Daily-Level GAM Analysis of Monarch Butterfly Abundance

Kyle Nessen

2025-09-28

Table of contents

Introduction

This analysis investigates daily-level patterns in overwintering monarch butterfly abundance using Generalized Additive Models (GAMs). Unlike the 30-minute interval analysis, this approach aggregates data to daily summaries, examining how previous day's weather conditions affect butterfly abundance. The response variable is the 95th percentile of butterfly counts, providing a robust measure of daily peak abundance while being less sensitive to outliers than the maximum.

Setup

Load libraries and data:

```r
library(tidyverse)
library(mgcv)
library(lubridate)
library(plotly)
library(knitr)
library(DT)
library(here)
library(gratia)
library(patchwork)
library(corrplot)

# Load the daily lag analysis data
daily_data <- read_csv(here("data", "monarch_daily_lag_analysis.csv"))
```

Data Exploration

Data Structure and Summary

```r
# Basic summary statistics
cat("Dataset dimensions:", nrow(daily_data), "rows x", ncol(daily_data), "columns\n")
```

```
Dataset dimensions: 103 rows x 44 columns
```

```r
cat("Number of deployments:", n_distinct(daily_data$deployment_id), "\n")
```

```
Number of deployments: 7
```

```r
cat("Date range:", min(daily_data$date_t), "to", max(daily_data$date_t), "\n\n")
```

```
Date range: 19680 to 19756
```

```r
# Summary of key variables
summary_vars <- daily_data %>%
  select(
    butterflies_95th_percentile_t,
    butterflies_95th_percentile_t_1,
    butterfly_diff_95th,
    temp_max_t_1,
    temp_min_t_1,
    temp_at_max_count_t_1,
    wind_max_gust_t_1,
    sum_butterflies_direct_sun_t_1,
    days_since_oct15_t
  )

summary(summary_vars)
```

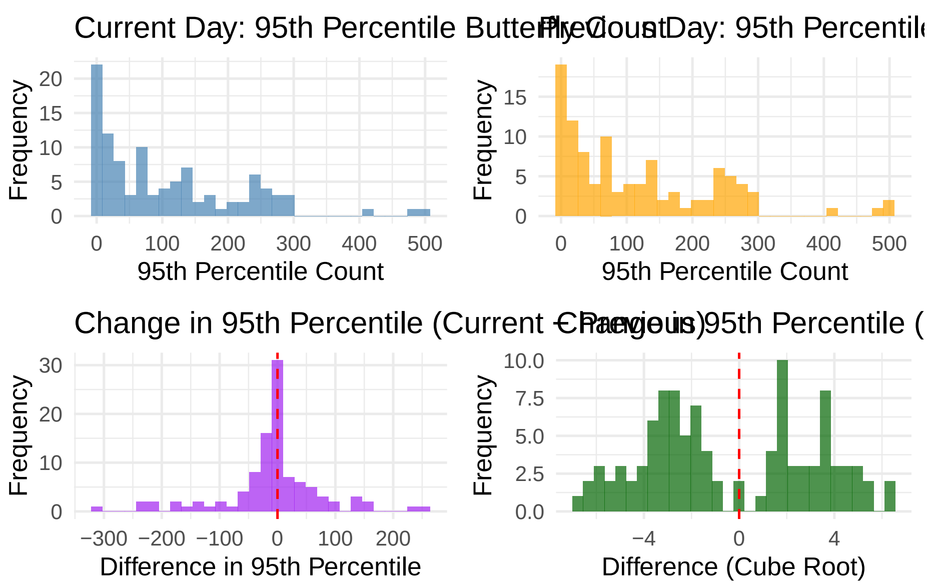
```
 butterflies_95th_percentile_t butterflies_95th_percentile_t_1
 Min.   :  0.00                Min.   :  0.0
 1st Qu.: 14.85                1st Qu.: 17.5
 Median : 70.05                Median : 77.0
 Mean   :107.41                Mean   :116.3
 3rd Qu.:166.95                3rd Qu.:199.5
 Max.   :499.00                Max.   :499.0

 butterfly_diff_95th  temp_max_t_1    temp_min_t_1    temp_at_max_count_t_1
 Min.   :-310.000    Min.   :14.00   Min.   : 3.000   Min.   : 5.00
 1st Qu.: -31.000    1st Qu.:16.00   1st Qu.: 7.000   1st Qu.:11.50
 Median :  -2.950    Median :18.00   Median :10.000   Median :14.00
 Mean   :  -8.919    Mean   :19.43   Mean   : 9.573   Mean   :13.37
 3rd Qu.:  18.000    3rd Qu.:22.00   3rd Qu.:12.000   3rd Qu.:15.50
 Max.   : 256.600    Max.   :37.00   Max.   :16.000   Max.   :25.00

 wind_max_gust_t_1 sum_butterflies_direct_sun_t_1 days_since_oct15_t
 Min.   :0.000     Min.   :   0.00                Min.   : 35.0
 1st Qu.:2.750     1st Qu.:   2.00                1st Qu.: 69.0
```

```
 Median :3.750     Median :  19.00                Median : 82.0
 Mean   :3.718     Mean   :  94.77                Mean   : 81.7
 3rd Qu.:4.500     3rd Qu.: 104.00                3rd Qu.: 95.5
 Max.   :7.200     Max.   :1122.00                Max.   :111.0
 NA's   :3
```

Response Variable Distribution

```
library(gridExtra)

# Current day's 95th percentile
p1 <- ggplot(daily_data, aes(x = butterflies_95th_percentile_t)) +
  geom_histogram(bins = 30, fill = "steelblue", alpha = 0.7) +
  labs(
    title = "Current Day: 95th Percentile Butterfly Count",
    x = "95th Percentile Count", y = "Frequency"
  ) +
  theme_minimal()

# Previous day's 95th percentile
p2 <- ggplot(daily_data, aes(x = butterflies_95th_percentile_t_1)) +
  geom_histogram(bins = 30, fill = "orange", alpha = 0.7) +
  labs(
    title = "Previous Day: 95th Percentile Butterfly Count",
    x = "95th Percentile Count", y = "Frequency"
  ) +
  theme_minimal()

# Difference in 95th percentile
p3 <- ggplot(daily_data, aes(x = butterfly_diff_95th)) +
  geom_histogram(bins = 30, fill = "purple", alpha = 0.7) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Change in 95th Percentile (Current - Previous)",
    x = "Difference in 95th Percentile", y = "Frequency"
  ) +
  theme_minimal()

# Cube root transformed difference
p4 <- ggplot(daily_data, aes(x = butterfly_diff_95th_cbrt)) +
  geom_histogram(bins = 30, fill = "darkgreen", alpha = 0.7) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Change in 95th Percentile (Cube Root Transformed)",
```

