Palmer Penguins Data Analysis Series (Part 4): Model Diagnostics and Interpretation

Ensuring our models meet assumptions and understanding what they really tell us

Your Name

2025-01-04

Table of contents

Figure 1: A penguin scientist with a magnifying glass, carefully examining model diagnostics and residual plots!

Photo: African penguins at Boulders Beach, South Africa. Licensed under CC BY 2.0 via Wikimedia Commons

Palmer Penguins Data Analysis Series

This is **Part 4** of a 5-part series exploring penguin morphometrics:

1. Part 1: EDA and Simple Regression
2. Part 2: Multiple Regression and Species Effects
3. Part 3: Advanced Models and Cross-Validation
4. **Part 4: Model Diagnostics and Interpretation** (This post)
5. Part 5: Random Forest vs Linear Models

1 Introduction

Welcome to the fourth chapter of our Palmer penguins journey! In Part 3, we validated our models through rigorous cross-validation and confirmed that our species-aware linear model offers excellent predictive performance. But excellent performance doesn't automatically mean our model is appropriate or that our assumptions are satisfied.

Today, we dive into the critical but often overlooked world of model diagnostics. Think of this as taking your high-performing car to a mechanic - it might run well, but are there underlying issues that could cause problems? In statistical modeling, diagnostic procedures help us understand whether our model is not just performing well, but performing well for the right reasons.

In this post, we'll explore:

- Comprehensive residual analysis and assumption checking
- Influence diagnostics to identify problematic observations
- Biological interpretation of model coefficients
- Prediction intervals and uncertainty quantification
- Best practices for model reporting in ecological research

By the end of this post, you'll have confidence that your model is not just accurate, but statistically sound and biologically meaningful.

2 Setup and Model Preparation

Let's reconstruct our best-performing model and prepare for diagnostic analysis:

```r
library(palmerpenguins)
library(tidyverse)
library(broom)
# Conditional loading of optional packages
optional_diagnostic_packages <- c("car", "performance", "see", "lmtest")
for (pkg in optional_diagnostic_packages) {
  if (requireNamespace(pkg, quietly = TRUE)) {
    library(pkg, character.only = TRUE)
  } else {
    cat("  Package '", pkg, "' not available. Install with: install.packages('", pkg, "')\n")
  }
}
library(knitr)
library(patchwork)

# Set theme and colors
theme_set(theme_minimal(base_size = 12))
penguin_colors <- c("Adelie" = "#FF6B6B", "Chinstrap" = "#4ECDC4", "Gentoo" = "#45B7D1")

# Load and prepare data
data(penguins)
penguins_clean <- penguins %>% drop_na()

# Recreate our best model from previous parts
best_model <- lm(body_mass_g ~ bill_length_mm + bill_depth_mm +
                 flipper_length_mm + species, data = penguins_clean)

# Add fitted values and residuals to our dataset
penguins_diagnostics <- penguins_clean %>%
  mutate(
    fitted_values = fitted(best_model),
    residuals = residuals(best_model),
    standardized_residuals = rstandard(best_model),
    studentized_residuals = rstudent(best_model),
    leverage = hatvalues(best_model),
    cooks_distance = cooks.distance(best_model)
  )

cat("  Model Diagnostic Setup:\n")
```

```
  Model Diagnostic Setup:
```

```
cat("==========================\n")
```

```
==========================
```

```
cat(sprintf("Model: body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm + specie
```

```
Model: body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm + species
```

```
cat(sprintf("Observations: %d\n", nrow(penguins_clean)))
```

```
Observations: 333
```

```
cat(sprintf("Parameters: %d\n", length(coef(best_model))))
```

```
Parameters: 6
```

```
cat(sprintf("R-squared: %.3f\n", summary(best_model)$r.squared))
```

```
R-squared: 0.849
```

3 Classical Regression Assumptions

Linear regression relies on several key assumptions. Let's check each systematically:

3.1 1. Linearity

The relationship between predictors and response should be linear:

```
# Check linearity using partial residual plots
par(mfrow = c(2, 2))
avPlots(best_model, main = "Added-Variable Plots for Linearity")
```