GR5065 Homework 6 Answer Key

Ben Goodrich

Due April 13, 2021 at 8PM New York Time

```r
set.seed(20210413)
options(mc.cores = parallel::detectCores(), width = 90)
```

1 Household Pulse Survey

```r
unzip("HPS_Week25_PUF_CSV.zip")
library(readr)
pulse <- read_csv("pulse2021_puf_25.csv")
```

After loading the data, we deal with some factor variables

```r
pulse$FEMALE <- as.integer(pulse$EGENDER == 2)
pulse$RRACE  <- factor(pulse$RRACE, levels = 1:4, labels = c("White", "Black", "Asian", "Other"))
pulse$EST_ST <- as.factor(pulse$EST_ST)
pulse$EST_MSA <- as.factor(pulse$EST_MSA)
pulse$EEDUC <- factor(pulse$EEDUC, levels = 1:7,
                      labels = c("Less than HS", "Some HS", "HS", "Some college",
                                 "Associates", "Bachelors", "Graduate"), ordered = TRUE)
pulse$WRKLOSS <- ifelse(pulse$WRKLOSS == -99, NA_integer_, pulse$WRKLOSS == 1)
```

1.1 Posterior Distribution

In addition to education, we include the number of adults who live in the household, since the outcome variable asks "has anyone in your household experienced a loss of employment income", which should be monotonically related to the number of people in the household (that work). In addition to sex and race categories, we include a spline for the year in which the person is born since there are plenty of reasons why the pandemic could affect people of different ages differently, but we have absolutely no reason to believe age is linearly related to the log-odds. Finally, in stratified random sampling designs, you almost always want to adjust for the strata, which is particularly true in this case because some areas were more severely affected by the pandemic than others and some places had more strict restrictions on businesses and movement.

```r
library(brms)
get_prior(WRKLOSS ~ mo(EEDUC) + mo(THHLD_NUMADLT) +
            FEMALE + RRACE + s(TBIRTH_YEAR) + EST_ST + EST_MSA,
          data = pulse, family = bernoulli, subset = WRKLOSS > 0)
```

```
##                 prior     class            coef group resp dpar nlpar bound       source
##                (flat)         b                                                  default
##                (flat)         b     EST_MSA14460                             (vectorized)
##                (flat)         b     EST_MSA16980                             (vectorized)
##                (flat)         b     EST_MSA19100                             (vectorized)
##                (flat)         b     EST_MSA19820                             (vectorized)
```

```
##                 (flat)          b      EST_MSA26420                              (vectorized)
##                 (flat)          b      EST_MSA31080                              (vectorized)
##                 (flat)          b      EST_MSA33100                              (vectorized)
##                 (flat)          b      EST_MSA35620                              (vectorized)
##                 (flat)          b      EST_MSA37980                              (vectorized)
##                 (flat)          b      EST_MSA38060                              (vectorized)
##                 (flat)          b      EST_MSA40140                              (vectorized)
##                 (flat)          b      EST_MSA41860                              (vectorized)
##                 (flat)          b      EST_MSA42660                              (vectorized)
##                 (flat)          b      EST_MSA47900                              (vectorized)
##                 (flat)          b          EST_ST02                              (vectorized)
##                 (flat)          b          EST_ST04                              (vectorized)
##                 (flat)          b          EST_ST05                              (vectorized)
##                 (flat)          b          EST_ST06                              (vectorized)
##                 (flat)          b          EST_ST08                              (vectorized)
##                 (flat)          b          EST_ST09                              (vectorized)
##                 (flat)          b          EST_ST10                              (vectorized)
##                 (flat)          b          EST_ST11                              (vectorized)
##                 (flat)          b          EST_ST12                              (vectorized)
##                 (flat)          b          EST_ST13                              (vectorized)
##                 (flat)          b          EST_ST15                              (vectorized)
##                 (flat)          b          EST_ST16                              (vectorized)
##                 (flat)          b          EST_ST17                              (vectorized)
##                 (flat)          b          EST_ST18                              (vectorized)
##                 (flat)          b          EST_ST19                              (vectorized)
##                 (flat)          b          EST_ST20                              (vectorized)
##                 (flat)          b          EST_ST21                              (vectorized)
##                 (flat)          b          EST_ST22                              (vectorized)
##                 (flat)          b          EST_ST23                              (vectorized)
##                 (flat)          b          EST_ST24                              (vectorized)
##                 (flat)          b          EST_ST25                              (vectorized)
##                 (flat)          b          EST_ST26                              (vectorized)
##                 (flat)          b          EST_ST27                              (vectorized)
##                 (flat)          b          EST_ST28                              (vectorized)
##                 (flat)          b          EST_ST29                              (vectorized)
##                 (flat)          b          EST_ST30                              (vectorized)
##                 (flat)          b          EST_ST31                              (vectorized)
##                 (flat)          b          EST_ST32                              (vectorized)
##                 (flat)          b          EST_ST33                              (vectorized)
##                 (flat)          b          EST_ST34                              (vectorized)
##                 (flat)          b          EST_ST35                              (vectorized)
##                 (flat)          b          EST_ST36                              (vectorized)
##                 (flat)          b          EST_ST37                              (vectorized)
##                 (flat)          b          EST_ST38                              (vectorized)
##                 (flat)          b          EST_ST39                              (vectorized)
##                 (flat)          b          EST_ST40                              (vectorized)
##                 (flat)          b          EST_ST41                              (vectorized)
##                 (flat)          b          EST_ST42                              (vectorized)
##                 (flat)          b          EST_ST44                              (vectorized)
##                 (flat)          b          EST_ST45                              (vectorized)
##                 (flat)          b          EST_ST46                              (vectorized)
##                 (flat)          b          EST_ST47                              (vectorized)
##                 (flat)          b          EST_ST48                              (vectorized)
##                 (flat)          b          EST_ST49                              (vectorized)
```

```
##                 (flat)         b          EST_ST50                        (vectorized)
##                 (flat)         b          EST_ST51                        (vectorized)
##                 (flat)         b          EST_ST53                        (vectorized)
##                 (flat)         b          EST_ST54                        (vectorized)
##                 (flat)         b          EST_ST55                        (vectorized)
##                 (flat)         b          EST_ST56                        (vectorized)
##                 (flat)         b            FEMALE                        (vectorized)
##                 (flat)         b           moEEDUC                        (vectorized)
##                 (flat)         b   moTHHLD_NUMADLT                        (vectorized)
##                 (flat)         b        RRACEAsian                        (vectorized)
##                 (flat)         b        RRACEBlack                        (vectorized)
##                 (flat)         b        RRACEOther                        (vectorized)
##                 (flat)         b    sTBIRTH_YEAR_1                        (vectorized)
##  student_t(3, 0, 2.5) Intercept                                               default
##  student_t(3, 0, 2.5)       sds                                               default
##  student_t(3, 0, 2.5)       sds    s(TBIRTH_YEAR)                        (vectorized)
##          dirichlet(1)      simo          moEEDUC1                             default
##          dirichlet(1)      simo  moTHHLD_NUMADLT1                             default
```

