Coliform Contamination in Private Well Water on the Crow Reservation: Supplement

Michaela Powell

How are the odds of coliform detection related to a well being in alluvium, after accounting for distance to the nearest river and well depth?

Recall that we fit a binary logistic regression model with the following structure:

$$logit(\pi_i) = \beta_0 + \beta_1 WD_i + \beta_2 DR_i + \beta_3 I_{alluvium_i} + \beta_4 DR_i * I_{alluvium_i}$$

Where:

$$I_{alluvium_i} = \begin{cases} 1 & if\ the\ well\ is\ in\ alluvium \\ 0 & if\ the\ well\ is\ not\ in\ alluvium \end{cases}$$

For a well not in alluvium it simplifies to:

$$logit(\pi_i | Not\ Alluvium) = \beta_0 + \beta_1 WD_i + \beta_2 DR_i$$

and, for a well in alluvium it simplifies to:

$$logit(\pi_i | Alluvium) = (\beta_0 + \beta_3) + \beta_1 WD_i + (\beta_2 + \beta_4) DR_i$$

To find the parameter representing how the odds of coliform detection differ between wells in alluvium and wells not in alluvium, after accounting for well depth and the distance to the nearest river, we can take the difference of the simplified models.

- The difference is the parameter representing the difference in the *log-odds* of contamination between wells in alluvium and wells not in alluvium, for a fixed well depth and distance to the nearest river.
- The exponentiated difference is the parameter representing the ratio of the *odds* of contamination between wells in alluvium and wells not in alluvium, for a fixed well depth and distance to the nearest river.

The difference (Alluvium - Not Alluvium) is:

$$\Big[(\beta_0 + \beta_3) + \beta_1 WD_i + (\beta_2 + \beta_4) DR_i\Big] - \Big[\beta_0 + \beta_1 WD_i + \beta_2 DR_i\Big] = \beta_3 + \beta_4 DR_i,$$

which depends on DR_i because of the interaction in the model between the alluvium indicator and DR_i.

To construct a confidence interval for this difference, the variance of the estimated quantity $\hat{\beta}_3 + \hat{\beta}_4 DR_i$ is needed. The details will not be explained, but the following is true:

$$Var(\hat{\beta}_3 + \hat{\beta}_4 DR_i) = Var(\hat{\beta}_3) + (DR_i)^2 Var(\hat{\beta}_4) + (2)(DR_i) Cov(\hat{\beta}_3, \hat{\beta}_4)$$

See Section 10.4.3 in the Statistical Sleuth for the details of this formula (Ramsey and Schafer 2012, 293–94).

