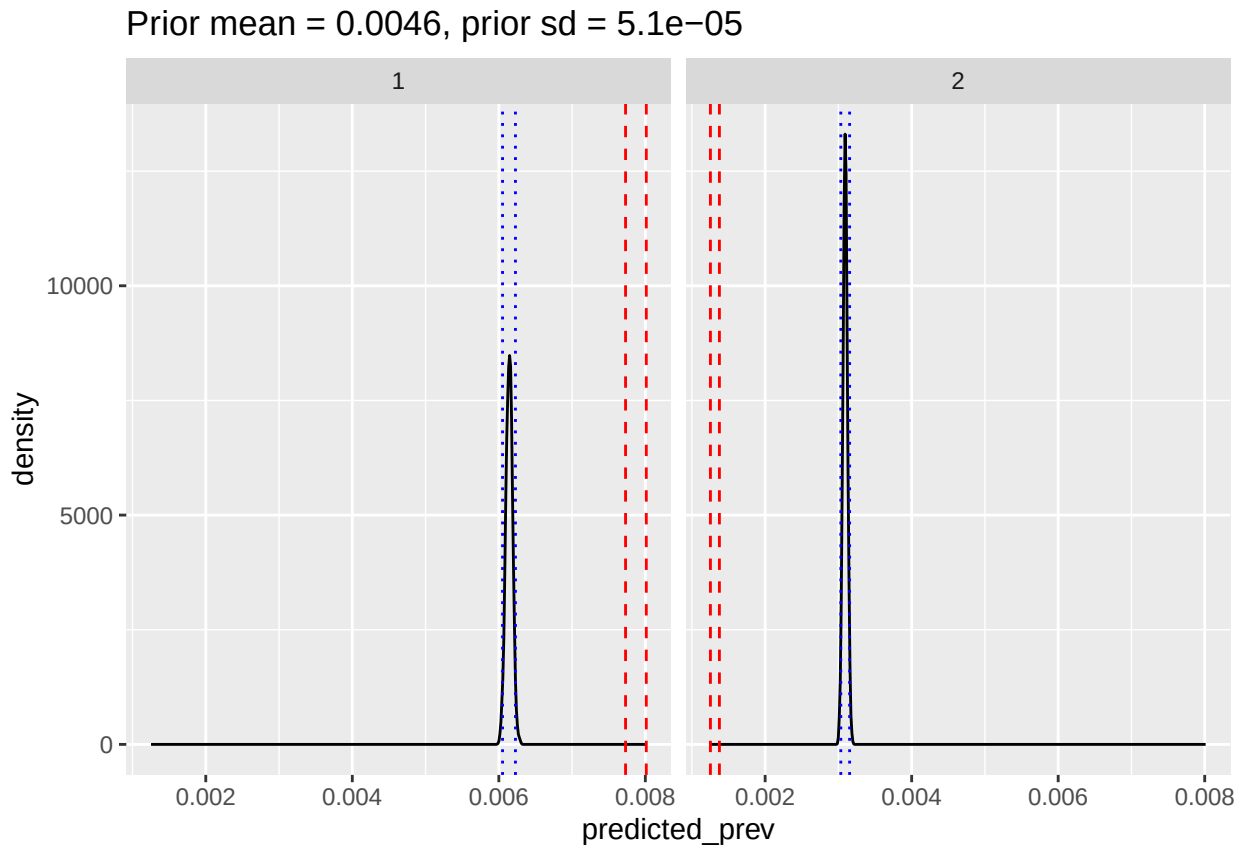
```
aut_prev_adj_m <- get_adjusted_prev(aut_prev_m, grouping_vars = c()) %>% mutate(sex = 1)
```

```r
aut_prev_sex_adj <- rbind(aut_prev_adj_m, aut_prev_adj_f)
# have to put m first because 1 comes before 2 and otherwise will mess up naming in do_jags_rand_model

aut_prev_sex_post <- do_jags_rand_model(x = aut_prev_sex_adj,
                                feat = "sex",
                                model = rand_model,
                                theta_mu = theta_mu_prior,
                                theta_sigma = theta_sigma_prior,
                                pars = pars,
                                convergence_checks = FALSE) %>%
  rename("sex" = "Feat_names")

plot_post_density(aut_prev_sex_post, aut_prev_sex_adj, feat = "sex", theta_mu = theta_mu_prior, theta_s
```

```
## Warning: Combining variables of class <factor> and <numeric> 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 <numeric> 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.
```



Random effect on region, sexes separate

Need sex specific priors - use age-adjusted prevalence from above

```
theta_mu_prior_f <- aut_prev_adj_f$adjusted_rate
theta_sigma_prior_f <- sqrt(aut_prev_adj_f$var)

theta_mu_prior_m <- aut_prev_adj_m$adjusted_rate
theta_sigma_prior_m <- sqrt(aut_prev_adj_m$var)
```

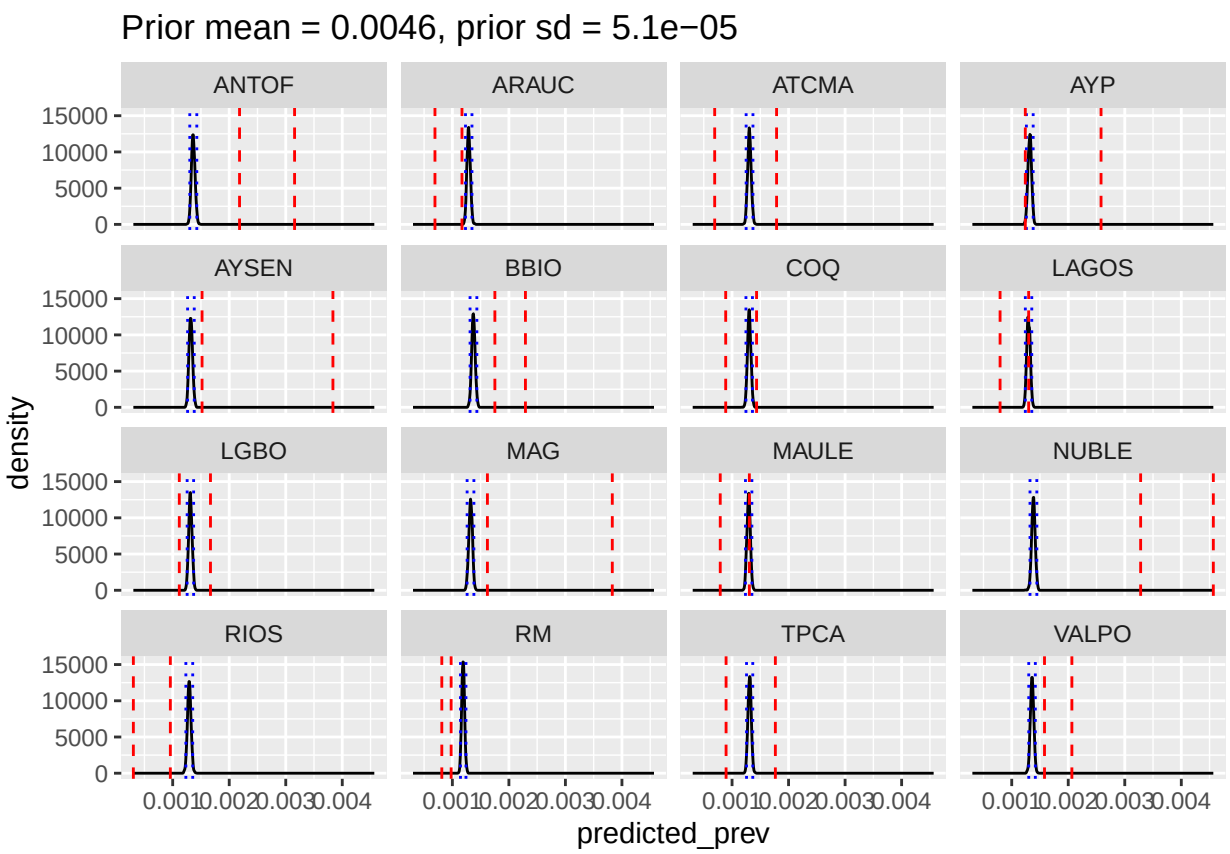
```
# Females
aut_prev_region_f <- chile_bayes_aut %>%
  filter(sex == 2) %>%
  get_grouped_prev(stdpop = chile_stdpop_f, grouping_vars = c("school_region_name_abr", "age_june30", "
```

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

```
aut_prev_region_adj_f <- get_adjusted_prev(aut_prev_region_f, grouping_vars = "school_region_name_abr")

aut_prev_region_post_f <- do_jags_rand_model(x = aut_prev_region_adj_f,
                                feat = "school_region_name_abr",
                                model = rand_model,
                                theta_mu = theta_mu_prior_f,
                                theta_sigma = theta_sigma_prior_f,
                                pars = pars,
                                convergence_checks = FALSE) %>%
```

```
  rename("school_region_name_abr" = "Feat_names")

plot_post_density(aut_prev_region_post_f, aut_prev_region_adj_f, feat = "school_region_name_abr", theta
```

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

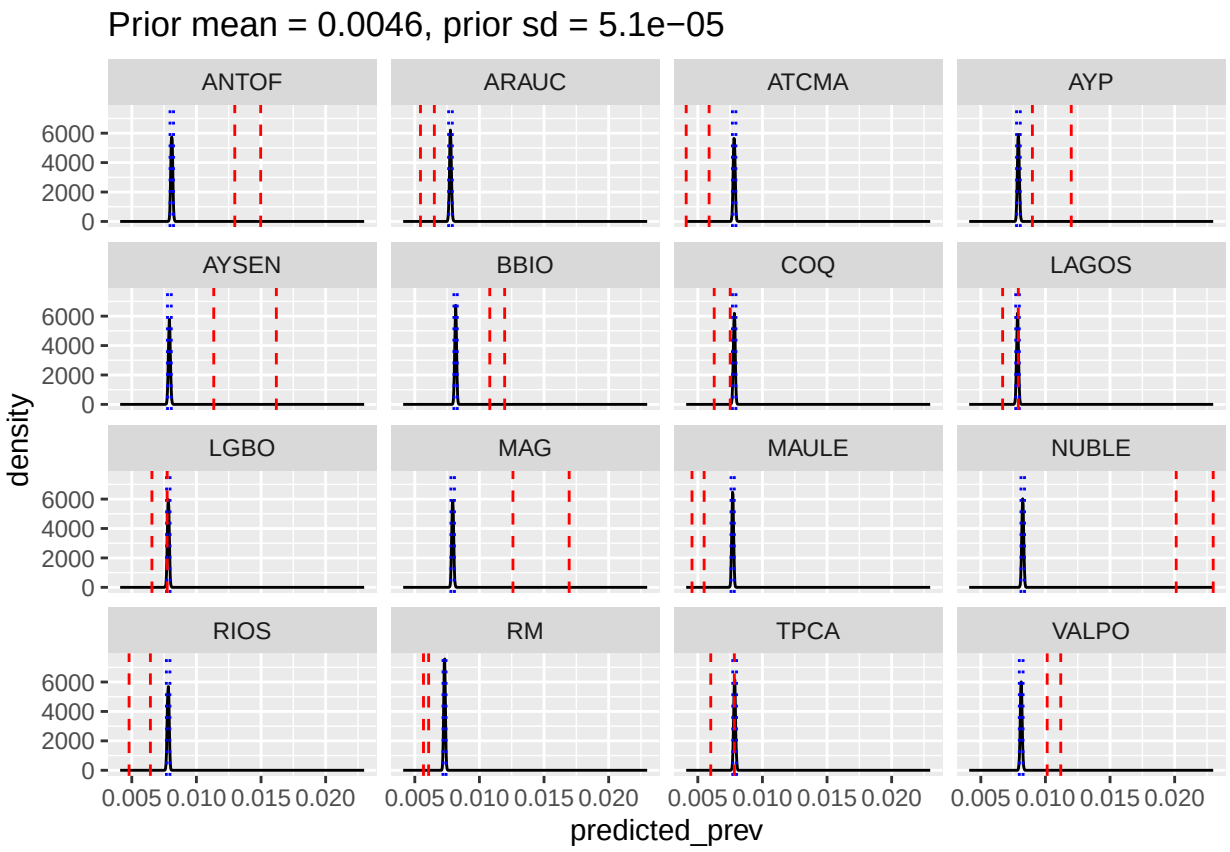



```
# Males
aut_prev_region_m <- chile_bayes_aut %>%
  filter(sex == 1) %>%
  get_grouped_prev(stdpop = chile_stdpop_m, grouping_vars = c("school_region_name_abr", "age_june30", "
```

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

```
aut_prev_region_adj_m <- get_adjusted_prev(aut_prev_region_m, grouping_vars = "school_region_name_abr")

aut_prev_region_post_m <- do_jags_rand_model(x = aut_prev_region_adj_m,
                                feat = "school_region_name_abr",
                                model = rand_model,
                                theta_mu = theta_mu_prior_m,
                                theta_sigma = theta_sigma_prior_m,
                                pars = pars,
                                convergence_checks = FALSE) %>%
  rename("school_region_name_abr" = "Feat_names")

plot_post_density(aut_prev_region_post_m, aut_prev_region_adj_m, feat = "school_region_name_abr", theta
```