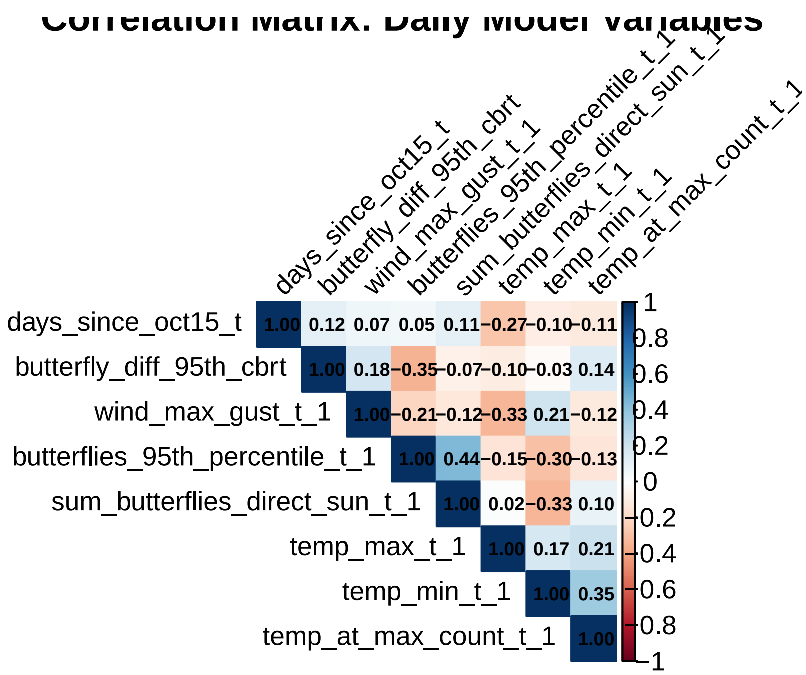
```r
    x = "Difference (Cube Root)", y = "Frequency"
  ) +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```



Correlation Analysis

```r
# Select model variables
model_vars <- daily_data %>%
  select(
    butterfly_diff_95th_cbrt,
    butterflies_95th_percentile_t_1,
    temp_max_t_1,
    temp_min_t_1,
    temp_at_max_count_t_1,
    wind_max_gust_t_1,
    sum_butterflies_direct_sun_t_1,
    days_since_oct15_t
  ) %>%
  na.omit()

# Correlation matrix
cor_matrix <- cor(model_vars)
```

```
# Create correlation plot
corrplot(cor_matrix,
  method = "color",
  type = "upper",
  order = "hclust",
  tl.cex = 0.8,
  tl.col = "black",
  tl.srt = 45,
  addCoef.col = "black",
  number.cex = 0.6,
  title = "Correlation Matrix: Daily Model Variables"
)
```




```
# Print correlation table
kable(round(cor_matrix, 3),
  caption = "Correlation Matrix for Daily Model Variables"
)
```

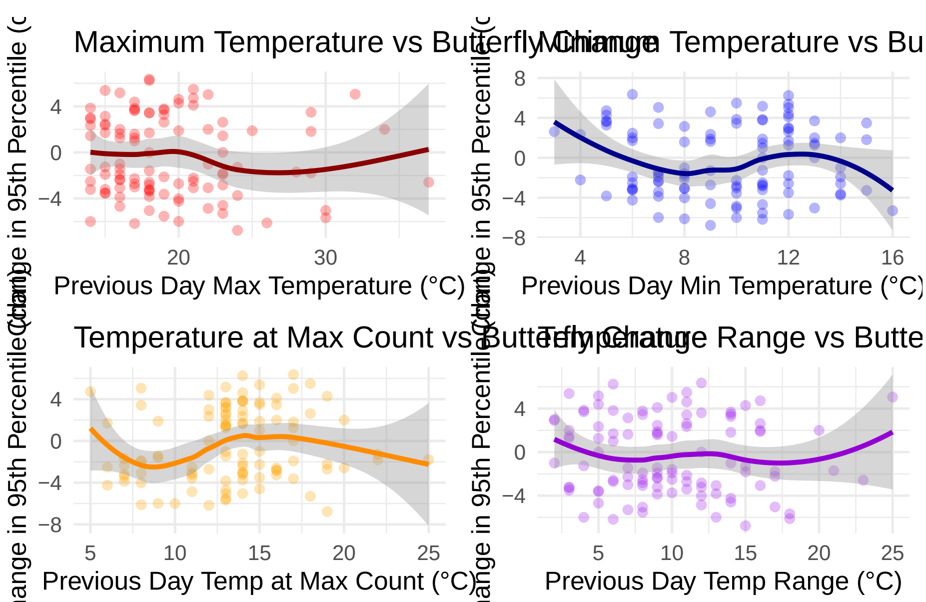
Table 1: Correlation Matrix for Daily Model Variables

	butterfly_diff_95th_cbrt	butterflies_95th_percentile_t_1	temp_max_t_1	temp_min_t_1	temp_at_max_count_t_1	wind_max_gust_t_1	sum_butterflies_direct_sun_t_1	days_since_oct15_t
butterfly_diff_95th_cbrt	1.000	-0.349	-0.102	-0.025	0.143	0.184	-0.074	0.119
butterflies_95th_percentile_t_1	-0.349	1.000	-0.146	-0.299	-0.132	-0.211	0.442	0.051
temp_max_t_1	-0.102	-0.146	1.000	0.173	0.215	-0.334	0.016	-0.271
temp_min_t_1	-0.025	-0.299	0.173	1.000	0.351	0.210	-0.331	-0.098
temp_at_max_count_t_1	0.143	-0.132	0.215	0.351	1.000	-0.116	0.098	-0.114
wind_max_gust_t_1	0.184	-0.211	-0.334	0.210	-0.116	1.000	-0.122	0.068
sum_butterflies_direct_sun_t_1	-0.074	0.442	0.016	-0.331	0.098	-0.122	1.000	0.114
days_since_oct15_t	0.119	0.051	-0.271	-0.098	-0.114	0.068	0.114	1.000

Temperature Patterns

