Applied Bayesian modeling - HW4

Score: Each question is worth 10 points. The maximum number of points in this HW is 40 points, with 10 points extra credit. For calculating a final HW grade, the points will be rescaled to a maximum score of (50)/40*100% = 125%.

In this HW, we are going to analyze the wells data (briefly mentioned in class) using logistic regression models, and do model checking. HW4 is based on module 11, part 1 (in-sample checking). Later parts of this analysis include approximate leave-one-out validation and testing sensitivity of results to choice of priors.

If you'd like a refresher on logistic regression, and want to read about it in a Bayesian context, you may find these texts helpful (do add others you recommend on the slack!):

- https://bookdown.org/marklhc/notes_bookdown/generalized-linear-models.html#binary-logistic-regression
- https://www.bayesrulesbook.com/chapter-13.html

For model fitting, you can choose if you want to fit the models using the brms or rstan package functions (or both!). Either way, you will need to investigate how to fit a logistic regression model. Consider using help functions (i.e. check out the family option in brm), consider the resources, and/or do a google search for vignettes or tutorials to do logistic regression with brm or stan.

Choice of priors will be discussed further in part 2. A default recommendation (based on centered covariates) varies across references but generally, distributions with fatter tails (as compared to normal densities) are recommended, such as a t-distribution. For part 1 of the HW, when using brm, you may use brm-default priors (based on centered covariates, the default here is to use a student_t(3, 0, 2.5) for the intercept, flat priors are used for other coefficients). When using stan, you may use the same priors, or consider a student_t(df = 7, location = 0, scale = 2.5), as recommended here https://avehtari.github.io/modelselection/diabetes.html.

For all model fits, include centered covariates (i.e. subtract the mean of the covariate) and make sure Rhat and effectve sample sizes don't suggest any issues.

Some functions

```
invlogit <- function(x) 1/(1+exp(-x))
logit <- function(p) log(p/(1-p))

# A function to slightly jitter the binary data
jitt <- function(...) {
  geom_point(aes_string(...), position = position_jitter(height = 0.05, width = 0.1),
             size = 2, shape = 21, stroke = 0.2)
}
```

Wells data

Information taken from https://cran.r-project.org/web/packages/rstanarm/vignettes/binomial.html. The data are described here https://vincentarelbundock.github.io/Rdatasets/doc/carData/Wells.html

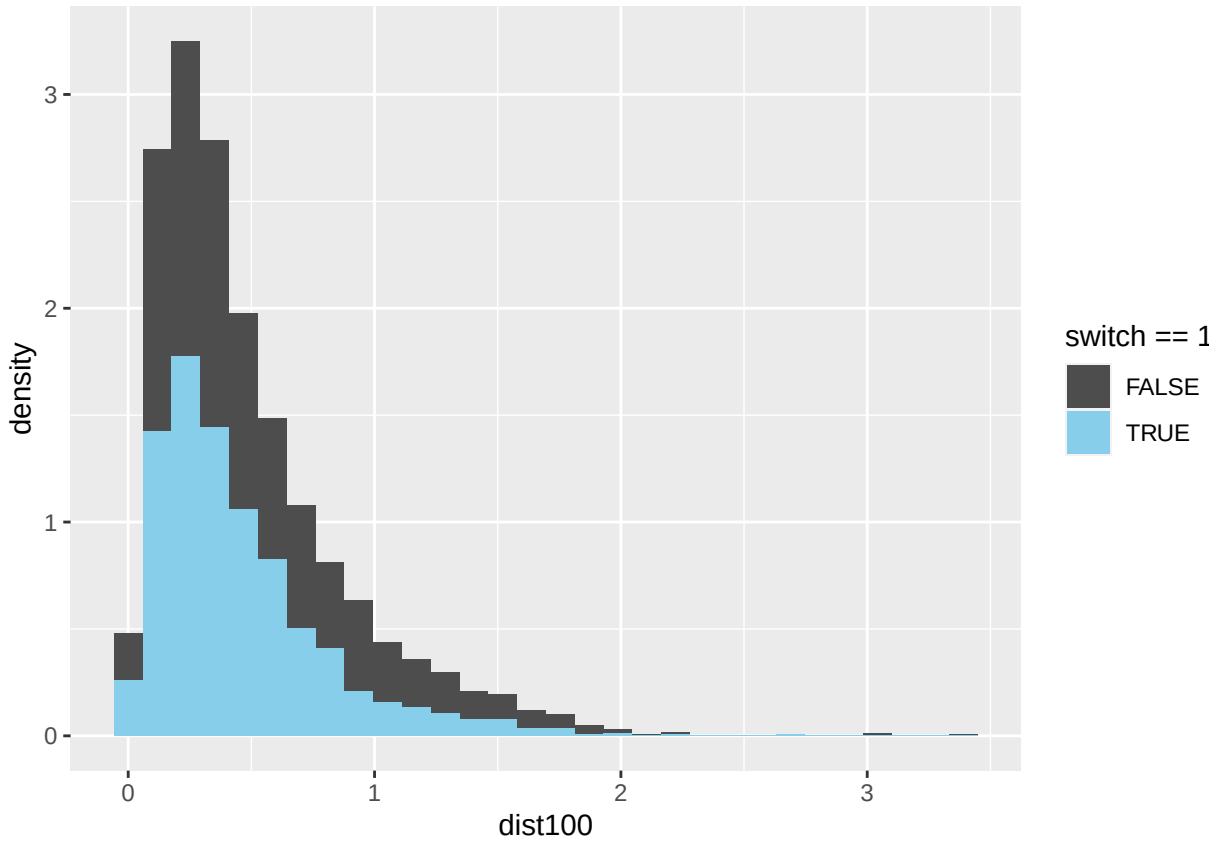
Gelman and Hill describe a survey of 3200 residents in a small area of Bangladesh suffering from arsenic contamination of groundwater. Respondents with elevated arsenic levels in their wells had been encouraged to switch their water source to a safe public or private well in the nearby area and the survey was conducted several years later to learn which of the affected residents had switched wells. The goal of the analysis presented by Gelman and Hill is to learn about the factors associated with switching wells.

