Analytic Examples

A Gentle Introduction to Bayesian Analysis with Applications to QuantCrit
ASHE Workshop

Alberto Guzman-Alvarez
Taylor Burtch
Benjamin Skinner

18 November 2023

In this part of the workshop, we'll work through a couple of examples of a Bayesian analysis. The first will be a very simple regression and the second a more complex regression with deep interactions. We'll also consider a few model checks and ways of presenting the results.

Libraries

We'll load a few libraries for our analysis:

- *tidyverse*: useful for data wrangling
- *brms*: our main Bayesian regression tool
- *bayesplot*: support library for plotting Bayesian results
- *tidybayes*: support library for wrangling/plotting Bayesian results
- *patchwork*: combine plots
- *shinystan*: interactive inspection of Bayesian objects
- *parallel*: take advantage of multiple cores

If you don't have these libraries, you can install them quickly with the following:

```r
install.packages(c("tidyverse", "brms", "bayesplot", "tidybayes", "patchwork", "shinystan", "parallel"), dependenc

## ----------------------------
## libraries
## ----------------------------

libs <- c("tidyverse", "brms", "bayesplot", "tidybayes", "patchwork","shinystan")
sapply(libs, require, character.only = TRUE)
```

```
## Loading required package: tidyverse

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Loading required package: brms
```

```
## 
## Loading required package: Rcpp
## 
## Loading 'brms' package (version 2.20.4). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
## 
## 
## Attaching package: 'brms'
## 
## 
## The following object is masked from 'package:stats':
## 
##     ar
## 
## 
## Loading required package: bayesplot
## 
## This is bayesplot version 1.10.0
## 
## - Online documentation and vignettes at mc-stan.org/bayesplot
## 
## - bayesplot theme set to bayesplot::theme_default()
## 
##     * Does _not_ affect other ggplot2 plots
## 
##     * See ?bayesplot_theme_set for details on theme setting
## 
## 
## Attaching package: 'bayesplot'
## 
## 
## The following object is masked from 'package:brms':
## 
##     rhat
## 
## 
## Loading required package: tidybayes
## 
## 
## Attaching package: 'tidybayes'
## 
## 
## The following objects are masked from 'package:brms':
## 
##     dstudent_t, pstudent_t, qstudent_t, rstudent_t
## 
## 
## Loading required package: patchwork
## 
## Loading required package: shinystan
## 
## Loading required package: shiny
## 
```

```
## 
## This is shinystan version 2.6.0

## tidyverse      brms bayesplot tidybayes patchwork shinystan
##      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
```

Settings

We have a couple of settings that will help us. First, we'll take advantage of our computers' multiple cores with `options(mc.cores=parallel::detectCores())`.

```
## ---------------------------
## settings
## ---------------------------


## set number of cores to use to speed things up
options(mc.cores=parallel::detectCores())


## set a seed so things stay the same
my_seed <- 20231118
```

```
## ---------------------------
## input
## ---------------------------


df <- readRDS("college.RDS")
```

```
## ---------------------------
## show data set
## ---------------------------


df
```

```
## # A tibble: 17,202 × 7
##        id gender raceeth birthyr pov185 region    college
##     <int> <chr>  <chr>     <int> <chr>  <chr>       <dbl>
##  1 10001 male   white      1995 above  midwest         1
##  2 10002 female white      1995 below  northeast       1
##  3 10003 female blackaa    1995 above  west            1
##  4 10004 female white      1995 above  south           0
##  5 10005 male   white      1995 above  south           0
##  6 10007 female white      1994 above  northeast       1
##  7 10008 male   white      1994 above  northeast       1
##  8 10009 male   white      1995 above  south           1
##  9 10012 female white      1995 above  midwest         1
## 10 10013 male   white      1995 above  south           1
## # ⏷  17,192 more rows
```

```
## -------------------------------------------------------------------------------
## simple regression: intercept only (average college-going rate)
## -------------------------------------------------------------------------------


## likelihood of going to college
fit <- brm(college ~ 1,
           data = df,
           family = bernoulli("logit"),
           seed = my_seed)
```

```r
## show summary stats
summary(fit)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: college ~ 1
##    Data: df (Number of observations: 17202)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.12      0.02     1.08     1.15 1.00     1534     1792
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
## show distribution of intercept (our main parameter)
mcmc_areas(fit, prob = 0.95, pars = "b_Intercept") +
  labs(
    title = "Posterior distribution (log scale)",
    subtitle = "with median and 95% interval"
  )
```

Posterior distribution (log scale)
with median and 95% interval

b_Intercept

1.05   1.10   1.15
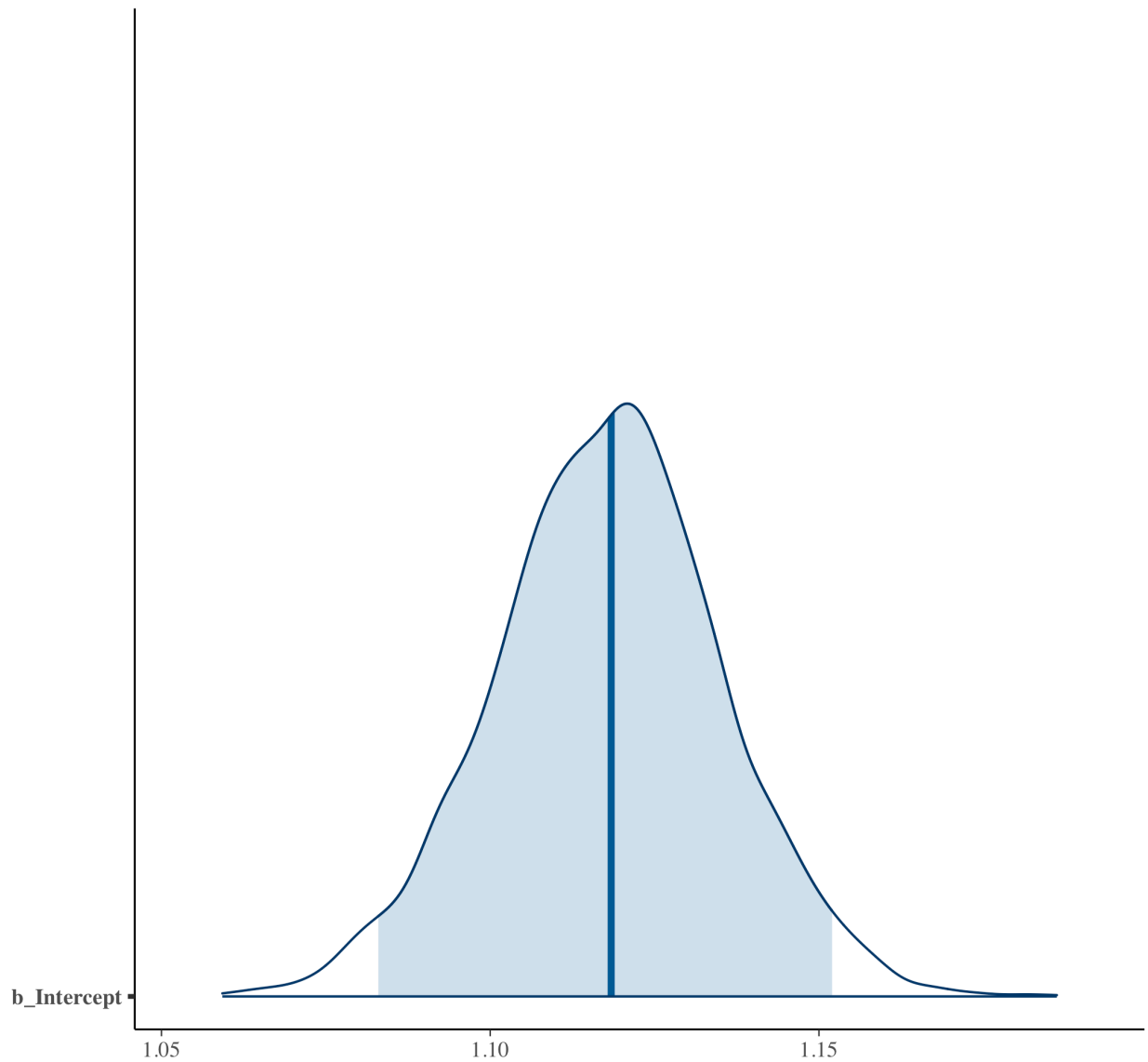
```r
## helper function: inverse logit
inv_logit <- function(x) { exp(-log(1 + exp(-x))) }

## show distribution of transformed intercept (our main parameter)
## using helper function
mcmc_areas(fit, prob = 0.95, pars = "b_Intercept",
           transformation = list("b_Intercept" = inv_logit)) +
  labs(
    title = "Posterior distribution (probability)",
    subtitle = "with median and 95% interval; prior: student_t(0,3,2.5)"
  )
```

