

Bayesian prevalence analysis of ADHD prevalence in Chile

2023-06-22

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

library(nleqslv) # Only needed for robinice bayesian prevalence
library(janitor)
library(gridExtra)
library(readxl)
library(viridis)
library(wesanderson)
library(rgbif)
library(chilemapas) # For commune maps
library(leaflet)
library(plotly)
library(sf)
library(sp)
library(broom)
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 impaired
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_2, # The second one has DD
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 = ADHD, 203 = ADHD. See
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 <- read_excel("04_Data/Outputs/region_service_commune.xlsx") %>%
  clean_names() %>%
  rename(school_region_name_abr = region_name,
         school_commune_name = commune_name)

# https://www.ine.gob.cl/estadisticas/sociales/demografia-y-vitales/proyecciones-de-poblacion
chile_regionpop_raw <- read_excel("04_Data/ine_estimaciones-y-proyecciones-2002-2035_base-2017_region_a"
  clean_names())

chile_regionpop <- chile_regionpop_raw %>%
  mutate(nombre_region = ifelse(region == "Aysén", "Aysen del General Carlos Ibanez del Campo",
                                ifelse(region == "Biobío", "Biobio",

```

```

ifelse(region == "La Araucanía", "La Araucania",
       ifelse(region == "Los Ríos", "Los Rios",
              ifelse(region == "Magallanes", "Magallanes y de la
                 ifelse(region == "Metropolitana", "Metropol
                    ifelse(region == "Ñuble", "Nuble",
                           ifelse(region == "O'Higgins",
                                  ifelse(region == "Tarapacá",
                                         ifelse(region == "Aysén",
                                                ifelse(region == "B
                                                   ifelse(region == "Biobío",
                                                      ifelse(region == "La Araucanía",
                                                         "La Araucania",
                                                         ifelse(region == "Los Ríos",
                                                            "Los Rios",
                                                               ifelse(region == "Magallanes",
                                                                  "Magallanes y de la
                                                                    ifelse(region == "Metropolitana",
                                                                       "Metropol
                                                                          ifelse(region == "Ñuble",
                                                                             "Nuble",
                                                                               ifelse(region == "O'Higgins",
                                                                                      ifelse(region == "Tarapacá",
                                                                                         ifelse(region == "Aysén",
                                                                                            ifelse(region == "B
                                                              
# https://www.ine.gob.cl/estadisticas/sociales/ingresos-y-gastos/encuesta-suplementaria-de-ingresos
# Income is from Supplementary Income Survey (ESI) (2021), data is from Instituto Nacional de Estadística y Censo
chile_regionincome_raw <- read_excel("04_Data/ingreso-medio-mensual-por-región-2010---2021.xlsx", skip = 1, n_max = 10000)
clean_names()

chile_regionincome <- chile_regionincome_raw %>%
  filter(ano == 2021,
         region != "Nacional") %>%
  mutate(region_short = ifelse(grepl("Región de ", region), substr(region, start = 11, stop = nchar(region) - 1),
                               ifelse(grepl("Región del ", region), substr(region, start = 12, stop = nchar(region) - 1),
                                      ifelse(region == "Región Metropolitana", "Metropolitana", NA))),
        nombre_region = ifelse(region_short == "Aysén", "Aysen del General Carlos Ibanez del Campo",
                               ifelse(region_short == "Biobío", "Biobio",
                                     ifelse(region_short == "La Araucanía", "La Araucania",
                                         ifelse(region_short == "Los Ríos", "Los Rios",
                                               ifelse(region_short == "Magallanes", "Magallanes y de la
                                                 ifelse(region_short == "Metropolitana", "Metropol
                                                    ifelse(region_short == "Ñuble", "Nuble",
                                                       ifelse(region_short == "O'Higgins",
                                                          ifelse(region_short == "Tarapacá",
                                                             ifelse(region_short == "Aysén",
                                                                ifelse(region_short == "B
                                                              
select(nombre_region, ingreso_medio_nominal)

```

Try Bayesian analysis of ADHD 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 ADHD Detection in Very Young Children” Josephine Barbaro, Nancy Sadka, Melissa Gilbert, et al <https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2789926>

```
chile_bayes_ADHD <- chile_merged %>%
  filter(age_june30 >= 6 & age_june30 <= 18,
        #special_needs_status == 1,
        sex != 0) %>%
  mutate(ADHD = ifelse(special_needs_code == 203, 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, "12-14", "15-18")),
        # 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-$50,000",
                            ifelse(school_fee == "$50.001 A $100.000", "$50,001-$100,000",
                                   ifelse(school_fee == "MAS DE $100.000", "More than $100,000",
                                          ifelse(school_fee == "SIN INFORMACION", "No information"
                                               school_fee = factor(school_fee, levels = c("Free", "$1,000-$10,000", "$10,001-$25,000", "$25,001-$50,000", "$50,001-$100,000", "More than $100,000", "No information")
                                               school_fee_group = ifelse(school_fee == "Free", "Free",
                                                               ifelse(school_fee %in% c("$1,000-$10,000", "$10,001-$25,000", "$25,001-$50,000", "$50,001-$100,000", "More than $100,000"), "High",
                                                               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_comunes, by = c("school_commune_name", "school_region_name_abr")) %>%
                                                               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 ADHD
                                                               ADHD,
                                                               school_commune_name,
                                                               health_service_name
                                                               )
                                                               )

# Prevalence of ADHD in Chile dataset
sum(chile_bayes_ADHD$ADHD) / nrow(chile_bayes_ADHD) # 0.00476 = 0.476%, very low

## [1] 0.01512414

# 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_plot <- function(x, grouping_vars) {
  # Calculates sample prevalence and its confidence intervals for supplied feature grouping
  # x = chile_bayes_ADHD, needs columns called ADHD, count

```

```

# grouping_vars = variables in x to group by

x_grouped <- x %>%
  group_by(across(all_of(grouping_vars))) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = ADHD, values_from = count) %>%
  rename("n_noADHD" = "0", "n_ADHD" = "1") %>% #, "age" = "age_june30") %>%
  mutate(n_ADHD = ifelse(is.na(n_ADHD), 0, n_ADHD), # If there are no cases of ADHD in the group, in
         sample_pop_size = n_noADHD + n_ADHD, # Total sample population is ADHD cases + not cases
         sample_prevalence = n_ADHD / sample_pop_size, # Prevalence of ADHD in the group
         ci_lower = sample_prevalence - (1.96 * sqrt(sample_prevalence * (1 - sample_prevalence)) / sam
         ci_upper = sample_prevalence + (1.96 * sqrt(sample_prevalence * (1 - sample_prevalence)) / sam
  ungroup()
return(x_grouped)
}

get_grouped_prev <- function(x, stdpop, grouping_vars) {
  # Calculates sample prevalence, age- and sex-standardised prevalence and group weighting for supplied
  # x = chile_bayes_ADHD, needs columns called ADHD, 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 = ADHD, values_from = count) %>%
    rename("n_noADHD" = "0", "n_ADHD" = "1", "age" = "age_june30") %>%
    mutate(n_ADHD = ifelse(is.na(n_ADHD), 0, n_ADHD), # If there are no cases of ADHD in the group, in
           sample_pop_size = n_noADHD + n_ADHD, # Total sample population is ADHD cases + not cases
           sample_prevalence = n_ADHD / sample_pop_size) %>% # Prevalence of ADHD in the group
  left_join(stdpop, by = c("age", "sex")) %>%
  mutate(ADHD_prev_std = n_ADHD / sample_pop_size * pop_prop, # Prevalence of ADHD in the group, stan
        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 confidence
  # 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_ADHD) / sum(sample_pop_size),
              crude_count = sum(n_ADHD),
              adjusted_rate = sum(n_ADHD / 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_ADHD / sample_pop_size^2),
#se2 = sqrt(sum((std_pop/sum(std_pop))^2 * n_ADHD/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_rate),
                  ci_upper = (var + w_M^2) / (2*(adjusted_rate + w_M)) * qchisq(p = 1-0.05/2, df = 2*(adjusted_rate + w_M))
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$sum_sample_pop_size,
                    ADHD_sample = x$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("ADHD_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, title_text = "") {
  # 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 = ", signif(theta_mu, 3), ", prior sd = ", signif(theta_sigma,
}

```

Set up Chile maps

```
commune_geoms <- data.frame(mapa_comunas)
communes_lookup <- data.frame(codigos_territoriales)
regions_lookup <- communes_lookup %>%
  group_by(codigo_region, nombre_region) %>%
  summarise() %>%
  mutate(school_region_name_abr = ifelse(nombre_region == "Antofagasta", "ANTOF",
                                         ifelse(nombre_region == "Arica y Parinacota", "AYP",
                                               ifelse(nombre_region == "Atacama", "ATCMA",
                                                 ifelse(nombre_region == "Aysen del General Carlos Ibanez del Campo", "AYSEN",
                                                       ifelse(nombre_region == "Biobio", "BBIO",
                                                         ifelse(nombre_region == "Coquimbo", "COQ",
```

```

        ifelse(nombre_region == "La Araucania", "ARAUC",
        ifelse(nombre_region == "Libertador General Bernardo OHiggins", "LGBO",
        ifelse(nombre_region == "Los Lagos", "LAGOS",
        ifelse(nombre_region == "Los Rios", "RIOS",
        ifelse(nombre_region == "Magallanes y de la Antartica Chilena", "MAG",
        ifelse(nombre_region == "Maule", "MAULE",
        ifelse(nombre_region == "Metropolitana de Santiago", "RM",
        ifelse(nombre_region == "Nuble", "NUBLE",
        ifelse(nombre_region == "Tarapaca", "TPCA",
        ifelse(nombre_region == "Valparaiso", "VALPO", NA))))))))))))))),

ADM1_PCODE = paste0("CL", codigo_region))

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

#region_geoms <- generar_circunscripciones(mapa = chilemapas::mapa_comunas) %>% # This has Nuble for all
#   full_join(regions_lookup, by = "codigo_region")

# https://data.humdata.org/dataset/cod-ab-chl
chile.adm1 <- st_read("04_Data/CHL_adm_humdata/chl_admbnda_adm1_bcn_20211008.shp") #%>%

## Reading layer `chl_admbnda_adm1_bcn_20211008` from data source
##   `C:\Users\delat\OneDrive\MPhil Population Health Sciences 2022-2023\12 Dissertation\04_Data\CHL_adm1_bcn_20211008.shp` with driver `ESRI Shapefile'
## Simple feature collection with 16 features and 12 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -109.4549 ymin: -55.98 xmax: -66.41821 ymax: -17.4984
## Geodetic CRS: WGS 84
#st_transform(crs = 32629)

demog_geom <- chile.adm1 %>%
  left_join(regions_lookup, by = "ADM1_PCODE") %>%
  left_join(chile_regionpop, by = "nombre_region") %>%
  left_join(chile_regionincome, by = "nombre_region") %>%
  select(-region)

#chile.adm2 <- st_read("04_Data/CHL_adm/CHL_adm2.shp") # Also doesn't have Biobio
#chile.adm2 <- st_transform(chile.adm2, crs = 32629)

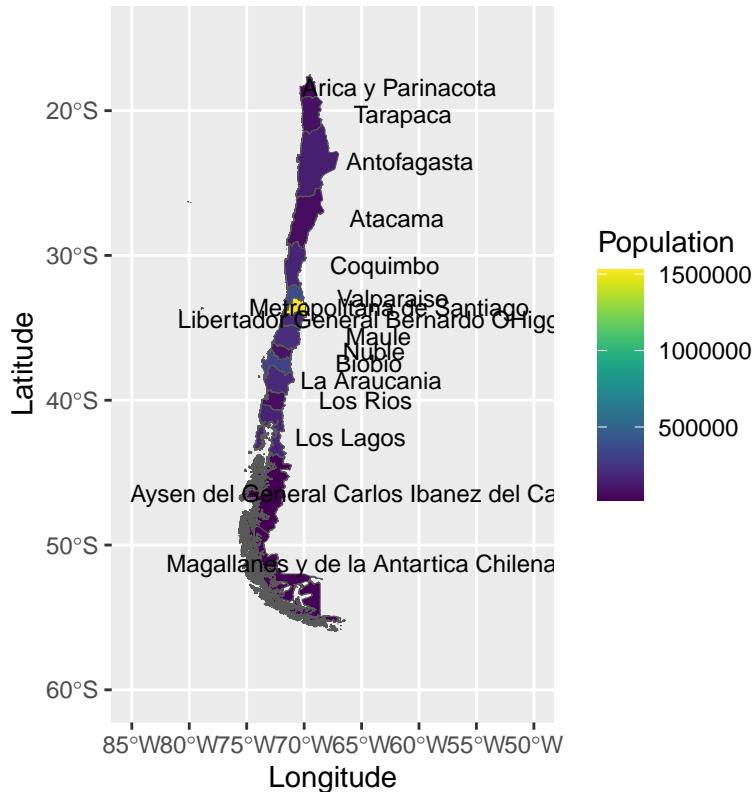
#region_geoms_demog <- region_geoms %>%
#  left_join(chile_regionpop, by = "nombre_region") %>%
#  left_join(chile_regionincome, by = "nombre_region") %>%
#  select(-region)

# Population
ggplot(demog_geom) +
  geom_sf(mapping = aes(geometry = geometry, fill = youth_pop_2021), linewidth = 0.01) +
  geom_sf_text(mapping = aes(geometry = geometry, label = nombre_region), nudge_x = 8, size = 3) +
  scale_fill_viridis(option = "viridis", direction = 1, name = "Population") + #, limits = c(23000, 40000)
  labs(title = "Population aged 0-14 (2021 projections)") +
  xlab("Longitude") +
  ylab("Latitude") +
  coord_sf(xlim = c(-85, -50), ylim = c(-60, -15))

```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```

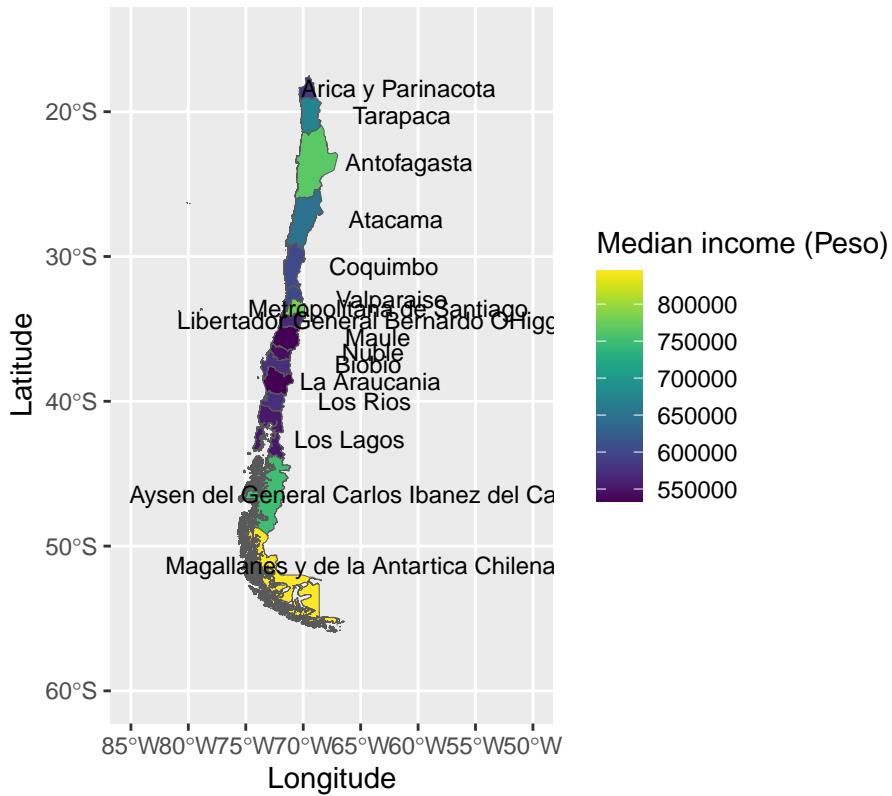
Population aged 0–14 (2021 projections)



```
# Income
ggplot(demog_geom) +
  geom_sf(mapping = aes(geometry = geometry, fill = ingreso_medio_nominal), linewidth = 0.01) +
  geom_sf_text(mapping = aes(geometry = geometry, label = nombre_region), nudge_x = 8, size = 3) +
  scale_fill_viridis(option = "viridis", direction = 1, name = "Median income (Peso)") +
  labs(title = "Net income from main job (2021)") +
  xlab("Longitude") +
  ylab("Latitude") +
  coord_sf(xlim = c(-85, -50), ylim = c(-60, -15))

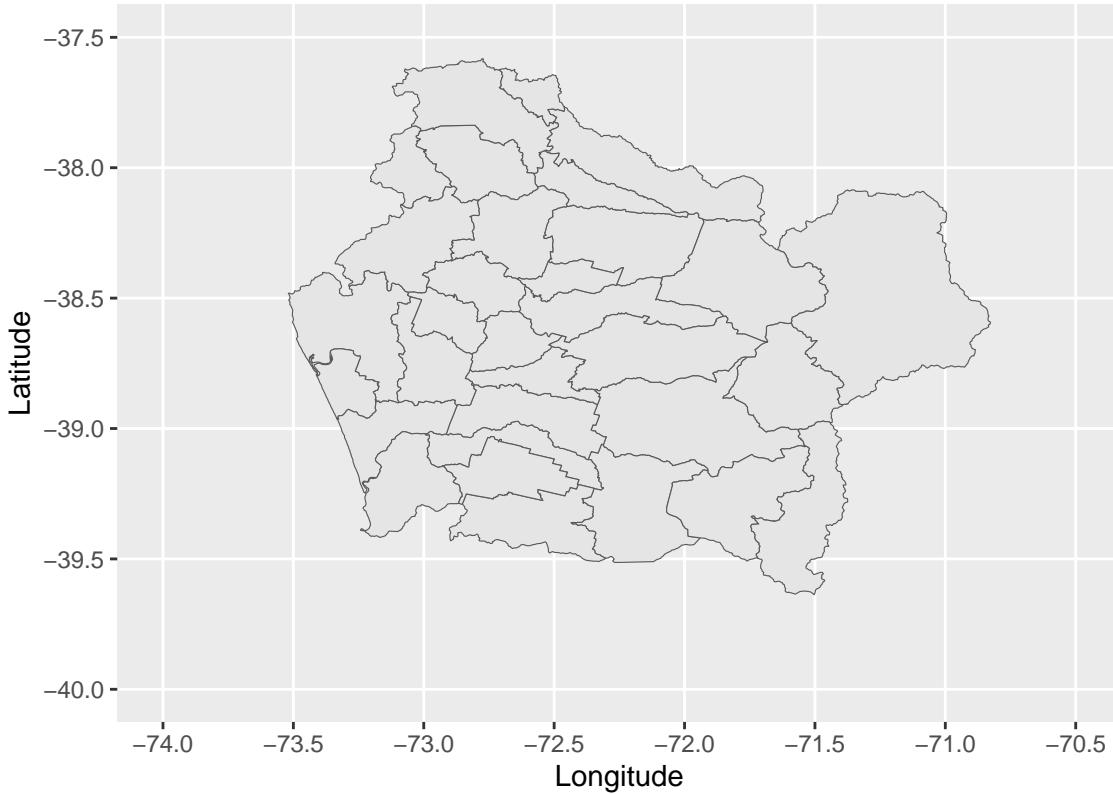
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```

Net income from main job (2021)



```
# ARAUC communes
arauc <- commune_geoms %>% filter(codigo_region == "09")
ggplot(arauc) +
  geom_sf(mapping = aes(geometry = geometry)) +
  coord_sf(xlim = c(-74, -70.5), ylim = c(-40, -37.5)) +
  labs(title = "Communes in La Araucania") +
  xlab("Longitude") +
  ylab("Latitude")
```

Communes in La Araucania



Bayesian prevalence analysis

Find age- and sex-adjusted sample prevalence

```
ADHD_prev <- get_grouped_prev(x = chile_bayes_ADHD, stdpop = chile_stdpop,
                                grouping_vars = c("age_june30", "age_cat_name", "sex", "sex_desc", "ADHD"))

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

ADHD_prev_adj <- ADHD_prev %>%
  summarise(sum_sample_pop_size = sum(sample_pop_size),
           crude_rate = sum(n_ADHD) / sum(sample_pop_size),
           crude_count = sum(n_ADHD),
           adjusted_rate = sum(n_ADHD / sample_pop_size * pop_prop),
           adjusted_count = round(adjusted_rate * sum_sample_pop_size, 0), # had to fudge this to get a whole number
           #adjusted_count = adjusted_rate * sum_sample_pop_size,
           var = sum(pop_prop^2 * n_ADHD / sample_pop_size^2),
           #se2 = sqrt(sum((std_pop/sum(std_pop))^2 * n_ADHD/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_rate)),
           ci_upper = (var + w_M^2) / (2*(adjusted_rate + w_M)) * qchisq(p = 1-0.05/2, df = 2*(adjusted_rate)))
```

Set global parameters

```
nObs <- nrow(chile_bayes_ADHD)
nIter <- 1000
```

```

nBurn <- 1000
pars <- c("theta_a", "theta_b", "theta", "ADHD_sample", "ADHD_pred")

theta_mu_prior <- ADHD_prev_adj$adjusted_rate
theta_sigma_prior <- (ADHD_prev_adj$ci_upper - ADHD_prev_adj$ci_lower) / (2*1.96)
theta_mu_extrapolate <- c(0.02, 0.03, 0.05) # 0.5%, 1%, 1.5%, 3% prevalence
theta_sigma_extrapolate <- c(rep(0.0001/1.96, 3)) # 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)
    ADHD_sample[i] ~ dbin(theta[i], nObs[i])
    ADHD_pred[i] ~ dbin(theta[i], nObs[i])
  }
}"

```

Common effects model with unadjusted or age- and sex-adjusted sample prevalence

Do age- and sex-adjustment (same as below for random effect models)

```

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

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

  ADHD_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)
  #ADHD_post <- ADHD_sample/nObs * sens + (1 - ADHD_sample/nObs) * spec
}"

common_data <- list(theta_a = theta_a_common,
                     theta_b = theta_b_common,
                     nObs = nObs,

```

```

#ADHD_sample = sum(chile_bayes_ADHD$ADHD) # unadjusted sample prevalence
ADHD_sample = ADHD_prev_adj$adjusted_count # age- and sex- adjusted sample prevalence
#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)
)

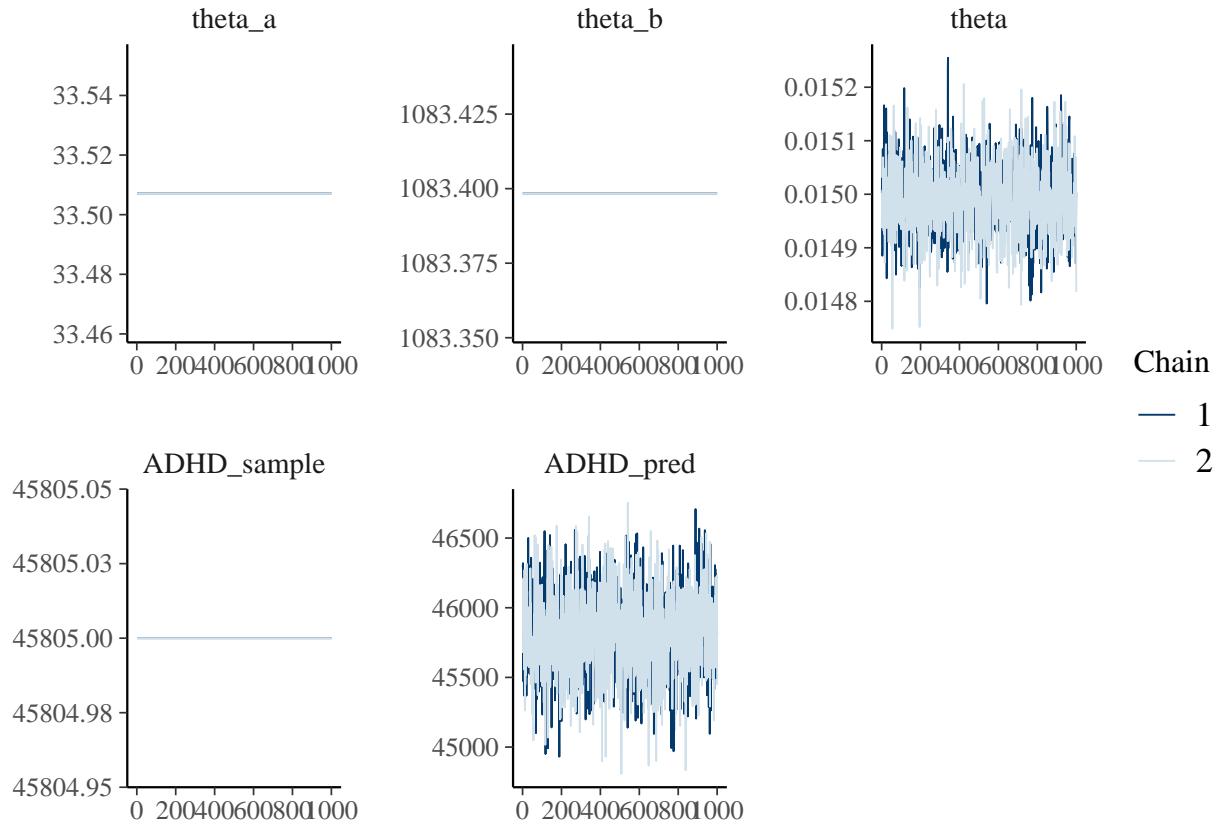
common_ini <- list(list(theta = 0.001), #, spec = 0.5, sens = 0.5),
                     list(theta = 0.01)) #, spec = 0.9, sens = 0.9))

common_pars <- c("theta_a", "theta_b", "theta",
                 #'spec', "sens",
                 "ADHD_sample", "ADHD_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

```

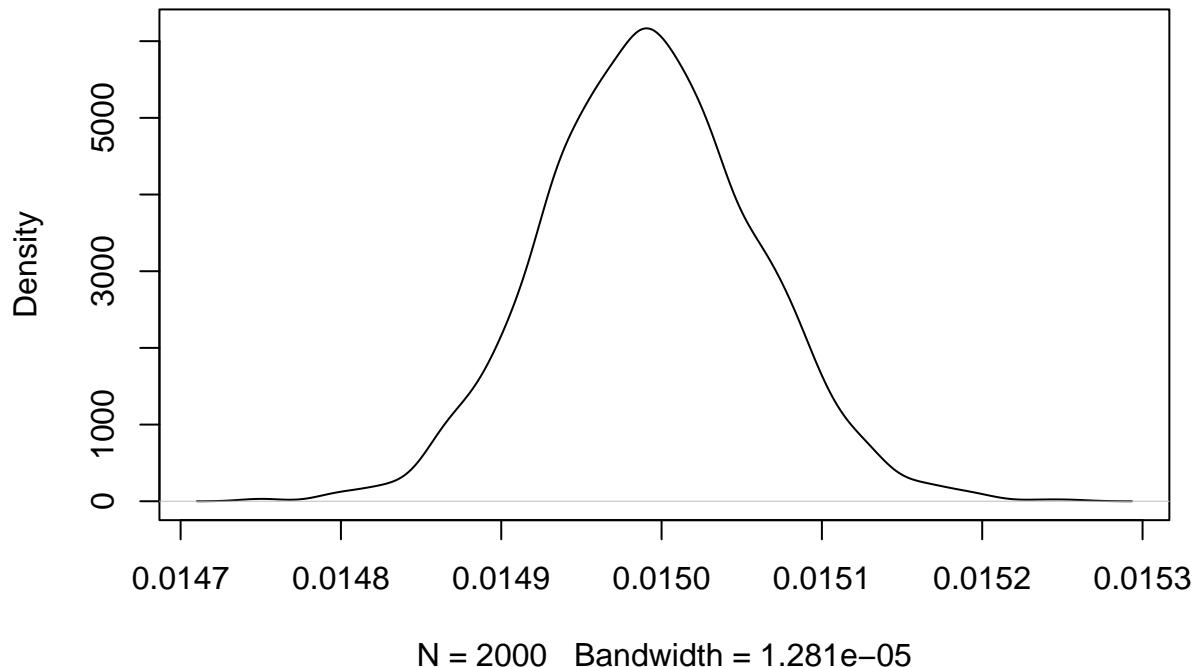


```
summary(as_draws(common_sam)) # mean posterior theta is 0.00477
```

```
## # A tibble: 5 x 10
##   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 ADHD_p~ 4.58e+4 4.58e+4 2.92e+2 2.89e+2 4.53e+4 4.63e+4 0.999  1649.  1783.
## 2 ADHD_s~ 4.58e+4 4.58e+4 0         0         4.58e+4 4.58e+4 NA      NA      NA
## 3 theta    1.50e-2 1.50e-2 6.64e-5 6.51e-5 1.49e-2 1.51e-2 1.00   1423.  1329.
## 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: variable, 2: ess_bulk, 3: ess_tail
```

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

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



```
#plot(density(extract_variable(common_sam, "theta")), xlim = c(0.004,0.0055))
# Very very narrow posterior distribution centered approx at sample prevalence of 0.00476.
# Not that surprising given uniform prior was used.

# Informative prior made no difference to posterior distribution
```

Random effects analysis

95% CIs for aggregate data visualisations using:

<https://openstax.org/books/introductory-business-statistics/pages/8-3-a-confidence-interval-for-a-population-proportion>

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 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))

ADHD_prev_f <- chile_bayes_ADHD %>%
  filter(sex == 2) %>%
  get_grouped_prev(stdpop = chile_stdpop_f, grouping_vars = c("age_june30", "sex", "ADHD"))

## `summarise()` has grouped output by 'age_june30', 'sex'. You can override using
## the ` `.groups` argument.
ADHD_prev_adj_f <- get_adjusted_prev(ADHD_prev_f, grouping_vars = c()) %>% mutate(sex_desc = "Female")

ADHD_prev_m <- chile_bayes_ADHD %>%
  filter(sex == 1) %>%
  get_grouped_prev(stdpop = chile_stdpop_m, grouping_vars = c("age_june30", "sex", "ADHD"))

## `summarise()` has grouped output by 'age_june30', 'sex'. You can override using
## the ` `.groups` argument.
ADHD_prev_adj_m <- get_adjusted_prev(ADHD_prev_m, grouping_vars = c()) %>% mutate(sex_desc = "Male")

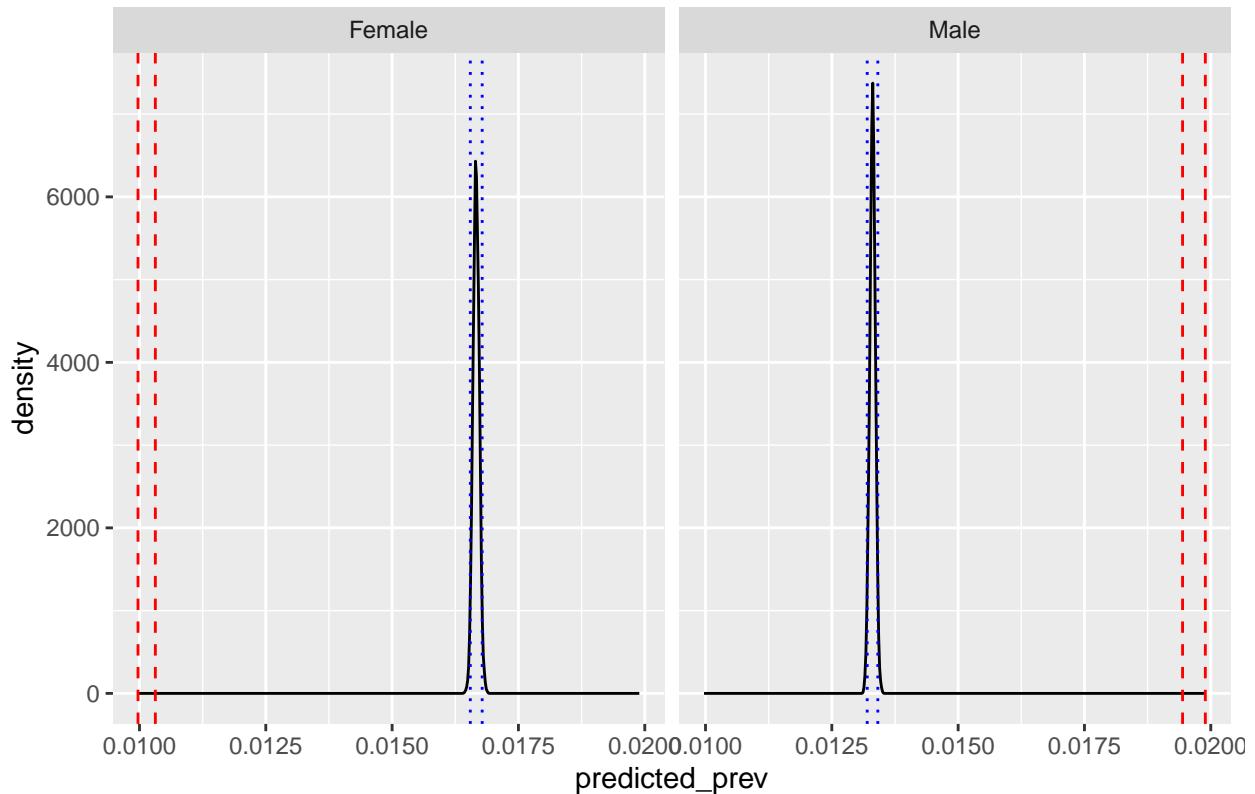
ADHD_prev_sex_adj <- rbind(ADHD_prev_adj_m, ADHD_prev_adj_f)
# have to put m first because 1 comes before 2 and otherwise will mess up naming in do_jags_rand_model