Reading in the data and creating some transformed variables:

```
url <- "http://stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat"
wells <- read.table(url)
wells <- wells %>%
  # adding some transformed and centered variables
  mutate(y = switch,
         dist100 = dist / 100,
         # rescale the dist variable (measured in meters) so that it is measured in units of 100 meters
        c_dist100 = dist100 - mean (dist100),
        c_arsenic = arsenic - mean (arsenic))
```

A simple plot: blue bars correspond to the 1737 residents who said they switched wells and darker bars show the distribution of dist100 for the 1283 residents who didn't switch. As we would expect, for the residents who switched wells, the distribution of dist100 is more concentrated at smaller distances.

```
ggplot(wells, aes(x = dist100, y = ..density.., fill = switch == 1)) +
  geom_histogram() +
  scale_fill_manual(values = c("gray30", "skyblue"))
```



Question 1: fitting a logistic regression model (warm-up exercise)

Fit the following simple logistic regression model:

$$\begin{aligned} y_i|\theta_i &\sim Bern(\theta_i), \\ logit(\theta_i) &= \beta_0 + \beta_1 \cdot (d_i - \bar{d}), \end{aligned}$$

where $y_i = 1$ if household i switched wells , 0 otherwise (recorded by the variable `switch` in the dataset), θ_i refers to its probability of switching and d_i to its distance to the nearest safe well (measured in 100 meters, `dist100` in the well dataset).

Report point estimates and 95% CIs for β_0 and β_1. Interpret these estimates in terms of odds ratios.

Solution

```
fit <- brm(y ~ c_dist100, data = wells,
           family = bernoulli(link = "logit"),
           seed = 12,
           chains = 4,
            iter = 2000, thin = 1,
            cores = getOption("mc.cores", 4),
             file = "output/hw4_fit",
           file_refit = "on_change")
```

```
print(fit)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: y ~ c_dist100
##    Data: wells (Number of observations: 3020)
##   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     0.30      0.04     0.23     0.38 1.00     3055     2202
## c_dist100    -0.62      0.10    -0.82    -0.43 1.00     3449     2736
##
## 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).
```

The estimates and 95% CIs for the β's are given by 0.30 (0.23,0.37) and -0.62 (-0.81,-0.43) respectively.

Interpretation:

- The odds of switching are $\exp(0.3) \approx 1.3$ (with CI $\exp((0.23, 0.37))$ for households that are an average distance away from a safe well.
- Odds ratio for slope: $exp(\hat{\beta}_1) = exp(-0.62) \approx 0.54$ thus the odds of switching are 46% lower for a household that lives (x+100) meters away from a safe well as compared to one that lives (x) meters away from it.

Extra notes on interpretation (GH p.89/90P) GH's point 2 and 3:

- Point 2: We can calculate the slope in the inverse-logit function at the mean value of the predictor to find the ? in the statement "1 unit change in x is associated with ? change in the probability of switching around the mean value of x"''". To do so, first note that the derivative of the inverse-logit function

  $$f(x) = 1/(1 + \exp(-(\beta_0 + \beta_1 x)))$$

  is given by

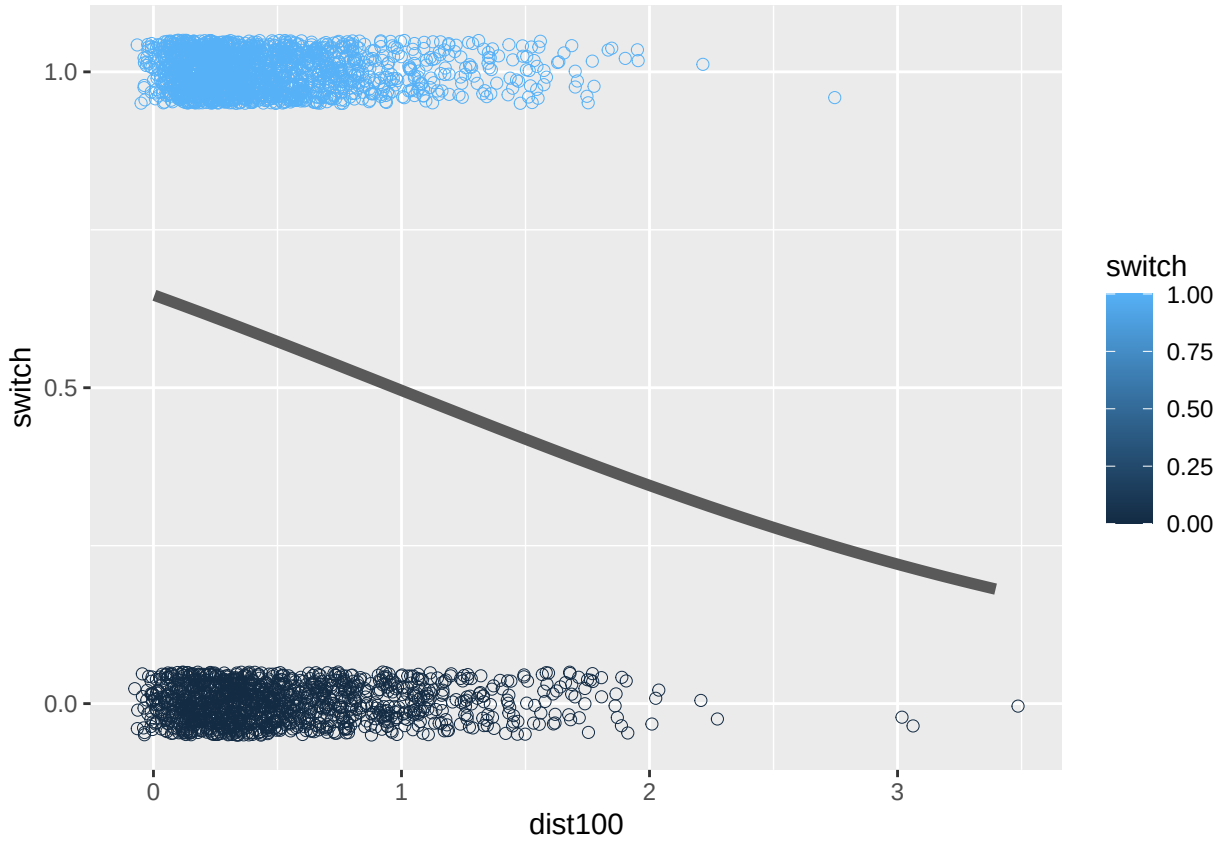
  $$f'(x) = 1/(1 + \exp(-(\beta_0 + \beta_1 x)))^2 \cdot \beta_1 \exp(-(\beta_0 + \beta_1 x)).$$

  The mean value for dist100 is given by 0.48, and $f'(0.48) \approx -0.16$, thus an increase in distance by 100 meters is associated with a decrease of 0.16 in the probability of switching around the mean distance of 48 meter.

