

Day8 exercise solutions

Ali Movasati, Isabelle Caroline Rose Cretton, Tristan Koning

Oct. 4th, 2024

```
# Set global code chunk options
knitr::opts_chunk$set(warning = FALSE)

# load required libraries
library("skimr")
library("dplyr")

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
library("magrittr")
library("ggplot2")

# define functions
`%notin%` <- Negate(`%in%`)
```

Problem 1

```
data(bliss, package = "faraway")
bliss %<>% mutate(ratio = dead/(alive + dead))
```

1.A)

```
skim(bliss)
```

Table 1: Data summary

Name	bliss
Number of rows	5
Number of columns	4
Column type frequency:	
numeric	4

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
dead	0	1	15.0	10.32	2.00	8.00	15.0	23.00	27.0	
alive	0	1	15.0	10.32	3.00	7.00	15.0	22.00	28.0	
conc	0	1	2.0	1.58	0.00	1.00	2.0	3.00	4.0	
ratio	0	1	0.5	0.34	0.07	0.27	0.5	0.77	0.9	

```
head(bliss)
```

```
##   dead alive conc      ratio
## 1     2    28   0 0.06666667
## 2     8    22   1 0.26666667
## 3    15    15   2 0.50000000
## 4    23     7   3 0.76666667
## 5    27     3   4 0.90000000
```

```
# Load the data
```

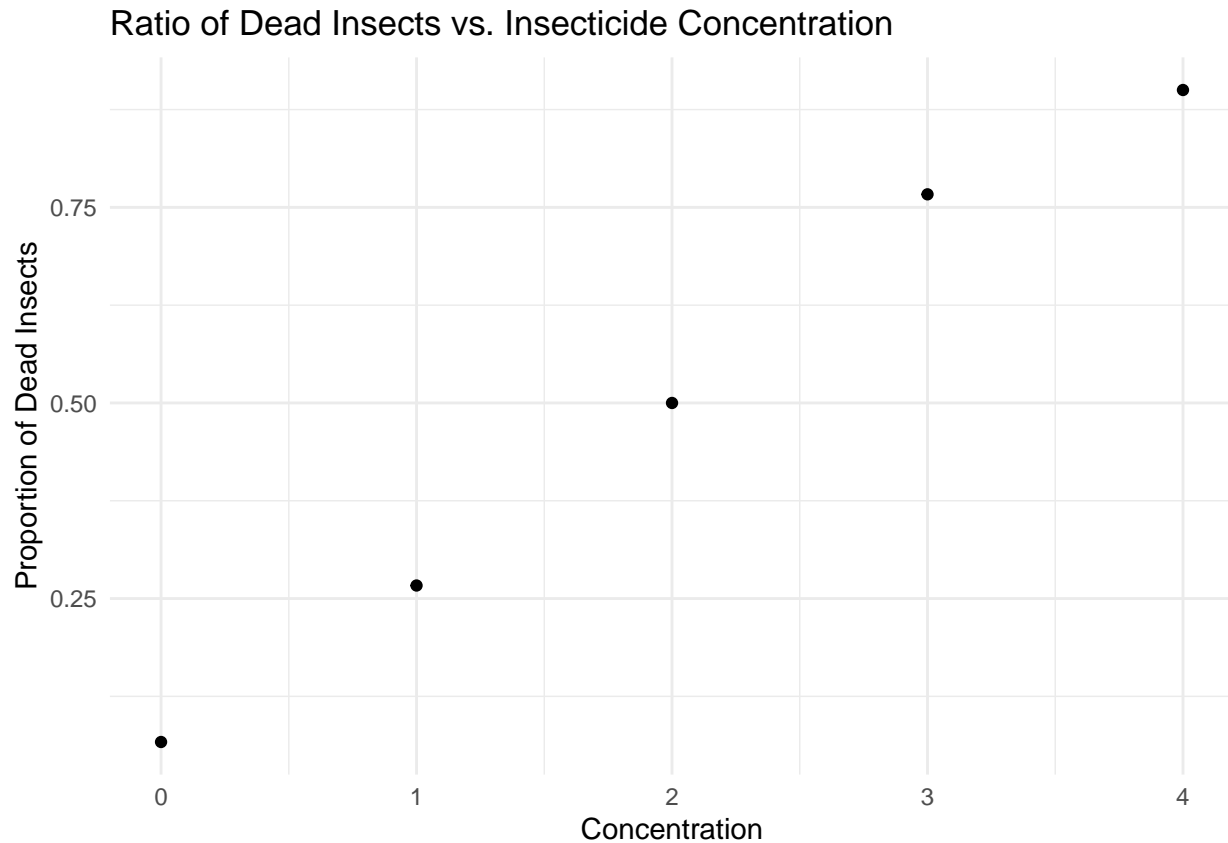
```
data(bliss, package = "faraway")
```

```
# 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()
```