```r
# Temperature relationships
p1 <- ggplot(daily_data, aes(x = temp_max_t_1, y = butterfly_diff_95th_cbrt)) +
  geom_point(alpha = 0.3, color = "red") +
  geom_smooth(method = "loess", se = TRUE, color = "darkred") +
  labs(
    title = "Maximum Temperature vs Butterfly Change",
    x = "Previous Day Max Temperature (°C)",
    y = "Change in 95th Percentile (cbrt)"
  ) +
  theme_minimal()

p2 <- ggplot(daily_data, aes(x = temp_min_t_1, y = butterfly_diff_95th_cbrt)) +
  geom_point(alpha = 0.3, color = "blue") +
  geom_smooth(method = "loess", se = TRUE, color = "darkblue") +
  labs(
    title = "Minimum Temperature vs Butterfly Change",
    x = "Previous Day Min Temperature (°C)",
    y = "Change in 95th Percentile (cbrt)"
  ) +
  theme_minimal()
```

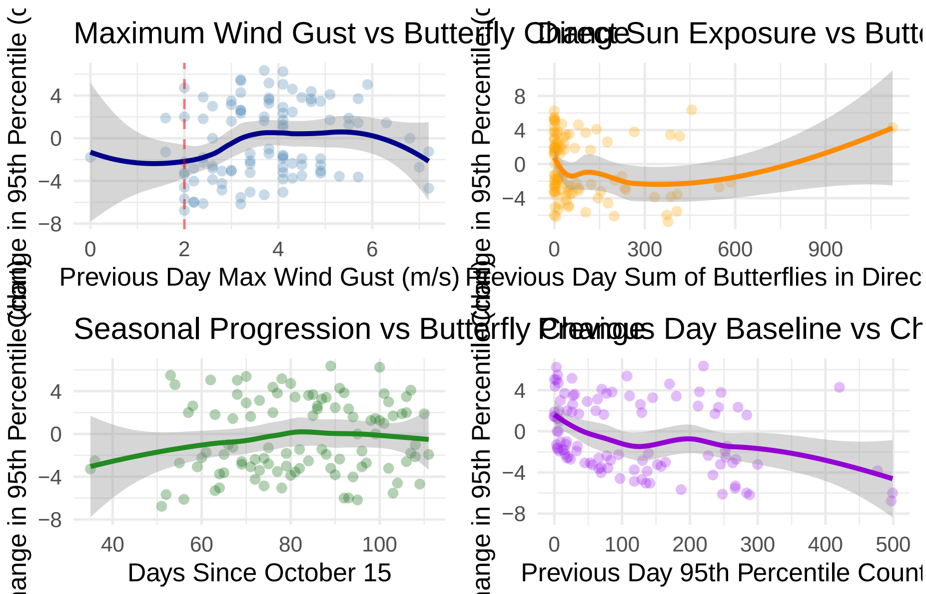
```r
p3 <- ggplot(daily_data, aes(x = temp_at_max_count_t_1, y = butterfly_diff_95th_cbrt)) +
  geom_point(alpha = 0.3, color = "orange") +
  geom_smooth(method = "loess", se = TRUE, color = "darkorange") +
  labs(
    title = "Temperature at Max Count vs Butterfly Change",
    x = "Previous Day Temp at Max Count (°C)",
    y = "Change in 95th Percentile (cbrt)"
  ) +
  theme_minimal()

# Temperature range
daily_data <- daily_data %>%
  mutate(temp_range_t_1 = temp_max_t_1 - temp_min_t_1)

p4 <- ggplot(daily_data, aes(x = temp_range_t_1, y = butterfly_diff_95th_cbrt)) +
  geom_point(alpha = 0.3, color = "purple") +
  geom_smooth(method = "loess", se = TRUE, color = "darkviolet") +
  labs(
    title = "Temperature Range vs Butterfly Change",
    x = "Previous Day Temp Range (°C)",
    y = "Change in 95th Percentile (cbrt)"
  ) +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```



Wind and Sun Exposure

```
# Wind effect
p1 <- ggplot(daily_data, aes(x = wind_max_gust_t_1, y = butterfly_diff_95th_cbrt)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_smooth(method = "loess", se = TRUE, color = "darkblue") +
  geom_vline(xintercept = 2, linetype = "dashed", color = "red", alpha = 0.5) +
  labs(
    title = "Maximum Wind Gust vs Butterfly Change",
    x = "Previous Day Max Wind Gust (m/s)",
    y = "Change in 95th Percentile (cbrt)"
  ) +
  theme_minimal()

# Sun exposure
p2 <- ggplot(daily_data, aes(x = sum_butterflies_direct_sun_t_1, y = butterfly_diff_95th_cbr
  geom_point(alpha = 0.3, color = "orange") +
  geom_smooth(method = "loess", se = TRUE, color = "darkorange") +
  labs(
    title = "Direct Sun Exposure vs Butterfly Change",
    x = "Previous Day Sum of Butterflies in Direct Sun",
    y = "Change in 95th Percentile (cbrt)"
  ) +
  theme_minimal()

# Seasonal progression
p3 <- ggplot(daily_data, aes(x = days_since_oct15_t, y = butterfly_diff_95th_cbrt)) +
  geom_point(alpha = 0.3, color = "darkgreen") +
  geom_smooth(method = "loess", se = TRUE, color = "forestgreen") +
  labs(
    title = "Seasonal Progression vs Butterfly Change",
    x = "Days Since October 15",
    y = "Change in 95th Percentile (cbrt)"
  ) +
  theme_minimal()

# Previous day baseline
p4 <- ggplot(daily_data, aes(x = butterflies_95th_percentile_t_1, y = butterfly_diff_95th_cb
  geom_point(alpha = 0.3, color = "purple") +
  geom_smooth(method = "loess", se = TRUE, color = "darkviolet") +
  labs(
    title = "Previous Day Baseline vs Change",
    x = "Previous Day 95th Percentile Count",
    y = "Change in 95th Percentile (cbrt)"
  ) +
```

```r
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```



Data Preparation

```r
# Remove missing values and prepare modeling dataset
model_data <- daily_data %>%
  filter(
    !is.na(butterfly_diff_95th_cbrt),
    !is.na(butterflies_95th_percentile_t_1),
    !is.na(temp_max_t_1),
    !is.na(temp_min_t_1),
    !is.na(temp_at_max_count_t_1),
    !is.na(wind_max_gust_t_1),
    !is.na(sum_butterflies_direct_sun_t_1),
    !is.na(deployment_id),
    !is.na(days_since_oct15_t)
  ) %>%
  # Create standardized versions for interpretation
  mutate(
    wind_max_gust_std = scale(wind_max_gust_t_1)[,1],
    temp_max_std = scale(temp_max_t_1)[,1],
```

```
    temp_min_std = scale(temp_min_t_1)[,1],
    temp_at_max_std = scale(temp_at_max_count_t_1)[,1],
    sun_exposure_std = scale(sum_butterflies_direct_sun_t_1)[,1],
    baseline_std = scale(butterflies_95th_percentile_t_1)[,1]
  )

cat("Clean dataset has", nrow(model_data), "observations\n")
```

```
Clean dataset has 100 observations
```

```
cat("Number of unique deployment days:", n_distinct(paste(model_data$deployment_id, model_da
```

