Logistic Regression Example — HAB Concentration and Cattle Mortality

Your Name

2025-10-04

Introduction

Harmful algal blooms (HABs) produce toxins that can poison livestock and wildlife.
This analysis examines whether **HAB concentration (µg/L)** predicts **cattle mortality** in 12 sampled ponds.

Research Question

Does HAB concentration increase the likelihood of cattle mortality?

Hypotheses

- **Null hypothesis (H):**
  HAB concentration does **not** affect the probability of cattle mortality.
- **Alternative hypothesis (H):**
  Higher HAB concentration **increases** the probability of cattle mortality.

Statistical Method

We use a **logistic regression model**, appropriate when the response variable is binary (Yes/No).

Model:

$$\text{logit}(P(\text{Mortality} = \text{Yes})) = \beta_0 + \beta_1 \times \text{HAB Concentration}$$

Where: - β_1 = change in **log-odds** of mortality per 1 µg/L increase in HAB concentration.
- $\exp(\beta_1)$ = **odds ratio** (OR), expressing multiplicative change in odds.

Data

The following data summarize 12 pond observations:

Pond_ID	HAB_Concentration (µg/L)	Cattle_Mortality
1	120	Yes
2	80	Yes

Pond_ID	HAB_Concentration (µg/L)	Cattle_Mortality
3	50	No
4	100	Yes
5	60	No
6	130	Yes
7	90	Yes
8	40	No
9	110	Yes
10	70	No
11	30	No
12	20	No

Logistic Regression Analysis

Create dataset

```
Pond_ID <- 1:12
HAB_Concentration <- c(120, 80, 50, 100, 60, 130, 90, 40, 110, 70, 30, 20)
Cattle_Mortality <- c("Yes", "Yes", "No", "Yes", "No", "Yes", "Yes", "No", "Yes", "No", "No", "No")

data <- data.frame(Pond_ID, HAB_Concentration, Cattle_Mortality)

# Convert response variable to binary numeric (Yes=1, No=0)
data$Mortality_Binary <- ifelse(data$Cattle_Mortality == "Yes", 1, 0)

# Display dataset
data
```

```
##    Pond_ID HAB_Concentration Cattle_Mortality Mortality_Binary
## 1        1               120              Yes                1
## 2        2                80              Yes                1
## 3        3                50               No                0
## 4        4               100              Yes                1
## 5        5                60               No                0
## 6        6               130              Yes                1
## 7        7                90              Yes                1
## 8        8                40               No                0
## 9        9               110              Yes                1
## 10      10                70               No                0
## 11      11                30               No                0
## 12      12                20               No                0
```

```
# Fit logistic regression
log_model <- glm(Mortality_Binary ~ HAB_Concentration,
                 data = data, family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
# Display model summary
summary(log_model)
```

```
##
## Call:
## glm(formula = Mortality_Binary ~ HAB_Concentration, family = binomial,
##     data = data)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.484e-05  -2.110e-08   0.000e+00   2.110e-08   2.484e-05
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -328.49  367078.45  -0.001    0.999
## HAB_Concentration      4.38    4883.54   0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.6636e+01  on 11  degrees of freedom
## Residual deviance: 1.2340e-09  on 10  degrees of freedom
## AIC: 4
##
## Number of Fisher Scoring iterations: 25
```

```r
# Calculate odds ratio and 95% confidence interval
odds_ratio <- exp(cbind(OR = coef(log_model),
                        confint(log_model)))
```

```
## Waiting for profiling to be done...

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
odds_ratio
```

