# Lab #10 - Logistic Regression Part II

*Econ 224*

*September 27th, 2018*

## Contaminated Wells in Bangladesh

Today we'll work with a dataset containing household-level information from Bangladesh: `wells.csv`. You can download the dataset from the course website at http://ditraglia.com/econ224/wells.csv. Here is some background on the dataset from Gelman and Hill (2007):

> Many of the wells used for drinking water in Bangladesh and other South Asian countries are contaminated with natural arsenic ... a research team from the United States and Bangladesh measured all the wells [in a small region] and labeled them with their arsenic level as well as a characterization of "safe" (below 0.5 in units of hundreds of micrograms per liter, the Bangladesh standard for arsenic in drinking water) or "unsafe" (above 0.5). People with unsafe wells were encouraged to switch to nearby private or community wells or to new wells of their own construction. A few years later, the researchers returned to find out who had switched wells.

Our goal is to predict which households will switch wells using the following information:

| Name | Description |
|---|---|
| dist | Distance to closest known safe well (meters) |
| arsenic | Arsenic level of respondent's well (100s of micrograms per liter) |
| switch | Dummy variable equal to 1 if switched to a new well |
| assoc | Dummy variable equal to 1 if any member of the household is active in community organizations |
| educ | Education level of head of household (years) |

To be clear, our dataset contains *only* information for households with an arsenic level of 0.5 or above, as these are the households that were encouraged to switch wells.

## Exercises

1. Preliminaries:
   (a) Load the data and store it in a tibble called `wells`.
   (b) Use `dplyr` to create a variable called `larsenic` that equals the natural logarithm of `arsenic`.
   (c) Use `ggplot2` to make a histogram of `arsenic` and `larsenic`. Be sure to label your plots appropriately. Comment on your findings.
   (d) Use `dplyr` to create a variable called `dist100` that contains the same information as `dist` but measured in *hundreds of meters* rather than in meters.
   (e) Use `dplyr` to create a variable called `zeduc` that equals the *z-score* of `educ`.
2. First Regression: `fit1`
   (a) Run a logistic regression using `dist100` to predict `switch` and store the result in an object called `fit1`.

(b) Use `ggplot2` to plot the logistic regression function from part (a) along with the data, jittered appropriately.
(c) Discuss your results from parts (a)-(b). In particular: based on `fit1`, is `dist100` a statistically significant predictor of `switch`? Does the sign of its coefficient make sense? Explain.
(d) Use `fit1` to calculate the predicted probability of switching wells for the *average* household in the dataset.
(e) Use `fit1` to calculate the marginal effect of `dist100` for the *average* household in the dataset. Interpret your result. How does is compare to the "divide by 4" rule?

3. Predictive performance of `fit1`

(a) Add a column called `p1` to `wells` containing the predicted probabilities that `switch` equals one based on `fit1`.
(b) Add a column called `pred1` to `wells` that gives the predicted *values* of $y$ that correspond to `p1`.
(c) Use `pred1` to calculate the *Bayes error rate* of `fit1` based on the full training dataset, i.e. `wells`. Hint: you can do this using the `summarize` function from `dplyr`.
(d) Use `pred1` to construct the *confusion matrix* for `fit1`. Hint: use the function `table`.
(e) Based on your results from (d), calculate the *sensitivity* and *specificity* of the predictions from `fit1`.
(f) Comment on your results from (c) and (e). In particular, compare them to the error rate that you would obtain by simply predicting the *most common* value of `switch` for every observation in the dataset. (This is called the "null model" since it doesn't use any predictors.)

4. Additional regressions: `fit2`, `fit3`, and `fit4`

(a) Run a logistic regression using `larsenic` to predict `switch` and store the results in an object called `fit2`.
(b) Run a logistic regression using `zeduc` to predict `switch` and store the results in an object called `fit3`.
(c) Run a logistic regression using `dist100`, `larsenic`, and `zeduc` to predict `switch` and store the results in an object called `fit4`.
(d) Use `stargazer` to make a nicely formatted summary table of the results from `fit1`, `fit2`, `fit3`, and `fit4`. Make sure to add appropriate labels and captions, use a reasonable number of decimal places, etc.

5. Interpreting `fit2`, `fit3` and `fit4`

(a) Repeat parts (b) and (c) of question #2 above with `fit2` in place of `fit1`.
(b) Repeat parts (b) and (c) of question #2 above with `fit3` in place of `fit1`.
(c) Calculate the marginal effect of *each predictor* in `fit4` for the *average* household in the dataset. Interpret your results. How do they compare to the "divide by 4" rule?