- In point 3, GH use their "divide by 4 rule'' to get the maximum slope, which gives -0.62/4 $\approx$ -0.15. This again is based on the derivate (slope) of the inverse-logit function, which is largest (in absolute sense) for $\beta_0 + \beta_1 x = 0$, thus for $x = -\beta_0/\beta_1$, then $f'(-\beta_0/\beta_1) = \beta_1/4$. here $-\beta_0/\beta_1 \approx 0.5$ thus the maximum slope of -0.15 is attained around a distance of 50 meters (close to the mean distance).

```
# Predicted probability as a function of x
pr_switch <- function(x, ests) invlogit(ests[1] + ests[2] * (x - mean(wells$dist100)))
# test
# curve(pr_switch(x, ests = fixef(fit)))
```

```
ggplot(wells, aes(x = dist100, y = switch, color = switch)) +
  scale_y_continuous(breaks = c(0, 0.5, 1)) +
  jitt(x="dist100") +
  stat_function(fun = pr_switch, args = list(ests = fixef(fit)),
                size = 2, color = "gray35")
```



Aside, LA checking priors

```
get_prior(y ~ c_dist100, data = wells, family = bernoulli(link = "logit"))
```

```
##                 prior     class      coef group resp dpar nlpar lb ub
##                (flat)         b
##                (flat)         b c_dist100
##  student_t(3, 0, 2.5) Intercept
##        source
##       default
##  (vectorized)
##       default
```

```
stancode(fit)
```

```
## // generated with brms 2.18.0
## functions {
## }
## data {
```

```
##   int<lower=1> N;  // total number of observations
##   int Y[N];  // response variable
##   int<lower=1> K;  // number of population-level effects
##   matrix[N, K] X;  // population-level design matrix
##   int prior_only;  // should the likelihood be ignored?
## }
## transformed data {
##   int Kc = K - 1;
##   matrix[N, Kc] Xc;  // centered version of X without an intercept
##   vector[Kc] means_X;  // column means of X before centering
##   for (i in 2:K) {
##     means_X[i - 1] = mean(X[, i]);
##     Xc[, i - 1] = X[, i] - means_X[i - 1];
##   }
## }
## parameters {
##   vector[Kc] b;  // population-level effects
##   real Intercept;  // temporary intercept for centered predictors
## }
## transformed parameters {
##   real lprior = 0;  // prior contributions to the log posterior
##   lprior += student_t_lpdf(Intercept | 3, 0, 2.5);
## }
## model {
##   // likelihood including constants
##   if (!prior_only) {
##     target += bernoulli_logit_glm_lpmf(Y | Xc, Intercept, b);
##   }
##   // priors including constants
##   target += lprior;
## }
## generated quantities {
##   // actual population-level intercept
##   real b_Intercept = Intercept - dot_product(means_X, b);
## }
```

Question 2: Models with distance and arsenic

Now consider models that include a second predictor, which is the arsenic level in the respondents' well (called `arsenic` in the dataset). Fit model (2), which has distance/100 and arsenic levels as predictors, as well as model (3), which has both predictors and their interaction term.

Write out the equations for both models, and construct one plot that shows the relation between the estimated switch probability and arsenic levels for both models for households that are 100 meters away from a safe well (use posterior means of the regression coefficients and show the model with the interaction term in a dashed red line). Interpret the difference between the fitted regression lines.

Solution

$$
\begin{aligned}
y_i|\theta_i &\sim Bern(\theta_i), \\
logit(\theta_i) &= \beta_0 + \beta_1 \cdot (d_i - \bar{d}) + \beta_2 \cdot (a_i - \bar{a}), \text{ for model 2,} \\
logit(\theta_i) &= \beta_0 + \beta_1 \cdot (d_i - \bar{d}) + \beta_2 \cdot (a_i - \bar{a}) + \beta_3 \cdot (d_i - \bar{d})(a_i - \bar{a}), \text{ for model 3,}
\end{aligned}
$$

where a_i is the arsenic variable for household i.\

```
fit2 <- brm(y ~ c_dist100 + c_arsenic, data = wells,
           family = bernoulli(link = "logit"),
           seed = 12,
           chains = 4,
            iter = 2000, thin = 1,
            cores = getOption("mc.cores", 4),
             file = "output/hw4_fit2",
           file_refit = "on_change")
```

```
print(fit2)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: y ~ c_dist100 + c_arsenic
##    Data: wells (Number of observations: 3020)
##   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     0.33      0.04     0.26     0.41 1.00     3228     2793
## c_dist100    -0.90      0.10    -1.11    -0.70 1.00     3643     2696
## c_arsenic     0.46      0.04     0.38     0.55 1.00     3564     3130
##
## 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).
```

```
fit3 <- brm(y ~ c_dist100*c_arsenic, data = wells,
           family = bernoulli(link = "logit"),
           seed = 12,
           chains = 4,
            iter = 2000, thin = 1,
            cores = getOption("mc.cores", 4),
             file = "output/hw4_fit3",
           file_refit = "on_change")
```

```
print(fit3)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: y ~ c_dist100 * c_arsenic
##    Data: wells (Number of observations: 3020)
##   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               0.35      0.04     0.27     0.43 1.00     3941     3245
## c_dist100              -0.88      0.10    -1.07    -0.68 1.00     3397     2961
```

```
## c_arsenic               0.47      0.04     0.39     0.56 1.00     4032     2916
## c_dist100:c_arsenic    -0.18      0.10    -0.37     0.03 1.00     4093     2938
##
## 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).
```