ADHD_prev_sex_post <- do_jags_rand_model(x = ADHD_prev_sex_adj,
                                           feat = "sex_desc",
                                           model = rand_model,
                                           theta_mu = theta_mu_prior,
                                           theta_sigma = theta_sigma_prior,
                                           pars = pars,
                                           convergence_checks = FALSE) %>%
  rename("sex_desc" = "Feat_names")

plot_post_density(ADHD_prev_sex_post, ADHD_prev_sex_adj, feat = "sex_desc", theta_mu = theta_mu_prior,
                  ...
## 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 = 7.25e-05



Pretty clear difference across sexes, as expected

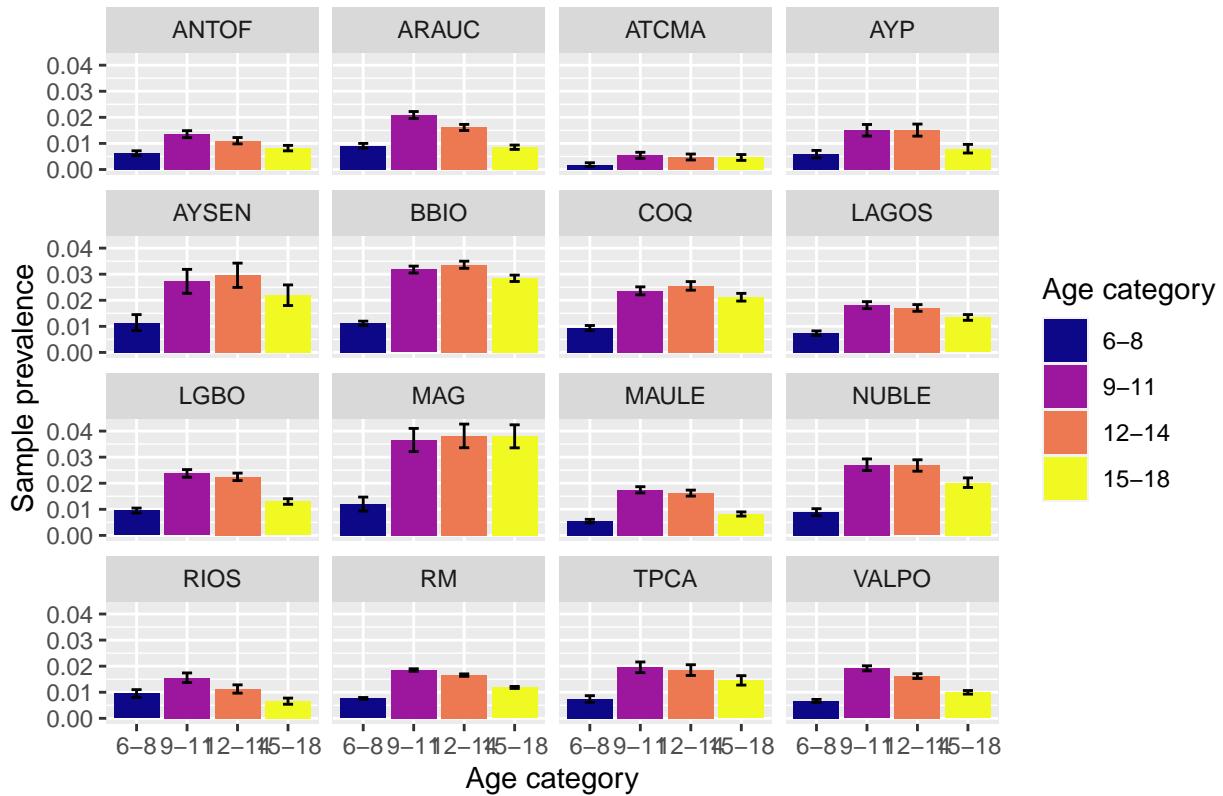
Random effect on region

```
ADHD_prev_region.agecat <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_region_name_abr", "age_cat_name"))

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

ggplot(data = ADHD_prev_region.agecat) +
  geom_col(aes(x = age_cat_name, y = sample_prevalence, group = age_cat_name, fill = age_cat_name), position = "dodge")
  geom_errorbar(aes(x = age_cat_name, ymin = ci_lower, ymax = ci_upper, group = age_cat_name), width = 0.5)
  scale_fill_viridis_d(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~school_region_name_abr) +
  labs(title = "ADHD prevalence by school region",
       x = "Age category",
       y = "Sample prevalence",
       fill = "Age category")
```

ADHD prevalence by school region

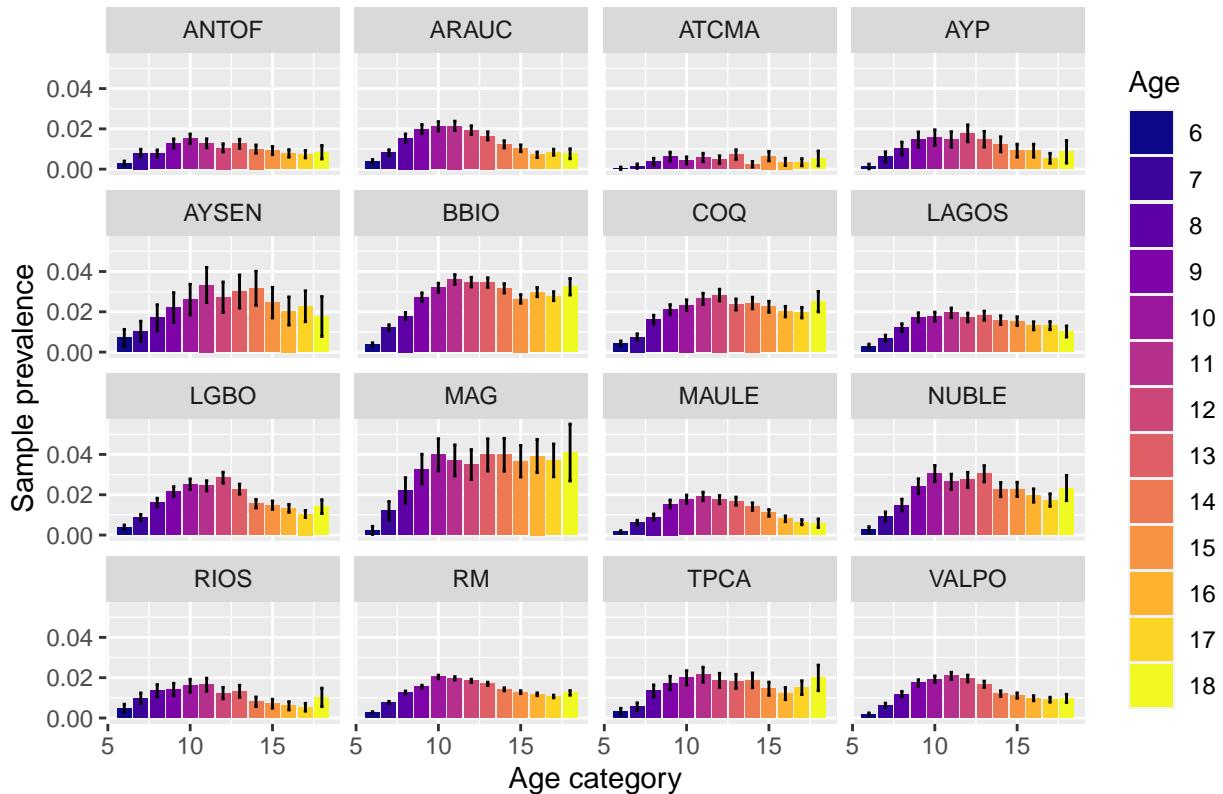


```
ADHD_prev_region.age <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_region_name_abr", "age_june30"))

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

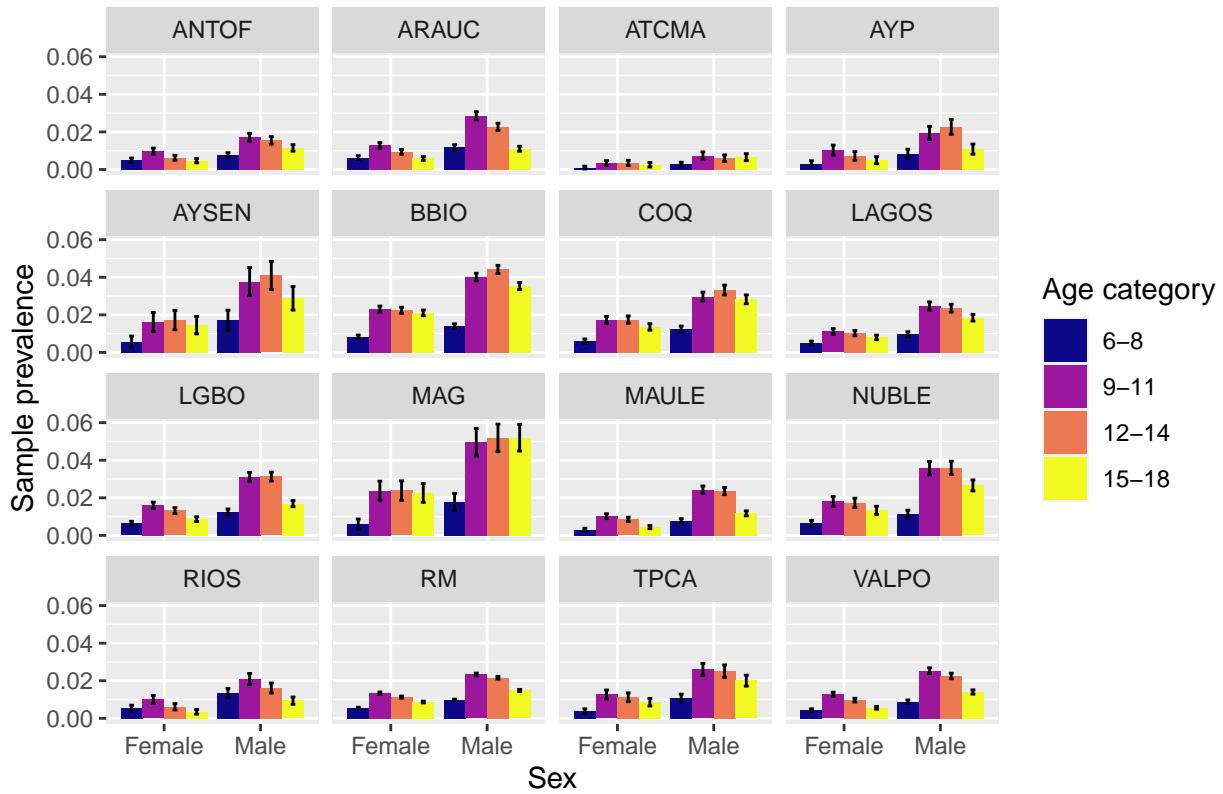
ggplot(data = ADHD_prev_region.age) +
  geom_col(aes(x = age_june30, y = sample_prevalence, group = age_june30, fill = as.factor(age_june30)))
  geom_errorbar(aes(x = age_june30, ymin = ci_lower, ymax = ci_upper, group = age_june30), width = 0.2,
    scale_fill_viridis_d(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  #xlim(c(5.5, 18.5)) +
  facet_wrap(~school_region_name_abr) +
  labs(title = "ADHD prevalence by school region",
       x = "Age category",
       y = "Sample prevalence",
       fill = "Age")
```

ADHD prevalence by school region



```
ADHD_prev_region.agecat.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_region_name_abr",
## `summarise()` has grouped output by 'school_region_name_abr', 'age_cat_name',
## 'sex_desc'. You can override using the `.groups` argument.
ggplot(data = ADHD_prev_region.agecat.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_cat_name, fill = age_cat_name), position =
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_cat_name), width = 0.2,
  scale_fill_viridis_d(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~school_region_name_abr) +
  labs(title = "ADHD prevalence by sex",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age category")
```

ADHD prevalence by sex



```

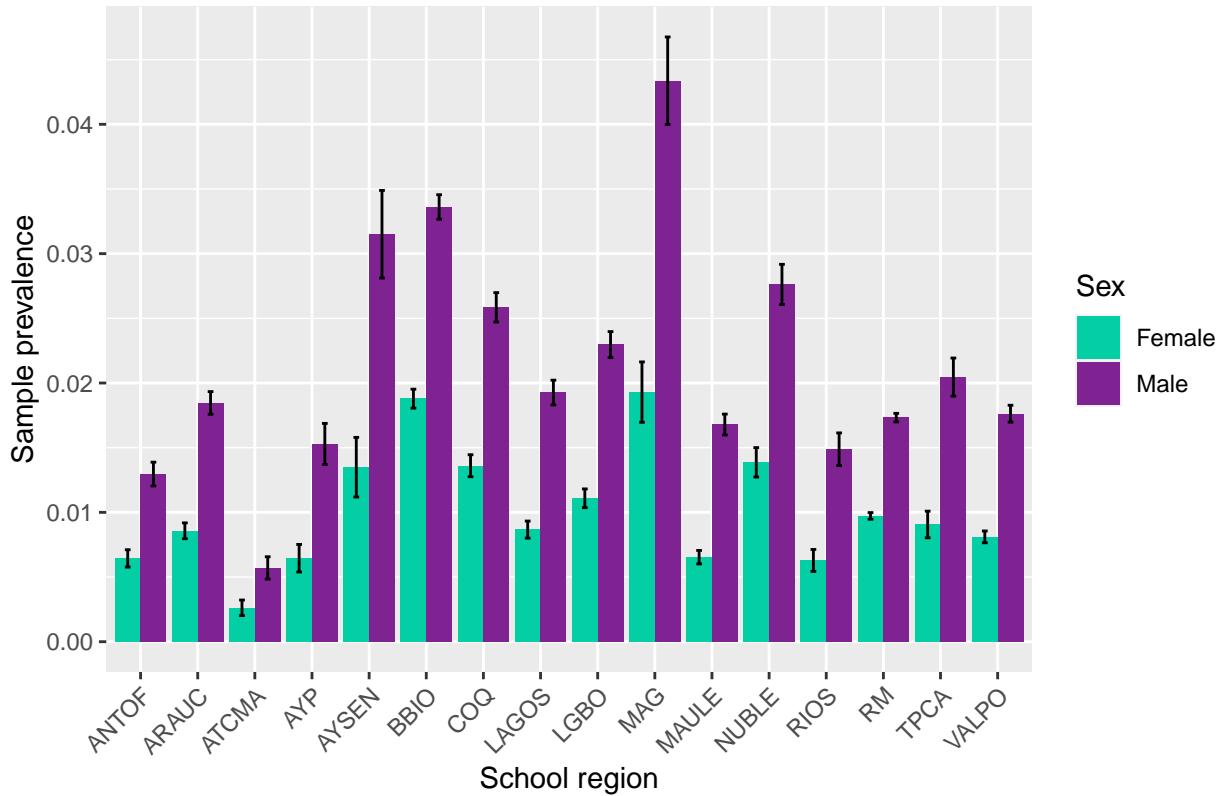
ADHD_prev_region.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_region_name_abr", "sex_desc"))

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

ggplot(data = ADHD_prev_region.sex) +
  geom_col(aes(x = school_region_name_abr, y = sample_prevalence, group = sex_desc, fill = as.factor(sex_desc)))
  # 1 is male, 2 is female
  geom_errorbar(aes(x = school_region_name_abr, ymin = ci_lower, ymax = ci_upper, group = sex_desc), width = 0.2)
  #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 = "ADHD prevalence",
       x = "School region",
       y = "Sample prevalence",
       fill = "Sex")

```

ADHD prevalence



```

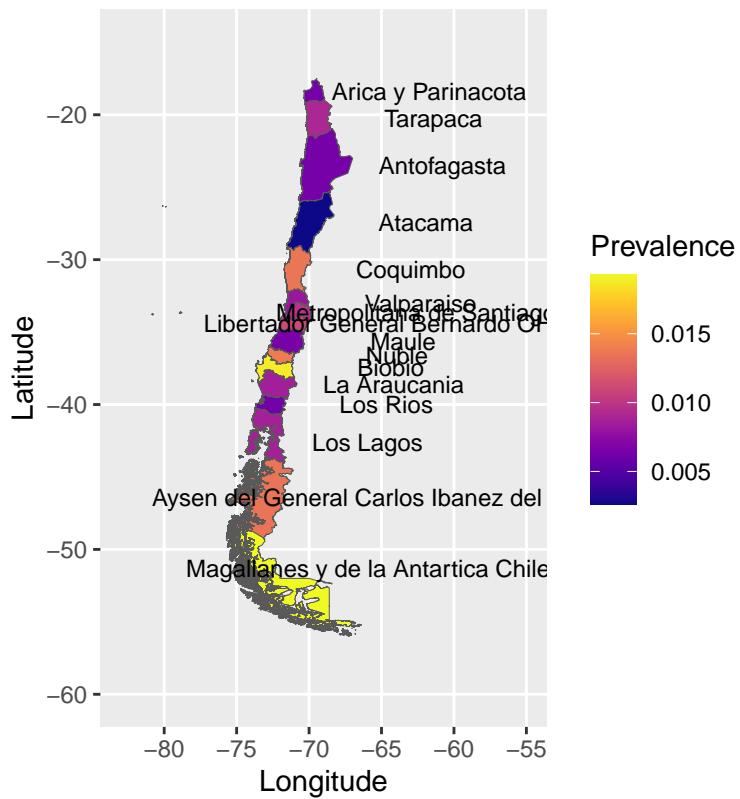
ADHD_prev_region.geom.f <- ADHD_prev_region.sex %>%
  filter(sex_desc == "Female") %>%
  left_join(regions_lookup, by = "school_region_name_abr") %>%
  left_join(demog_geom, by = c("nombre_region", "codigo_region"))

ggplot(ADHD_prev_region.geom.f) +
  geom_sf(mapping = aes(geometry = geometry, fill = sample_prevalence), linewidth = 0.01) +
  geom_sf_text(mapping = aes(geometry = geometry, label = nombre_region), nudge_x = 8, size = 3) +
  scale_fill_viridis(option = "plasma", name = "Prevalence") + #, limits = c(0, 0.025)) +
  labs(title = "ADHD prevalence (females)") +
  xlab("Longitude") +
  ylab("Latitude") +
  coord_sf(xlim = c(-83, -55), ylim = c(-60, -15))

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data

```

ADHD prevalence (females)



```

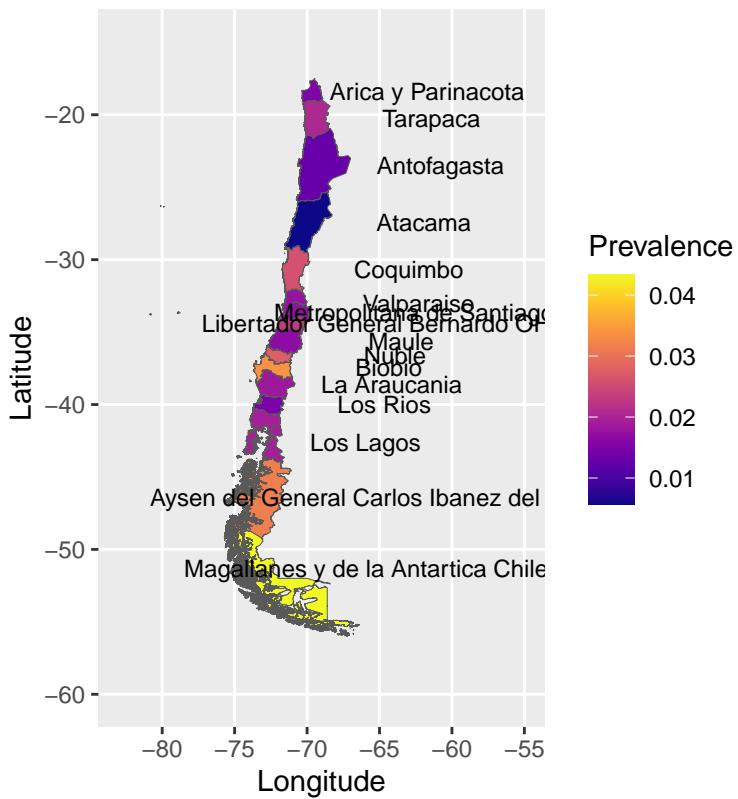
ADHD_prev_region.geom.m <- ADHD_prev_region.sex %>%
  filter(sex_desc == "Male") %>%
  left_join(regions_lookup, by = "school_region_name_abr") %>%
  left_join(demog_geom, by = c("nombre_region", "codigo_region"))

ggplot(ADHD_prev_region.geom.m) +
  geom_sf(mapping = aes(geometry = geometry, fill = sample_prevalence), linewidth = 0.01) +
  geom_sf_text(mapping = aes(geometry = geometry, label = nombre_region), nudge_x = 8, size = 3) +
  scale_fill_viridis(option = "plasma", name = "Prevalence") + #, limits = c(0, 0.025)) +
  labs(title = "ADHD prevalence (males)") +
  xlab("Longitude") +
  ylab("Latitude") +
  coord_sf(xlim = c(-83, -55), ylim = c(-60, -15))

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data

```

ADHD prevalence (males)



```
# Need to add standardised scale to both f and m
```

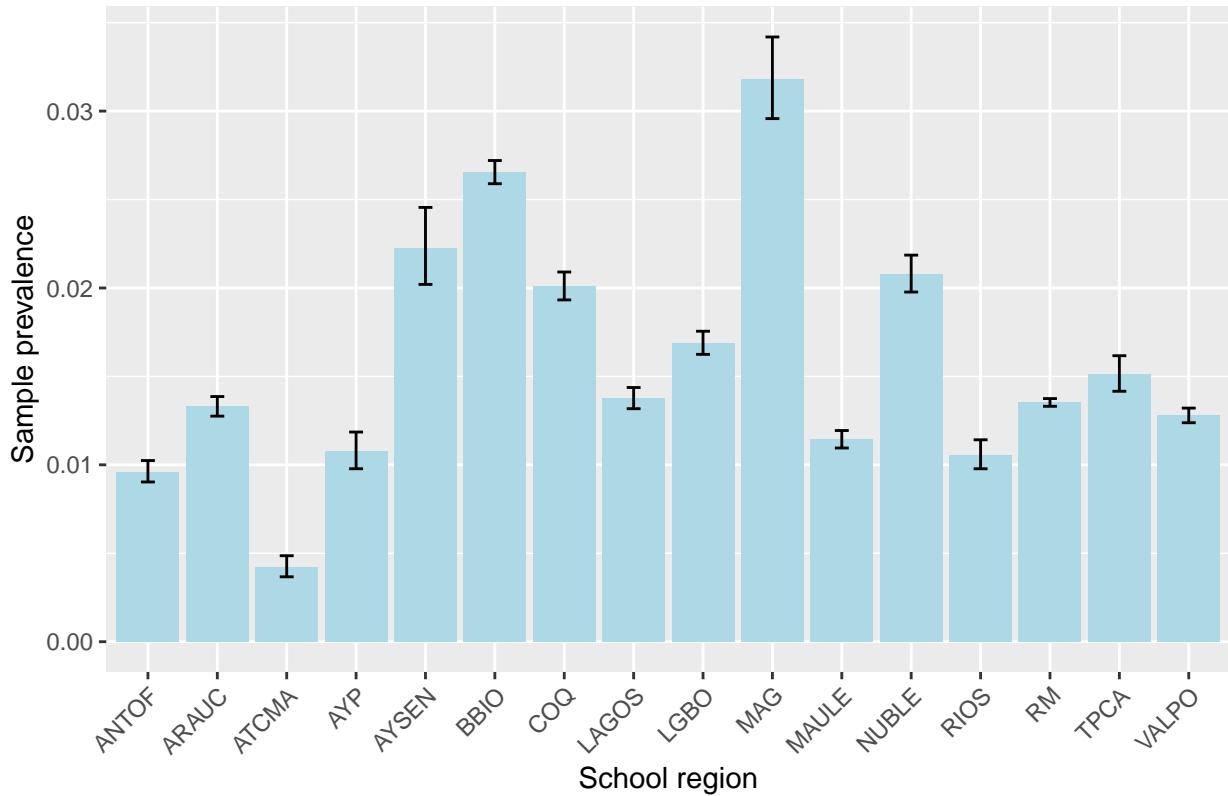
```
ADHD_prev_region <- get_grouped_prev(x = chile_bayes_ADHD, stdpop = chile_stdpop,
                                         grouping_vars = c("school_region_name_abr", "age_june30", "age_cat_name",
                                         "sex", "sex_desc"))

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

ADHD_prev_region_adj <- get_adjusted_prev(ADHD_prev_region, grouping_vars = "school_region_name_abr")

ggplot(data = ADHD_prev_region_adj) +
  geom_col(aes(x = school_region_name_abr, y = adjusted_rate), fill = "lightblue", position = "dodge") +
  geom_errorbar(aes(x = school_region_name_abr, ymin = ci_lower, ymax = ci_upper), width = 0.2) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Age- and sex-adjusted ADHD prevalence",
       x = "School region",
       y = "Sample prevalence")
```

Age- and sex-adjusted ADHD prevalence



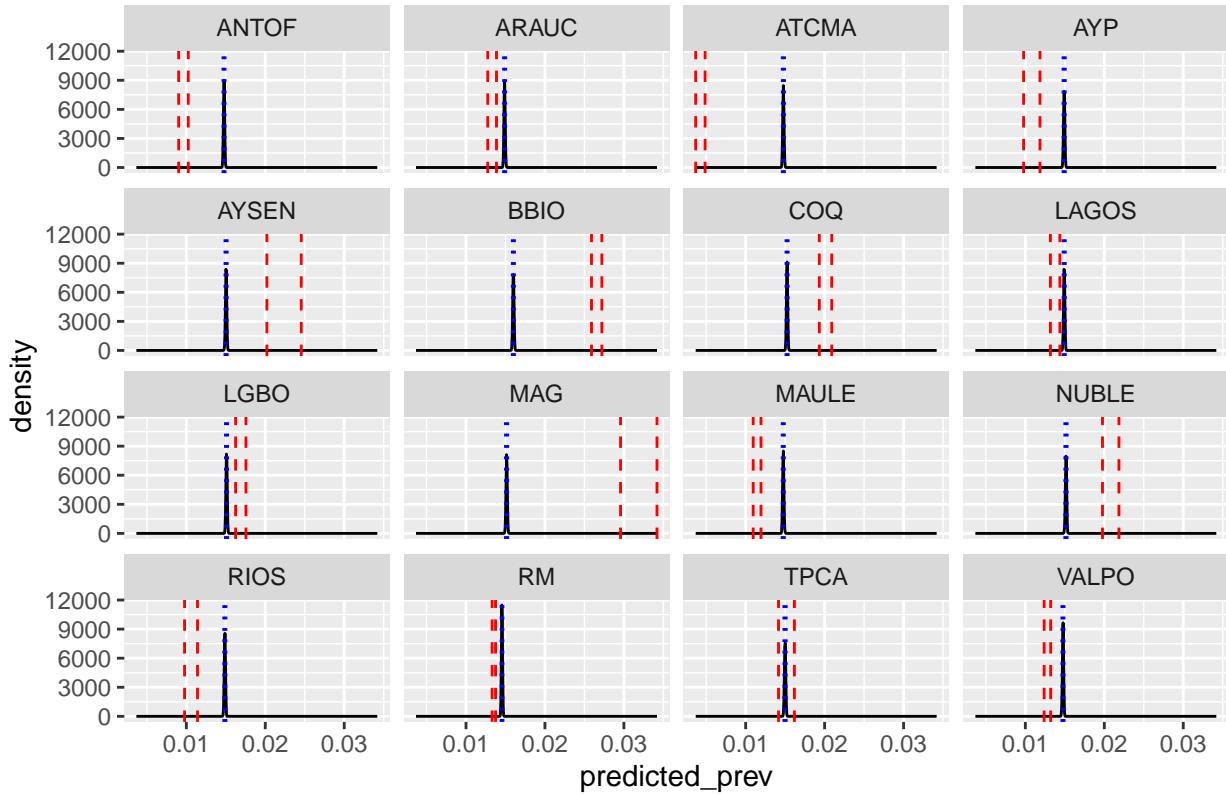
```

ADHD_prev_region_post <- do_jags_rand_model(x = ADHD_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(ADHD_prev_region_post, ADHD_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 = 0.015, prior sd = 7.25e-05



Predictions for higher population prevalence - increase prior mean

```

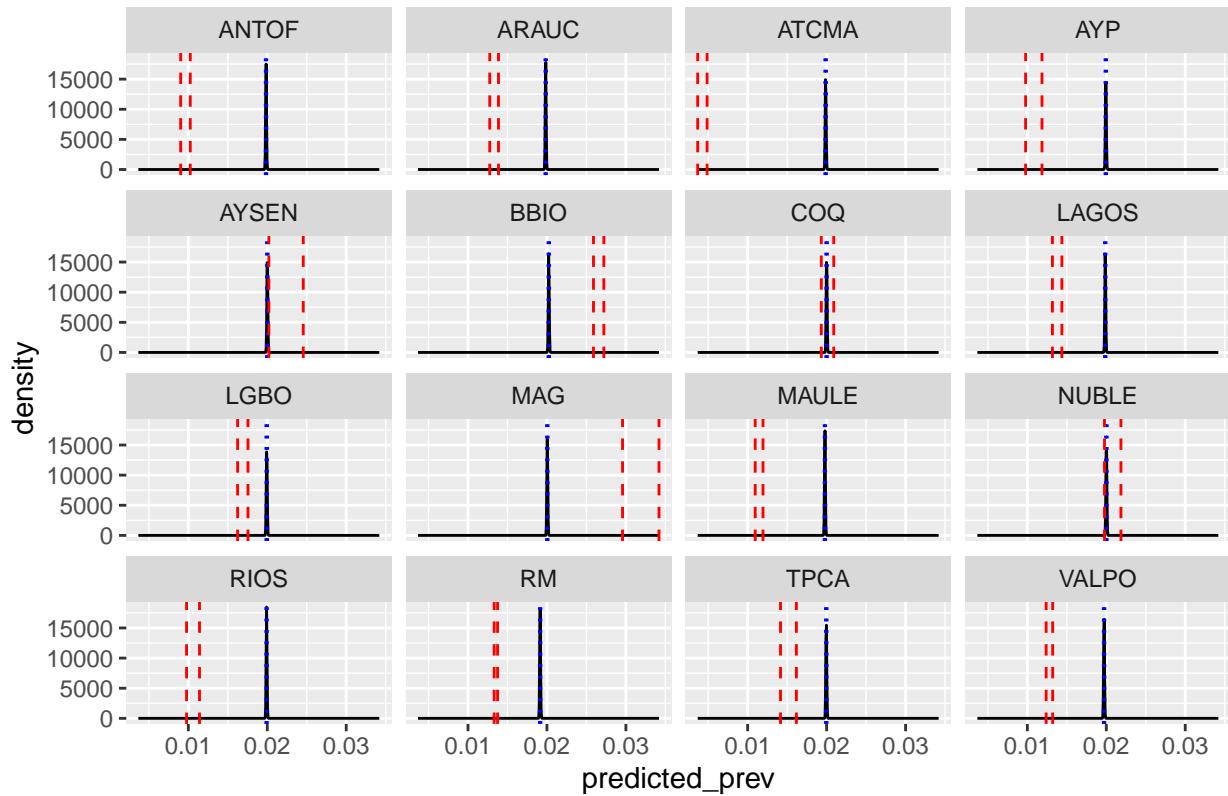
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_region_post <- do_jags_rand_model(x = ADHD_prev_region_adj,
                                                feat = "school_region_name_abr",
                                                model = rand_model,
                                                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(ADHD_prev_region_post,
                    ADHD_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.

