GR5065 Assignment 4

Juan Lopez Martin

1 The Impact of Medicaid Expansion on Voter Participation

```
oregon <- as_factor(read_dta(file.path("19026_supp", "Data", "individual_voting_data.dta")))
table(oregon$treatment) # this indicates who won the lottery
```

```
##
##     0     1
## 45088 29834
```

I'm going to analyze *Table 3 Row 2*. As an initial exploration, I will reproduce the results of columns 2 and 3 in order to check I'm using the correct variables. For column 2, we also get that the control group mean is 22.895.

```
fit <- lm(vote_midterm_2010_2 ~ treatment, data = oregon, weights = weight_nov2010)
summary(fit)
```

```
##
## Call:
## lm(formula = vote_midterm_2010_2 ~ treatment, data = oregon,
##     weights = weight_nov2010)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1560 -0.2289 -0.2100  0.0000  8.2135
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.228947   0.002427  94.340  < 2e-16 ***
## treatment   -0.018909   0.003837  -4.929 8.31e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5128 on 48765 degrees of freedom
## Multiple R-squared:  0.0004979,  Adjusted R-squared:  0.0004774
## F-statistic: 24.29 on 1 and 48765 DF,  p-value: 8.312e-07
```

For column 3, I get the same point estimates. Standard errors are different because I'm not applying the clustering correction.

```
fit <- lm(vote_midterm_2010_2 ~ treatment + nnnnumhh_li_2 + nnnnumhh_li_3 + prevote, data = oregon, wei
summary(fit)
```

```
##
## Call:
## lm(formula = vote_midterm_2010_2 ~ treatment + nnnnumhh_li_2 +
##     nnnnumhh_li_3 + prevote, data = oregon, weights = weight_nov2010)
```

```
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9021 -0.0938 -0.0821  0.0000  9.2371
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.093792   0.002265  41.405  < 2e-16 ***
## treatment     -0.011704   0.003242  -3.611 0.000306 ***
## nnnnumhh_li_2  0.006748   0.003793   1.779 0.075264 .
## nnnnumhh_li_3  0.064178   0.034999   1.834 0.066701 .
## prevote        0.547405   0.003653 149.854  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.424 on 48762 degrees of freedom
## Multiple R-squared:  0.3167, Adjusted R-squared:  0.3167
## F-statistic:  5651 on 4 and 48762 DF,  p-value: < 2.2e-16
```

```
set.seed(1234)
oregon <- oregon %>% sample_n(10000)
```

1.1 Monotonic Predictor

```
prior_a <- prior(normal(0.5, 1), class = "Intercept")
prior_b <- prior(normal(0, 1), class = "b")
prior_mo <- prior(dirichlet(1, 1), class = "simo", coef = "monumhh_list1")
```

```
oregon$numhh_list <- factor(oregon$numhh_list, ordered = TRUE, levels = c("signed self up",
                                                                          "signed self up + 1 additional
                                                                          "signed self up + 2 additional
```

Note that we are ignoring the weights (as suggested by Prof. Goodrich), which could be added with "vote_midterm_2010_2 | weights(weight_nov2010)".

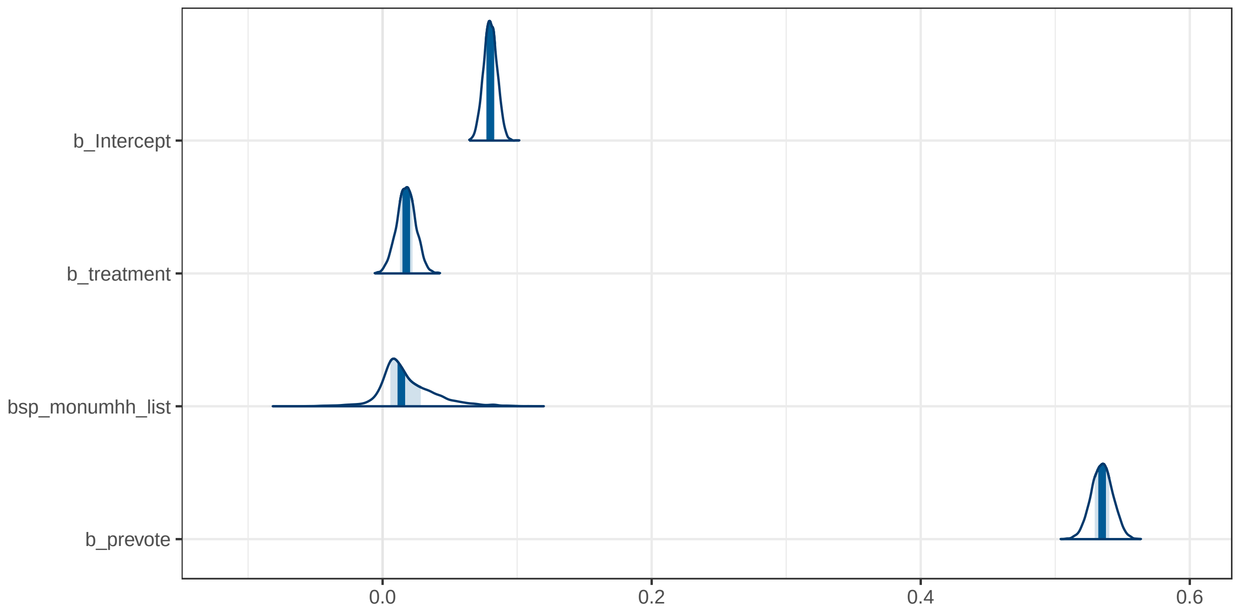
```
prior_a <- prior(normal(0, 5), class = "Intercept")
prior_b <- prior(normal(0, 2), class = "b")
prior_mo <- prior(dirichlet(1, 1), class = "simo", coef = "monumhh_list1")

fit_glm <- brm(vote_midterm_2010_2  ~ treatment + mo(numhh_list) + prevote, data = oregon, prior = prio
```

```
print(fit_glm, digits = 3)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: vote_midterm_2010_2 ~ treatment + mo(numhh_list) + prevote
##    Data: oregon (Number of observations: 10000)
## Samples: 6 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 6000
##
## Population-Level Effects:
##              Estimate Est.Error l-95% CI u-95% CI  Rhat Bulk_ESS Tail_ESS
## Intercept       0.080     0.005    0.070    0.090 1.001     6639     4344
## treatment       0.018     0.007    0.004    0.031 1.004     5891     4415
## prevote         0.535     0.008    0.519    0.550 1.001     5542     3685
```

```
## monumhh_list    0.018     0.021   -0.017    0.069 1.003     2836     2238
##
## Simplex Parameters:
##                   Estimate Est.Error l-95% CI u-95% CI  Rhat Bulk_ESS Tail_ESS
## monumhh_list1[1]    0.354     0.266    0.018    0.930 1.002     2817     3179
## monumhh_list1[2]    0.646     0.266    0.070    0.982 1.002     2817     3179
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI  Rhat Bulk_ESS Tail_ESS
## sigma    0.341     0.002    0.337    0.346 1.003     6445     4328
##
## Samples were drawn 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).
```

```
mcmc_areas(fit_glm, pars = c("b_Intercept", "b_treatment", "bsp_monumhh_list", "b_prevote")) + theme_bw
```