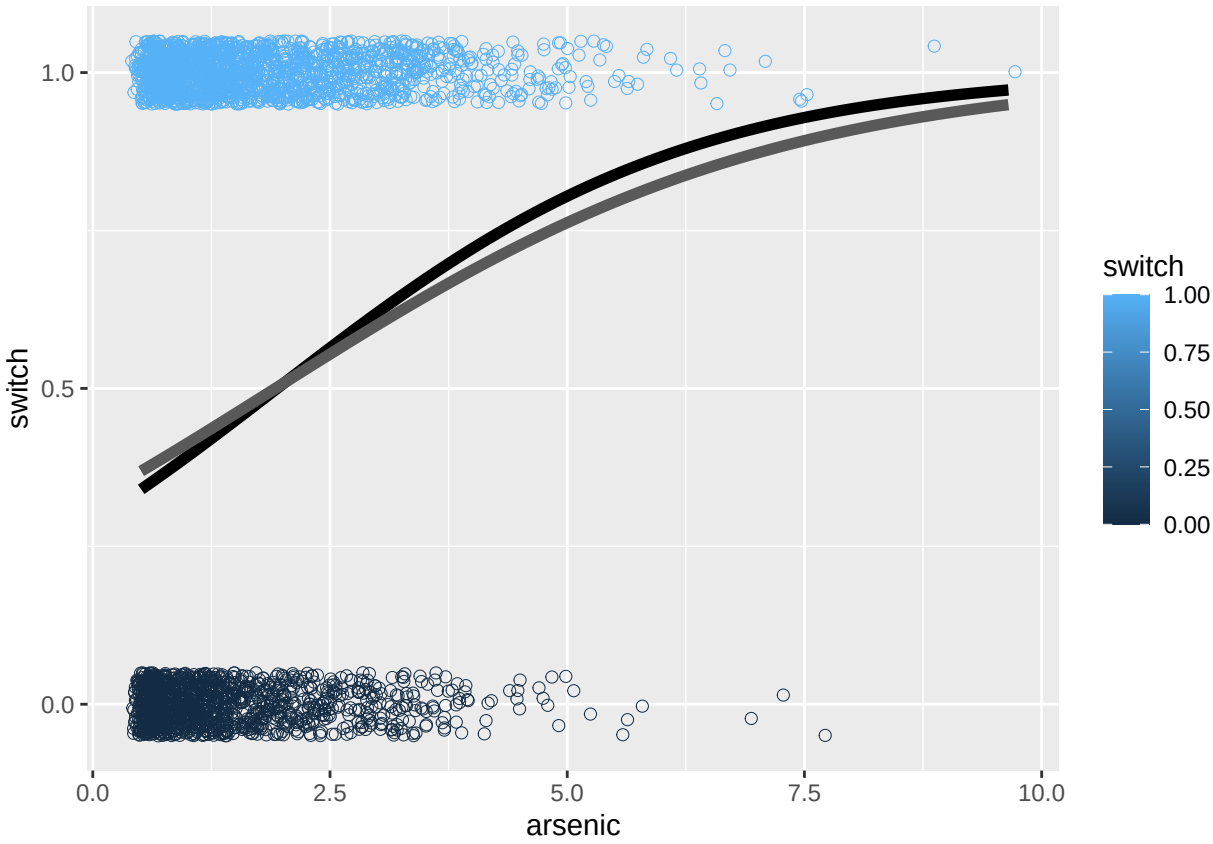
To get the plot below, I used posterior mean estimates of the βs and plugged those into the regression function, eg for the one with the interaction term I used: $InvLogit(\hat{\beta}_0 + \hat{\beta}_1 \cdot (d_i - \bar{d}) + \hat{\beta}_2 \cdot (a_i - \bar{a}) + \hat{\beta}_3 \cdot (d_i - \bar{d})(a_i - \bar{a}))$.

Interpreting the plot turns out to be a little uneventful because the relationships are not that different (sorry, I thought that there was more of a difference but didn't check in detail!). We see similar relations between arsenic and the probability of switching, where the probability increases with arsenic levels. For the model with the interaction term, at higher arsenic, the probability of switching is a little lower.

```
pr_switch_arsenic_fit2 <- function(x, dist = 1, ests)
  invlogit(ests[1] + ests[2] * (dist - mean(wells$dist100)) + ests[3]*(x -  mean(wells$arsenic)))

pr_switch_arsenic_fit3 <- function(x, dist = 1, ests)
  invlogit(ests[1] + ests[2] * (dist - mean(wells$dist100)) + ests[3]*(x -  mean(wells$arsenic)) +
             ests[4]* (dist - mean(wells$dist100))*(x -  mean(wells$arsenic)))

ggplot(wells, aes(x = arsenic, y = switch, color = switch)) +
  scale_y_continuous(breaks = c(0, 0.5, 1)) +
  jitt(x="arsenic") +
  stat_function(fun = pr_switch_arsenic_fit2, args = list(ests = fixef(fit2)[, 'Estimate']),
                size = 2) +
  stat_function(fun = pr_switch_arsenic_fit3, args = list(ests = fixef(fit3)[, 'Estimate']),
                size = 2, color = "gray35")
```



Question 3: Residual plots

Produce residual plots for model 3, to show how residuals in that model vary with distance and arsenic. Start by calculating the residuals as discussed in module 11. Then, because this is logistic regression with binary outcomes, consider how to best display the residuals. Note that just plotting residuals will not result in an informative plot because the y's are binary.