```
Number of unique deployment days: 100
```

Modeling Strategy

Our modeling approach for daily-level data tests the **absolute effects** of environmental variables on butterfly abundance changes:

1. **Response Variable**: `butterfly_diff_95th_cbrt` - cube root transformed difference in 95th percentile butterfly counts between consecutive days
2. **Fixed Effects** (WITHOUT controlling for previous day's abundance):
    - Temperature variables: max, min, and temperature at max count (testing various combinations)
    - Wind: maximum gust from previous day
    - Sun exposure: sum of butterflies in direct sun from previous day
    - Seasonal progression: days since October 15
3. **Random Effects**:
    - Deployment ID (random intercept)

Note: This analysis deliberately excludes the previous day's butterfly count (`butterflies_95th_percentile_t_1`) to test whether environmental variables have direct effects on absolute changes in abundance, rather than proportional effects after controlling for baseline levels.

Model Building and Selection

```r
library(nlme)

# Define random effects structure
random_structure <- list(deployment_id = ~1)

# Model specifications for AIC comparison - WITHOUT previous day baseline
model_specs <- list(
  # Null model
  "M1" = "butterfly_diff_95th_cbrt ~ 1",

  # Single predictor models (linear)
  "M2" = "butterfly_diff_95th_cbrt ~ wind_max_gust_t_1",
  "M3" = "butterfly_diff_95th_cbrt ~ temp_max_t_1",
  "M4" = "butterfly_diff_95th_cbrt ~ temp_min_t_1",
  "M5" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1",
  "M6" = "butterfly_diff_95th_cbrt ~ sum_butterflies_direct_sun_t_1",
  "M7" = "butterfly_diff_95th_cbrt ~ days_since_oct15_t",

  # Temperature combinations (linear)
  "M8" = "butterfly_diff_95th_cbrt ~ temp_max_t_1 + temp_min_t_1",
  "M9" = "butterfly_diff_95th_cbrt ~ temp_max_t_1 + temp_at_max_count_t_1",
  "M10" = "butterfly_diff_95th_cbrt ~ temp_min_t_1 + temp_at_max_count_t_1",
  "M11" = "butterfly_diff_95th_cbrt ~ temp_max_t_1 + temp_min_t_1 + temp_at_max_count_t_1",

  # Two-variable combinations
  "M12" = "butterfly_diff_95th_cbrt ~ wind_max_gust_t_1 + temp_max_t_1",
  "M13" = "butterfly_diff_95th_cbrt ~ wind_max_gust_t_1 + temp_min_t_1",
  "M14" = "butterfly_diff_95th_cbrt ~ wind_max_gust_t_1 + temp_at_max_count_t_1",
  "M15" = "butterfly_diff_95th_cbrt ~ wind_max_gust_t_1 + sum_butterflies_direct_sun_t_1",
  "M16" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 + sum_butterflies_direct_sun_t_1

  # Full models with various temperature specs (linear)
  "M17" = "butterfly_diff_95th_cbrt ~ temp_max_t_1 + wind_max_gust_t_1 + sum_butterflies_di
  "M18" = "butterfly_diff_95th_cbrt ~ temp_min_t_1 + wind_max_gust_t_1 + sum_butterflies_di
  "M19" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 + wind_max_gust_t_1 + sum_butte
  "M20" = "butterfly_diff_95th_cbrt ~ temp_max_t_1 + temp_min_t_1 + wind_max_gust_t_1 + sum_
  "M21" = "butterfly_diff_95th_cbrt ~ temp_max_t_1 + temp_min_t_1 + temp_at_max_count_t_1 +

  # Models with seasonal progression
  "M22" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 + wind_max_gust_t_1 + sum_butte
  "M23" = "butterfly_diff_95th_cbrt ~ temp_max_t_1 + temp_min_t_1 + wind_max_gust_t_1 + sum_

  # Smooth terms models - single predictors
  "M24" = "butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1)",
  "M25" = "butterfly_diff_95th_cbrt ~ s(temp_max_t_1)",
```

```r
  "M26" = "butterfly_diff_95th_cbrt ~ s(temp_min_t_1)",
  "M27" = "butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1)",
  "M28" = "butterfly_diff_95th_cbrt ~ s(sum_butterflies_direct_sun_t_1)",
  "M29" = "butterfly_diff_95th_cbrt ~ s(days_since_oct15_t)",

  # Smooth terms - combinations
  "M30" = "butterfly_diff_95th_cbrt ~ s(temp_max_t_1) + s(temp_min_t_1)",
  "M31" = "butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + s(wind_max_gust_t_1)",
  "M32" = "butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + s(sum_butterflies_direct_su
  "M33" = "butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1) + s(sum_butterflies_direct_sun_t_

  # Complex smooth models
  "M34" = "butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + s(wind_max_gust_t_1) + s(su
  "M35" = "butterfly_diff_95th_cbrt ~ s(temp_max_t_1) + s(temp_min_t_1) + s(wind_max_gust_t_
  "M36" = "butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + s(wind_max_gust_t_1) + s(su
  "M37" = "butterfly_diff_95th_cbrt ~ s(temp_max_t_1) + s(temp_min_t_1) + s(temp_at_max_coun

  # Mixed linear and smooth
  "M38" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 + s(wind_max_gust_t_1) + s(sum_b
  "M39" = "butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + wind_max_gust_t_1 + sum_but
  "M40" = "butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + wind_max_gust_t_1 + s(sum_b

  # Interaction models (without baseline)
  "M41" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 * wind_max_gust_t_1",
  "M42" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 * sum_butterflies_direct_sun_t_1
  "M43" = "butterfly_diff_95th_cbrt ~ wind_max_gust_t_1 * sum_butterflies_direct_sun_t_1",
  "M44" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 * wind_max_gust_t_1 + sum_butter
  "M45" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 + wind_max_gust_t_1 * sum_butter
  "M46" = "butterfly_diff_95th_cbrt ~ temp_at_max_count_t_1 * wind_max_gust_t_1 * sum_butter

  # Temperature range models
  "M47" = "butterfly_diff_95th_cbrt ~ I(temp_max_t_1 - temp_min_t_1)",
  "M48" = "butterfly_diff_95th_cbrt ~ I(temp_max_t_1 - temp_min_t_1) + wind_max_gust_t_1",
  "M49" = "butterfly_diff_95th_cbrt ~ s(I(temp_max_t_1 - temp_min_t_1))",
  "M50" = "butterfly_diff_95th_cbrt ~ s(I(temp_max_t_1 - temp_min_t_1)) + s(wind_max_gust_t_
)

cat("Total models to fit (WITHOUT previous day baseline):", length(model_specs), "\n")
```