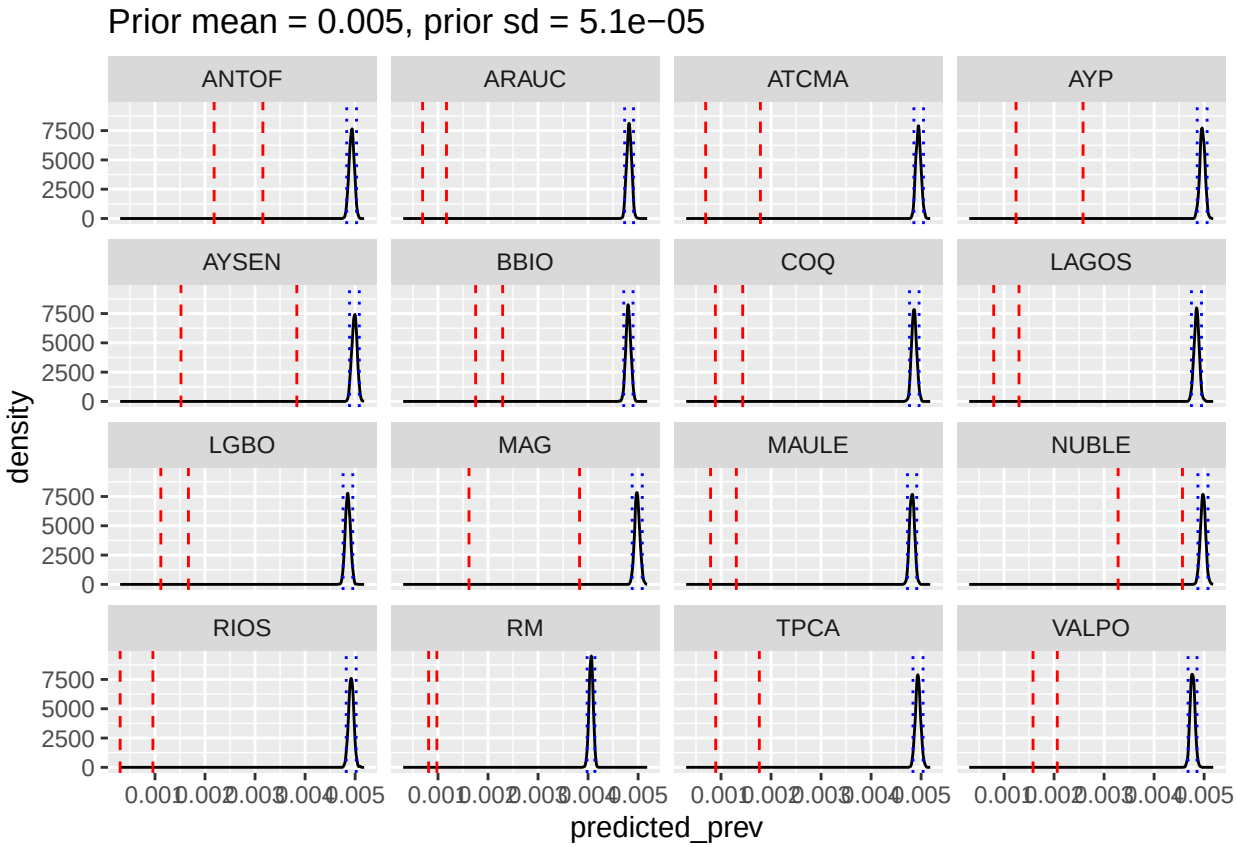
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



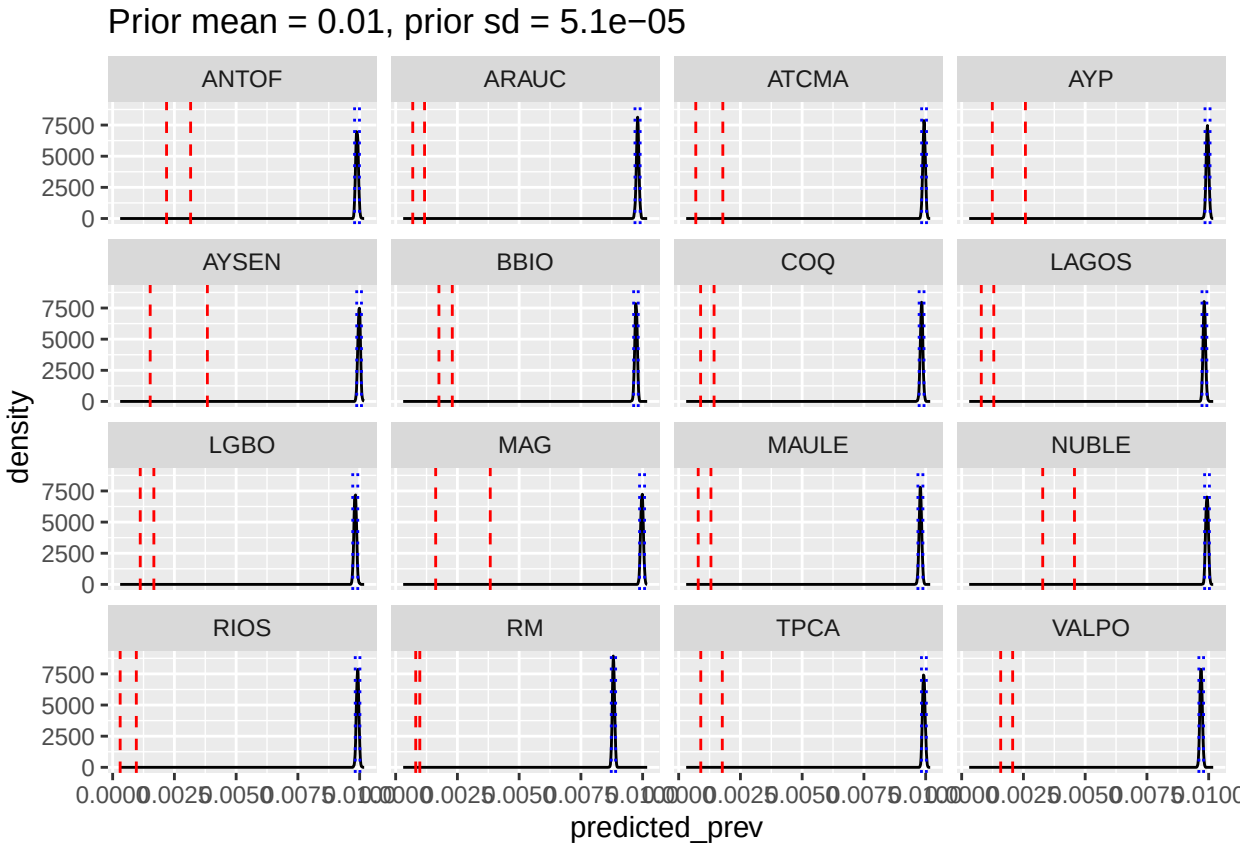
Maybe need different extrapolation priors for females?

```
# Females
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_region_post_f <- do_jags_rand_model(x = aut_prev_region_adj_f,
                                               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_f,
                    aut_prev_region_adj_f,
                    feat = "school_region_name_abr",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])

}
```

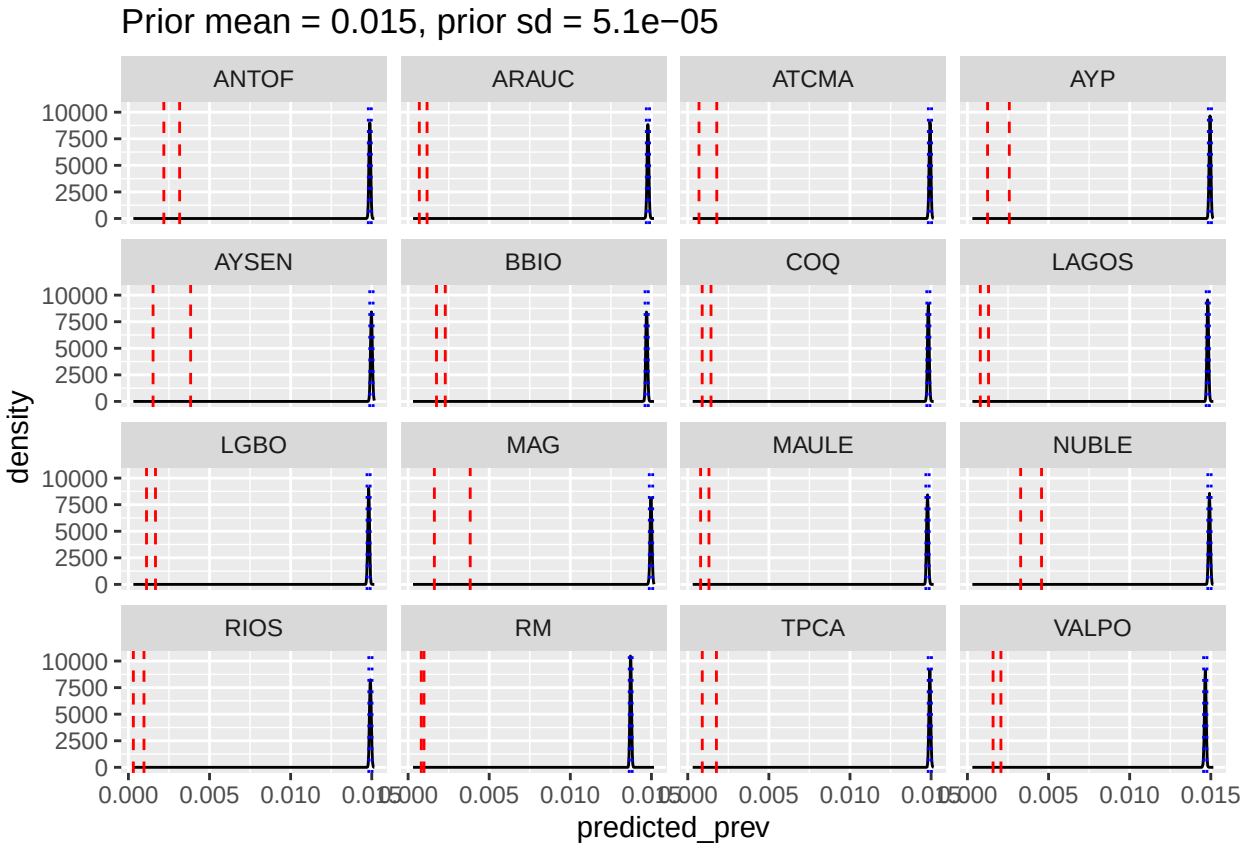
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



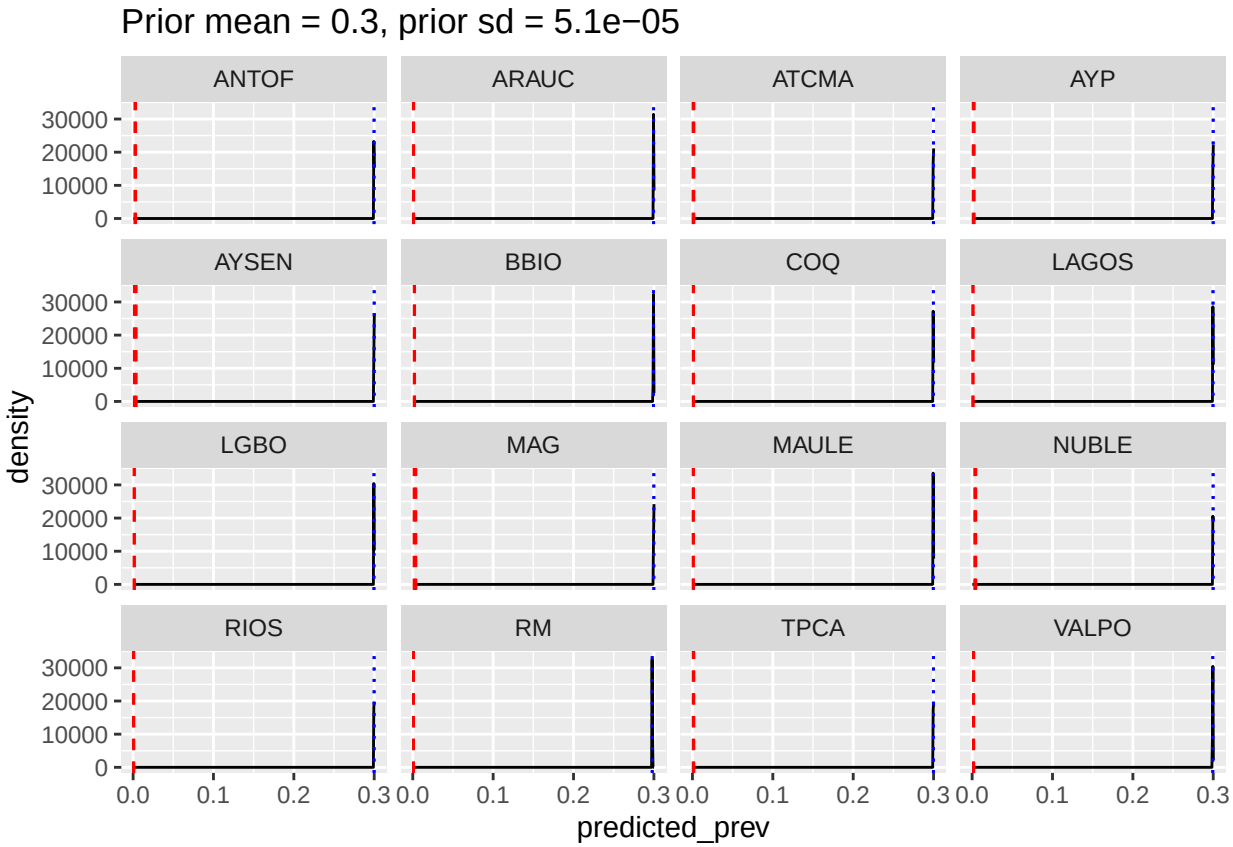

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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