1.B)

```
logit_model <- glm(cbind(dead,alive)~conc, family = binomial(link=logit), data = bliss)
summary(logit_model)
```

```
##
## Call:
## glm(formula = cbind(dead, alive) ~ conc, family = binomial(link = logit),
##      data = bliss)
##
## Deviance Residuals:
##      1       2       3       4       5
## -0.4510  0.3597  0.0000  0.0643 -0.2045
##
## 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
```

1.C)

```
# 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)

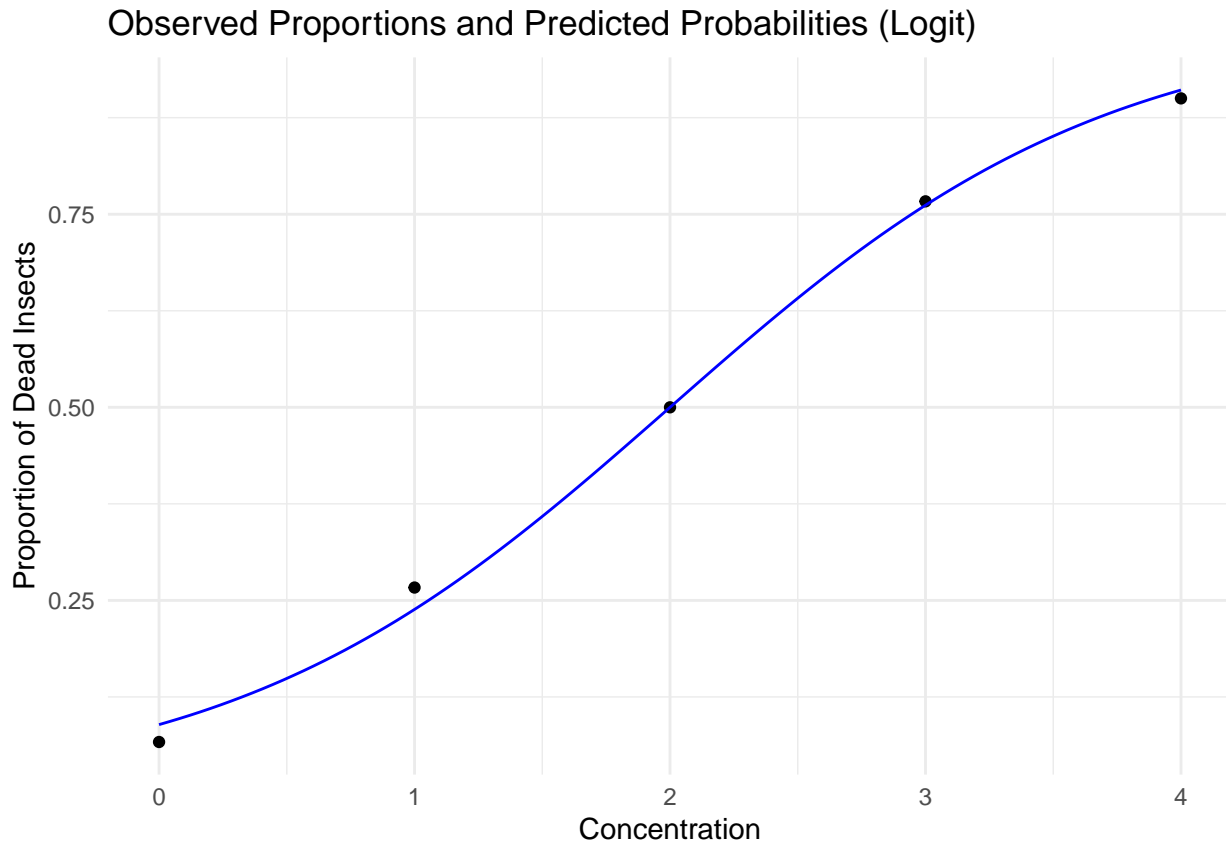
# Compare results
comparison_df <- data.frame(
  concentration = bliss$conc,
  manual = manual_pred,
  fitted = fitted_pred,
  difference = abs(manual_pred - fitted_pred)
)
print(comparison_df)
```

```
##   concentration    manual    fitted difference
## 1             0 0.08917177 0.08917177         0
## 2             1 0.23832314 0.23832314         0
## 3             2 0.50000000 0.50000000         0
## 4             3 0.76167686 0.76167686         0
## 5             4 0.91082823 0.91082823         0
```

