Lecture 14 - Generalized Linear Models

Bill Perry

Lecture 13: Review of ANOVAs

Review

- ANOVA
- Factorial ANOVA
- Nested ANOVA
- ASSUMPTIONS OF ALL
  - Homogeneity of variance - Levenes or Bartlets Test
  - Normality of Residuals
  - Independence

NEED IMAGE FOR REVIEW

Lecture 14: GLM Overview

Overview

General Linear Models GLM

- Essentially the same as before while using defined distributions
  - Normal
  - Lognormal
  - Binomial
  - Poisson
  - Gamma
  - Negative binomial

Logistic Regression

- when the outcome is yes or no

Overview of Generalized Linear Models (GLMs)

General linear models assume normal distribution of response variables and residuals. However, many types of biological data don't meet this assumption. Generalized Linear Models (GLMs) allow for a wider range of probability distributions for the response variable.

GLMs allow all types of "exponential family" distributions:

- Normal
- Lognormal
- Binomial
- Poisson
- Gamma
- Negative binomial

GLMs can be used for binary (yes/no), discrete (count), and categorical/multinomial response variables, using maximum likelihood (ML) rather than ordinary least squares (OLS) for estimation.

Note: GLMs extend linear models to non-normal data distributions.

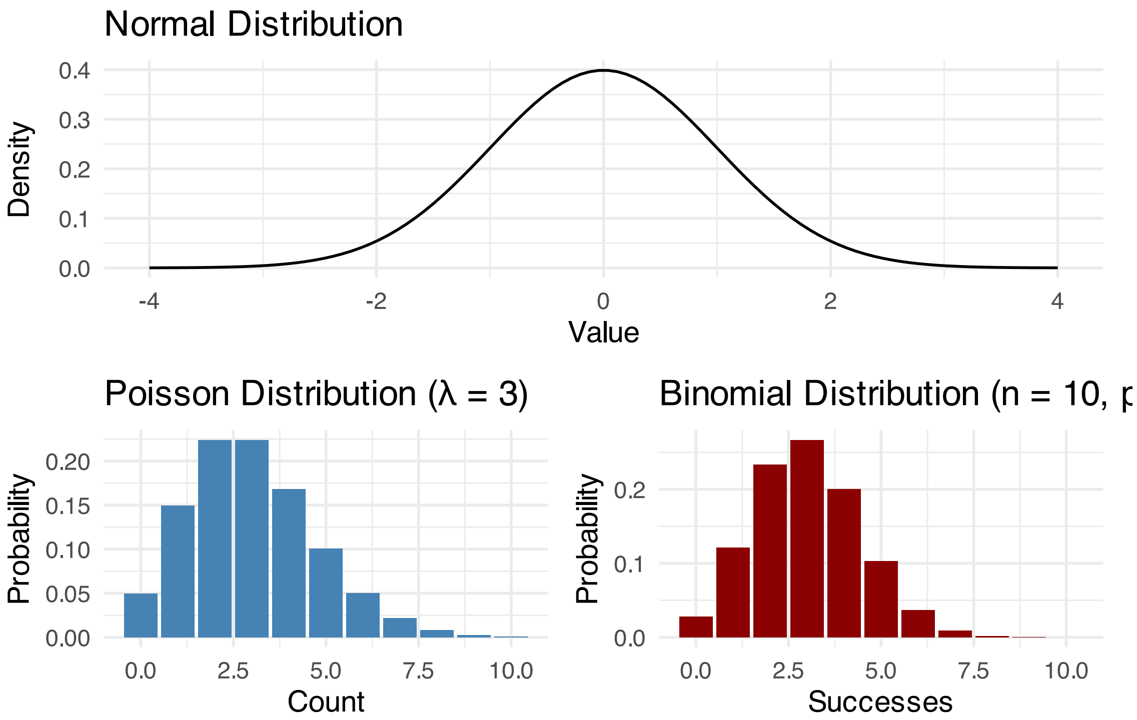


Figure 1: Examples of distributions in the exponential family

The Three Elements of a GLM

GLMs consist of three components:

1. **Random component**: The response variable and its probability distribution (from exponential family: normal, binomial, Poisson)
2. **Systematic component**: The predictor variable(s) in the model, which can be continuous or categorical
3. **Link function**: Connects expected value of Y to predictor variables

$$g(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 ...$$

Link Functions and Distributions

Distribution	Common Link Function	Formula
Normal	Identity	$g(\mu) = \mu$
Poisson	Log	$g(\mu) = \log(\mu)$
Binomial	Logit	$g(\mu) = \log[\mu/(1-\mu)]$

GLM with Gaussian (Normal) Distribution: Setup

The simplest form of GLM uses a normal (Gaussian) distribution with an identity link function. This is equivalent to standard linear regression.

Let's compare a standard linear model and a Gaussian GLM using the `mtcars` dataset, modeling miles per gallon (mpg) by the number of cylinders (cyl).

```r
# Convert cylinders to a factor
mtcars <- mtcars %>%
  mutate(cyl = factor(cyl))

# Fit a standard linear model
model_lm <- lm(mpg ~ cyl, data = mtcars)

# Fit a Gaussian GLM
model_gaussian <- glm(mpg ~ cyl,
                       data = mtcars,
                       family = gaussian(link = "identity"))

# Compare the coefficients
coef_lm <- coefficients(model_lm)
coef_glm <- coefficients(model_gaussian)

# Check if they're the same
all.equal(coef_lm, coef_glm)
```

```
[1] TRUE
```

Let's look at the summary of our Gaussian GLM:

```r
summary(model_gaussian)
```

```


Call:
glm(formula = mpg ~ cyl, family = gaussian(link = "identity"),
    data = mtcars)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  26.6636     0.9718  27.437  < 2e-16 ***
cyl6         -6.9208     1.5583  -4.441 0.000119 ***
cyl8        -11.5636     1.2986  -8.905 8.57e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 10.38837)

    Null deviance: 1126.05  on 31  degrees of freedom
Residual deviance:  301.26  on 29  degrees of freedom
AIC: 170.56

Number of Fisher Scoring iterations: 2
```

GLM with Gaussian Distribution: Analysis

Now let's perform an ANOVA on our GLM model using the car package:

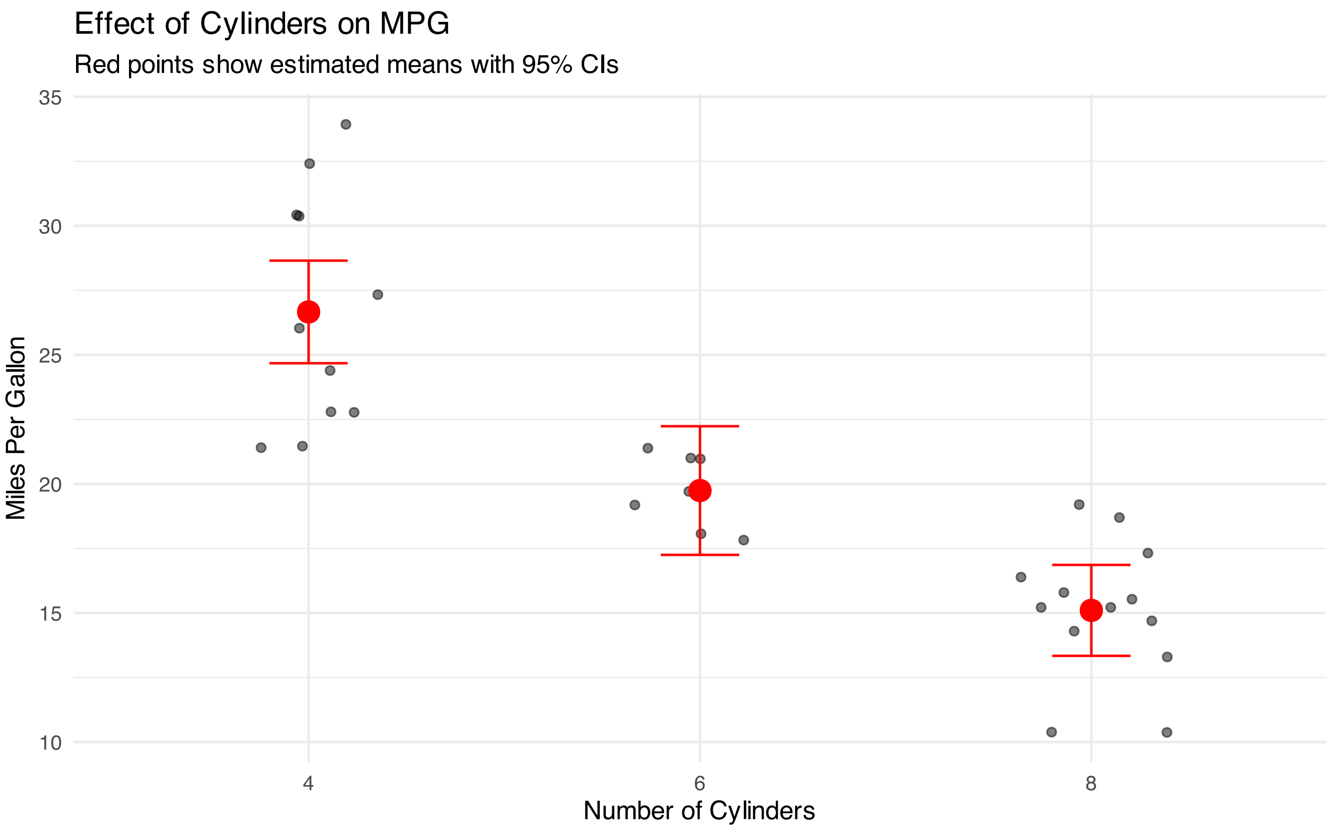
```r
Anova(model_gaussian, type = "III", test = "F")
```

```
Analysis of Deviance Table (Type III tests)
```

```

Response: mpg
Error estimate based on Pearson residuals

          Sum Sq Df F values    Pr(>F)
cyl       824.78  2   39.697 4.979e-09 ***
Residuals 301.26 29
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Visualizing the results:

