Week 8 Solutions

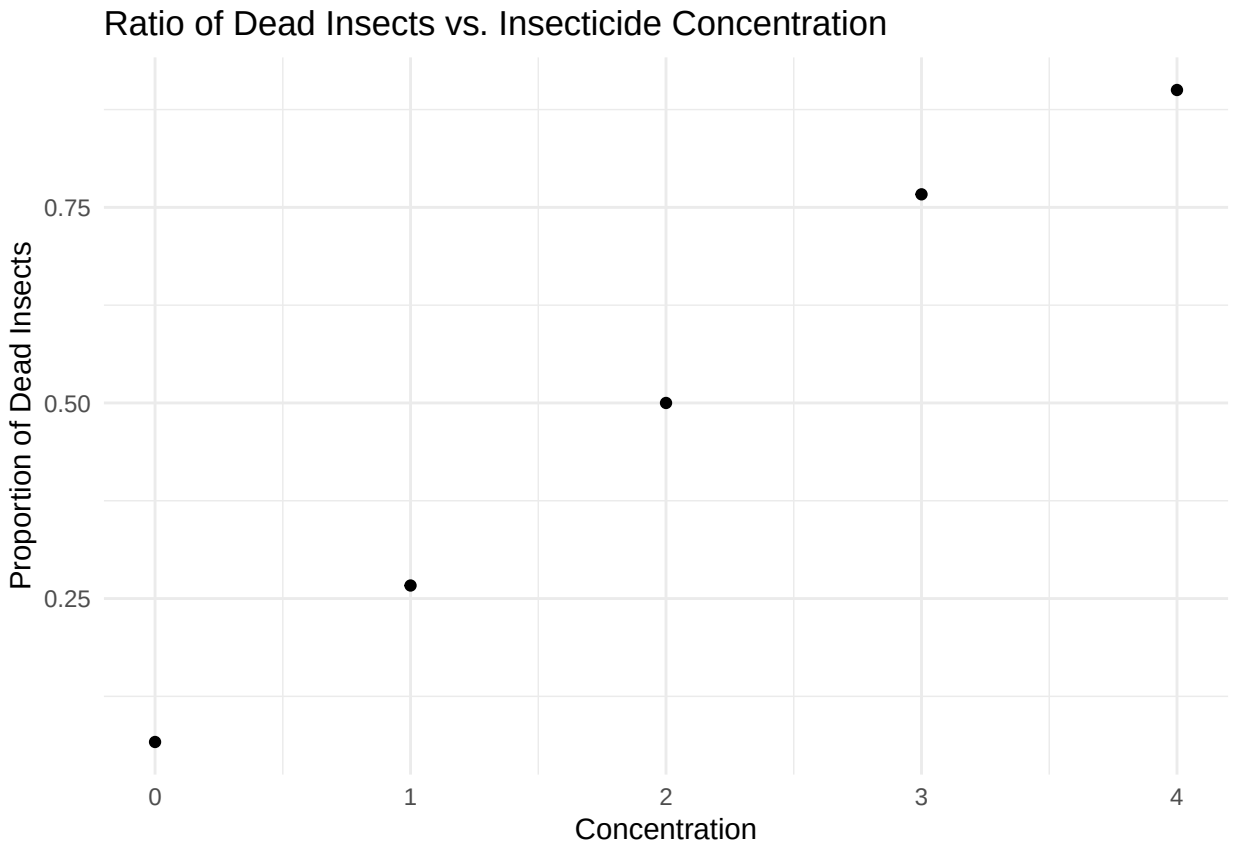
Isabelle Caroline Rose Cretton

2024-11-07

Problem 1: Logistic Regression

(a) Data Exploration

```
# Load the data
data(bliss, package = "faraway")

# Calculate total insects and ratio
bliss$ratio <- bliss$dead / (bliss$dead + bliss$alive)

# Plot the ratio vs concentration
ggplot(bliss, aes(x = conc, y = ratio)) +
  geom_point() +
  labs(title = "Ratio of Dead Insects vs. Insecticide Concentration",
       x = "Concentration",
       y = "Proportion of Dead Insects") +
  theme_minimal()
```



(b) Logistic Regression with Logit Link

```r
# Fit logistic regression
logit_model <- glm(cbind(dead, alive) ~ conc,
                   family = binomial(link = "logit"),
                   data = bliss)

# Display results
summary(logit_model)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ conc, family = binomial(link = "logit"),
##     data = bliss)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.3238     0.4179  -5.561 2.69e-08 ***
## conc          1.1619     0.1814   6.405 1.51e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 64.76327  on 4  degrees of freedom
```

```
## Residual deviance:  0.37875  on 3  degrees of freedom
## AIC: 20.854
##
## Number of Fisher Scoring iterations: 4
```