The effect of the treatment has an opposite sign relative to what is reported in the paper. I suspect not using the weighting they are using is an important part of the explanation, because when I fitted the exact same model they're fitting with the standanrd lm I obtained the same point estimates with weighting (as shown above), but when ignoring weighting the results indicated a positive effect of the treatment effect instead of negative.

1.2 Bernoulli likelihood

The results are quite similar to the previous model.

```
fit_ber <- brm(vote_midterm_2010_2 ~ treatment + mo(numhh_list) + prevote, data = oregon, family = berno
```

```
print(fit_ber, digits = 3)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: vote_midterm_2010_2 ~ treatment + mo(numhh_list) + prevote
##    Data: oregon (Number of observations: 10000)
## Samples: 6 chains, each with iter = 2000; warmup = 1000; thin = 1;
```

```
##          total post-warmup samples = 6000
##
## Population-Level Effects:
##               Estimate Est.Error l-95% CI u-95% CI  Rhat Bulk_ESS Tail_ESS
## Intercept      -2.402     0.049   -2.499   -2.307 1.001     3906     3728
## treatment       0.149     0.062    0.024    0.268 1.002     4838     3802
## prevote         2.832     0.059    2.717    2.945 1.000     4444     3926
## monumhh_list    0.137     0.159   -0.145    0.509 1.002     2987     2773
##
## Simplex Parameters:
##                   Estimate Est.Error l-95% CI u-95% CI  Rhat Bulk_ESS Tail_ESS
## monumhh_list1[1]    0.346     0.262    0.017    0.932 1.001     2739     2794
## monumhh_list1[2]    0.654     0.262    0.068    0.983 1.001     2739     2794
##
## Samples were drawn 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).
```

1.3 PSISLOOCV

It seems like the linear regression worked quite better than the logistic regression in this case.

```r
loo_glm <- loo(fit_glm)
loo_glm
```

```
##
## Computed from 6000 by 10000 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  -3446.2  95.6
## p_loo         6.6   0.2
## looic      6892.4 191.1
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```r
loo_ber <- loo(fit_ber)
loo_ber
```

```
##
## Computed from 6000 by 10000 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  -3839.8  61.2
## p_loo         4.2   0.1
## looic      7679.7 122.4
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```r
loo_compare(loo_glm, loo_ber)
```

```
##         elpd_diff se_diff
## fit_glm    0.0       0.0
## fit_ber -393.7      35.0
```

1.4 Stacking Weights

This function tries to find a combination of models that maximizes ELPD, thus assigning weights (not probabilities) to each model. Note that the first model is given much more weight, something that is consistent with the findings above that suggested that this linear regression performed better than logistic regression.

```
loo_model_weights(list(loo_glm, loo_ber), method = "stacking")
```

```
## Method: stacking
## ------
##        weight
## model1 0.822
## model2 0.178
```

1.5 Posterior Probability Over Models

```
bridge_glm <- bridge_sampler(fit_glm, silent = TRUE)

bridge_ber <- bridge_sampler(fit_ber, silent = TRUE)
```

The errors are sufficiently small

```
sapply(list(glm = bridge_glm, ber = bridge_ber), error_measures)
```

```
##            glm         ber
## re2        0.000122577 0.0001080565
## cv         0.01107145  0.01039502
## percentage "1%"        "1%"
```

The result of post__prob indicates that, given the (very strong) assumption that one of these two models is right, the linear regression has a probability of 1 being the one which is right. This extreme results highlights what we have found in the previous two questions.

```
post_prob(bridge_glm, bridge_ber)
```

```
##    bridge_glm    bridge_ber
##  1.000000e+00 6.737512e-166
```

1.6 Projection Pursuit

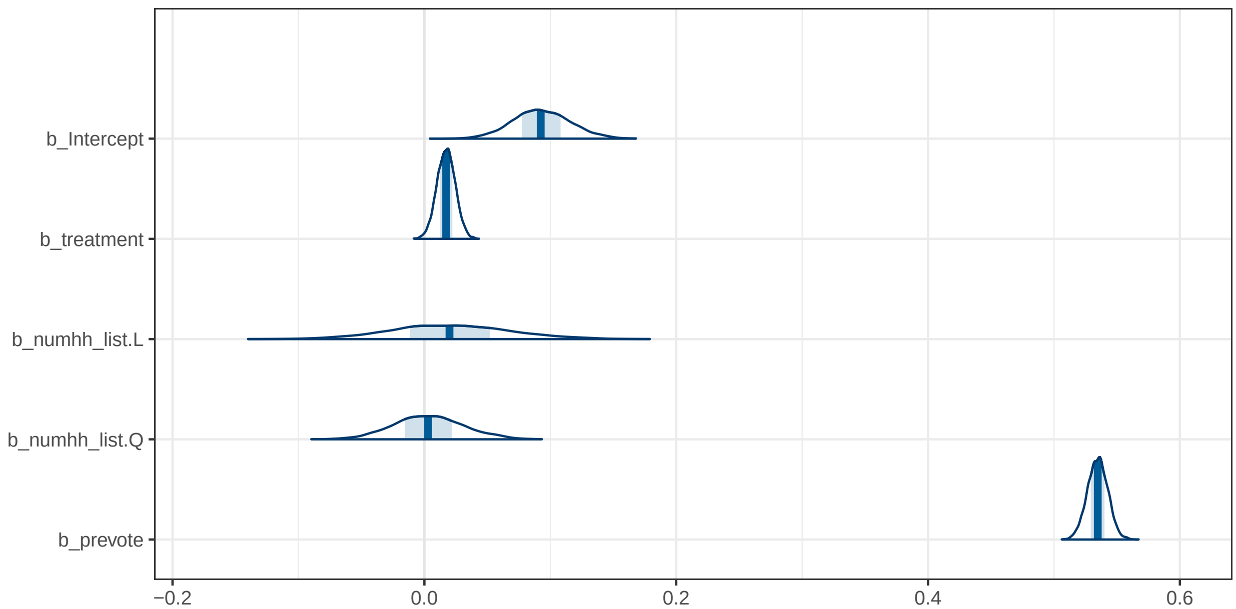
After removing the monotonic predictor we get the following model:

```
prior_a <- prior(normal(0.5, 1), class = "Intercept")
prior_b <- prior(normal(0, 1), class = "b")

fit_glm_2 <- brm(vote_midterm_2010_2  ~ treatment + numhh_list + prevote, data = oregon, prior = prior_a
print(fit_glm_2, digits = 2)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: vote_midterm_2010_2 ~ treatment + numhh_list + prevote
##    Data: oregon (Number of observations: 10000)
## Samples: 6 chains, each with iter = 2000; warmup = 1000; thin = 1;
```

```
##          total post-warmup samples = 6000
##
## Population-Level Effects:
##                Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept          0.09      0.02     0.05     0.14 1.00     2915     3621
## treatment          0.02      0.01     0.00     0.03 1.00     4969     3451
## numhh_list.L       0.02      0.05    -0.07     0.12 1.00     2784     3156
## numhh_list.Q       0.00      0.03    -0.05     0.06 1.00     2768     3269
## prevote            0.53      0.01     0.52     0.55 1.00     4988     3909
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.34      0.00     0.34     0.35 1.00     4940     4098
##
## Samples were drawn 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).
```

```
mcmc_areas(fit_glm_2, pars = c("b_Intercept", "b_treatment", "b_numhh_list.L", "b_numhh_list.Q", "b_pre
```