We go with fairly informative priors on the effect of education and number of adults in the household and neutral priors on all the other coefficients. The location of the prior on the intercept was chosen to place about a 0.33 probability of an "average" person experiencing an employment income loss. Finally, we put a unit-exponential prior on the standard deviation of the (normally distributed) coefficients in the spline term.

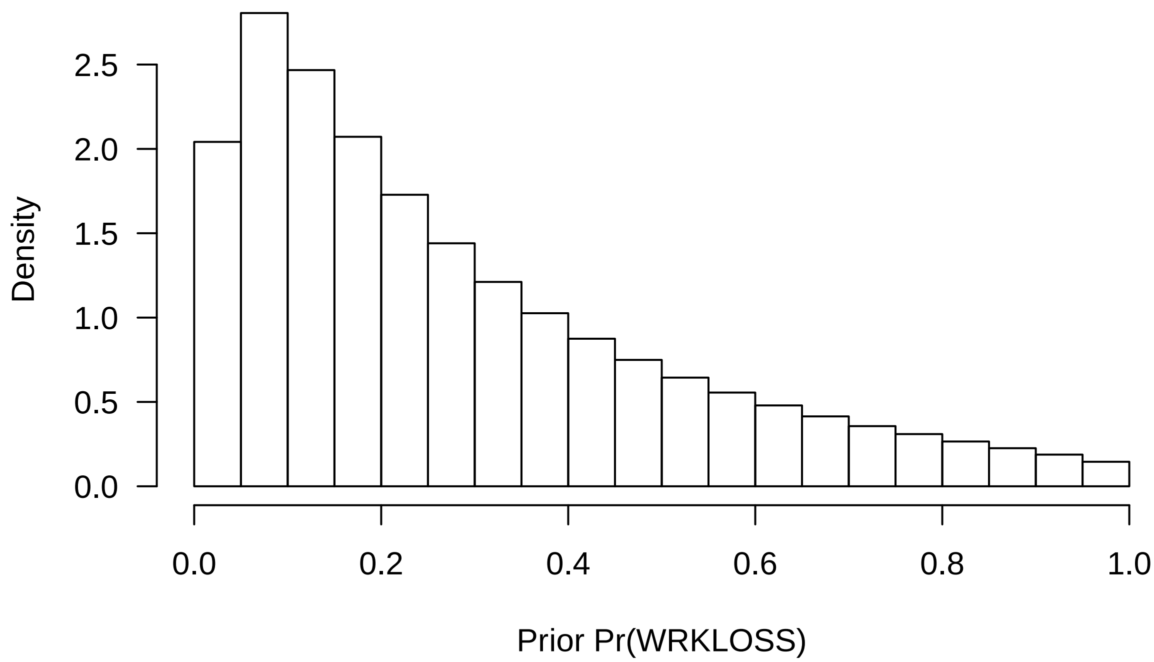
```
my_prior <- prior(normal(0, 0.5), class = "b") +
            prior(normal(-0.25, 0.1), class = "b", coef = "moEEDUC") +
            prior(normal(0.5, 0.2), class = "b", coef = "moTHHLD_NUMADLT") +
            prior(logistic(-0.7, 0.25), class = "Intercept") +
            prior(exponential(1), class = "sds")
```

We can check that these priors on the parameters are reasonable by verifying whether the implied prior probability of employment income loss is reasonable. First, we draw from the prior distribution of the parameters

```
prior_draws <- brm(WRKLOSS ~ mo(EEDUC) + mo(THHLD_NUMADLT) +
                     FEMALE + RRACE + s(TBIRTH_YEAR) + EST_ST + EST_MSA,
                   data = pulse, family = bernoulli, prior = my_prior, sample_prior = "only")
```

and then plot the prior probability of loss of employment income

```
hist(c(posterior_epred(prior_draws)), prob = TRUE, main = "", las = 1, xlab = "Prior Pr(WRKLOSS)")
```



As can be seen, most people had little probability of losing their job under the model but the right tail is somewhat heavy so that some people are almost guaranteed to lose their job.