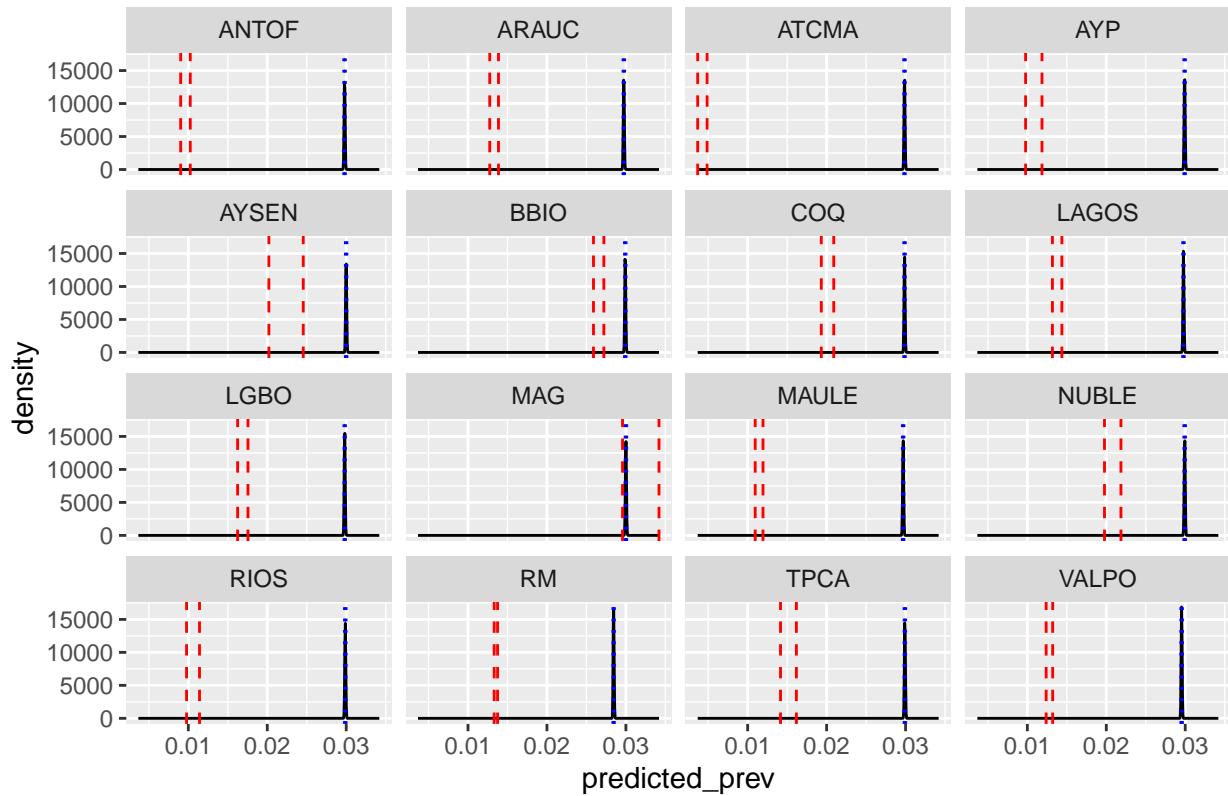
```

Prior mean = 0.02, prior sd = 5.1e-05



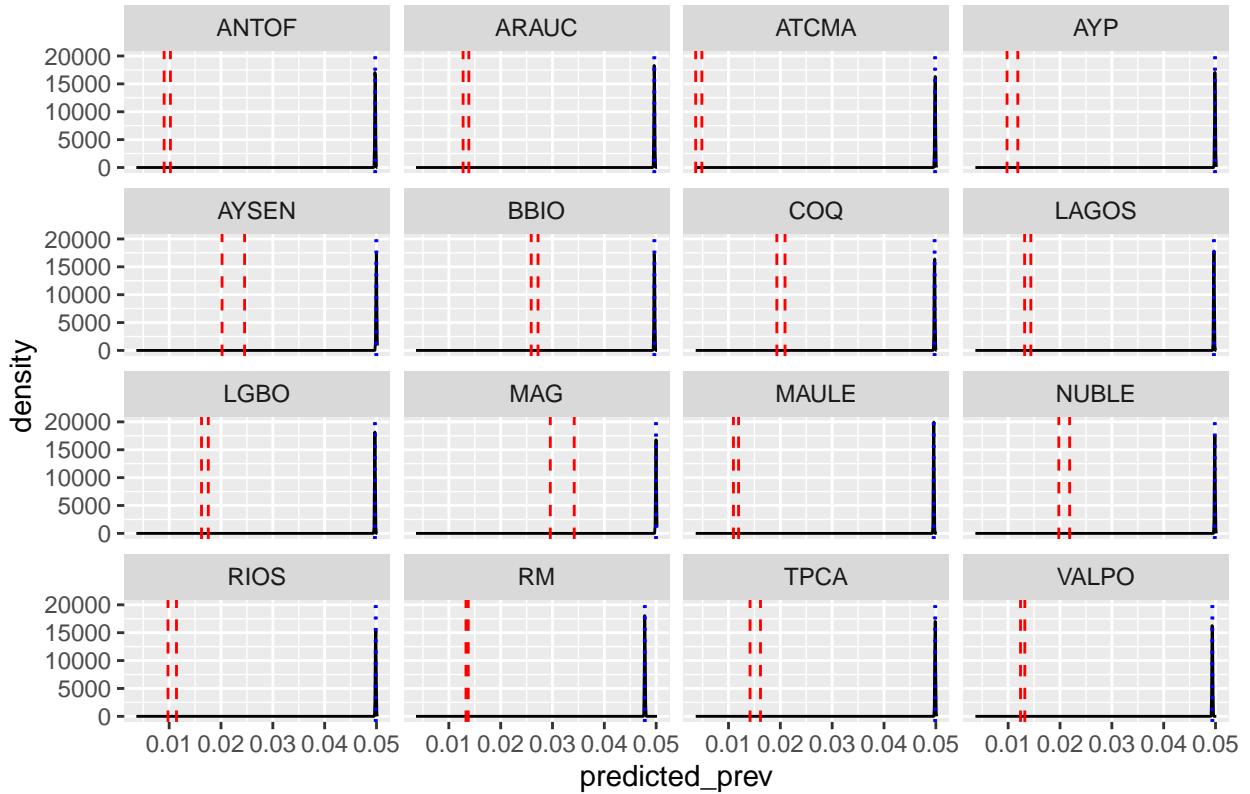
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, 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 = 0.05, prior sd = 5.1e-05



Random effect on region, sexes separate

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

```

theta_mu_prior_f <- ADHD_prev_adj_f$adjusted_rate
theta_sigma_prior_f <- sqrt(ADHD_prev_adj_f$var)

theta_mu_prior_m <- ADHD_prev_adj_m$adjusted_rate
theta_sigma_prior_m <- sqrt(ADHD_prev_adj_m$var)

# Females
ADHD_prev_region_f <- chile_bayes_ADHD %>%
  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.

ADHD_prev_region_adj_f <- get_adjusted_prev(ADHD_prev_region_f, grouping_vars = "school_region_name_abr")

ADHD_prev_region_post_f <- do_jags_rand_model(x = ADHD_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) %>%
  
```

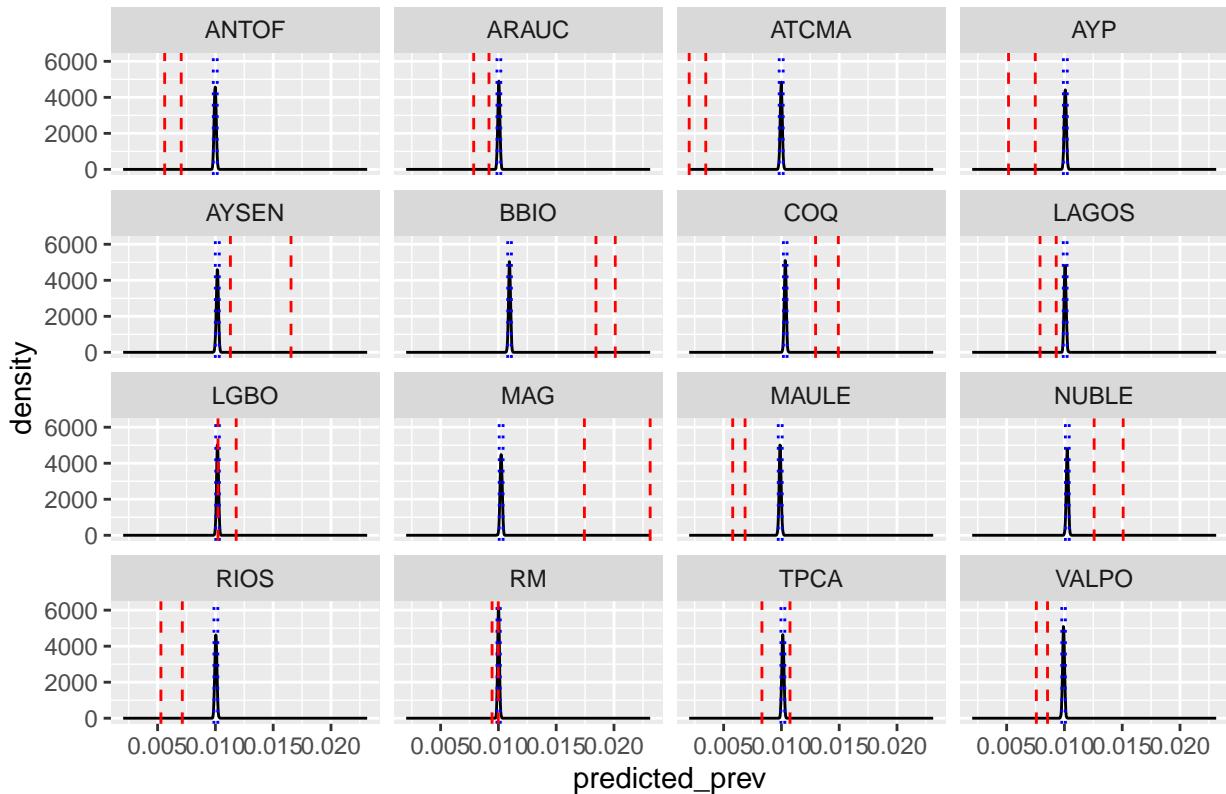
```

rename("school_region_name_abr" = "Feat_names")

plot_post_density(ADHD_prev_region_post_f, ADHD_prev_region_adj_f, feat = "school_region_name_abr", the
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.

```

Prior mean = 0.0101, prior sd = 8.7e-05 (females)



```

# Males
ADHD_prev_region_m <- chile_bayes_ADHD %>%
  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.

ADHD_prev_region_adj_m <- get_adjusted_prev(ADHD_prev_region_m, grouping_vars = "school_region_name_abr"

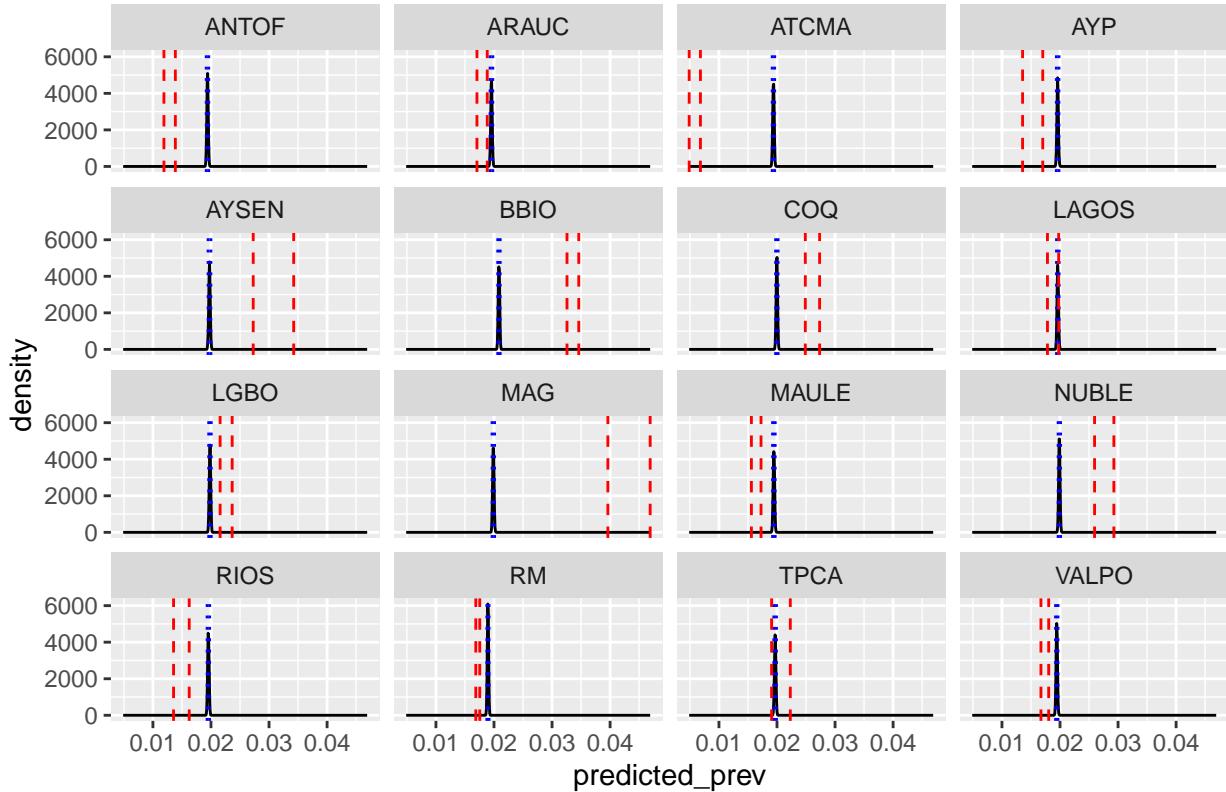
ADHD_prev_region_post_m <- do_jags_rand_model(x = ADHD_prev_region_adj_m,
                                               feat = "school_region_name_abr",
                                               model = rand_model,
                                               theta_mu = theta_mu_prior_m,
   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

plot_post_density(ADHD_prev_region_post_m, ADHD_prev_region_adj_m, feat = "school_region_name_abr", the

```

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

Prior mean = 0.0197, prior sd = 0.000114 (males)



Maybe need different extrapolation priors for females?

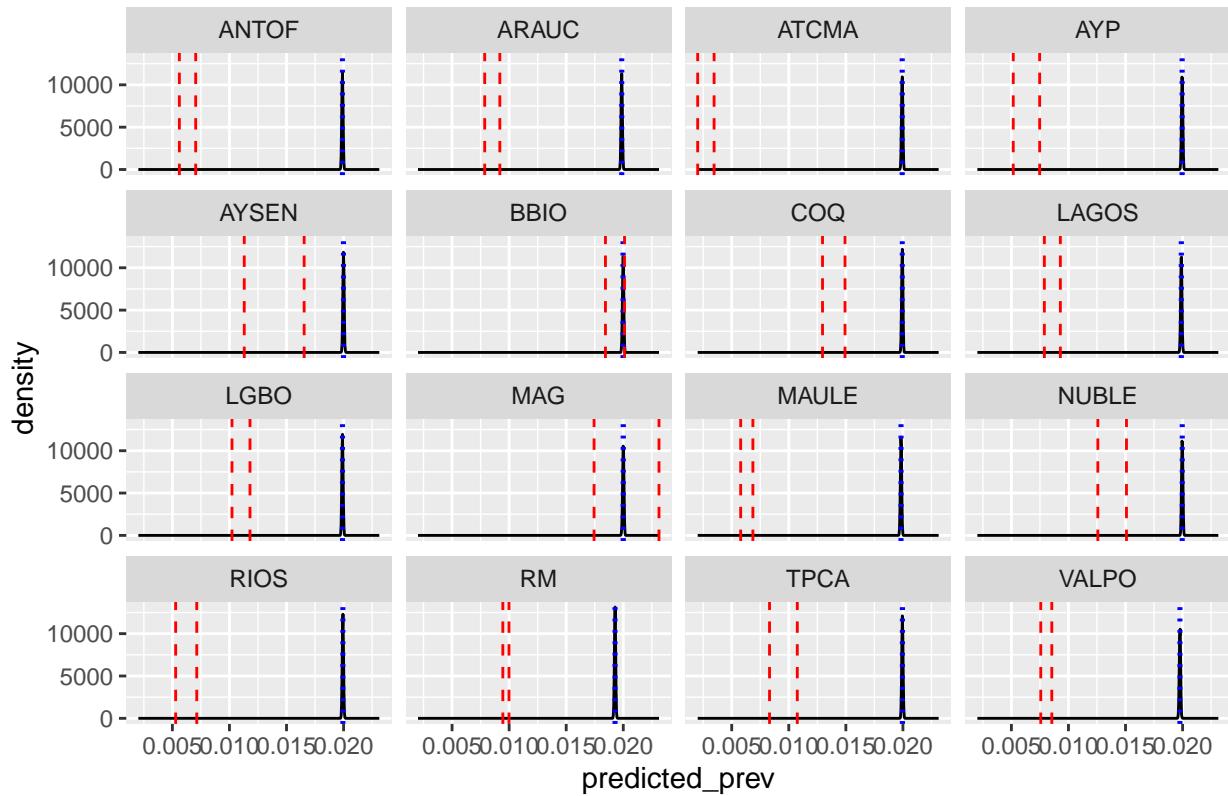
```
# Females
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_region_post_f <- do_jags_rand_model(x = ADHD_prev_region_adj_f,
                                                 feat = "school_region_name_abr",
                                                 model = rand_model,
                                                 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(ADHD_prev_region_post_f,
                    ADHD_prev_region_adj_f,
                    feat = "school_region_name_abr",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j],
                    title_text = " (females)")

}

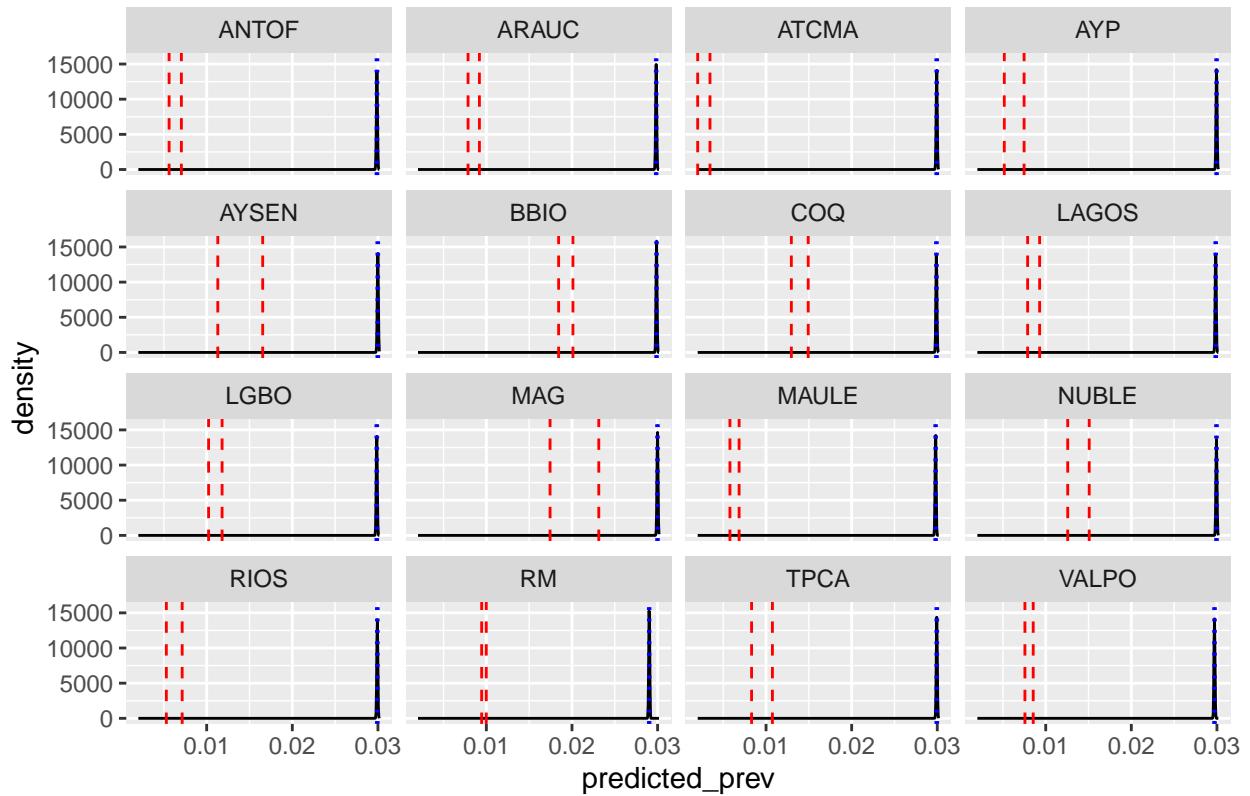
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05 (females)



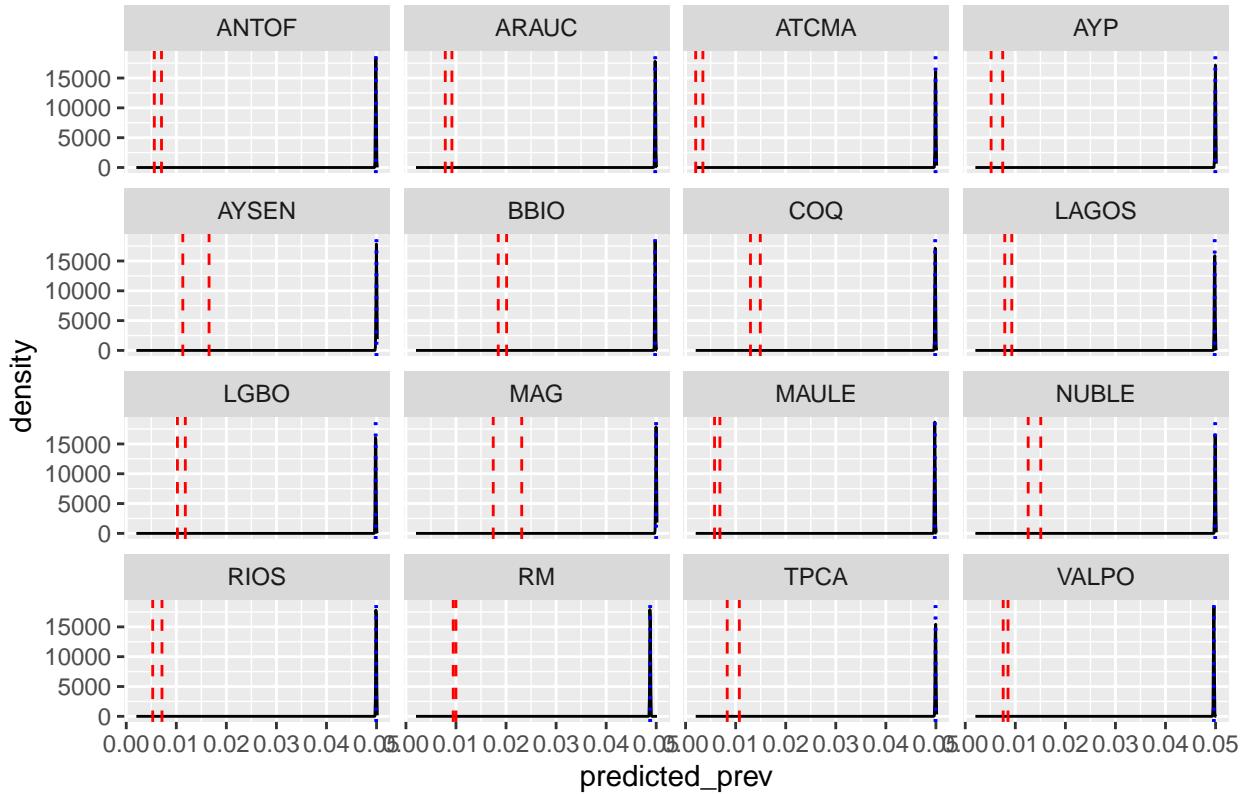
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, prior sd = 5.1e-05 (females)



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

Prior mean = 0.05, prior sd = 5.1e-05 (females)



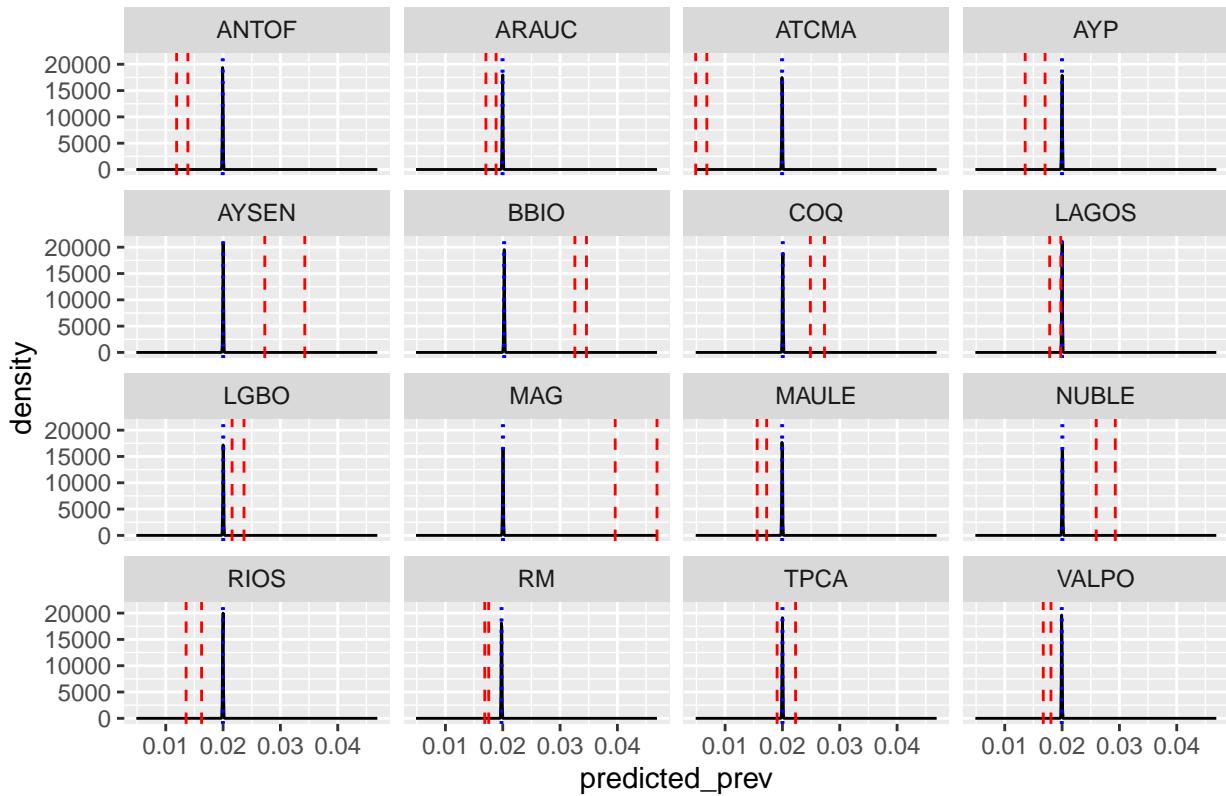
```
# Males
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_region_post_m <- do_jags_rand_model(x = ADHD_prev_region_adj_m,
                                                feat = "school_region_name_abr",
                                                model = rand_model,
                    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

}

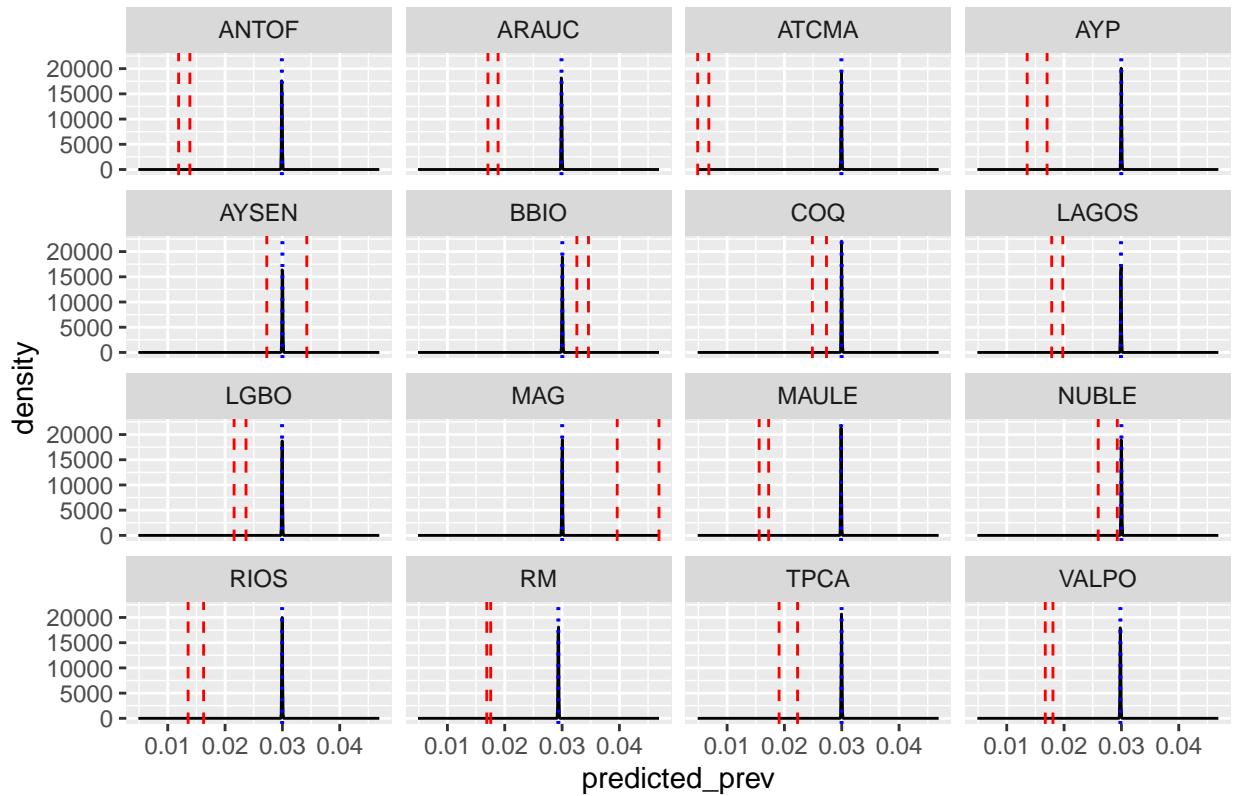
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05 (males)



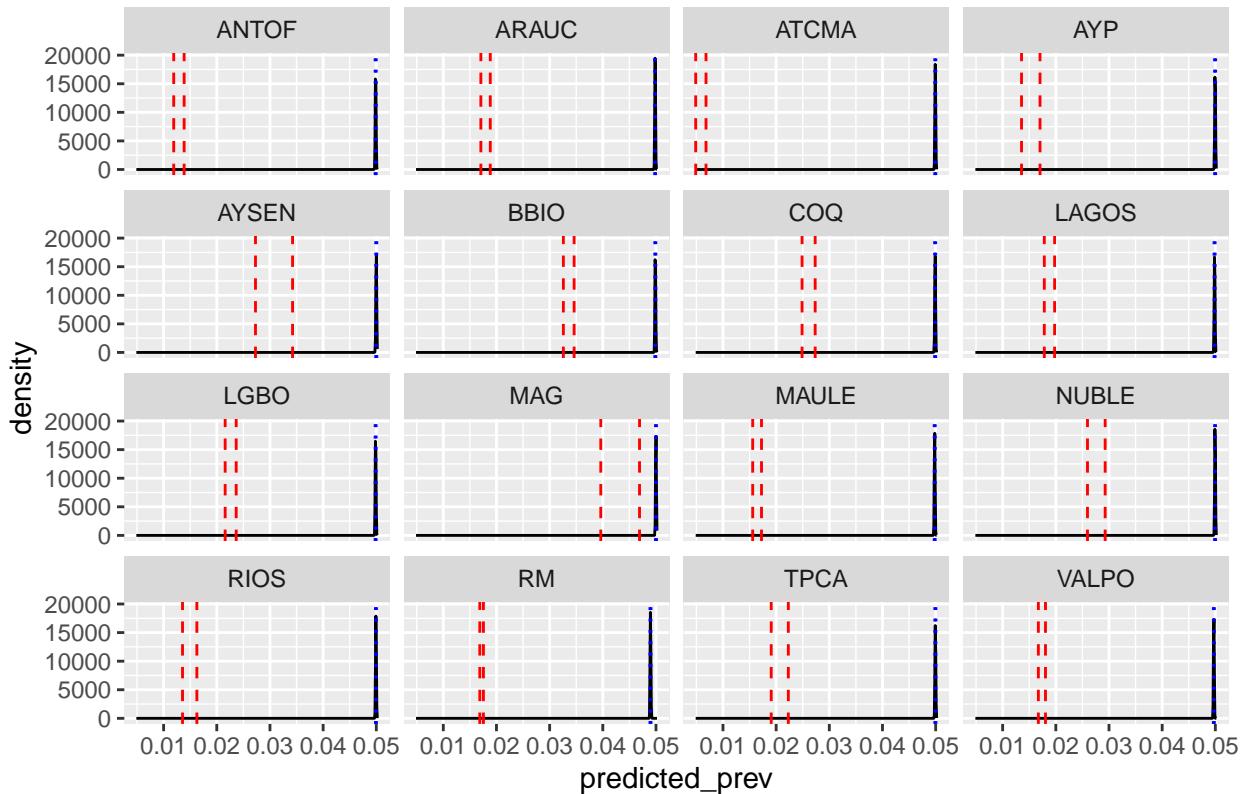
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, prior sd = 5.1e-05 (males)



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

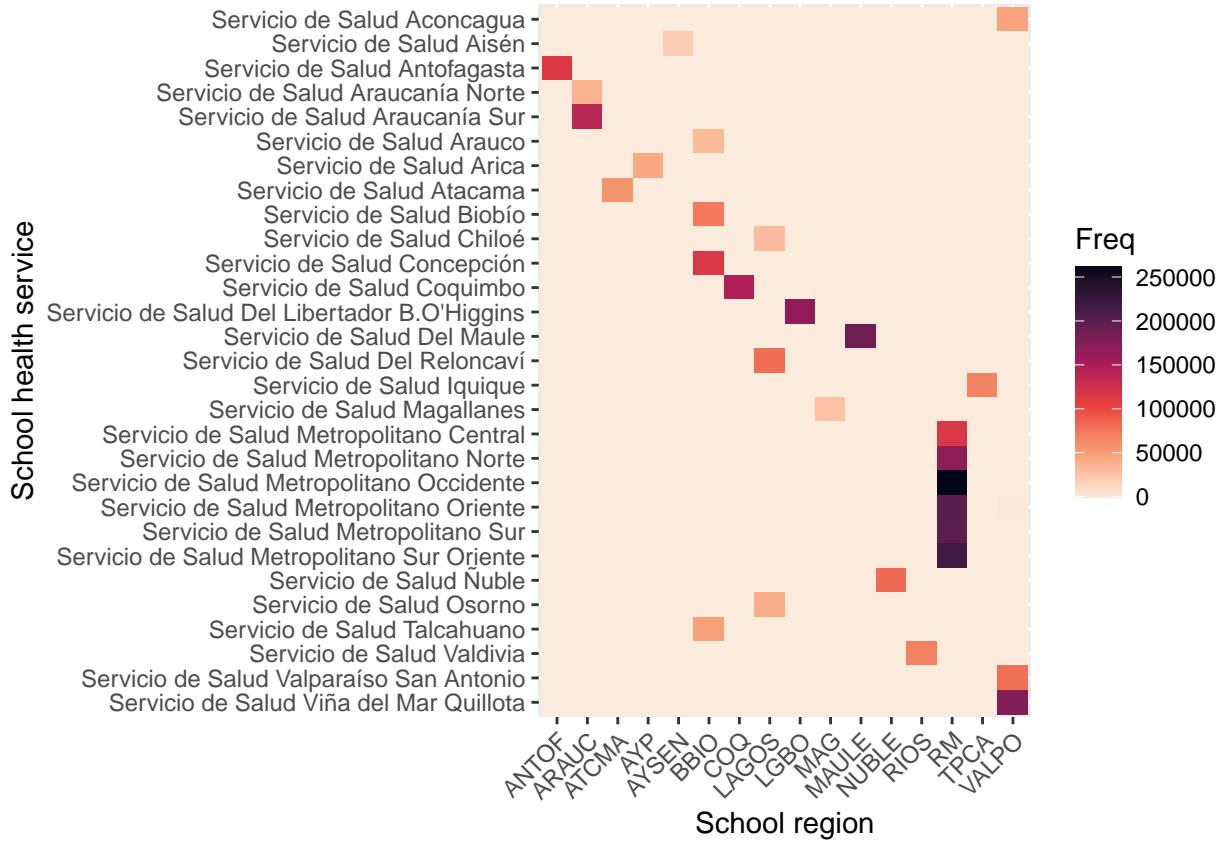
Prior mean = 0.05, prior sd = 5.1e-05 (males)



Random effect on health service

Show which regions each health service operates in