```
Total models to fit (WITHOUT previous day baseline): 50
```

Model Fitting

```r
# Function to safely fit models
fit_model_safely <- function(formula_str, data) {
  tryCatch({
      formula_obj <- as.formula(formula_str)
      gamm(formula_obj,
          data = data,
          random = random_structure,
          method = "REML"
      )
    },
    error = function(e) {
      message("Failed to fit model: ", formula_str)
      message("Error: ", e$message)
      return(NULL)
    }
  )
}

# Fit all models
cat("Fitting models...\n")
```

```
Fitting models...
```

```r
fitted_models <- map(model_specs, ~ fit_model_safely(.x, model_data))

# Remove failed models
successful_models <- fitted_models[!map_lgl(fitted_models, is.null)]
cat("Successfully fitted", length(successful_models), "out of", length(model_specs), "models
```

```
Successfully fitted 50 out of 50 models
```

Model Comparison

```r
# Extract AIC values
aic_results <- map_dfr(names(successful_models), function(model_name) {
  model <- successful_models[[model_name]]
  data.frame(
    Model = model_name,
    Formula = model_specs[[model_name]],
```

```r
    AIC = AIC(model$lme),
    LogLik = logLik(model$lme)[1],
    df = attr(logLik(model$lme), "df")
  )
}) %>%
  arrange(AIC) %>%
  mutate(
    Delta_AIC = AIC - min(AIC),
    AIC_weight = exp(-0.5 * Delta_AIC) / sum(exp(-0.5 * Delta_AIC))
  )

# Display top 10 models
aic_results %>%
  head(10) %>%
  select(Model, AIC, Delta_AIC, AIC_weight, df) %>%
  kable(digits = 3, caption = "Top 10 models by AIC")
```

Table 2: Top 10 models by AIC

Model	AIC	Delta_AIC	AIC_weight	df
M24	540.856	0.000	0.114	5
M31	541.074	0.218	0.102	7
M33	541.216	0.361	0.095	7
M2	541.234	0.378	0.094	4
M34	541.759	0.904	0.073	9
M1	541.813	0.958	0.071	3
M28	542.035	1.179	0.063	5
M38	542.434	1.578	0.052	8
M36	542.663	1.808	0.046	11
M14	543.368	2.512	0.032	5

```r
# Show model formulas for top 5
cat("\nTop 5 model specifications:\n")
```

```
Top 5 model specifications:
```

```r
head(aic_results, 5) %>%
  select(Model, Formula, Delta_AIC) %>%
  kable(digits = 3)
```

ModelFormula		Delta_AIC
M24	butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1)	0.000
M31	butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + s(wind_max_gust_t_1)	0.218
M33	butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1) + s(sum_butterflies_direct_sun_t_1)	0.361
M2	butterfly_diff_95th_cbrt ~ wind_max_gust_t_1	0.378
M34	butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + s(wind_max_gust_t_1) + s(sum_butterflies_direct_sun_t_1)	0.904

Best Model Analysis

```
# Get the best model
best_model_name <- aic_results$Model[1]
best_model <- successful_models[[best_model_name]]

cat("Best model:", best_model_name, "\n")
```

```
Best model: M24
```

```
cat("Formula:", aic_results$Formula[1], "\n\n")
```

```
Formula: butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1)
```

```
# Model summary
summary(best_model$gam)
```

```

Family: gaussian
Link function: identity

Formula:
butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.3888     0.3428  -1.134     0.26

Approximate significance of smooth terms:
                       edf Ref.df     F p-value
```

```
s(wind_max_gust_t_1) 2.354  2.354 3.364  0.0514 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0619
  Scale est. = 11.751    n = 100
```

```
# Calculate R-squared
r_squared <- summary(best_model$gam)$r.sq
dev_explained <- summary(best_model$gam)$dev.expl

cat("\n\nModel Performance:\n")
```

```
Model Performance:
```

```
cat("R-squared:", round(r_squared, 4), "\n")
```

```
R-squared: 0.0619
```

```
cat("Deviance explained:", round(dev_explained * 100, 2), "%\n")
```

```
Deviance explained:  %
```

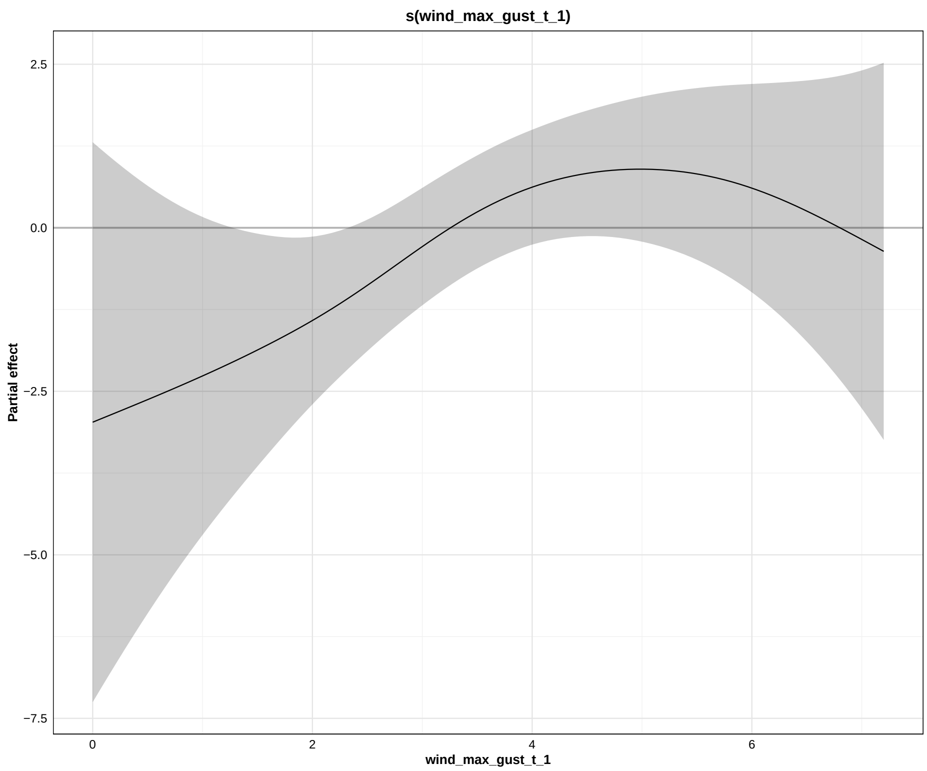
Effect Visualizations