Posterior distribution (probability)
with median and 95% interval; prior: student_t(0,3,2.5)

t(b_Intercept)

0.75   0.76
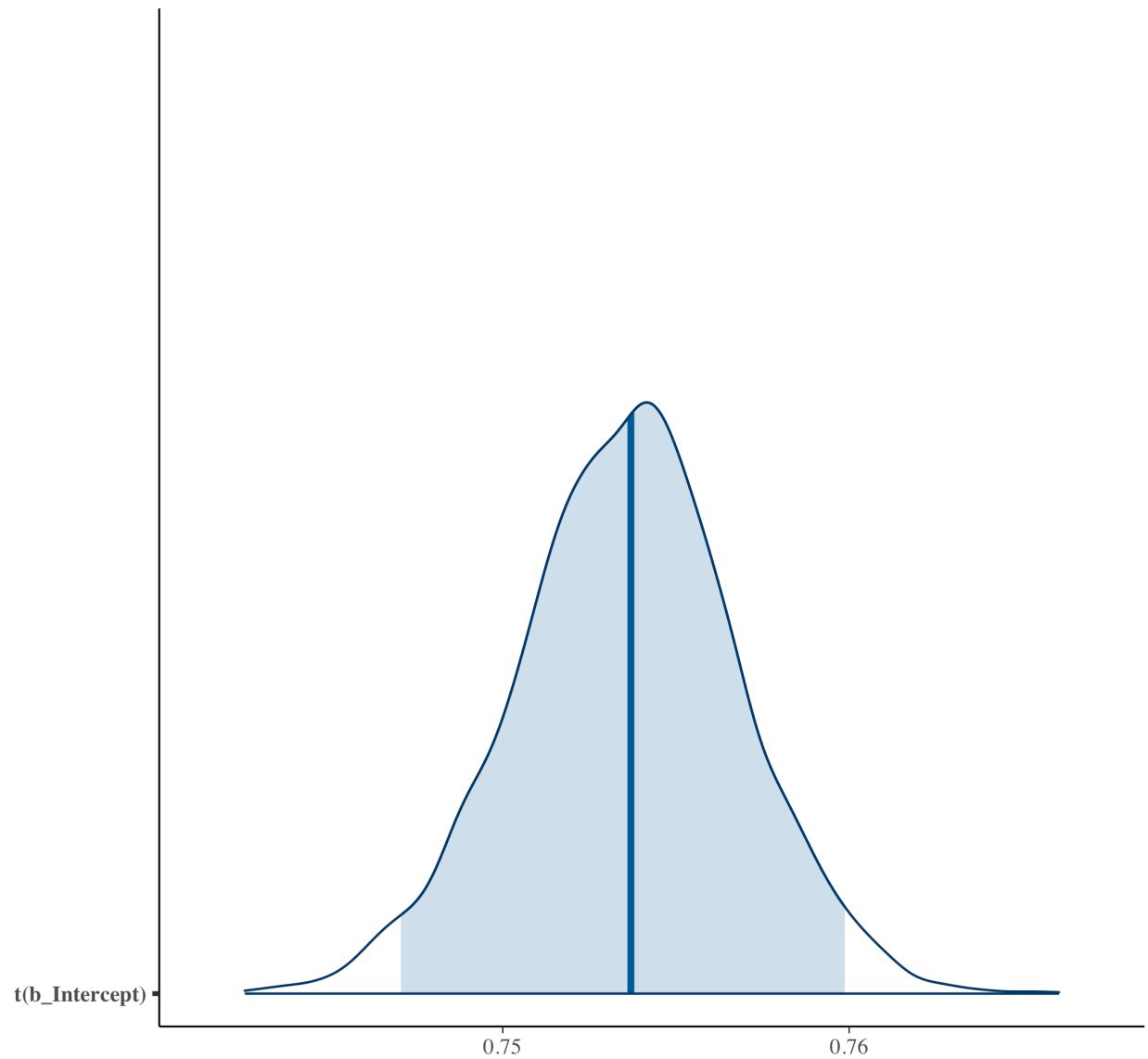
```
## ---------------------------
## check prior
## ---------------------------

## show prior from first model
prior_summary(fit)

## Intercept ~ student_t(3, 0, 2.5)

## change to normal prior for comparison
fit <- brm(college ~ 1,
           data = df,
           family = bernoulli("logit"),
           seed = my_seed,
           prior = set_prior("normal(0,20)", class = "Intercept"))
```

```
## Compiling Stan program...

## Start sampling
## check prior
prior_summary(fit)

## Intercept ~ normal(0,20)
## show summary stats
summary(fit)

##  Family: bernoulli
##   Links: mu = logit
## Formula: college ~ 1
##     Data: df (Number of observations: 17202)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.12      0.02     1.08     1.15 1.00     1498     2192
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
