

Day8 exercise solutions

Ali Movasati, Isabelle Cretton, Tristan Koning

Oct. 31st, 2024

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

```
# load required libraries
library(faraway)
library(skimr)
library(ggplot2)

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

Problem 1

(a)

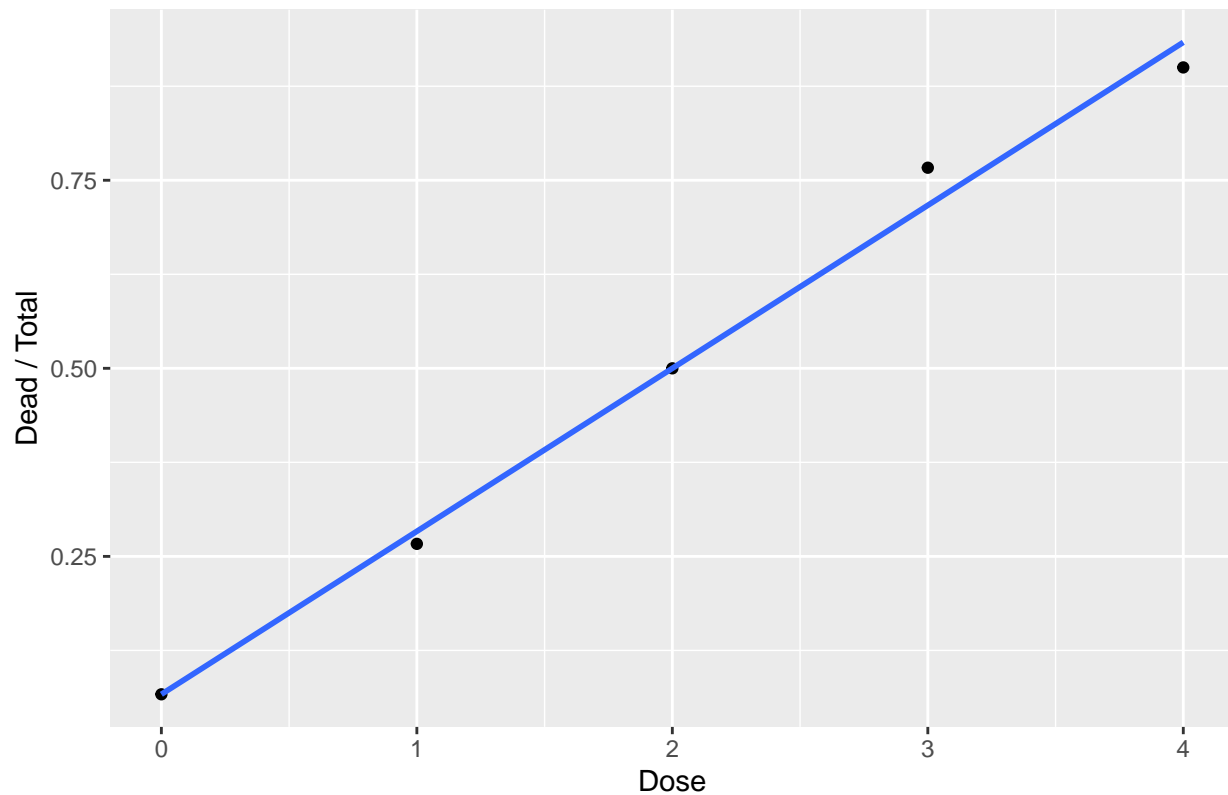
```
data(bliss)
head(bliss)
```

```
##   dead alive conc
## 1    2    28    0
## 2    8    22    1
## 3   15    15    2
## 4   23     7    3
## 5   27     3    4
```

```
# Plot ratio of dead and total insects
ggplot(bliss, aes(x = conc, y = dead / 30)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Ratio of Dead and Total Insects", x = "Dose", y = "Dead / Total")
```

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

Ratio of Dead and Total Insects



(b)

```
# Fit logistic model, predicting dead and alive insects (cbind)
model_A <- glm(cbind(dead, 30 - dead) ~ conc, data = bliss, family = binomial)
summary(model_A)
```

```
##
## Call:
## glm(formula = cbind(dead, 30 - dead) ~ conc, family = binomial,
##      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

# Plot the fitted model
ggplot(bliss, aes(x = conc, y = dead / 30)) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = binomial), se = FALSE) +
  labs(title = "Ratio of Dead and Total Insects", x = "Dose", y = "Dead / Total")

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



We get a significant results for both the Intercept and the concentration variable. The higher the concentration is, the higher the ratio of dead insects.