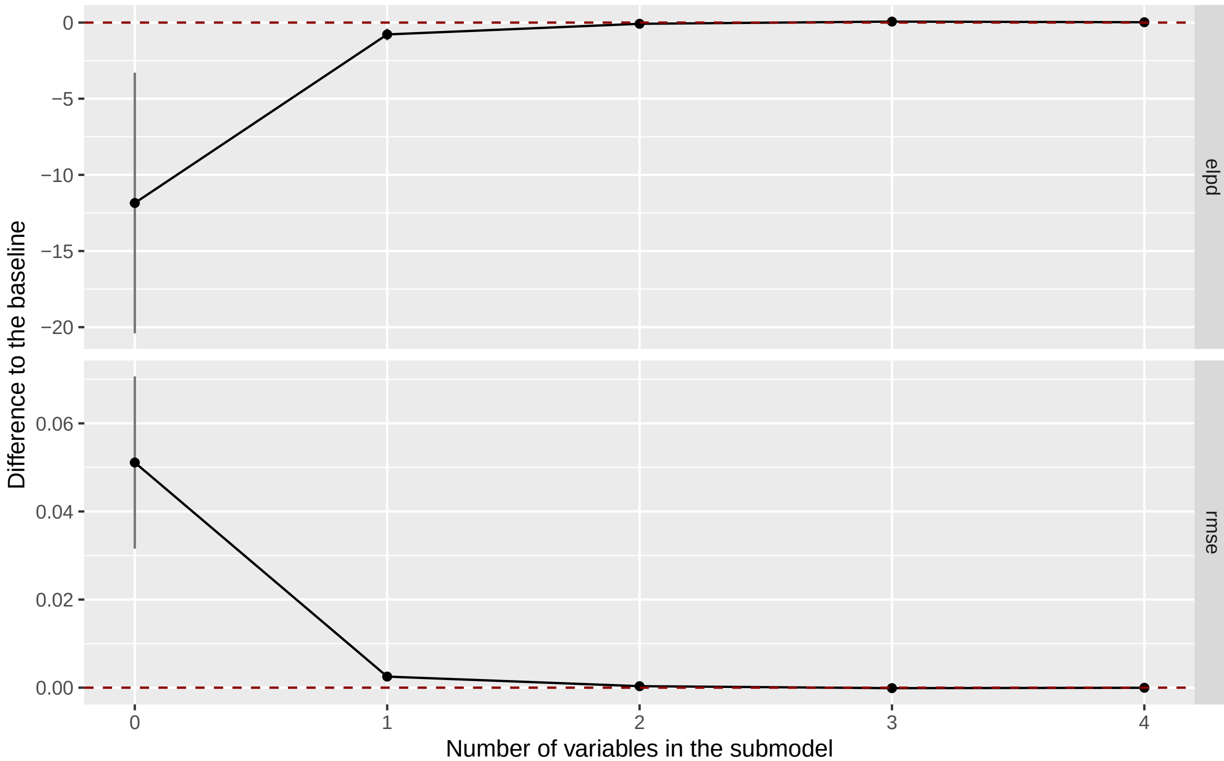


In general, we almost never want to exclude the treatment variable given that its associated β is the main parameter of interest, but in this case we debate whether it should be excluded from a prediction perspective.

FIrst, we see that using projpred reveals that the two numhh parameters can be excluded without almost any increase to the ELPD. Whether we should exclude the treatment variable or not is more open to discussion. one the one hand, the increase in ELPD is quite small with wide SE, something that should not be a surprise given that the prevote parameter is (obviously) strongly related with voting behavior while the treatment is, at least as far as we know, slightly (if at all) related to voting behavior in this particular case. Remember that including the weights even changed the sign of the treatment, probably suggesting its effect is very noisy and unclear. However, even with that small increase in ELPD, the function suggest_size with its defaults parameters recommends keeping it.

```
library(projpred)
cvs <- cv_varsel(fit_glm_2, method = "forward", nloo = 100)
```

```r
varsel_plot(cvs, stats = c('elpd', 'rmse'), deltas = TRUE)
```




```r
cvs
```