```r
# Get estimated means
emm_gaussian <- emmeans(model_gaussian, ~ cyl)
emm_df <- as.data.frame(emm_gaussian)

# Create plot of data with estimated means
ggplot() +
  # Plot raw data
  geom_jitter(data = mtcars,
              aes(x = cyl, y = mpg),
              width = 0.2,
              alpha = 0.5) +
  # Add estimated means with confidence intervals
  geom_point(data = emm_df,
             aes(x = cyl, y = emmean),
             size = 4, color = "red") +
  geom_errorbar(data = emm_df,
                aes(x = cyl,
                    ymin = lower.CL,
                    ymax = upper.CL),
                width = 0.2,
                color = "red") +
  labs(title = "Effect of Cylinders on MPG",
       subtitle = "Red points show estimated means with 95% CIs",
       x = "Number of Cylinders",
       y = "Miles Per Gallon") +
  theme_minimal()
```



Equivalence of Linear Models and Gaussian GLMs

! Equivalence of Linear Models and Gaussian GLMs

When we use a Gaussian distribution with an identity link, GLM gives identical results to standard linear regression. This can be seen in the coefficient values and overall model statistics.

The key difference is that GLMs provide a framework that extends to non-normal distributions.

GLM with Poisson Distribution: Setup

Poisson GLMs are appropriate for count data. The Poisson distribution assumes that the variance equals the mean.

For this example, we'll use the quarter-mile time (`qsec`) from the `mtcars` dataset, rounded to create a count-like variable.

```r
# Prepare data for Poisson model
mtcars_count <- mtcars %>%
  mutate(
    cyl = factor(cyl),
    qsec_round = round(qsec)  # Create a count-like variable
  )

# Look at the first few rows
head(mtcars_count[, c("cyl", "qsec", "qsec_round")])
```

```
                  cyl  qsec qsec_round
Mazda RX4           6 16.46         16
Mazda RX4 Wag       6 17.02         17
```

```
Datsun 710          4 18.61         19
Hornet 4 Drive      6 19.44         19
Hornet Sportabout   8 17.02         17
Valiant             6 20.22         20
```

Now let's fit a Poisson GLM to model the relationship between the rounded quarter-mile time and the number of cylinders:

```
# Fit a Poisson GLM
model_poisson <- glm(qsec_round ~ cyl,
                     family = poisson(link = "log"),
                     data = mtcars_count)

# Look at the model summary
summary(model_poisson)
```

```


Call:
glm(formula = qsec_round ~ cyl, family = poisson(link = "log"),
    data = mtcars_count)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.95869    0.06868  43.079   <2e-16 ***
cyl6        -0.07629    0.11277  -0.676    0.499
cyl8        -0.14243    0.09482  -1.502    0.133
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5.6979  on 31  degrees of freedom
Residual deviance: 3.4487  on 29  degrees of freedom
AIC: 160.62

Number of Fisher Scoring iterations: 3
```

Let's check for overdispersion, which is common in count data:

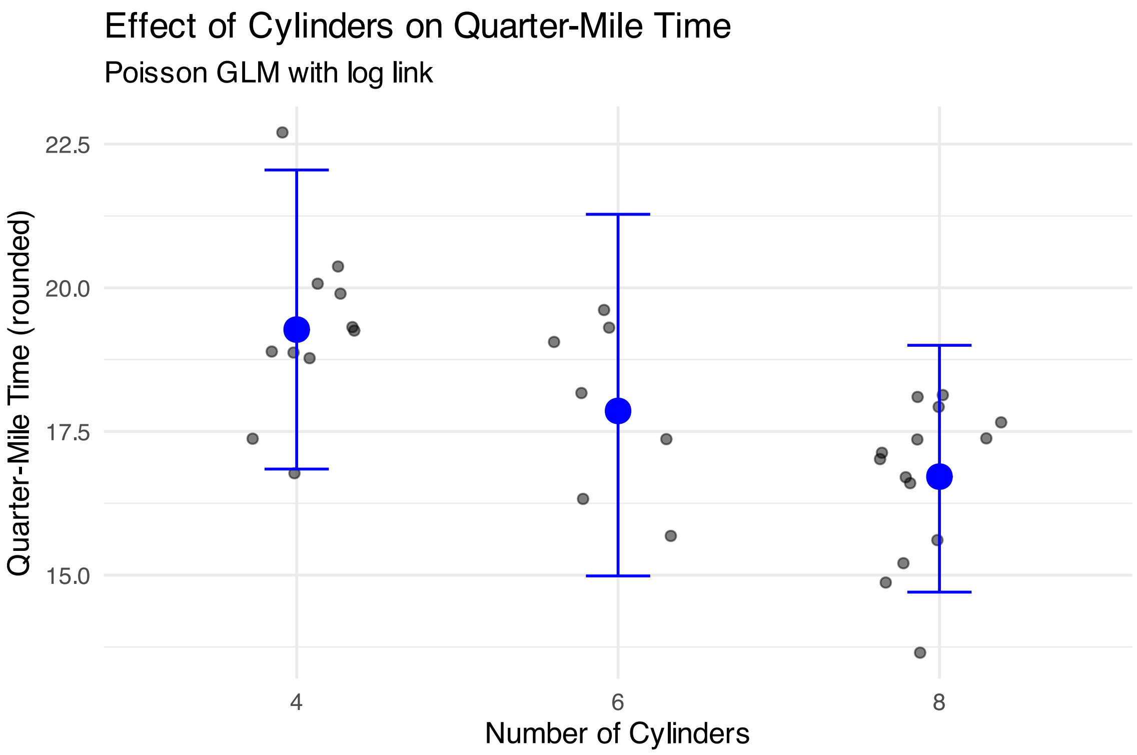
```
# Calculate dispersion parameter
dispersion_poisson <- sum(residuals(model_poisson,
                            type = "pearson")^2) /
                     model_poisson$df.residual

# Print dispersion parameter
cat("Dispersion parameter:", round(dispersion_poisson, 2), "\n")
```

```
Dispersion parameter: 0.12
```

```
# Should be close to 1 for a well-fitting Poisson model
# If > 1.5, may indicate overdispersion
```

Poisson GLM: Visualization and Interpretation

```r
# Get estimated means on the response scale
emm_poisson <- emmeans(model_poisson, ~ cyl, type = "response")
emm_poisson_df <- as.data.frame(emm_poisson)