```
map_tab_df <- as.data.frame(table(chile_bayes_ADHD$school_region_name_abr, chile_bayes_ADHD$health_serv)
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")
```



```
#map2_tab_df <- as.data.frame(table(chile_bayes_ADHD$school_commune_name, chile_bayes_ADHD$health_service_name))
#ggplot(map2_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(map2_tab_df$Var2))) +
#  labs(x = "School commune", y = "School health service", fill = "Freq")
```

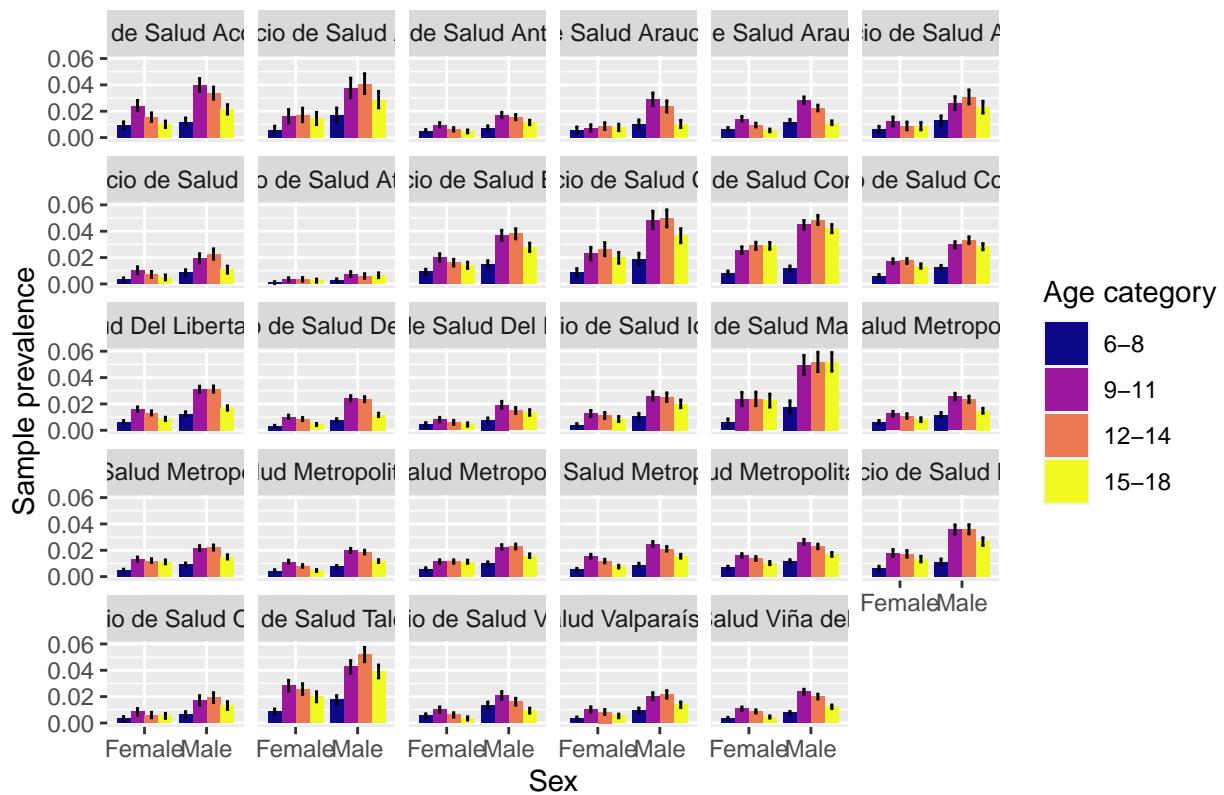
Each health service operates in only one region, some regions (RM, VALPO, LAGOS, BBIO, ARAUC) have multiple health services.

```
ADHD_prev_health.agecat.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("health_service_name", "age_cat_name", "sex_desc"))

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

ggplot(data = ADHD_prev_health.agecat.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_cat_name, fill = age_cat_name), position = "stack") +
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_cat_name), width = 0.2, position = "stack") +
  scale_fill_viridis_d(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~health_service_name) +
  labs(title = "ADHD prevalence by school health service",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age category")
```

ADHD prevalence by school health service

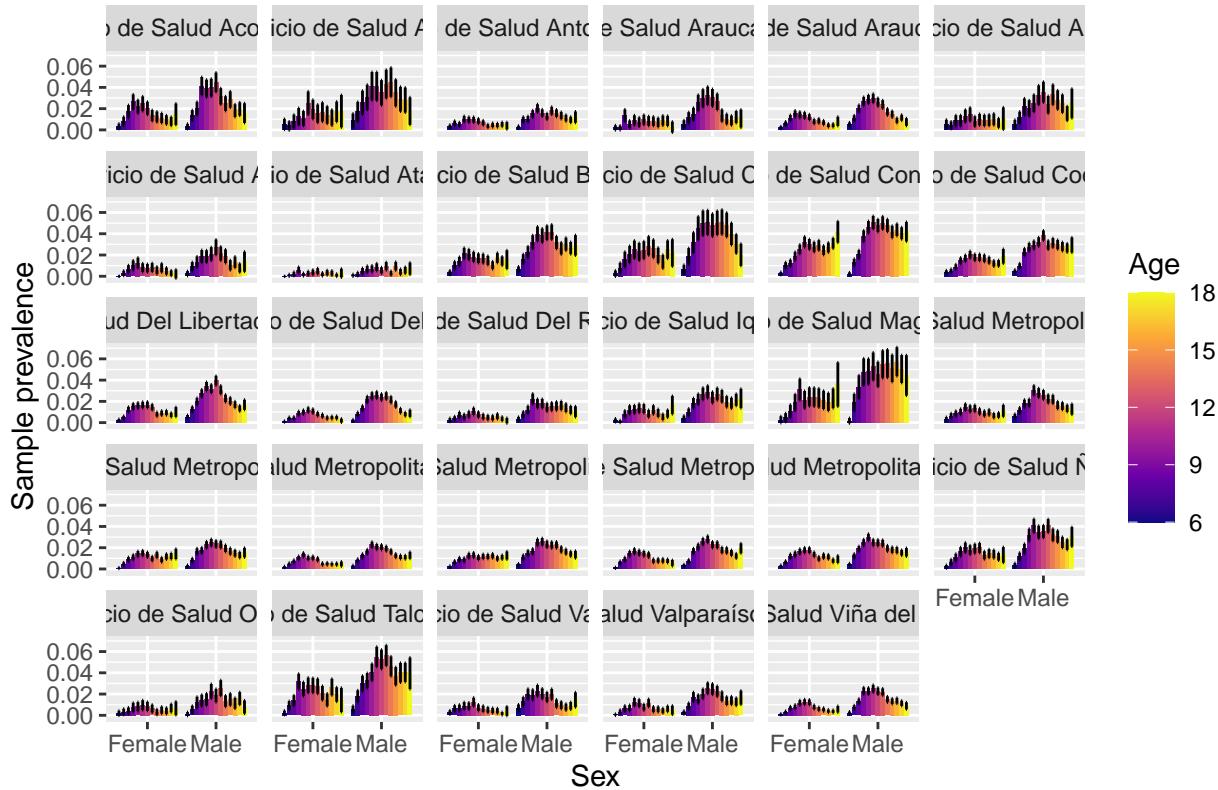


```
ADHD_prev_health.age.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("health_service_name", "age_june30"))

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

ggplot(data = ADHD_prev_health.age.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_june30, fill = age_june30), position = position_dodge())
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_june30), width = 0.2, position = position_dodge())
  scale_fill_viridis_c(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~health_service_name) +
  labs(title = "ADHD prevalence by school health service",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age")
```

ADHD prevalence by school health service



```

ADHD_prev_health <- get_grouped_prev(x = chile_bayes_ADHD, 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.

ADHD_prev_health_adj <- get_adjusted_prev(ADHD_prev_health, grouping_vars = "health_service_name")

# ggplot(data = ADHD_prev_health_adj) +
#   geom_col(aes(x = health_service_name, y = adjusted_rate), fill = "lightblue", position = "dodge") +
#   geom_errorbar(aes(x = health_service_name, ymin = ci_lower, ymax = ci_upper), width = 0.2) +
#   theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
#   labs(title = "Age- and sex-adjusted ADHD prevalence",
#        x = "School health service",
#        y = "Sample prevalence")

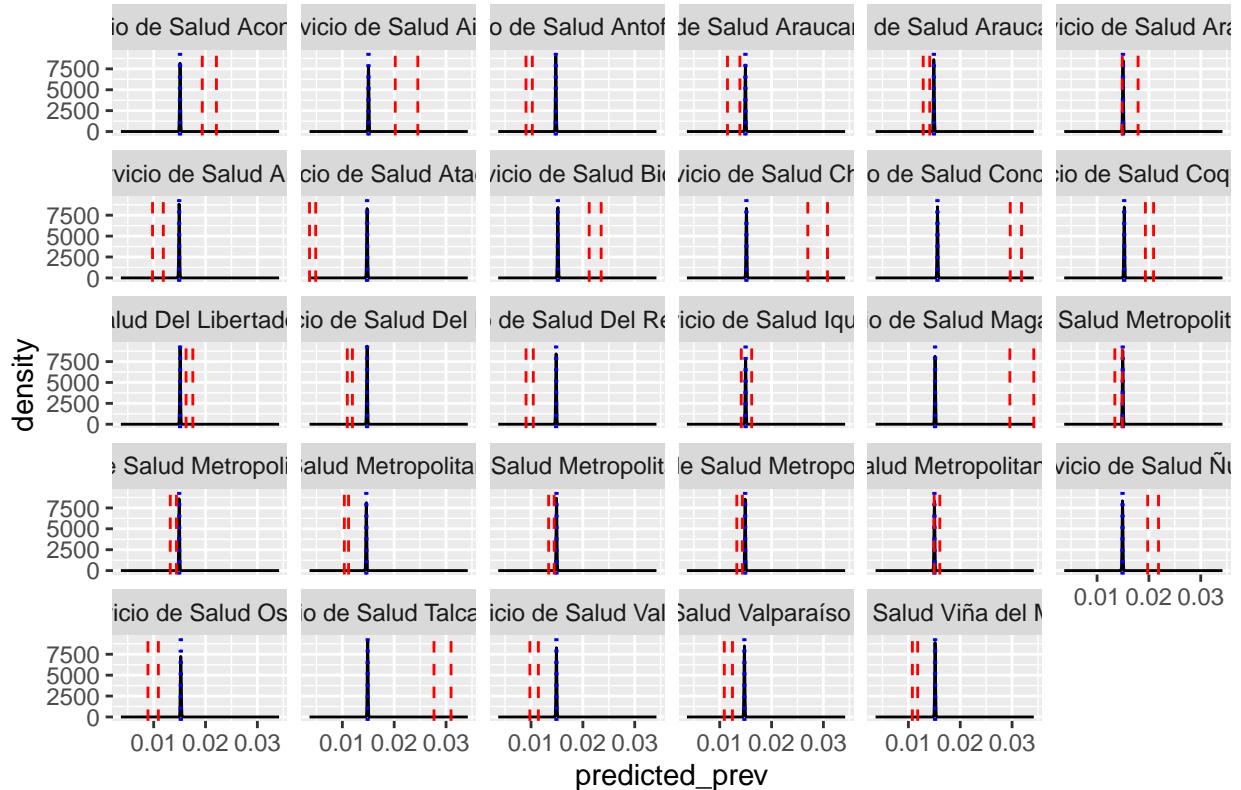
ADHD_prev_health_post <- do_jags_rand_model(x = ADHD_prev_health_adj,
                                              feat = "health_service_name",
                                              model = rand_model,
                                              theta_mu = theta_mu_prior,
               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

plot_post_density(ADHD_prev_health_post, ADHD_prev_health_adj, feat = "health_service_name", theta_mu =

```

```
## 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 = 7.25e-05



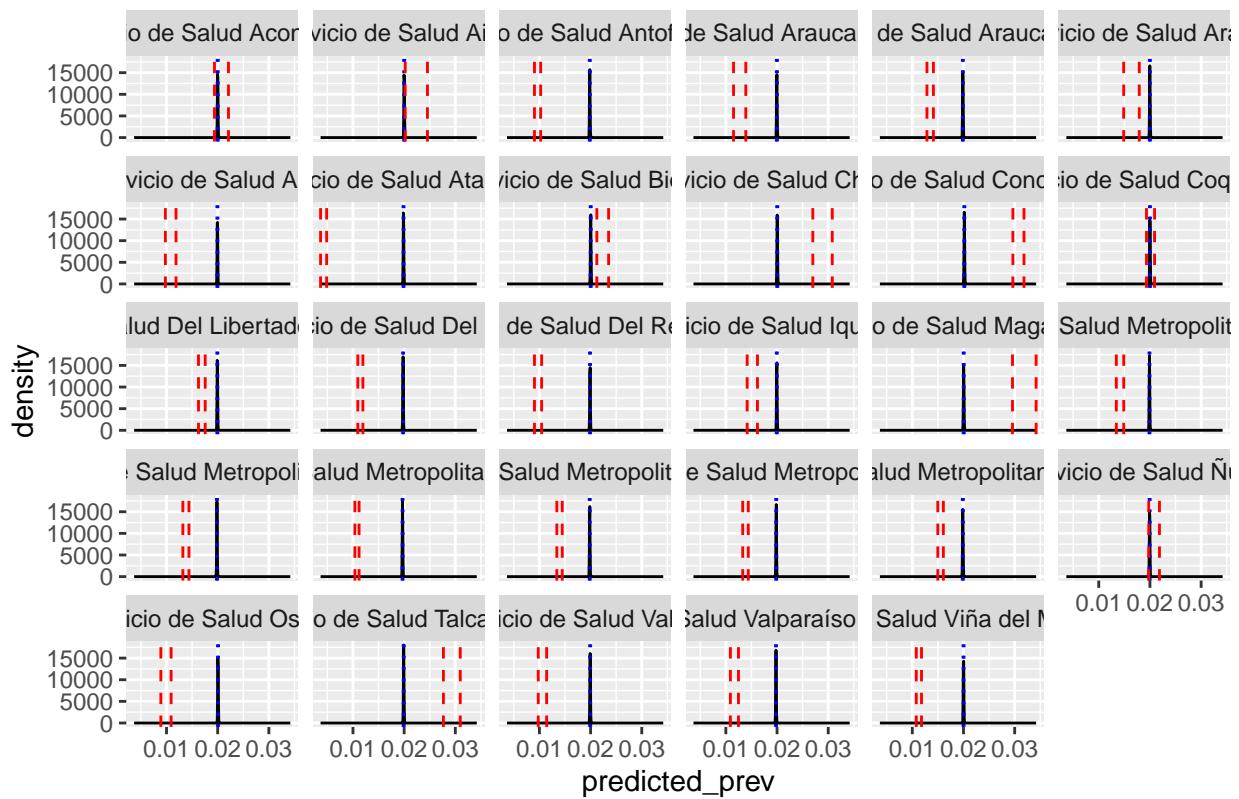
Predictions for higher population prevalence - increase prior mean

```
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_health_post <- do_jags_rand_model(x = ADHD_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(ADHD_prev_health_post,
                    ADHD_prev_health_adj,
                    feat = "health_service_name",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}

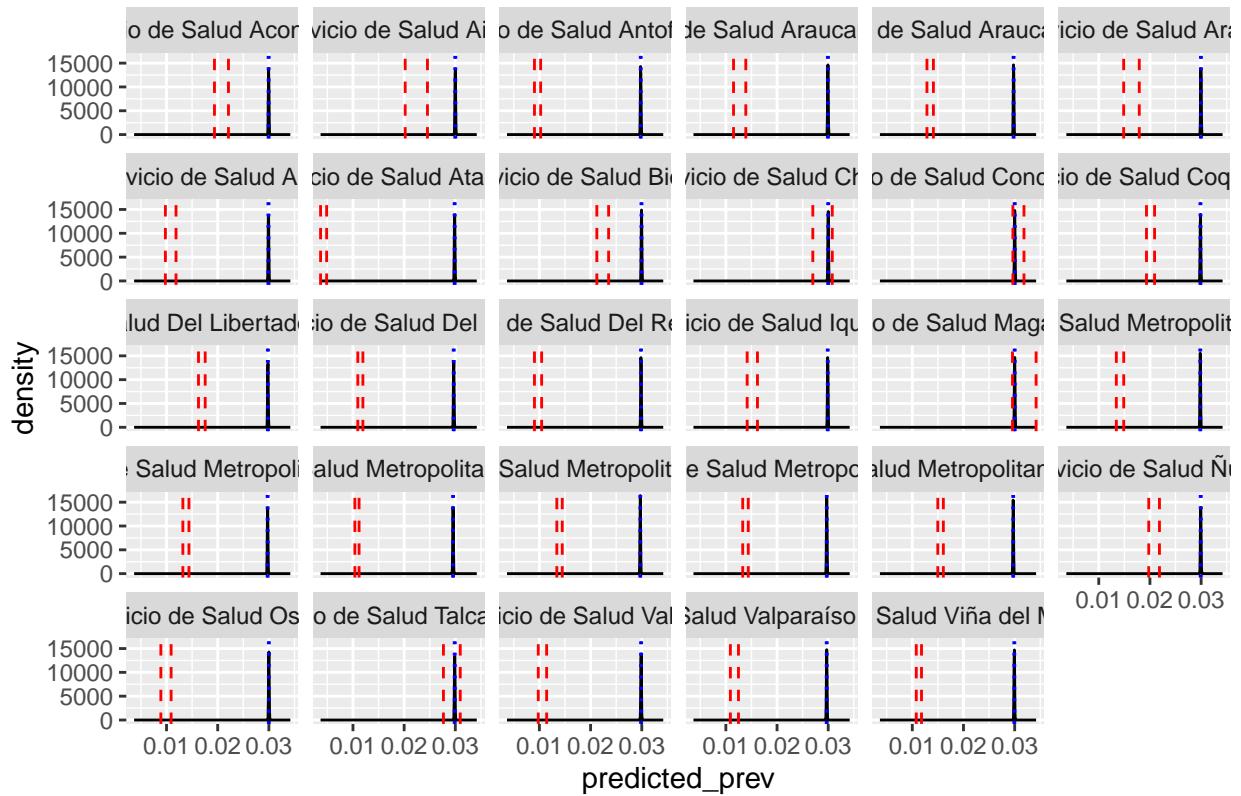
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05



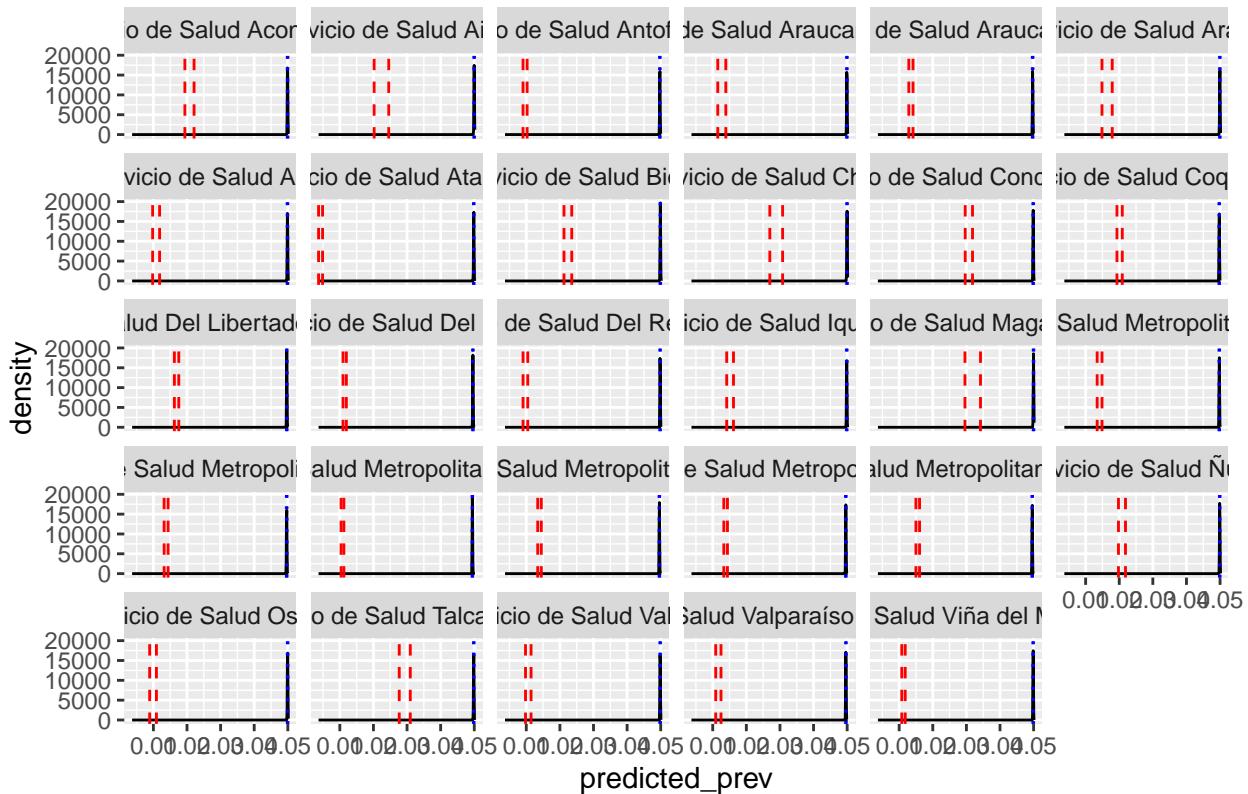
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, 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 = 0.05, prior sd = 5.1e-05



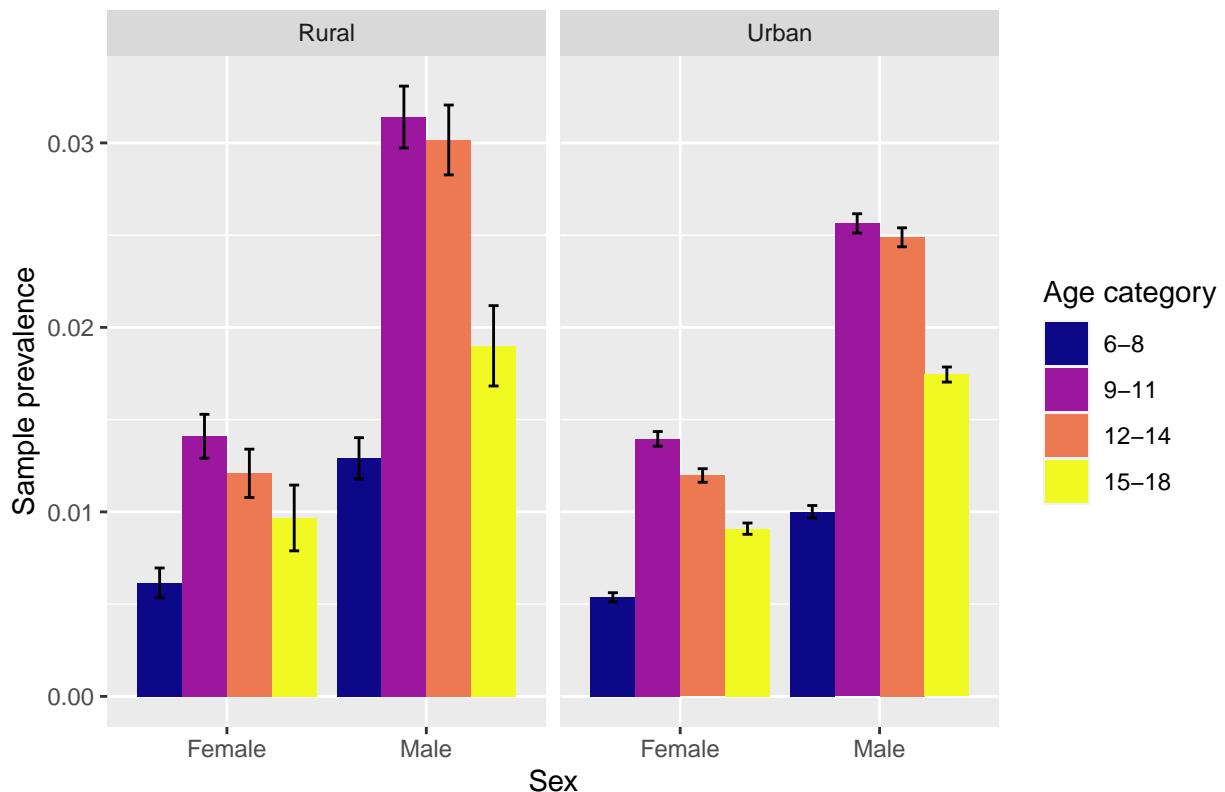
Random effect on rurality

```
ADHD_prev_rural.agecat.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_rurality_code", "age_cat_name"),
  mutate(rurality_desc = ifelse(school_rurality_code == 0, "Urban", "Rural"))

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

ggplot(data = ADHD_prev_rural.agecat.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_cat_name, fill = age_cat_name), position = "dodge") +
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_cat_name), width = 0.2,
    scale_fill_viridis_d(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~rurality_desc) +
  labs(title = "ADHD prevalence by school rurality",
    x = "Sex",
    y = "Sample prevalence",
    fill = "Age category")
```

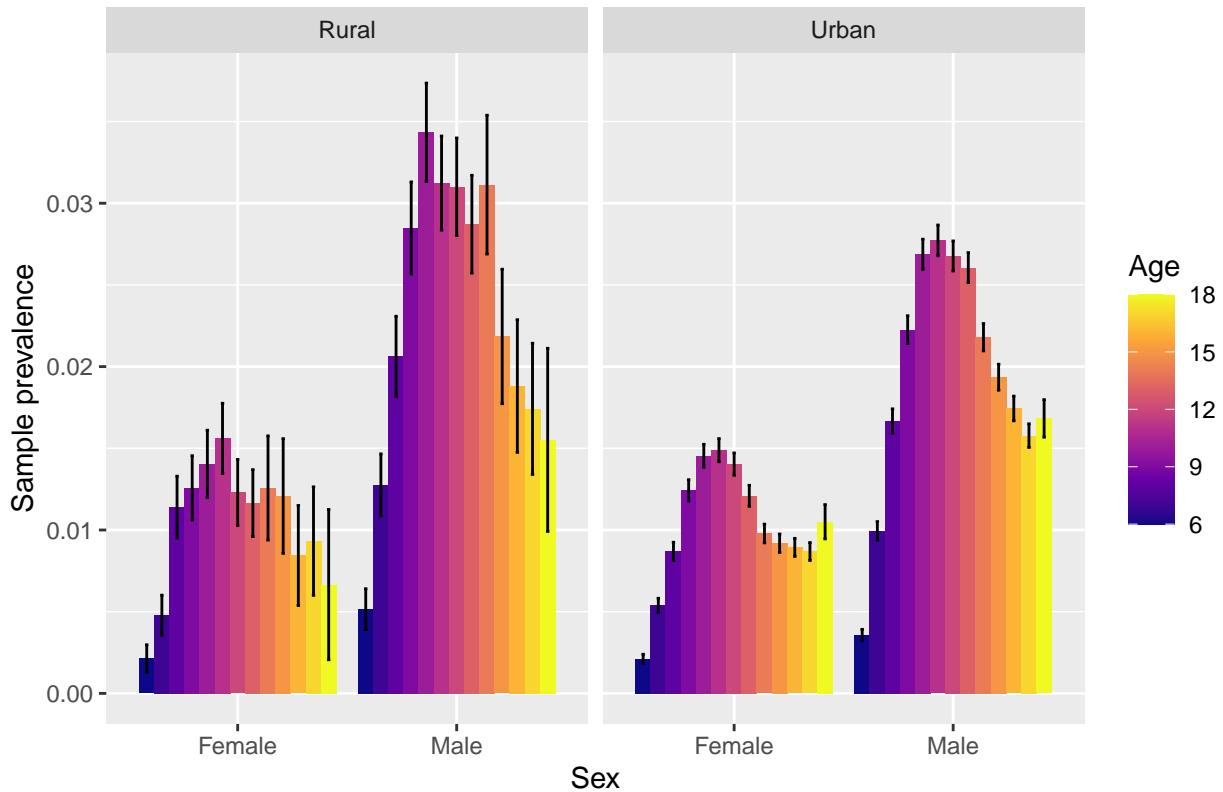
ADHD prevalence by school rurality



```
ADHD_prev_rural.age.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_rurality_code", "age_june30"))
  mutate(rurality_desc = ifelse(school_rurality_code == 0, "Urban", "Rural"))

## `summarise()` has grouped output by 'school_rurality_code', 'age_june30',
## 'sex_desc'. You can override using the `.groups` argument.
ggplot(data = ADHD_prev_rural.age.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_june30, fill = age_june30), position = position_dodge())
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_june30), width = 0.2, position = position_dodge())
  scale_fill_viridis_c(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~rurality_desc) +
  labs(title = "ADHD prevalence by school rurality",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age")
```

ADHD prevalence by school rurality



```
# ggplot(data = ADHD_prev_rural) +
#   geom_col(aes(x = as.factor(school_rurality_code), y = sample_prevalence, group = age_cat_name, fill = "Age category"))

# ggplot(data = ADHD_prev_rural) +
#   geom_col(aes(x = as.factor(school_rurality_code), y = sample_prevalence, group = age_cat_name, fill = "Age category"))

# ggplot(data = ADHD_prev_rural) +
#   geom_col(aes(x = as.factor(school_rurality_code), y = sample_prevalence, group = sex, fill = as.factor(sex)))
```

```

#           fill = "Sex")

ADHD_prev_rural <- get_grouped_prev(x = chile_bayes_ADHD, stdpop = chile_stdpop,
                                      grouping_vars = c("school_rurality_code", "age_june30", "age_cat_name"),
                                      mutate(rurality_desc = ifelse(school_rurality_code == 0, "Urban", "Rural")))

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

ADHD_prev_rural_adj <- get_adjusted_prev(ADHD_prev_rural, grouping_vars = "rurality_desc")