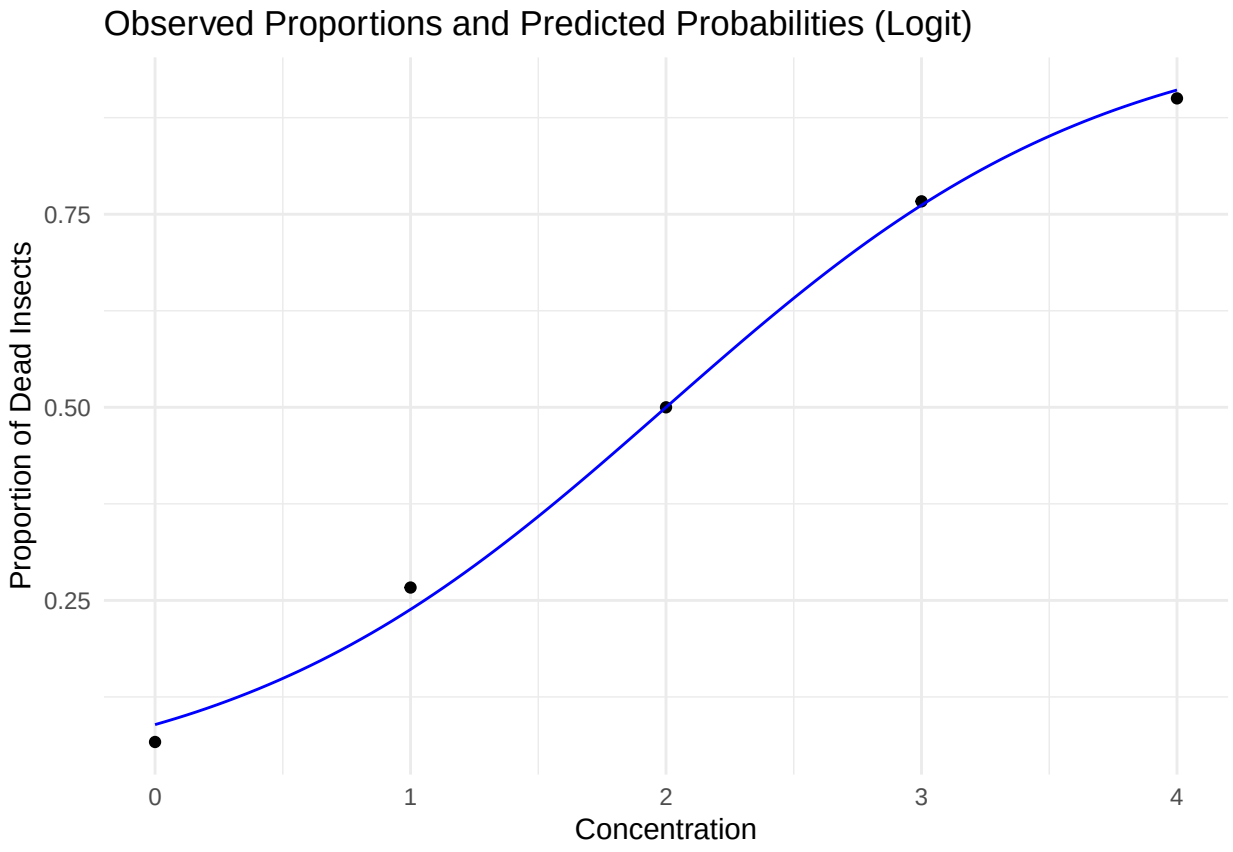
(c) Manual Prediction Calculation

```r
# Calculate predicted values manually
coeffs <- coef(logit_model)
linear_pred <- coeffs[1] + coeffs[2] * bliss$conc
manual_pred <- exp(linear_pred) / (1 + exp(linear_pred))

# Compare with fitted values
fitted_pred <- fitted(logit_model)

# Compare results
comparison_df <- data.frame(
  concentration = bliss$conc,
  manual = manual_pred,
  fitted = fitted_pred,
  difference = abs(manual_pred - fitted_pred)
)
print(comparison_df)
```

```
##   concentration     manual     fitted   difference
## 1             0 0.08917177 0.08917177 0.000000e+00
## 2             1 0.23832314 0.23832314 0.000000e+00
## 3             2 0.50000000 0.50000000 0.000000e+00
## 4             3 0.76167686 0.76167686 1.110223e-16
## 5             4 0.91082823 0.91082823 0.000000e+00
```