The estimate and variance of the estimate depend on DR_i, therefore a distance of interest (or several) must be chosen. In this example we will use 105, 250, and 470 feet - approximately the first, second, and third quartiles of the observed distances to the nearest river - but practically meaningful distances should be chosen by the client. The formal "Coliform Contamination in Private Well Water on the Crow Reservation" report can be referenced for the example interpretations of the exponentiated point estimates and exponentiated confidence intervals.

```r
# LOAD REQUIRED PACKAGES

library(knitr)
library(ggplot2)


# LOAD DATA AND KEEP NECESSARY COLUMNS ONLY

well <- read.csv("crowhomewelldata4.csv")
well <- well[,c("MPN_100mL_1_Coliform", "MPN_100mL_2_Coliform", "MPN_100mL_3_Coliform",
                "Well_Depth_ft", "Alluvium_or_Not", "Near_Distance_Stream")]


# CONSTRUCT BINARY CONTAMINATED OR NOT CONTAMINATED VARIABLE

well$sum_MPN <- (well$MPN_100mL_1_Coliform + well$MPN_100mL_2_Coliform
                 + ifelse(is.na(well$MPN_100mL_3_Coliform)==TRUE, 0, well$MPN_100mL_3_Coliform))

well$contaminated <- ifelse(well$sum_MPN==0, 0, 1)


# CONSTRUCT AUQIFER TYPE VARIABLE WITH 'NOT ALLUVIUM' AS BASELINE

well$Auqifer_Type <- relevel(well$Alluvium_or_Not, ref="Not Alluvium")


# FIT THE MODEL

m1 <- glm(data=well, contaminated ~ Well_Depth_ft + Near_Distance_Stream * Auqifer_Type,
          family=binomial(link="logit"))


# EXTRACT THE NECESSARY PARAMETER ESTIMATES AND VARIANCES/COVARIANCES FROM THE MODEL SUMMARY

beta3 <- summary(m1)$coef[4,1]
beta4 <- summary(m1)$coef[5,1]

var_beta3 <- vcov(m1)[4,4]
var_beta4 <- vcov(m1)[5,5]
cov_beta3_beta4 <- vcov(m1)[4,5]


# WRITE A FUNCTION TO CALCULATE THE POINT ESTIMATE

log.odds.difference_point <- function(DR, B3, B4) {B3 + B4*DR}


# WRITE A FUNCTION TO CALCULATE THE VARIANCE OF THE POINT ESTIMATE

log.odds.difference_var <- function(DR, VarB3, VarB4, CovB3B4) {VarB3 + (DR^2)*VarB4 + 2*DR*CovB3B4}


# CALCULATE/STORE ESTIMATE, VARIANCE, AND LOWER/UPPER BOUNDS OF 95% CI FOR DR=105

estimate_105 <- log.odds.difference_point(105, beta3, beta4)
var_105 <- log.odds.difference_var(105, var_beta3, var_beta4, cov_beta3_beta4)
lb_ci_105 <- estimate_105 - qnorm(0.975) * sqrt(var_105)
ub_ci_105 <- estimate_105 + qnorm(0.975) * sqrt(var_105)
```

```
# CALCULATE/STORE ESTIMATE, VARIANCE, AND LOWER/UPPER BOUNDS OF 95% CI FOR DR=250

estimate_250 <- log.odds.difference_point(250, beta3, beta4)
var_250 <- log.odds.difference_var(250, var_beta3, var_beta4, cov_beta3_beta4)
lb_ci_250 <- estimate_250 - qnorm(0.975) * sqrt(var_250)
ub_ci_250 <- estimate_250 + qnorm(0.975) * sqrt(var_250)


# CALCULATE/STORE ESTIMATE, VARIANCE, AND LOWER/UPPER BOUNDS OF 95% CI FOR DR=470

estimate_470 <- log.odds.difference_point(470, beta3, beta4)
var_470 <- log.odds.difference_var(470, var_beta3, var_beta4, cov_beta3_beta4)
lb_ci_470 <- estimate_470 - qnorm(0.975) * sqrt(var_470)
ub_ci_470 <- estimate_470 + qnorm(0.975) * sqrt(var_470)


# CREATE/PRINT TABLE OF ESTIMATES, VARIANCES, AND LOWER/UPPER BOUNDS

distance <- c(105, 250, 470)
estimated.difference_log.odds <- c(estimate_105, estimate_250, estimate_470)
variance_estimated.difference_log.odds <- c(var_105, var_250, var_470)
estimated.difference_log.odds_95.ci.lb <- c(lb_ci_105, lb_ci_250, lb_ci_470)
estimated.difference_log.odds_95.ci.ub <- c(ub_ci_105, ub_ci_250, ub_ci_470)

summary <- as.data.frame((cbind(distance, estimated.difference_log.odds,
                                variance_estimated.difference_log.odds,
                                estimated.difference_log.odds_95.ci.lb,
                                estimated.difference_log.odds_95.ci.ub)))

names <- c("Distance", "Estimate", "Variance of Estimate",
           "Lower Bound of 95% CI", "Upper Bound of 95% CI")
kable(summary, col.names=names)
```