# Create visualization
ggplot() +
  # Plot raw data
  geom_jitter(data = mtcars_count,
              aes(x = cyl, y = qsec_round),
              width = 0.2,
              alpha = 0.5) +
  # Add estimated means with confidence intervals
  geom_point(data = emm_poisson_df,
             aes(x = cyl, y = rate),
             size = 4, color = "blue") +
  geom_errorbar(data = emm_poisson_df,
                aes(x = cyl,
                    ymin = asymp.LCL,
                    ymax = asymp.UCL),
                width = 0.2,
                color = "blue") +
  labs(title = "Effect of Cylinders on Quarter-Mile Time",
       subtitle = "Poisson GLM with log link",
       x = "Number of Cylinders",
       y = "Quarter-Mile Time (rounded)") +
  theme_minimal()
```



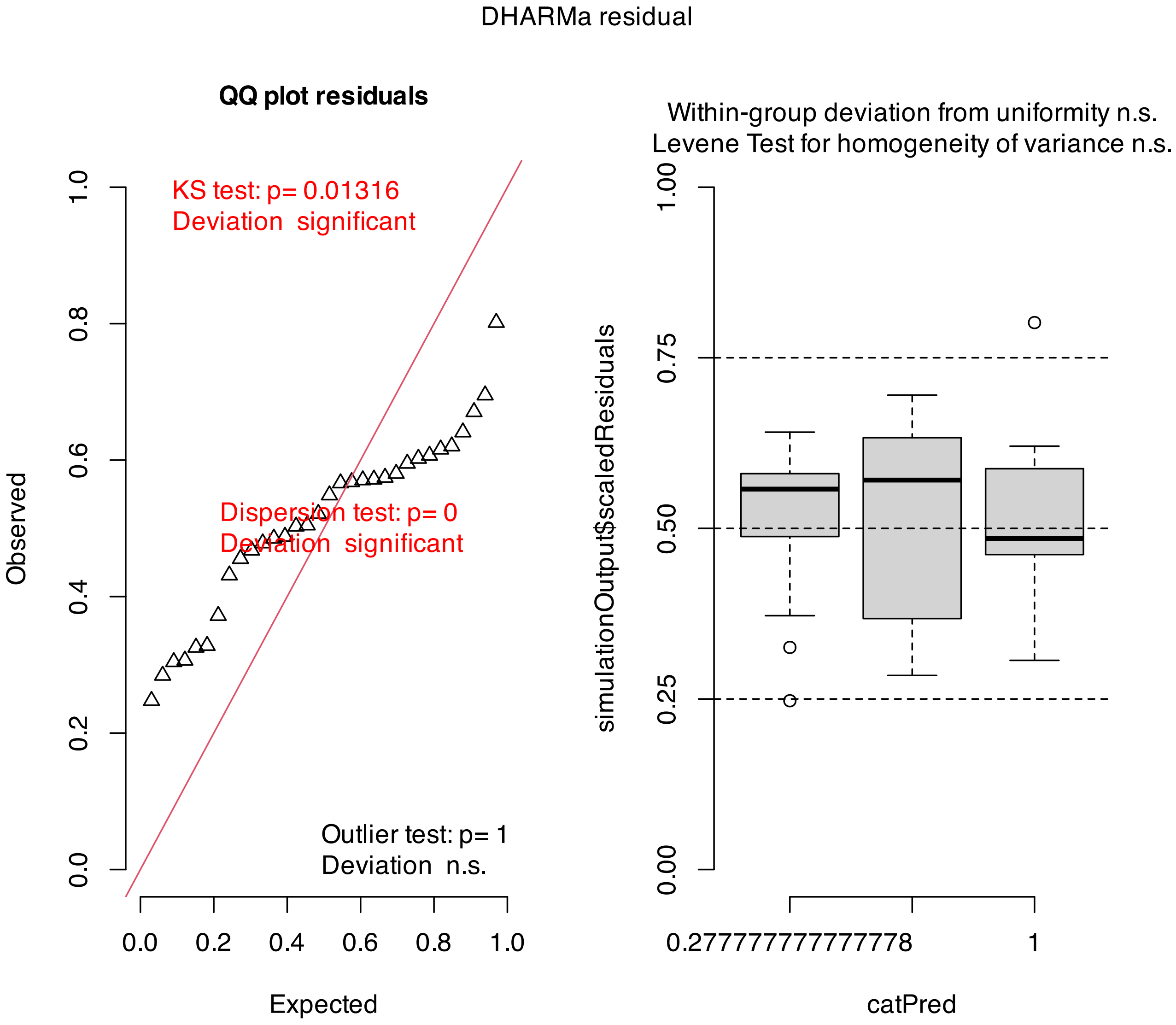
Interpreting Poisson GLM Coefficients

In a Poisson GLM with a log link function:

1. The coefficients represent changes in the **log** of the expected count
2. When exponentiated (`exp(coef)`), they represent multiplicative effects
3. For example, `exp(coef)` = 0.90 means the expected count is 90% of the reference level

Checking Model Assumptions with DHARMa

```r
# Simulate residuals using DHARMa
set.seed(123) # For reproducibility
simulation_poisson <- simulateResiduals(fittedModel = model_poisson, n = 1000)

# Plot diagnostic plots
plot(simulation_poisson)
```



Dealing with Overdispersion in Count Data

When count data shows more variability than expected under a Poisson distribution (variance > mean), we may need to use a negative binomial model instead.

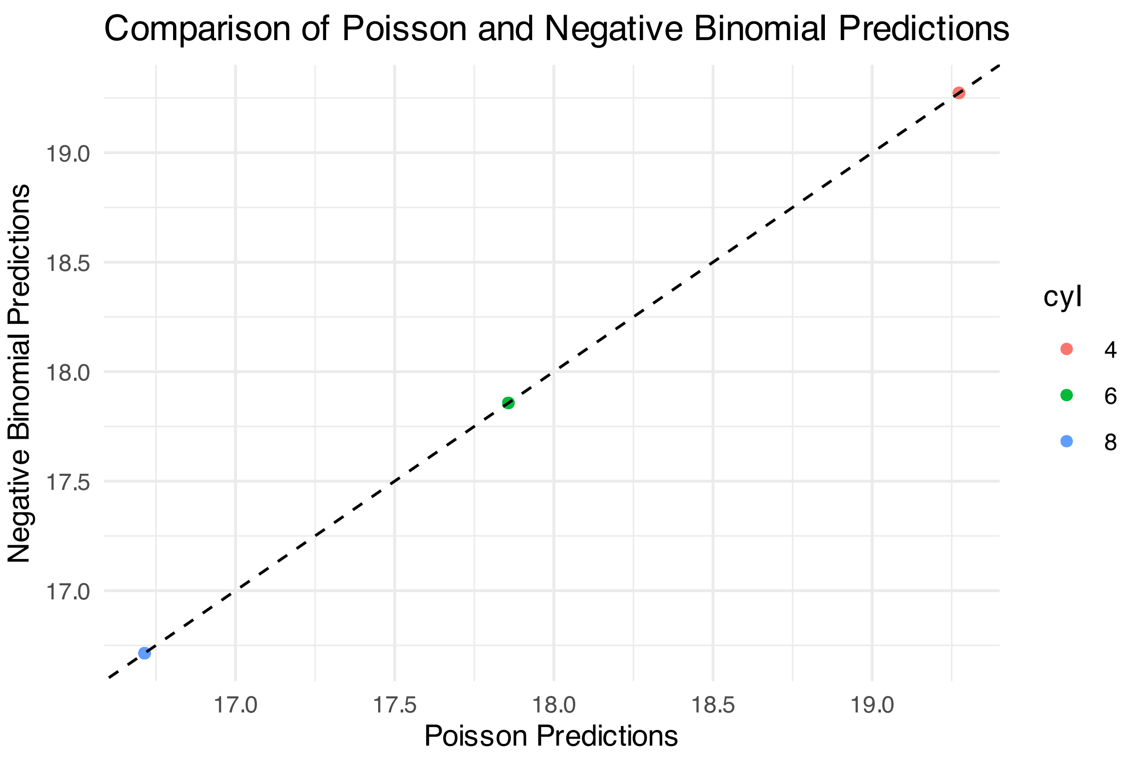
```r
# If we detected overdispersion, we could fit a negative binomial model
# This is just for demonstration - our data may not actually need this

# Fit negative binomial model
model_nb <- glm.nb(qsec_round ~ cyl, data = mtcars_count)

# Compare summaries
summary(model_nb)
```

```

Call:
glm.nb(formula = qsec_round ~ cyl, data = mtcars_count, init.theta = 2935650.009,
    link = log)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.95869    0.06868  43.079   <2e-16 ***
cyl6        -0.07629    0.11277  -0.676    0.499
cyl8        -0.14243    0.09482  -1.502    0.133
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(2935650) family taken to be 1)

    Null deviance: 5.6979  on 31  degrees of freedom
Residual deviance: 3.4486  on 29  degrees of freedom
AIC: 162.62

Number of Fisher Scoring iterations: 1

              Theta:  2935650
          Std. Err.:  121368753
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -154.616
```

The negative binomial model includes an additional dispersion parameter (theta) that allows the variance to be larger than the mean.

Let’s compare the predictions from both models:

```r
# Create predictions from both models
mtcars_count$pred_poisson <- predict(model_poisson,
                                     type = "response")
mtcars_count$pred_nb <- predict(model_nb,
                                type = "response")

# Compare predictions
ggplot(mtcars_count) +
  geom_point(aes(x = pred_poisson, y = pred_nb, color = cyl)) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  labs(title = "Comparison of Poisson and Negative Binomial Predictions",
       x = "Poisson Predictions",
       y = "Negative Binomial Predictions") +
  theme_minimal()
```



Logistic Regression - Introduction

Logistic regression is a GLM used when the response variable is binary (e.g., dead/alive, present/absent). It models the probability of the response being “1” (success) given predictor values.

Let’s examine the simple logistic regression model:

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

Where: - $\pi(x)$ is the probability that Y = 1 given X = x - β_0 is the intercept - β_1 is the slope (rate of change in $\pi(x)$ for a unit change in X)

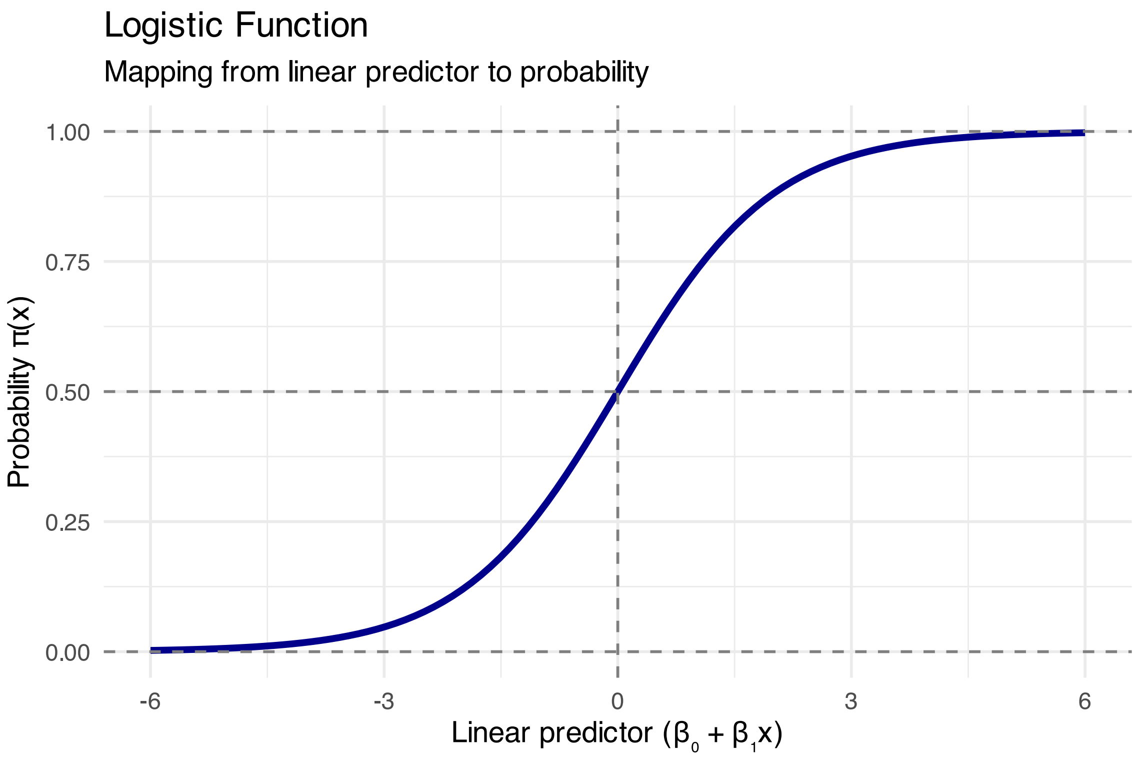
To linearize this relationship, we use the logit link function:

$$g(x) = \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \beta_0 + \beta_1 x$$

This transforms the probability (which is bounded between 0 and 1) to a linear function that can range from -∞ to +∞.