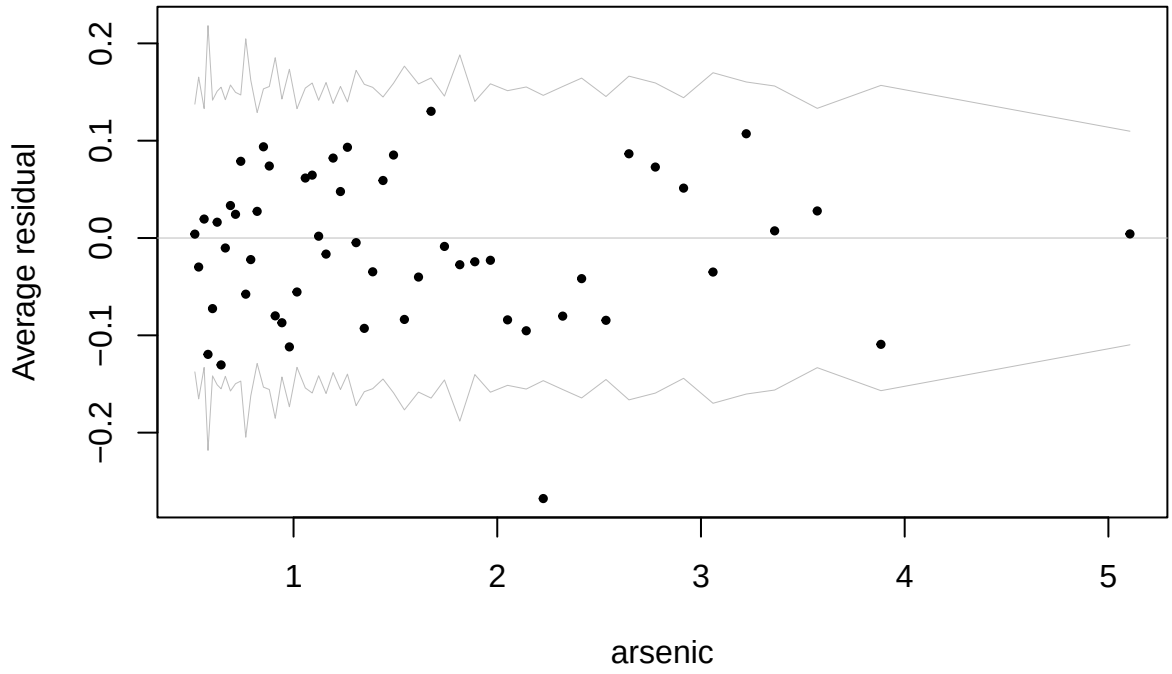
You may be able to find better resources but in case it's still helpful, in my pre-tidyverse and ggplot life, I have used a function from GH for plotting residuals

```
#----
# function for binned residual plots from GH
#-----
binned.resids <- function (x, # what to bin over?
                           y, # what to bin, eg. residuals
                           nclass=sqrt(length(x))){
  breaks.index <- floor(length(x)*(1:(nclass-1))/nclass)
  breaks <- c (-Inf, sort(x)[breaks.index], Inf)
  output <- NULL
  xbreaks <- NULL
  x.binned <- as.numeric (cut (x, breaks))
  for (i in 1:nclass){
    items <- (1:length(x))[x.binned==i]
    x.range <- range(x[items])
    xbar <- mean(x[items])
    ybar <- mean(y[items])
```

```
    n <- length(items)
    sdev <- sd(y[items])
    output <- rbind (output, c(xbar, ybar, n, x.range, 2*sdev/sqrt(n)))
  }
  colnames (output) <- c ("xbar", "ybar", "n", "x.lo", "x.hi", "2se")
  return (list (binned=output, xbreaks=xbreaks))
}
```

Example use for made up residuals

```
n <- length(wells$y)
resid <-  runif(n, -1,1)# just making up something
result <- data.frame(binned.resids(wells$arsenic, resid)$binned)
plot(range(result$xbar), range(result$ybar,result$X2se, -result$X2se),
     ylab="Average residual", type="n", xlab = "arsenic")
abline (0,0, col="gray", lwd=.5)
lines (result$xbar, result$X2se, col="gray", lwd=.5)
lines (result$xbar, -result$X2se, col="gray", lwd=.5)
points (result$xbar,result$ybar, pch=19, cex=.5)
```



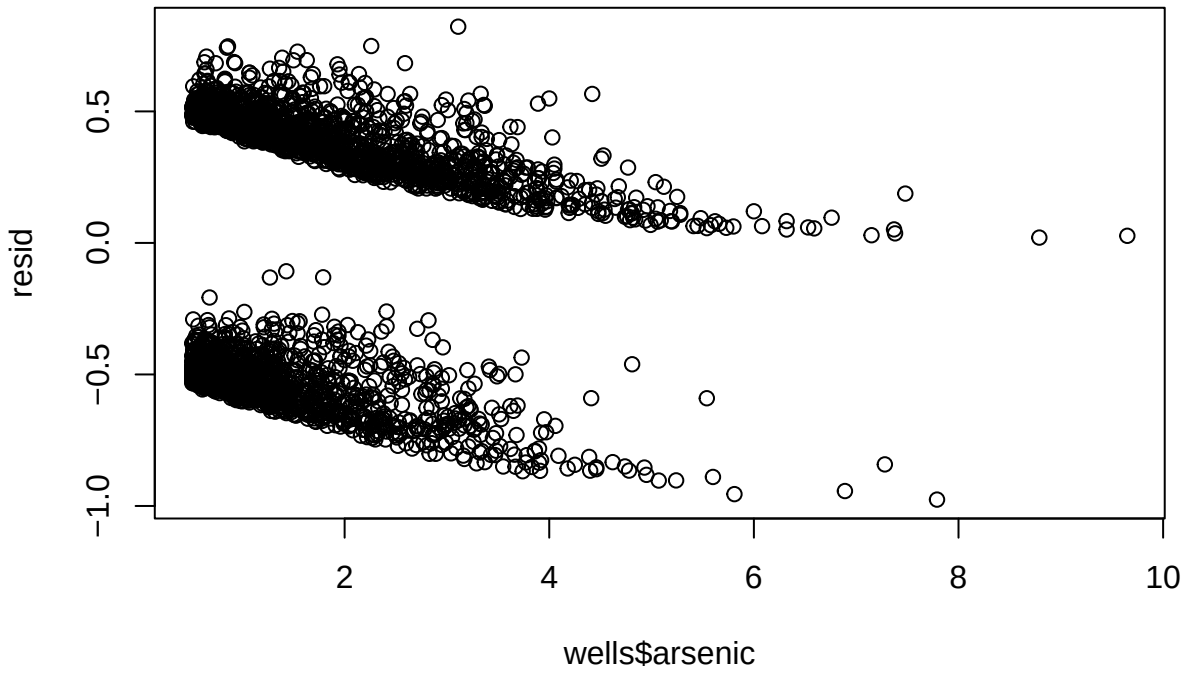
##

Solution

```
ynew_si <- posterior_predict(fit3)
ytildehat_i <- apply(ynew_si,2,mean)
resid <- wells$y - ytildehat_i
```