## show distribution of transformed intercept (our main parameter)
## using helper function
mcmc_areas(fit, prob = 0.95, pars = "b_Intercept",
           transformation = list("b_Intercept" = inv_logit)) +
  labs(
    title = "Posterior distribution (probability)",
    subtitle = "with median and 95% interval; prior: normal(0,20)",
  )
```
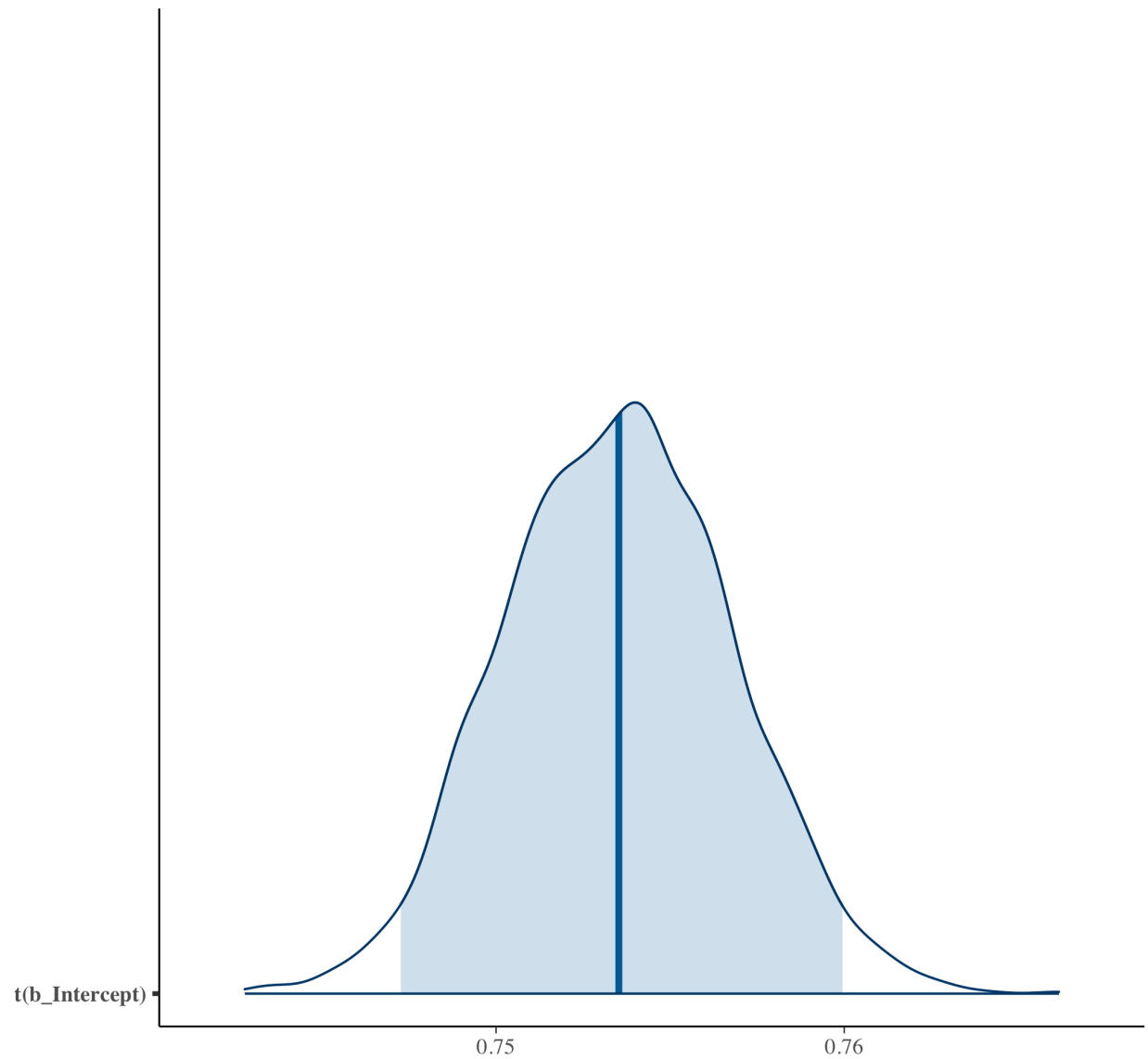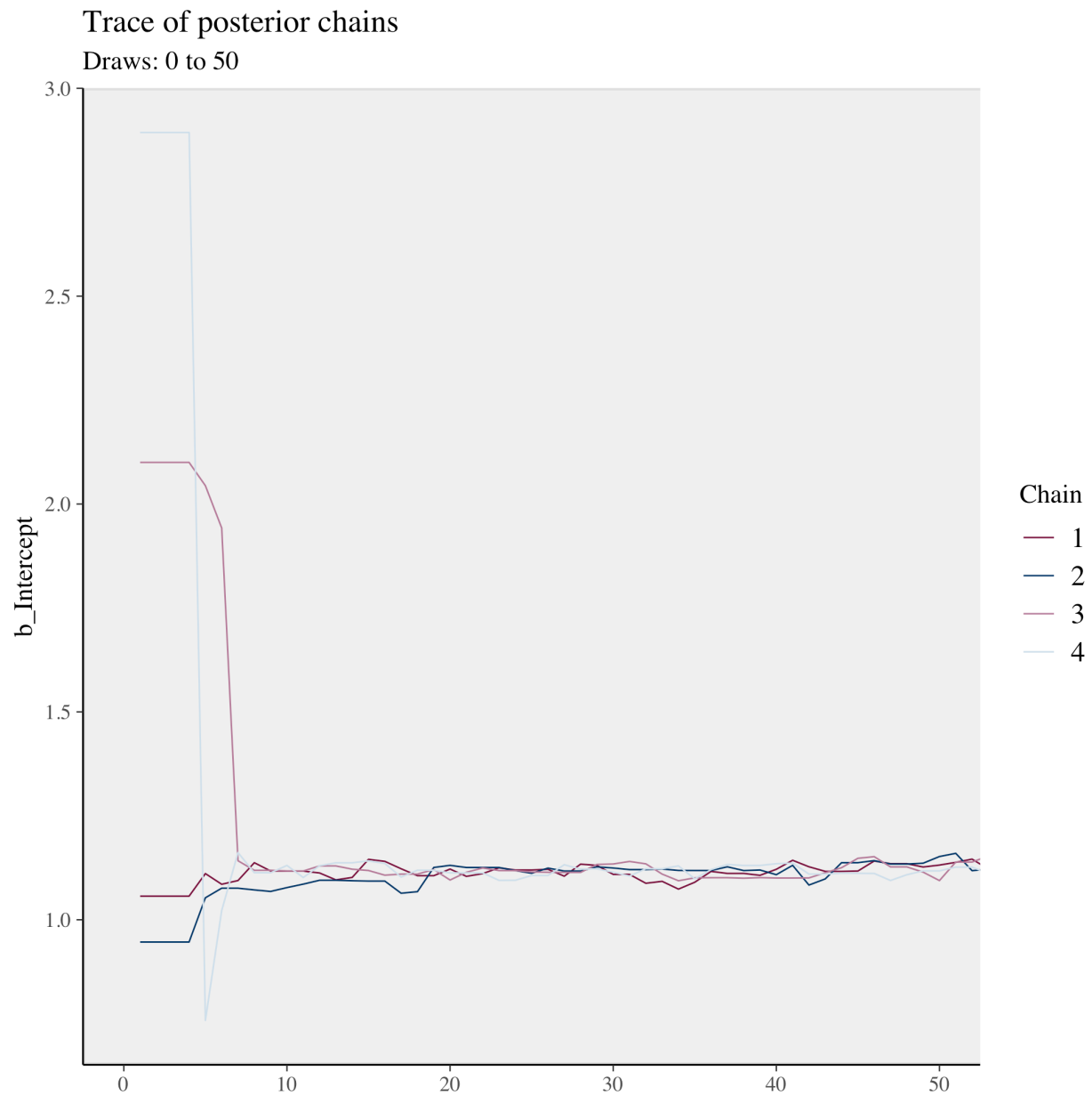
Posterior distribution (probability)
with median and 95% interval; prior: normal(0,20)

t(b_Intercept)

0.75   0.76

```r
## show trace of chains for intercept (our main parameter)
color_scheme_set("mix-blue-pink")
mcmc_trace(fit |> as_draws(inc_warmup = TRUE),
           pars = "b_Intercept", n_warmup = 1000,
           window = c(0, 50)) +
  labs(
    title = "Trace of posterior chains",
    subtitle = "Draws: 0 to 50"
  )
```