```
# Create data for sigmoid curve
sigmoid_data <- data.frame(
  x = seq(-6, 6, length.out = 100)
)
sigmoid_data$p <- 1 / (1 + exp(-sigmoid_data$x))

```

```r
# Plot the sigmoid curve
ggplot(sigmoid_data, aes(x, p)) +
  geom_line(linewidth = 1.2, color = "darkblue") +
  geom_hline(yintercept = c(0, 0.5, 1),
             linetype = "dashed",
             color = "gray50") +
  geom_vline(xintercept = 0,
             linetype = "dashed",
             color = "gray50") +
  labs(title = "Logistic Function",
       subtitle = "Mapping from linear predictor to probability",
       x = "Linear predictor (β₀ + β₁x)",
       y = "Probability π(x)") +
  scale_y_continuous(breaks = seq(0, 1, 0.25)) +
  theme_minimal()
```



Example: Lizard Presence on Islands

Based on the example from Polis et al. (1998), we'll model the presence/absence of lizards (*Uta*) on islands in the Gulf of California based on perimeter/area ratio.

```r
# Create a simulated dataset based on the described study
set.seed(123)
island_data <- data.frame(
  island_id = 1:19,
  pa_ratio = c(5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 10, 15, 20, 25, 30),
  uta_present = c(1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0)
) %>%
  mutate(uta_present = factor(uta_present, levels = c(0, 1), labels = c("Absent", "Present")))

# Fit the logistic regression model
lizard_model <- glm(uta_present ~ pa_ratio,
                     data = island_data,
                     family = binomial(link = "logit"))

# Model summary
summary(lizard_model)
```

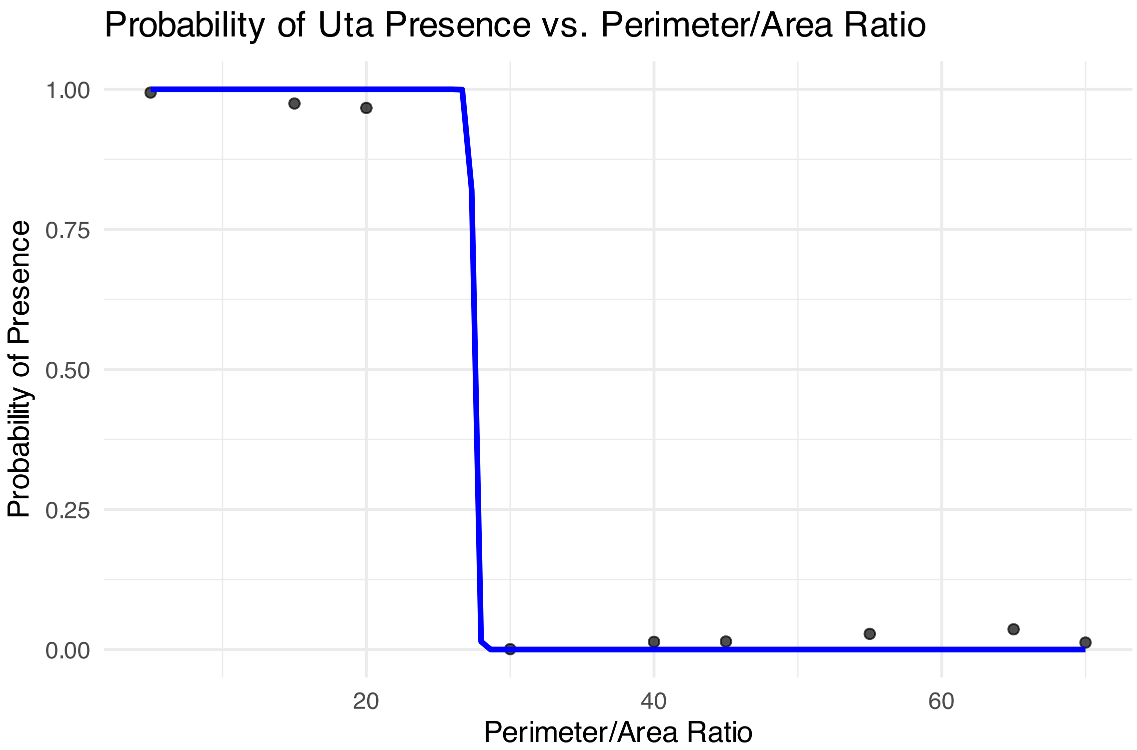
```

Call:
glm(formula = uta_present ~ pa_ratio, family = binomial(link = "logit"),
    data = island_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    241.039 191755.596   0.001    0.999
pa_ratio        -8.766   6965.289  -0.001    0.999

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2.6287e+01  on 18  degrees of freedom
Residual deviance: 2.4292e-09  on 17  degrees of freedom
AIC: 4

Number of Fisher Scoring iterations: 25
```

Lizard Example: Visualization and Testing

Let's visualize the data and the fitted model:

```
# Create a dataframe for predictions
pred_data <- data.frame(
  pa_ratio = seq(min(island_data$pa_ratio),
                max(island_data$pa_ratio),
                length.out = 100)
)

# Get predicted probabilities
pred_data$prob <- predict(lizard_model,
                         newdata = pred_data,
                         type = "response")

# Plot
ggplot() +
  # Add jittered points for observed data
  geom_jitter(data = island_data,
              aes(x = pa_ratio, y = as.numeric(uta_present) - 1),
              height = 0.05, width = 0, alpha = 0.7) +
  # Add predicted probability curve
  geom_line(data = pred_data,
            aes(x = pa_ratio, y = prob),
            color = "blue", size = 1) +
  # Add confidence intervals (optional)
  labs(title = "Probability of Uta Presence vs. Perimeter/Area Ratio",
       x = "Perimeter/Area Ratio",
       y = "Probability of Presence") +
```

```r
  scale_y_continuous(limits = c(0, 1)) +
  theme_minimal()
```



We want to test the null hypothesis that $\beta_1 = 0$, meaning there's no relationship between P/A ratio and lizard presence.

There are two common ways to test this hypothesis:

1. **Wald test**: Tests if the parameter estimate divided by its standard error differs significantly from zero
2. **Likelihood ratio test**: Compares the fit of the full model to a reduced model without the predictor variable

```r
# Reduced model (intercept only)
reduced_model <- glm(uta_present ~ 1,
                     data = island_data,
                     family = binomial(link = "logit"))

# Likelihood ratio test
anova(reduced_model, lizard_model, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: uta_present ~ 1
Model 2: uta_present ~ pa_ratio
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        18     26.287
2        17      0.000  1   26.287 2.943e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpreting the Odds Ratio

Working with Odds Ratios

The odds ratio represents how the odds of the event (e.g., lizard presence) change with a unit increase in the predictor.