Since that seems plausible in the aggregate, we go ahead and condition on the observed outcomes to obtain posterior draws of the parameters.

```
post <- brm(WRKLOSS ~ mo(EEDUC) + mo(THHLD_NUMADLT) +
              FEMALE + RRACE + s(TBIRTH_YEAR) + EST_ST + EST_MSA,
            data = pulse, family = bernoulli, prior = my_prior)
```

```
post
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: WRKLOSS ~ mo(EEDUC) + mo(THHLD_NUMADLT) + FEMALE + RRACE + s(TBIRTH_YEAR) + EST_ST + EST_MSA
##    Data: pulse (Number of observations: 25866)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Smooth Terms:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sds(sTBIRTH_YEAR_1)     2.18      0.63     1.28     3.68 1.00     1537     2254
##
## Population-Level Effects:
##                  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           -0.40      0.20    -0.78     0.01 1.00     1612     2201
## FEMALE               0.03      0.03    -0.03     0.08 1.00     8066     2991
## RRACE2               0.27      0.04     0.18     0.35 1.00     6793     2795
## RRACE3              -0.13      0.05    -0.23    -0.03 1.00     7598     2983
## RRACE4               0.29      0.07     0.16     0.42 1.00     7144     2902
## EST_ST02            -0.01      0.49    -0.99     0.95 1.00     9032     2660
## EST_ST04            -0.11      0.37    -0.82     0.59 1.00     3466     2647
## EST_ST05            -0.00      0.50    -1.01     0.99 1.00     7169     2648
## EST_ST06             0.14      0.29    -0.41     0.70 1.00     1551     2204
```