Distance	Estimate	Variance of Estimate	Lower Bound of 95% CI	Upper Bound of 95% CI
105	0.132	0.479	-1.224	1.49
250	0.732	0.425	-0.545	2.01
470	1.642	0.638	0.076	3.21

```
# CREATE/PRINT TABLE OF EXPONENTIATED ESTIMATES AND LOWER/UPPER BOUNDS

`Distance` <- distance
`Exponentiated Estimate` <- exp(estimated.difference_log.odds)
`Exponentiated Lower Bound` <- exp(estimated.difference_log.odds_95.ci.lb)
`Exponentiated Upper Bound` <- exp(estimated.difference_log.odds_95.ci.ub)

ExponentiatedSummary <- as.data.frame(cbind(`Distance`, `Exponentiated Estimate`,
                                            `Exponentiated Lower Bound`,
                                            `Exponentiated Upper Bound`))
kable(ExponentiatedSummary)
```

Distance	Exponentiated Estimate	Exponentiated Lower Bound	Exponentiated Upper Bound
105	1.14	0.294	4.43
250	2.08	0.580	7.46
470	5.17	1.079	24.74

The following code calculates the exponentiated point estimates and exponentiated confidence intervals for the sequence of distances $(0,\ 20,\ 40, ..., 1660)$. The plot of the results can be found on the following page.

```
# STORE SEQUENCE OF DIFFERENCES

distances <- as.array(seq(0,1660,by=20))


# CALCULATE/STORE POINT ESTIMATES AND VARIANCES OF THE POINT ESTIMATES

vector_estimates <- apply(distances, MARGIN=1, FUN=log.odds.difference_point, B3=beta3, B4=beta4)

vector_variances <- apply(distances, MARGIN=1, FUN=log.odds.difference_var,
                          VarB3=var_beta3, VarB4=var_beta4, CovB3B4=cov_beta3_beta4)


# CALCULATE/STORE LOWER/UPPER BOUNDS OF 95% CIS

vector_lower.bounds <- vector_estimates - qnorm(0.975) * sqrt(vector_variances)
vector_upper.bounds <- vector_estimates + qnorm(0.975) * sqrt(vector_variances)


# EXPONENTIATE ESTIMATES AND LOWER/UPPER BOUNDS

vector_exp.point.estimates <- exp(vector_estimates)
vector_exp.lower.bounds <- exp(vector_lower.bounds)
vector_exp.upper.bounds <- exp(vector_upper.bounds)


# BIND EXPONENTIATED ESTIMATES AND LOWER/UPPER BOUNDS INTO A DATAFRAME

exp_summary <- as.data.frame(cbind(distances, vector_exp.point.estimates, vector_exp.lower.bounds,
                                   vector_exp.upper.bounds))


# RESHAPE DATAFRAME FROM LONG TO WIDE FORMAT

exp_summary_long <- reshape(exp_summary, varying=c("vector_exp.point.estimates",
                                                   "vector_exp.lower.bounds",
                                                   "vector_exp.upper.bounds"),
                            timevar="Estimate", direction="long", sep = "_")


# PLOT EXPONENTIATED ESTIMATES AND LOWER/UPPER BOUNDS FOR THE SEQUENCE OF DISTANCES

ggplot(exp_summary_long, aes(x=distances, y=vector)) +
  scale_y_continuous(trans='log10', breaks=c(-1,0,1,10,100,1000,10000,100000,1000000)) +
  geom_point(aes(shape=Estimate), show.legend=FALSE, size=0.7) +
  scale_shape_manual(values=c(18,15,18)) +
  geom_vline(xintercept=440, lty="longdash", size=0.2, color="gray8") +
  annotate("text", x=540, y=500000, label="440 feet", size=3, color="gray8") +
  geom_rug(data=well, aes(x=Near_Distance_Stream, color=Auqifer_Type), inherit.aes = F) +
  scale_color_manual(values=c("#E31A1C", "dodgerblue2")) +
  labs(x="Distance to the Nearest River \n (in feet)",
       y="Ratio of the Odds of Detection \n (Alluvium / Not in Alluvium)",
       color="Auqifer Type of Observed Wells:") +
  theme(legend.position="top")
```