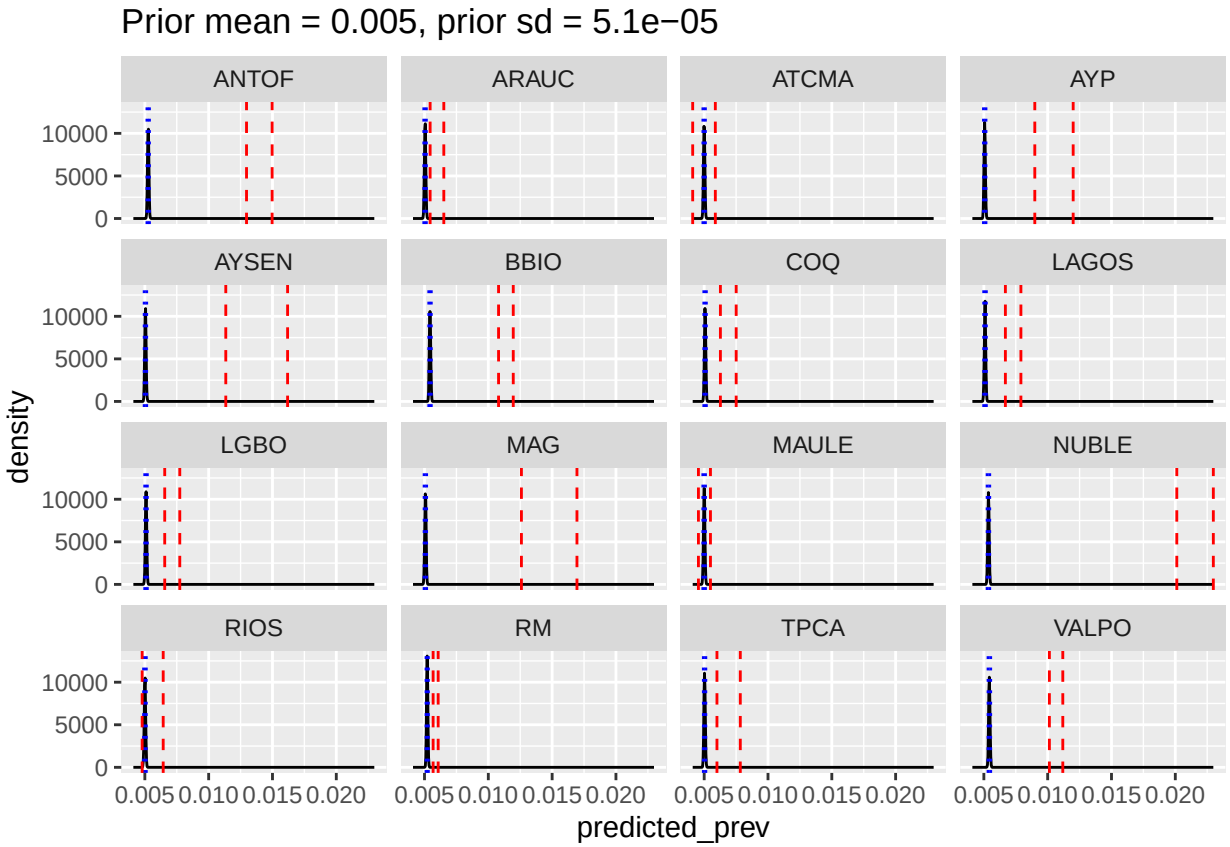



```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

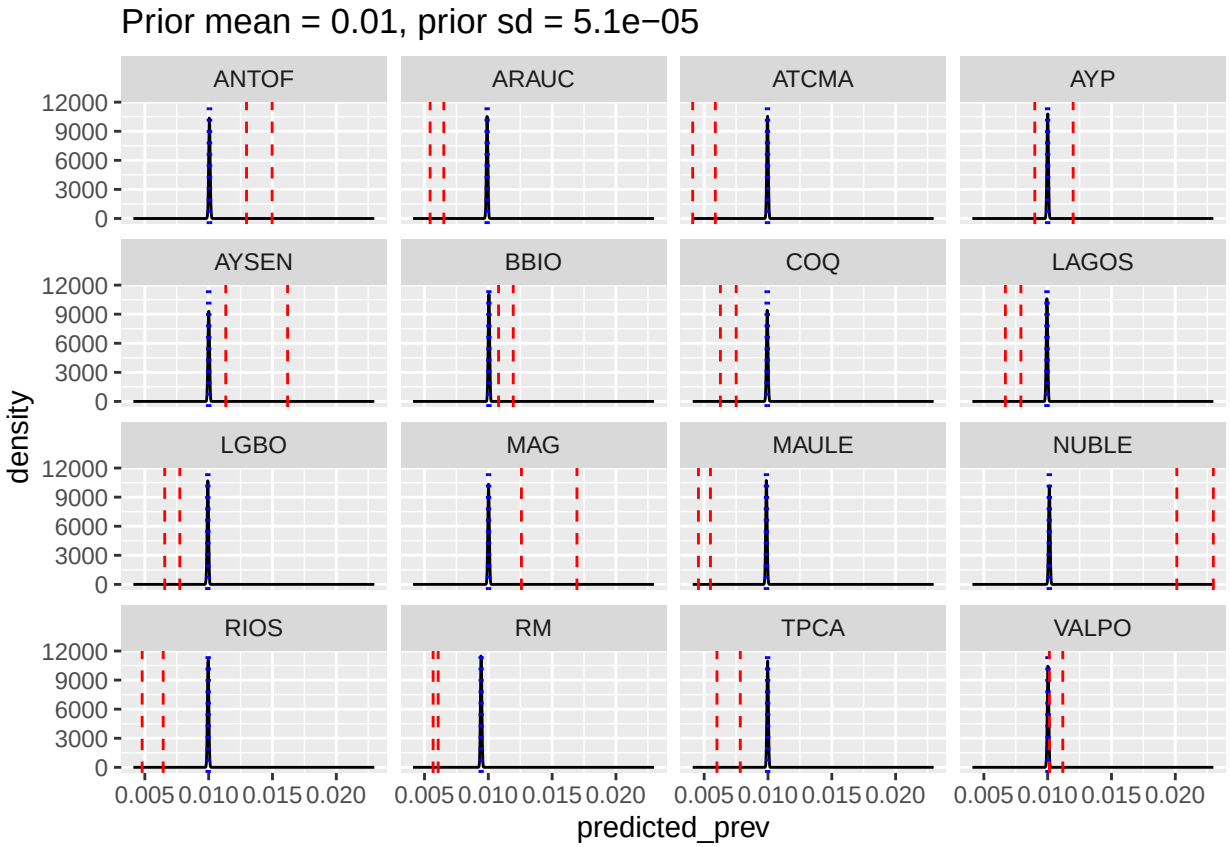



```
# Males
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_region_post_m <- do_jags_rand_model(x = aut_prev_region_adj_m,
                                               feat = "school_region_name_abr",
                                               model = rand_model,
                                               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_m,
                    aut_prev_region_adj_m,
                    feat = "school_region_name_abr",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])

}
```

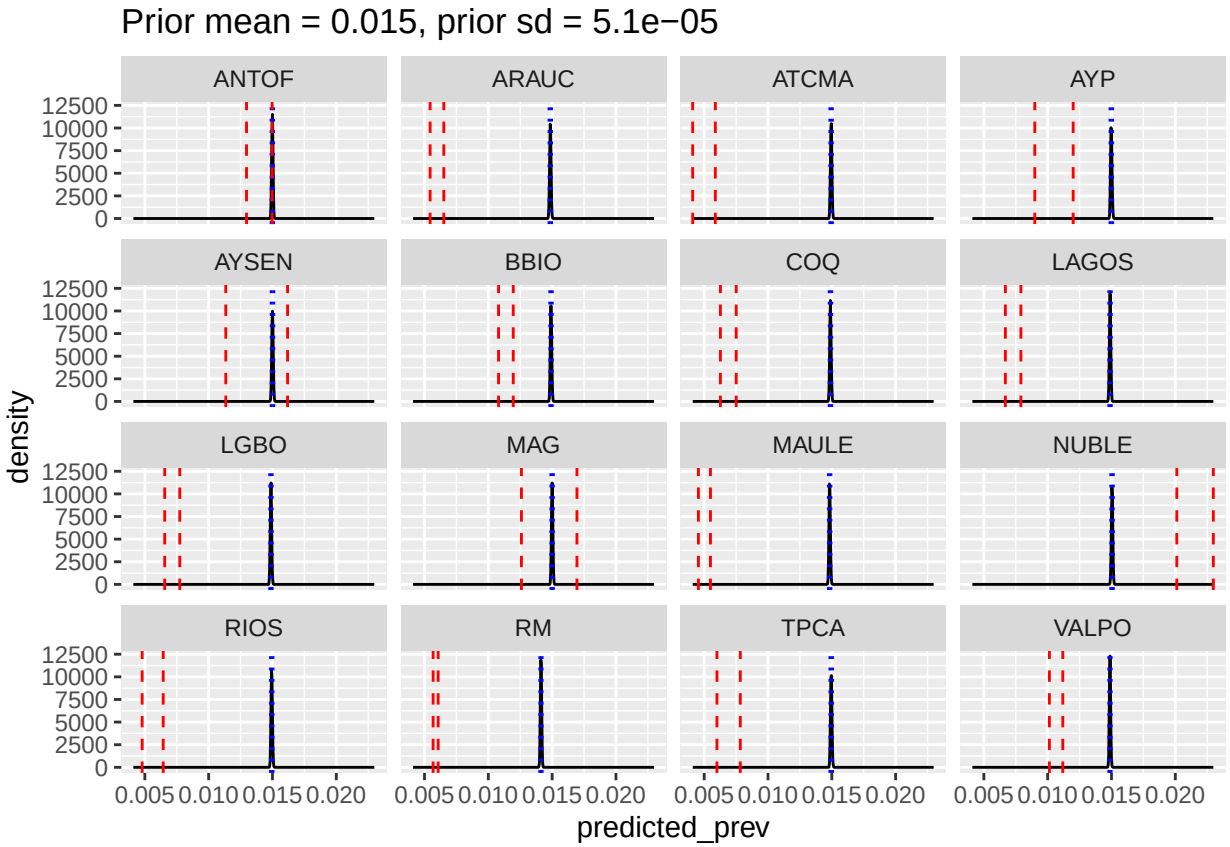
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



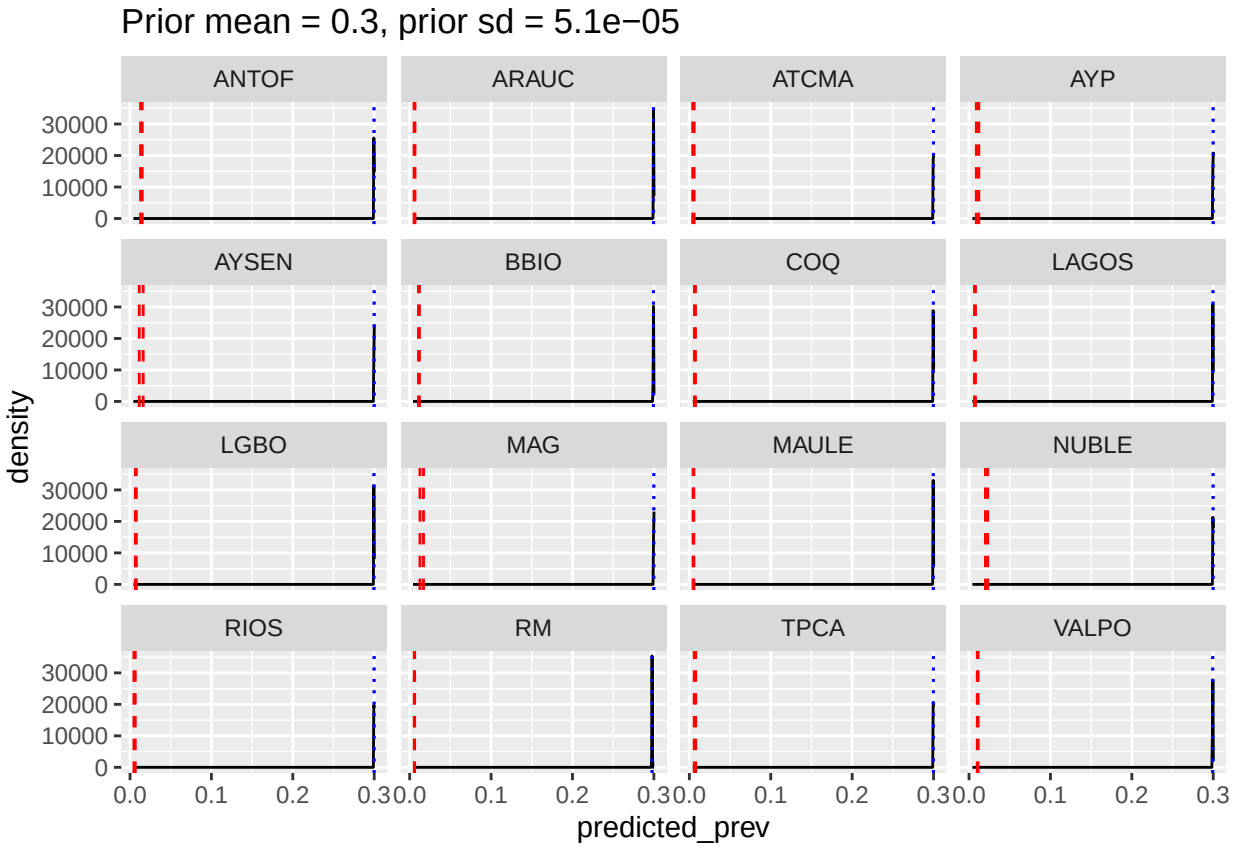

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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




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



Random effect on ethnicity, sexes separate

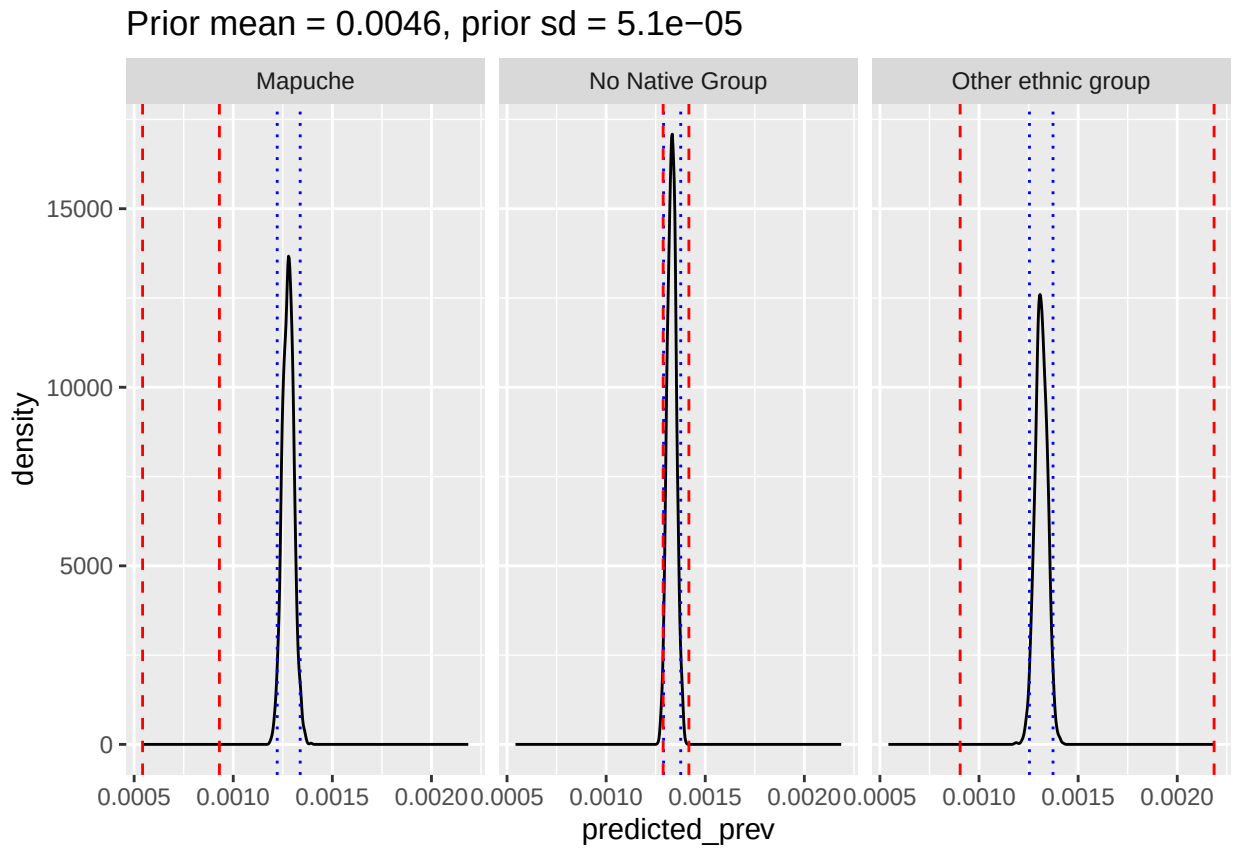
Need sex-specific priors