(c)

```
plogis(coef(model_A)[1] + coef(model_A)[2] * bliss$conc)

## [1] 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823

fitted(model_A)

##          1          2          3          4          5
## 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823
```

(d)

```
newdata <- data.frame(conc = c(bliss$conc))
predictions <- predict.glm(model_A, newdata = newdata, type = "response")

# Plot ratio of dead and total insects and add the fitted values
ggplot(bliss, aes(x = conc, y = dead / 30)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, aes(color = "Original Ratio")) +
  geom_line(aes(y = predictions, color = "Fitted Line")) +
  scale_color_manual(name = "Legend", values = c("Original Ratio" = "blue", "Fitted Line" = "black")) +
  labs(title = "Ratio of Dead and Total Insects", x = "Dose", y = "Dead / Total")
```

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



(e)

```
lin_predictors <- predict(model_A, newdata = newdata, type = "link", se.fit = TRUE)
lin_predictors
```

```
## $fit
```

```
##           1           2           3           4           5
## -2.323790e+00 -1.161895e+00 -4.440892e-16  1.161895e+00  2.323790e+00
##
## $se.fit
##           1           2           3           4           5
## 0.4178878 0.2754909 0.2073247 0.2754909 0.4178878
##
## $residual.scale
## [1] 1
```

```
# Get 90% confidence intervals
```

```
alpha <- 0.10
```

```
z <- qnorm(1 - alpha / 2)
```

```
ci_logit <- cbind(lin_predictors$fit - z * lin_predictors$se.fit, lin_predictors$fit + z * lin_predictors$se.fit)
```

```
ci <- plogis(ci_logit)
```

```
fitted_values_ci <- data.frame(
```

```
  conc = newdata$conc,
```

```
  dead = bliss$dead,
```

```
  predictions = predictions,
```

```
  ci_lower = ci[, 1],
```

```
  ci_upper = ci[, 2]
```

```
)
```

```
# Plot the data and the resulting fit with 90% CI
```

```
ggplot(data = bliss, aes(x = conc, y = dead / 30)) +
```

```
  geom_point() +
```