Trace of posterior chains
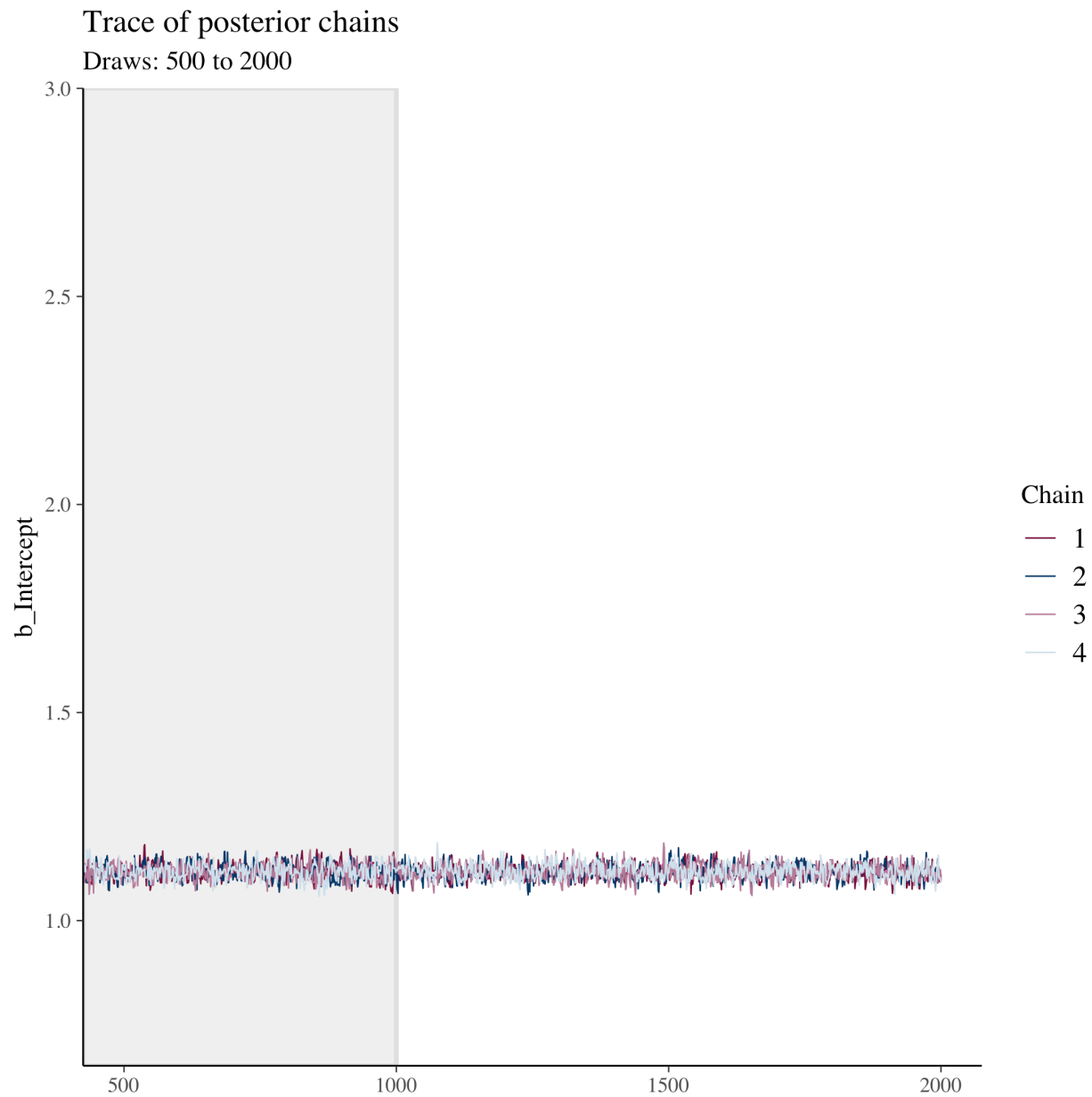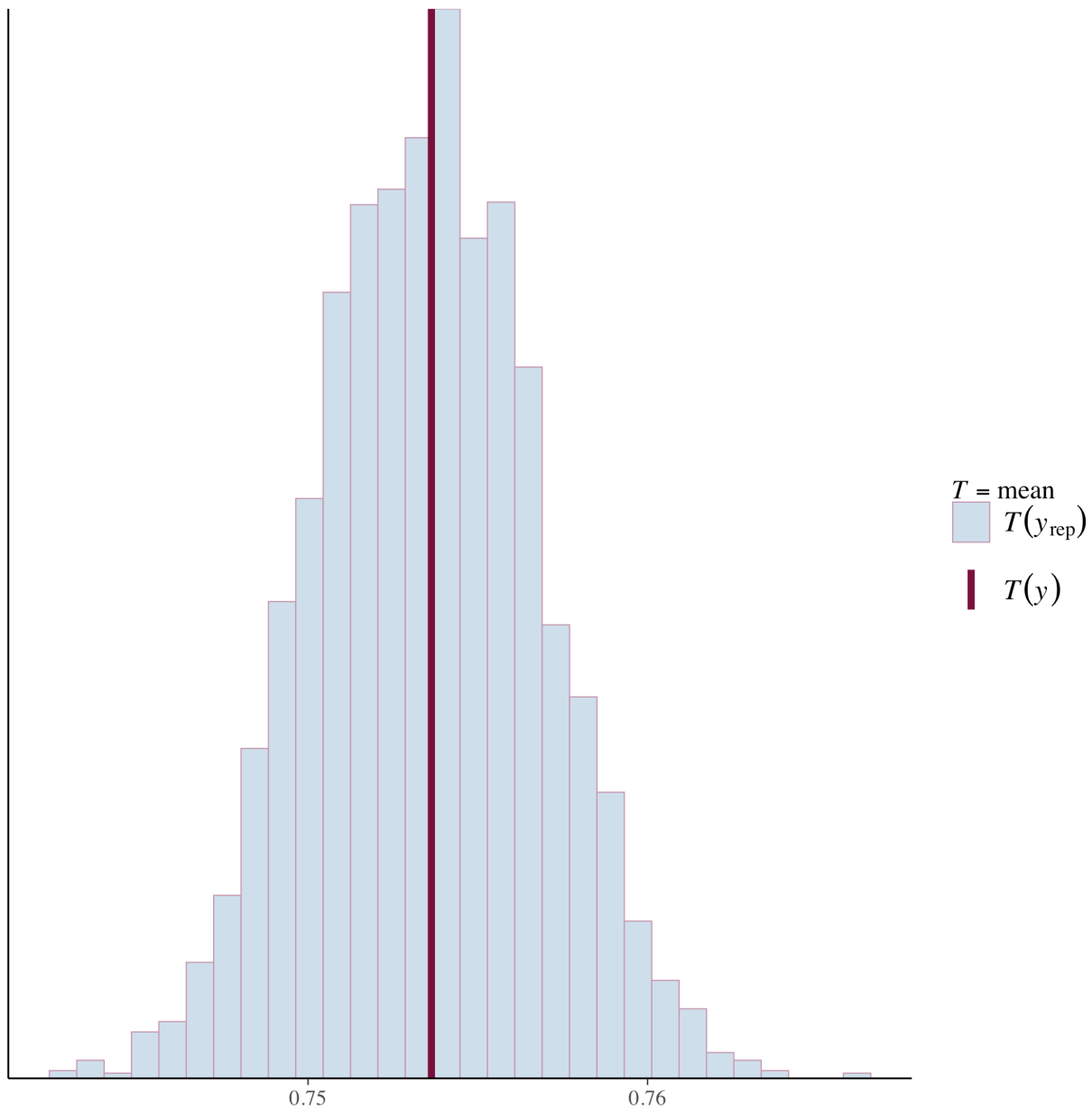Draws: 0 to 50


```r
## show trace of chains for intercept (our main parameter)
color_scheme_set("mix-blue-pink")
mcmc_trace(fit |> as_draws(inc_warmup = TRUE),
           pars = "b_Intercept", n_warmup = 1000,
           window = c(500, 2000)) +
  labs(
    title = "Trace of posterior chains",
    subtitle = "Draws: 500 to 2000"
  )
```