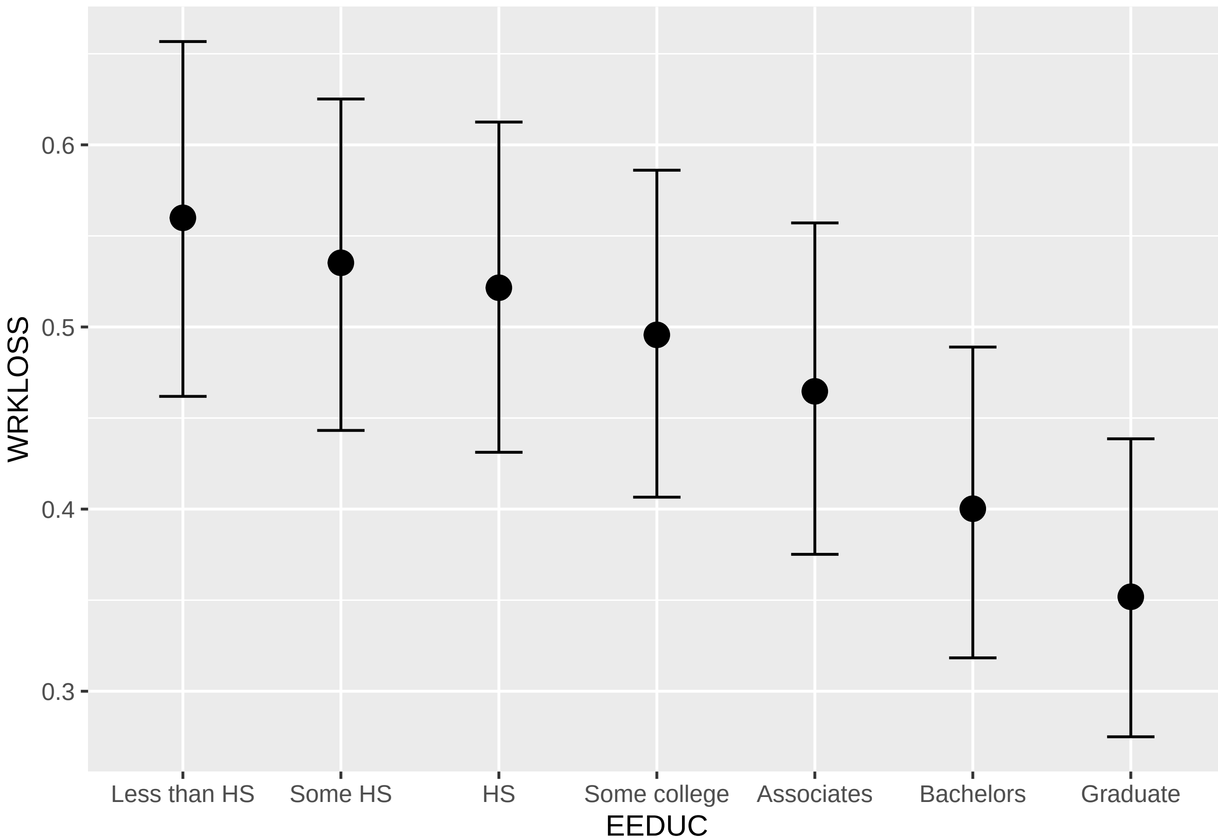
```
## EST_ST08              -0.01      0.51    -1.01      0.98 1.00      8120      3132
## EST_ST09               0.00      0.49    -0.97      0.97 1.00      6899      3014
## EST_ST10               0.10      0.21    -0.31      0.51 1.00      1754      2693
## EST_ST11              -0.33      0.24    -0.80      0.14 1.00      1739      2516
## EST_ST12               0.11      0.36    -0.62      0.81 1.00      3652      2853
## EST_ST13               0.03      0.19    -0.34      0.40 1.00      1543      2017
## EST_ST15              -0.00      0.50    -0.99      0.99 1.00      7896      3210
## EST_ST16              -0.00      0.48    -0.94      0.92 1.00      8831      2723
## EST_ST17               0.15      0.27    -0.38      0.68 1.00      2399      2782
## EST_ST18               0.06      0.29    -0.52      0.63 1.00      2661      2670
## EST_ST19               0.01      0.50    -0.98      0.99 1.00      6442      2626
## EST_ST20               0.00      0.50    -0.96      0.97 1.00      8931      2807
## EST_ST21              -0.01      0.47    -0.94      0.92 1.00      7433      2783
## EST_ST22               0.01      0.51    -0.99      0.98 1.00      6751      2625
## EST_ST23               0.01      0.49    -0.98      0.98 1.00      7192      2410
## EST_ST24              -0.21      0.24    -0.68      0.24 1.00      1665      2641
## EST_ST25               0.07      0.30    -0.54      0.66 1.00      2490      2794
## EST_ST26               0.12      0.37    -0.63      0.86 1.00      3902      2896
## EST_ST27               0.01      0.51    -1.01      1.00 1.01      7798      2183
## EST_ST28              -0.00      0.51    -0.98      0.98 1.00      6396      2503
## EST_ST29               0.00      0.50    -0.99      0.99 1.00      7561      2887
## EST_ST30               0.01      0.50    -0.97      1.01 1.00      8241      3300
## EST_ST31              -0.00      0.52    -1.02      1.01 1.00      8354      2730
## EST_ST32               0.01      0.51    -0.97      0.99 1.00      7361      2907
## EST_ST33              -0.02      0.30    -0.62      0.57 1.00      2576      2796
## EST_ST34               0.08      0.20    -0.33      0.48 1.00      1646      2269
## EST_ST35               0.00      0.50    -0.95      0.99 1.00      7538      3034
## EST_ST36               0.22      0.21    -0.21      0.64 1.00      1765      2609
## EST_ST37               0.01      0.49    -0.95      0.96 1.00      8960      3096
## EST_ST38              -0.00      0.51    -0.97      0.98 1.00      7219      3105
## EST_ST39              -0.01      0.50    -0.96      0.96 1.00      7988      2577
## EST_ST40              -0.01      0.50    -0.96      0.94 1.00      7243      2831
## EST_ST41               0.00      0.48    -0.92      0.93 1.00      7683      2902
## EST_ST42               0.29      0.21    -0.12      0.69 1.00      1538      2341
## EST_ST44              -0.00      0.49    -0.95      0.96 1.00      8421      3081
## EST_ST45              -0.01      0.49    -0.96      0.93 1.00      7432      2892
## EST_ST46               0.00      0.50    -0.96      0.98 1.00      9175      2697
## EST_ST47              -0.00      0.49    -0.96      0.93 1.00      8512      3013
## EST_ST48              -0.00      0.31    -0.62      0.62 1.00      2146      2212
## EST_ST49               0.00      0.50    -0.97      1.00 1.00      8192      2879
## EST_ST50              -0.00      0.50    -0.97      0.97 1.00      8792      2677
## EST_ST51              -0.36      0.24    -0.82      0.11 1.00      1690      2602
## EST_ST53              -0.02      0.36    -0.75      0.68 1.00      3011      2752
## EST_ST54              -0.01      0.51    -0.99      1.00 1.00      9080      2766
## EST_ST55              -0.35      0.34    -1.01      0.31 1.00      4031      3380
## EST_ST56               0.00      0.51    -0.99      1.01 1.00      8214      2906
## EST_MSA14460           0.02      0.31    -0.60      0.62 1.00      2177      2646
## EST_MSA16980          -0.16      0.29    -0.73      0.42 1.00      2074      2748
## EST_MSA19100          -0.03      0.30    -0.63      0.55 1.00      2333      2870
## EST_MSA19820           0.12      0.37    -0.60      0.86 1.00      3875      2654
## EST_MSA26420           0.01      0.29    -0.58      0.59 1.00      2308      2689
## EST_MSA31080           0.26      0.26    -0.26      0.76 1.00      1662      2394
## EST_MSA33100           0.12      0.36    -0.59      0.85 1.00      3236      2602
## EST_MSA35620           0.03      0.23    -0.44      0.49 1.01      1465      2339
```

```
## EST_MSA37980       -0.21      0.23     -0.66      0.24 1.01     1357     2228
## EST_MSA38060       -0.10      0.36     -0.81      0.64 1.00     3333     2866
## EST_MSA40140       -0.09      0.26     -0.60      0.41 1.00     1585     2176
## EST_MSA41860       -0.00      0.26     -0.52      0.50 1.00     1652     2326
## EST_MSA42660       -0.01      0.37     -0.73      0.71 1.00     2537     2517
## EST_MSA47900       -0.04      0.26     -0.57      0.48 1.00     1485     2236
## sTBIRTH_YEAR_1     -0.06      0.48     -1.00      0.90 1.00     4918     3016
## moEEDUC            -0.14      0.02     -0.18     -0.12 1.00     2230     2339
## moTHHLD_NUMADLT     0.16      0.02      0.13      0.20 1.00     2281     2115
##
## Simplex Parameters:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## moEEDUC1[1]              0.12      0.08     0.01     0.30 1.00     2258     2720
## moEEDUC1[2]              0.06      0.05     0.00     0.18 1.00     4018     2047
## moEEDUC1[3]              0.13      0.05     0.02     0.23 1.00     3600     1385
## moEEDUC1[4]              0.14      0.06     0.03     0.27 1.00     2724     1807
## moEEDUC1[5]              0.32      0.07     0.19     0.46 1.00     3163     2859
## moEEDUC1[6]              0.24      0.04     0.16     0.33 1.00     3871     2898
## moTHHLD_NUMADLT1[1]      0.21      0.03     0.16     0.27 1.00     2754     3046
## moTHHLD_NUMADLT1[2]      0.31      0.04     0.24     0.40 1.00     2941     2818
## moTHHLD_NUMADLT1[3]      0.21      0.04     0.13     0.29 1.00     4676     3230
## moTHHLD_NUMADLT1[4]      0.03      0.03     0.00     0.10 1.00     6104     1861
## moTHHLD_NUMADLT1[5]      0.04      0.03     0.00     0.12 1.00     5404     2106
## moTHHLD_NUMADLT1[6]      0.05      0.04     0.00     0.15 1.00     5068     2217
## moTHHLD_NUMADLT1[7]      0.04      0.04     0.00     0.13 1.00     5860     2462
## moTHHLD_NUMADLT1[8]      0.04      0.04     0.00     0.13 1.00     4799     2419
## moTHHLD_NUMADLT1[9]      0.07      0.06     0.00     0.22 1.00     2594     2134
##
## 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).
```