(d) Add Predicted Probabilities to Plot

```r
# Create prediction grid
pred_grid <- data.frame(conc = seq(min(bliss$conc), max(bliss$conc), length.out = 100))
pred_grid$pred <- predict(logit_model, newdata = pred_grid, type = "response")

ggplot() +
  geom_point(data = bliss, aes(x = conc, y = ratio)) +
  geom_line(data = pred_grid, aes(x = conc, y = pred), color = "blue") +
  labs(title = "Observed Proportions and Predicted Probabilities (Logit)",
       x = "Concentration",
       y = "Proportion of Dead Insects") +
  theme_minimal()
```



(e) Add Confidence Intervals

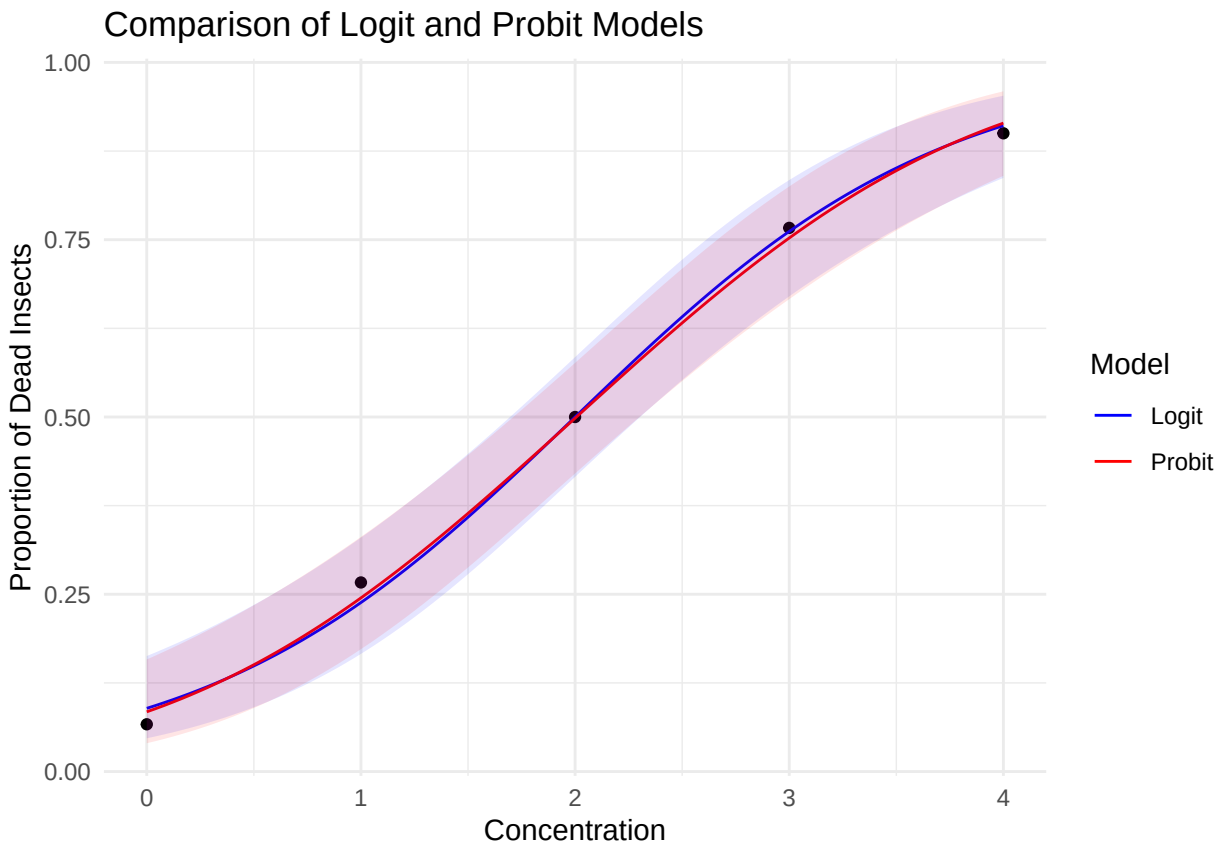
```r
# Calculate confidence intervals
pred_ci <- predict(logit_model,
                   newdata = pred_grid,
                   type = "link",
                   se.fit = TRUE)

# Transform to probability scale
ci_lower <- plogis(pred_ci$fit - 1.645 * pred_ci$se.fit)  # 90% CI
ci_upper <- plogis(pred_ci$fit + 1.645 * pred_ci$se.fit)

# Add to plot
pred_grid$lower <- ci_lower
pred_grid$upper <- ci_upper

ggplot() +
  geom_point(data = bliss, aes(x = conc, y = ratio)) +
  geom_line(data = pred_grid, aes(x = conc, y = pred), color = "blue") +
  geom_ribbon(data = pred_grid,
              aes(x = conc, ymin = lower, ymax = upper),
              alpha = 0.2) +
  labs(title = "Logistic Regression with 90% Confidence Interval",
       x = "Concentration",
       y = "Proportion of Dead Insects") +
```

```
  theme_minimal()
```