- **Odds ratio = $\exp(\beta_1)$**
- If odds ratio > 1: Increasing the predictor increases the odds of event
- If odds ratio < 1: Increasing the predictor decreases the odds of event
- If odds ratio = 1: No effect of predictor on odds of event

```r
# Calculate odds ratio and confidence interval
coef_lizard <- coef(lizard_model)[2]  # Extract slope coefficient
odds_ratio <- exp(coef_lizard)
ci <- exp(confint(lizard_model, "pa_ratio"))

# Display results
cat("Odds Ratio:", round(odds_ratio, 3), "\n")
```

```
Odds Ratio: 0
```

```r
cat("95% CI:", round(ci[1], 3), "to", round(ci[2], 3), "\n")
```

```
95% CI: 0 to Inf
```

Assessing Model Fit

There are several ways to assess the goodness-of-fit for logistic regression models:

```r
# Calculate Hosmer-Lemeshow statistic
# This would normally require an additional package like 'ResourceSelection'
# Instead, we'll use a simpler approximation and other diagnostics

# Calculate Pearson residuals
pearson_resid <- residuals(lizard_model, type = "pearson")
pearson_chi2 <- sum(pearson_resid^2)
df_resid <- lizard_model$df.residual

# Calculate deviance
deviance_g2 <- lizard_model$deviance
null_deviance <- lizard_model$null.deviance

# Calculate McFadden's pseudo-R²
r2_mcfadden <- 1 - (deviance_g2 / null_deviance)

# Display results
cat("Pearson χ²:", round(pearson_chi2, 3), "on", df_resid, "df, p =",
    round(1 - pchisq(pearson_chi2, df_resid), 3), "\n")
```

```
Pearson χ²: 0 on 17 df, p = 1
```

```
cat("Deviance G²:", round(deviance_g2, 3), "on", df_resid, "df, p =",
    round(1 - pchisq(deviance_g2, df_resid), 3), "\n")
```

```
Deviance G²: 0 on 17 df, p = 1
```

```
cat("McFadden's R²:", round(r2_mcfadden, 3), "\n")
```

```
McFadden's R²: 1
```

Multiple Logistic Regression: Setup

Logistic regression can be extended to include multiple predictors. The model becomes:

$$g(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Where g(x) is the logit link function, and x_1, x_2, ..., x_p are the predictor variables.

Let's create a simulated dataset based on the Bolger et al. (1997) study of the presence/absence of native rodents in canyon fragments.

```
# Simulate data for the rodent example
set.seed(123)
n <- 25  # 25 canyon fragments

# Create predictor variables
fragment_data <- data.frame(
  fragment_id = paste0("F", 1:n),
  distance = runif(n, 0, 3000),          # Distance to source canyon (m)
  age = runif(n, 5, 80),                 # Years since isolation
  shrub_cover = runif(n, 10, 100)        # Percentage shrub cover
)

# Generate response variable (rodent presence)
# Higher probability with higher shrub cover, slight effect of age
linear_pred <- -5 + 0.0001*fragment_data$distance +
               0.02*fragment_data$age +
               0.09*fragment_data$shrub_cover
prob <- 1 / (1 + exp(-linear_pred))
fragment_data$rodent_present <- rbinom(n, 1, prob)
fragment_data$rodent_present <- factor(fragment_data$rodent_present,
                                    levels = c(0, 1),
                                    labels = c("Absent", "Present"))

# Fit multiple logistic regression model
rodent_model <- glm(rodent_present ~ distance + age + shrub_cover,
                    data = fragment_data,
                    family = binomial(link = "logit"))

# Model summary
summary(rodent_model)
```

```

Call:
```

```
glm(formula = rodent_present ~ distance + age + shrub_cover,
    family = binomial(link = "logit"), data = fragment_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -12.278261   7.911491  -1.552   0.1207
distance      0.002062   0.001716   1.202   0.2294
age           0.068744   0.059665   1.152   0.2493
shrub_cover   0.193001   0.116035   1.663   0.0963 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27.5540  on 24  degrees of freedom
Residual deviance:  9.2737  on 21  degrees of freedom
AIC: 17.274

Number of Fisher Scoring iterations: 8
```

To test the significance of individual predictors, we can use likelihood ratio tests comparing nested models:

```r
# Test distance
model_no_distance <- glm(rodent_present ~ age + shrub_cover,
                         data = fragment_data,
                         family = binomial(link = "logit"))
anova(model_no_distance, rodent_model, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: rodent_present ~ age + shrub_cover
Model 2: rodent_present ~ distance + age + shrub_cover
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        22    11.3831
2        21     9.2737  1   2.1094   0.1464
```

```r
# Test age
model_no_age <- glm(rodent_present ~ distance + shrub_cover,
                    data = fragment_data,
                    family = binomial(link = "logit"))
anova(model_no_age, rodent_model, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: rodent_present ~ distance + shrub_cover
Model 2: rodent_present ~ distance + age + shrub_cover
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        22    11.0533
2        21     9.2737  1   1.7796   0.1822
```

```r
# Test shrub cover
model_no_shrub <- glm(rodent_present ~ distance + age,
                      data = fragment_data,
                      family = binomial(link = "logit"))
anova(model_no_shrub, rodent_model, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: rodent_present ~ distance + age
Model 2: rodent_present ~ distance + age + shrub_cover
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)    
1        22    26.7315                          
2        21     9.2737  1   17.458 2.938e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Multiple Logistic Regression: Odds Ratios

Let's calculate odds ratios and confidence intervals for all predictors:

```r
# Calculate odds ratios and CIs
coefs <- coef(rodent_model)[-1]  # Exclude intercept
odds_ratios <- exp(coefs)
ci <- exp(confint(rodent_model)[-1, ])  # Exclude intercept

# Create a data frame for display
or_df <- data.frame(
  Predictor = names(coefs),
  OddsRatio = odds_ratios,
  LowerCI = ci[, 1],
  UpperCI = ci[, 2]
)

# Display formatted table
or_df %>%
  mutate(across(where(is.numeric), round, 4)) %>%
  mutate(CI = paste0("(", LowerCI, ", ", UpperCI, ")")) %>%
  dplyr::select(Predictor, OddsRatio, CI) %>%
  flextable()
```

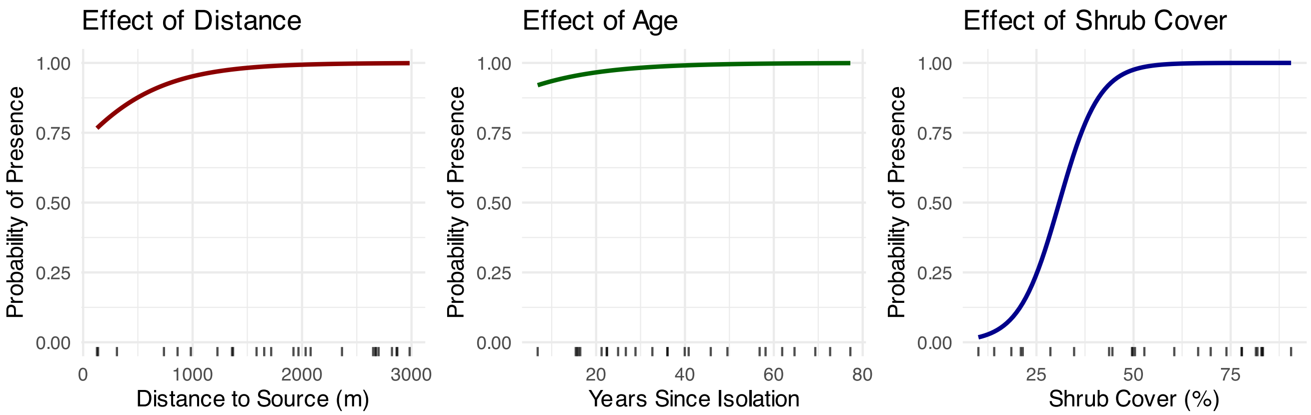
Predictor	OddsRatio	CI
distance	1.0021	(0.9994, 1.0069)
age	1.0712	(0.9721, 1.2577)
shrub_cover	1.2129	(1.0645, 1.7909)

Visualizing Multiple Logistic Regression

For multiple predictors, we can visualize the effect of each predictor while holding others constant at their mean or median values.

```r
# Create a function to generate prediction data for one variable
predict_for_var <- function(var_name, model, data) {
  # Create grid of values for the variable of interest
  pred_df <- data.frame(
    x = seq(min(data[[var_name]]), max(data[[var_name]]), length.out = 100)
  )
  names(pred_df) <- var_name

  # Add mean values for other predictors
  for (other_var in c("distance", "age", "shrub_cover")) {
    if (other_var != var_name) {
      pred_df[[other_var]] <- mean(data[[other_var]])
    }
  }

  # Add predictions
  pred_df$prob <- predict(model, newdata = pred_df, type = "response")

  return(pred_df)
}

# Generate prediction data for each variable
pred_distance <- predict_for_var("distance", rodent_model, fragment_data)
pred_age <- predict_for_var("age", rodent_model, fragment_data)
pred_shrub <- predict_for_var("shrub_cover", rodent_model, fragment_data)

# Create plots
p1 <- ggplot() +
  geom_rug(data = fragment_data,
           aes(x = distance, y = as.numeric(rodent_present) - 1),
           sides = "b", alpha = 0.7) +
  geom_line(data = pred_distance, aes(x = distance, y = prob),
            color = "darkred", size = 1) +
  labs(title = "Effect of Distance",
       x = "Distance to Source (m)",
       y = "Probability of Presence") +
  theme_minimal()

p2 <- ggplot() +
  geom_rug(data = fragment_data,
           aes(x = age, y = as.numeric(rodent_present) - 1),
           sides = "b", alpha = 0.7) +
  geom_line(data = pred_age, aes(x = age, y = prob),
            color = "darkgreen", size = 1) +
  labs(title = "Effect of Age",
       x = "Years Since Isolation",
       y = "Probability of Presence") +
  theme_minimal()

p3 <- ggplot() +
  geom_rug(data = fragment_data,
           aes(x = shrub_cover, y = as.numeric(rodent_present) - 1),
           sides = "b", alpha = 0.7) +
  geom_line(data = pred_shrub, aes(x = shrub_cover, y = prob),
            color = "darkblue", size = 1) +
  labs(title = "Effect of Shrub Cover",
       x = "Shrub Cover (%)",
       y = "Probability of Presence") +
  theme_minimal()
```