Trace of posterior chains
Draws: 500 to 2000


```
## ---------------------------
## posterior predictive check
## ---------------------------

## plot our posterior predictive values against the college-going rate
## that is observed in the data
ppc_stat(y = df |> pull(college) |> c(),
         yrep = posterior_epred(fit),
         stat = mean)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

The legend reads:

T = mean
$T(y_\text{rep})$
$T(y)$

```
## --------------------------------------------------------------------------
## speed up trick
## --------------------------------------------------------------------------

## since we're using categorical variables (rather than continuous variables),
## collapse binary data (bernoulli) into smaller data set of successes/trials
## (binomial) to take advantage of sufficient statistics
df_tmp <- df |>
  summarise(college = sum(college),
            n = n())

## likelihood of going to college using binomial
fit <- brm(college | trials(n) ~ 1,
           data = df_tmp,
```

```
        family = binomial("logit"),
        seed = my_seed)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
##  Family: binomial
##   Links: mu = logit
## Formula: college | trials(n) ~ 1
##    Data: df_tmp (Number of observations: 1)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.12      0.02     1.08     1.15 1.00     1533     2023
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
## -------------------------------------------------------------------------------
## multiple regression across groups
## -------------------------------------------------------------------------------
```

```
## collapse into groups of race/ethnicity by gender by poverty level
df_tmp <- df |>
  group_by(raceeth, gender, pov185) |>
  summarise(college = sum(college),
            n = n(),
            .groups = "drop")
```

```
## likelihood of going to college using binomial
fit <- brm(college | trials(n) ~ raceeth + gender + pov185 +
             (1 | raceeth:gender:pov185),
           data = df_tmp,
           family = binomial("logit"),
           seed = my_seed)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
## show summary stats
summary(fit)
```

```
##  Family: binomial
##   Links: mu = logit
## Formula: college | trials(n) ~ raceeth + gender + pov185 + (1 | raceeth:gender:pov185)
##    Data: df_tmp (Number of observations: 52)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
```

```
## Group-Level Effects:
## ~raceeth:gender:pov185 (Number of levels: 52)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.11      0.04     0.04     0.18 1.00      984      649
##
## Population-Level Effects:
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept         1.46      0.21     1.03     1.89 1.01      717     1389
## raceethasian      1.75      0.23     1.30     2.21 1.00      785     1665
## raceethblackaa    0.47      0.22     0.05     0.91 1.00      769     1481
## raceethhispnr     0.40      0.25    -0.09     0.91 1.00      913     1866
## raceethhisprs     0.48      0.22     0.05     0.91 1.01      697     1508
## raceethmoretor    0.59      0.22     0.15     1.03 1.00      759     1412
## raceethnhpi       0.27      0.35    -0.41     0.99 1.00     1379     2329
## raceethunknown    1.00      0.25     0.52     1.48 1.00      904     1684
## raceethwhite      0.50      0.22     0.07     0.93 1.01      702     1532
## gendermale       -0.48      0.06    -0.59    -0.37 1.00     2653     2620
## pov185below      -1.21      0.08    -1.35    -1.06 1.00     1992     2762
## pov185unknown    -1.30      0.08    -1.44    -1.14 1.00     1869     2048
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
## ----------------------------
## posterior predictions
## ----------------------------

## create a design matrix (data frame) of all possible groups in our model
df_design <- expand.grid(raceeth = df |> distinct(raceeth) |> pull() |> c(),
                         gender = df |> distinct(gender) |> pull() |> c(),
                         pov185 = df |> distinct(pov185) |> pull() |> c(),
                         stringsAsFactors = FALSE) |>
  as_tibble() |>
  arrange(raceeth, gender, pov185) |>
  mutate(n = 100,
         group = paste(raceeth, gender, pov185, sep = "_"))

## get posterior predictions but in long form that's better for plotting
pp <- df_design |>
  add_epred_draws(fit,
                  ndraws = 500,
                  allow_new_levels = TRUE)

## compute mean posterior by group to get order for plot
pp_mean <- pp |>
  summarise(pp_mean = mean(.epred),
            .groups = "drop") |>
  arrange(pp_mean) |>
  mutate(plot_index = row_number(),
         plot_index = factor(plot_index,
                             levels = plot_index,
                             labels = group)) |>
  select(group, pp_mean, plot_index)
```

```r
## join means back to main pp tibble and plot densities for each group
bayes_g <- pp |>
  left_join(pp_mean, by = "group") |>
  ggplot(aes(y = plot_index, x = .epred)) +
  stat_pointinterval(.width = 0.95, linewidth = 0.7, size = 1) +
  scale_x_continuous(breaks = seq(0, 100, 10),
                     minor_breaks = seq(0, 100, 5)) +
  labs(
    title = "Posterior predictive distributions of college enrollment",
    y = "Group: race/ethnicity + gender + poverty status (185%)",
    x = "Rate of college attendance"
  ) +
  theme_bw()
bayes_g
```

Posterior predictive distributions of college enrollment
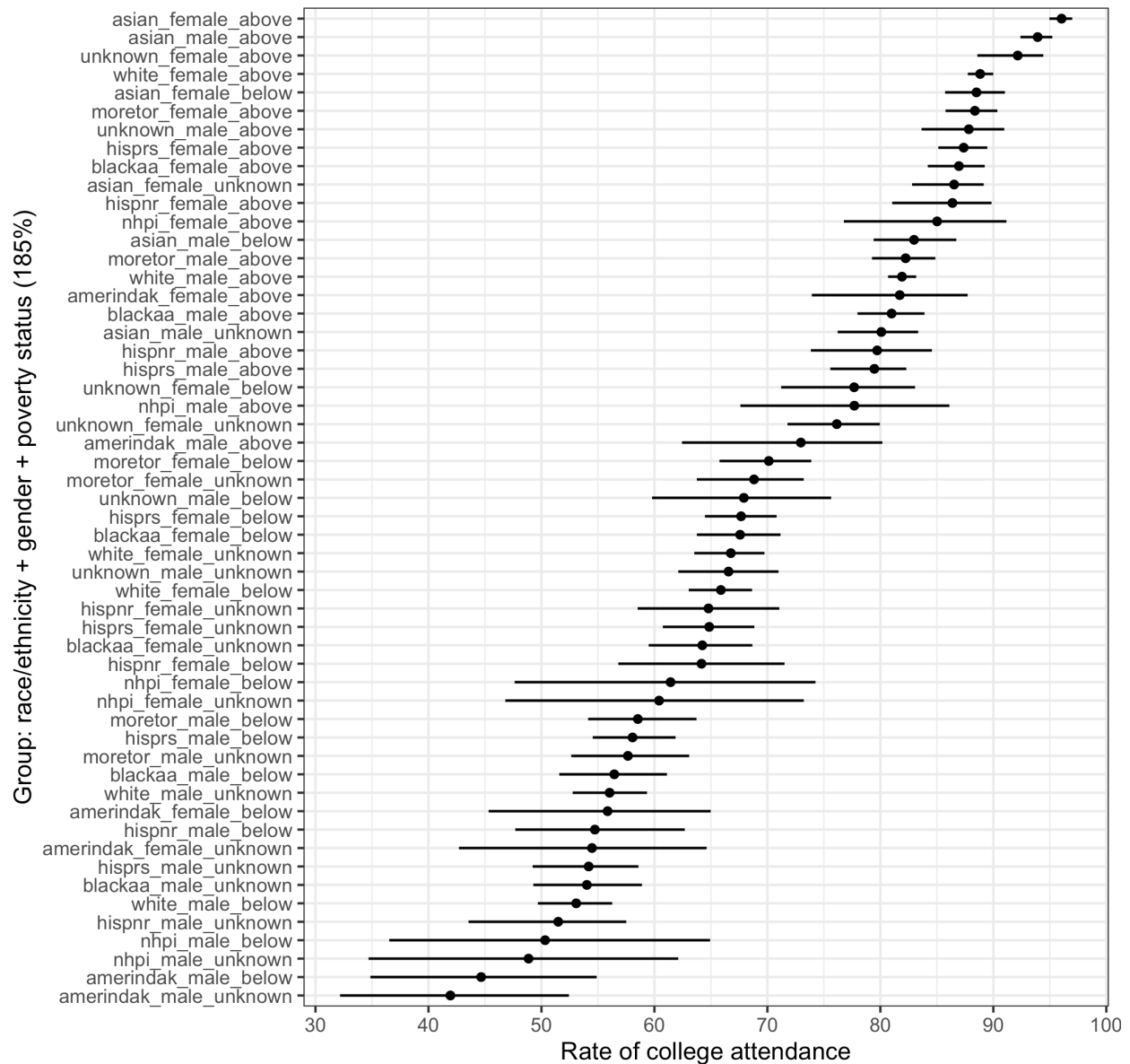