(f) Probit Link Function

```
# Fit probit model
probit_model <- glm(cbind(dead, alive) ~ conc,
                    family = binomial(link = "probit"),
                    data = bliss)

# Get predictions
pred_grid$probit_pred <- predict(probit_model,
                                 newdata = pred_grid,
                                 type = "response")

# Calculate confidence intervals
probit_ci <- predict(probit_model,
                     newdata = pred_grid,
                     type = "link",
                     se.fit = TRUE)

pred_grid$probit_lower <- pnorm(probit_ci$fit - 1.645 * probit_ci$se.fit)
pred_grid$probit_upper <- pnorm(probit_ci$fit + 1.645 * probit_ci$se.fit)

# Plot both models
```

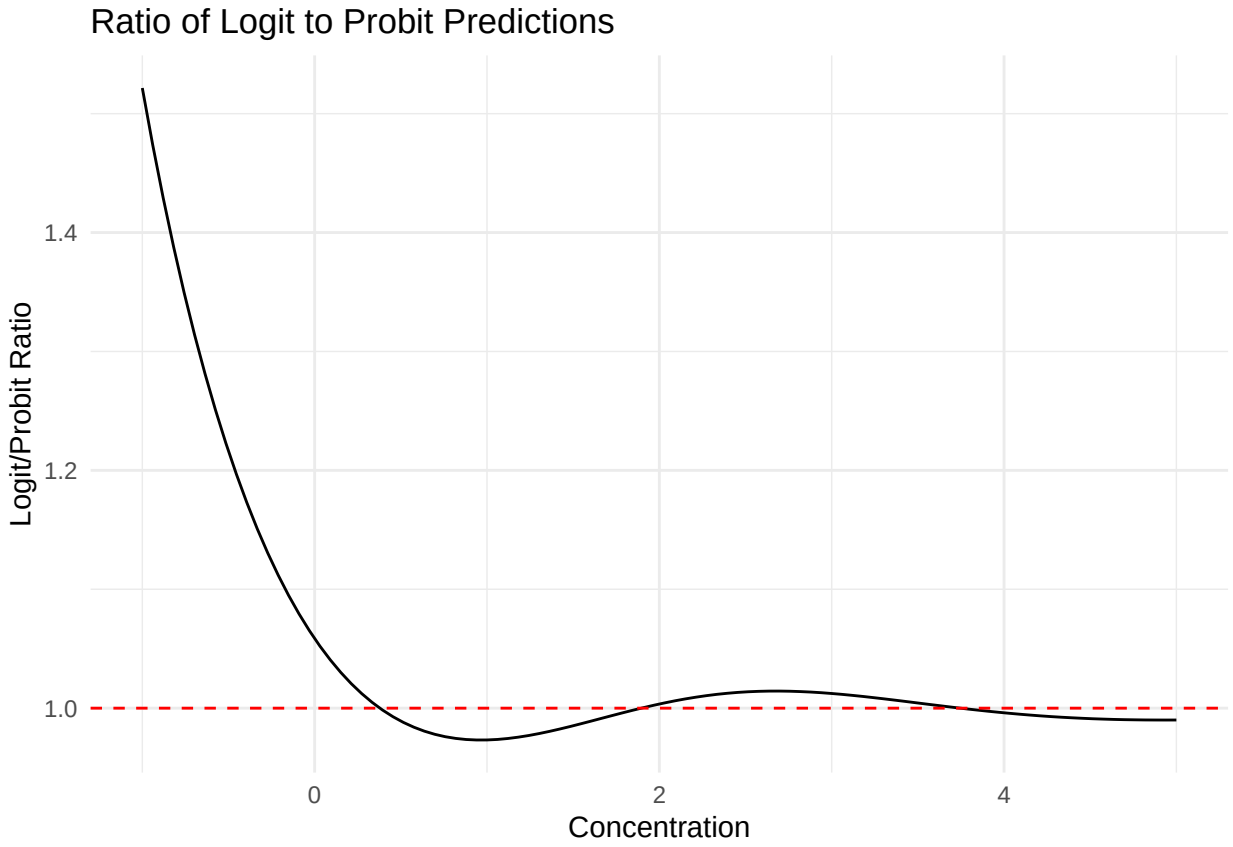
```r
ggplot() +
  geom_point(data = bliss, aes(x = conc, y = ratio)) +
  geom_line(data = pred_grid, aes(x = conc, y = pred, color = "Logit")) +
  geom_line(data = pred_grid, aes(x = conc, y = probit_pred, color = "Probit")) +
  geom_ribbon(data = pred_grid,
              aes(x = conc, ymin = lower, ymax = upper),
              alpha = 0.1, fill = "blue") +
  geom_ribbon(data = pred_grid,
              aes(x = conc, ymin = probit_lower, ymax = probit_upper),
              alpha = 0.1, fill = "red") +
  scale_color_manual(values = c("blue", "red")) +
  labs(title = "Comparison of Logit and Probit Models",
       x = "Concentration",
       y = "Proportion of Dead Insects",
       color = "Model") +
  theme_minimal()
```