```
# Females
aut_prev_ethnic_f <- chile_bayes_aut %>%
  filter(sex == 2) %>%
  get_grouped_prev(stdpop = chile_stdpop_f, grouping_vars = c("ethnic_2_group", "age_june30", "sex", "au
```

```
## `summarise()` has grouped output by 'ethnic_2_group', 'age_june30', 'sex'. You
## can override using the `.groups` argument.
```

```
aut_prev_ethnic_adj_f <- get_adjusted_prev(aut_prev_ethnic_f, grouping_vars = "ethnic_2_group")

aut_prev_ethnic_post_f <- do_jags_rand_model(x = aut_prev_ethnic_adj_f,
                                feat = "ethnic_2_group",
                                model = rand_model,
                                theta_mu = theta_mu_prior_f,
                                theta_sigma = theta_sigma_prior_f,
                                pars = pars,
                                convergence_checks = FALSE) %>%
  rename("ethnic_2_group" = "Feat_names")

plot_post_density(aut_prev_ethnic_post_f, aut_prev_ethnic_adj_f, feat = "ethnic_2_group", theta_mu = the
```

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

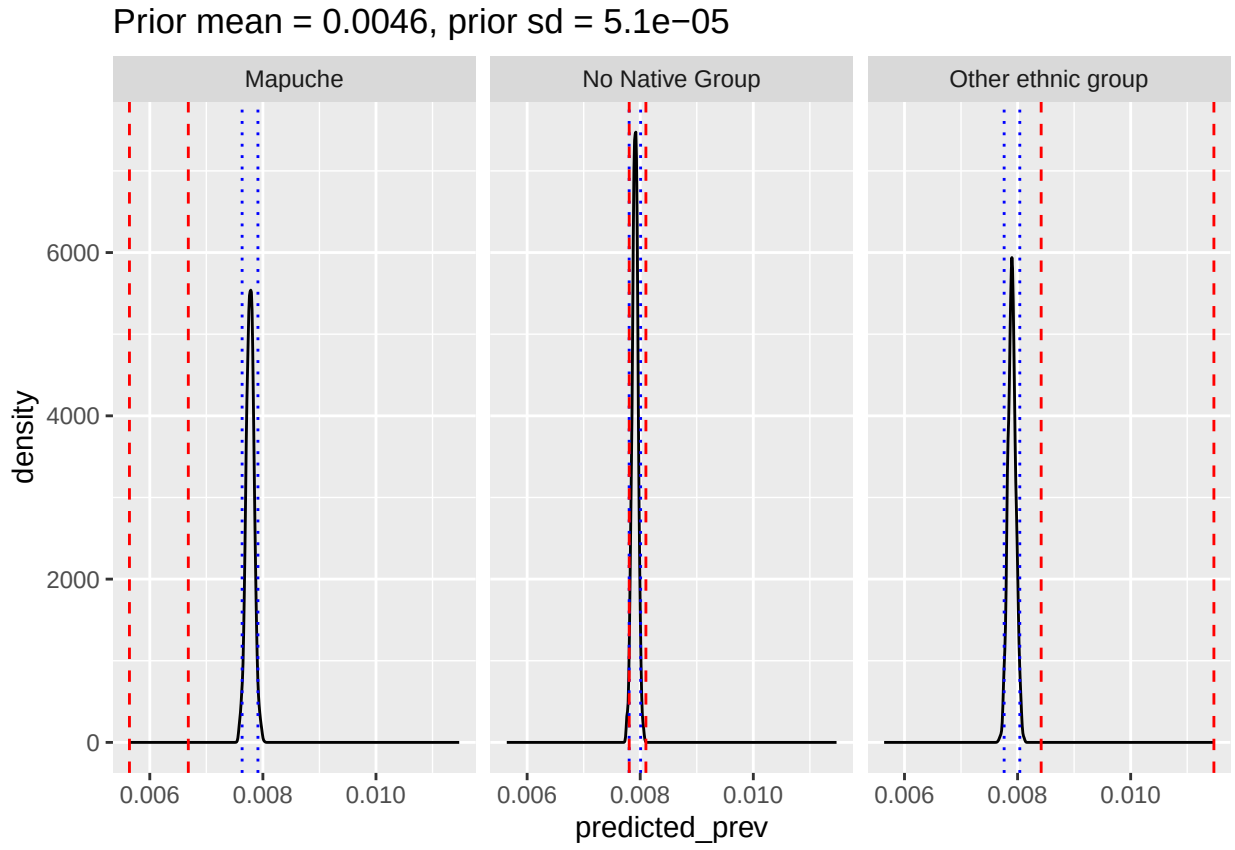



```
# Males
aut_prev_ethnic_m <- chile_bayes_aut %>%
  filter(sex == 1) %>%
  get_grouped_prev(stdpop = chile_stdpop_m, grouping_vars = c("ethnic_2_group", "age_june30", "sex", "au
```

```
## `summarise()` has grouped output by 'ethnic_2_group', 'age_june30', 'sex'. You
## can override using the `.groups` argument.
```

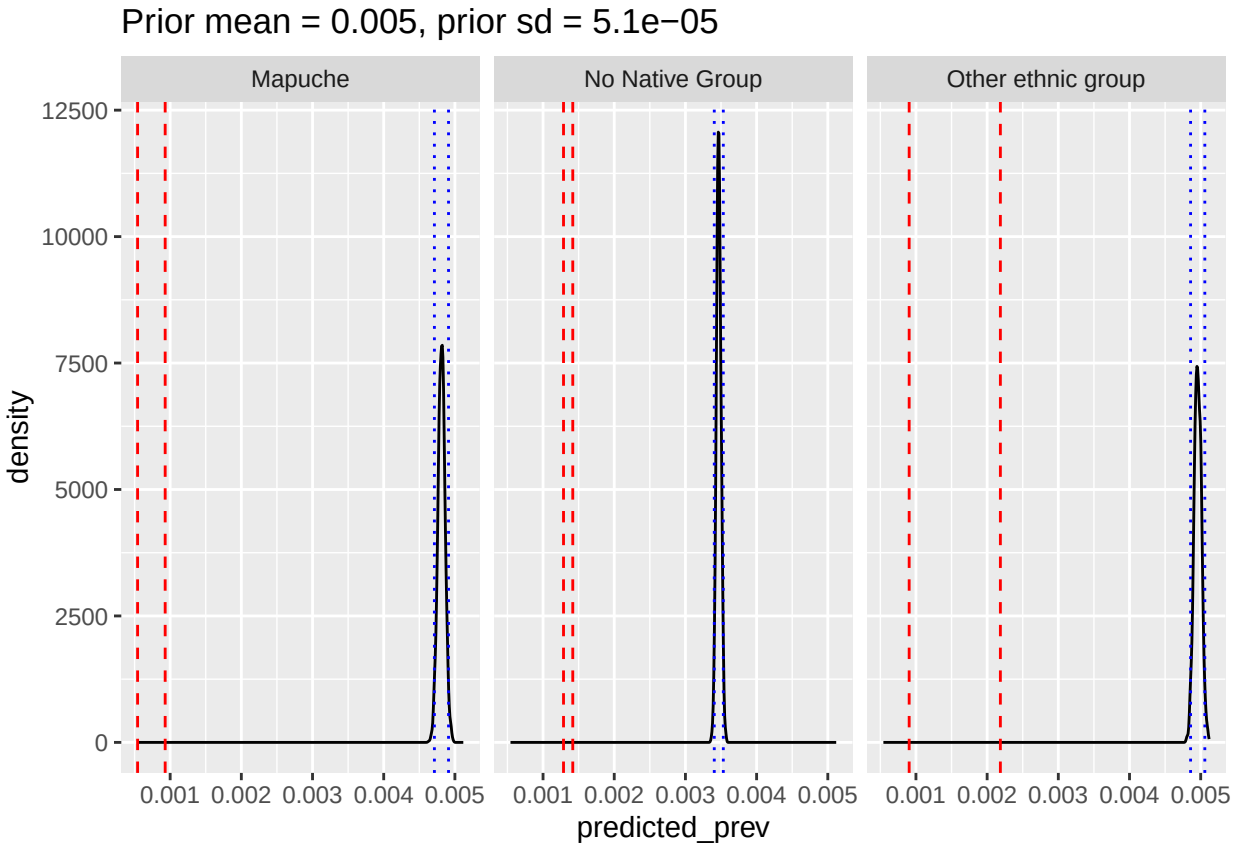
```
aut_prev_ethnic_adj_m <- get_adjusted_prev(aut_prev_ethnic_m, grouping_vars = "ethnic_2_group")

aut_prev_ethnic_post_m <- do_jags_rand_model(x = aut_prev_ethnic_adj_m,
                              feat = "ethnic_2_group",
                              model = rand_model,
                              theta_mu = theta_mu_prior_m,
                              theta_sigma = theta_sigma_prior_m,
                              pars = pars,
                              convergence_checks = FALSE) %>%
  rename("ethnic_2_group" = "Feat_names")

plot_post_density(aut_prev_ethnic_post_m, aut_prev_ethnic_adj_m, feat = "ethnic_2_group", theta_mu = the
```

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

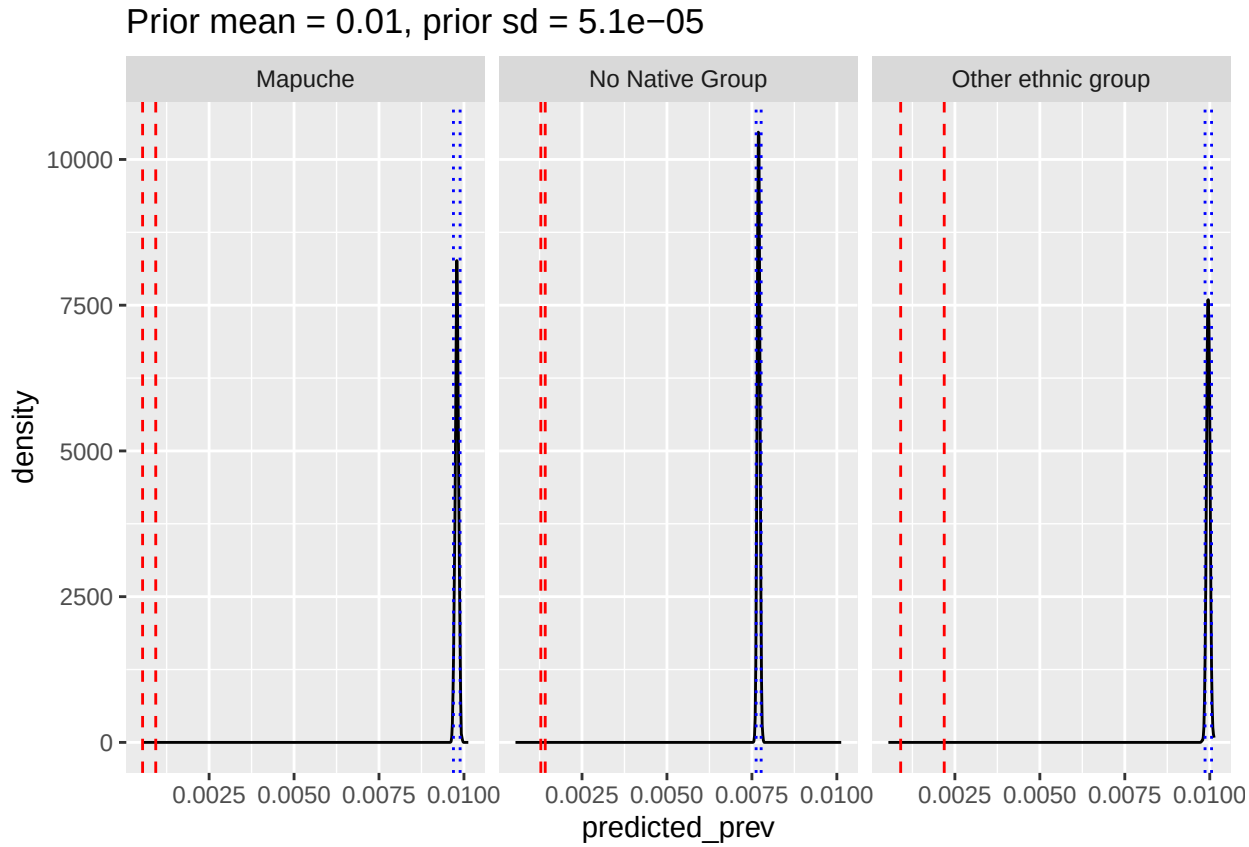