1.2 Interpretation

The effect of education is restricted to be monotonic and it has a very good chance of being decreasing under our prior on `moEEDUC`. In the posterior distribution, the relationship remains decreasing

```r
plot(conditional_effects(post, effect = "EEDUC"))
```



and we become a lot more certain about the magnitude of the dropoff in the probability of a loss in employment income. It is fairly steep for people with graduate or bachelor's degrees compared to lesser levels of education.

1.3 Frequentism

After loading the data.frame of household weights

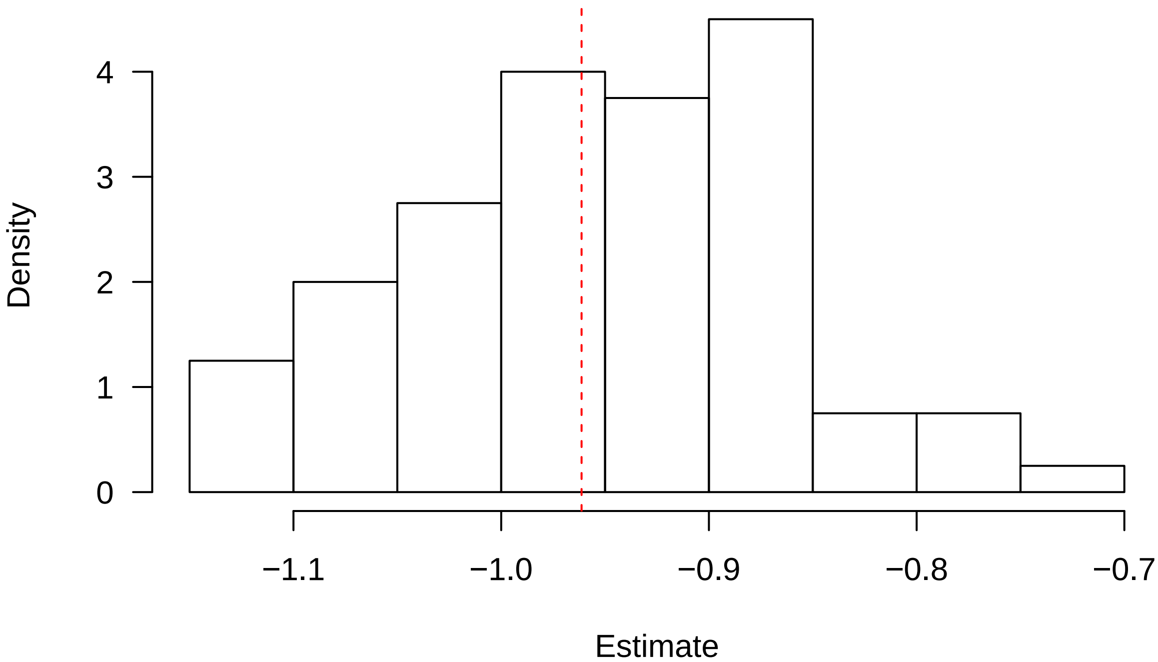
```
pulse_weights <- read_csv("pulse2021_repwgt_puf_25.csv")[ , 3:82]
```

we can estimate a fairly comparable model using maximum likelihood and then re-estimate the effect of having a graduate degree with 80 different sets of weights

```
MLE <- mgcv::gam(WRKLOSS ~ EEDUC + as.factor(THHLD_NUMADLT) +
                  FEMALE + RRACE + s(TBIRTH_YEAR) + EST_ST + EST_MSA,
                 data = pulse, family = binomial, weights = HWEIGHT / sum(HWEIGHT))
EEDUCGraduate <- sapply(pulse_weights, FUN = function(w) {
  coef(update(MLE, weights = w / sum(w)))["EEDUCGraduate"]
})
```