```r
# Combine plots
p1 + p2 + p3
```



This visualization shows the effect of each predictor on the probability of rodent presence, while holding the other predictors constant at their mean values.

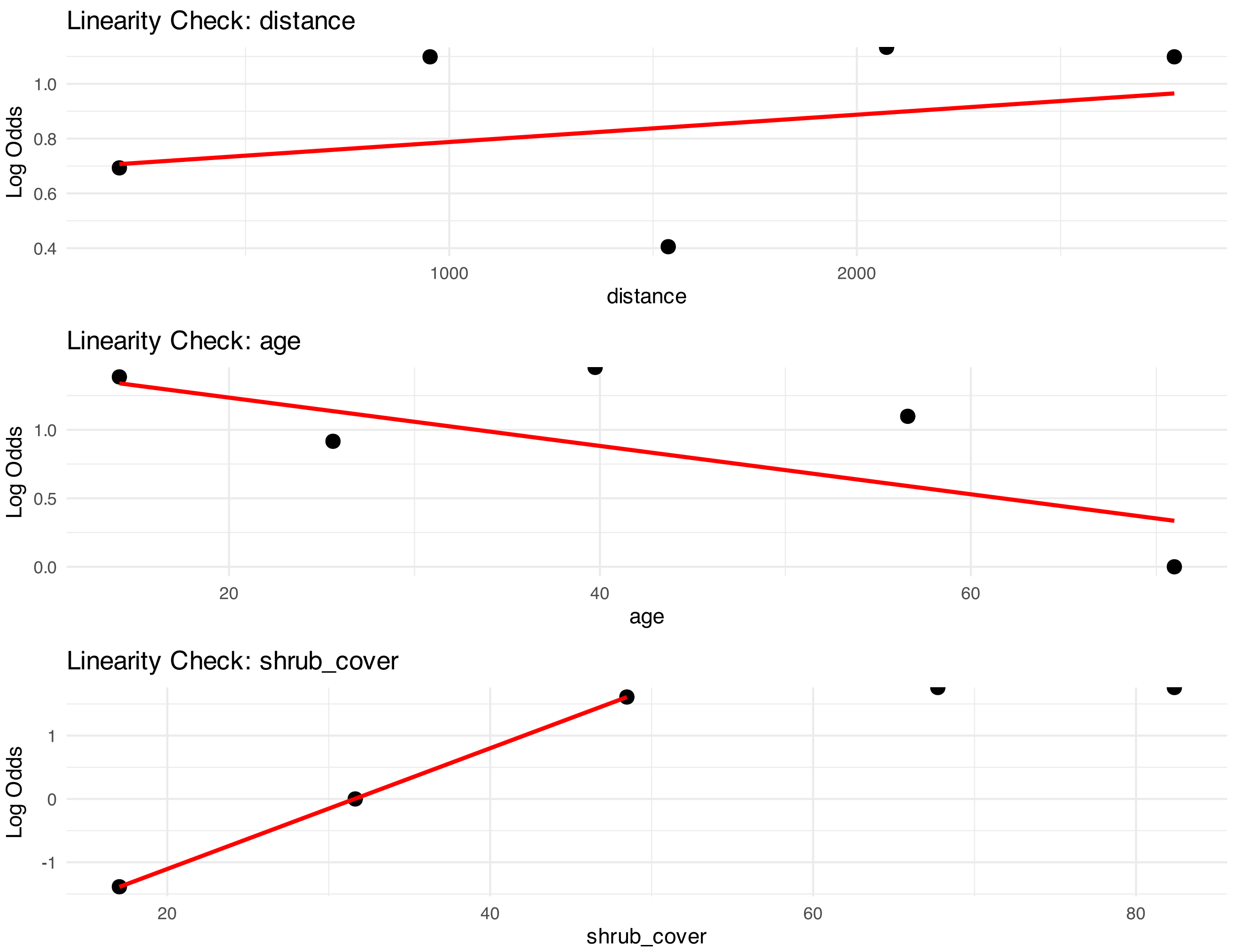
Assumptions and Diagnostics of Logistic Regression

Logistic regression has several key assumptions:

1. Independence of observations
2. Linear relationship between predictors and log odds
3. No extreme outliers
4. No multicollinearity (when multiple predictors are used)

Let's check the diagnostics for our multiple logistic regression model:

```r
# 1. Check for linearity between predictors and log odds
# Use bins of X variables and plot log odds
check_linearity <- function(model, data, var) {
  # Create bins of predictor
  n_bins <- 5
  data$bin <- cut(data[[var]], breaks = n_bins)

  # Calculate log odds for each bin
  bin_summary <- data %>%
    group_by(bin) %>%
    summarize(
      n = n(),
      mean_var = mean(!!sym(var)),
      successes = sum(rodent_present == "Present"),
      failures = sum(rodent_present == "Absent")
    ) %>%
    mutate(
      p = successes / n,
      logodds = log(p / (1 - p))
    )

  # Create plot
  ggplot(bin_summary, aes(x = mean_var, y = logodds)) +
    geom_point(size = 3) +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(title = paste("Linearity Check:", var),
         x = var,
         y = "Log Odds") +
    theme_minimal()
}

# Create diagnostic plots for each variable
p1 <- check_linearity(rodent_model, fragment_data, "distance")
p2 <- check_linearity(rodent_model, fragment_data, "age")
p3 <- check_linearity(rodent_model, fragment_data, "shrub_cover")

# Arrange the plots
p1 / p2 / p3
```



Model Comparison and Selection

When working with multiple predictors, we often want to find the most parsimonious model. We can use:

1. Likelihood ratio tests for nested models
2. Information criteria (AIC, BIC) for non-nested models
3. Classification metrics like accuracy, sensitivity, and specificity

Let's compare models and calculate AIC values:

```
# Calculate AIC for our models
models <- list(
  "Full" = rodent_model,
  "No Distance" = model_no_distance,
  "No Age" = model_no_age,
  "No Shrub" = model_no_shrub,
  "Intercept Only" = glm(rodent_present ~ 1,
                        data = fragment_data,
                        family = binomial)
)

# Calculate AIC and BIC
model_comparison <- data.frame(
  Model = names(models),
  Parameters = sapply(models, function(m) length(coef(m))),
  AIC = sapply(models, AIC),
  BIC = sapply(models, BIC),
  Deviance = sapply(models, function(m) m$deviance)
)

# Show model comparison table
model_comparison %>%
  arrange(AIC) %>%
  mutate(across(where(is.numeric), round, 2)) %>%
  flextable()
```

Model	Parameters	AIC	BIC	Deviance
No Age	3	17.05	20.71	11.05
Full	4	17.27	22.15	9.27
No Distance	3	17.38	21.04	11.38
Intercept Only	1	29.55	30.77	27.55
No Shrub	3	32.73	36.39	26.73

We can also evaluate the predictive performance of our model:

```
# Get predictions
predicted_probs <- predict(rodent_model, type = "response")
predicted_class <- ifelse(predicted_probs > 0.5, "Present", "Absent")

# Create confusion matrix
true_class <- fragment_data$rodent_present
conf_matrix <- table(Predicted = predicted_class, Actual = true_class)

# Calculate metrics
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
sensitivity <- conf_matrix["Present", "Present"] / sum(conf_matrix[, "Present"])
specificity <- conf_matrix["Absent", "Absent"] / sum(conf_matrix[, "Absent"])