6. Predictive Performance of `fit4`

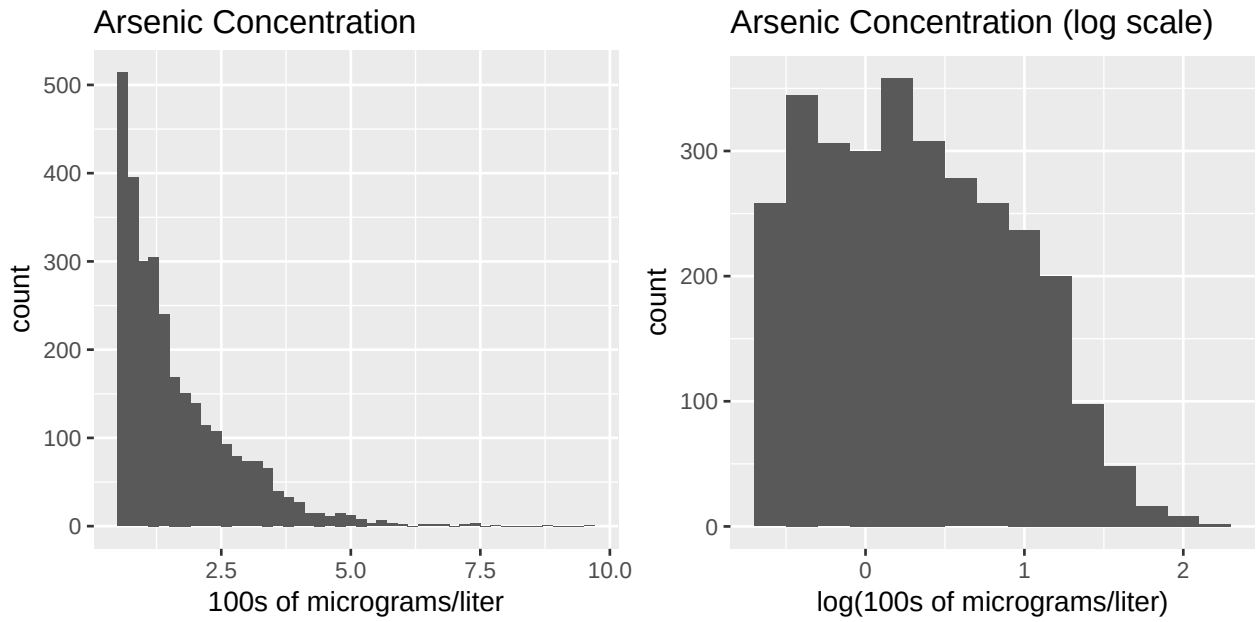
(a) Repeat question #3 from above with `fit4`, `p4`, and `pred4` in place of `fit1`, `p1` and `pred1`.
(b) Using your results from (a) and question #3 above, compare the in-sample predictive performance of `fit1` and `fit4`.

# Solutions

## 1 - Preliminaries

```
#------- Load data
library(gridExtra)
library(tidyverse)
```

```
library(ggplot2)
wells <- read_csv('~/econ224/labs/wells.csv')

#------- Create log arsenic variable
wells <- wells %>%
  mutate(larsenic = log(arsenic))

#------- Histogram of arsenic and larsenic
arsenic_hist <- ggplot(wells) +
  geom_histogram(aes(x = arsenic), binwidth = 0.2) +
  xlab('100s of micrograms/liter') +
  ggtitle('Arsenic Concentration')

larsenic_hist <- ggplot(wells) +
  geom_histogram(aes(x = larsenic), binwidth = 0.2) +
  xlab('log(100s of micrograms/liter)') +
  ggtitle('Arsenic Concentration (log scale)')

grid.arrange(arsenic_hist, larsenic_hist, ncol = 2)
```



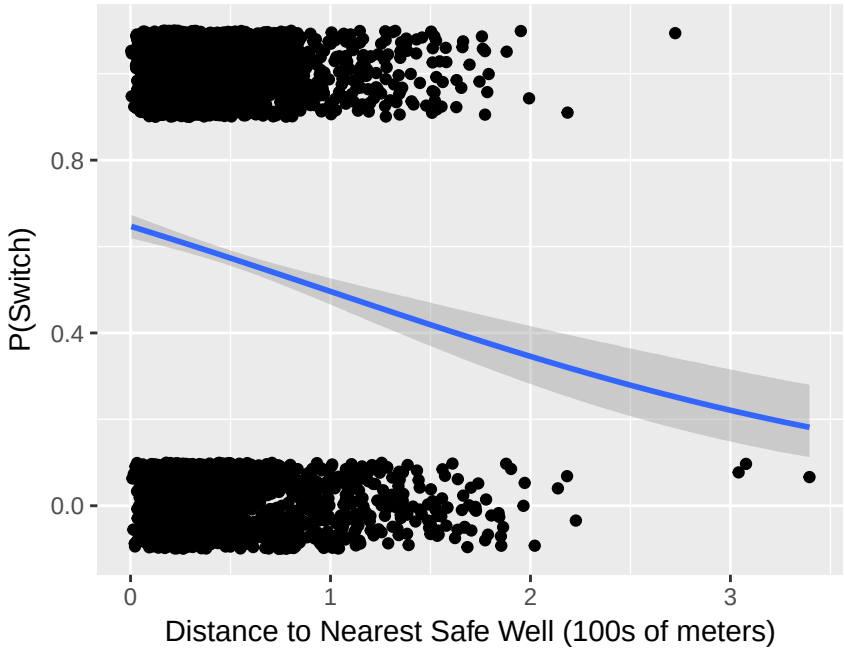

```
#-------- Create dist100 and zeduc
wells <- wells %>%
  mutate(dist100 = dist / 100,
         zeduc = scale(educ))
```