(g) Compare Logit and Probit Predictions

```r
# Create comparison grid
comp_grid <- data.frame(conc = seq(-1, 5, length.out = 100))

# Get predictions
comp_grid$logit_pred <- predict(logit_model,
```

```
                                newdata = comp_grid,
                                type = "response")
comp_grid$probit_pred <- predict(probit_model,
                                 newdata = comp_grid,
                                 type = "response")
comp_grid$ratio <- comp_grid$logit_pred / comp_grid$probit_pred

ggplot(comp_grid, aes(x = conc, y = ratio)) +
  geom_line() +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Ratio of Logit to Probit Predictions",
       x = "Concentration",
       y = "Logit/Probit Ratio") +
  theme_minimal()
```



(h) Calculate LD50

```
# For logit model, LD50 is when logit = 0
coeffs <- coef(logit_model)
ld50 <- -coeffs[1] / coeffs[2]
print(paste("LD50 (concentration at which 50% die):", round(ld50, 3)))
```

```
## [1] "LD50 (concentration at which 50% die): 2"
```

Problem 2: Exponential Family

The exponential family has the form:

$f(y;\theta,\phi) = exp((y\theta - b(\theta))/\phi + c(y,\phi))$

(a) Exponential Distribution

The probability density function is: $f(y;\lambda) = \lambda e^{-\lambda y}$

We can rewrite this as: $f(y;\lambda) = exp(log(\lambda) - \lambda y) = exp(-\lambda y + log(\lambda))$

This belongs to the exponential family with:

- $\theta = -\lambda$
- $\phi = 1$
- $b(\theta) = -log(-\theta)$
- $c(y,\phi) = 0$

(b) Binomial Distribution

The probability density function is: $f(y;\pi) = \binom{n}{y}\pi^y(1-\pi)^{n-y}$

We can rewrite this as: $f(y;\pi) = exp(ylog(\pi/(1-\pi)) + nlog(1-\pi) + log(\binom{n}{y})) \quad = exp(y\theta - nlog(1+e^\theta) + log(\binom{n}{y}))$

where $\theta = log(\pi/(1-\pi))$

This belongs to the exponential family with:

- $\theta = log(\pi/(1-\pi))$
- $\phi = 1$
- $b(\theta) = nlog(1+e^\theta)$
- $c(y,\phi) = log(\binom{n}{y})$

(c) Uniform Distribution

The probability density function is: $f(y;\theta) = 1/\theta, \quad 0 < y < \theta$

This cannot be written in exponential family form because the support of y depends on θ.

(d) Normal Distribution (known variance)

The probability density function is: $f(y;\mu,\sigma^2) = (1/\sqrt{2\pi\sigma^2})exp(-(y-\mu)^2/(2\sigma^2))$

We can rewrite this as: $f(y;\mu,\sigma^2) = exp(y\mu/\sigma^2 - \mu^2/(2\sigma^2) - y^2/(2\sigma^2) - (1/2)log(2\pi\sigma^2))$

This belongs to the exponential family with:

- $\theta = \mu$
- $\phi = \sigma^2$
- $b(\theta) = \theta^2/2$
- $c(y,\phi) = -y^2/(2\phi) - (1/2)log(2\pi\phi)$