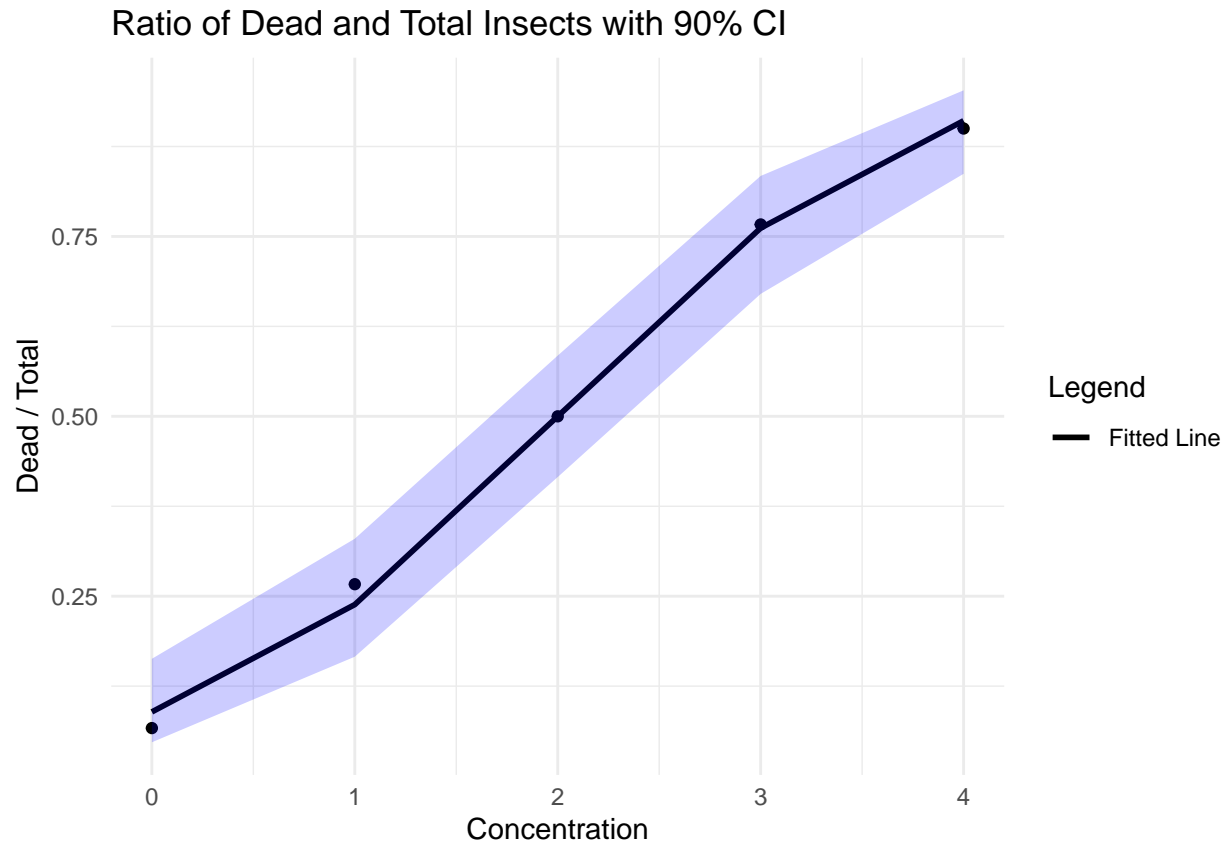
```
  geom_line(aes(y = predictions, color = "Fitted Line"), data = fitted_values_ci, size = 1) +
```

```
  geom_ribbon(aes(ymin = ci_lower, ymax = ci_upper), data = fitted_values_ci, alpha = 0.2, fill = "blue") +
```

```
  scale_color_manual(name = "Legend", values = c("Original Ratio" = "blue", "Fitted Line" = "black")) +
```

```
  labs(title = "Ratio of Dead and Total Insects with 90% CI", x = "Concentration", y = "Dead / Total") +
```

```
  theme_minimal()
```



(f)

```
model_B <- glm(cbind(dead, 30 - dead) ~ conc, data = bliss, family = binomial(link = "probit"))
summary(model_B)
```

```
##
## Call:
## glm(formula = cbind(dead, 30 - dead) ~ conc, family = binomial(link = "probit"),
##      data = bliss)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.37709    0.22781  -6.045 1.49e-09 ***
## conc         0.68638    0.09677   7.093 1.31e-12 ***
## ---
## 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:  0.31367  on 3  degrees of freedom
## AIC: 20.789
##
## Number of Fisher Scoring iterations: 4
```

```
predict(model_B, newdata = newdata, type = "response")
```

```
##           1           2           3           4           5
## 0.08424186 0.24487335 0.49827210 0.75239612 0.91441122
```

```
# Get 90% confidence intervals
```

```
lin_predictors <- predict(model_B, newdata = newdata, type = "link", se.fit = TRUE)
```

```
ci_probit <- cbind(lin_predictors$fit - z * lin_predictors$se.fit, lin_predictors$fit + z * lin_predictors$se.fit)
```

```
ci <- pnorm(ci_probit)
```

```
fitted_values_ci <- data.frame(
```

```
  conc = newdata$conc,
```

```
  dead = bliss$dead,
```

```
  predictions = predictions,
```

```
  ci_lower = ci[, 1],
```

```
  ci_upper = ci[, 2]
```

```
)
```

```
# Plot the data and the resulting fit with 90% CI
```

```
ggplot(data = bliss, aes(x = conc, y = dead / 30)) +
```

```
  geom_point() +
```