## 2 - First Regression: `fit1`

```
#---------- Generate and summarize fit1
fit1 <- glm(switch ~ dist100, family = binomial(link = 'logit'), wells)
summary(fit1)
```

```
Call:
glm(formula = switch ~ dist100, family = binomial(link = "logit"),
    data = wells)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4406  -1.3058   0.9669   1.0308   1.6603

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.60596    0.06031  10.047  < 2e-16 ***
dist100     -0.62188    0.09743  -6.383 1.74e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4118.1  on 3019  degrees of freedom
Residual deviance: 4076.2  on 3018  degrees of freedom
AIC: 4080.2

Number of Fisher Scoring iterations: 4
```

```r
#---------- Plot fit1 with jittering
ggplot(wells, aes(x = dist100, y = switch)) +
  geom_jitter(height = 0.1) +
  stat_smooth(method='glm',
              method.args = list(family = "binomial"),
              formula = y ~ x) +
  xlab('Distance to Nearest Safe Well (100s of meters)') +
  ylab("P(Switch)")
```



```r
#--------- Prob(switch) at the avg of dist100
avgdist <- wells %>%
  summarize(avgdist = mean(dist100)) %>%
  pull(avgdist)
predict(fit1, newdata = data.frame(dist100 = avgdist), type = 'response')
```

```
        1
0.5757602
```

```r
#--------- Marginal effect of dist100 for the average household
```

## 3 - Predictive Performance of `fit1`

```r
#---------- In-sample predictions from fit1
wells <- wells %>%
  mutate(p1 = predict(fit1, type = 'response'),
         pred1 = 1 * (p1 > 0.5))

#---------- In-sample Bayes Error Rate from fit1
wells %>%
  summarize(error_rate = mean((pred1 == 1 & switch == 0) | (pred1 == 0 & switch == 1)))
```

```
# A tibble: 1 x 1
  error_rate
       <dbl>
1      0.405
```

```r
#---------- Confusion Matrix for fit1
confusion1 <- wells %>%
  select(switch, pred1) %>%
  table
confusion1
```

```
      pred1
switch    0    1
     0  194 1089
     1  133 1604
```

```r
#---------- Sensitivity
confusion1[2,2] / (confusion1[2,1] + confusion1[2,2])
```

```
[1] 0.9234312
```

```r
#---------- Specificity
confusion1[1,1] / (confusion1[1,1] + confusion1[1,2])
```

```
[1] 0.1512081
```

## 4 - Additional Regressions: `fit2`, `fit3`, and `fit4`

```
library(stargazer)
fit2 <- glm(switch ~ larsenic, wells, family = binomial(link = 'logit'))
fit3 <- glm(switch ~ zeduc, wells, family = binomial(link = 'logit'))
fit4 <- glm(switch ~ dist100 + larsenic + zeduc, wells, family = binomial(link = 'logit'))
stargazer(fit1, fit2, fit3, fit4,
          type = 'latex', digits = 2,
          title = 'Logistic Regression Results',
          header = FALSE)
```

Table 2: Logistic Regression Results

| | *Dependent variable:* | | | |
|---|---|---|---|---|
| | switch | | | |
| | (1) | (2) | (3) | (4) |
| dist100 | −0.62*** | | | −0.98*** |
| | (0.10) | | | (0.11) |
| larsenic | | 0.71*** | | 0.89*** |
| | | (0.06) | | (0.07) |
| zeduc | | | 0.16*** | 0.17*** |
| | | | (0.04) | (0.04) |
| Constant | 0.61*** | 0.10** | 0.30*** | 0.53*** |
| | (0.06) | (0.04) | (0.04) | (0.06) |
| Observations | 3,020 | 3,020 | 3,020 | 3,020 |
| Log Likelihood | −2,038.12 | −1,994.64 | −2,050.19 | −1,939.08 |
| Akaike Inf. Crit. | 4,080.24 | 3,993.29 | 4,104.39 | 3,886.15 |
| *Note:* | | | | *p<0.1; **p<0.05; ***p<0.01 |