```
# Females
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_ethnic_post_f <- do_jags_rand_model(x = aut_prev_ethnic_adj_f,
                                               feat = "ethnic_2_group",
                                               model = rand_model,
                                               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_f,
                    aut_prev_ethnic_adj_f,
                    feat = "ethnic_2_group",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])

}
```

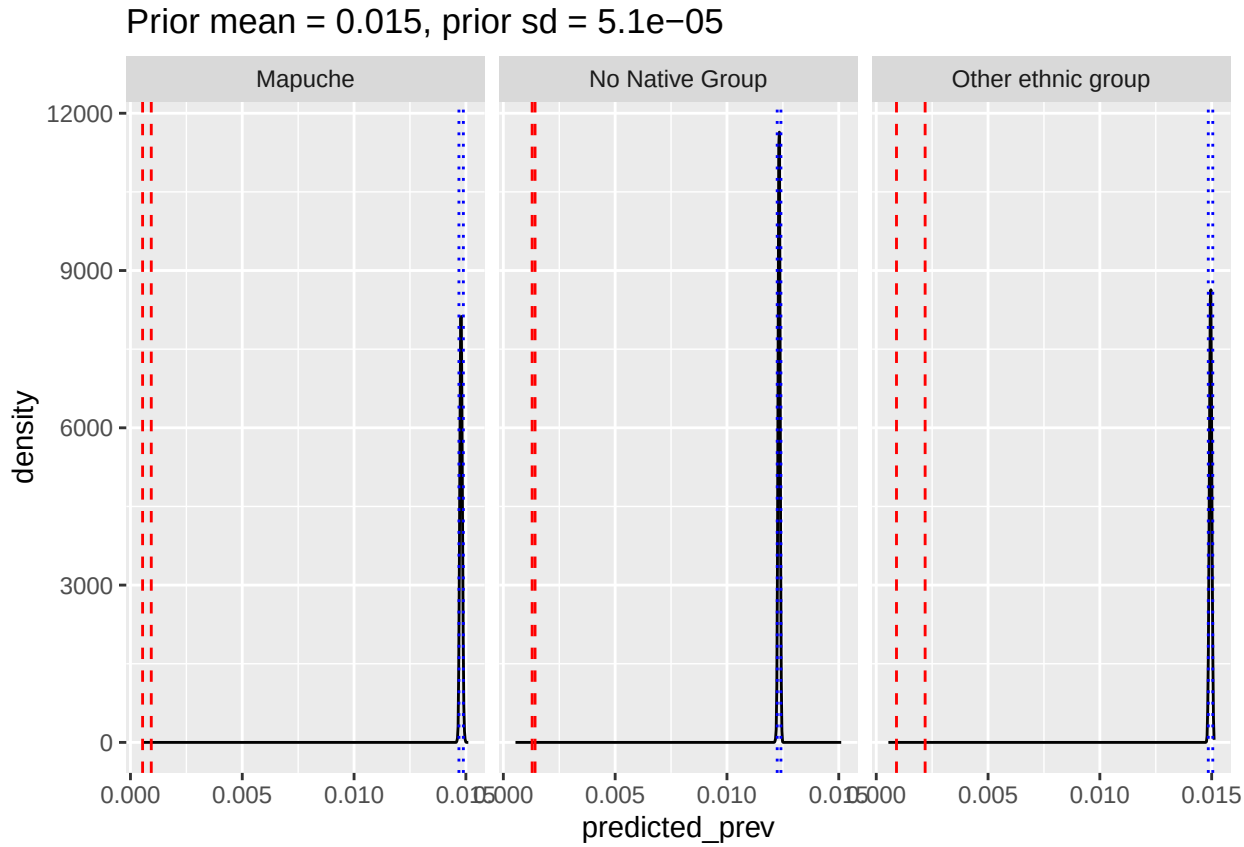
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



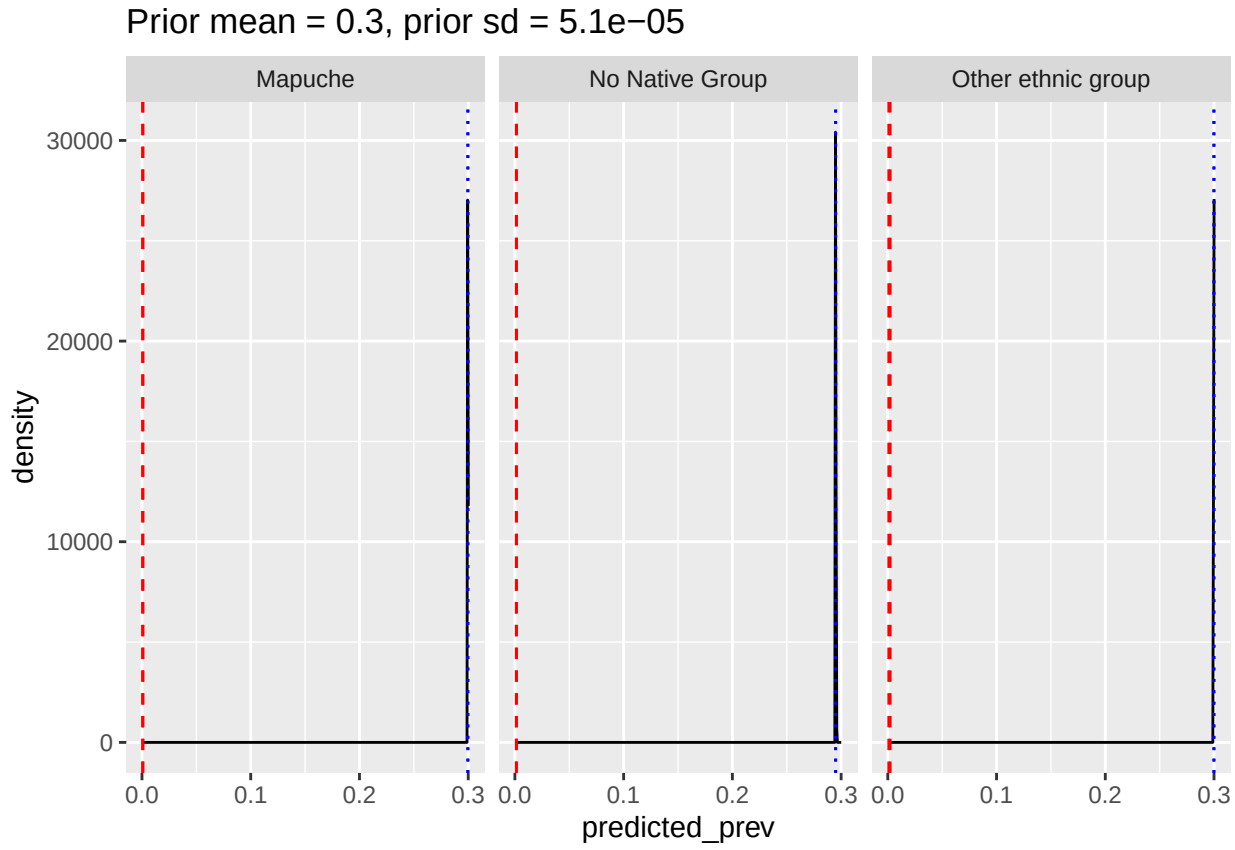

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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

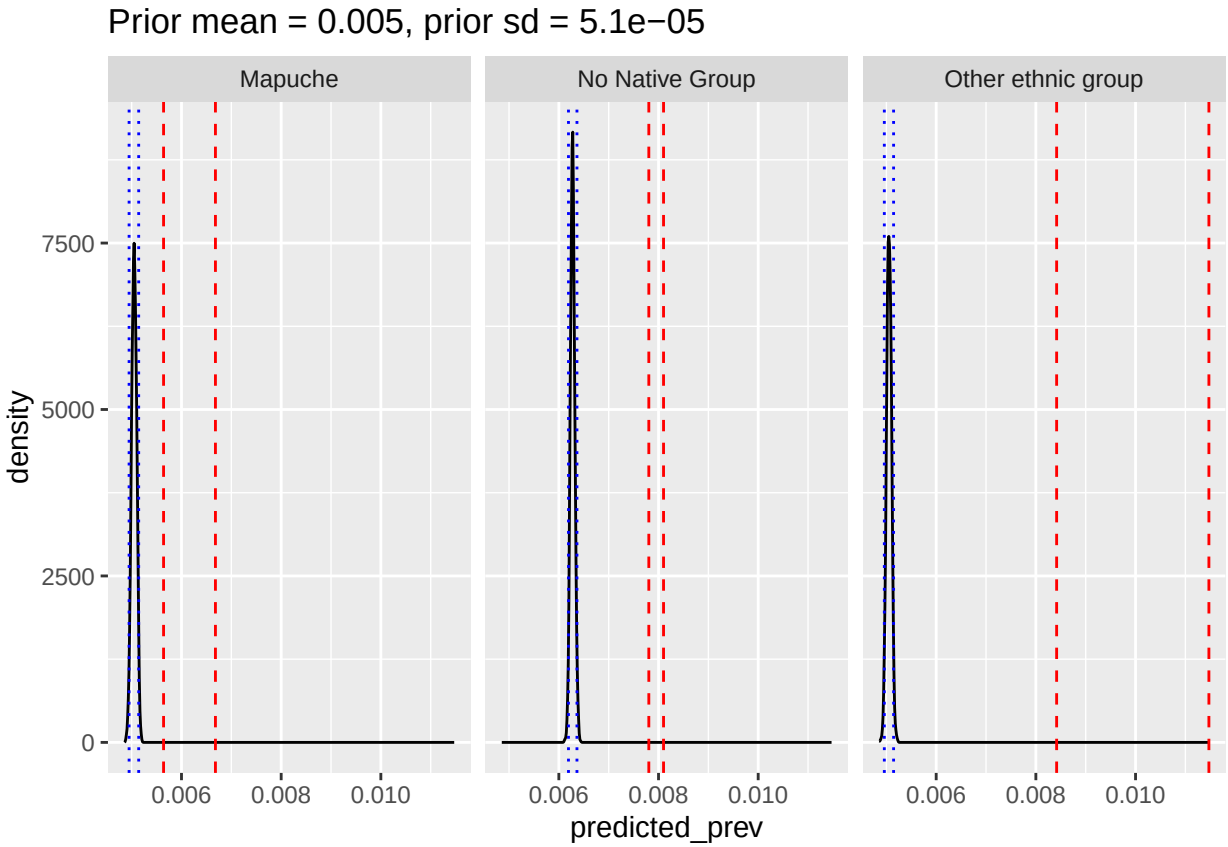



```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

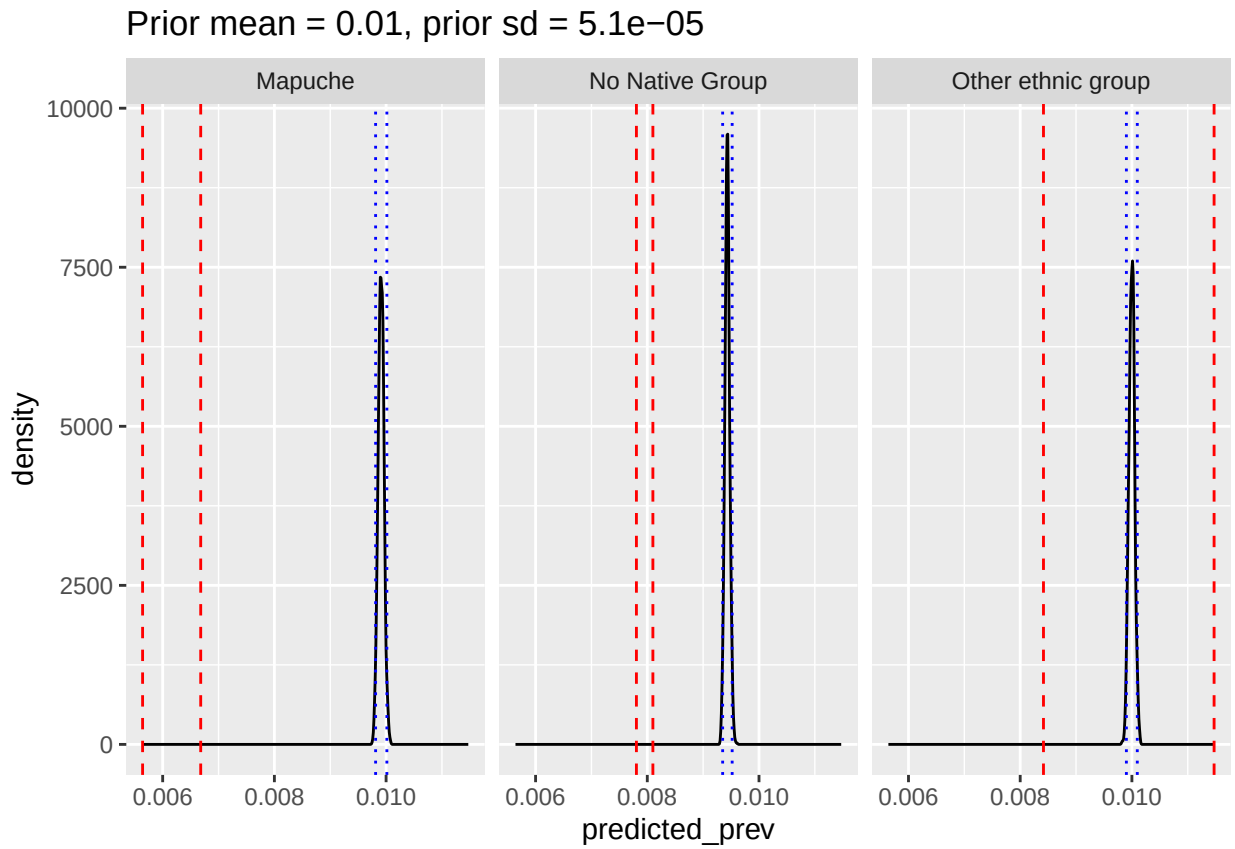



```
# Males
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_ethnic_post_m <- do_jags_rand_model(x = aut_prev_ethnic_adj_m,
                                               feat = "ethnic_2_group",
                                               model = rand_model,
                                               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_m,
                    aut_prev_ethnic_adj_m,
                    feat = "ethnic_2_group",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])

}
```

