Monarch Butterfly Abundance Analysis (Updated Approach)

Setup and Libraries

```r
# Load necessary libraries
library(tidyverse)
library(lubridate)
library(mgcv) # For GAMs
library(data.table) # For fast data prep
library(knitr) # For nice tables
```

Load Raw Data

```r
# Load raw data
counts <- readr::read_csv("../data/butterfly_abundance_index.csv", show_col_types = FALSE)
deployments <- readr::read_csv("../data/deployments.csv", show_col_types = FALSE)
temp <- readr::read_csv("../data/temperature_data_2023.csv", show_col_types = FALSE)
wind <- readr::read_csv("../data/wind_all.csv", show_col_types = FALSE)

glimpse(counts)
```

```
Rows: 11,885
Columns: 4
$ deployment_id          <chr> "SC1", "SC1", "SC1", "SC1", "SC1", "SC1", "SC1"~
$ image_filename         <chr> "SC1_20231117114001.JPG", "SC1_20231117114501.J~
$ total_butterflies      <dbl> 121, 156, 122, 134, 108, 114, 86, 90, 82, 89, 7~
$ butterflies_direct_sun <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

```r
glimpse(deployments)
```

```
Rows: 19
Columns: 23
$ camera_name                  <chr> "EM03", "JINX", "IRIS", "NOVA", "LYNX", "~
$ wind_meter_name              <chr> "JazzPlay", "Stardust", "FunStorm", "Blue~
$ Deployed_time                <chr> "2023/11/17 11:40:00", "2024/01/05 16:06:~
$ Recovered_time               <chr> "2023/12/15 11:00:00", "2024/01/31 20:30:~
$ notes                        <chr> "Location and height estimated after depl~
$ height_m                     <dbl> 7.0, 5.9, 6.1, 7.1, 6.2, NA, 5.6, 8.3, 6.~
$ horizontal_dist_to_cluster_m <dbl> 4.2, 7.5, NA, 12.6, 6.8, NA, 6.3, 5.7, NA~
$ view_direction               <dbl> 90, 320, 90, 230, 155, 30, 335, 320, 210,~
$ cluster_count                <dbl> NA, NA, NA, NA, NA, NA, 750, 0, 0, 300, N~
$ deployment_id                <chr> "SC1", "SC10", "SC11", "SC12", "SC2", "SC~
$ status                       <chr> "Complete", "Complete", "Complete", "Comp~
$ photo_interval_min           <dbl> 5, 10, 10, 10, 5, 10, 10, 10, 10, 10, 10,~
$ monarchs_present             <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,~
$ youtube_url                  <chr> "https://youtu.be/-zeGd_WTeWo", "https://~
$ latitude                     <dbl> 34.63186, 34.63126, 34.63172, 34.63162, 3~
$ longitude                    <dbl> -120.6182, -120.6181, -120.6185, -120.618~
$ label_status                 <chr> "Complete", "Complete", "No Monarchs", "C~
$ `Percent Complete`           <chr> "100%", "5%", "100%", "100%", "100%", "0%~
$ Observer                     <chr> "Skyler", "Emery", NA, "Vincent", "Vincen~
$ Effort                       <chr> "Medium", "Hard", NA, "Hard", "Medium", "~
$ label_notes                  <chr> NA, NA, NA, NA, NA, "Need to check if the~
$ label_youtube_url            <chr> "https://www.youtube.com/watch?v=-zeGd_WT~
$ view_id                      <dbl> 1, 2, 3, 4, 5, 6, 2, 7, 4, 8, 4, 9, 10, 1~
```

```r
glimpse(temp)
```

```
Rows: 56,066
Columns: 6
$ filename          <chr> "SC1_20231117114001.JPG", "SC1_20231117114501.JPG", ~
$ deployment_id     <chr> "SC1", "SC1", "SC1", "SC1", "SC1", "SC1", "SC1", "SC~
$ timestamp         <dbl> 2.023112e+13, 2.023112e+13, 2.023112e+13, 2.023112e+~
$ temperature       <dbl> 22, 21, 20, 20, 19, 19, 19, 19, 19, 18, 18, 18, 18, ~
$ confidence        <dbl> 0.7918647, 0.4684700, 0.6609230, 0.5185271, 0.515489~
$ extraction_status <chr> "success", "success", "success", "success", "success~
```

```r
glimpse(wind)
```

```
Rows: 681,013
Columns: 5
$ db_path        <chr> "data/wind/BlueLake.s3db", "data/wind/BlueLake.s3db", ~
$ wind_meter_name <chr> "BlueLake", "BlueLake", "BlueLake", "BlueLake", "BlueL~
$ time           <dttm> 2008-01-01 08:01:00, 2023-12-15 20:23:00, 2023-12-15 ~
$ speed          <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.4, 1.5, 1.0, 1.3, 1.3,~
$ gust           <dbl> 0.4, 0.4, 0.0, 0.0, 0.0, 0.0, 2.0, 2.2, 1.6, 2.4, 1.8,~
```

Data Prep

```r
# Filter and prepare deployments data
deployments_filtered <- deployments %>%
    filter(label_status == "Complete") %>%
    mutate(view_id = as.factor(view_id)) %>%
    select(deployment_id, view_id, wind_meter_name)

# Parse timestamps in counts data (YYYYMMDDHHMMSS in image_filename)
counts_with_datetime <- counts %>%
    mutate(
        datetime_str = stringr::str_extract(image_filename, "\\d{14}"),
        datetime     = lubridate::ymd_hms(datetime_str, tz = "UTC")
    ) %>%
    select(-datetime_str)

# Create master dataframe
master_df <- counts_with_datetime %>%
    left_join(deployments_filtered, by = "deployment_id") %>%
    left_join(temp, by = c("image_filename" = "filename"))