# ggplot(data = ADHD_prev_rural_adj) +
#   geom_col(aes(x = school_rurality_code, y = adjusted_rate), fill = "lightblue", position = "dodge") +
#   geom_errorbar(aes(x = school_rurality_code, ymin = ci_lower, ymax = ci_upper), width = 0.2) +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   labs(title = "Age- and sex-adjusted ADHD prevalence",
#        x = "Rurality status",
#        y = "Sample prevalence")

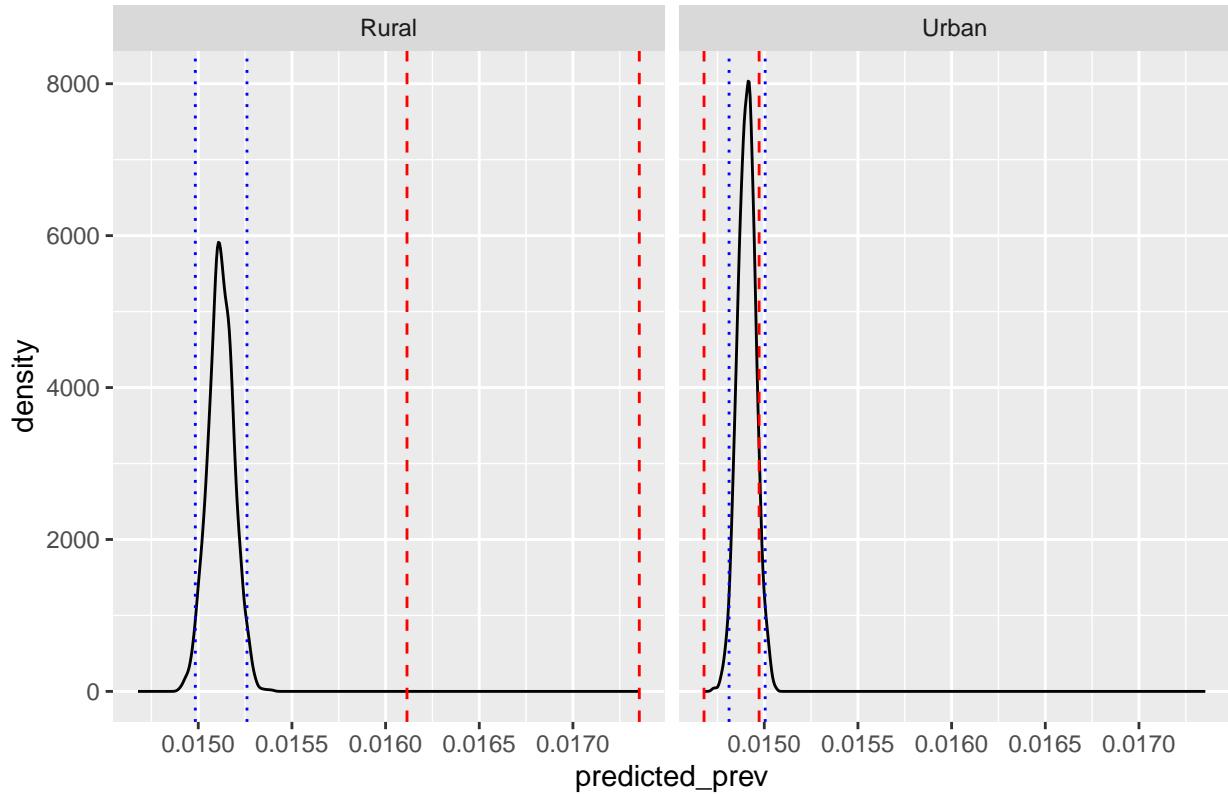
ADHD_prev_rural_post <- do_jags_rand_model(x = ADHD_prev_rural_adj,
                                              feat = "rurality_desc",
                                              model = rand_model,
                                              theta_mu = theta_mu_prior,
                                              theta_sigma = theta_sigma_prior,
                                              pars = pars,
                                              convergence_checks = FALSE) %>%
  rename("rurality_desc" = "Feat_names")

plot_post_density(ADHD_prev_rural_post, ADHD_prev_rural_adj, feat = "rurality_desc", 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 = 0.015, prior sd = 7.25e-05



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

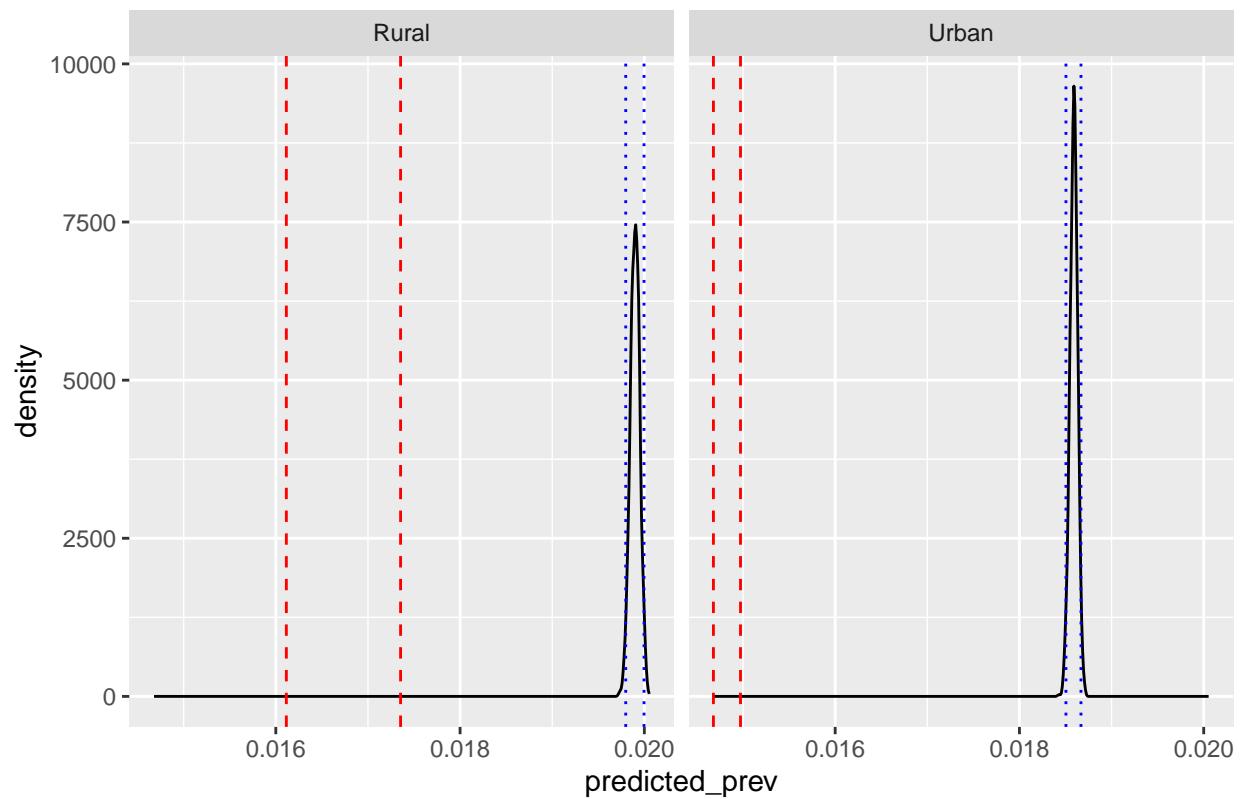
Predictions for higher population prevalence - increase prior mean

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

  plot_post_density(ADHD_prev_rural_post,
                    ADHD_prev_rural_adj,
                    feat = "rurality_desc",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}

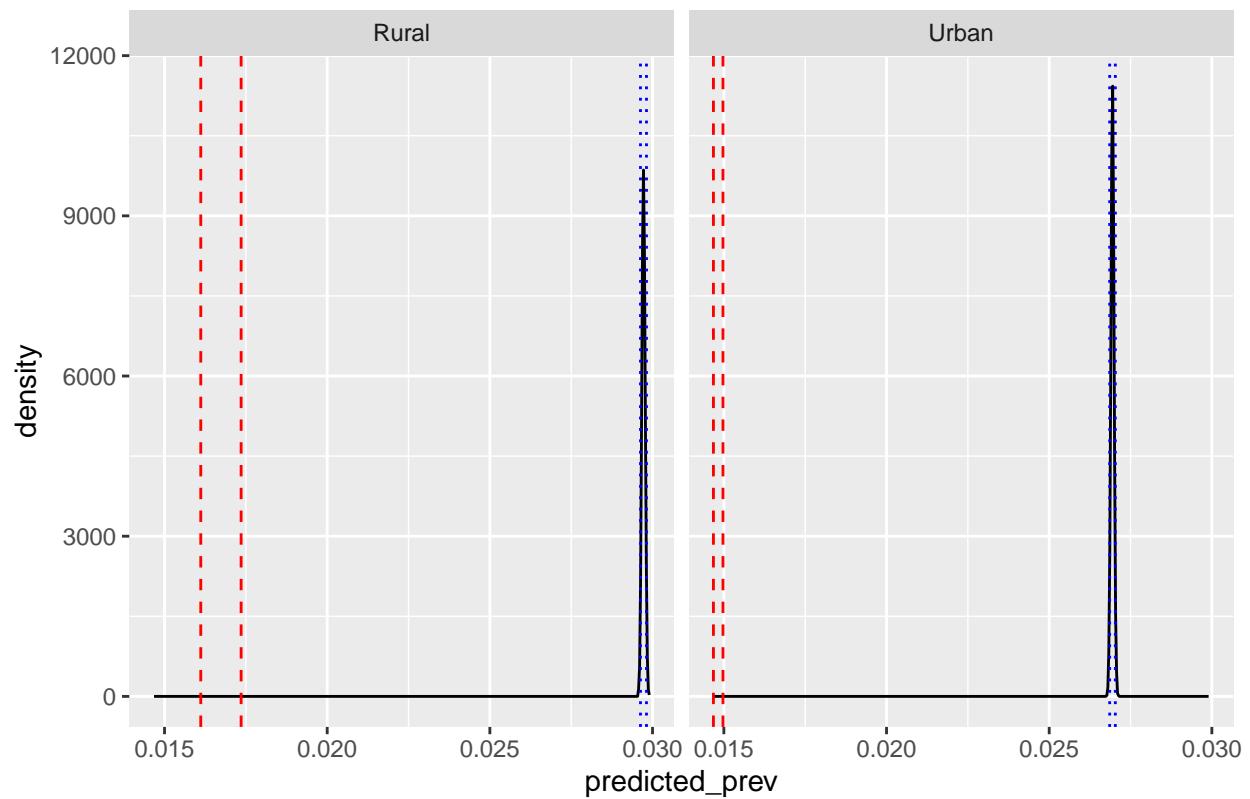
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05



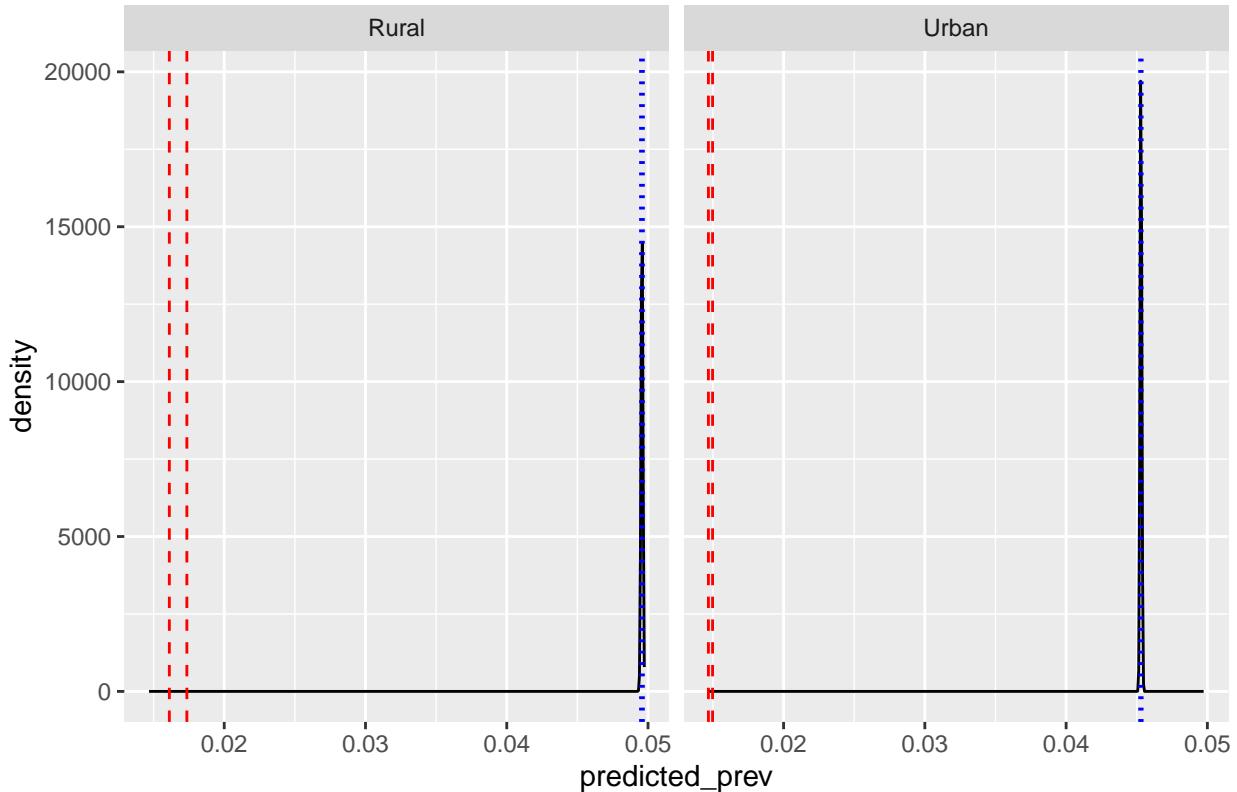
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, 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 = 0.05, prior sd = 5.1e-05



Random effect on ethnicity

Find the regions with high proportion of Mapuche population

```
mapuche_count <- chile_bayes_ADHD %>%
  group_by(school_region_name_abr, ethnicity) %>%
  summarise(count = n()) %>%
  filter(ethnicity == "Mapuche") %>%
  arrange(desc(count))

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

mapuche_prop <- chile_bayes_ADHD %>%
  group_by(school_region_name_abr, ethnicity) %>%
  summarise(count = n()) %>%
  mutate(eth_mapuche = ifelse(ethnicity == "Mapuche", "mapuche", "not")) %>%
  group_by(school_region_name_abr, eth_mapuche) %>%
  summarise(count_eth = sum(count)) %>%
  pivot_wider(names_from = eth_mapuche, values_from = count_eth) %>%
  mutate(prop_mapuche = mapuche / (mapuche + not)) %>%
  arrange(desc(prop_mapuche))

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

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

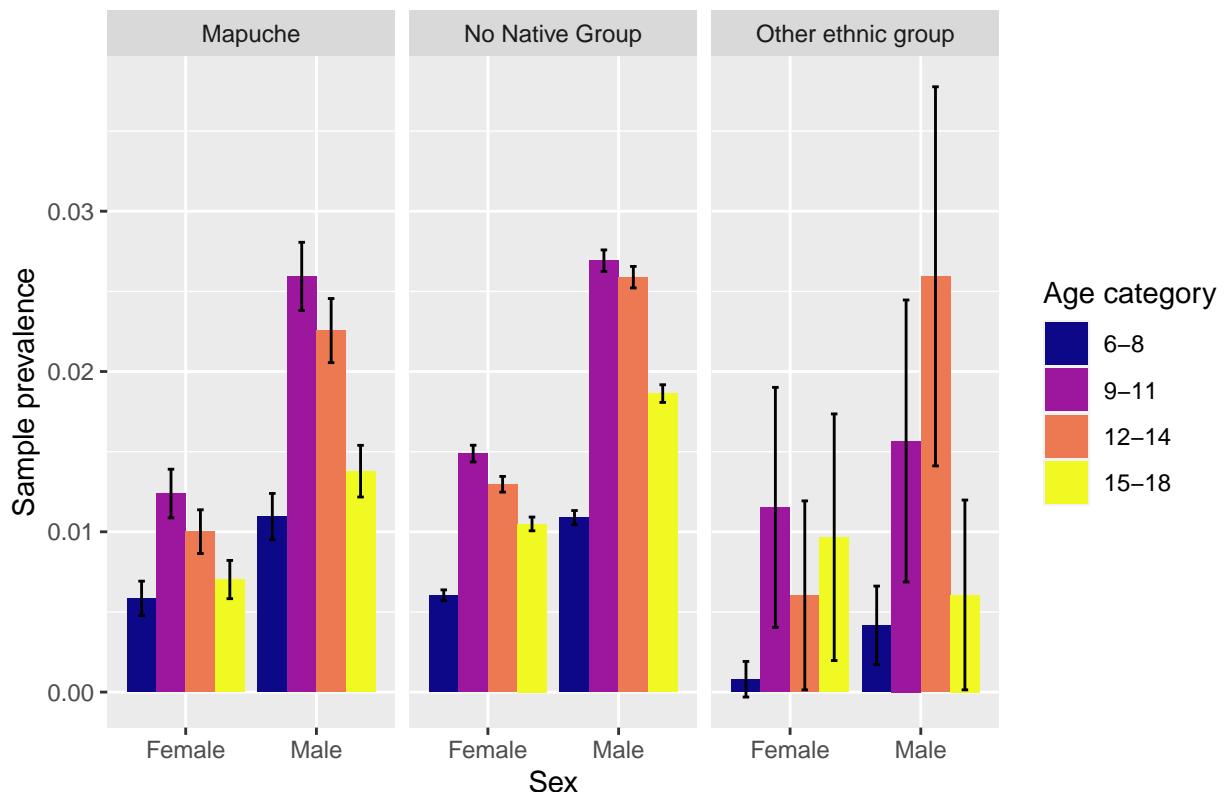
Only use regions with large Mapuche populations - specifically top 5 by count of Mapuche people (ARAUC, RM, LAGOS, BBIO, RIOS) and top 5 by proportion of region's population (ARAUC, LAGOS, RIOS, MAG, AYSEN).

```
ADHD_prev_ethnic.agecat.sex <- get_grouped_prev_plot(x = filter(chile_bayes_ADHD, school_region_name_abr %in%
  c("ARAUC", "RM", "LAGOS", "BBIO", "RIOS", "MAG", "AYSEN"),
  #c("ARAUC", "BBIO", "LAGOS", "RIOS", "MAG", "AYSEN"),
  grouping_vars = c("ethnic_2_group", "age_cat_name", "sex_desc"))

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

ggplot(data = ADHD_prev_ethnic.agecat.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_cat_name, fill = age_cat_name), position = "dodge") +
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_cat_name), width = 0.2, position = "dodge") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~ethnic_2_group) +
  labs(title = "ADHD prevalence by ethnicity",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age category")
```

ADHD prevalence by ethnicity



```
ADHD_prev_ethnic.age.sex <- get_grouped_prev_plot(x = filter(chile_bayes_ADHD, school_region_name_abr %in%
  c("ARAUC", "RM", "LAGOS", "BBIO", "RIOS", "MAG", "AYSEN"),
  #c("ARAUC", "BBIO", "LAGOS", "RIOS", "MAG", "AYSEN"),
  grouping_vars = c("ethnic_2_group", "age_june30", "sex_desc"))
```

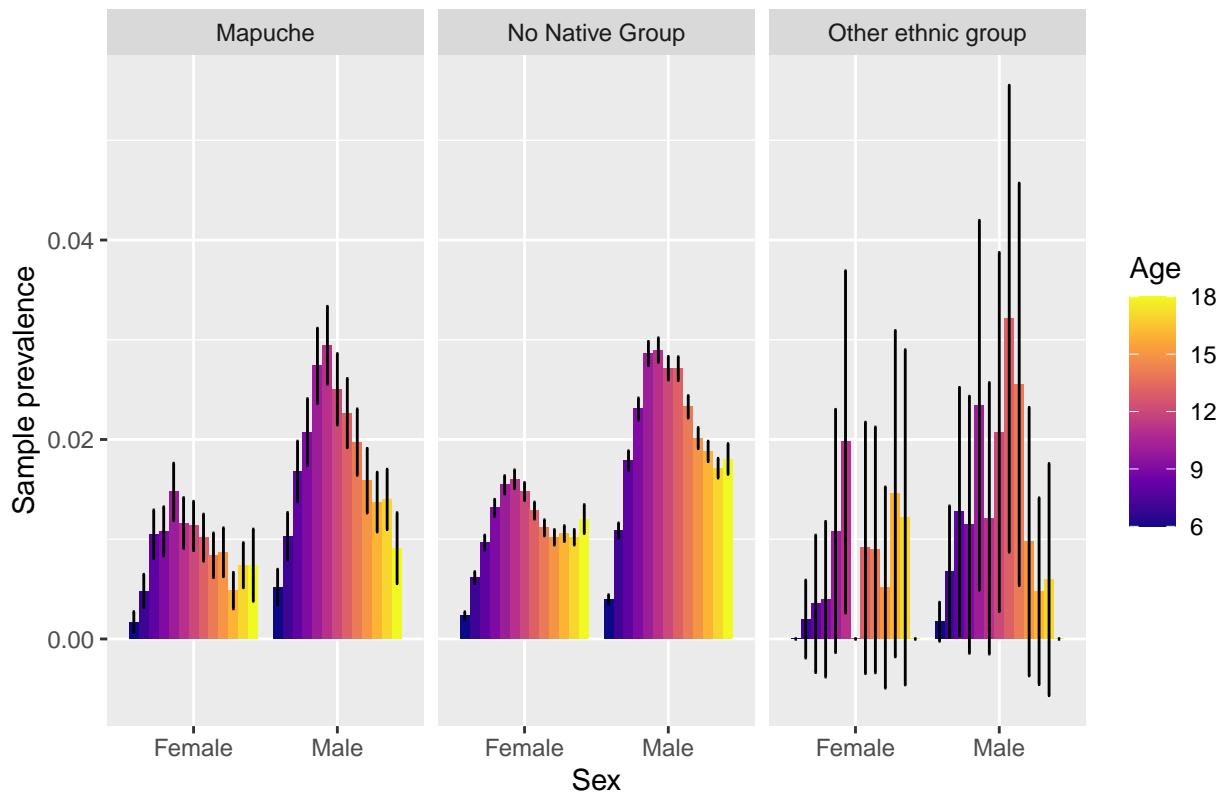
```

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

ggplot(data = ADHD_prev_ethnic.age.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_june30, fill = age_june30), position = position_dodge())
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_june30), width = 0.2, position = position_dodge())
  scale_fill_viridis_c(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~ethnic_2_group) +
  labs(title = "ADHD prevalence by ethnicity",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age")

```

ADHD prevalence by ethnicity



```

#
# ggplot(data = ADHD_prev_ethnic) +
#   geom_col(aes(x = as.factor(ethnic_2_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 = "ADHD prevalence",
#                 x = "Ethnicity",
#                 y = "Sample prevalence",
#                 fill = "Age category")
# #
# ggplot(data = ADHD_prev_ethnic) +
#   geom_col(aes(x = as.factor(ethnic_2_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 = "ADHD prevalence",
#                 x = "Ethnicity",
#                 y = "Sample prevalence",
#                 fill = "Age category")
# 
```

```

#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   facet_wrap(~sex_desc) +
#   labs(title = "ADHD prevalence by sex",
#        x = "Ethnicity",
#        y = "Sample prevalence",
#        fill = "Age category")
#
# ggplot(data = ADHD_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 = "ADHD prevalence",
#        x = "Ethnicity",
#        y = "Sample prevalence",
#        fill = "Sex")

ADHD_prev_ethnic <- chile_bayes_ADHD %>%
  #filter(school_region_name_abr %in% c("ARAUC", "BBIO", "LAGOS", "RIOS", "RM")) %>%
  filter(school_region_name_abr %in% c("ARAUC", "RM", "LAGOS", "BBIO", "RIOS", "MAG", "AYSEN")) %>%
  get_grouped_prev(stdpop = chile_stdpop,
                   grouping_vars = c("ethnic_2_group", "age_june30", "age_cat_name", "sex", "sex_desc",
                                     "grouping_vars"))
  ## `summarise()` has grouped output by 'ethnic_2_group', 'age_june30',
  ## 'age_cat_name', 'sex', 'sex_desc'. You can override using the `.groups` argument.

ADHD_prev_ethnic_adj <- get_adjusted_prev(ADHD_prev_ethnic, grouping_vars = "ethnic_2_group")

# ggplot(data = ADHD_prev_ethnic_adj) +
#   geom_col(aes(x = ethnic_2_group, y = adjusted_rate), fill = "lightblue", position = "dodge") +
#   geom_errorbar(aes(x = ethnic_2_group, ymin = ci_lower, ymax = ci_upper), width = 0.2) +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   labs(title = "Age- and sex-adjusted ADHD prevalence",
#        x = "Ethnicity",
#        y = "Sample prevalence")

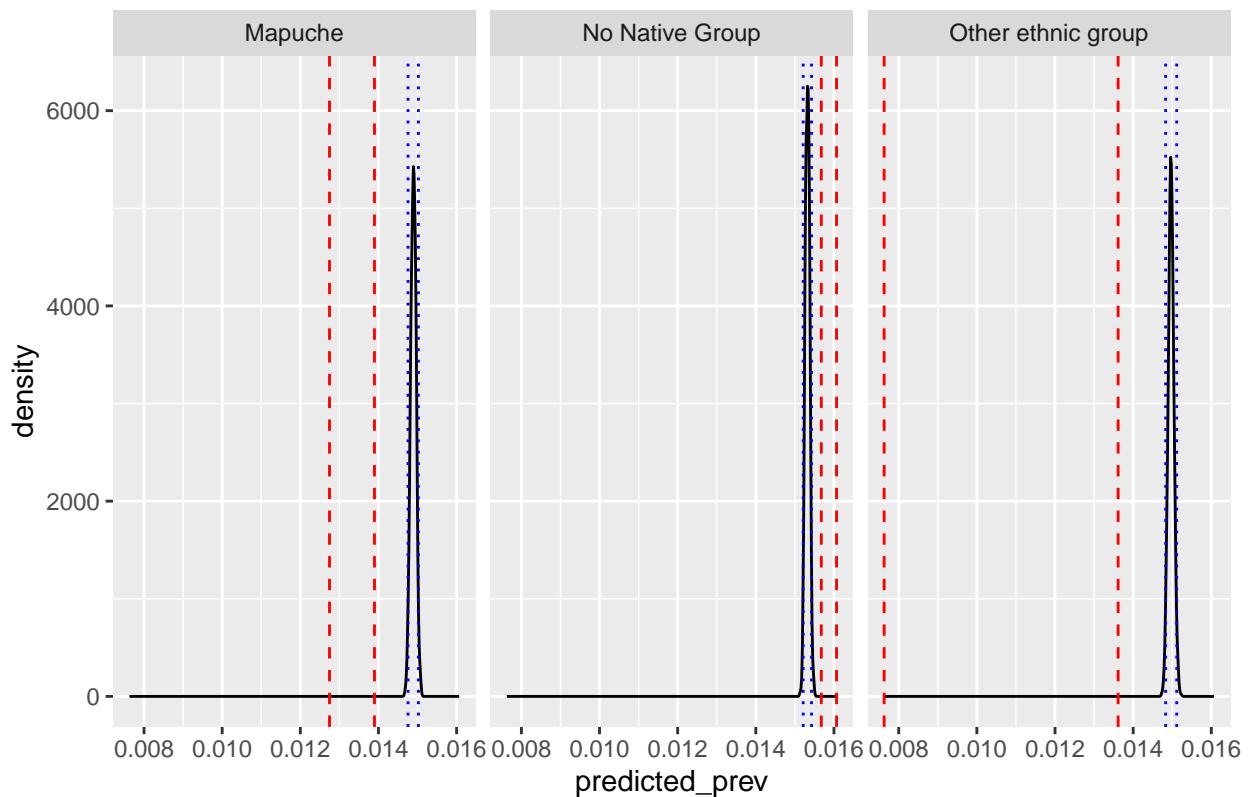
ADHD_prev_ethnic_post <- do_jags_rand_model(x = ADHD_prev_ethnic_adj,
                                              feat = "ethnic_2_group",
                                              model = rand_model,
                                              theta_mu = theta_mu_prior,
                                              theta_sigma = theta_sigma_prior,
                                              pars = pars,
                                              convergence_checks = FALSE) %>%
  rename("ethnic_2_group" = "Feat_names")

plot_post_density(ADHD_prev_ethnic_post, ADHD_prev_ethnic_adj, feat = "ethnic_2_group", theta_mu = theta_mu_prior)