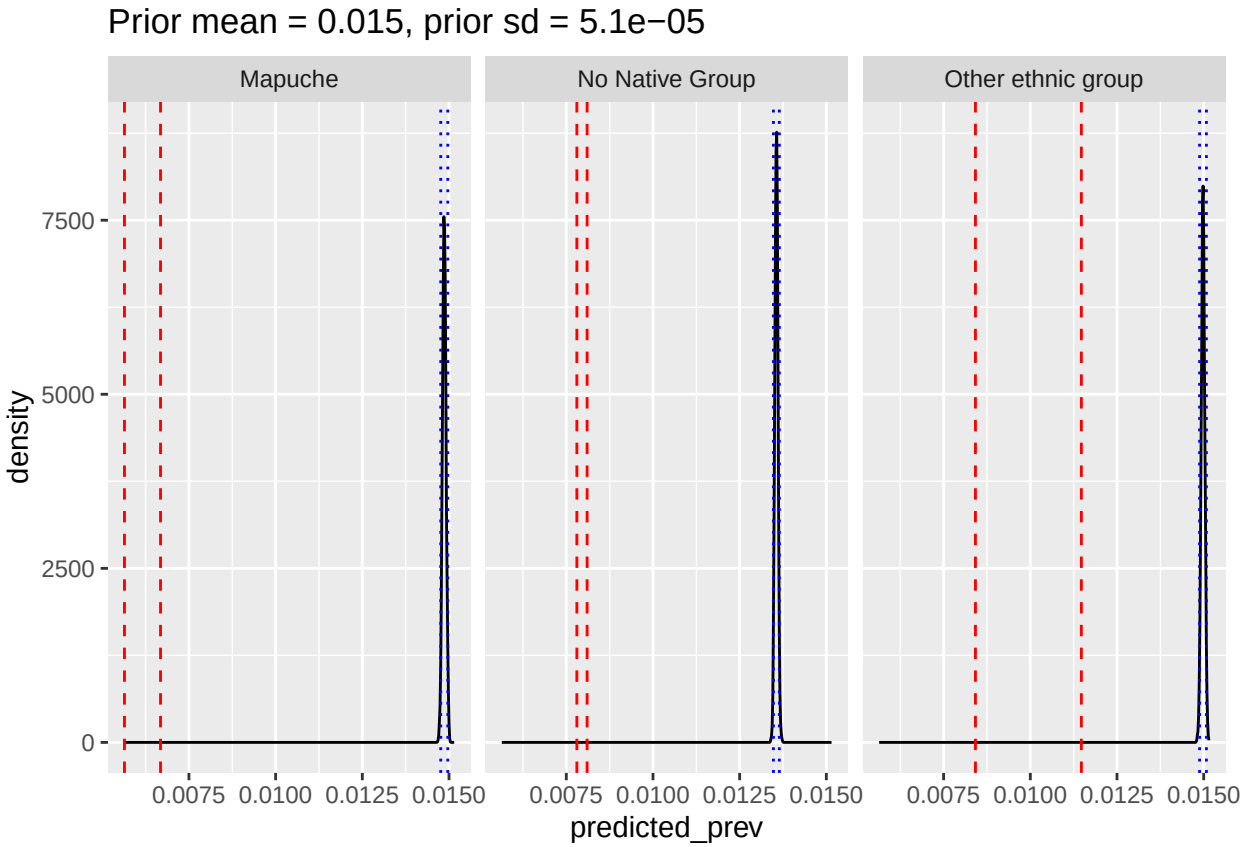
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



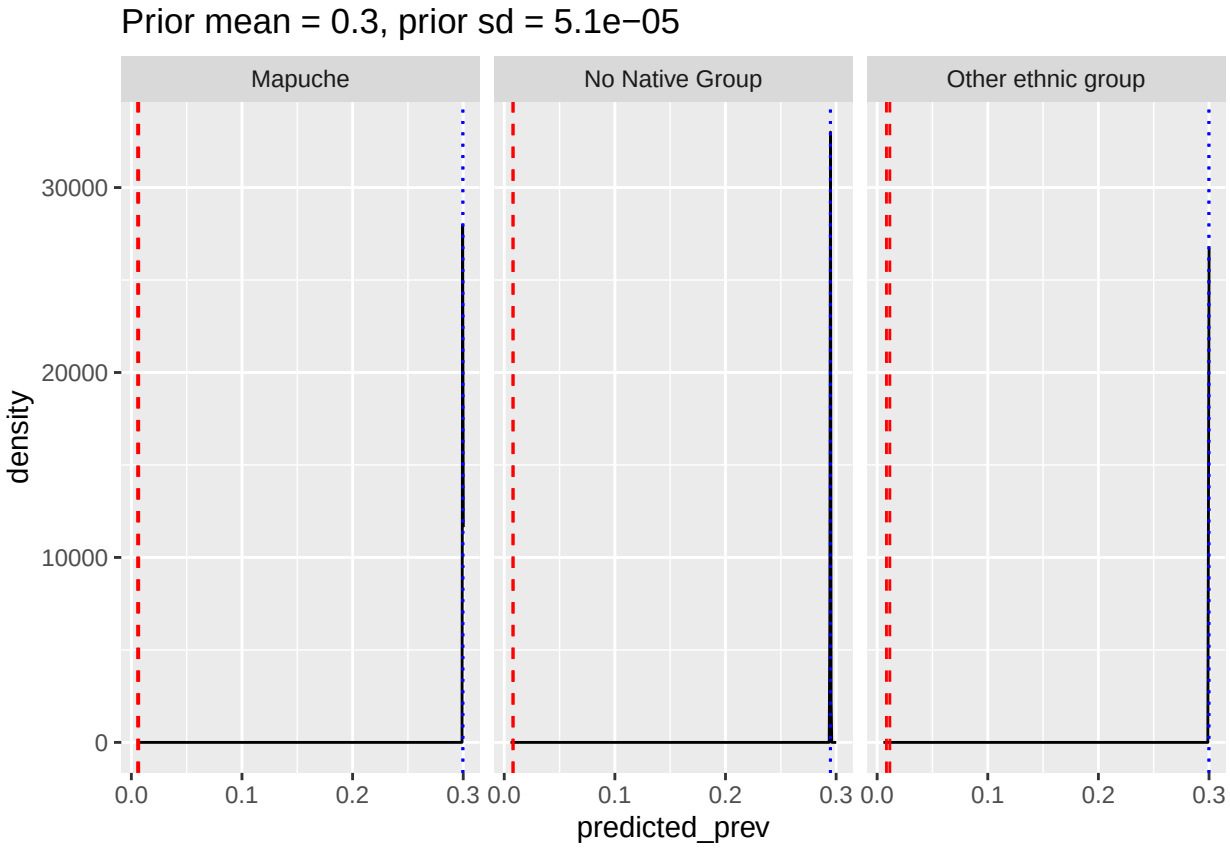

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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




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



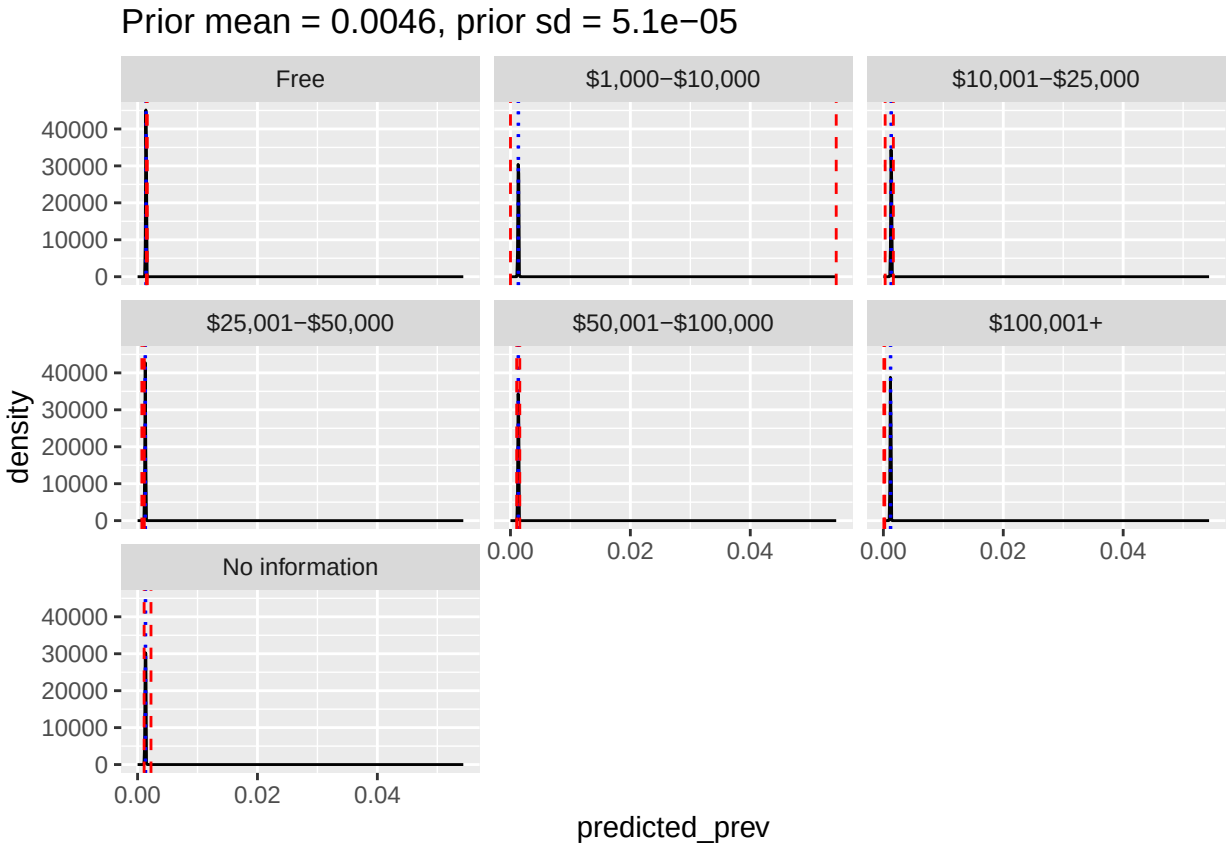
Random effect on economic status, sexes separate

```
# Females
aut_prev_econ_f <- chile_bayes_aut %>%
  filter(sex == 2) %>%
  get_grouped_prev(stdpop = chile_stdpop_f, grouping_vars = c("school_fee", "age_june30", "sex", "autis
```

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

```
aut_prev_econ_adj_f <- get_adjusted_prev(aut_prev_econ_f, grouping_vars = "school_fee")

aut_prev_econ_post_f <- do_jags_rand_model(x = aut_prev_econ_adj_f,
                              feat = "school_fee",
                              model = rand_model,
                              theta_mu = theta_mu_prior_f,
                              theta_sigma = theta_sigma_prior_f,
                              pars = pars,
                              convergence_checks = FALSE) %>%
  rename("school_fee" = "Feat_names")

plot_post_density(aut_prev_econ_post_f, aut_prev_econ_adj_f, feat = "school_fee", theta_mu = theta_mu_pr
```

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

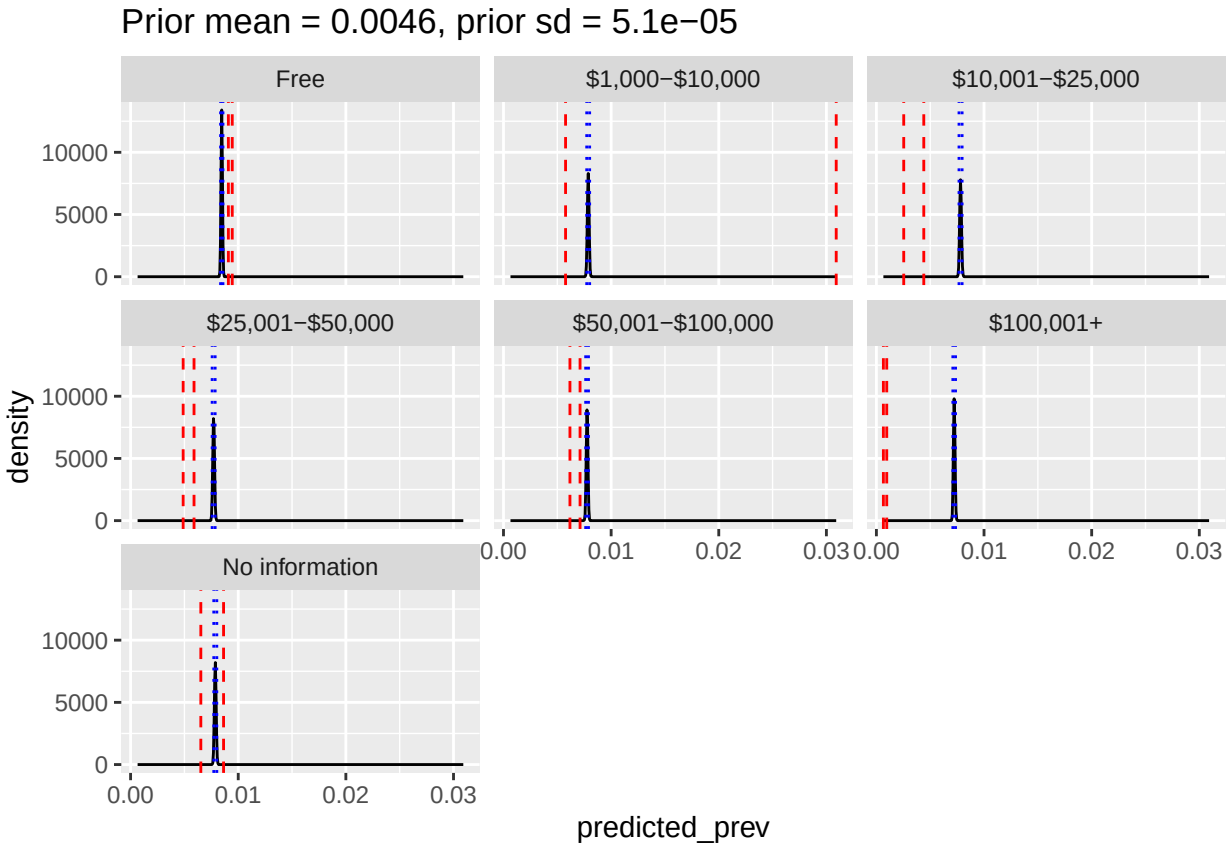



```
# Males
aut_prev_econ_m <- chile_bayes_aut %>%
  filter(sex == 1) %>%
  get_grouped_prev(stdpop = chile_stdpop_m, grouping_vars = c("school_fee", "age_june30", "sex", "autis
```