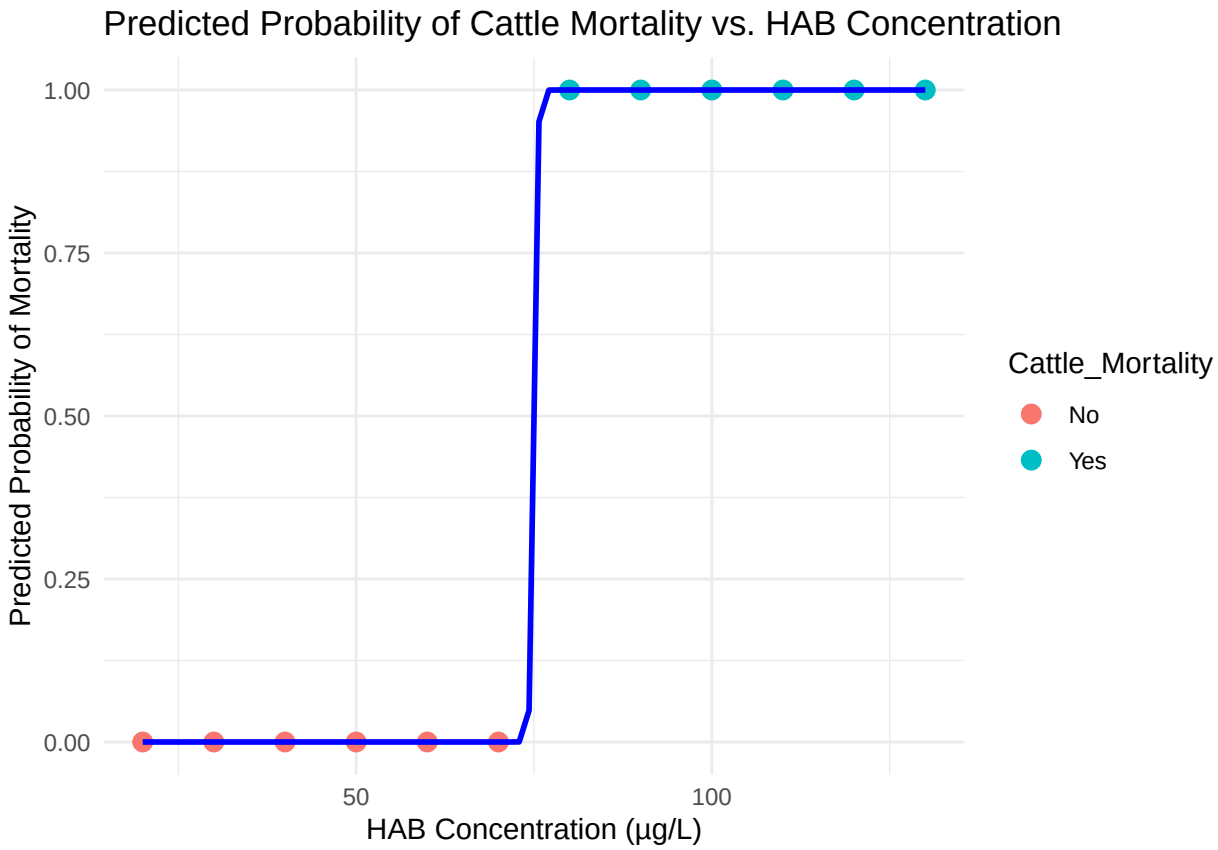
```
##                              OR       2.5 % 97.5 %
## (Intercept)       2.182400e-143  0.0000e+00    Inf
## HAB_Concentration  7.982658e+01 1.9564e-150     NA
```

```
# Generate predicted probabilities
data$Predicted_Prob <- predict(log_model, type = "response")

# Plot predicted probability curve
library(ggplot2)

ggplot(data, aes(x = HAB_Concentration, y = Predicted_Prob)) +
  geom_point(aes(color = Cattle_Mortality), size = 3) +
  geom_smooth(method = "glm", method.args = list(family = "binomial"),
              se = FALSE, color = "blue") +
  labs(title = "Predicted Probability of Cattle Mortality vs. HAB Concentration",
       x = "HAB Concentration (µg/L)",
       y = "Predicted Probability of Mortality") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```


Predicted Probability of Cattle Mortality vs. HAB Concentration
Predicted Probability of Mortality
1.00
0.75
0.50
0.25
0.00
50
100
HAB Concentration (µg/L)
Cattle_Mortality
No
Yes