```
## ------------------------------------------------------------------------------
## quick comparison to frequentist approach
## ------------------------------------------------------------------------------

## fit logit model
lm_fit <- glm(cbind(college, n - college) ~ raceeth * gender * pov185,
              data = df_tmp,
              family = binomial("logit"))

## show summary
summary(lm_fit)

##
## Call:
```

```
## glm(formula = cbind(college, n - college) ~ raceeth * gender *
##     pov185, family = binomial("logit"), data = df_tmp)
##
## Coefficients: (2 not defined because of singularities)
##                                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                                   2.079e+00  7.500e-01   2.773  0.00556
## raceethasian                                  1.048e+00  7.923e-01   1.323  0.18598
## raceethblackaa                               -2.652e-01  7.674e-01  -0.346  0.72965
## raceethhispnr                                -9.808e-01  1.377e+00  -0.712  0.47625
## raceethhisprs                                -1.215e-01  7.628e-01  -0.159  0.87345
## raceethmoretor                               -1.515e-01  7.657e-01  -0.198  0.84311
## raceethnhpi                                   6.931e-01  1.275e+00   0.544  0.58661
## raceethunknown                               2.182e+01  6.648e+04   0.000  0.99974
## raceethwhite                                  3.918e-02  7.524e-01   0.052  0.95847
## gendermale                                   -1.192e+00  8.742e-01  -1.364  0.17265
## pov185below                                  -2.166e+00  8.583e-01  -2.524  0.01160
## pov185unknown                                -1.520e+00  9.774e-01  -1.555  0.11996
## raceethasian:gendermale                       8.133e-01  9.381e-01   0.867  0.38596
## raceethblackaa:gendermale                     8.365e-01  9.006e-01   0.929  0.35299
## raceethhispnr:gendermale                      1.480e+00  1.650e+00   0.897  0.36979
## raceethhisprs:gendermale                      4.404e-01  8.932e-01   0.493  0.62200
## raceethmoretor:gendermale                     7.120e-01  8.980e-01   0.793  0.42785
## raceethnhpi:gendermale                        2.654e-01  1.487e+00   0.178  0.85841
## raceethunknown:gendermale                     7.526e-01  8.025e-01   0.938  0.34831
## raceethwhite:gendermale                       5.916e-01  8.775e-01   0.674  0.50022
## raceethasian:pov185below                      1.409e+00  9.413e-01   1.496  0.13453
## raceethblackaa:pov185below                    1.123e+00  8.815e-01   1.274  0.20261
## raceethhispnr:pov185below                     1.132e+00  1.483e+00   0.764  0.44512
## raceethhisprs:pov185below                     9.822e-01  8.745e-01   1.123  0.26142
## raceethmoretor:pov185below                    1.122e+00  8.851e-01   1.268  0.20490
## raceethnhpi:pov185below                      -1.453e+00  1.508e+00  -0.964  0.33528
## raceethunknown:pov185below                   -4.614e+01  1.036e+05   0.000  0.99964
## raceethwhite:pov185below                      6.669e-01  8.632e-01   0.773  0.43980
## raceethasian:pov185unknown                    7.653e-03  1.027e+00   0.007  0.99405
## raceethblackaa:pov185unknown                  2.307e-01  9.991e-01   0.231  0.81735
## raceethhispnr:pov185unknown                   1.250e+00  1.528e+00   0.818  0.41338
## raceethhisprs:pov185unknown                   1.244e-01  9.953e-01   0.125  0.90053
## raceethmoretor:pov185unknown                  4.557e-01  1.006e+00   0.453  0.65060
## raceethnhpi:pov185unknown                    -7.419e-01  1.597e+00  -0.465  0.64228
## raceethunknown:pov185unknown                 -2.125e+01  6.648e+04   0.000  0.99974
## raceethwhite:pov185unknown                    1.104e-01  9.823e-01   0.112  0.91055
## gendermale:pov185below                        1.279e+00  1.058e+00   1.209  0.22683
## gendermale:pov185unknown                      1.348e-02  1.173e+00   0.011  0.99083
## raceethasian:gendermale:pov185below          -1.368e+00  1.174e+00  -1.165  0.24391
## raceethblackaa:gendermale:pov185below        -1.401e+00  1.093e+00  -1.281  0.20012
## raceethhispnr:gendermale:pov185below         -1.161e+00  1.821e+00  -0.638  0.52363
## raceethhisprs:gendermale:pov185below         -9.009e-01  1.082e+00  -0.832  0.40522
## raceethmoretor:gendermale:pov185below        -1.334e+00  1.100e+00  -1.213  0.22496
## raceethnhpi:gendermale:pov185below            4.949e-01  1.856e+00   0.267  0.78974
## raceethunknown:gendermale:pov185below               NA         NA      NA       NA
## raceethwhite:gendermale:pov185below          -1.245e+00  1.066e+00  -1.168  0.24290
## raceethasian:gendermale:pov185unknown         4.412e-02  1.246e+00   0.035  0.97175
## raceethblackaa:gendermale:pov185unknown      -5.135e-03  1.207e+00  -0.004  0.99661
## raceethhispnr:gendermale:pov185unknown       -1.221e+00  1.851e+00  -0.659  0.50959
```