## 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 = 7.25e-05



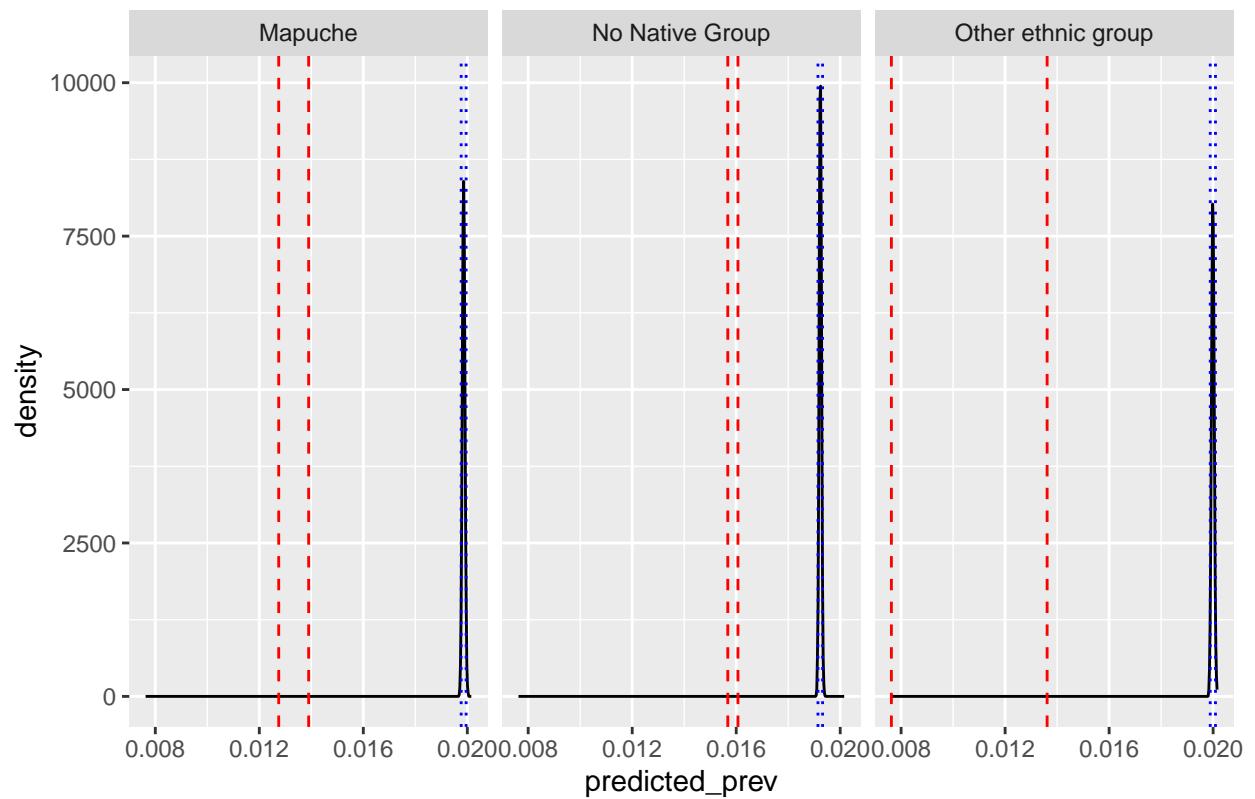
Predictions for higher population prevalence - increase prior mean

```
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_ethnic_post <- do_jags_rand_model(x = ADHD_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(ADHD_prev_ethnic_post,
                    ADHD_prev_ethnic_adj,
                    feat = "ethnic_2_group",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j])
}

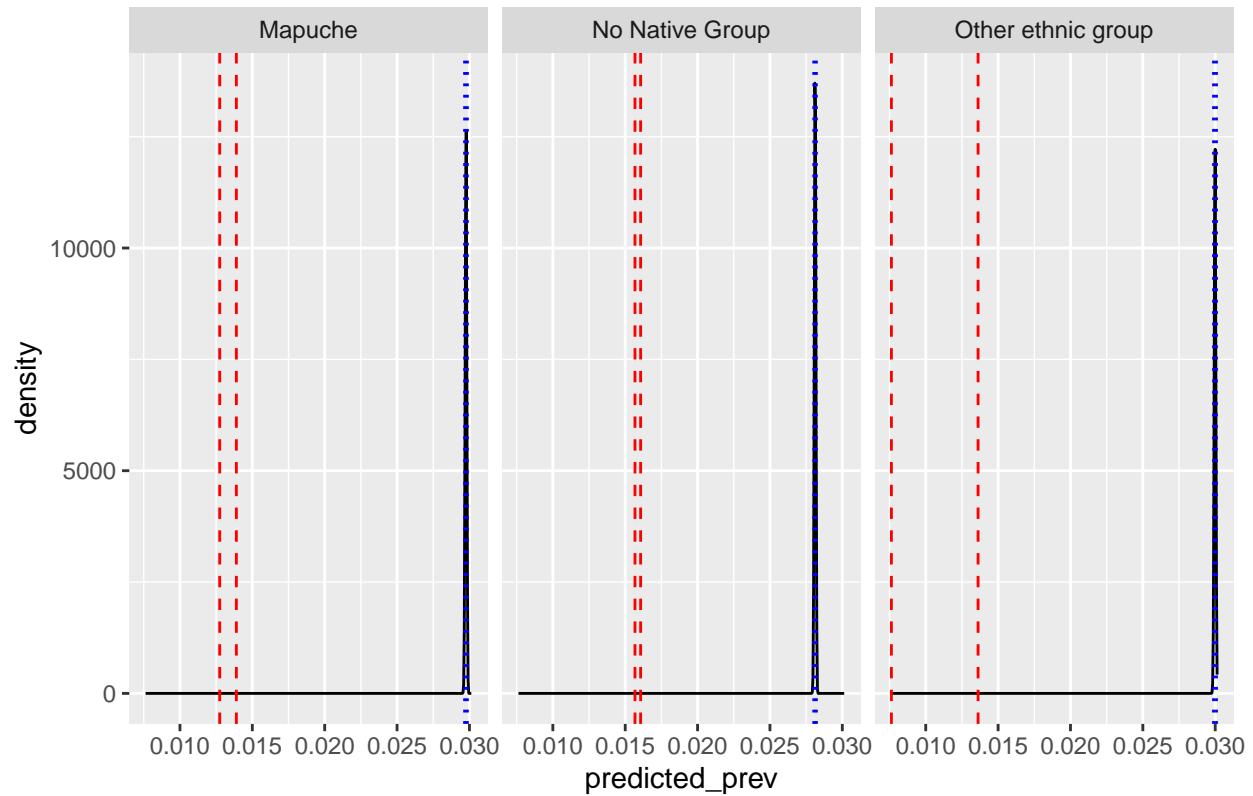
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05



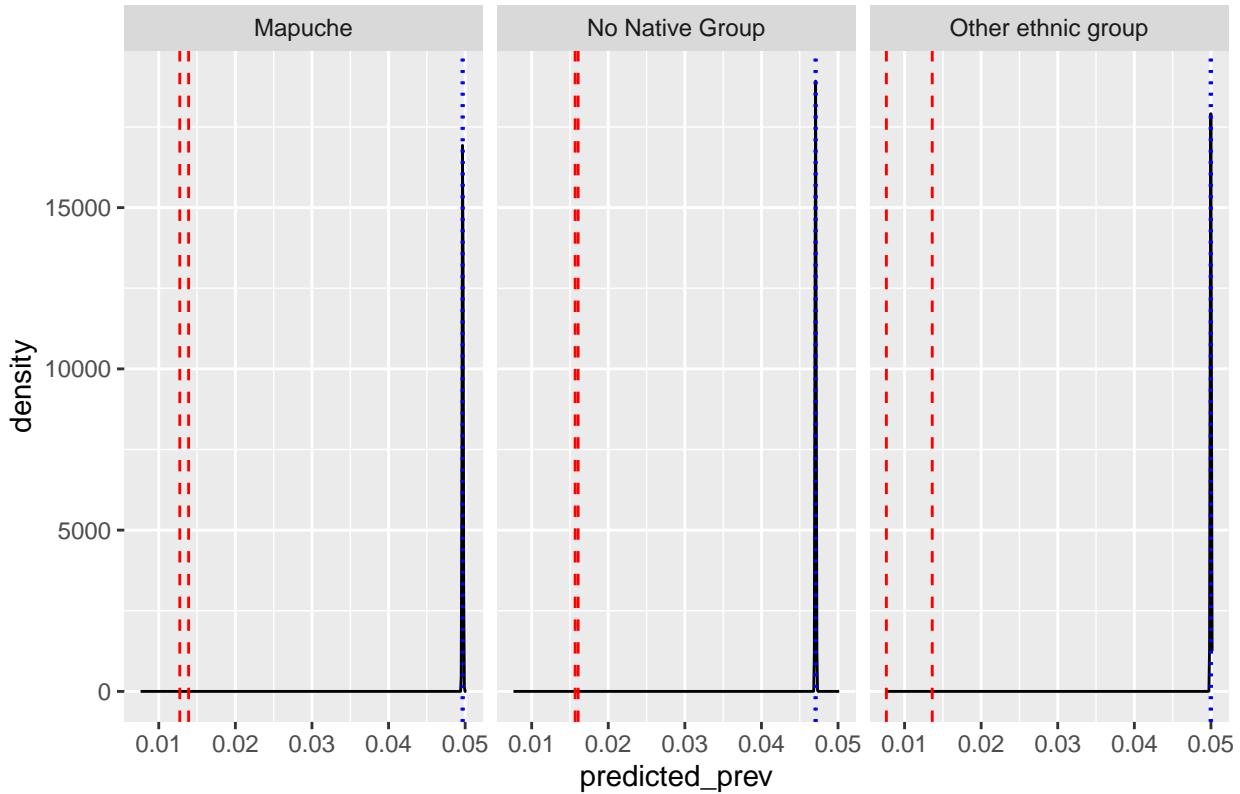
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, 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 = 0.05, prior sd = 5.1e-05



Random effect on ethnicity, sexes separate

```
# Females
ADHD_prev_ethnic_f <- chile_bayes_ADHD %>%
  filter(school_region_name_abr %in% c("ARAUC", "RM", "LAGOS", "BBIO", "RIOS", "MAG", "AYSEN")) %>%
  filter(sex == 2) %>%
  get_grouped_prev(stdpop = chile_stdpop_f, grouping_vars = c("ethnic_2_group", "age_june30", "sex", "A"))

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

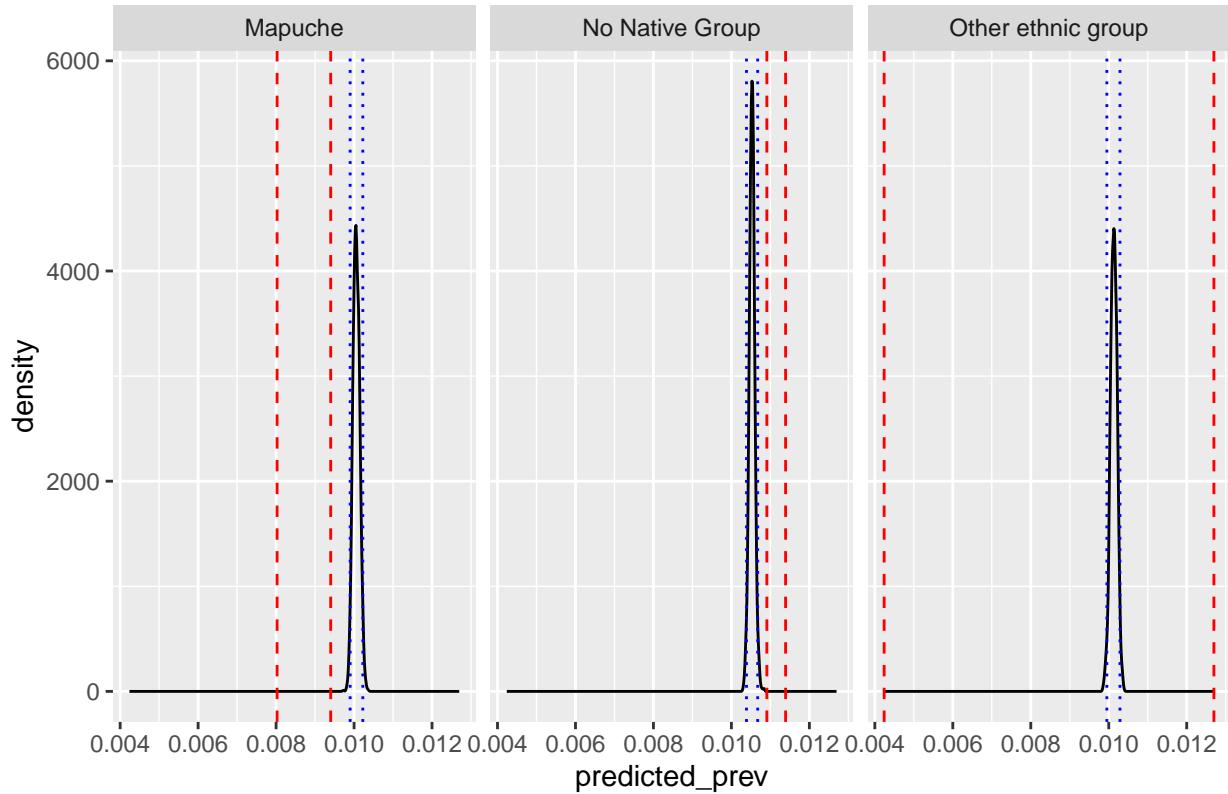
ADHD_prev_ethnic_adj_f <- get_adjusted_prev(ADHD_prev_ethnic_f, grouping_vars = "ethnic_2_group")

ADHD_prev_ethnic_post_f <- do_jags_rand_model(x = ADHD_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(ADHD_prev_ethnic_post_f, ADHD_prev_ethnic_adj_f, feat = "ethnic_2_group", theta_mu = 0.05)

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

Prior mean = 0.0101, prior sd = 8.7e-05 (females)



```
# Males
ADHD_prev_ethnic_m <- chile_bayes_ADHD %>%
  filter(school_region_name_abr %in% c("ARAUC", "RM", "LAGOS", "BBIO", "RIOS", "MAG", "AYSEN")) %>%
  filter(sex == 1) %>%
  get_grouped_prev(stdpop = chile_stdpop_m, grouping_vars = c("ethnic_2_group", "age_june30", "sex", "A"))

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

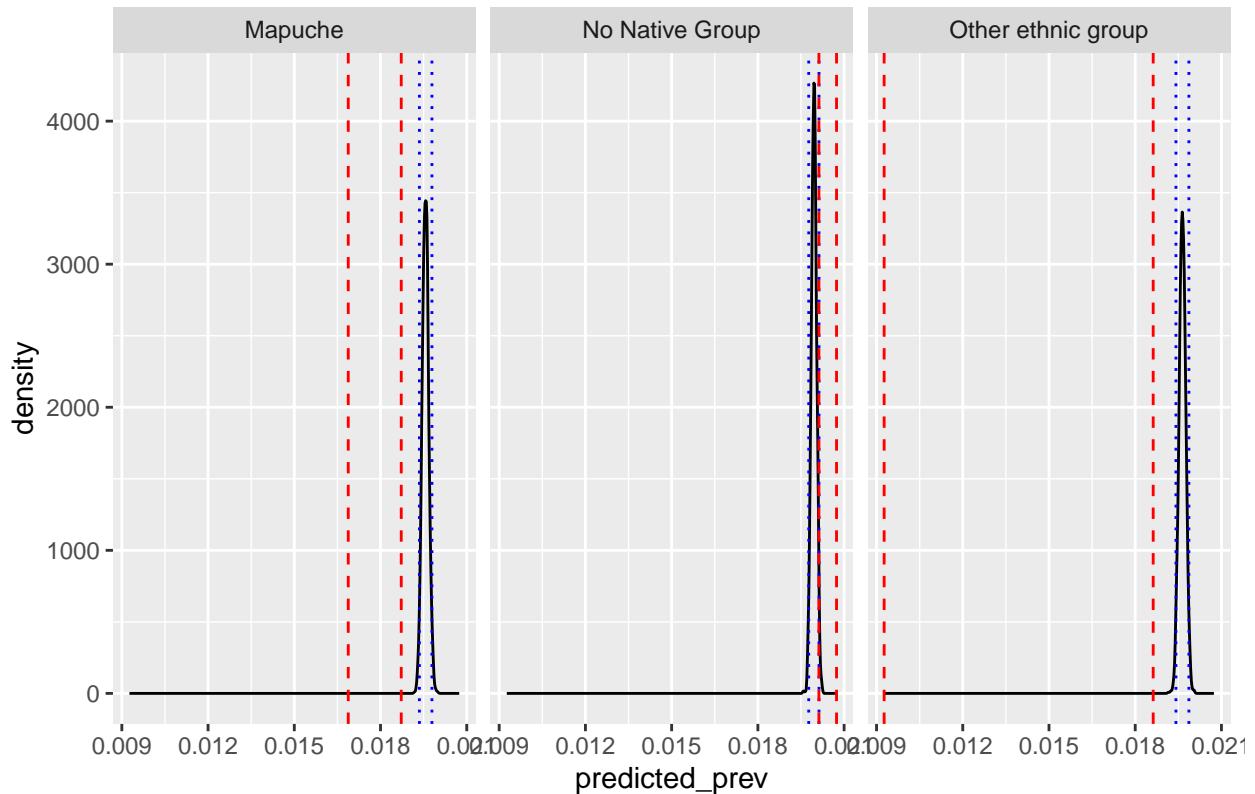
ADHD_prev_ethnic_adj_m <- get_adjusted_prev(ADHD_prev_ethnic_m, grouping_vars = "ethnic_2_group")

ADHD_prev_ethnic_post_m <- do_jags_rand_model(x = ADHD_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(ADHD_prev_ethnic_post_m, ADHD_prev_ethnic_adj_m, feat = "ethnic_2_group", theta_mu = ...)

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

Prior mean = 0.0197, prior sd = 0.000114 (males)



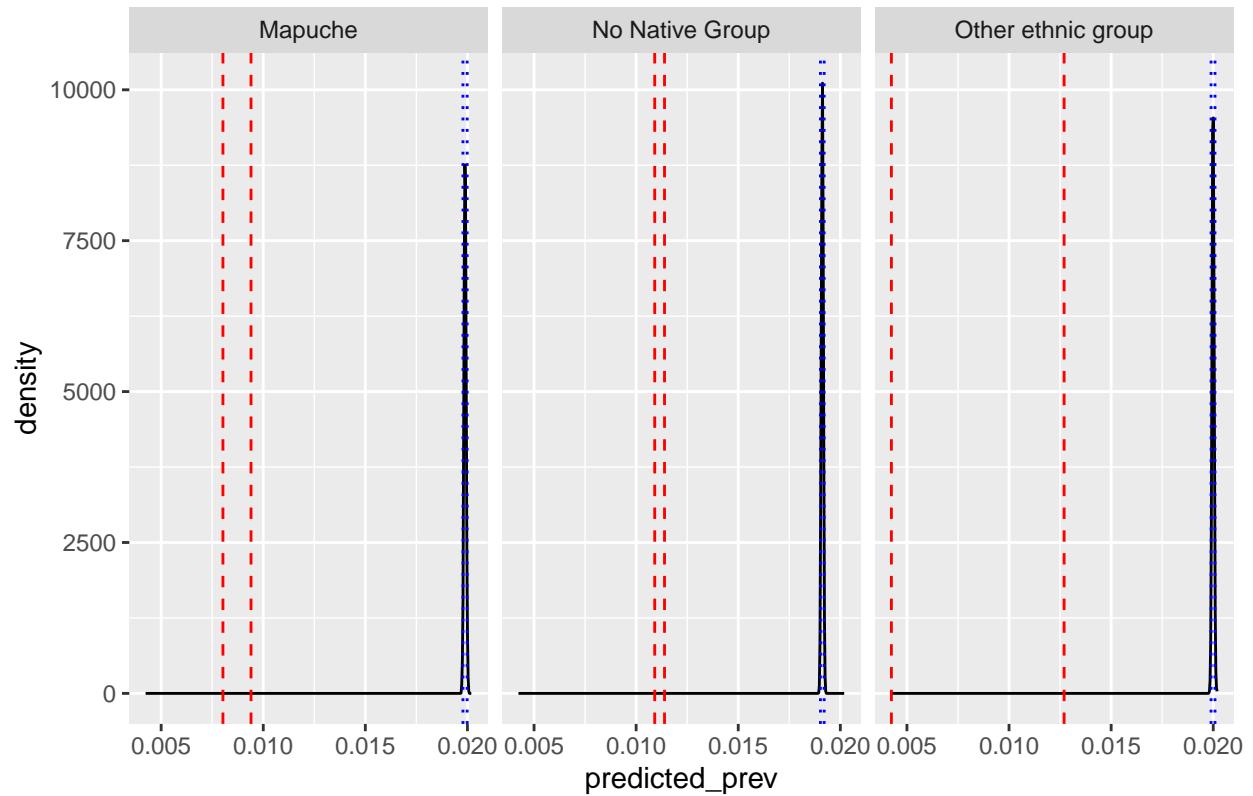
```
# Females
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_ethnic_post_f <- do_jags_rand_model(x = ADHD_prev_ethnic_adj_f,
                                                 feat = "ethnic_2_group",
                                                 model = rand_model,
                        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

}

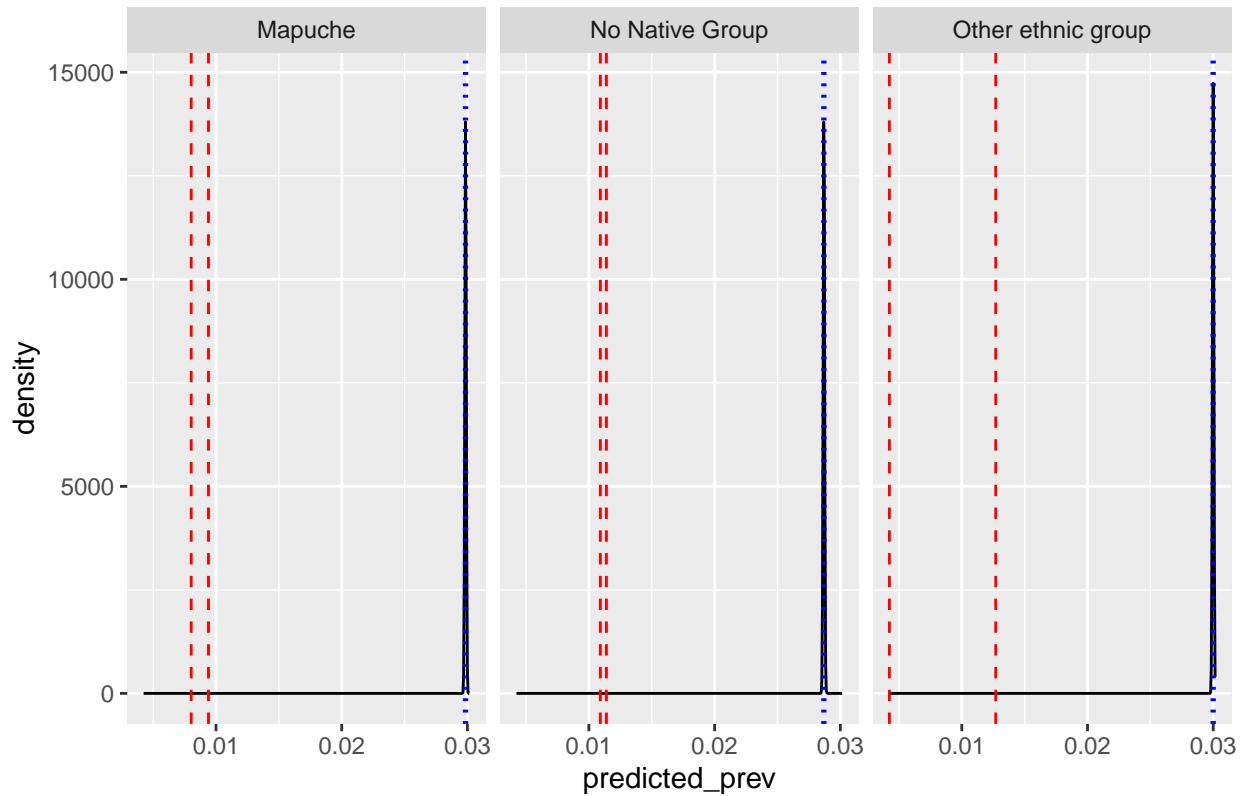
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05 (females)



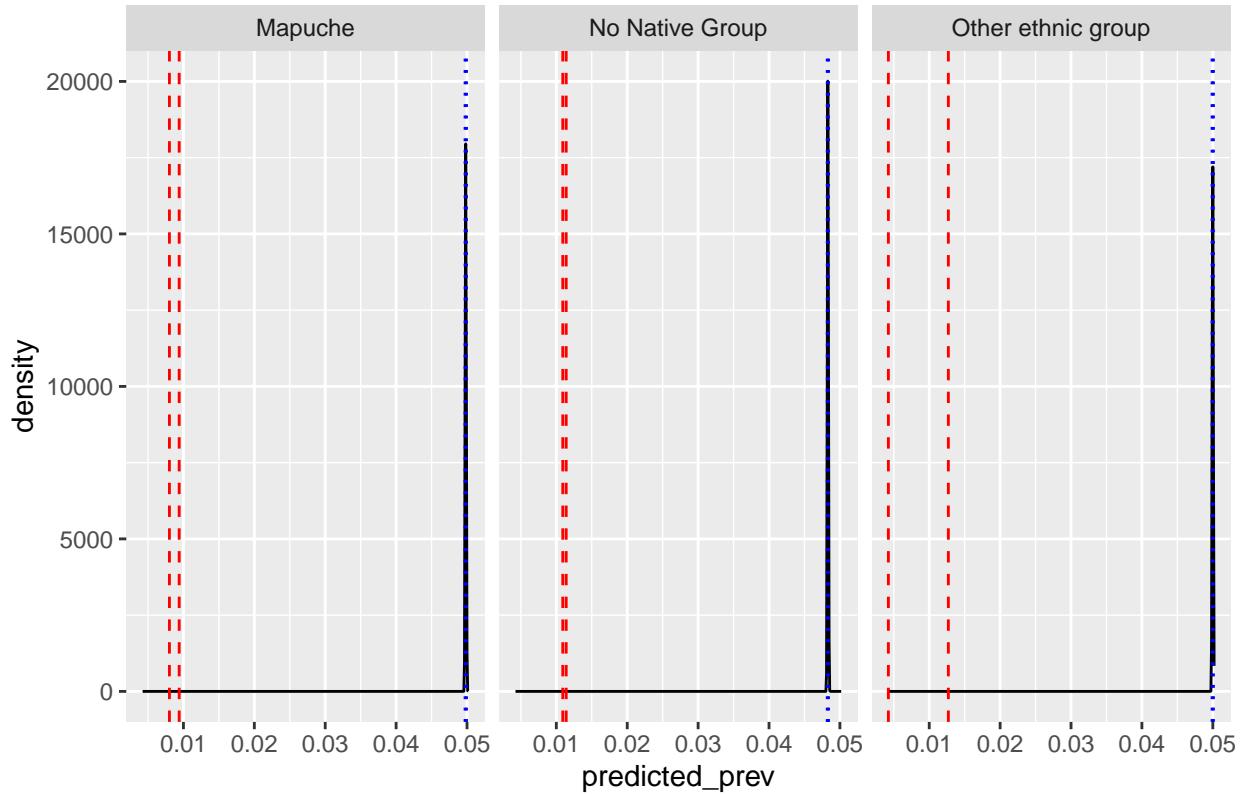
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, prior sd = 5.1e-05 (females)



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

Prior mean = 0.05, prior sd = 5.1e-05 (females)



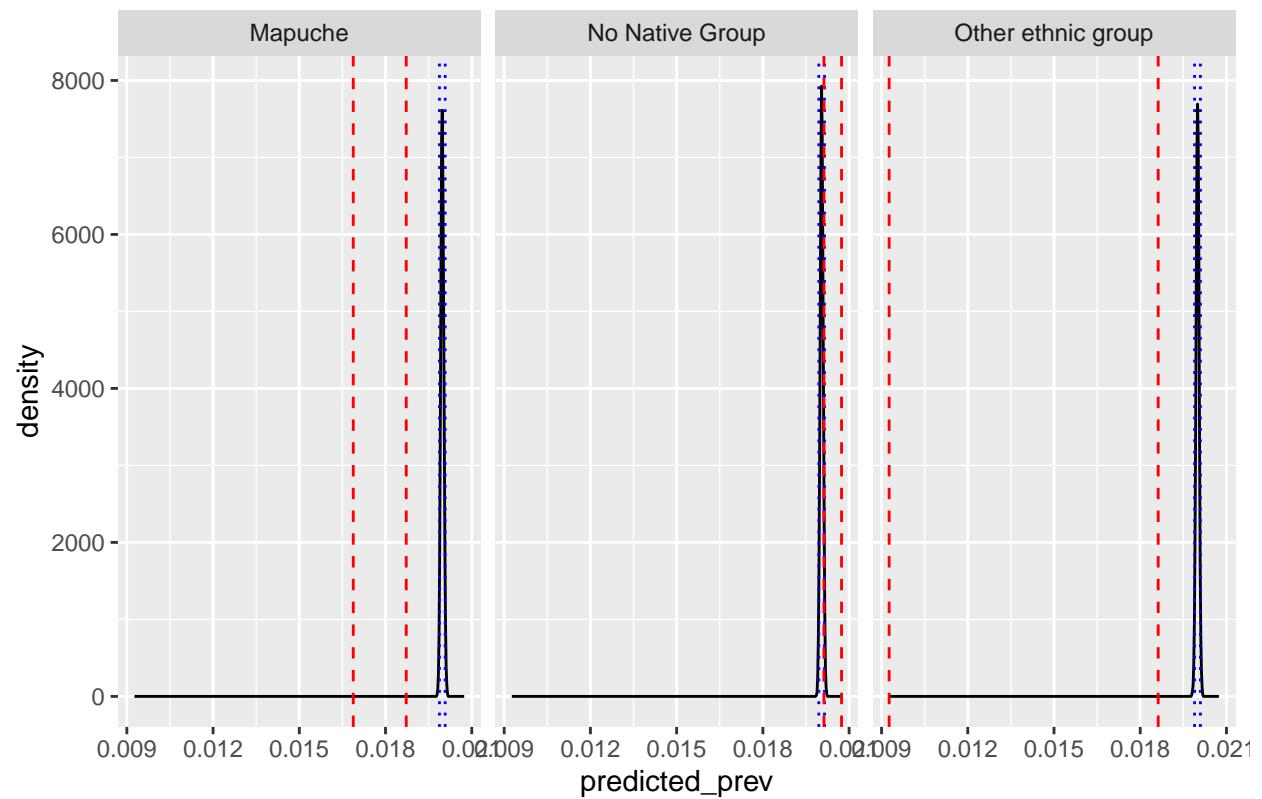
```
# Males
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_ethnic_post_m <- do_jags_rand_model(x = ADHD_prev_ethnic_adj_m,
                                                feat = "ethnic_2_group",
                                                model = rand_model,
                            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

}

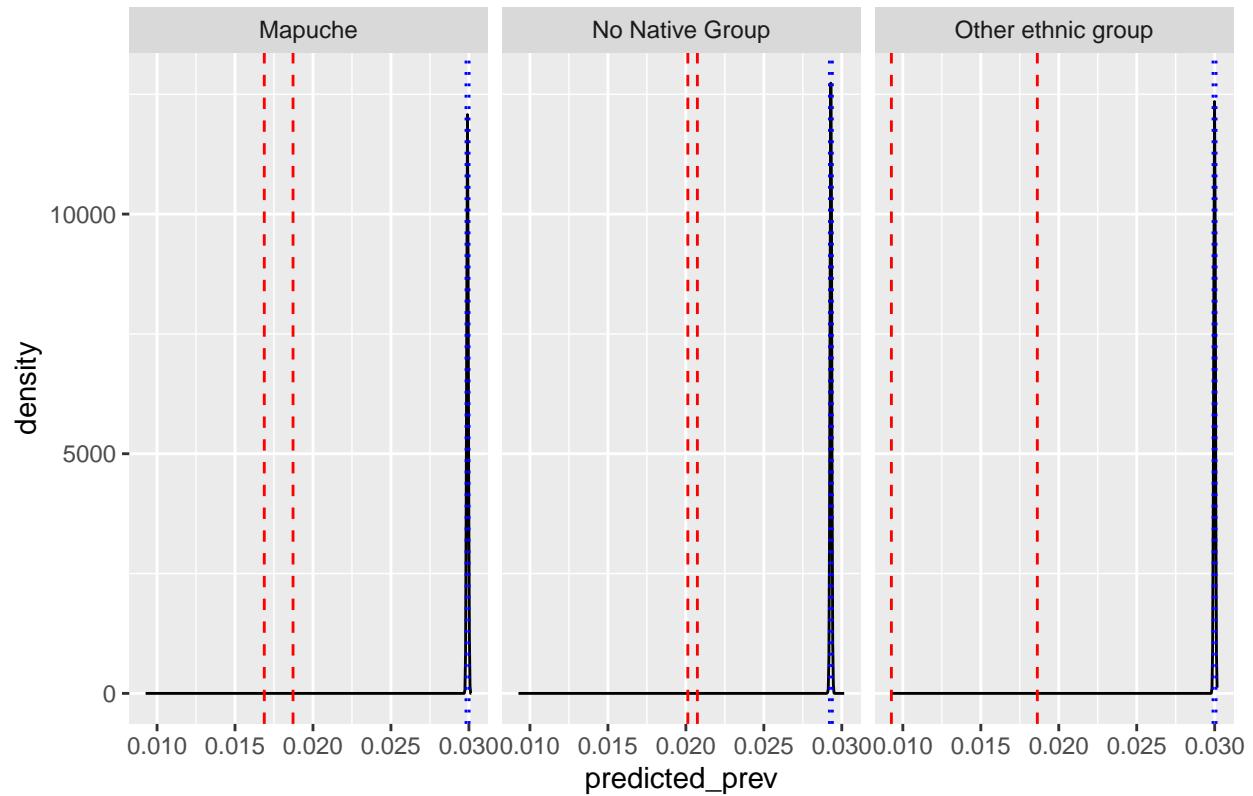
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05 (males)



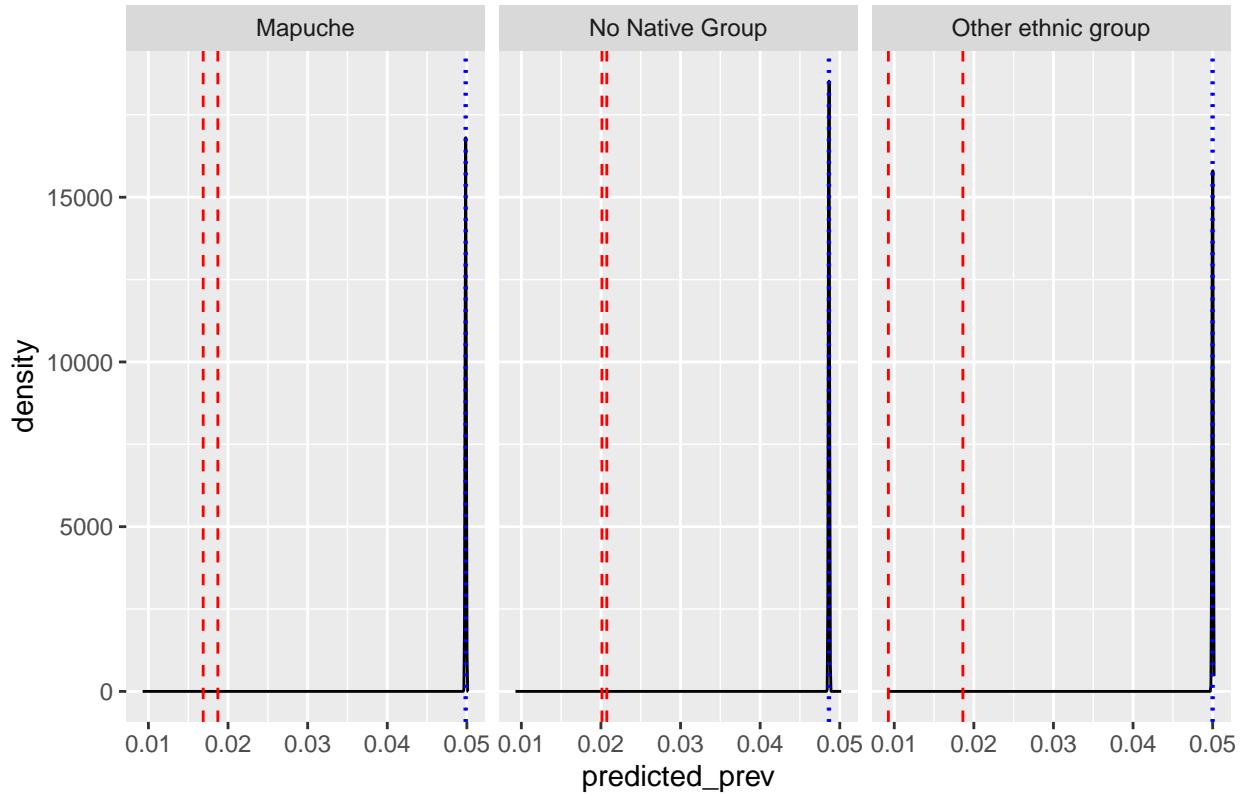
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, prior sd = 5.1e-05 (males)



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

Prior mean = 0.05, prior sd = 5.1e-05 (males)



Random effect on economic status - all economic groups

```
ADHD_prev_econA.agecat.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_fee", "age_cat_name"))