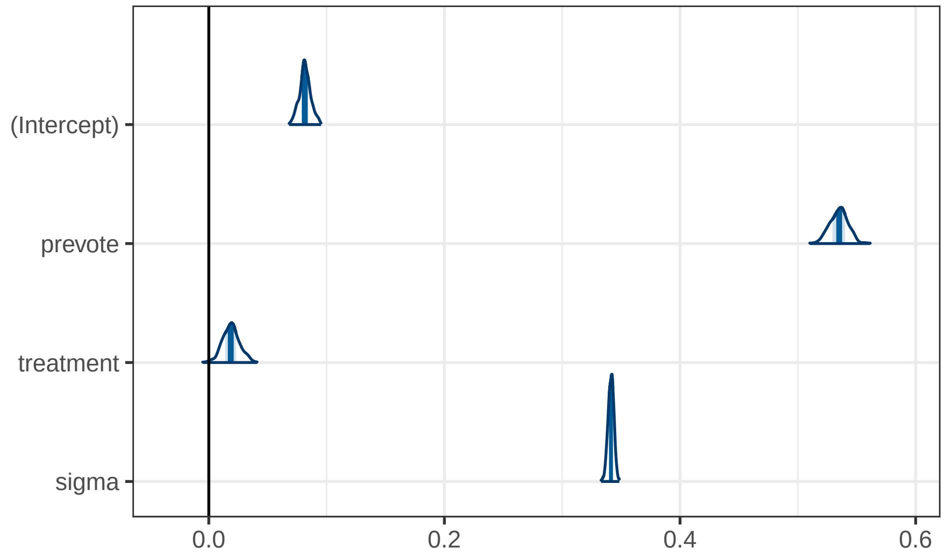
```
##              size     elpd elpd.se pctch
##                 0 -3458.89   95.95    NA
## prevote         1 -3447.82   95.57     1
## treatment       2 -3447.12   95.57     1
## numhh_list.L    3 -3446.98   95.57     1
## numhh_list.Q    4 -3447.02   95.57     1
```

```r
suggest_size(cvs)
```

```
## [1] 2
```

Note that, even if we keep it, the coefficient is very close to zero and relatively noisy. This is consistent with finding in the paper that this coefficient is not statistically significant, although in this case (i.e. with this particular subset of the data and having removed the other parameters) it turns out that its CI barely excludes 0.

```r
newfit <- project(cvs, nv = 2)
mcmc_areas(as.matrix(newfit)) + theme_bw() + geom_vline(aes(xintercept = 0))
```




```r
mean(as.matrix(newfit)[,"treatment"]) - 2*sd(as.matrix(newfit)[,"treatment"])
```

```
## [1] 0.004312926
```

```r
mean(as.matrix(newfit)[,"treatment"]) + 2*sd(as.matrix(newfit)[,"treatment"])
```

```
## [1] 0.03345442
```

1.7 Unbiasedness

The frequentist approaches have placed great emphasis on unbiasedness as a desirable principle of parameter estimation. It is quite intuitive that, over repeated sampling, the mean of a parameter estimate should be equal to its true value. This motivates the reasoning in the paragraph, arguing that the least squares estimator will tend to be better in that case as it is unbiased, while a logit or probit model is not. However, in practical terms there are some complications with this reasoning.

- First, it is obvious that in all cases we have only a finite sample, something that is particularly problematic when having small sample sizes. When having a finite sample size, it is in many cases good to use a particular modelling approach that, even if biased, offers a more realistic model that performs better (i.e. reduces variance). Therefore, in some cases we may want to use a logit/probit model instead of the least squares estimator, as the logit/probit function offers a reasonable (although potentially biased) approach that maps the result of a linear model to the estimated probabilities associated with a of a Bernoulli-distributed outcome. This is also the principle behind the use of regularization, which introduces bias in exchange for a reduced variance. In the Bayesian context, this is also the role of priors, include relevant information that can improve performance at the cost of making the modelling biased.

- Furthermore, in multiparameter models (which are almost all of the models used in practice) it is not possible to estimate all the parameters in an even unbiased manner – commonly leading to an overly optimistic estimate of the variances of the parameters. Bayesian modelling is able to propagate uncertainty very effectively, and therefore is preferred to avoid this problem.

In sum, I think it is reasonable to say that the importance of unbiased models has been, in general, overstated. As we often have limited data to estimate multiparameter models, we may trade unbiasedness for the benefit of including prior information in the priors and/or in the model form, the use of regularization, and the propagation of uncertainty – something that will ultimately lead to a better modelling.

2 General Social Survey

```
GSS <- as_factor(read_dta("GSS2018.dta"))
```

```
GSS_clean <- GSS %>% select(age, race, sex, educ, partyid, relig, godswill) %>%
  mutate(relig = factor(replace(relig, relig %in% c("NA", "DK", "IAP"), NA)),
         age = as.numeric(replace(age, age %in% c("NA", "DK", "IAP"), NA)),
         race = factor(replace(race, race %in% c("NA", "DK", "IAP"), NA)),
         sex = factor(replace(sex, sex %in% c("NA", "DK", "IAP"), NA)),
         educ = as.numeric(replace(educ, educ %in% c("na", "dk", "iap"), NA)),
         partyid = as.integer(factor(partyid, ordered = TRUE, levels = c("strong democrat",
                                                                          "not str democrat",
                                                                          "ind, near dem",
                                                                          "independent",
                                                                          "ind, near rep",
                                                                          "not str republican",
                                                                          "strong republican"))),
         godswill = factor(godswill, ordered = TRUE, levels = c("not at all likely",
                                                                "not very likely",
                                                                "somewhat likely",
                                                                "very likely"))) %>%
           mutate(relig = relevel(factor(replace(relig, !(relig %in% c("protestant", "catholic", "none"
```

One interesting section of the 2018 GSS narrated a mental-health related situation that an individual had suffered The identity of the person was randomly sampled (changing his/her name, gender, race, education), and so was the topic of the narration (alcohol dependence, major depression, schizofrenia, drug problem, or no problem). An example is shown below:

$$John/Juan/Mary/Maria$$

is a

$$white/AfricanAmerican/Hispanic$$

$$man/woman$$

who has completed an

$$eighthgrade/highschool/college$$

. About a year ago

$$John/Juan/Mary/Maria$$

was prescribed prescription pain medication for back pain s/he developed folling a car accident. S/He took the pain medication regularly, and after a few weeks found that s/he increasingly felt the desire for more, even though his/her back pain had improved. [John/Juan/Mary/Maria] went to several different doctors to get more prescriptions from them and then started getting them from a friend. Each time [John/Juan/Mary/Maria] tried to cut down, s/he felt anxious and became sweaty and nauseated for hours on end and also could not sleep. These symptoms lasted until s/he resumed taking the prescription pain medication.

$$John/Juan/Mary/Maria$$

's friends complained that s/he had become unreliable - making plans one day, and cancelling the next. His/Her family said s/he had changed and they could no longer count on him/her.