Having 80 different sets of random weights is (somewhat) like having 80 different datasets of size N, which allows us to accomplish the primary goal of Frequentist inference, which is to describe using probability distributions how much an estimator varies across datasets of size N that are induced by the stratified random sampling design. We can plot that distribution as

```
hist(EEDUCGraduate, prob = TRUE, main = "", las = 1, xlab = "Estimate")
theta <- coef(MLE)["EEDUCGraduate"]
abline(v = theta, col = 2, lty = 2)
```



which would look more normal if we had more than 80 sets of weights. These coefficients are roughly centered at the one estimate (red dashed line) we obtained using the original weights, and we can apply the formula to estimate the standard deviation of the estimator:

```
sqrt(4 * mean( (EEDUCGraduate - theta) ^ 2 ))
```

```
## [1] 0.1801617
```

This number is about twice as large as the posterior standard deviation,

```
sd(as.data.frame(post)$bsp_moEEDUC * 6)
```

```
## [1] 0.0932167
```

The fact that both numbers are small relative to the magnitude of the effect of having a graduate degree would lead a lot of people to the unwarranted conclusion that it does not matter whether you used a Bayesian or Frequentist estimator. However, in order to decide what policies to implement as a result of the pandemic, it is completely insufficient to reject the (preposterous) null hypothesis that education has no effect in favor of the (vacuous) alternative hypothesis that it has a negative effect; you have to have a good estimate of the magnitude of the negative effect of education and your uncertainty about that effect in order to come up with a good estimate of how many people at each education level lost their jobs.

Even if the numbers were considered similar, they represent different concepts. The Frequentist formula estimates the standard deviation of point estimates across datasets and the Bayesian calculation is of the standard deviation of MCMC draws conditional on the one dataset that was most recently collected. It is far from intuitive why those numbers should ever be similar, although under some conditions as $N \uparrow \infty$ the distribution of the posterior mode *across datasets* is the same as the distribution of the maximum likelihood estimator (because the prior becomes negligible). For finite N — even a fairly large N of 25866 in this case — the posterior standard deviation should typically be less than a Frequentist standard error if the priors were proper.

The fact that most people analyzing these data would call `glm` without specifying `weights` or if they did, would not utilize the 80 random variations on the household weights to estimate the standard error strongly suggests that most people do not really care about the variation in the point estimator across datasets, or else did not learn how to perform Frequentist estimation beyond the simplest case of simple random sampling from an infinite population, which is what is assumed by `glm` and similar software. But no one does simple random sampling because it is too cost-inefficient compared to alternatives like stratified random sampling, cluster random sampling, etc. So, even if there is some form of random sampling — which there was in this case but often is absent in the social sciences — and some potential to repeat the random sampling process — which there was in this case but usually is absent in the social sciences — the estimated standard errors (and thus of functions of the estimated standard errors like p-values and confidence intervals) do not even correctly reflect the standard deviation in the point estimates across datasets of size N that are induced by the randomization in the sampling process. Moreover, although you can randomize who is *contacted* for a survey, who among those contacted actually agrees to provide responses to the survey is not that random, which is not accounted for by `glm`.

2 General Social Survey

Here I am going to model a question about support for gay marriage that was first asked in 1988, next asked in 2004, and then asked every two years thereafter. As usual, we have to deal with several factor variables.

```
data(gss_all, package = "gssr")
gss_all <- gss_all[gss_all$year == 1988 | gss_all$year >= 2004, ]
gss_all$year_born <- as.integer(gss_all$year) - gss_all$age
gss_all$year <- as.ordered(gss_all$year)
gss_all$female <- as.integer(gss_all$sex == 2)
gss_all$relig16 <- haven::as_factor(gss_all$relig16)
gss_all$y <- haven::as_factor(gss_all$marhomo, ordered = TRUE)
```

All of these are fairly self-explanatory except for `relig16`, which is the religion the respondent was raised in, as opposed to the religion that they identify with today. There is a large literature on "age-period-cohort" models (spoiler: you should estimate them with Bayesian methods because it is impossible to separately identify the three effects using the data alone) but here we are going to eliminate age and just try to distinguish between the period of the survey and the cohort to which the respondent was born.

2.1 Prior Predictive Distribution

It seems plausible that opposition to gay marriage should monotonically decrease over time but implausible that the year in which the respondent was born would have a linear effect on anything. It is well-established from every other survey that women are more supportive of gay marriage than are men, although we could go with a neutral prior on that. Similarly, it is well-known that some religions are more supportive of gay marriage than are others, but there are a lot of religions that I am not that familiar with so I will go with a neutral prior on all of them.

```
get_prior(y ~ mo(year) + female + s(year_born) + relig16,
          data = gss_all, family = cumulative)
```

```
##                 prior     class                          coef group resp dpar nlpar
##                (flat)         b
##                (flat)         b                        female
##                (flat)         b                        moyear
##                (flat)         b               relig16BUDDHISM
##                (flat)         b               relig16CATHOLIC
##                (flat)         b              relig16CHRISTIAN
##                (flat)         b               relig16HINDUISM
##                (flat)         b relig16INTERMNONDENOMINATIONAL
##                (flat)         b                 relig16JEWISH
##                (flat)         b            relig16MOSLEMDISLAM
##                (flat)         b          relig16NATIVEAMERICAN
##                (flat)         b                   relig16NONE
##                (flat)         b       relig16ORTHODOXMCHRISTIAN
```

```
##                (flat)         b                   relig16OTHER
##                (flat)         b            relig16OTHEREASTERN
##                (flat)         b                   syear_born_1
##   student_t(3, 0, 2.5) Intercept
##   student_t(3, 0, 2.5) Intercept                              1
##   student_t(3, 0, 2.5) Intercept                              2
##   student_t(3, 0, 2.5) Intercept                              3
##   student_t(3, 0, 2.5) Intercept                              4
##   student_t(3, 0, 2.5)       sds
##   student_t(3, 0, 2.5)       sds                   s(year_born)
##           dirichlet(1)      simo                        moyear1
##   bound       source
##              default
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##              default
##         (vectorized)
##         (vectorized)
##         (vectorized)
##         (vectorized)
##              default
##         (vectorized)
##              default
```

```r
my_prior <- prior(normal(0, 0.5), class = "b") +
            prior(normal(-0.5, 0.2), class = "b", coef = "moyear") +
            prior(normal(-3.5, 1), class = "Intercept", coef = "1") +
            prior(normal(-2.5, 1), class = "Intercept", coef = "2") +
            prior(normal(-1.5, 1), class = "Intercept", coef = "3") +
            prior(normal(-0.5, 1), class = "Intercept", coef = "4") +
            prior(exponential(1), class = "sds")
```