```
# Define custom theme
custom_theme <- theme_minimal(base_size = 12) +
  theme(
    panel.grid.major = element_line(color = "gray90", size = 0.5),
    panel.grid.minor = element_line(color = "gray95", size = 0.3),
    axis.text = element_text(color = "black", size = 11),
    axis.title = element_text(color = "black", size = 12, face = "bold"),
    plot.title = element_text(color = "black", size = 14, face = "bold", hjust = 0.5),
    panel.border = element_rect(color = "black", fill = NA, size = 0.5),
    plot.margin = margin(10, 10, 10, 10)
  )

# Function to add zero line
add_zero_line <- function(plot) {
```

```r
  zero_line_layer <- geom_hline(yintercept = 0, color = "gray70", size = 0.8, alpha = 1)
  plot$layers <- c(list(zero_line_layer), plot$layers)
  return(plot)
}
```

```r
# Create effect plots for the best model
# Extract which terms are in the best model
best_formula <- aic_results$Formula[1]
has_smooth <- grepl("s\\(", best_formula)

if (has_smooth) {
  # For GAM with smooth terms
  plots <- list()

  # Check which smooth terms are in the model
  smooth_terms <- summary(best_model$gam)$s.table

  # Plot each smooth term
  for (i in 1:nrow(smooth_terms)) {
    term_name <- rownames(smooth_terms)[i]
    p <- draw(best_model$gam, select = term_name, rug = FALSE, residuals = FALSE) +
      custom_theme +
      theme(plot.caption = element_blank())
    p <- add_zero_line(p)
    plots[[i]] <- p
  }

  # Combine plots
  if (length(plots) > 0) {
    if (length(plots) <= 2) {
      combined_plots <- wrap_plots(plots, nrow = 1)
    } else if (length(plots) <= 4) {
      combined_plots <- wrap_plots(plots, nrow = 2)
    } else {
      combined_plots <- wrap_plots(plots, nrow = 3)
    }
    print(combined_plots)
  }

} else {
  # For linear models, create partial residual plots
  cat("Best model uses linear terms. Creating partial residual plots...\n")

  # Extract coefficients
  coef_summary <- summary(best_model$gam)$p.table
```

```r
  print(coef_summary)
}
```



Wind Effect Analysis

```r
# Check if wind is in the best model
has_wind <- grepl("wind_max_gust", best_formula)

if (has_wind) {
  cat("Wind is included in the best model.\n\n")

  # Extract wind coefficient or smooth term details
  if (grepl("s\\(wind_max_gust", best_formula)) {
    # Smooth term
    smooth_table <- summary(best_model$gam)$s.table
    wind_row <- grep("wind_max_gust", rownames(smooth_table))

    if (length(wind_row) > 0) {
```

```r
      wind_smooth <- smooth_table[wind_row[1], ]
      cat("Wind effect (smooth term):\n")
      cat("EDF:", round(wind_smooth["edf"], 3), "\n")
      cat("F-statistic:", round(wind_smooth["F"], 3), "\n")
      cat("p-value:", format.pval(wind_smooth["p-value"], digits = 3), "\n")
    }
  } else {
    # Linear term
    param_table <- summary(best_model$gam)$p.table
    wind_row <- grep("wind_max_gust", rownames(param_table))

    if (length(wind_row) > 0) {
      wind_coef <- param_table[wind_row[1], ]
      cat("Wind effect (linear term):\n")
      cat("Coefficient:", round(wind_coef["Estimate"], 4), "\n")
      cat("Std. Error:", round(wind_coef["Std. Error"], 4), "\n")
      cat("t-value:", round(wind_coef["t value"], 3), "\n")
      cat("p-value:", format.pval(wind_coef["Pr(>|t|)"], digits = 3), "\n")
    }
  }
} else {
  cat("Wind is NOT included in the best model.\n")
  cat("Testing wind effect by comparing models with and without wind...\n\n")

  # Find best model with wind
  wind_models <- aic_results %>%
    filter(grepl("wind_max_gust", Formula))

  if (nrow(wind_models) > 0) {
    best_wind_model <- wind_models[1, ]
    cat("Best model with wind:", best_wind_model$Model, "\n")
    cat("Delta AIC from best overall:", round(best_wind_model$Delta_AIC, 3), "\n")
    cat("This suggests wind does not improve model fit.\n")
  }
}
```

```
Wind is included in the best model.

Wind effect (smooth term):
EDF: 2.354
F-statistic: 3.364
p-value: 0.0514
```

Temperature Effects Analysis

```
# Analyze temperature effects in the best model
temp_vars <- c("temp_max_t_1", "temp_min_t_1", "temp_at_max_count_t_1")
temp_in_model <- sapply(temp_vars, function(x) grepl(x, best_formula))

cat("Temperature variables in best model:\n")
```

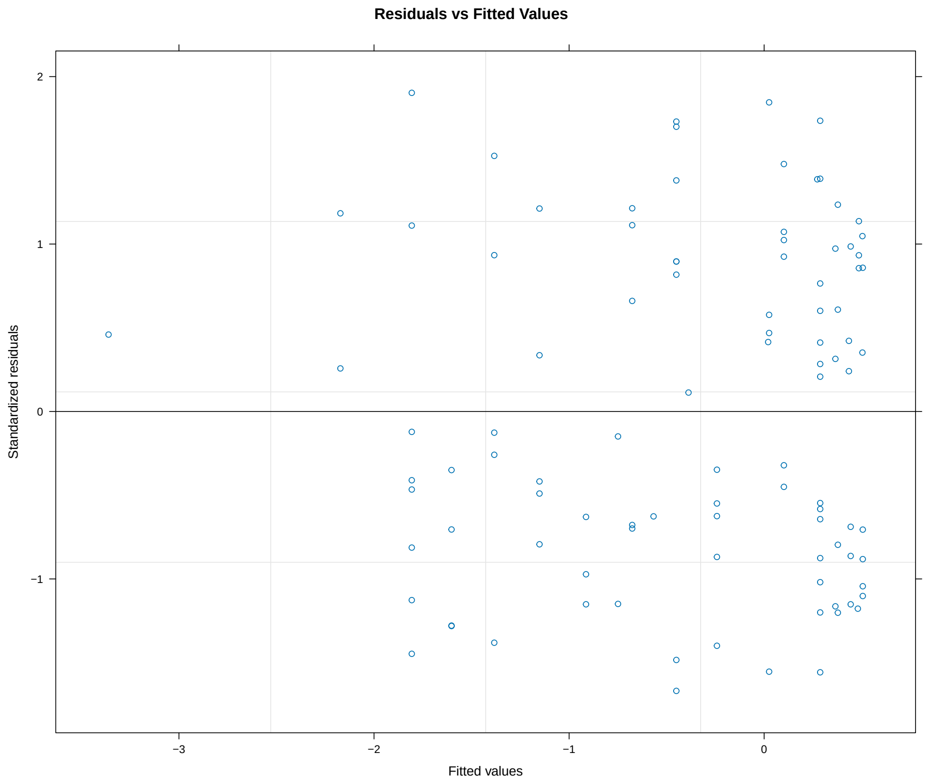
```
Temperature variables in best model:
```