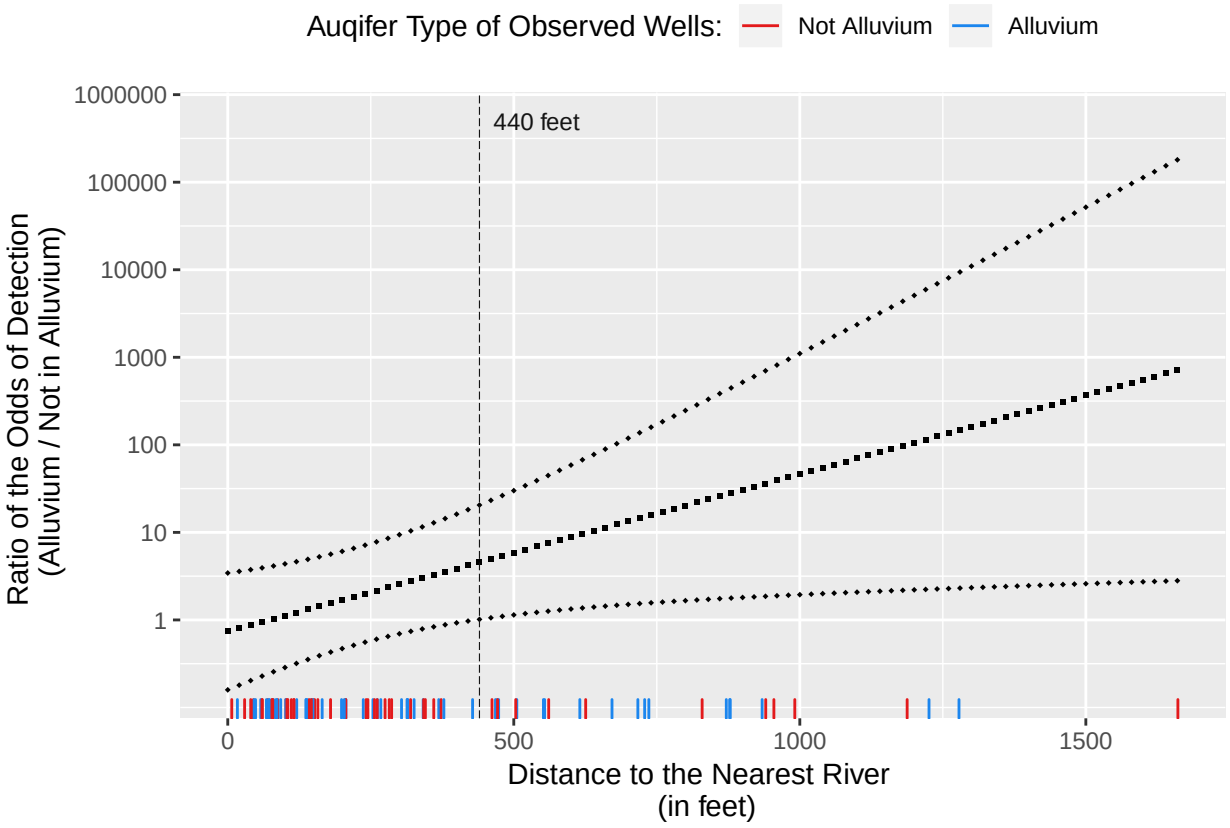

Figure 1: Plot of the estimated ratio in the odds of detection between wells in alluvium and wells not in alluvium, after accounting for well depth, for the sequence of distances (0, 20, ..., 1660) feet with 95% confidence bounds. The vertical dashed line marks the first distance in the sequence for which the lower bound of the confidence interval is above one. The distances from the wells actually observed are jittered at the bottom of the plot and colored by auqifer type.

Works Cited

Ramsey, F., and D. Schafer. 2012. *The Statistical Sleuth: A Course in Methods of Data Analysis*. Cengage Learning. https://books.google.com/books?id=eSlLjA9TwkUC.