$$John/Juan/Mary/Maria$$

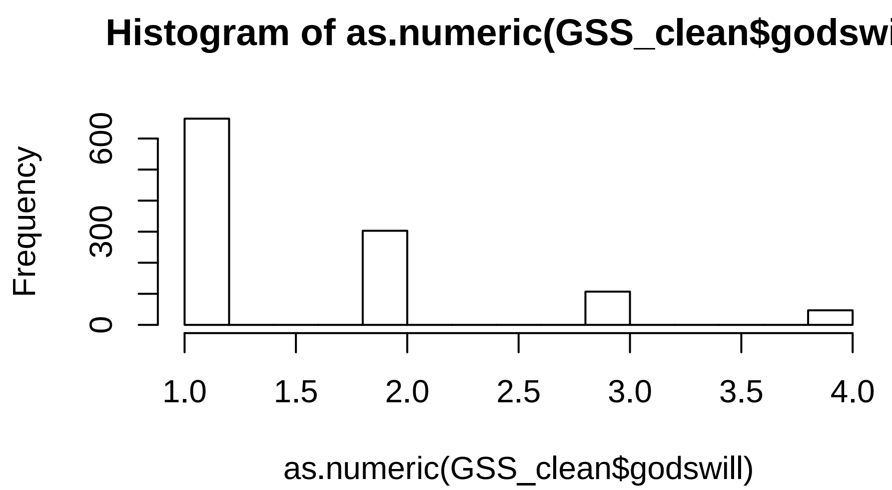
has been livng this way for six months

A series of questions concerning the cause of the negative situation followed the narration. For instance, it asked about the potential effects of

$$John's/Juan's/Mary's/Maria's$$

bad character, brain chemical imbalances, or the way she was raised on his/her problem. A particularly interesting cause for which the survey asked was the potential effect of God's will in the problem. The specific question was: In your opinion, how likely is it that NAME's situation might be caused by God's will [(1) not at all likely, (2) not very likely, (3) somewhat likely, (4) very likely]. Historically, mental health problemas have been framed, in many cases, as particuarly related to supernatural forces. To my surprise, the categories (2), (3), and (4) were more frequently used than I expected – although, being, still, a minority relative to the responses to the first level.

```
hist(as.numeric(GSS_clean$godswill))
```



My goal, therefore, was to explain the potential characteristics that are associated with these responses. Being the variable godswill the outcome, I started including as predictors age, gender (male/female), and race (white/black/other). I also included age^2, in case it could detect any nonlinearity in the relationship. Additionally, I included education level (years of education), religion (none/protestant/catholic/other) and party leaning (Democrat_republican). Note that religion was included as a categorical variable, while party ID was converted to an ordinal variable. My expectation was that age, religion, and party were positively related to the belief in God's will, while education was expected to be negatively related.

Note that not all individuals in the dataset got asked all the questions or gave valid responses, which gives a dataset with only 781 rows. I would have liked to include a variable that referred to trust in science. Multiple variables in the dataset exist that measure similar concepts, but all of them were asked for people for which the God's will question was not presented. Therefore, I finally couldn't include it.

```
GSS_clean %>% na.omit() %>% nrow()
```

```
## [1] 781
```

2.1 Prior predictive distribution

```
prior_fit <- brm(godswill ~ age + I(age^2) + educ + sex + partyid + race + relig, data = GSS_clean, fam
           prior = prior(normal(0, 5), class = "Intercept") + prior(normal(0, 2), class = "b"), sample_
```

```r
print(prior_fit, digits = 2)
```

```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: godswill ~ age + I(age^2) + educ + sex + partyid + race + relig
##    Data: GSS_clean (Number of observations: 781)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept[1]      -56.10   2693.32 -5264.39  5445.16 1.00     5868     2702
## Intercept[2]      -51.81   2693.23 -5256.92  5452.63 1.00     5868     2702
## Intercept[3]      -47.51   2693.32 -5247.49  5457.22 1.00     5868     2702
## age                 0.00      1.99    -3.84     3.95 1.00     5156     2872
## IageE2             -0.04      1.97    -3.82     3.98 1.00     5889     2739
## educ               -0.02      2.03    -4.04     4.00 1.00     6650     2792
## sexfemale           0.02      1.98    -3.89     3.78 1.00     6757     2639
## partyid             0.00      2.00    -3.92     3.91 1.00     5953     2690
## raceblack          -0.01      1.98    -3.85     3.80 1.00     6214     3059
## raceother          -0.02      1.97    -3.86     3.85 1.00     6179     2837
## religprotestant    -0.02      1.98    -3.88     3.97 1.00     5505     2856
## religcatholic       0.03      1.99    -3.81     3.91 1.00     3833     2387
## religother         -0.03      2.05    -4.00     4.04 1.00     5734     2542
##
## Samples were drawn 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).
```

2.2 Posterior Distribution

```r
fit <- brm(godswill ~ age + I(age^2) + educ + sex + partyid + race + relig, data = GSS_clean, family = c
           prior = prior(normal(0, 5), class = "Intercept") + prior(normal(0, 2), class = "b"))
```

```r
print(fit, digits = 2)
```

```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: godswill ~ age + I(age^2) + educ + sex + partyid + race + relig
##    Data: GSS_clean (Number of observations: 781)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept[1]       -1.14      0.48    -2.09    -0.19 1.00     4748     2523
## Intercept[2]        0.37      0.48    -0.57     1.32 1.00     4809     2615
## Intercept[3]        1.78      0.50     0.78     2.78 1.00     4944     2959
## age                -0.02      0.02    -0.05     0.01 1.00     4690     2867
## IageE2              0.00      0.00    -0.00     0.00 1.00     4522     3183
## educ               -0.11      0.03    -0.16    -0.06 1.00     4769     2966
## sexfemale          -0.17      0.14    -0.45     0.11 1.00     4353     2774
## partyid             0.08      0.04     0.00     0.15 1.00     4166     2870
## raceblack           0.65      0.20     0.24     1.05 1.00     4409     2769
```

```
## raceother            0.25      0.26     -0.26      0.75 1.00      4402      2658
## religprotestant      0.45      0.19      0.07      0.84 1.00      3205      2674
## religcatholic        0.40      0.24     -0.08      0.87 1.00      3131      2637
## religother           0.20      0.35     -0.48      0.87 1.00      3698      3195
##
## Samples were drawn 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).
```