The hyperparameters to the normal prior on the "Intercepts" (really cutpoints) have been tweaked so that they induce a seemingly reasonable prior predictive distribution of responses. We can verify that by drawing from the prior distribution of all of the parameters,

```r
prior_draws <- brm(y ~ mo(year) + female + s(year_born) + relig16,
                   data = gss_all, family = cumulative, prior = my_prior, sample_prior = "only")
```

and calculating the proportion of draws of the predictive distribution that fall in each of the five categories of the Likert scale for the outcome (from strongly agree to strongly oppose).

```r
round(prop.table(table(c(posterior_predict(prior_draws)))), digits = 2)
```

```
##
##    1    2    3    4    5
## 0.24 0.16 0.17 0.18 0.23
```

As can be seen, the proportions are fairly uniform but I left a little bit of extra probability on the extreme categories to reflect that fact that gay marriage has historically been a contentious issue in the United States.

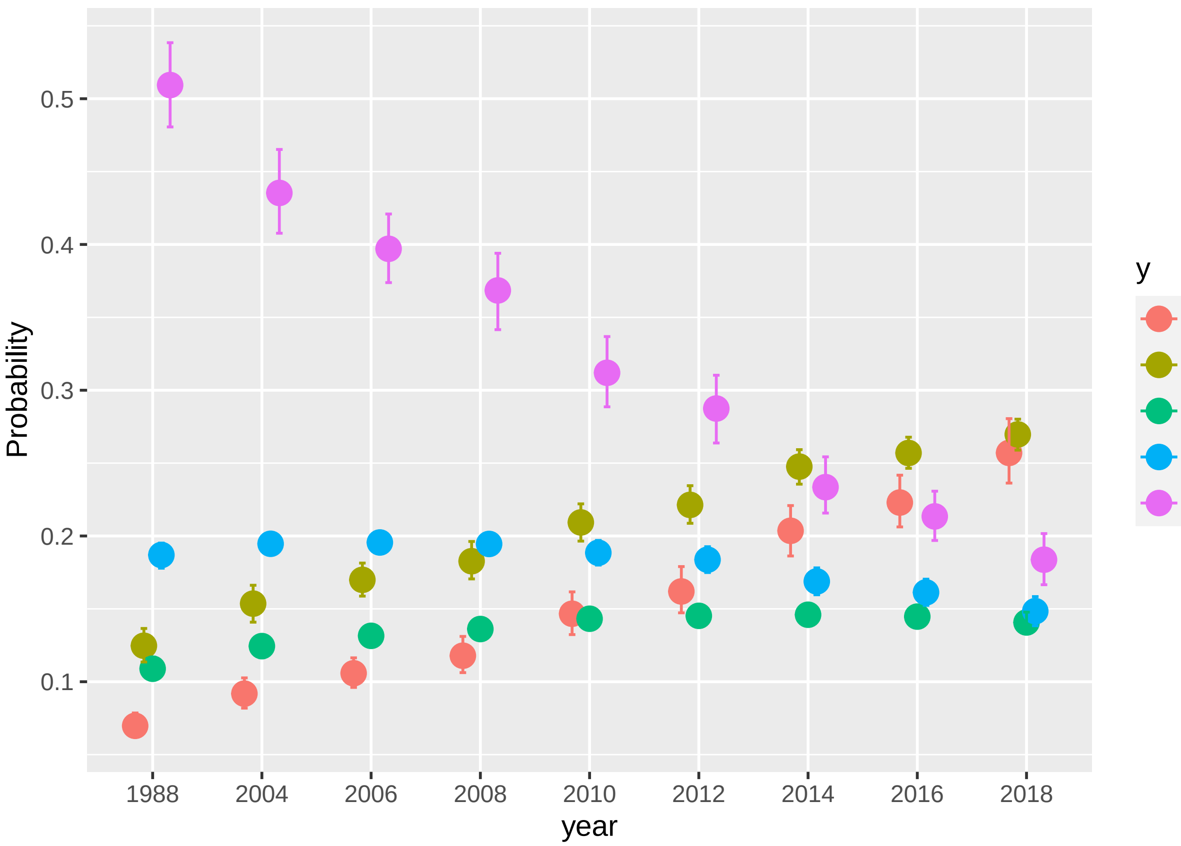
2.2 Posterior Distribution

We can then condition on the observed data to draw from the posterior distribution of the parameters.

```r
post <- brm(y ~ mo(year) + female + s(year_born) + relig16,
            data = gss_all, family = cumulative, prior = my_prior)
```