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

ggplot(data = ADHD_prev_econA.agecat.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_cat_name, fill = age_cat_name), position = "stack") +
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_cat_name), width = 0.2, scale_fill_viridis_d(option = "plasma")) +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~school_fee) +
  labs(title = "ADHD prevalence by economic status (all)",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age category")
```

ADHD prevalence by economic status (all)

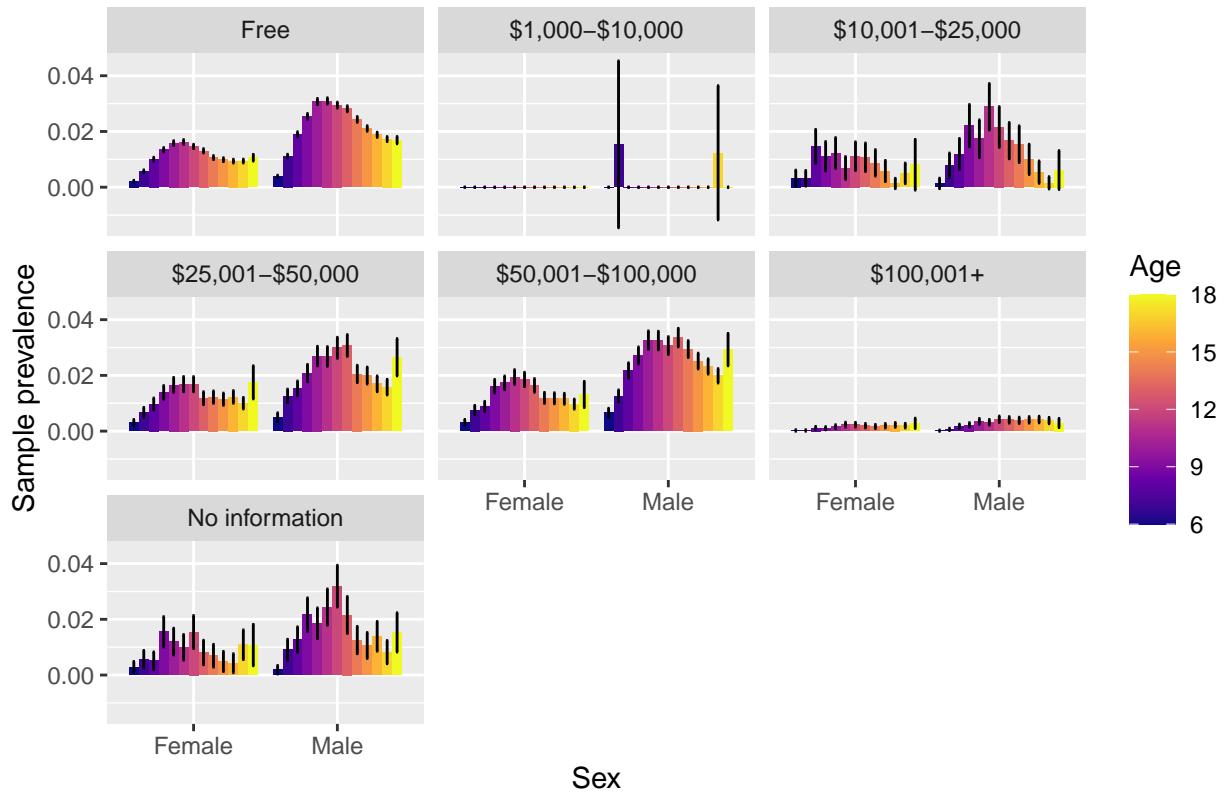


```
ADHD_prev_econA.age.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_fee", "age_june30", "sex_desc"))

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

ggplot(data = ADHD_prev_econA.age.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_june30, fill = age_june30), position = position_dodge())
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_june30), width = 0.2, position = position_dodge())
  scale_fill_viridis_c(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~school_fee) +
  labs(title = "ADHD prevalence by economic status (all)",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age")
```

ADHD prevalence by economic status (all)



```
# ggplot(data = ADHD_prev_econ) +
#   geom_col(aes(x = as.factor(school_fee), y = sample_prevalence, group = age_cat_name, fill = age_cat_name),
#   scale_fill_viridis_d() +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   facet_wrap(~sex_desc) +
#   labs(title = "ADHD prevalence by sex",
#       x = "Economic status (all)",
#       y = "Sample prevalence",
#       fill = "Age category")
#
# ggplot(data = ADHD_prev_econ) +
#   geom_col(aes(x = as.factor(school_fee), 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 = "ADHD prevalence",
#       x = "Economic status (all)",
#       y = "Sample prevalence",
#       fill = "Sex")
#
ADHD_prev_econ <- get_grouped_prev(x = chile_bayes_ADHD, stdpop = chile_stdpop,
                                     grouping_vars = c("school_fee", "school_fee_group", "age_june30", "age_cat_name"))
## `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.
```

```

ADHD_prev_econ_adj <- get_adjusted_prev(ADHD_prev_econ, grouping_vars = "school_fee")

# ggplot(data = ADHD_prev_econ_adj) +
#   geom_col(aes(x = school_fee, y = adjusted_rate), fill = "lightblue", position = "dodge") +
#   geom_errorbar(aes(x = school_fee, ymin = ci_lower, ymax = ci_upper), width = 0.2) +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   labs(title = "Age- and sex-adjusted ADHD prevalence",
#        x = "Economic status (all)",
#        y = "Sample prevalence")

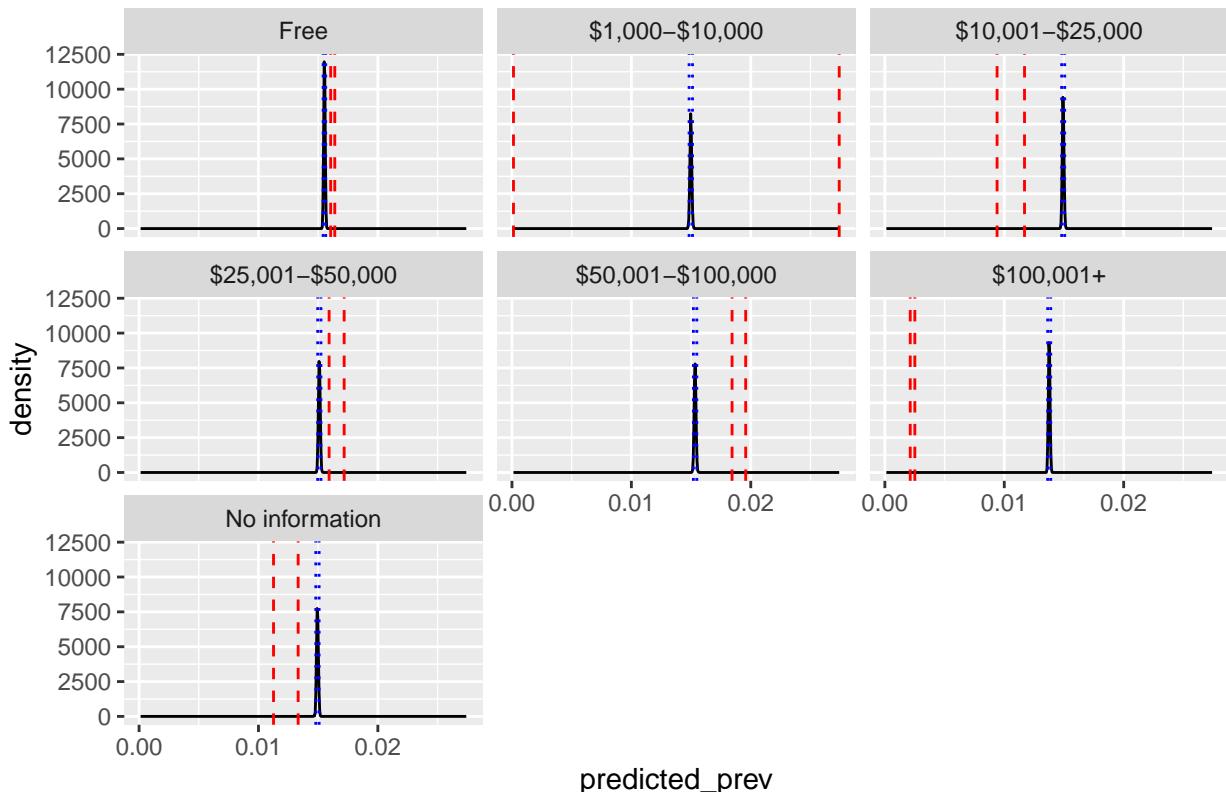
ADHD_prev_econ_post <- do_jags_rand_model(x = ADHD_prev_econ_adj,
                                             feat = "school_fee",
                                             model = rand_model,
                                             theta_mu = theta_mu_prior,
                               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

plot_post_density(ADHD_prev_econ_post, ADHD_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.

```

Prior mean = 0.015, prior sd = 7.25e-05



Predictions for higher population prevalence - increase prior mean

```

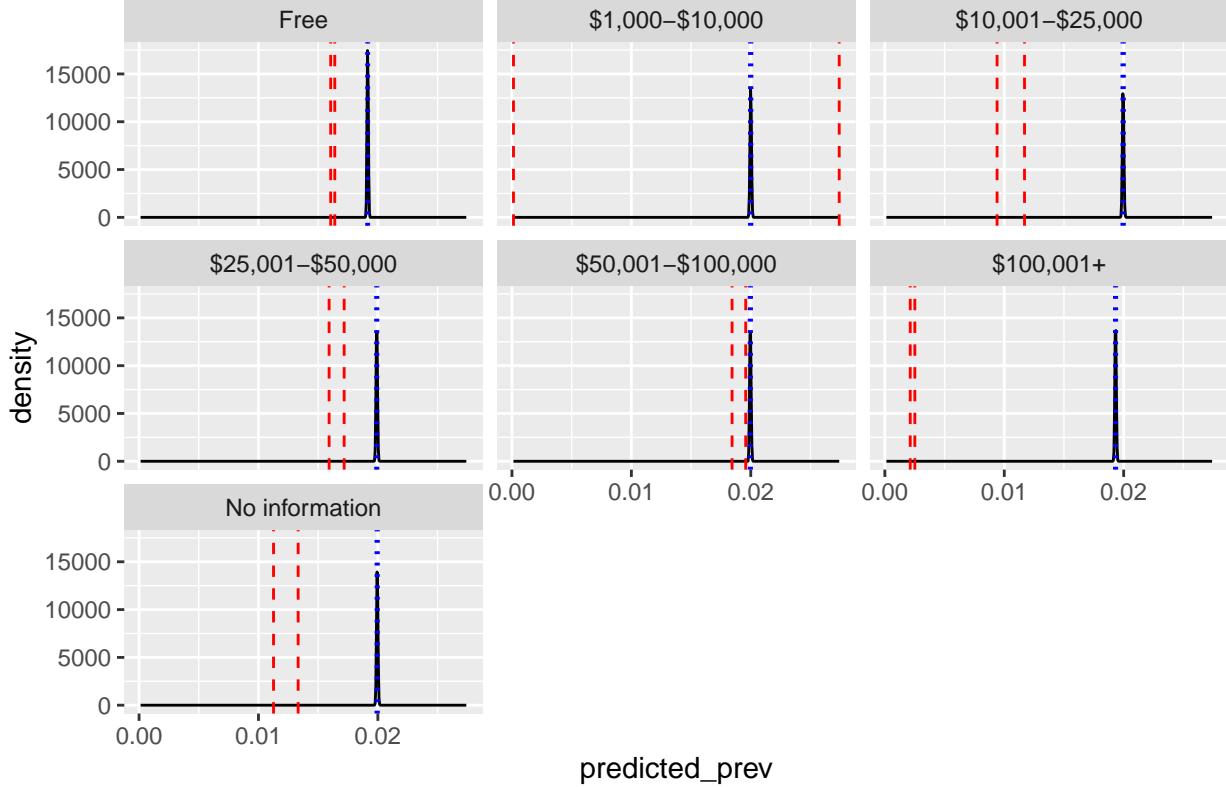
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_econ_post <- do_jags_rand_model(x = ADHD_prev_econ_adj,
                                              feat = "school_fee",
                                              model = rand_model,
                                              theta_mu =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

  plot_post_density(ADHD_prev_econ_post,
                    ADHD_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.

```

Prior mean = 0.02, prior sd = 5.1e-05

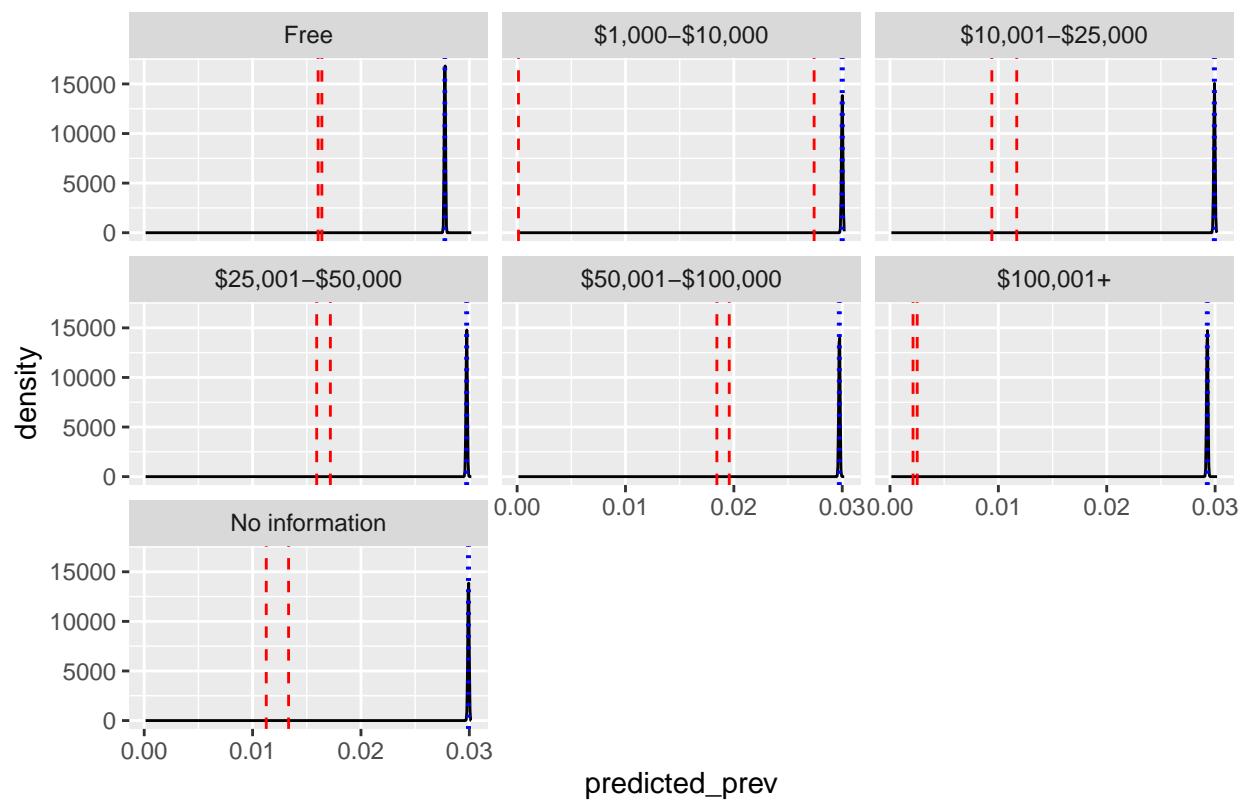


```

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

```

Prior mean = 0.03, 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 = 0.05, prior sd = 5.1e-05



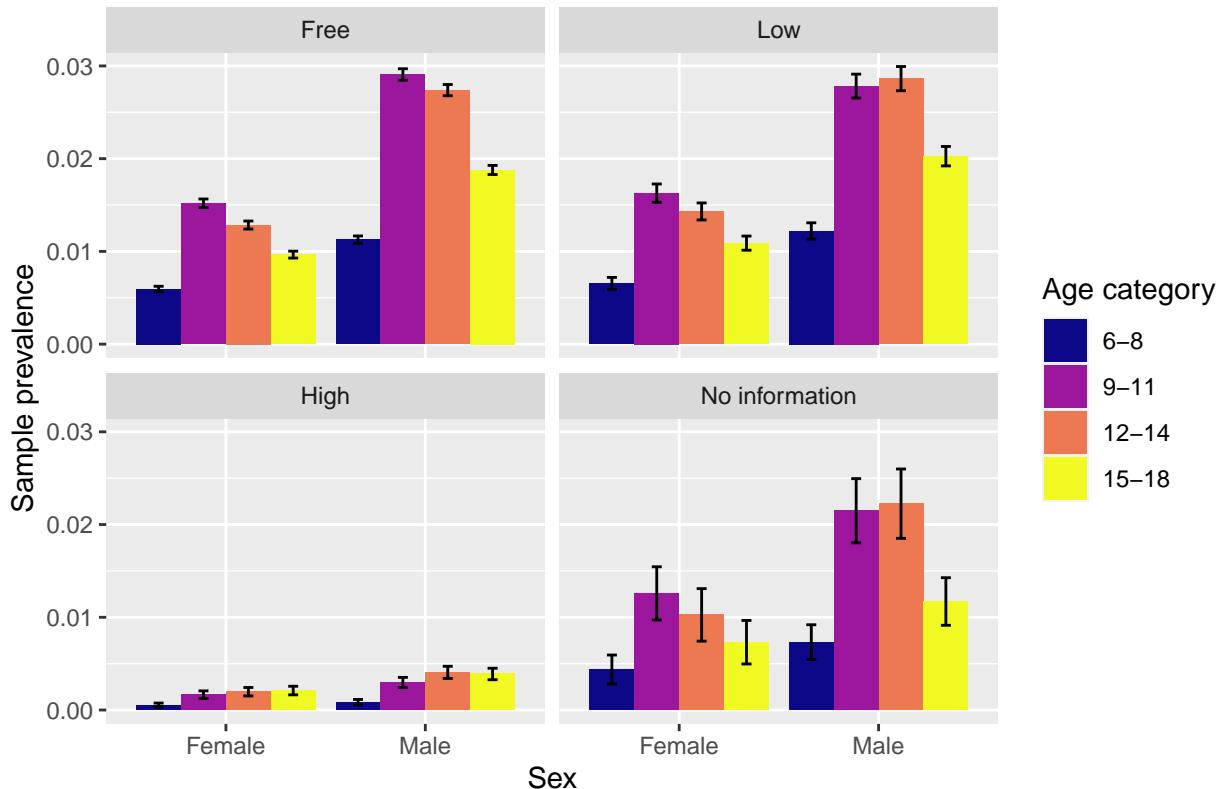
Random effect on economic status - free education, low status, high status and no information

```
ADHD_prev_econG.agecat.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_fee_group", "age_cat_name"))

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

ggplot(data = ADHD_prev_econG.agecat.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_cat_name, fill = age_cat_name), position =
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_cat_name), width = 0.2,
  scale_fill_viridis_d(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~school_fee_group) +
  labs(title = "ADHD prevalence by economic status (grouped)",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age category")
```

ADHD prevalence by economic status (grouped)

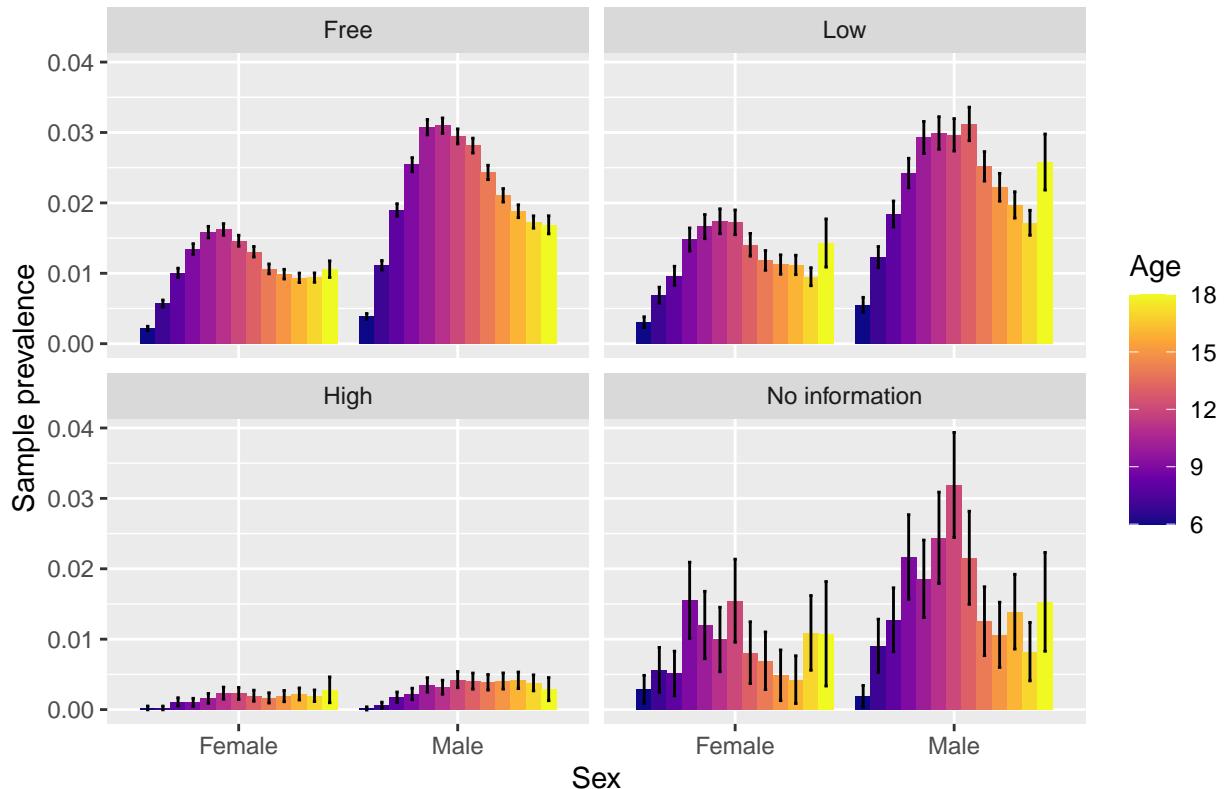


```
ADHD_prev_econG.age.sex <- get_grouped_prev_plot(x = chile_bayes_ADHD, c("school_fee_group", "age_june30"))

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

ggplot(data = ADHD_prev_econG.age.sex) +
  geom_col(aes(x = sex_desc, y = sample_prevalence, group = age_june30, fill = age_june30), position = position_dodge())
  geom_errorbar(aes(x = sex_desc, ymin = ci_lower, ymax = ci_upper, group = age_june30), width = 0.2, position = position_dodge())
  scale_fill_viridis_c(option = "plasma") +
  #theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2)) +
  facet_wrap(~school_fee_group) +
  labs(title = "ADHD prevalence by economic status (grouped)",
       x = "Sex",
       y = "Sample prevalence",
       fill = "Age")
```

ADHD prevalence by economic status (grouped)



```
# ggplot(data = ADHD_prev_econ) +
#   geom_col(aes(x = as.factor(school_fee_group), y = sample_prevalence, group = age_cat_name, fill = age_cat_name),
#   scale_fill_viridis_d() +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   facet_wrap(~sex_desc) +
#   labs(title = "ADHD prevalence by sex",
#       x = "Economic status (grouped)",
#       y = "Sample prevalence",
#       fill = "Age category")
#
# ggplot(data = ADHD_prev_econ) +
#   geom_col(aes(x = as.factor(school_fee_group), y = sample_prevalence, group = sex, fill = sex),
#   scale_fill_manual(values = c("#03CEA4", "#802392")) +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   labs(title = "ADHD prevalence",
#       x = "Economic status (grouped)",
#       y = "Sample prevalence",
#       fill = "Sex")
#
ADHD_prev_econG_adj <- get_adjusted_prev(ADHD_prev_econ, grouping_vars = "school_fee_group")
#
# ggplot(data = ADHD_prev_econG_adj) +
#   geom_col(aes(x = school_fee_group, y = adjusted_rate), fill = "lightblue", position = "dodge") +
#   geom_errorbar(aes(x = school_fee_group, ymin = ci_lower, ymax = ci_upper), width = 0.2) +
#   theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
#   labs(title = "Age- and sex-adjusted ADHD prevalence",
```

```

#           x = "Economic status (grouped)",
#           y = "Sample prevalence")

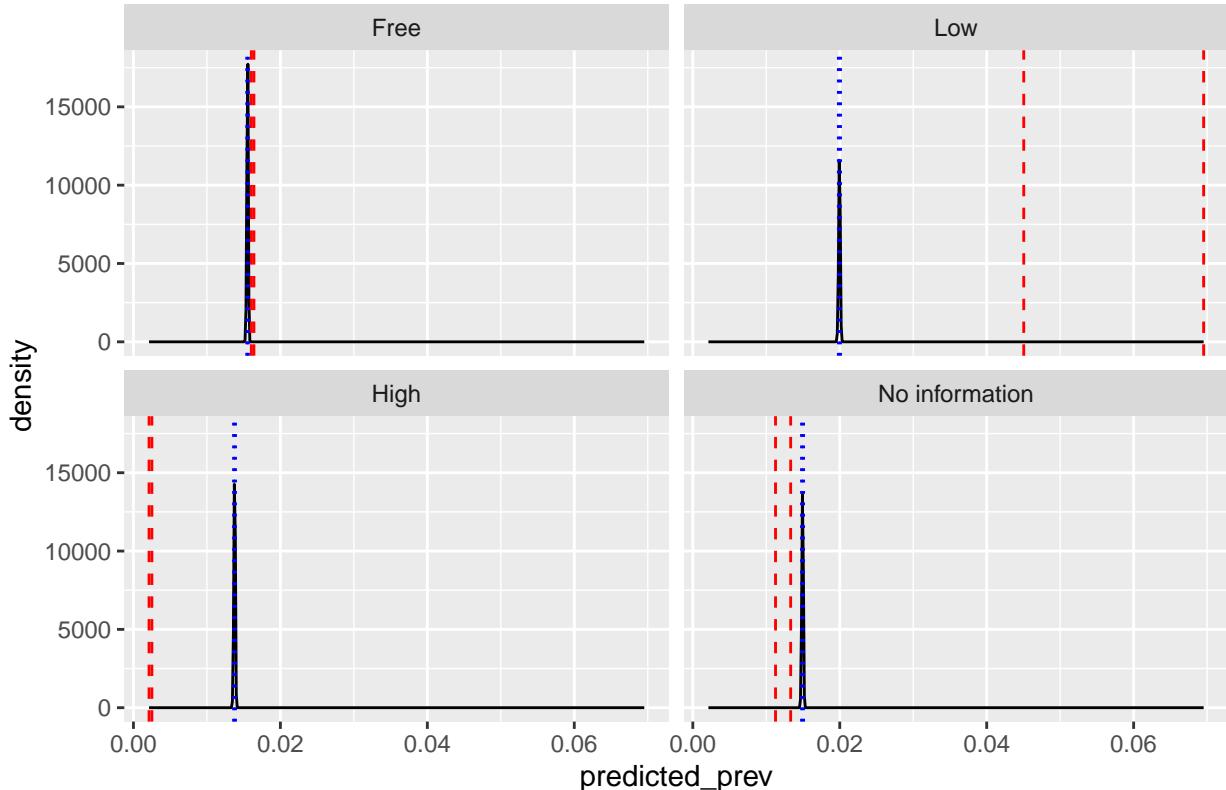
ADHD_prev_econG_post <- do_jags_rand_model(x = ADHD_prev_econG_adj,
                                             feat = "school_fee_group",
                                             model = rand_model,
                                             theta_mu = theta_mu_prior,
                       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

plot_post_density(ADHD_prev_econG_post, ADHD_prev_econG_adj, feat = "school_fee_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 = 0.015, prior sd = 7.25e-05



Predictions for higher population prevalence - increase prior mean

```

for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_econG_post <- do_jags_rand_model(x = ADHD_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) %>%

```

```

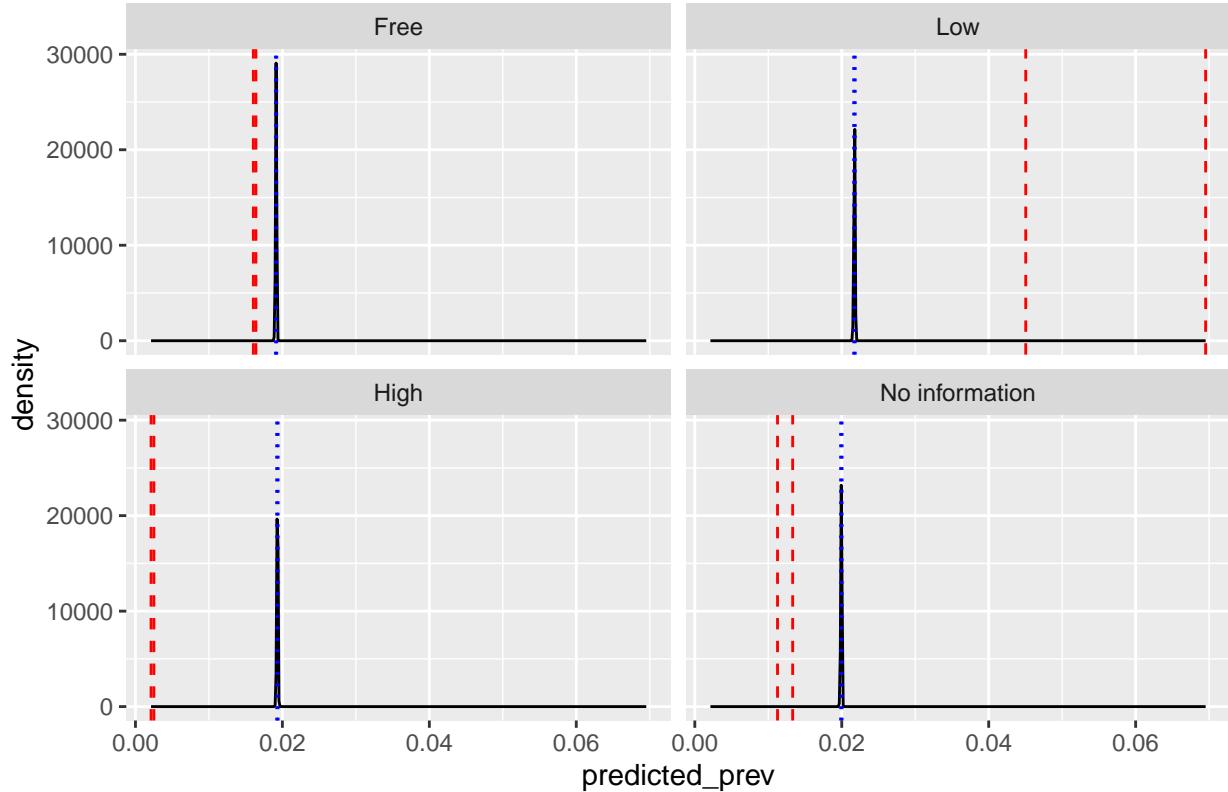
    rename("school_fee_group" = "Feat_names")

    plot_post_density(ADHD_prev_econG_post,
                      ADHD_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.