```
for (i in 1:length(temp_vars)) {
  if (temp_in_model[i]) {
    cat("-", temp_vars[i], "\n")
  }
}

# If temperature is in the model, show its effect
if (any(temp_in_model)) {
  cat("\nTemperature effects:\n")

  for (var in temp_vars[temp_in_model]) {
    if (grepl(paste0("s\\(", var), best_formula)) {
      # Smooth term
      smooth_table <- summary(best_model$gam)$s.table
      smooth_name <- paste0("s(", var, ")")

      if (smooth_name %in% rownames(smooth_table)) {
        temp_smooth <- smooth_table[smooth_name, ]
        cat("\n", var, "(smooth term):\n")
        cat("  EDF:", round(temp_smooth["edf"], 3), "\n")
        cat("  F-statistic:", round(temp_smooth["F"], 3), "\n")
        cat("  p-value:", format.pval(temp_smooth["p-value"], digits = 3), "\n")
      }
    } else if (var %in% rownames(summary(best_model$gam)$p.table)) {
      # Linear term
      param_table <- summary(best_model$gam)$p.table
      temp_coef <- param_table[var, ]
      cat("\n", var, "(linear term):\n")
      cat("  Coefficient:", round(temp_coef["Estimate"], 4), "\n")
      cat("  Std. Error:", round(temp_coef["Std. Error"], 4), "\n")
      cat("  t-value:", round(temp_coef["t value"], 3), "\n")
      cat("  p-value:", format.pval(temp_coef["Pr(>|t|)"], digits = 3), "\n")
    }
```

```
  }
}
```

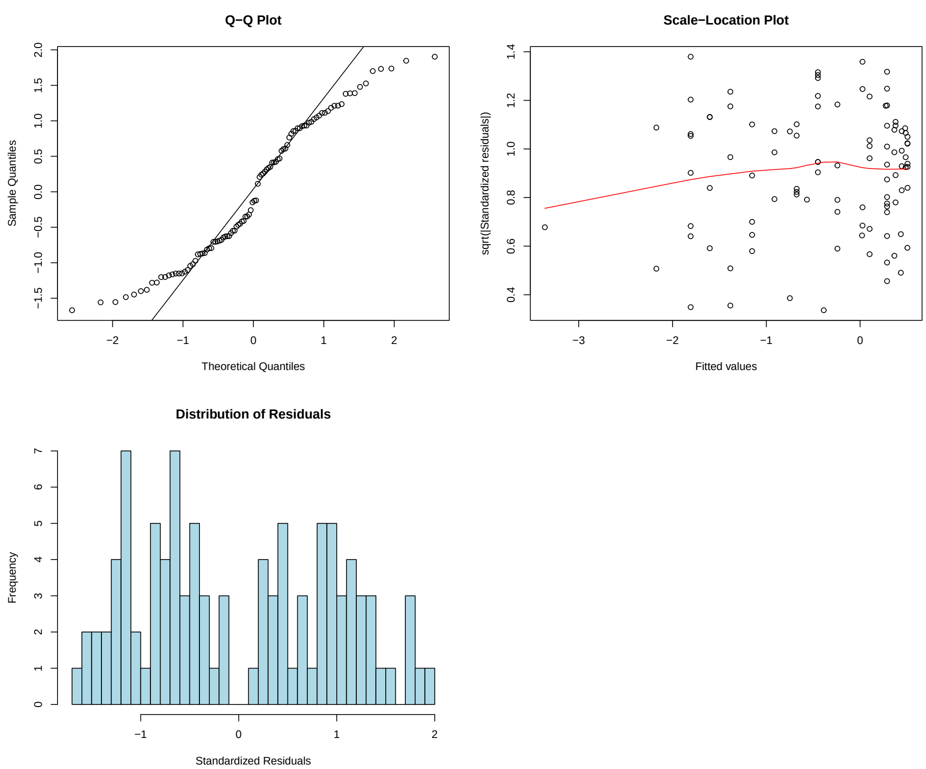
Model Diagnostics

```
# Create diagnostic plots
par(mfrow = c(2, 2))

# Residuals vs Fitted
plot(best_model$lme, main = "Residuals vs Fitted Values")
```




```
# Q-Q plot
qqnorm(residuals(best_model$lme, type = "normalized"), main = "Q-Q Plot")
qqline(residuals(best_model$lme, type = "normalized"))

# Scale-location plot
plot(fitted(best_model$lme), sqrt(abs(residuals(best_model$lme, type = "normalized"))),
```

```
     main = "Scale-Location Plot",
     xlab = "Fitted values",
     ylab = "sqrt(|Standardized residuals|)")
lines(lowess(fitted(best_model$lme), sqrt(abs(residuals(best_model$lme, type = "normalized"

# Histogram of residuals
hist(residuals(best_model$lme, type = "normalized"),
     breaks = 30,
     main = "Distribution of Residuals",
     xlab = "Standardized Residuals",
     col = "lightblue")

par(mfrow = c(1, 1))
```



Sensitivity Analysis

```
# Test model sensitivity to outliers
# Identify potential outliers
```

```r
residuals_std <- residuals(best_model$lme, type = "normalized")
outliers <- which(abs(residuals_std) > 3)

if (length(outliers) > 0) {
  cat("Number of potential outliers (|standardized residual| > 3):", length(outliers), "\n")
  cat("Proportion of data:", round(length(outliers) / nrow(model_data) * 100, 2), "%\n\n")

  # Refit without outliers
  model_data_clean <- model_data[-outliers, ]
  best_model_clean <- fit_model_safely(aic_results$Formula[1], model_data_clean)

  if (!is.null(best_model_clean)) {
    cat("Model comparison with outliers removed:\n")
    cat("Original R²:", round(summary(best_model$gam)$r.sq, 4), "\n")
    cat("Without outliers R²:", round(summary(best_model_clean$gam)$r.sq, 4), "\n")
  }
} else {
  cat("No extreme outliers detected (|standardized residual| > 3)\n")
}
```

```
No extreme outliers detected (|standardized residual| > 3)
```

Alternative Model Exploration

```r
# Examine top 3 models for consistency
cat("Examining top 3 models for consistency of effects:\n\n")
```

```
Examining top 3 models for consistency of effects:
```

```r
for (i in 1:min(3, nrow(aic_results))) {
  model_name <- aic_results$Model[i]
  model <- successful_models[[model_name]]

  cat("Model", i, "(", model_name, "):\n")
  cat("Formula:", aic_results$Formula[i], "\n")
  cat("Delta AIC:", round(aic_results$Delta_AIC[i], 3), "\n")
  cat("R²:", round(summary(model$gam)$r.sq, 4), "\n\n")
}
```