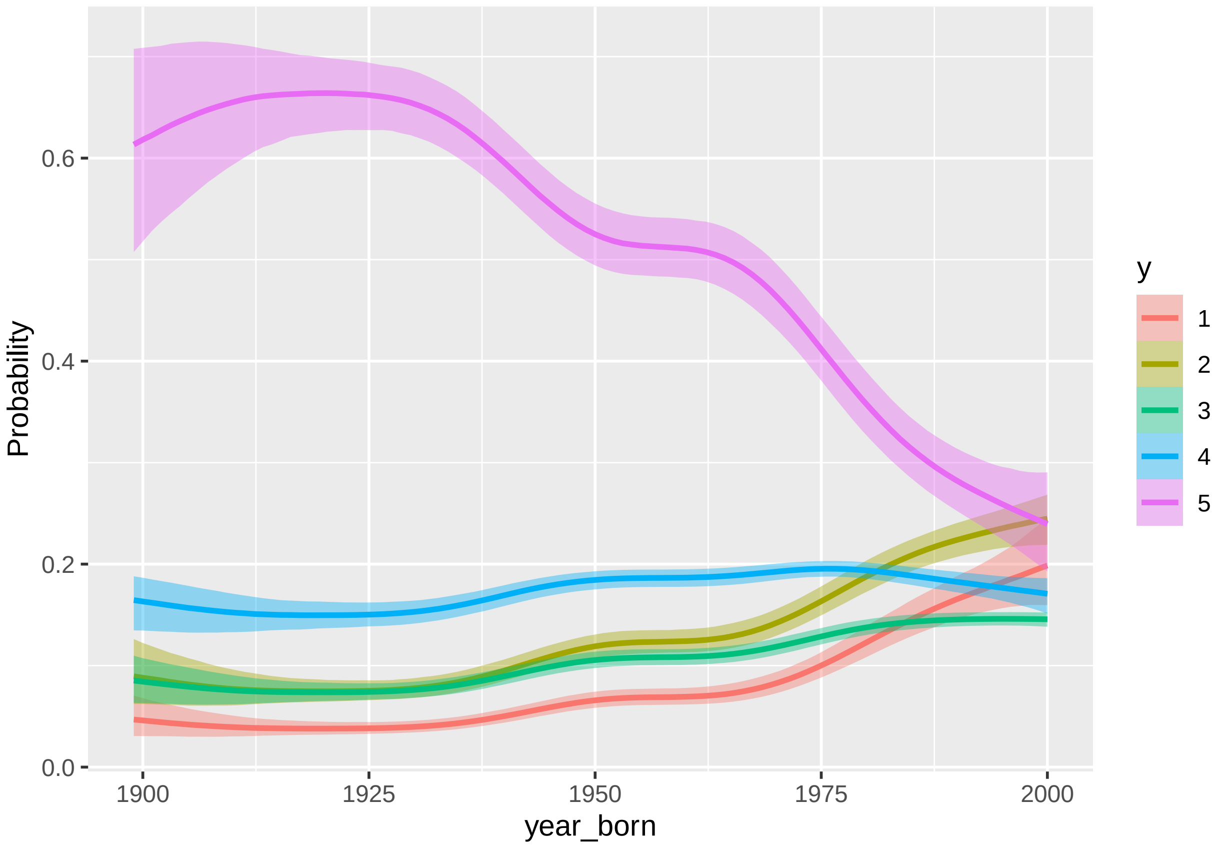
As can be seen, there is a fairly substantial and consistent period effect, where net of all the other predictors, strong opposition (5) to gay marriage to gay marriage dissipates over time and is mostly replaced by support (2) or strong support (1).

```r
plot(conditional_effects(post, effect = "year", categorical = TRUE))
```



This "period" effect seems to reflect many people changing their minds — as opposed to older survey respondents dying and being replaced by younger survey respondents — because the spline already captures that (nonlinear, strong, precise) "cohort" effect:

```r
plot(conditional_effects(post, effect = "year_born", categorical = TRUE))
```



2.3 Addendum

One question that always arises with Likert scale variables is whether distinguishing among the intermediate categories is worthwhile. Bayesians can investigate sych questions by drawing from the posterior distribution of an ordinal model and dichotomizing the outcome after-the-fact (often at the purportedly neutral category) to calculate a log-likelihood over the parameters evaluated at the binary outcome of whether responents support or oppose gay marriage:

```
eta <- posterior_linpred(post)
mu  <- plogis(eta - as.data.frame(post)$`b_Intercept[2]`)
ll  <- dbinom(as.integer(model.frame(post)$y) >= 3, size = 1, prob = mu, log = TRUE)
loo::loo(ll, r_eff = loo::relative_eff(exp(ll), chain_id = rep(1:4, each = 1000)))
```

```
##
## Computed from 4000 by 13378 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo -11520.1 25.4
## p_loo      2332.9 24.0
## looic     23040.2 50.8
## ------
## Monte Carlo SE of elpd_loo is 6.8.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

This estimated ELPD could then be compared to a logit model where the outcome has been dichotomized in advance to see which of the two models is better expected to predict support vs. opposition to gay marriage.