```
## raceethhisprs:gendermale:pov185unknown     3.401e-01  1.201e+00   0.283  0.77710
## raceethmoretor:gendermale:pov185unknown     2.999e-03  1.216e+00   0.002  0.99803
## raceethnhpi:gendermale:pov185unknown       -2.907e-01  2.204e+00  -0.132  0.89508
## raceethunknown:gendermale:pov185unknown            NA         NA      NA       NA
## raceethwhite:gendermale:pov185unknown        1.494e-01  1.181e+00   0.127  0.89931
##
## (Intercept)                               **
## raceethasian
## raceethblackaa
## raceethhispnr
## raceethhisprs
## raceethmoretor
## raceethnhpi
## raceethunknown
## raceethwhite
## gendermale
## pov185below                               *
## pov185unknown
## raceethasian:gendermale
## raceethblackaa:gendermale
## raceethhispnr:gendermale
## raceethhisprs:gendermale
## raceethmoretor:gendermale
## raceethnhpi:gendermale
## raceethunknown:gendermale
## raceethwhite:gendermale
## raceethasian:pov185below
## raceethblackaa:pov185below
## raceethhispnr:pov185below
## raceethhisprs:pov185below
## raceethmoretor:pov185below
## raceethnhpi:pov185below
## raceethunknown:pov185below
## raceethwhite:pov185below
## raceethasian:pov185unknown
## raceethblackaa:pov185unknown
## raceethhispnr:pov185unknown
## raceethhisprs:pov185unknown
## raceethmoretor:pov185unknown
## raceethnhpi:pov185unknown
## raceethunknown:pov185unknown
## raceethwhite:pov185unknown
## gendermale:pov185below
## gendermale:pov185unknown
## raceethasian:gendermale:pov185below
## raceethblackaa:gendermale:pov185below
## raceethhispnr:gendermale:pov185below
## raceethhisprs:gendermale:pov185below
## raceethmoretor:gendermale:pov185below
## raceethnhpi:gendermale:pov185below
## raceethunknown:gendermale:pov185below
## raceethwhite:gendermale:pov185below
## raceethasian:gendermale:pov185unknown
## raceethblackaa:gendermale:pov185unknown
```
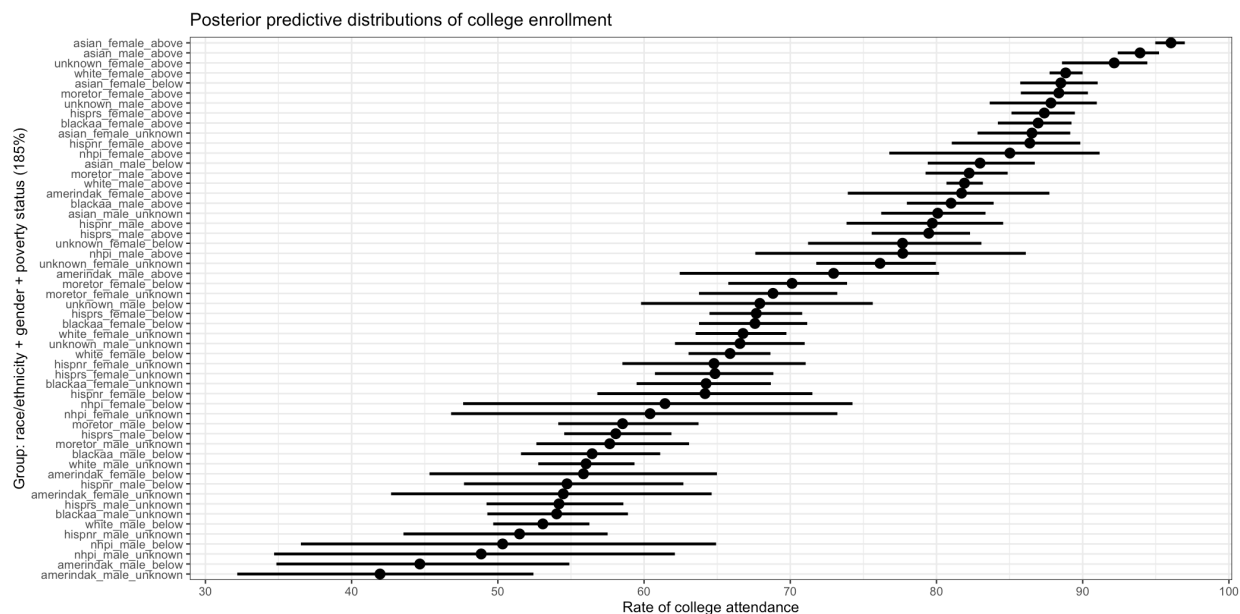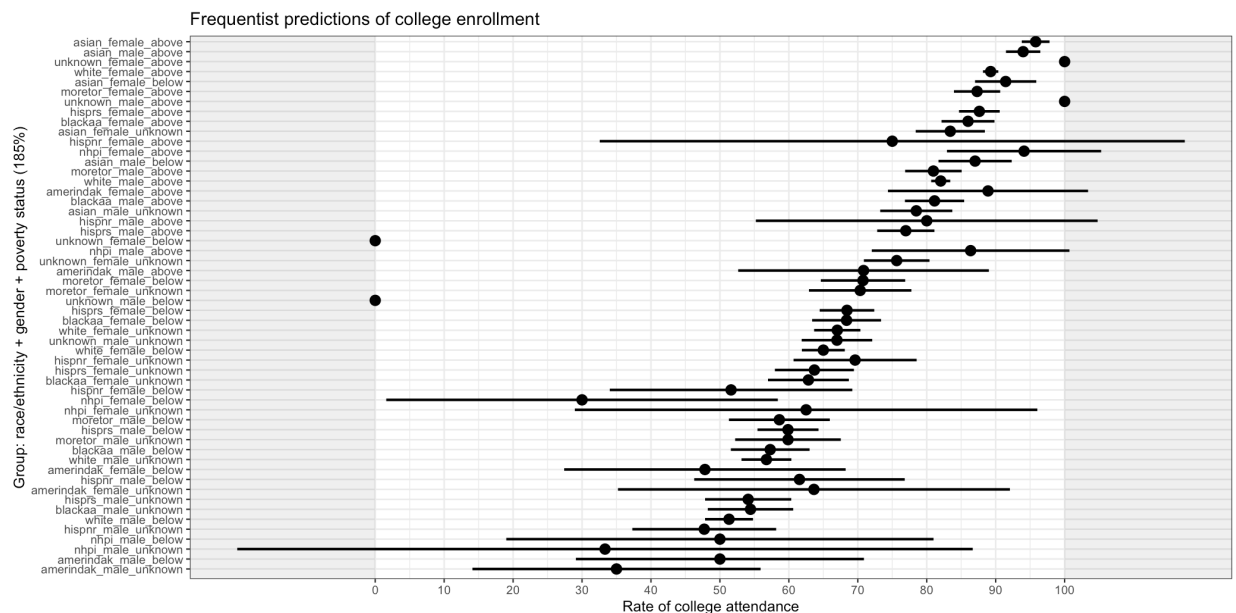
```
## raceethhispnr:gendermale:pov185unknown
## raceethhisprs:gendermale:pov185unknown
## raceethmoretor:gendermale:pov185unknown
## raceethnhpi:gendermale:pov185unknown
## raceethunknown:gendermale:pov185unknown
## raceethwhite:gendermale:pov185unknown
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.6327e+03  on 51  degrees of freedom
## Residual deviance: 2.8359e-10  on  0  degrees of freedom
## AIC: 350.1
##
## Number of Fisher Scoring iterations: 22
```

