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.

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

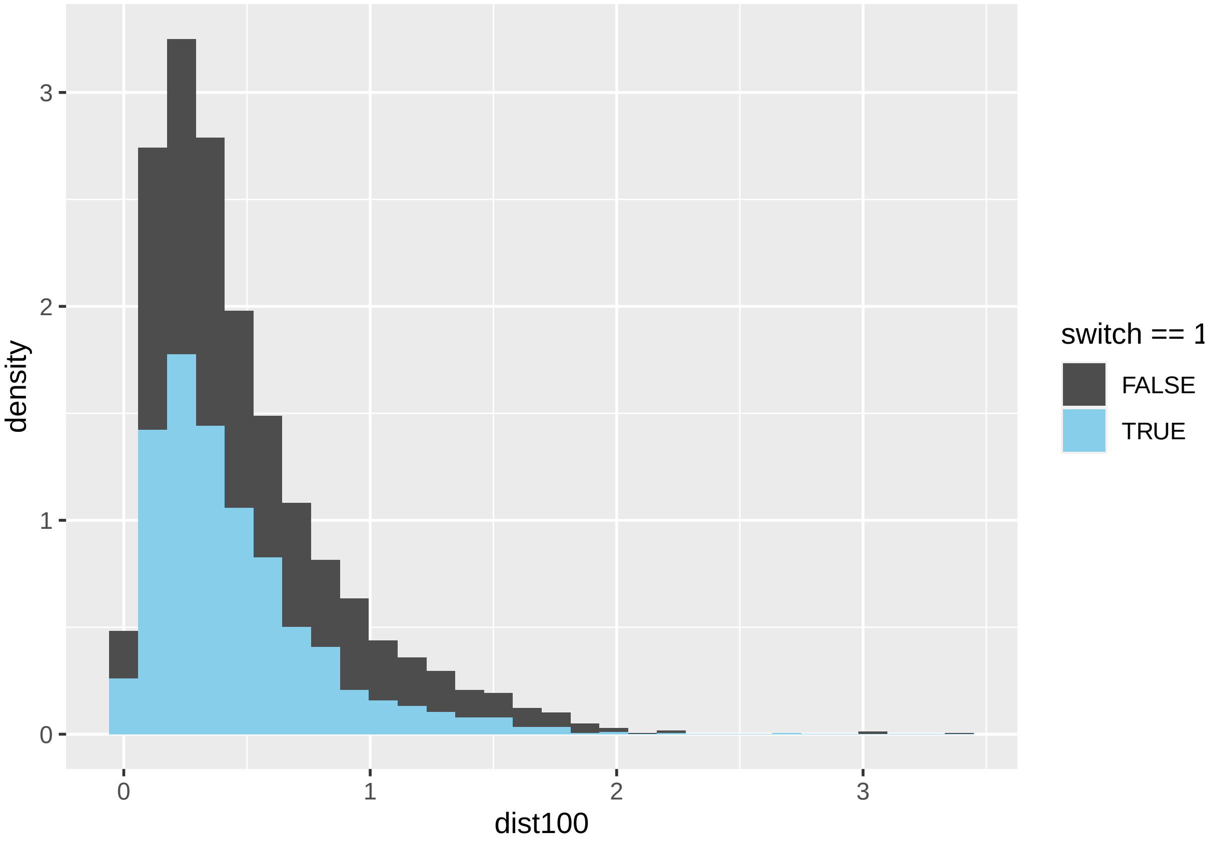
Gelman and Hill describe a survey of 3200 residents in a small area of Bangladesh suffering from arsenic contamination of groundwater. Respondents with elevated arsenic levels in their wells had been encouraged to switch their water source to a safe public or private well in the nearby area and the survey 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:

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

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.

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.

Question 3: Residual plots

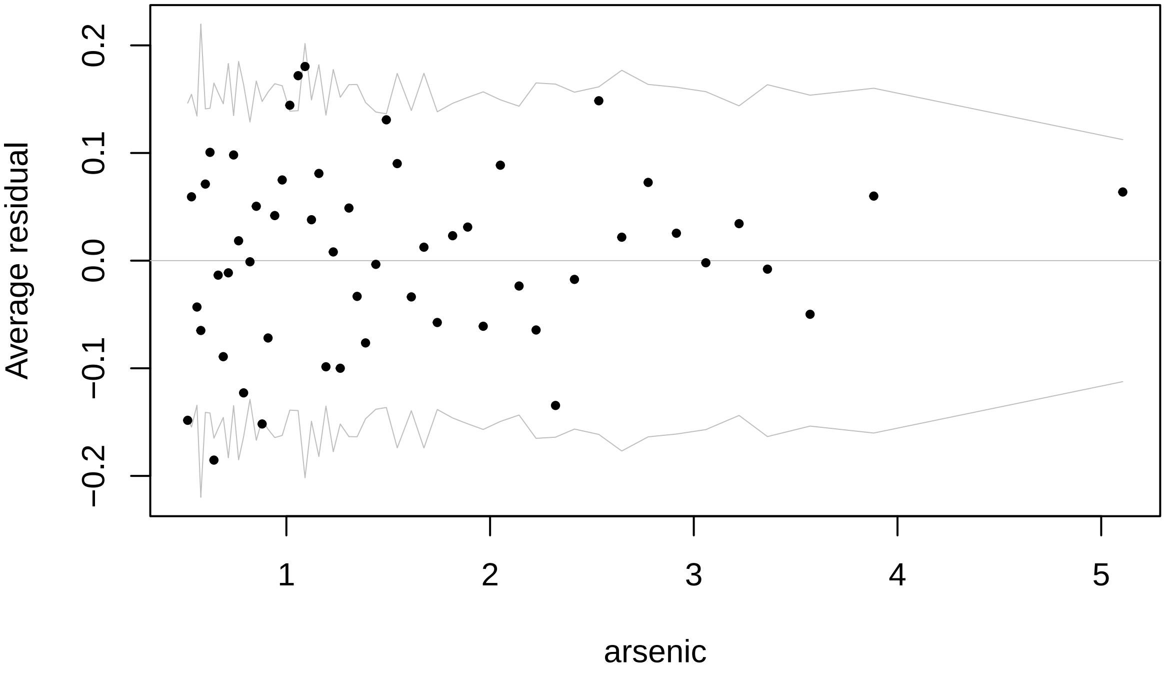
Produce residual plots for model 3, to show how residuals in that model vary with distance and arsenic. Start by calculating the residuals as discussed in module 11. Then, because this is logistic regression with binary outcomes, consider how to best display the residuals. Note that just plotting residuals 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

```r
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)
```



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:

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

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.

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?