1.D)

```
# 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()
```



1.E)

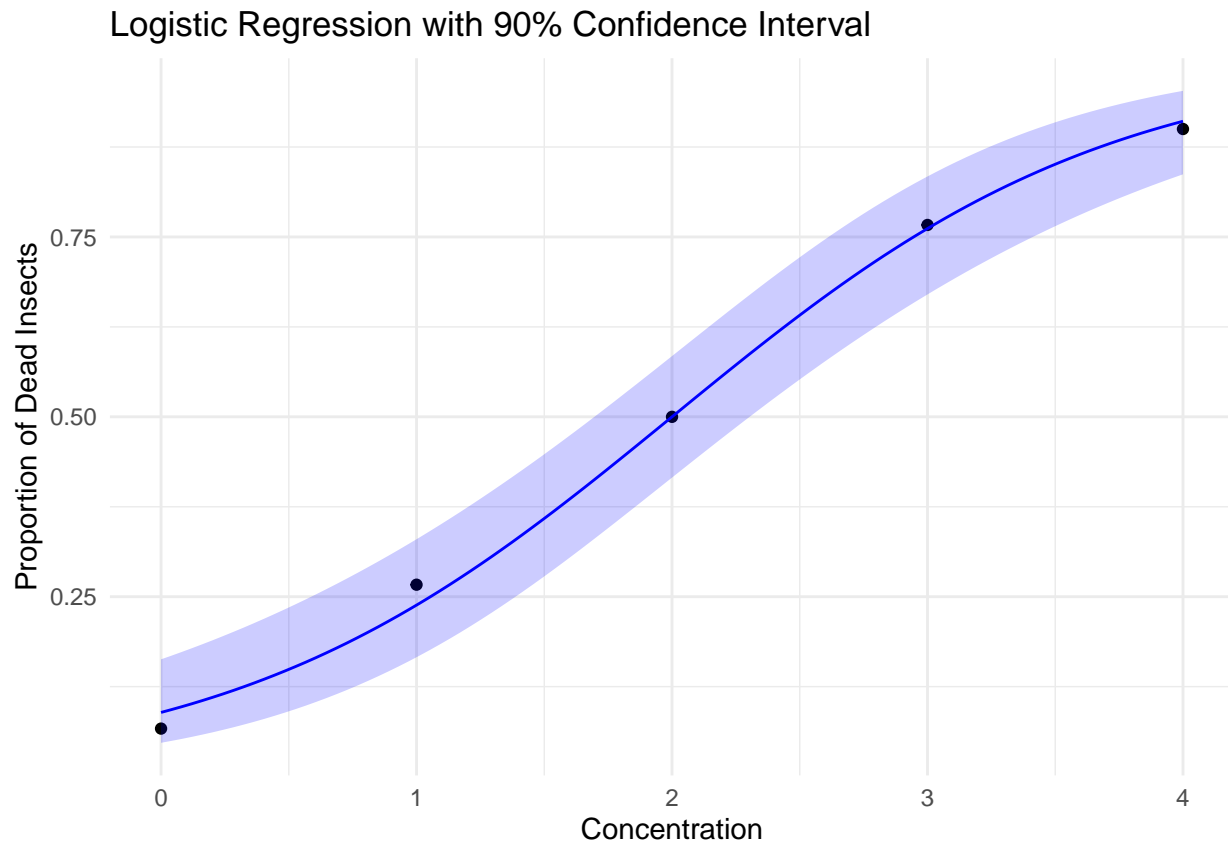
```
# Calculate confidence intervals
pred_ci <- predict(logit_model,
  newdata = pred_grid,
  type = "link",
  se.fit = TRUE)

# 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,
    fill = "blue") +
  labs(title = "Logistic Regression with 90% Confidence Interval",
    x = "Concentration",
    y = "Proportion of Dead Insects") +
```

```
theme_minimal()
```



1.F)