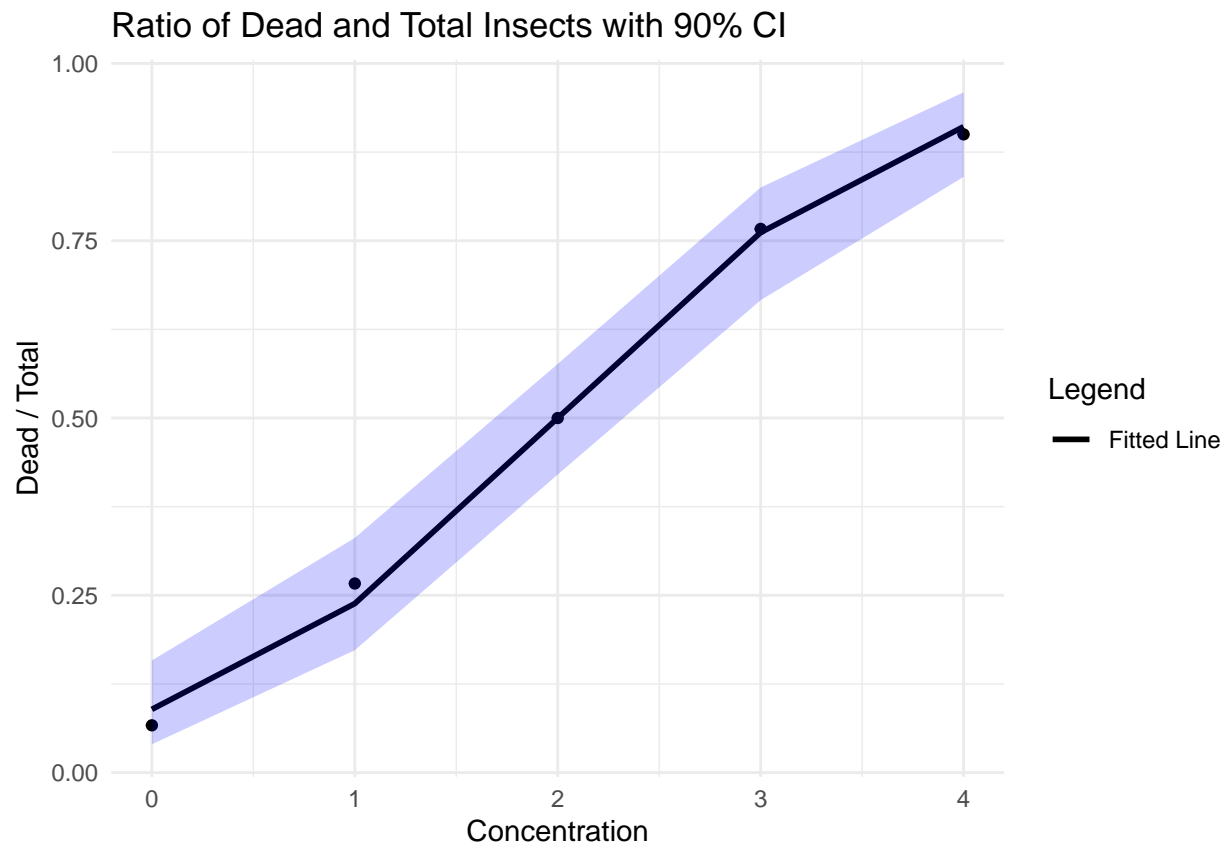
```
  geom_line(aes(y = predictions, color = "Fitted Line"), data = fitted_values_ci, size = 1) +
```

```
  geom_ribbon(aes(ymin = ci_lower, ymax = ci_upper), data = fitted_values_ci, alpha = 0.2, fill = "blue")
```

```
  scale_color_manual(name = "Legend", values = c("Original Ratio" = "blue", "Fitted Line" = "black")) +
```

```
  labs(title = "Ratio of Dead and Total Insects with 90% CI", x = "Concentration", y = "Dead / Total")
```

```
  theme_minimal()
```