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

```
aut_prev_econ_adj_m <- get_adjusted_prev(aut_prev_econ_m, grouping_vars = "school_fee")

aut_prev_econ_post_m <- do_jags_rand_model(x = aut_prev_econ_adj_m,
                               feat = "school_fee",
                               model = rand_model,
                               theta_mu = theta_mu_prior_m,
                               theta_sigma = theta_sigma_prior_m,
                               pars = pars,
                               convergence_checks = FALSE) %>%
  rename("school_fee" = "Feat_names")

plot_post_density(aut_prev_econ_post_m, aut_prev_econ_adj_m, feat = "school_fee", theta_mu = theta_mu_p
```

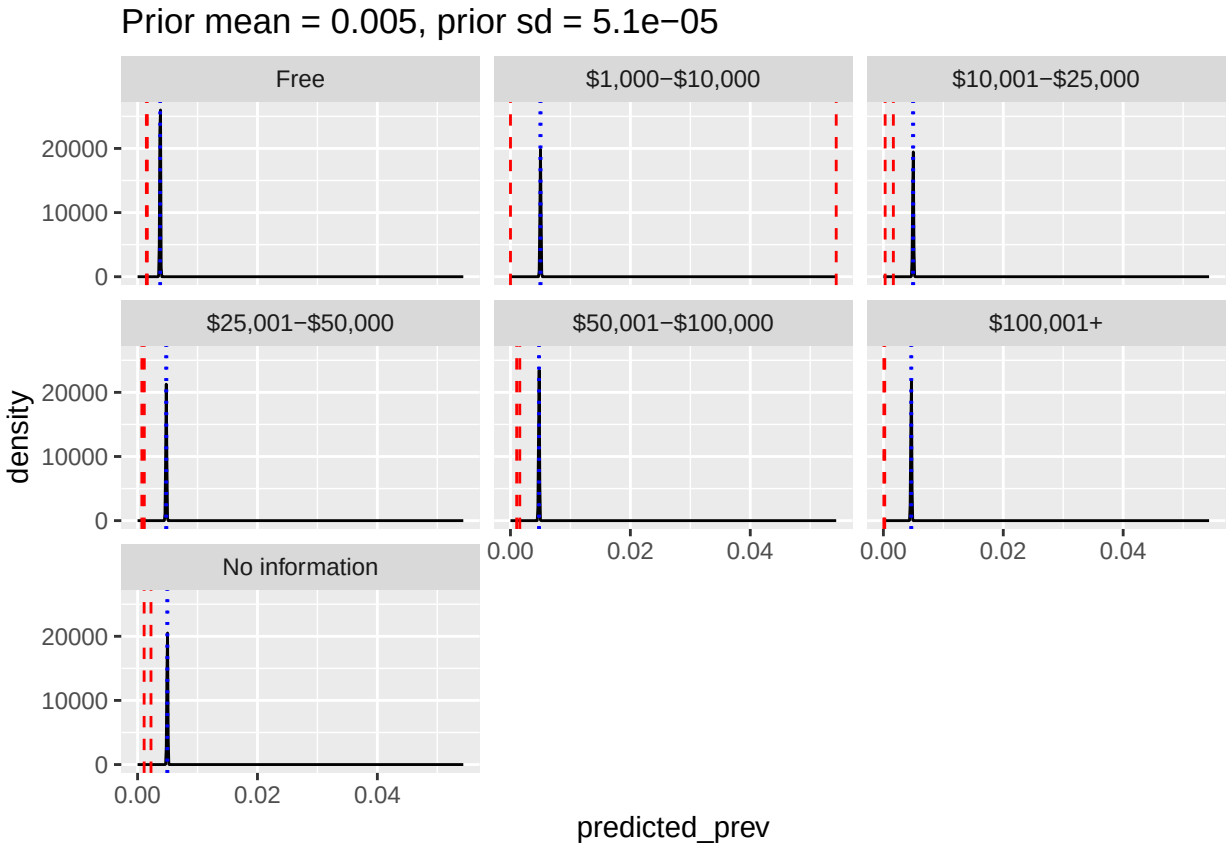
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



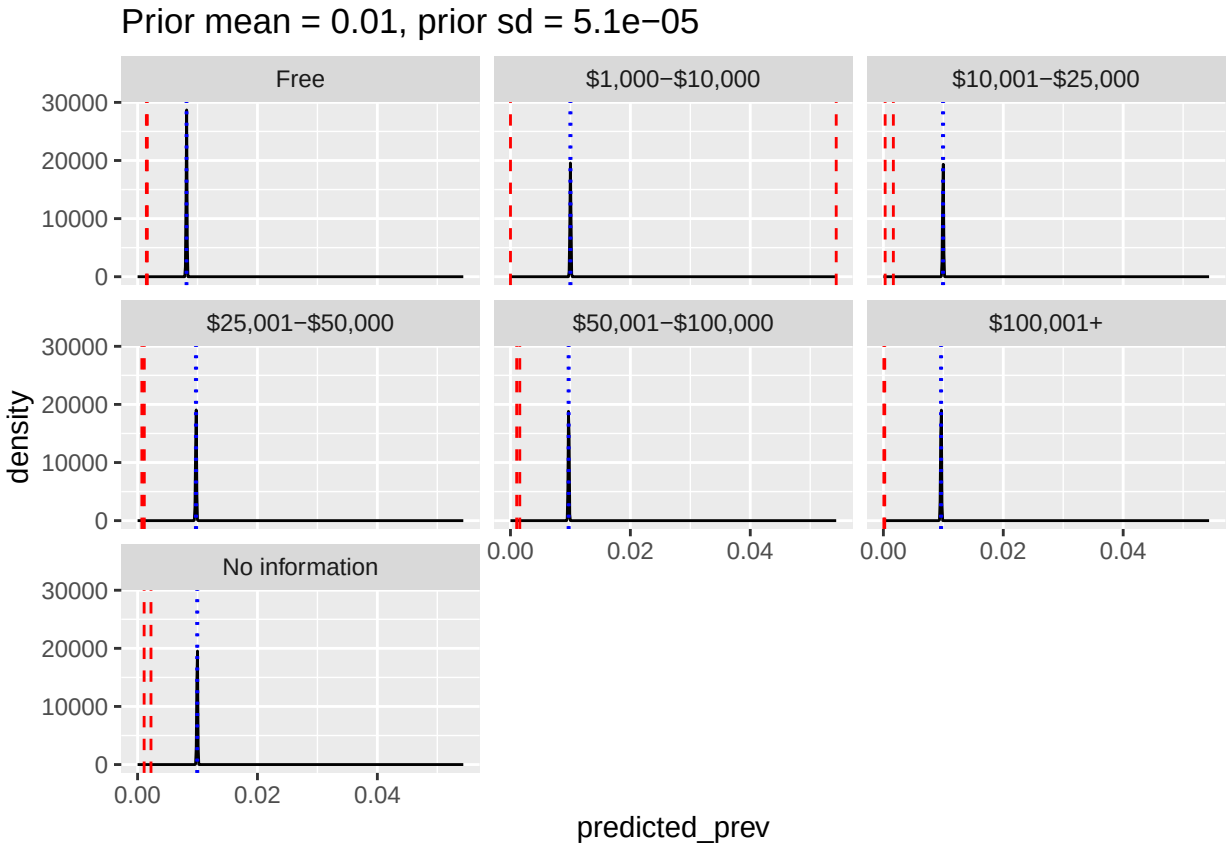
Maybe need to redefine gamma upper CI for female 1,000-10,000 because there are 0 cases.

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

  plot_post_density(aut_prev_econ_post_f,
                    aut_prev_econ_adj_f,
                    feat = "school_fee",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])

}
```

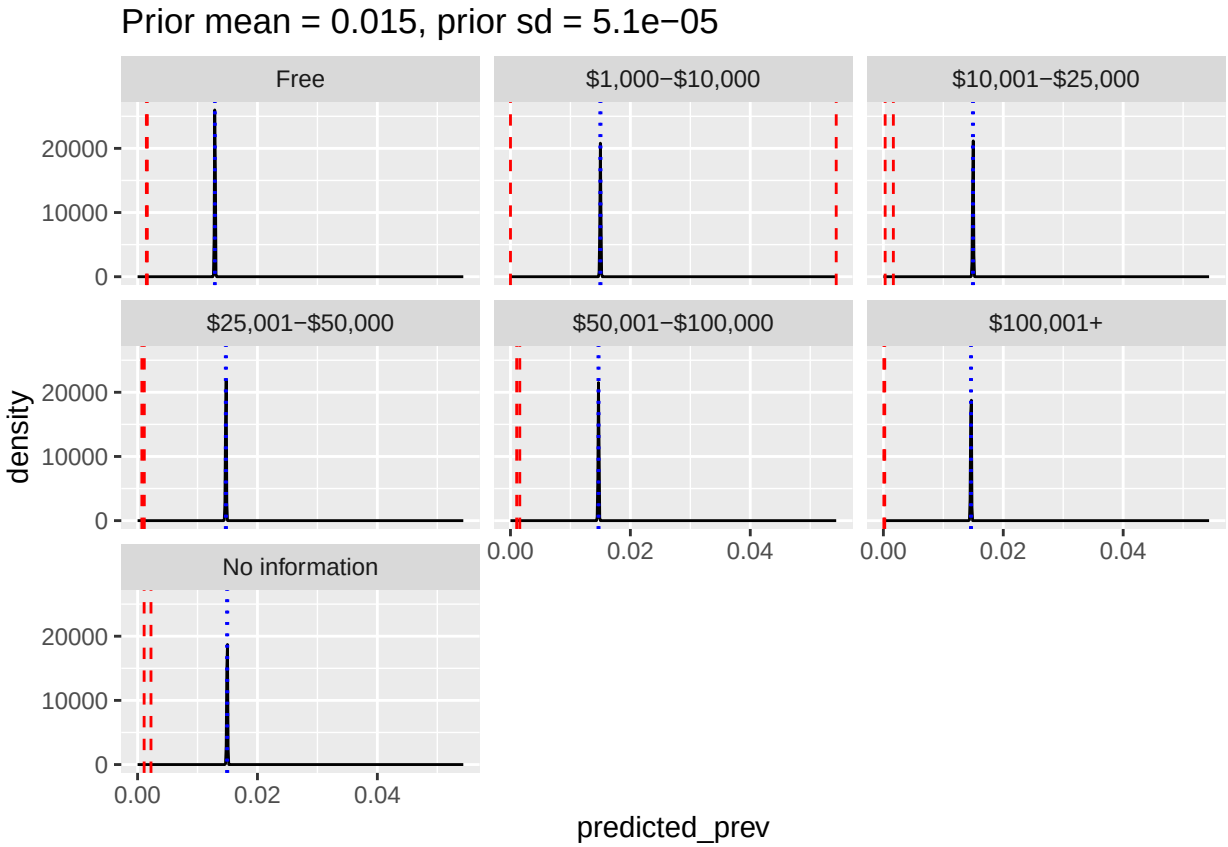
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



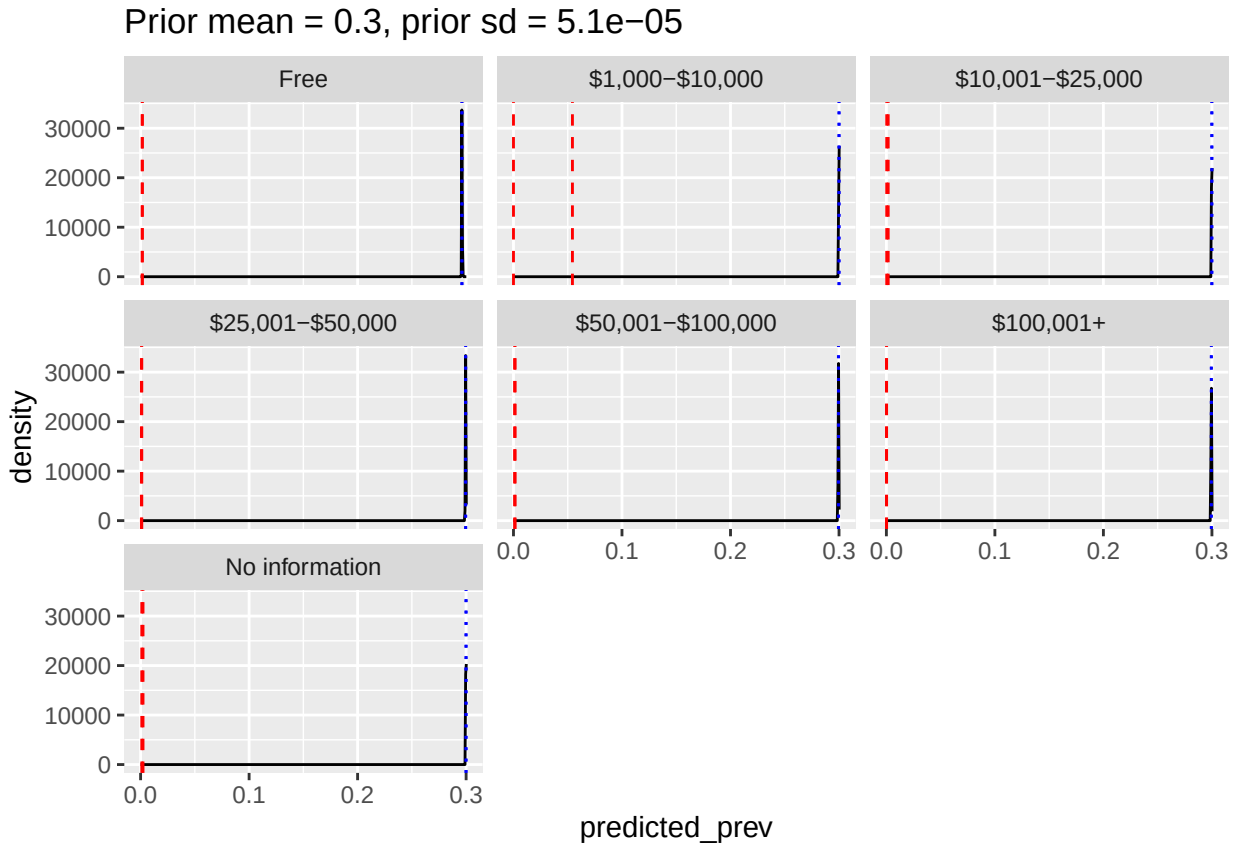

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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

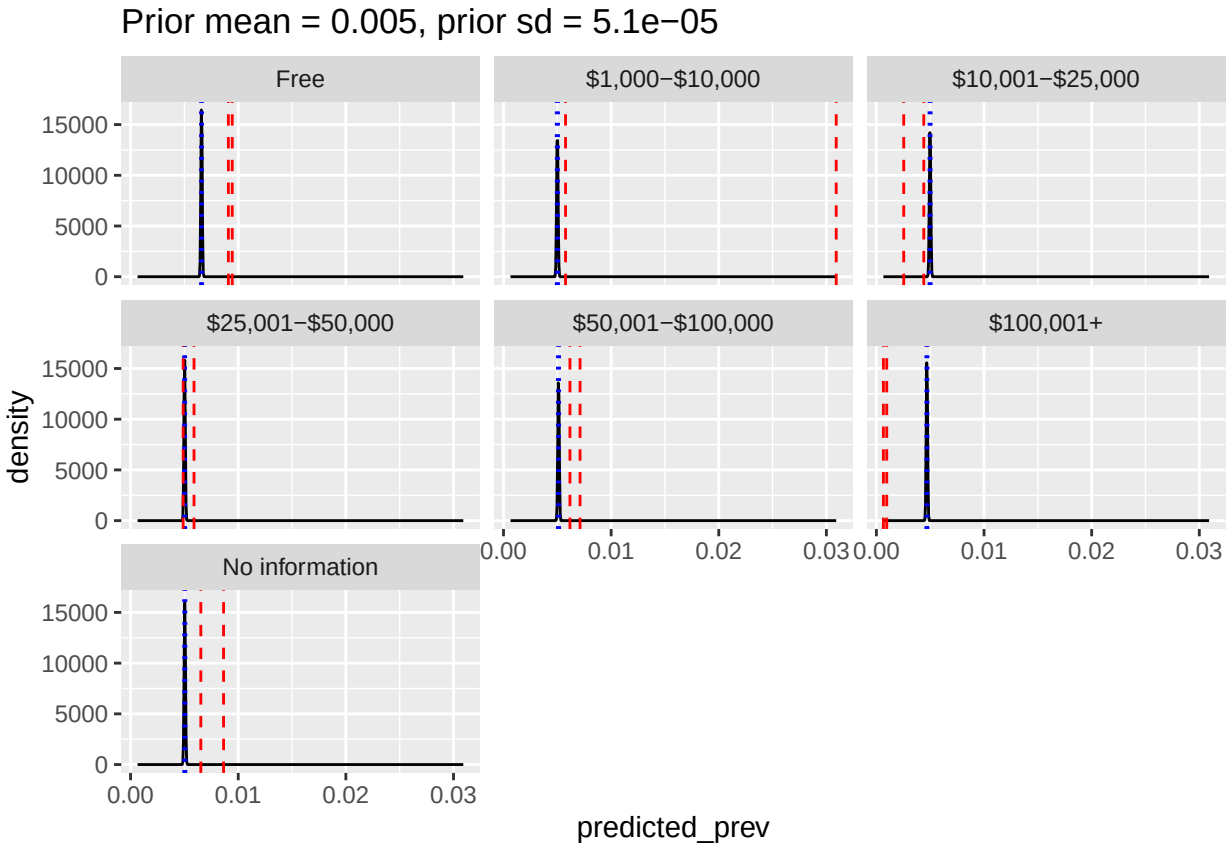