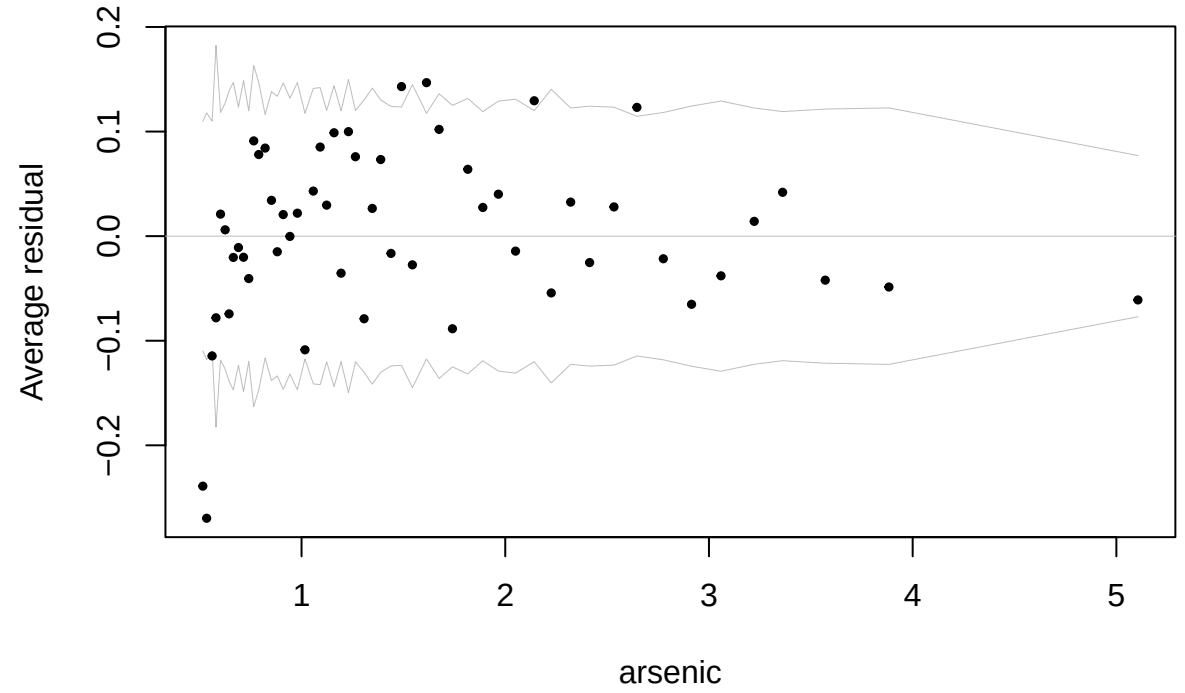
note that just residuals doesn't work, ie

```
plot(resid ~ wells$arsenic)
```

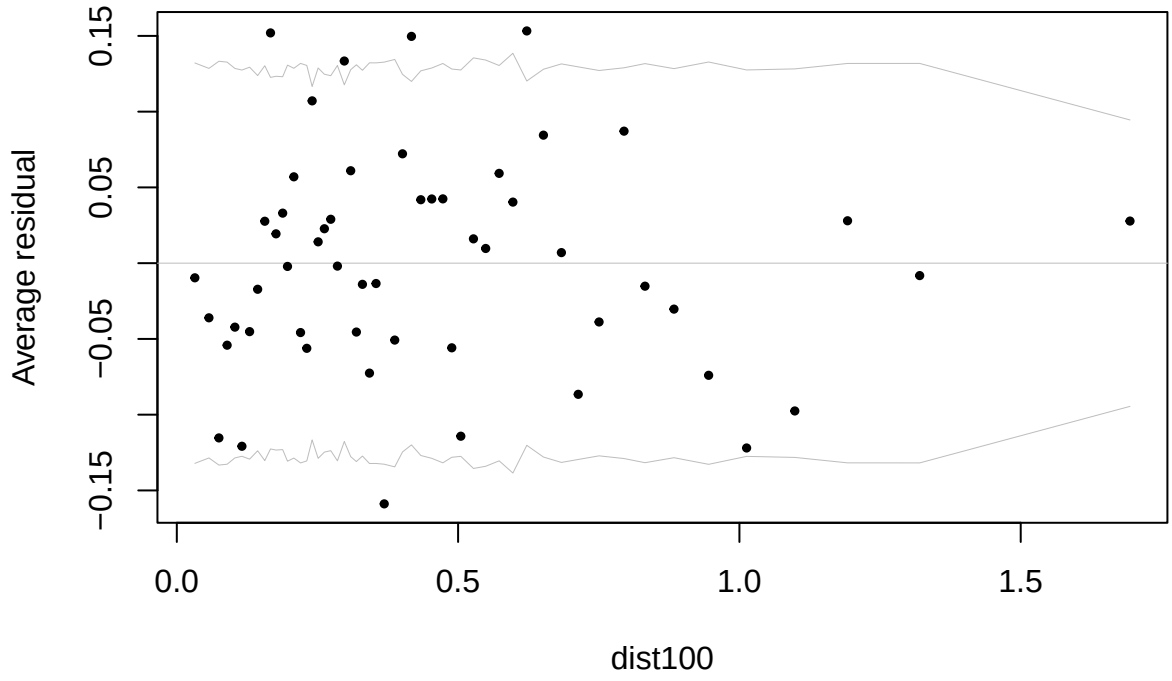


Here is the binned plot for arsenic, note that there may be an issue with a misfit at lower arsenic values.

```
n <- length(wells$y)
result <- data.frame(binned.resids(wells$arsenic, resid)$binned)
plot(range(result$xbar), range(result$ybar,result$X2se, -result$X2se),
     ylab="Average residual", type="n", xlab = "arsenic")
abline (0,0, col="gray", lwd=.5)
lines (result$xbar, result$X2se, col="gray", lwd=.5)
lines (result$xbar, -result$X2se, col="gray", lwd=.5)
points (result$xbar,result$ybar, pch=19, cex=.5)
```



```
n <- length(wells$y)
result <- data.frame(binned.resids(wells$dist100, resid)$binned)
plot(range(result$xbar), range(result$ybar,result$X2se, -result$X2se),
     ylab="Average residual", type="n", xlab = "dist100")
abline (0,0, col="gray", lwd=.5)
lines (result$xbar, result$X2se, col="gray", lwd=.5)
lines (result$xbar, -result$X2se, col="gray", lwd=.5)
points (result$xbar,result$ybar, pch=19, cex=.5)
```



Question 4: Posterior predictive check

The fit of model (3) is not great for low values of arsenic: the probability of switching is overpredicted at very low arsenic levels. To improve model diagnostics, let's consider another model (model 4) where arsenic levels are log-transformed:

$$\begin{aligned} y_i|\theta_i &\sim Bern(\theta_i), \\ logit(\theta_i) &= \beta_0 + \beta_1 \cdot (d_i - \bar{d}) + \beta_2 \cdot (a_i^* - \bar{a}_i^*) + \beta_3 \cdot (d_i - \bar{d})(a_i^* - \bar{a}^*), \text{ for model 4} \end{aligned}$$

where a_i^* refers to log-transformed arsenic.