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

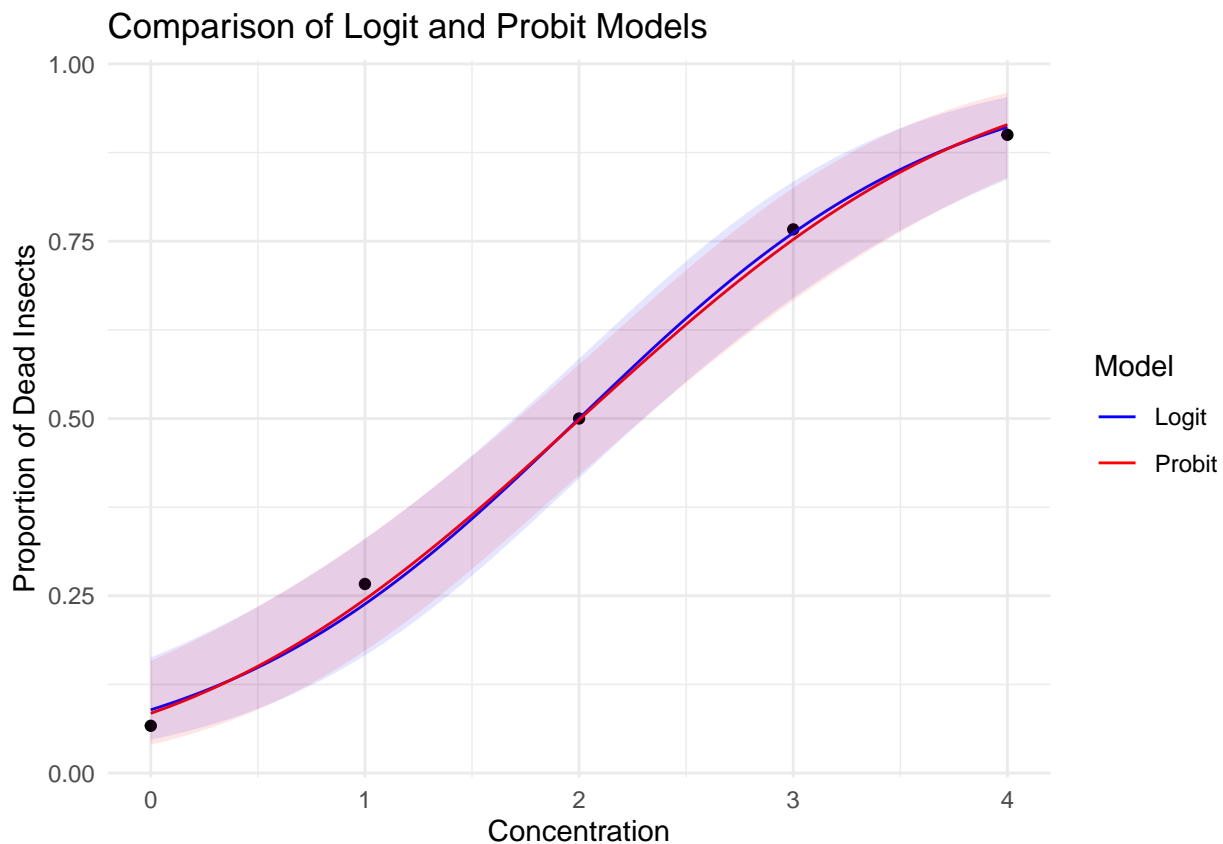
# Get predictions
pred_grid$probit_pred <- predict(probit_model,
                                newdata = pred_grid,
                                type = "response")

# 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)
```

1.G)

```
# Plot both models
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",
            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()
```



1.H)

```
logit_model_out <- predict(logit_model, newdata=data.frame(conc=seq(-1,to=5,length.out=100000)), type="probs")
ld50 <- round(seq(-1,to=5,length.out=100000)[which.min(abs(logit_model_out-0.5))])

print(paste0("LD50 or the dose at which there is 50% chance of survival is ", ld50))
```

```
## [1] "LD50 or the dose at which there is 50% chance of survival is 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: $f(y; \pi) = \exp(y \log(\pi/(1 - \pi)) + n \log(1 - \pi) + \log(\binom{n}{y})) = \exp(y\theta - n \log(1 + e^\theta) + \log(\binom{n}{y}))$

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

This belongs to the exponential family with:

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

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

- **Natural parameter:** $\theta = \frac{\mu}{\sigma^2}$
- **Dispersion parameter:** $\phi = \sigma^2$
- **Function $b(\theta)$:** $b(\theta) = \frac{\mu^2}{2\sigma^2}$
- **Function $c(y, \phi)$:** $c(y, \phi) = -\frac{y^2}{2\phi^2} - \frac{1}{2} \ln(2\pi\phi^2)$