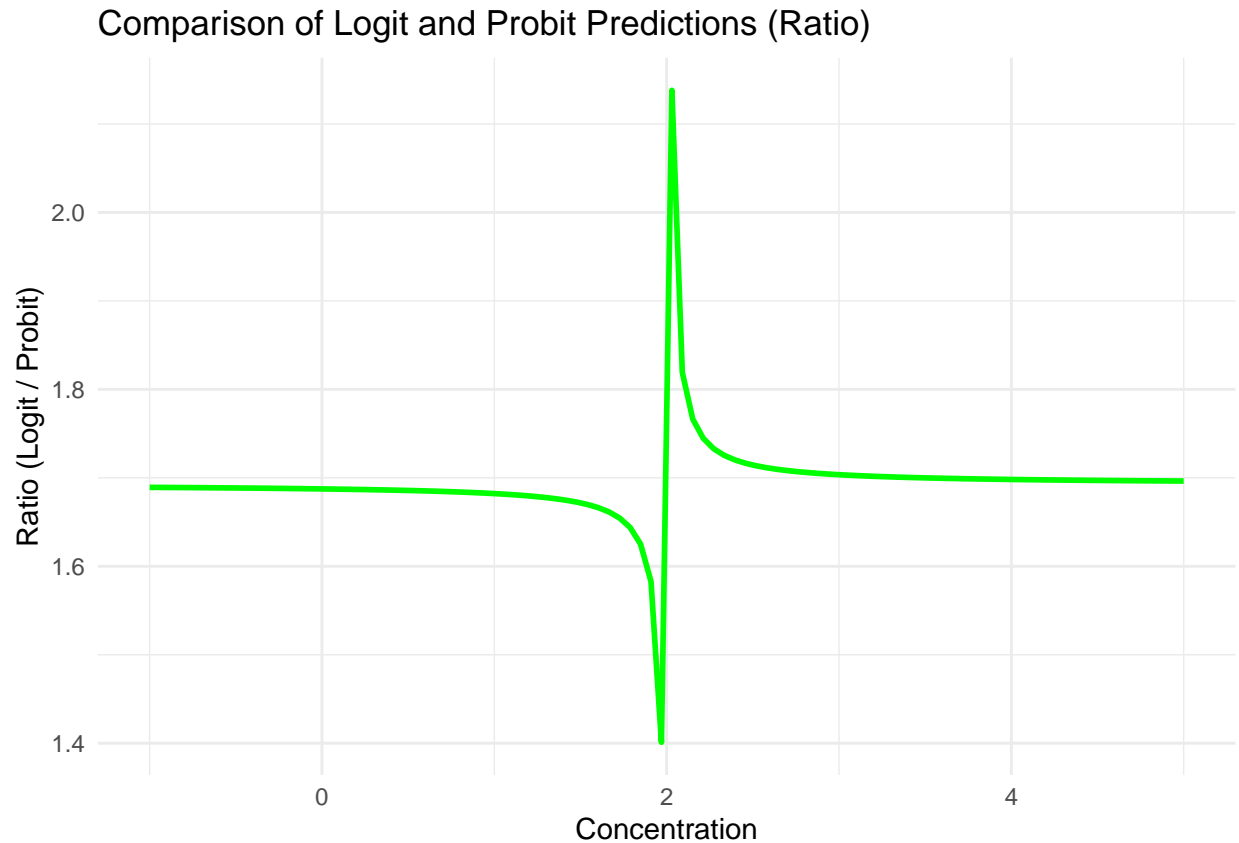


(g)

```
newdata <- data.frame(conc = seq(-1, 5, length.out = 100))
predicted_logit <- predict(model_A, newdata = newdata, type = "link")
predicted_probit <- predict(model_B, newdata = newdata, type = "link")
predicted_ratio <- predicted_logit / predicted_probit

comparison_data <- data.frame(
  conc = newdata$conc,
  predicted_ratio = predicted_ratio
)

ggplot(comparison_data, aes(x = conc, y = predicted_ratio)) +
  geom_line(color = "green", size = 1) +
  labs(title = "Comparison of Logit and Probit Predictions (Ratio)", x = "Concentration", y = "Ratio (Logit / Probit)") +
  theme_minimal()
```

They never reach the same predicted value, but the closest ratio is around the concentration of 2 with 1.4.

(h)

```
model_C <- glm(cbind(dead, 30 - dead) ~ conc, data = bliss, family = binomial(link = "logit"))

beta0 <- coef(model_C)[1]
beta1 <- coef(model_C)[2]
p = 0.5

log_odds = logit(p)
# logit(p) = beta0 + beta1 * conc
ld50_conc <- (log_odds - beta0) / beta1
ld50_conc

## (Intercept)
##          2
```

The lethal dose where 50% of insects die is 2.