- Surprisingly, age does not have a strong effect, and it is even in the opposite direction as what I would have expected. Moreover, the coefficient is Ageˆ2 with a large standard error and should probably be removed from subsequent models.
- As expected, education had a strong negative effect. Effects are relatively difficult to interpret in ordinal regression, but we will try to present it in an intuitive way. Let's start comparing, under the model, two people that have the average age, median political ideology, and that are both white protestant males, but that the first has average education and the second has one year more of education more than the first. Comparing these two individuals, the model expects the latter will have 2.7% more probability of responding with level 1 ("not at all likely"), 1.4% less chances of indicating level 2 ("not very likely"), 0.9% lower probability of indiciating level 3 ("somewhat likely"), and 0.4% smaller chances of selecting level 4 ("very likely") compared with the former. Overall, we see that a one year of extra education with respect to the average decreases the chances of believing in God's will. However, that results from a particular comparison between white protestant males. To get a more general idea, we will also study the effect of education expected by the model when comparing white female catholics with a white female catholics, black male protestants with black male protestants. For each of these comparisons we don't only get the estimated difference in probabilities, but the uncertainty that accompanies it (which we will ignore from simplicity). Averaging over these possibilites (note: for simplicity, we have excluded the case in which religion or race is 'other'), we get that one extra year of education with respect to the average increases the probabilities of by indicating level 1 by 1.4%, decreases the chances of level 2 by 1.1%, the probabilities of level 3 are decreased by 1.0%, and the chances of level 4 are reduced by 0.5%. To avoid wordiness, from now on we will refer to these differences as (1) 2.7% (2) -1.1% (3) -1.0% (4) -0.5%.
- Race seems to a have a strong effect. Although participants that indicated their race as 'other' do not show a consisent effect, people who identified as black did. Compared with whites (the baseline), more likely to believe in God's will as a cause of negative events, on average over all the comparisons we considered (and according to the model), (1) -16.0%, (2) 6.6%, (3) 6.3%, (4) 3.1%.
- Being female does not appear to have a consisent effect, presenting a noisy estimate.
- Party ID appears to suggest that being republican seems to be associated with a hgiher probability of agreeing with the God's will explanation. We obtain that an individual that is one extra level relative to the median (the median in the dataset is being independent, so one extra level means having indicated a preference of "independent, near rep") have, according to the model (1) -1.7% (2) 0.6 (3) 0.7 (4) 0.4 probabilities relative to a person in the median.
- Religion, surprisingly, did not had an effect as strong as expected, with a quite noisy estimate for both protestants [(1) -17.0 (2) 8.5 (3) 5.8 (4) 2.7] and catholics [(1) -17 (2) 8 (3) 6 (4) 3]. As expected, the cathegory that grouped other religions obtained a noisy estimate. Given that the sample size is not that big, that most people had responded with category one, and the similarity between the protestants and catholics, we could probably merge the two categories in subsequent models if we were interested in the effect of christianity on belief in God's will.