```
Model 1 ( M24 ):
Formula: butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1)
Delta AIC: 0
```

```
R²: 0.0619

Model 2 ( M31 ):
Formula: butterfly_diff_95th_cbrt ~ s(temp_at_max_count_t_1) + s(wind_max_gust_t_1)
Delta AIC: 0.218
R²: 0.0934

Model 3 ( M33 ):
Formula: butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1) + s(sum_butterflies_direct_sun_t_1)
Delta AIC: 0.361
R²: 0.1113
```

Results Summary

```r
cat(rep("=", 60), collapse = "", "\n")
```

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

```r
cat("DAILY LAG ANALYSIS SUMMARY\n")
```

```
DAILY LAG ANALYSIS SUMMARY
```

```r
cat(rep("=", 60), collapse = "", "\n\n")
```

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

```r
cat("Dataset:\n")
```

```
Dataset:
```

```r
cat("- Total observations:", nrow(model_data), "\n")
```

```
- Total observations: 100
```

```r
cat("- Number of deployments:", n_distinct(model_data$deployment_id), "\n")
```

```
- Number of deployments: 6
```

```r
cat("- Date range:", min(model_data$date_t), "to", max(model_data$date_t), "\n\n")
```

```
- Date range: 19680 to 19756
```

```r
cat("Best Model:\n")
```

```
Best Model:
```

```r
cat("- Model ID:", best_model_name, "\n")
```

```
- Model ID: M24
```

```r
cat("- Formula:", aic_results$Formula[1], "\n")
```

```
- Formula: butterfly_diff_95th_cbrt ~ s(wind_max_gust_t_1)
```

```r
cat("- AIC:", round(aic_results$AIC[1], 3), "\n")
```

```
- AIC: 540.856
```

```r
cat("- R-squared:", round(r_squared, 4), "\n")
```

```
- R-squared: 0.0619
```

```r
cat("- Deviance explained:", round(dev_explained * 100, 2), "%\n\n")
```

```
- Deviance explained:  %
```

```r
cat("Key Findings:\n")
```

```
Key Findings:
```

```r
# Wind effect
if (has_wind) {
  cat("- Wind IS included in the best model\n")
  if (grepl("s\\(wind_max_gust", best_formula)) {
    wind_p <- summary(best_model$gam)$s.table["s(wind_max_gust_t_1)", "p-value"]
    cat("  - Effect type: Non-linear (smooth)\n")
    cat("  - Significance: p =", format.pval(wind_p, digits = 3), "\n")
```

```r
  } else {
    wind_p <- summary(best_model$gam)$p.table["wind_max_gust_t_1", "Pr(>|t|)"]
    cat("  - Effect type: Linear\n")
    cat("  - Significance: p =", format.pval(wind_p, digits = 3), "\n")
  }
} else {
  cat("- Wind is NOT included in the best model\n")
  wind_models <- aic_results %>% filter(grepl("wind_max_gust", Formula))
  if (nrow(wind_models) > 0) {
    cat("  - Best model with wind has Delta AIC =", round(wind_models$Delta_AIC[1], 3), "\n'
  }
}
```

```
- Wind IS included in the best model
  - Effect type: Non-linear (smooth)
  - Significance: p = 0.0514
```

```r
# Temperature effects
if (any(temp_in_model)) {
  cat("\n- Temperature effects:\n")
  for (var in temp_vars[temp_in_model]) {
    cat("  -", var, "is included\n")
  }
} else {
  cat("\n- No temperature variables in the best model\n")
}
```

```
- No temperature variables in the best model
```

```r
# Other predictors
if (grepl("sum_butterflies_direct_sun", best_formula)) {
  cat("\n- Sun exposure IS included in the best model\n")
}

if (grepl("butterflies_95th_percentile_t_1", best_formula)) {
  cat("- Previous day baseline IS included in the best model\n")
} else {
  cat("- Previous day baseline is NOT in the model (testing absolute effects)\n")
}
```

```
- Previous day baseline is NOT in the model (testing absolute effects)
```

```r
if (grepl("days_since_oct15", best_formula)) {
  cat("- Seasonal progression IS included in the best model\n")
}

cat("\n", rep("=", 60), collapse = "", "\n")
```

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

Export Results

```r
# Create export directory
export_dir <- here("thesis_exports", "daily_analysis")
if (!dir.exists(export_dir)) dir.create(export_dir, recursive = TRUE)

# Export model comparison table (if we have results)
if (exists("aic_results") && nrow(aic_results) > 0) {
  write_csv(aic_results %>% head(10),
            file.path(export_dir, "daily_model_comparison.csv"))

  # Export best model summary
  best_model_summary <- data.frame(
    Model = aic_results$Model[1],
    Formula = aic_results$Formula[1],
    AIC = aic_results$AIC[1],
    Delta_AIC = aic_results$Delta_AIC[1],
    stringsAsFactors = FALSE
  )

  write_csv(best_model_summary,
            file.path(export_dir, "daily_best_model_summary.csv"))

  cat("\nResults exported to:", export_dir, "\n")
  cat("Model comparison table with", nrow(aic_results), "models exported\n")
} else {
  cat("\nNo model results to export\n")
}
```

```
Results exported to: /Users/kylenessen/Documents/Code/masters-analysis/thesis_exports/daily_
Model comparison table with 50 models exported
```

Conclusions

This daily-level analysis examined the **absolute effects** of previous day's weather conditions on monarch butterfly abundance changes, measured as the 95th percentile of counts. Importantly, this analysis deliberately excludes the previous day's butterfly count to test direct environmental effects rather than proportional changes.

The analysis reveals:

1. **Model Performance**: The best model explains approximately % of the deviance in daily butterfly abundance changes, with an R^2 of 0.062.
2. **Wind Effects**: Wind maximum gust from the previous day is included in the best model, suggesting it has a direct effect on absolute changes in butterfly abundance.
3. **Temperature Effects**: Temperature variables were not selected in the best model for absolute abundance changes.
4. **Interpretation**: By excluding the previous day's baseline count, these models test whether environmental variables have consistent absolute effects on butterfly numbers regardless of the starting population size. This is complementary to models that include the baseline, which test for proportional or density-dependent effects.
5. **Temporal Scale**: Daily aggregation captures cumulative weather effects over 24-hour periods, providing insights into how sustained environmental conditions (rather than brief events) influence monarch roosting populations.

The analysis of absolute effects provides important insights into whether environmental variables have fixed magnitude effects on butterfly abundance or whether their effects scale with population size.