```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

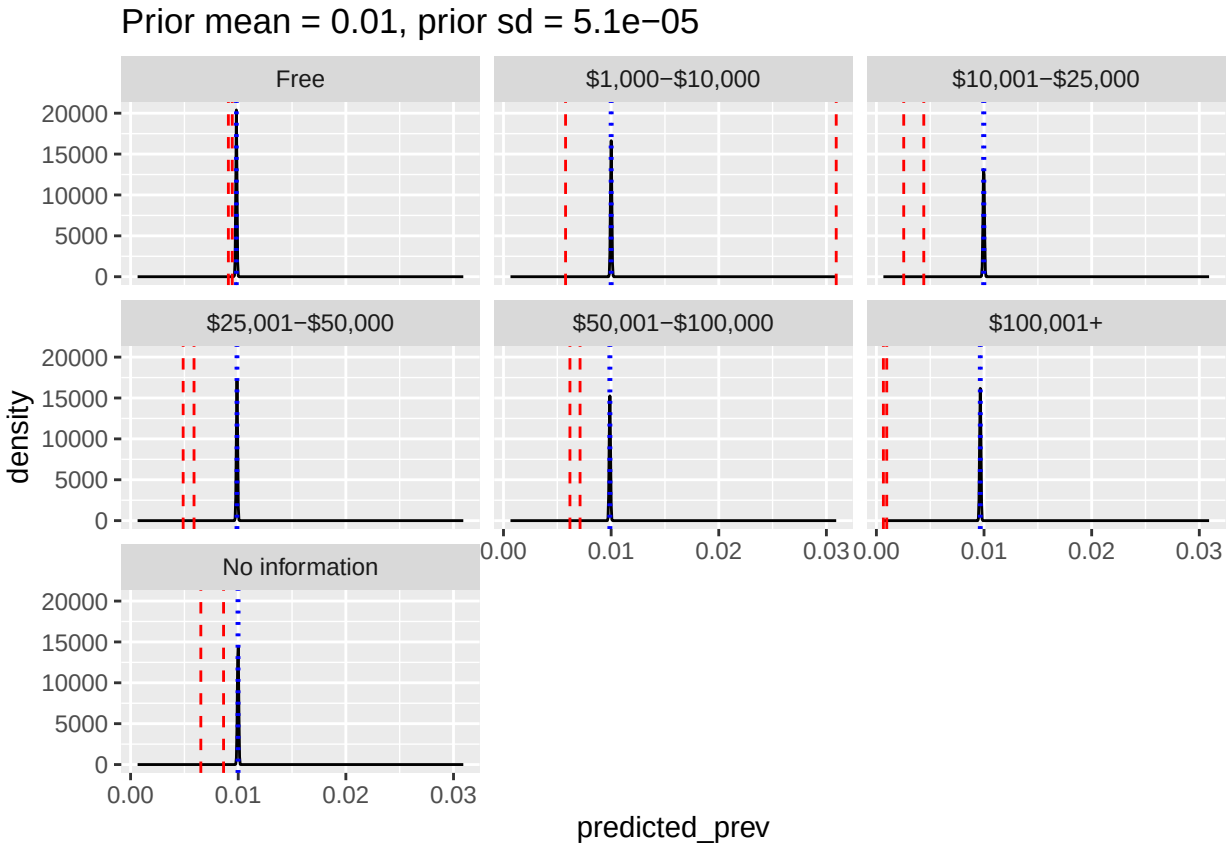



```
# Males
for(j in 1:length(theta_mu_extrapolate)) {
  aut_prev_econ_post_m <- do_jags_rand_model(x = aut_prev_econ_adj_m,
                                             feat = "school_fee",
                                             model = rand_model,
                                            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

  plot_post_density(aut_prev_econ_post_m,
                    aut_prev_econ_adj_m,
                    feat = "school_fee",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])

}
```

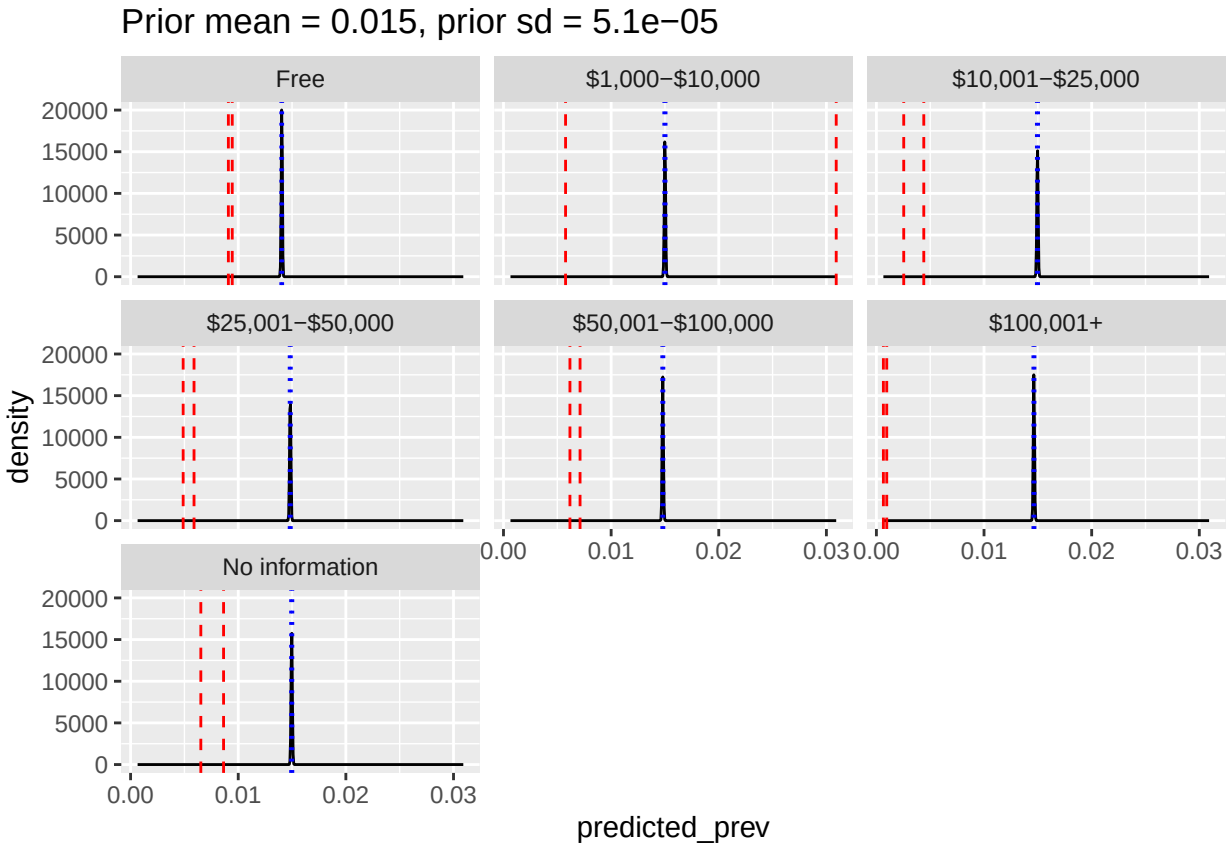
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```



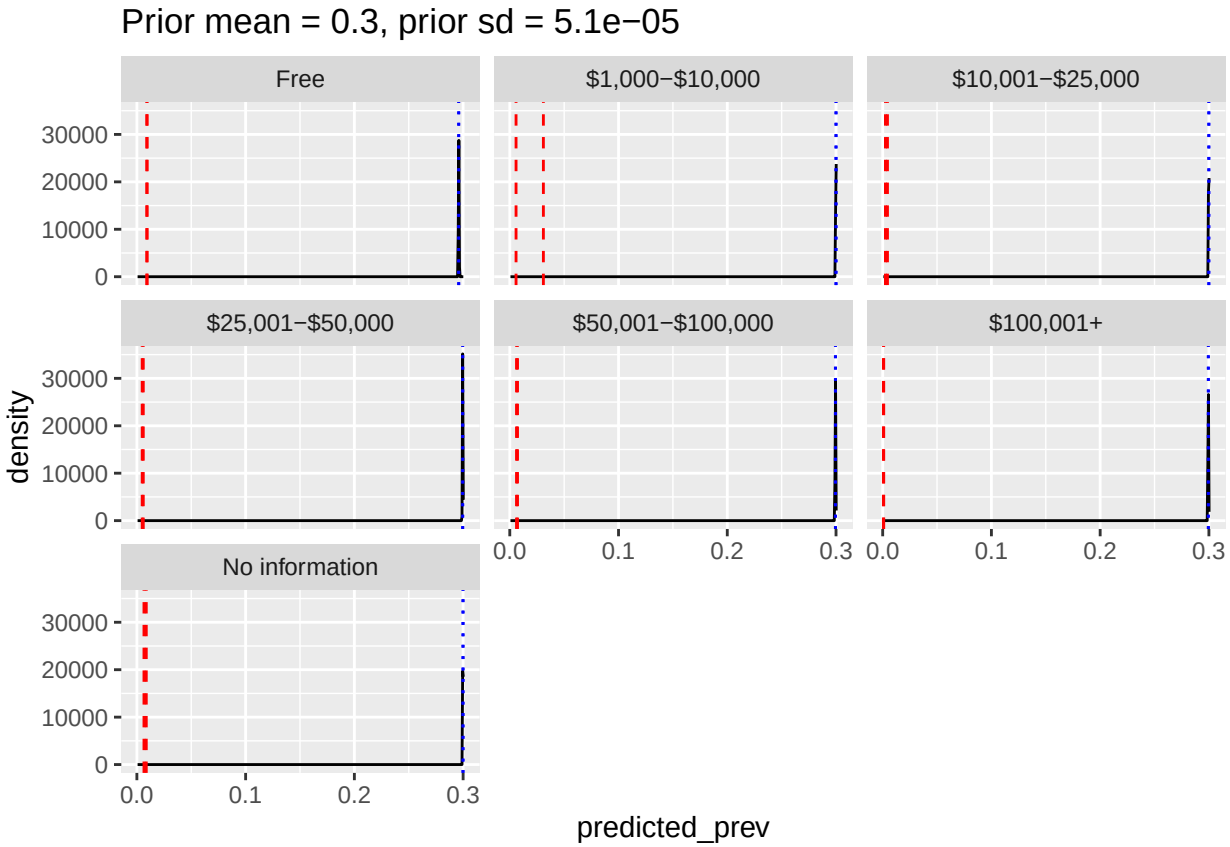

```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```




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




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



Could do random effect on sex and region together