Suppose that one of the outcomes of interest in this study is predicting whether or not a household that is using a well with "unsafe but relatively low arsenic levels'' (say arsenic levels up to 0.82, which is the 25th percentile of the observed sample of arsenic values) will switch. Carry out a posterior predictive check to verify whether model (3) with arsenic and/or model (4) with log(arsenic) give a reasonable prediction for the proportion of switching households (with arsenic levels less than 0.82).

Hint: specify a summary statistic $T(\boldsymbol{y})$ that summarizes the outcome of interest and calculate $T(\boldsymbol{y})$ for the data set. Then construct replicated data sets $\tilde{\boldsymbol{y}}^{(s)}$ with summary statistics $T(\tilde{\boldsymbol{y}}^{(s)})$ and evaluate how extreme $T(\boldsymbol{y})$ is compared to the sample of $T(\tilde{\boldsymbol{y}}^{(s)})$'s.

Solution

Let $T(\boldsymbol{y}) = 1/n_1 \sum_{i=1}^{n} 1(y_i = 1 | a_i < 0.82)$, with $n_1 = \sum_{i=1}^{n} 1(a_i < 0.82)$, the proportion of households that switch with arsenic level $a_i < 0.82$ (or use exact quantile as done in code below).

The histograms of the replicated summary statistics for models 3 and 4, with arsenic and log(arsenic) as predictors, as shown below. The posterior probability of an outcome less than the one observed in the data, $P(\boldsymbol{y}^{rep} < t(\boldsymbol{y}))$ is 0.5% for model 1 and 31.4% for model 2. Based on this check, we conclude that for model 3 the observed statistic is low relative to data replications while for model, the observed outcome is aligned with what the model predicts. This suggests that predictive performance of model 4 may be better than model 3 for this set of household. Note that this conclusion aligns with that of the residual plots where we saw some outlying residuals at low arsenic for model 3.

Function to construct the summary statistics

```
quantile(wells$arsenic, 0.25)
```

```
##  25%
## 0.82
```

```
quantile(wells$c_arsenic, 0.25)
```

```
##        25%
## -0.8369305
```

```
get_summ_stat <- function(y, c_arsenic)
  sum(y*(c_arsenic <  quantile(c_arsenic, 0.25)))/sum((c_arsenic <  quantile(c_arsenic, 0.25)))
```

Fit the model with log arsenic

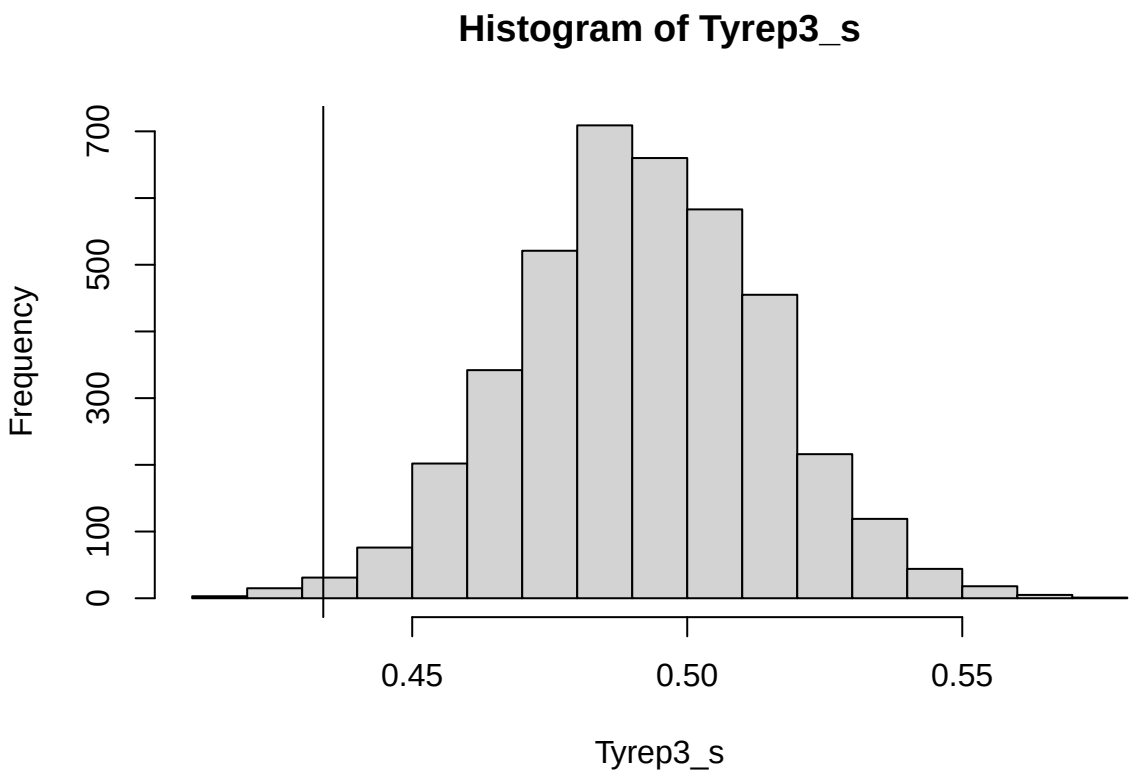
```
wells <-
  wells %>%
  mutate(logarsenic = log(arsenic)) %>%
  mutate(c_logarsenic = logarsenic - mean(logarsenic))

fit4 <- brm(y ~ c_dist100*c_logarsenic, data = wells,
            family = bernoulli(link = "logit"),
            seed = 12,
            chains = 4,
             iter = 2000, thin = 1,
             cores = getOption("mc.cores", 4),
              file = "output/hw4_fit4",
            file_refit = "on_change")
```

```
ynew3_si <- posterior_predict(fit3)
ynew4_si <- posterior_predict(fit4)
```

Histograms with observed value added:

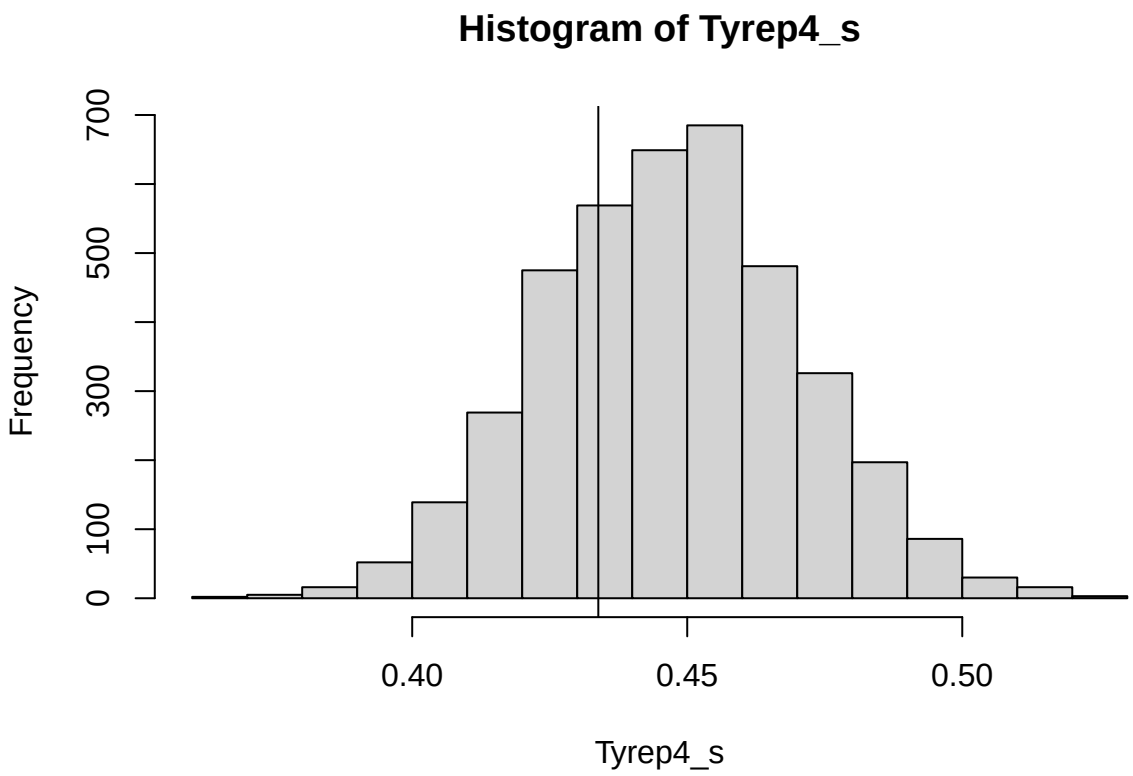
```
Tobs <- get_summ_stat(wells$y, wells$c_arsenic)
Tyrep3_s <- apply(ynew3_si, 1, get_summ_stat, wells$c_arsenic)
hist(Tyrep3_s)
abline(v = Tobs)
```




```
1 - mean(Tobs < Tyrep3_s)
```

```
## [1] 0.007
```

```
Tyrep4_s <- apply(ynew4_si, 1, get_summ_stat, wells$c_arsenic)
hist(Tyrep4_s)
abline(v = Tobs)
```




```
1 - mean(Tobs < Tyrep4_s)
```

```
## [1] 0.2985
```

Question 5: Multilevel logistic regression (extra credit)

According to GH 14.6 (Q2), the observations are obtained in different villages, which makes for a nice extension of the logistic regression model into a multilevel logistic regression model. However, I was not able to find the village grouping in the data sets provided online. To not deprive you from this nice extension and let you fit a multilevel logistic model, go ahead and construct your own groupings as follows:

```
set.seed(12345)
n <- length(wells$y)
# assign households to villages
J <- 300
getj1_i <- c(seq(1,J), sample(size = n-J, x = seq(1,J), replace = TRUE))
getj2_i <- sort(getj1_i) # now the households are assumed to be sorted by village
```

where the first grouping (summarized in 'getj1.i') is random while in the second grouping, the households are grouped in the order at which they appear in the dataset.

Write out in equations an extension for model (2), where each group has its own intercept, that is estimated hierarchically. Then fit the model, using both groupings (so fit the same model twice).

Comment on the difference in resulting fits between using grouping 1 and grouping 2. In particular, do you have any thoughts on why the across-village variance in intercept is smaller for the 1st grouping as compared to the second grouping?

Solution

The model is as follows:

$$\begin{aligned} y_i|\theta_i &\sim Bern(\theta_i), \\ logit(\theta_i) &= \alpha_{j[i]} + \beta_1 \cdot (d_i - \bar{d}) + \beta_2 \cdot (a_i - \bar{a}), \\ \alpha_j|\mu, \sigma_\alpha &\sim N(\mu, \sigma_\alpha^2), \end{aligned}$$

where $j[i]$ refers to the group of observation i and α_j refers to the group-specific intercept.

```
fit_group1 <- brm(y ~ (1|group) + c_dist100 + c_arsenic, data = cbind(wells, group = getj1_i),
           family = bernoulli(link = "logit"),
           seed = 12,
           chains = 4,
            iter = 2000, thin = 1,
            cores = getOption("mc.cores", 4),
             file = "output/hw4_group1_fit",
           file_refit = "on_change")
```

```
print(fit_group1)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: y ~ (1 | group) + c_dist100 + c_arsenic
##    Data: cbind(wells, group = getj1_i) (Number of observations: 3020)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~group (Number of levels: 300)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.18      0.09     0.01     0.35 1.01      502      899
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.34      0.04     0.26     0.41 1.00     4473     2983
## c_dist100    -0.90      0.11    -1.11    -0.69 1.00     4510     3078
## c_arsenic     0.47      0.04     0.38     0.55 1.00     4083     2870
##
## 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).
```

```
fit_group2 <- brm(y ~ (1|group) + c_dist100, data = cbind(wells, group = getj2_i),
           family = bernoulli(link = "logit"),
           seed = 12,
           chains = 4,
            iter = 2000, thin = 1,
            cores = getOption("mc.cores", 4),
             file = "output/hw4_group2_fit",
           file_refit = "on_change")
```

```
print(fit_group2)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: y ~ (1 | group) + c_dist100
##    Data: cbind(wells, group = getj2_i) (Number of observations: 3020)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~group (Number of levels: 300)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.11      0.08     0.96     1.27 1.00     1641     2074
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.41      0.08     0.26     0.56 1.00     1914     2523
## c_dist100    -0.72      0.13    -0.97    -0.46 1.00     5715     3355
##
## 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).
```

For the first grouping, $\hat{\sigma}_\alpha = 0.18$ with 95% CI given by (0.26,0.42), while for the 2nd grouping, the point estimate is 1.11 with 95% CI (0.95, 1.27).

Thoughts on what's going on: Most likely, the households in the data set were sorted by village. In the second grouping, this order is remained such that households within the same village are more likely to end up in the same group (as they should).