In general, it appears that all the effects that appear to be clearly consistent with the data were in the expected direction. However, further research should be done to clarify the strength of these and, potentially, to include other predictors.

```r
# Diferences by educ

dif1 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
```

```r
dif2 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
dif3 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
dif4 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
dif5 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
dif6 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
dif7 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
dif8 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
round(colMeans(rbind(apply(dif1, 2, mean), apply(dif2, 2, mean), apply(dif3, 2, mean), apply(dif4, 2, me
                     rbind(apply(dif5, 2, mean), apply(dif6, 2, mean), apply(dif7, 2, mean), apply(dif8
```

```
## [1]  2.7 -1.1 -1.0 -0.5
```

```r
rbind(Est_dif = round(apply(dif1, 2, mean)*100, 2), SE = round(apply(dif1, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
## Est_dif 2.71 -1.39 -0.91 -0.40
## SE      0.62  0.34  0.23  0.12
```

```r
rbind(Est_dif = round(apply(dif2, 2, mean)*100, 2), SE = round(apply(dif2, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
## Est_dif 2.61 -1.46 -0.81 -0.34
## SE      0.60  0.35  0.21  0.10
```

```r
rbind(Est_dif = round(apply(dif3, 2, mean)*100, 2), SE = round(apply(dif3, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
## Est_dif 2.67 -1.40 -0.88 -0.39
## SE      0.61  0.35  0.24  0.13
```

```r
rbind(Est_dif = round(apply(dif4, 2, mean)*100, 2), SE = round(apply(dif4, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
## Est_dif 2.57 -1.45 -0.79 -0.33
## SE      0.60  0.35  0.22  0.11
```

```r
rbind(Est_dif = round(apply(dif5, 2, mean)*100, 2), SE = round(apply(dif5, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
## Est_dif 2.71 -0.73 -1.26 -0.72
## SE      0.63  0.37  0.31  0.22
```

```r
rbind(Est_dif = round(apply(dif6, 2, mean)*100, 2), SE = round(apply(dif6, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
## Est_dif 2.76 -0.96 -1.18 -0.62
## SE      0.63  0.37  0.30  0.19
```

```r
rbind(Est_dif = round(apply(dif7, 2, mean)*100, 2), SE = round(apply(dif7, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
```

```
## Est_dif 2.71 -0.79 -1.23 -0.70
## SE      0.63  0.43  0.32  0.23
```

```r
rbind(Est_dif = round(apply(dif8, 2, mean)*100, 2), SE = round(apply(dif8, 2, sd)*100, 2))
```

```
##          [,1]  [,2]  [,3]  [,4]
## Est_dif 2.75 -1.00 -1.15 -0.60
## SE      0.63  0.41  0.31  0.21
```

```r
# Diferences by race

dif1 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif2 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif3 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif4 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean


round(colMeans(rbind(apply(dif1, 2, mean), apply(dif2, 2, mean), apply(dif3, 2, mean), apply(dif4, 2, me
```

```
## [1] -15.9   6.6   6.2   3.1
```

```r
rbind(Est_dif = round(apply(dif1, 2, mean)*100, 2), SE = round(apply(dif1, 2, sd)*100, 2))
```

```
##            [,1] [,2] [,3] [,4]
## Est_dif -15.99 6.01 6.58 3.40
## SE        4.94 1.89 2.30 1.37
```

```r
rbind(Est_dif = round(apply(dif2, 2, mean)*100, 2), SE = round(apply(dif2, 2, sd)*100, 2))
```

```
##            [,1] [,2] [,3] [,4]
## Est_dif -15.95 6.98 6.04 2.93
## SE        4.99 2.10 2.16 1.19
```

```r
rbind(Est_dif = round(apply(dif3, 2, mean)*100, 2), SE = round(apply(dif3, 2, sd)*100, 2))
```

```
##            [,1] [,2] [,3] [,4]
## Est_dif -15.92 6.20 6.42 3.30
## SE        4.94 2.02 2.37 1.46
```

```r
rbind(Est_dif = round(apply(dif4, 2, mean)*100, 2), SE = round(apply(dif4, 2, sd)*100, 2))
```

```
##            [,1] [,2] [,3] [,4]
## Est_dif -15.81 7.09 5.89 2.84
## SE        5.00 2.16 2.25 1.29
```

```r
# Diferences by partyid

dif1 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif2 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif3 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif4 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif5 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean
```

```r
dif6 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif7 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif8 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

round(colMeans(rbind(apply(dif1, 2, mean), apply(dif2, 2, mean), apply(dif3, 2, mean), apply(dif4, 2, me
                     rbind(apply(dif5, 2, mean), apply(dif6, 2, mean), apply(dif7, 2, mean), apply(dif8
```

```
## [1] -1.73  0.66  0.70  0.37
```

```r
rbind(Est_dif = round(apply(dif1, 2, mean)*100, 2), SE = round(apply(dif1, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.84 0.90 0.64 0.29
## SE       0.86 0.44 0.31 0.15
```

```r
rbind(Est_dif = round(apply(dif2, 2, mean)*100, 2), SE = round(apply(dif2, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.79 0.96 0.58 0.25
## SE       0.84 0.46 0.28 0.13
```

```r
rbind(Est_dif = round(apply(dif3, 2, mean)*100, 2), SE = round(apply(dif3, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.82 0.91 0.63 0.28
## SE       0.85 0.44 0.31 0.15
```

```r
rbind(Est_dif = round(apply(dif4, 2, mean)*100, 2), SE = round(apply(dif4, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.20 0.65 0.38 0.16
## SE       0.57 0.31 0.19 0.09
```

```r
rbind(Est_dif = round(apply(dif5, 2, mean)*100, 2), SE = round(apply(dif5, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.77 0.36 0.88 0.54
## SE       0.82 0.27 0.43 0.30
```

```r
rbind(Est_dif = round(apply(dif6, 2, mean)*100, 2), SE = round(apply(dif6, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.83 0.53 0.83 0.47
## SE       0.84 0.29 0.42 0.27
```

```r
rbind(Est_dif = round(apply(dif7, 2, mean)*100, 2), SE = round(apply(dif7, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.78 0.40 0.86 0.52
## SE       0.82 0.32 0.43 0.31
```

```r
rbind(Est_dif = round(apply(dif8, 2, mean)*100, 2), SE = round(apply(dif8, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -1.82 0.56 0.81 0.45
## SE       0.84 0.34 0.42 0.28
```

```r
# Diferences by protestant

dif1 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif2 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif3 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif4 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

round(colMeans(rbind(apply(dif1, 2, mean), apply(dif2, 2, mean), apply(dif3, 2, mean), apply(dif4, 2, me
```

```
## [1] -18.1   9.1   6.2   2.9
```

```r
rbind(Est_dif = round(apply(dif1, 2, mean)*100, 2), SE = round(apply(dif1, 2, sd)*100, 2))
```

```
##           [,1] [,2] [,3] [,4]
## Est_dif -10.39 5.78 3.24 1.38
## SE        4.39 2.53 1.39 0.63
```

```r
rbind(Est_dif = round(apply(dif2, 2, mean)*100, 2), SE = round(apply(dif2, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -9.88 5.85 2.86 1.17
## SE       4.15 2.53 1.22 0.53
```

```r
rbind(Est_dif = round(apply(dif3, 2, mean)*100, 2), SE = round(apply(dif3, 2, sd)*100, 2))
```

```
##           [,1]  [,2] [,3] [,4]
## Est_dif -26.38 11.79 9.81 4.78
## SE        6.34  2.97 2.70 1.58
```

```r
rbind(Est_dif = round(apply(dif4, 2, mean)*100, 2), SE = round(apply(dif4, 2, sd)*100, 2))
```

```
##           [,1]  [,2] [,3] [,4]
## Est_dif -25.83 12.83  8.9 4.09
## SE        6.25  3.12  2.5 1.36
```

```r
# Diferences by catholic

dif1 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif2 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif3 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

dif4 <- matrix(pp_expect(fit, newdata = data.frame(age = mean(GSS_clean$age, na.rm = TRUE), educ = mean

round(colMeans(rbind(apply(dif1, 2, mean), apply(dif2, 2, mean), apply(dif3, 2, mean), apply(dif4, 2, me
```

```
## [1] -17   8   6   3
```

```r
rbind(Est_dif = round(apply(dif1, 2, mean)*100, 2), SE = round(apply(dif1, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -9.11 5.08 2.83 1.20
## SE       5.45 3.04 1.75 0.78
```

```r
rbind(Est_dif = round(apply(dif2, 2, mean)*100, 2), SE = round(apply(dif2, 2, sd)*100, 2))
```

```
##          [,1] [,2] [,3] [,4]
## Est_dif -8.67 5.13 2.51 1.03
## SE       5.19 3.07 1.56 0.67
```