```

Prior mean = 0.02, prior sd = 5.1e-05

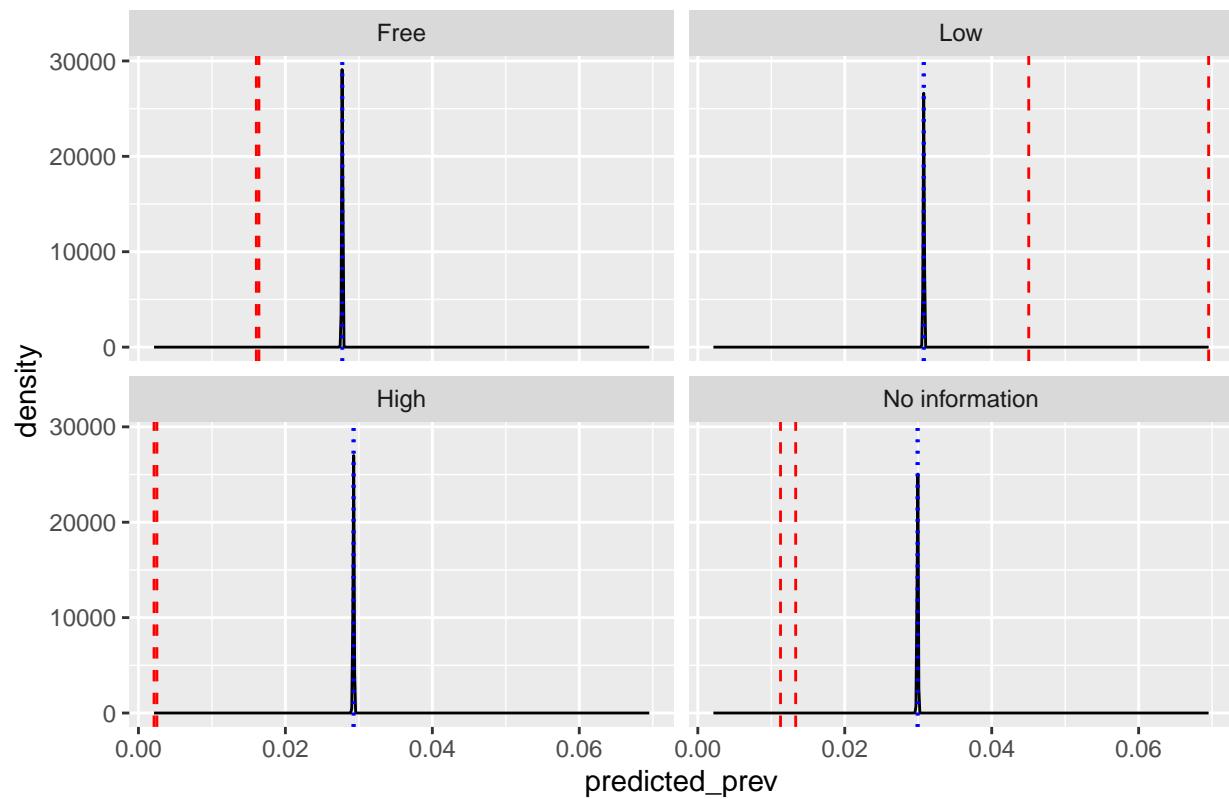


```

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

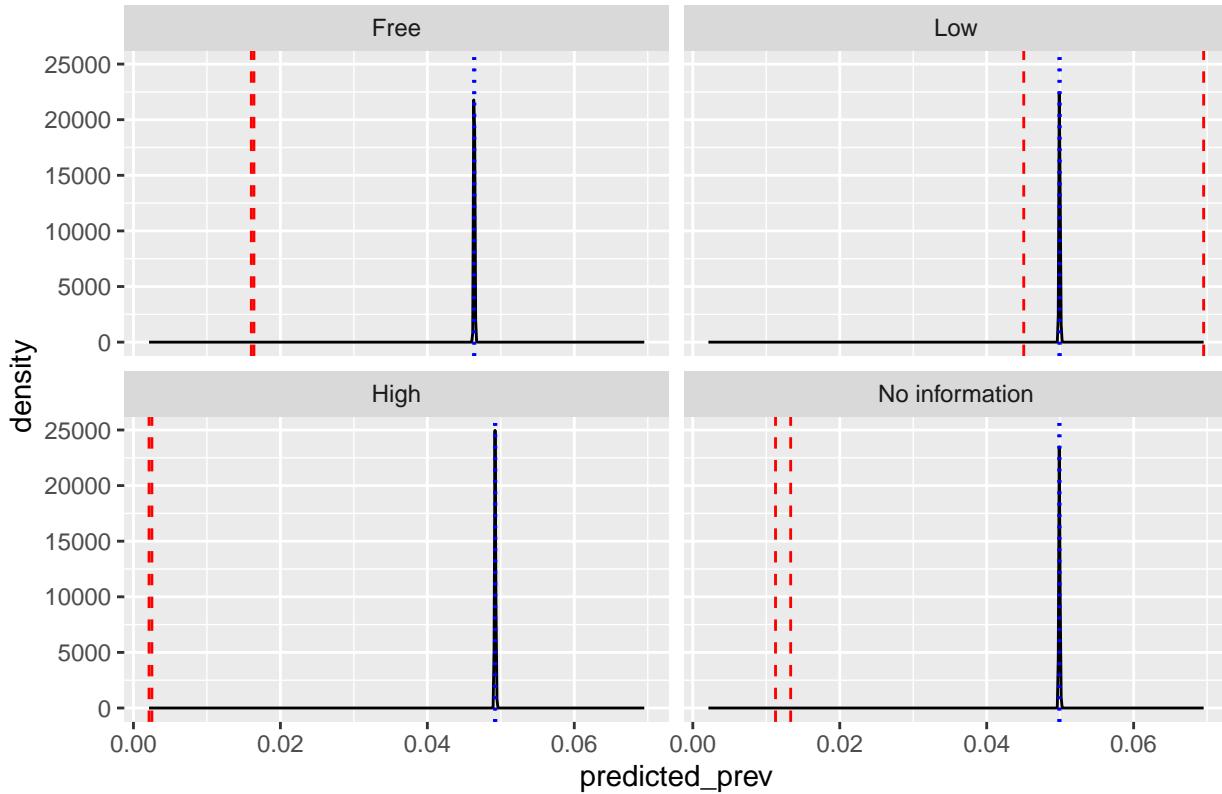
```

Prior mean = 0.03, 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 = 0.05, prior sd = 5.1e-05



Random effect on economic status, sexes separate

Use grouped fees because they ungrouped classes are too small and there are no females in the 1,000-10,000 group.

```
# Females
ADHD_prev_econG_f <- chile_bayes_ADHD %>%
  filter(sex == 2) %>%
  get_grouped_prev(stdpop = chile_stdpop_f, grouping_vars = c("school_fee_group", "age_june30", "sex"))

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

ADHD_prev_econG_adj_f <- get_adjusted_prev(ADHD_prev_econG_f, grouping_vars = "school_fee_group")

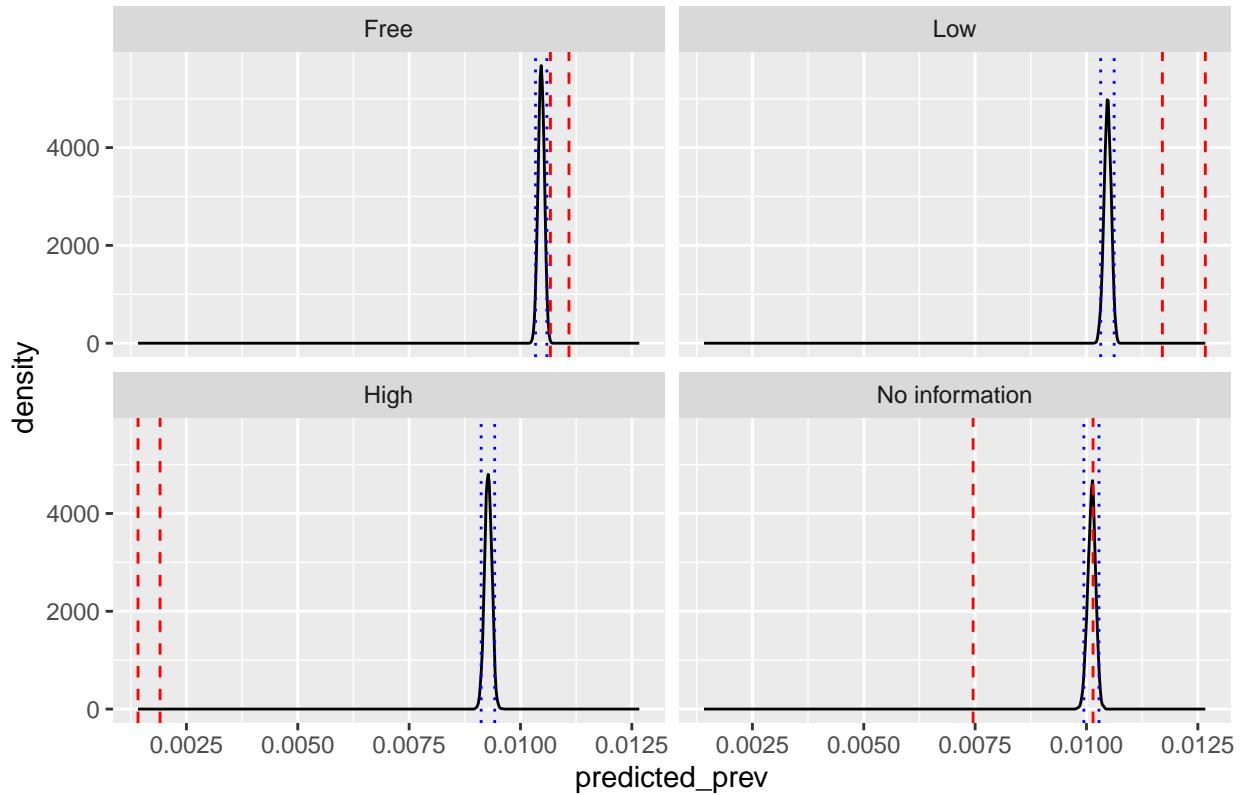
ADHD_prev_econG_post_f <- do_jags_rand_model(x = ADHD_prev_econG_adj_f,
                                              feat = "school_fee_group",
                                              model = rand_model,
                                              theta_mu = theta_mu_prior_f,
                                              theta_sigma = theta_sigma_prior_f,
                                              pars = pars,
                                              convergence_checks = FALSE) %>%
  rename("school_fee_group" = "Feat_names")

plot_post_density(ADHD_prev_econG_post_f, ADHD_prev_econG_adj_f, feat = "school_fee_group", theta_mu = -
```

Don't know how to automatically pick scale for object of type

```
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.0101, prior sd = 8.7e-05 (females)



```
# Males
```

```
ADHD_prev_econG_m <- chile_bayes_ADHD %>%
  filter(sex == 1) %>%
  get_grouped_prev(stdpop = chile_stdpop_m, grouping_vars = c("school_fee_group", "age_june30", "sex"))

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

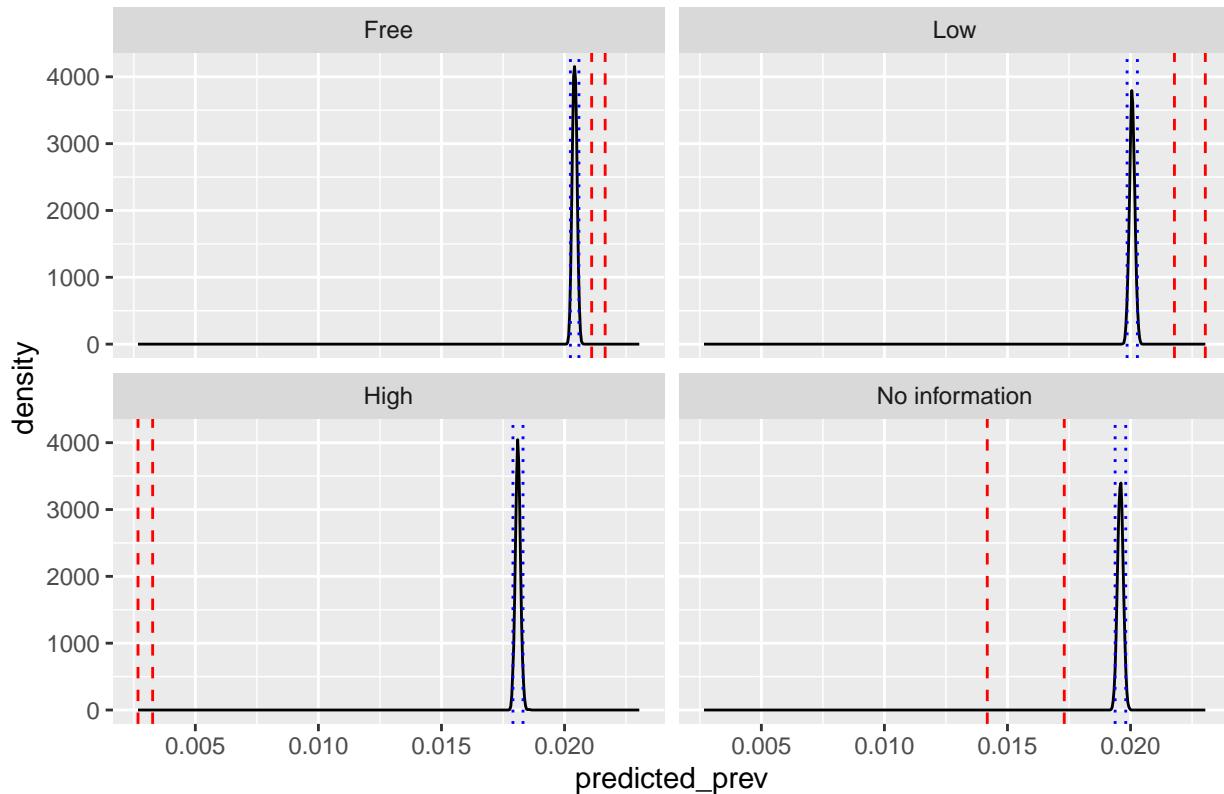
ADHD_prev_econG_adj_m <- get_adjusted_prev(ADHD_prev_econG_m, grouping_vars = "school_fee_group")

ADHD_prev_econG_post_m <- do_jags_rand_model(x = ADHD_prev_econG_adj_m,
                                              feat = "school_fee_group",
                                              model = rand_model,
                                              theta_mu = theta_mu_prior_m,
                                              theta_sigma = theta_sigma_prior_m,
                                              pars = pars,
                                              convergence_checks = FALSE) %>%
  rename("school_fee_group" = "Feat_names")

plot_post_density(ADHD_prev_econG_post_m, ADHD_prev_econG_adj_m, feat = "school_fee_group", theta_mu = ...)

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

Prior mean = 0.0197, prior sd = 0.000114 (males)



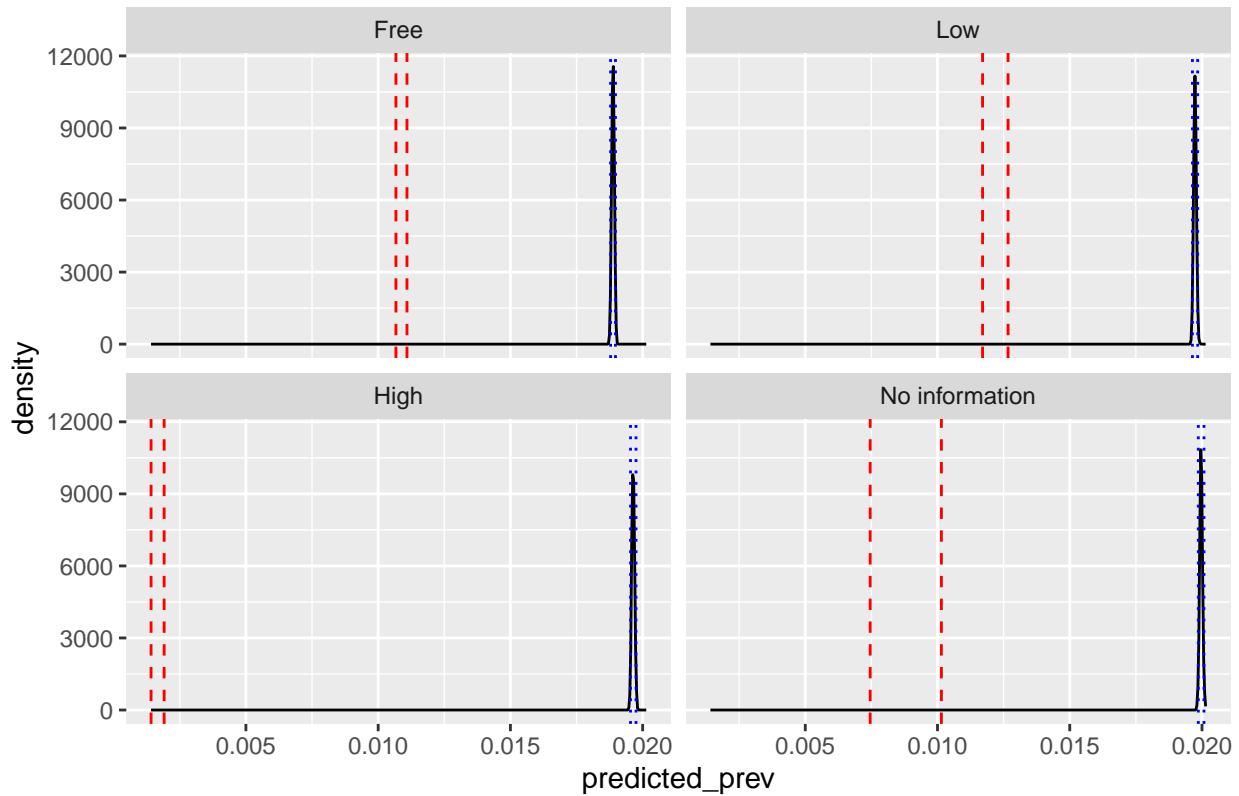
```
# Females
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_econG_post_f <- do_jags_rand_model(x = ADHD_prev_econG_adj_f,
                                                 feat = "school_fee_group",
                                                 model = rand_model,
                                                 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(ADHD_prev_econG_post_f,
                    ADHD_prev_econG_adj_f,
                    feat = "school_fee_group",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j],
                    title_text = " (females)")

}

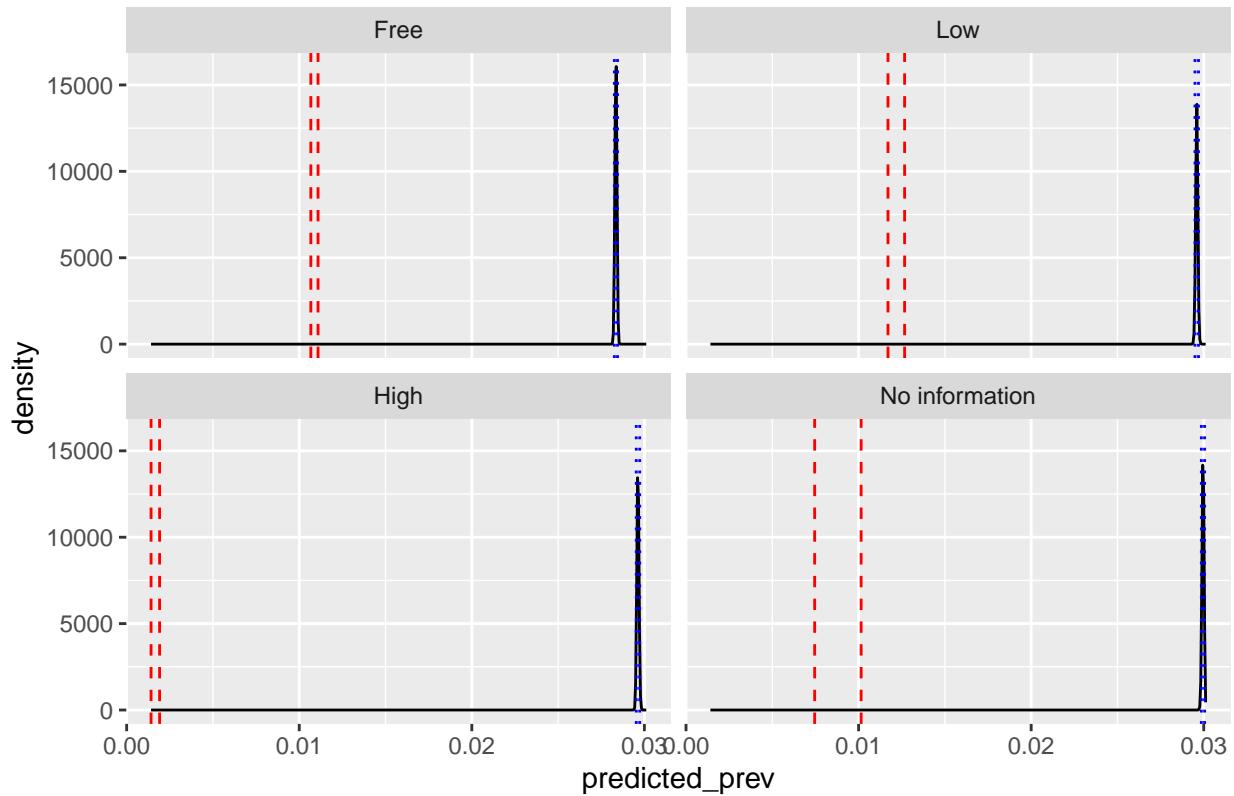
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05 (females)



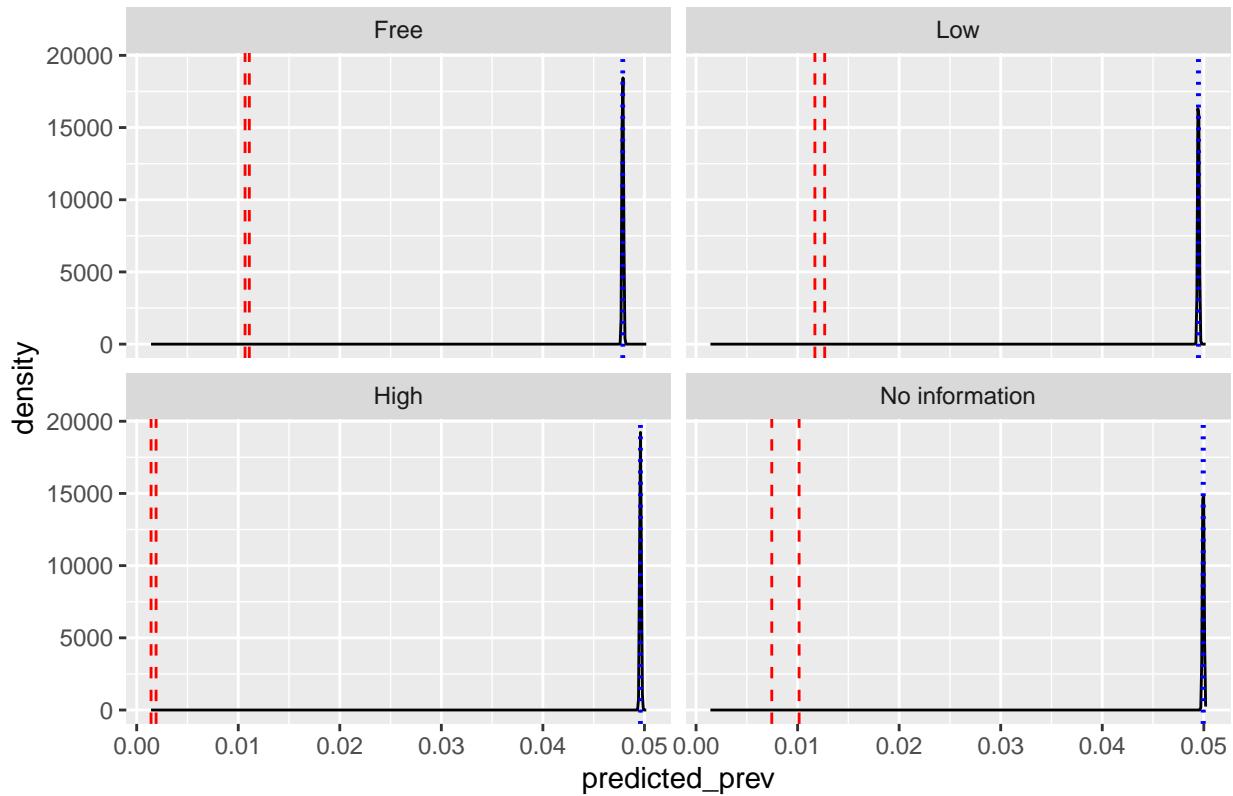
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, prior sd = 5.1e-05 (females)



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

Prior mean = 0.05, prior sd = 5.1e-05 (females)



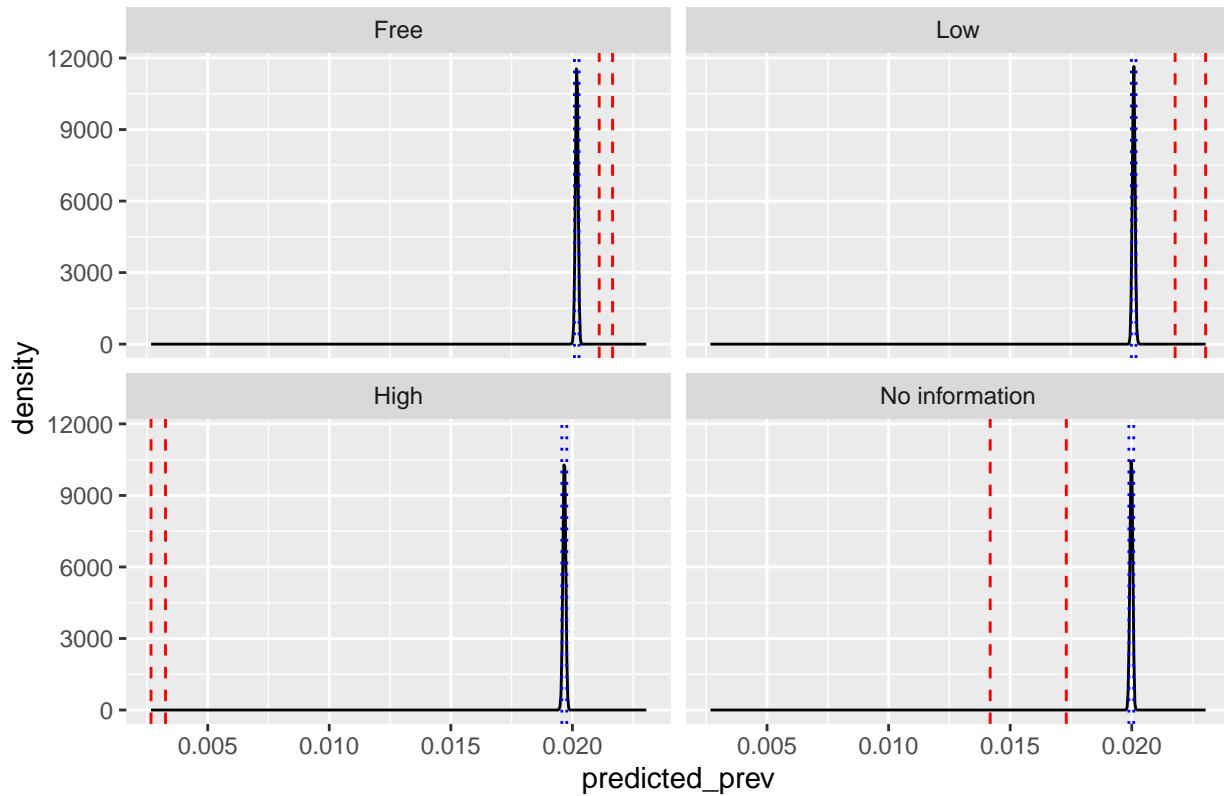
```
# Males
for(j in 1:length(theta_mu_extrapolate)) {
  ADHD_prev_econG_post_m <- do_jags_rand_model(x = ADHD_prev_econG_adj_m,
                                                feat = "school_fee_group",
                                                model = rand_model,
                                                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(ADHD_prev_econG_post_m,
                    ADHD_prev_econG_adj_m,
                    feat = "school_fee_group",
                    theta_mu = theta_mu_extrapolate[j],
                    theta_sigma = theta_sigma_extrapolate[j],
                    title_text = " (males)")

}

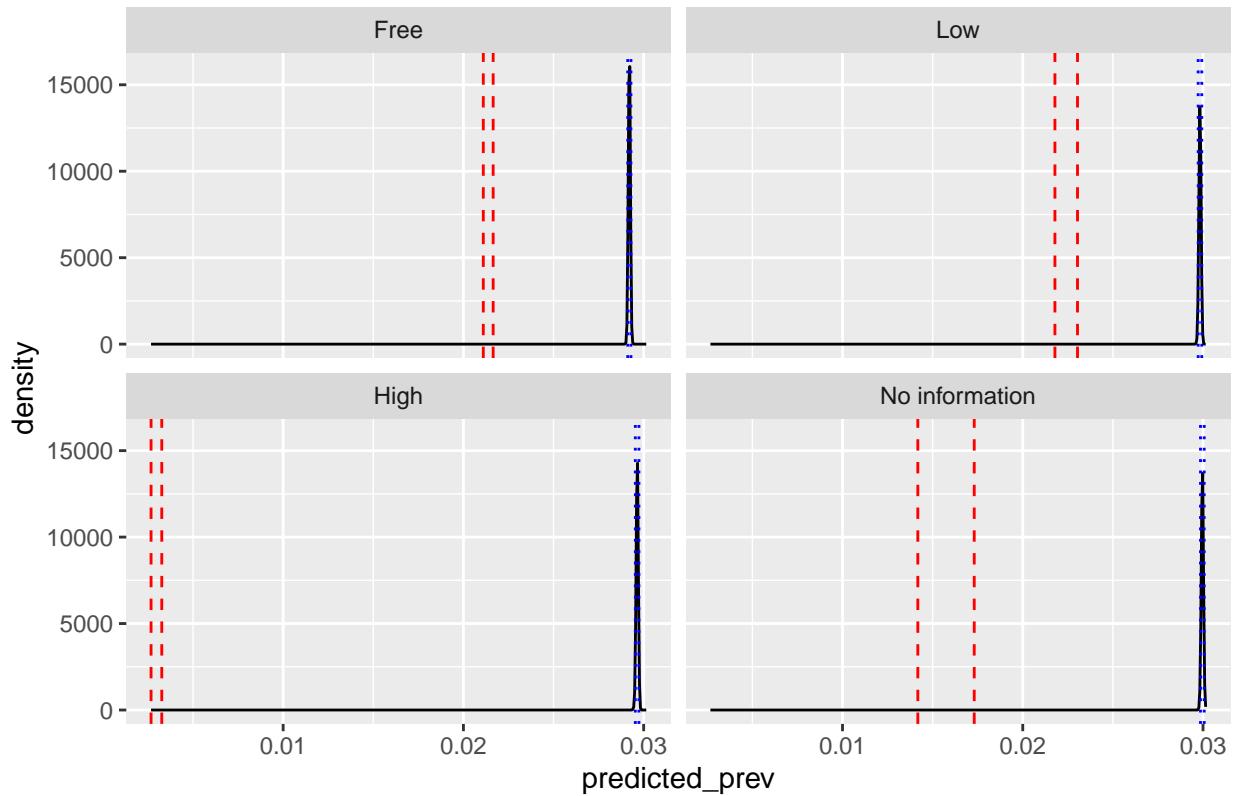
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.02, prior sd = 5.1e-05 (males)



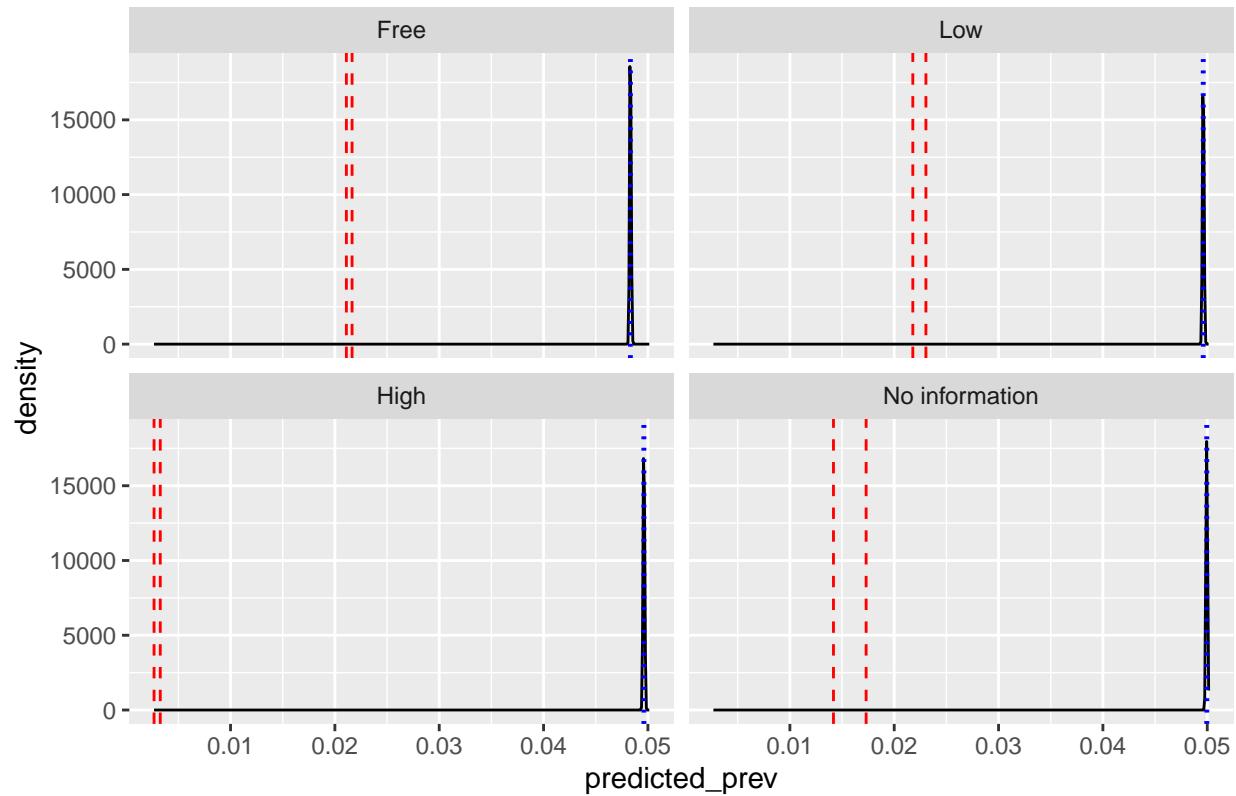
```
## Don't know how to automatically pick scale for object of type
## <draws_matrix/draws/matrix>. Defaulting to continuous.
```

Prior mean = 0.03, prior sd = 5.1e-05 (males)



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

Prior mean = 0.05, prior sd = 5.1e-05 (males)



Could do random effect on sex and region together

Rerun features that are most interesting using prevalence from clinical data, with sexes separate. Then graph prevalences and see if CI's overlap.