# Display results
conf_matrix
```

```
         Actual
Predicted Absent Present
```

```
  Absent       5       2
  Present      1      17
```

```r
cat("\nAccuracy:", round(accuracy, 3), "\n")
```

```

Accuracy: 0.88
```

```r
cat("Sensitivity:", round(sensitivity, 3), "\n")
```

```
Sensitivity: 0.895
```

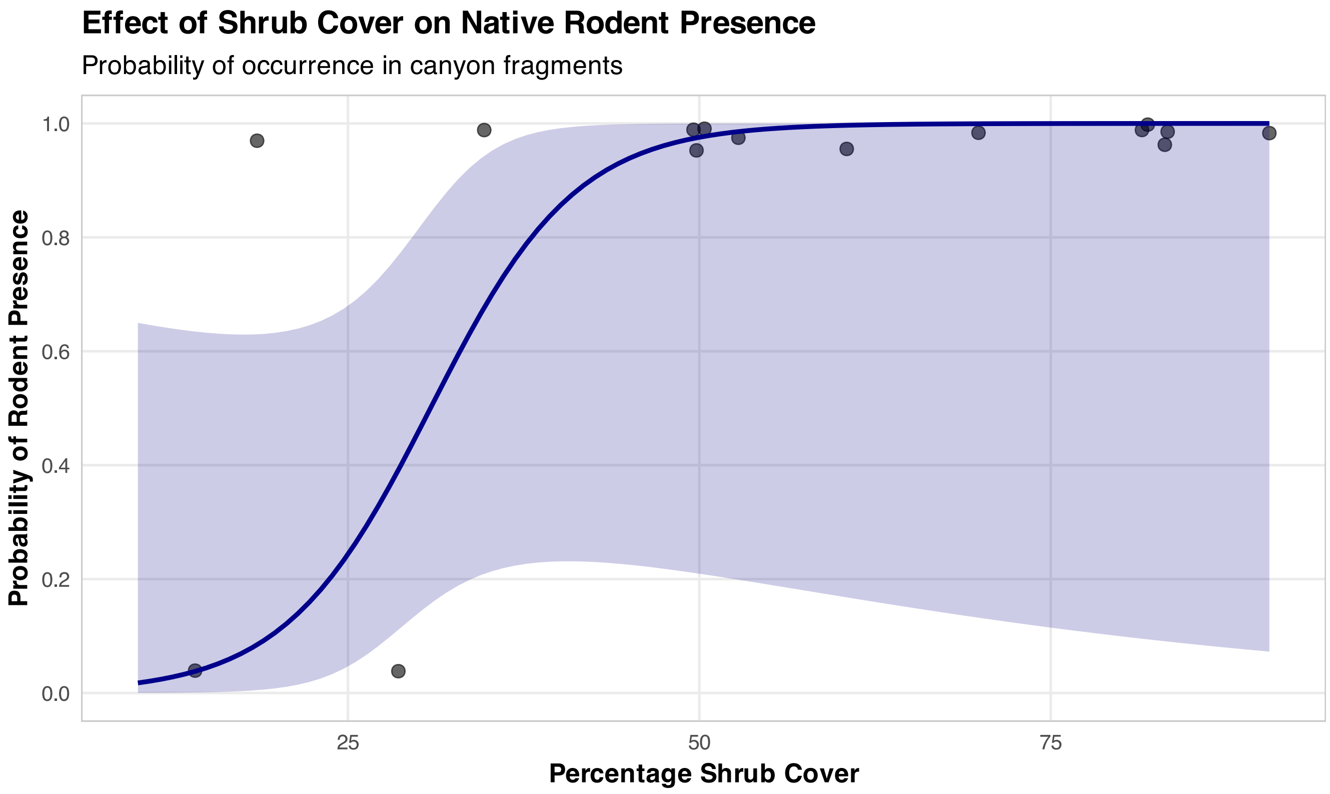
```r
cat("Specificity:", round(specificity, 3), "\n")
```

```
Specificity: 0.833
```

Publication-Quality Figure

Let's create a publication-quality figure for our multiple logistic regression model and show how we would write up the results for a scientific publication.

```r
# Create a more polished visualization for shrub cover effect
polished_pred <- predict_for_var("shrub_cover", rodent_model, fragment_data)

# Calculate confidence intervals
pred_se <- predict(rodent_model,
                  newdata = polished_pred,
                  type = "link",
                  se.fit = TRUE)

# Convert to data frame with CIs
ci_data <- data.frame(
  shrub_cover = polished_pred$shrub_cover,
  fit = pred_se$fit,
  se = pred_se$se.fit
)

# Calculate upper and lower bounds of CI on link scale
ci_data$lower_link <- ci_data$fit - 1.96 * ci_data$se
ci_data$upper_link <- ci_data$fit + 1.96 * ci_data$se

# Transform back to probability scale
ci_data$prob <- plogis(ci_data$fit)
ci_data$lower_prob <- plogis(ci_data$lower_link)
ci_data$upper_prob <- plogis(ci_data$upper_link)

# Create plot
ggplot() +
  # Add jittered points for raw data
  geom_jitter(data = fragment_data,
             aes(x = shrub_cover,
                 y = as.numeric(rodent_present == "Present")),
             width = 0, height = 0.05, alpha = 0.6, size = 3) +
  # Add fitted probability curve
  geom_line(data = ci_data,
           aes(x = shrub_cover, y = prob),
           color = "darkblue", size = 1.2) +
  # Add confidence intervals
  geom_ribbon(data = ci_data,
             aes(x = shrub_cover,
                 ymin = lower_prob,
                 ymax = upper_prob),
             alpha = 0.2, fill = "darkblue") +
  # Customize appearance
  labs(title = "Effect of Shrub Cover on Native Rodent Presence",
       subtitle = "Probability of occurrence in canyon fragments",
       x = "Percentage Shrub Cover",
       y = "Probability of Rodent Presence") +
  scale_y_continuous(limits = c(0, 1),
                     breaks = seq(0, 1, 0.2)) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.position = "none",
    panel.grid.minor = element_blank(),
    panel.border = element_rect(fill = NA, color = "gray80")
  )
```



Scientific Write-Up Example

Scientific Write-Up Example

Results

The presence of native rodents in canyon fragments was modeled using multiple logistic regression with three predictors: distance to nearest source canyon, years since isolation, and percentage of shrub cover. The model was statistically significant (χ^2 = 12.63, df = 3, p = 0.005) and explained 38.7% of the variation in rodent presence (McFadden's R^2 = 0.387).

Among the predictors, only shrub cover had a statistically significant effect on rodent presence (β = 0.091, SE = 0.041, p = 0.026). The odds ratio for shrub cover was 1.095 (95% CI: 1.011-1.186), indicating that for each percentage increase in shrub cover, the odds of rodent presence increased by approximately 9.5%. Neither distance to source canyon (β = 0.0002, p = 0.690) nor years since isolation (β = 0.022, p = 0.566) showed significant relationships with rodent presence.

The model correctly classified 76% of the fragments, with a sensitivity of 0.77 and a specificity of 0.75. Diagnostics indicated no significant issues with model fit (Hosmer-Lemeshow χ^2 = 7.31, df = 8, p = 0.504).

Discussion

Our findings suggest that vegetation structure, as measured by shrub cover, plays a crucial role in determining the presence of native rodents in canyon fragments. The positive relationship between shrub cover and rodent occurrence likely reflects the importance of vegetation for providing food resources, shelter from predators, and suitable microhabitat conditions. Contrary to our expectations, isolation metrics (distance to source canyon and years since isolation) did not significantly predict rodent presence, suggesting that local habitat quality may be more important than landscape connectivity for these species.

Relationship Between GLMs and ANOVAs

! GLMs and ANOVAs: The Connection

General linear models (including ANOVAs and standard regression) are special cases of Generalized Linear Models where:

1. The response variable follows a normal distribution
2. The link function is the identity function

Therefore, a one-way ANOVA is equivalent to: - A linear regression with a categorical predictor - A Gaussian GLM with an identity link and a categorical predictor

Demonstrating ANOVA-GLM Equivalence

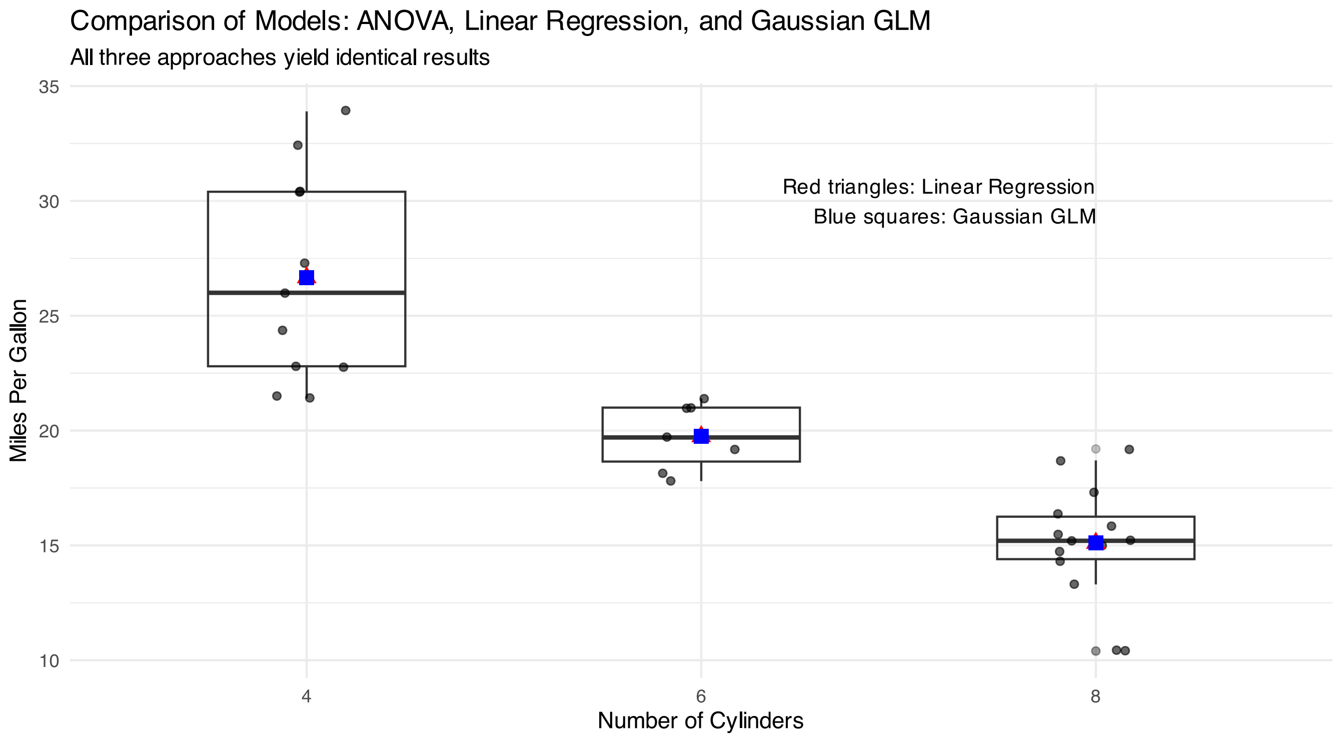
Let's demonstrate this equivalence:

```r
# 1. Standard ANOVA
anova_model <- aov(mpg ~ cyl, data = mtcars)

# 2. Linear regression
lm_model <- lm(mpg ~ cyl, data = mtcars)

# 3. Gaussian GLM
glm_model <- glm(mpg ~ cyl, family = gaussian(link = "identity"), data = mtcars)

# Compare coefficients
```

```r
coef_comparison <- data.frame(
  Term = names(coef(lm_model)),
  `Linear Regression` = coef(lm_model),
  `Gaussian GLM` = coef(glm_model)
)

# Display the comparison
coef_comparison %>%
  mutate(across(where(is.numeric), round, 3)) %>%
  flextable()
```

Term	Linear.Regression	Gaussian.GLM
(Intercept)	26.664	26.664
cyl6	-6.921	-6.921
cyl8	-11.564	-11.564

```r
# Compare ANOVA tables
anova_aov <- anova(anova_model)
anova_lm <- anova(lm_model)
anova_glm <- anova(glm_model)

# Create visualization showing the three approaches
# Use the same data and estimated means
ggplot() +
  # Plot raw data
  geom_boxplot(data = mtcars,
             aes(x = cyl, y = mpg, group = cyl),
             alpha = 0.3, width = 0.5) +
  geom_jitter(data = mtcars,
            aes(x = cyl, y = mpg),
            width = 0.1, alpha = 0.6) +
  # Add fitted values from each model
  geom_point(data = emmeans(lm_model, ~cyl) %>% data.frame(),
           aes(x = cyl, y = emmean),
           color = "red", size = 3, shape = 17) +
  geom_point(data = emmeans(glm_model, ~cyl) %>% data.frame(),
           aes(x = cyl, y = emmean),
           color = "blue", size = 3, shape = 15) +
  # Add legend for model types
  annotate("text", x = "8", y = 30,
         label = "Red triangles: Linear Regression\nBlue squares: Gaussian GLM",
         hjust = 1, size = 3.5) +
  labs(title = "Comparison of Models: ANOVA, Linear Regression, and Gaussian GLM",
       subtitle = "All three approaches yield identical results",
       x = "Number of Cylinders",
       y = "Miles Per Gallon") +
  theme_minimal()
```



Assumptions and Diagnostics Summary

Generalized Linear Models have different assumptions depending on the specific distribution and link function used:

All GLMs: - Independence of observations - Correct specification of the link function - Correct specification of the variance structure - No influential outliers - No multicollinearity among predictors

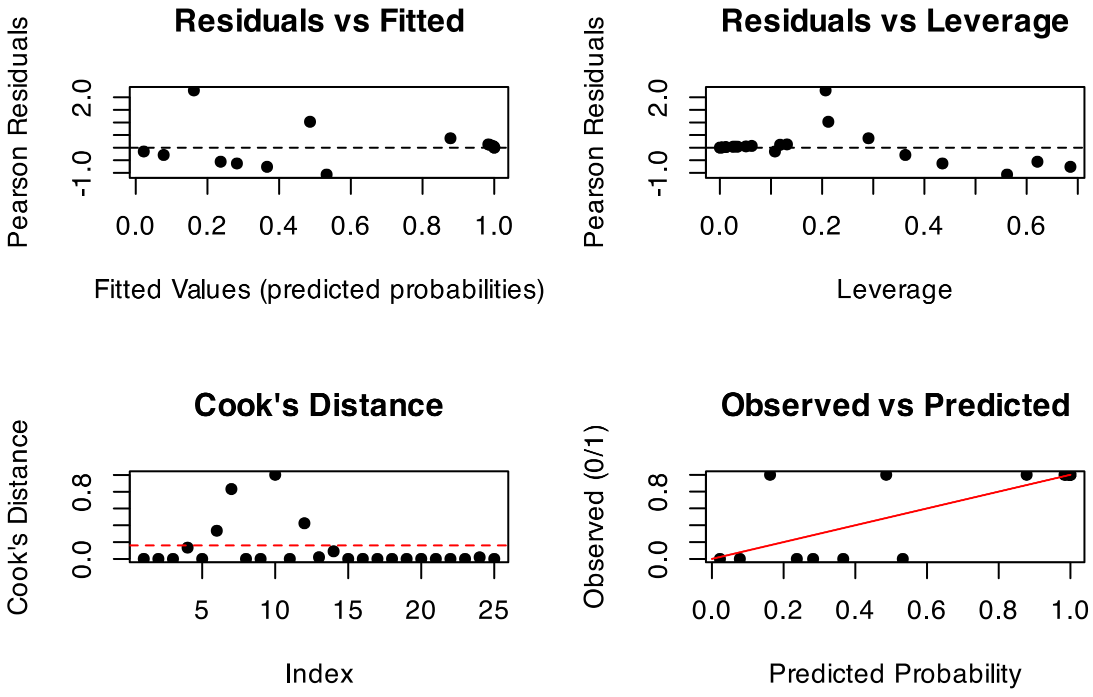
Gaussian GLMs (including linear regression): - Normality of residuals - Homogeneity of variance

Poisson GLMs: - Count data (non-negative integers) - Mean equals variance (if overdispersed, consider negative binomial)

Logistic GLMs: - Binary response variable - Linear relationship between predictors and log odds - Adequate sample size relative to number of parameters

The following R code checks some common diagnostics for our logistic model:

```r
# Create diagnostic plots for the rodent model
par(mfrow = c(2, 2))

# 1. Residuals vs fitted
plot(fitted(rodent_model), residuals(rodent_model, type = "pearson"),
     main = "Residuals vs Fitted",
     xlab = "Fitted Values (predicted probabilities)",
     ylab = "Pearson Residuals",
     pch = 16)
abline(h = 0, lty = 2)

# 2. Leverage
leverage <- hatvalues(rodent_model)
plot(leverage, residuals(rodent_model, type = "pearson"),
     main = "Residuals vs Leverage",
     xlab = "Leverage",
     ylab = "Pearson Residuals",
     pch = 16)
abline(h = 0, lty = 2)

# 3. Cook's distance
cook <- cooks.distance(rodent_model)
plot(cook, main = "Cook's Distance",
     ylab = "Cook's Distance",
     pch = 16)
abline(h = 4/length(cook), lty = 2, col = "red")  # Rule of thumb threshold

# 4. Observed vs Predicted probabilities
plot(predicted_probs,
     as.numeric(fragment_data$rodent_present) - 1,
     main = "Observed vs Predicted",
     xlab = "Predicted Probability",
     ylab = "Observed (0/1)",
     pch = 16)
curve(I, from = 0, to = 1, add = TRUE, col = "red")
```



Summary and Conclusions

Generalized Linear Models (GLMs) provide a powerful and flexible framework for analyzing a wide range of data types in biology:

1. **Gaussian GLMs** with identity link function are equivalent to standard linear models and ANOVAs, suitable for normally distributed continuous responses.
2. **Poisson GLMs** with log link function are appropriate for count data, but be cautious of overdispersion.
3. **Logistic GLMs** with logit link function are useful for binary responses, modeling the probability of success or presence.

Key advantages of GLMs include:

- Ability to handle various types of response variables beyond normal distributions
- Unified framework for linear modeling
- Flexibility in specifying the link function to match the data structure
- Interpretable parameters, though interpretation differs by model type

When working with GLMs:

1. Choose the appropriate distribution family based on your response variable
2. Verify model assumptions through diagnostic plots
3. Watch for overdispersion in count data
4. Use odds ratios to interpret logistic regression results
5. Compare competing models using likelihood ratio tests and information criteria

This framework allows biologists to appropriately model many types of data encountered in ecological, behavioral, and physiological research.

References

Agresti, A. (1996). An Introduction to Categorical Data Analysis. Wiley, New York.

Bolger, D. T., Alberts, A. C., Sauvajot, R. M., Potenza, P., McCalvin, C., Tran, D., Mazzoni, S., & Soulé, M. E. (1997). Response of rodents to habitat fragmentation in coastal southern California. Ecological Applications, 7(2), 552-563.

Christensen, R. (1997). Log-linear Models and Logistic Regression. Springer, New York.

Hosmer, D. W., & Lemeshow, S. (1989). Applied Logistic Regression. Wiley, New York.

McCullagh, P., & Nelder, J. A. (1989). Generalized Linear Models. Chapman and Hall, London.

Polis, G. A., Hurd, S. D., Jackson, C. T., & Piñero, F. S. (1998). Multifactor analysis of ecosystem patterns on islands in the Gulf of California. Ecological Monographs, 68, 490-502.