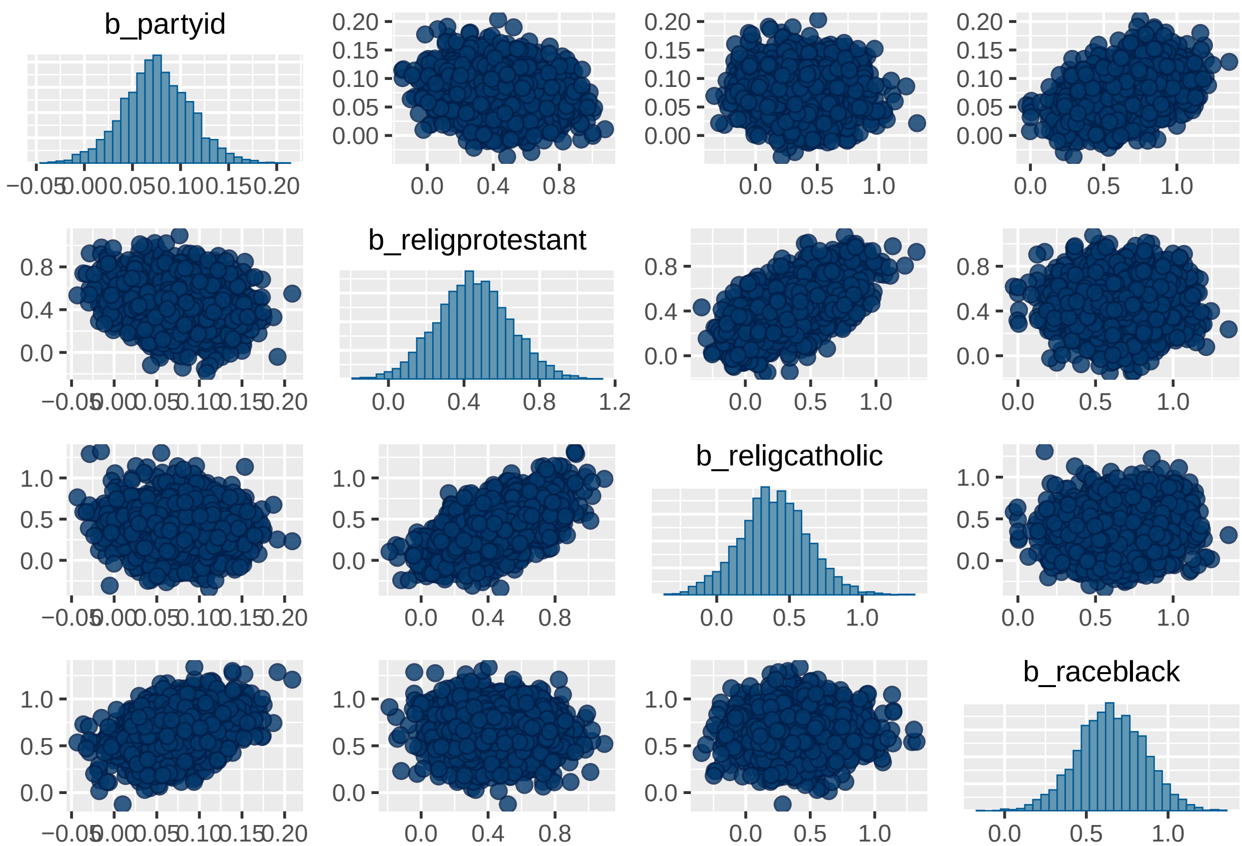
```r
rbind(Est_dif = round(apply(dif3, 2, mean)*100, 2), SE = round(apply(dif3, 2, sd)*100, 2))
```

```
##            [,1]  [,2] [,3] [,4]
## Est_dif -25.03 11.28 9.25  4.5
## SE        7.60  3.25 3.28  1.9
```

```r
rbind(Est_dif = round(apply(dif4, 2, mean)*100, 2), SE = round(apply(dif4, 2, sd)*100, 2))
```

```
##            [,1]  [,2] [,3] [,4]
## Est_dif -24.48 12.22 8.40 3.86
## SE        7.56  3.48 3.07 1.67
```

```r
pairs(fit, pars = c("partyid", "religprotestant", "religcatholic", "raceblack"))
```



We fit a new model removind the quadratic term for age and merging protestant and catholic categories. Note how age appears to be negatively related to beliving in God's will, something I did not expect and that should require further investigaion. In general, I think this second model is easier to interpret, but for the posterior predictive check in the next section we will maintain the first model as the rubric only asked for one model.

```r
GSS_clean$relig <- forcats::fct_collapse(GSS_clean$relig, christian = c("protestant", "catholic"))

fit2 <- brm(godswill ~ age + educ + sex + partyid + race + relig, data = GSS_clean, family = cumulative
```

```r
           prior = prior(normal(0, 5), class = "Intercept") + prior(normal(0, 2), class = "b"))
```

```r
print(fit2, digits = 3)
```

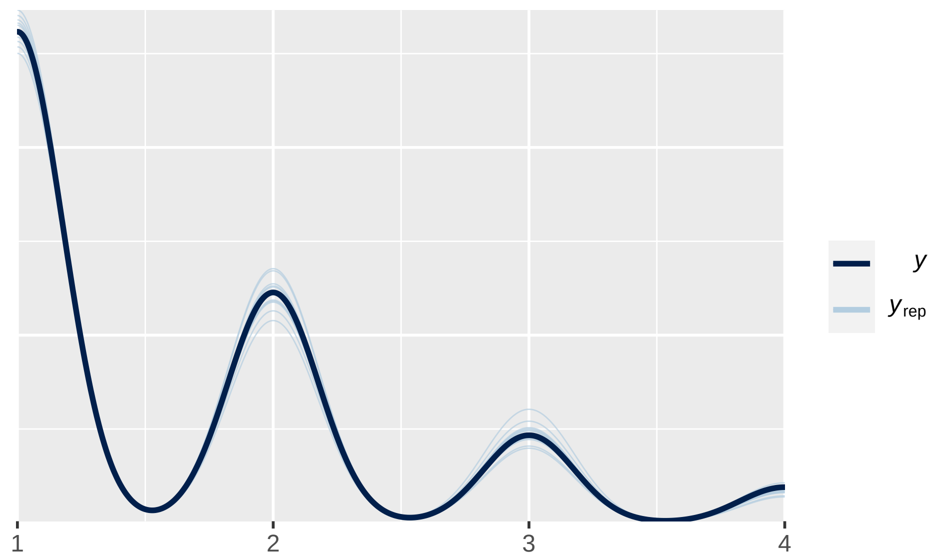
```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: godswill ~ age + educ + sex + partyid + race + relig
##    Data: GSS_clean (Number of observations: 781)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##                Estimate Est.Error l-95% CI u-95% CI  Rhat Bulk_ESS Tail_ESS
## Intercept[1]     -1.064     0.480   -1.995   -0.117 1.000     6440     3453
## Intercept[2]      0.445     0.479   -0.491    1.373 1.000     6671     3483
## Intercept[3]      1.857     0.502    0.879    2.819 1.001     6378     3611
## age              -0.012     0.004   -0.021   -0.004 1.002     7914     3349
## educ             -0.114     0.027   -0.166   -0.062 1.000     8020     3165
## sexfemale        -0.171     0.143   -0.458    0.107 1.001     9421     2814
## partyid           0.076     0.036    0.005    0.146 1.001     7253     3075
## raceblack         0.661     0.201    0.257    1.059 1.003     5509     3127
## raceother         0.243     0.251   -0.250    0.735 1.000     6463     2889
## religchristian    0.441     0.189    0.073    0.823 1.001     7038     3324
## religother        0.224     0.356   -0.475    0.916 1.003     6592     3293
##
## Samples were drawn 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).
```

I also fitted a model with an interaction, but it was too noisy. More data will be needed if we were interested in interactions.

2.3 Posterior Predictive Check

The posterior predictive check seems to imply the model fits the data well.

```r
pp_check(fit, plotfun = "ppc_bars")
```



y
y_{rep}
1
2
3
4