```r
## generage response predictions (meaning transform to probability scale)
lm_pred <- predict(lm_fit,
                   newdata = df_design,
                   se.fit = TRUE,
                   type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = residual.scale, type =
## if (type == : prediction from rank-deficient fit; attr(*, "non-estim") has
## doubtful cases
```

```r
## wrangle data and join plot_index from Bayes plot so everything aligns; plot
## means and 95 CIs to match prior plot
freq_g <- tibble(group = df_design$group,
                 pred = lm_pred$fit * 100,
                 se = lm_pred$se.fit * 100) |>
  mutate(ci95lo = pred + se * qnorm(0.025),
         ci95hi = pred + se * qnorm(0.975)) |>
  left_join(pp_mean, by = "group") |>
  ggplot(aes(y = plot_index, x = pred)) +
  geom_linerange(aes(xmin = ci95lo, xmax = ci95hi)) +
  geom_point(aes(x = pred)) +
  scale_x_continuous(breaks = seq(0, 100, 10),
                     minor_breaks = seq(0, 100, 5)) +
  annotate("rect", xmin = -Inf, xmax = 0, ymin = 0, ymax = Inf, alpha = 0.1) +
  annotate("rect", xmin = 100, xmax = Inf, ymin = 0, ymax = Inf, alpha = 0.1) +
  labs(
    title = "Frequentist predictions of college enrollment",
    y = "Group: race/ethnicity + gender + poverty status (185%)",
    x = "Rate of college attendance"
  ) +
  theme_bw()

## use patchwork to compare figures
freq_g / bayes_g &
  theme_bw(base_size = 6)
```

Frequentist predictions of college enrollment

Posterior predictive distributions of college enrollment

```
## ------------------------------------------------------------------------------
## comparison
## ------------------------------------------------------------------------------

## filter to two group comparison
pp_comp <- pp |>
  filter(group %in% c("white_male_above", "white_male_below"))

## plot two groups to make comparison clearer
comp_g <- pp_comp |>
  ggplot(aes(y = group, x = .epred)) +
  stat_pointinterval(.width = 0.95, linewidth = 0.7, size = 1) +
  scale_x_continuous(breaks = seq(0, 100, 5),
                     minor_breaks = seq(0, 100, 1)) +
```

```
  labs(
    title = "Posterior predictive distributions of college enrollment",
    y = "Group: race/ethnicity + gender + poverty status (185%)",
    x = "Rate of college attendance"
  ) +
  theme_bw()
comp_g
```


Posterior predictive distributions of college enrollment
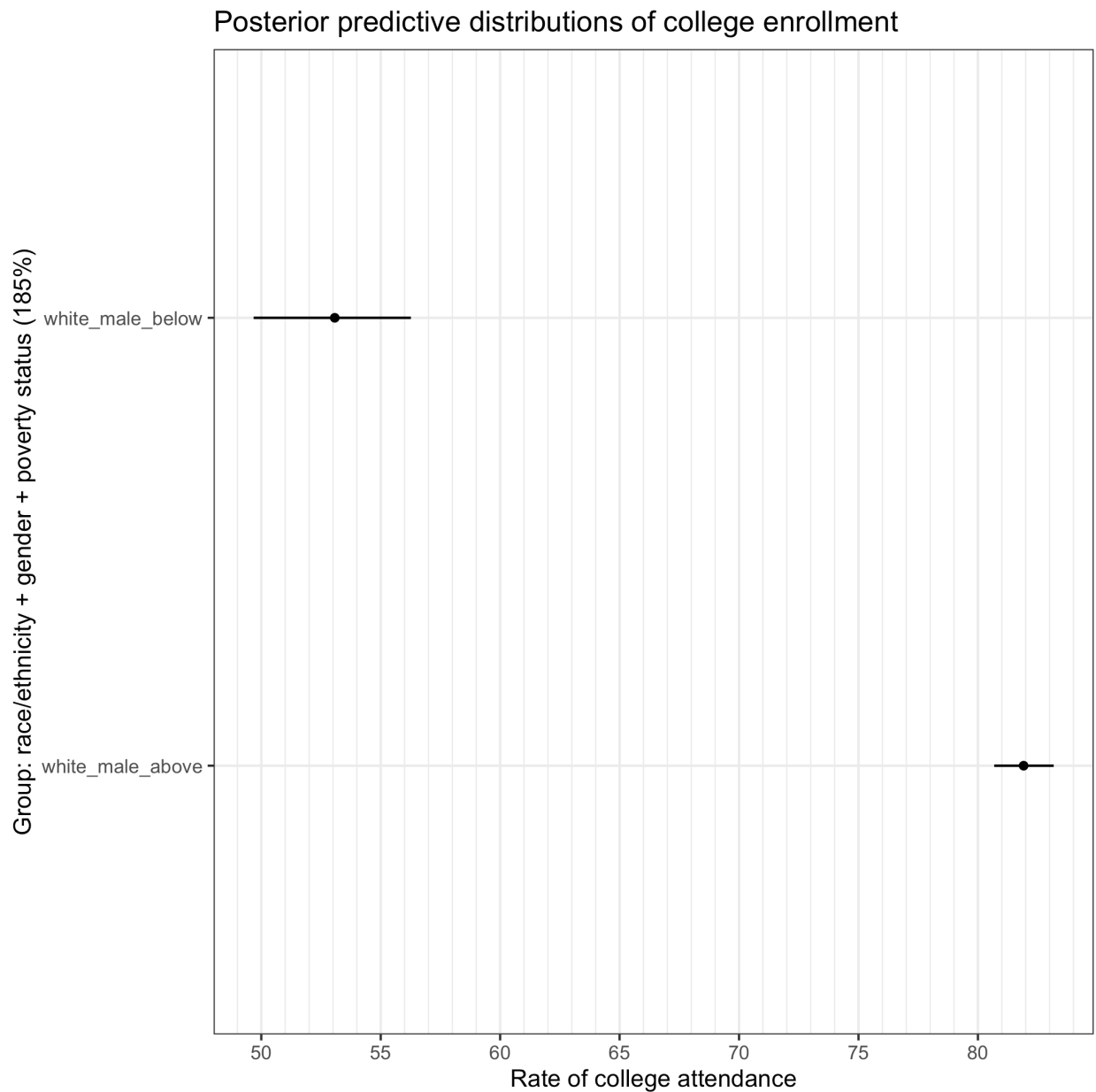
```
## wrangle data to subtract one group of predicitions from the other to get
## estimate of difference
pp_diff <- tibble(wma = pp_comp |> filter(group == "white_male_above") |> pull(.epred),
                  wmb = pp_comp |> filter(group == "white_male_below") |> pull(.epred),
                  diff = wma - wmb)
```

```
## plot density of difference
diff_g <- pp_diff |>
  ggplot(aes(x = diff)) +
  geom_density() +
  geom_vline(xintercept = pp_diff$diff |> mean(), linetype = "dashed") +
  scale_x_continuous(breaks = seq(0, 100, 1),
                     minor_breaks = seq(0, 100, 0.5)) +
   labs(
     title = "Difference in attendance rates",
     subtitle = "(White, male, above 185%) - (White, male, below 185%)",
     y = "Density",
     x = "Percentage point difference"
   ) +
  theme_bw()
diff_g
```

Difference in attendance rates

(White, male, above 185%) - (White, male, below 185%)