# Coerce critical time columns; keep wind in UTC as well
master_df <- master_df %>%
    mutate(
        view_id  = as.factor(view_id),
        datetime = as.POSIXct(datetime, tz = "UTC")
    )

# Wind table must have clear names: wind_meter_name, time, speed, gust
wind <- wind %>%
    mutate(
        time = as.POSIXct(time, tz = "UTC")
    )

cat("Master dataframe created with", nrow(master_df), "rows\n")
```

```
Master dataframe created with 11885 rows
```

Lag Builder Function

```
# This version uses data.table non-equi joins to aggregate over [t-Δ, t] intervals
# per (view_id, wind_meter_name). It replaces the slow R loop.

prepare_lag_data <- function(master_df, wind_df, lag_minutes) {
    cat("Preparing data for", lag_minutes, "minute lag...\n")

    # Ensure unique column names in wind_df
    wind_df <- wind_df %>% rename(wind_time = time)

    df_t <- master_df %>%
        filter(!is.na(datetime) & !is.na(view_id))

    df_t_lag <- df_t %>%
        rename(
            abundance_t_minus_1         = total_butterflies,
            datetime_t_minus_1          = datetime,
            butterflies_sun_t_minus_1   = butterflies_direct_sun,
            temperature_t_minus_1       = temperature
        ) %>%
        mutate(datetime_expected = datetime_t_minus_1 + minutes(lag_minutes)) %>%
        select(
            view_id, datetime_t_minus_1, datetime_expected,
            abundance_t_minus_1, butterflies_sun_t_minus_1, temperature_t_minus_1
        )

    final_df <- df_t %>%
        left_join(
            df_t_lag,
            by = c("view_id" = "view_id", "datetime" = "datetime_expected")
        ) %>%
```

```r
        filter(!is.na(abundance_t_minus_1)) %>%
        filter(!(total_butterflies == 0 & abundance_t_minus_1 == 0)) %>%
        mutate(
            time_delta_mins = as.numeric(difftime(datetime, datetime_t_minus_1, units = "mins
            butterfly_diff = total_butterflies - abundance_t_minus_1,
            butterfly_log_diff = log((total_butterflies + 0.1) / (abundance_t_minus_1 + 0.1)
        )

    cat("Valid pairs after filtering:", nrow(final_df), "\n")
    if (nrow(final_df) == 0) {
        return(final_df[0, ])
    }

    intervals <- final_df %>%
        select(view_id, wind_meter_name, datetime_t_minus_1, datetime) %>%
        distinct() %>%
        mutate(interval_id = dplyr::row_number())

    wind_dt <- as.data.table(wind_df)
    master_dt <- as.data.table(master_df)
    ints_dt <- as.data.table(intervals)

    setkey(wind_dt, wind_meter_name, wind_time)
    setkey(master_dt, view_id, datetime)

    # Now use wind_time in non-equi join (remove duplicate wind_time from join)
    wind_ag <- wind_dt[
        ints_dt,
        on = .(
            wind_meter_name,
            wind_time >= datetime_t_minus_1,
            wind_time <= datetime
        ),
        allow.cartesian = TRUE,
        nomatch = 0L
    ][, .(
        interval_id                 = interval_id[1L],
        mean_wind_speed             = mean(speed, na.rm = TRUE),
        max_wind_speed              = suppressWarnings(max(gust, na.rm = TRUE)),
        sd_wind_speed               = sd(speed, na.rm = TRUE),
        cumulative_wind             = sum(speed, na.rm = TRUE),
        gust_factor                 = mean(gust, na.rm = TRUE) / mean(speed, na.rm = TRUE),
        sustained_minutes_above_2mps = sum(speed > 2, na.rm = TRUE),
        gust_minutes_above_2mps      = sum(gust > 2, na.rm = TRUE)
    ), by = .(wind_meter_name, interval_id)]

    # Guard against all-NA => max = -Inf
    if (nrow(wind_ag)) {
        wind_ag[, max_wind_speed := ifelse(is.finite(max_wind_speed), max_wind_speed, NA_real
        wind_ag[, gust_factor := ifelse(is.finite(gust_factor), gust_factor, NA_real_)]
    }

    # TEMPERATURE & SUNLIGHT metrics over [start, end]
    master_ag <- master_dt[
        ints_dt,
        on = .(
            view_id,
            datetime >= datetime_t_minus_1,
            datetime <= datetime
        ),
        allow.cartesian = TRUE,
        nomatch = 0L
    ][,
        {
            props <- ifelse(total_butterflies > 0,
                butterflies_direct_sun / total_butterflies,
                NA_real_
            )
            sum_total <- sum(total_butterflies, na.rm = TRUE)
            sum_sun <- sum(butterflies_direct_sun, na.rm = TRUE)
            sunlight_prop <- if (sum_total > 0) sum_sun / sum_total else NA_real_

            pmax <- suppressWarnings(max(props, na.rm = TRUE))
            pmin <- suppressWarnings(min(props, na.rm = TRUE))
            psd <- sd(props, na.rm = TRUE)
            pmax <- ifelse(is.finite(pmax), pmax, NA_real_)
            pmin <- ifelse(is.finite(pmin), pmin, NA_real_)
            psd <- ifelse(is.finite(psd), psd, NA_real_)

            .(
                interval_id         = interval_id[1L],
                mean_temp           = mean(temperature, na.rm = TRUE),
                max_temp            = suppressWarnings(max(temperature, na.rm = TRUE)),
                min_temp            = suppressWarnings(min(temperature, na.rm = TRUE)),
                sd_temp             = sd(temperature, na.rm = TRUE),
                sunlight_proportion = sunlight_prop,
                max_prop_sunlight   = pmax,
                min_prop_sunlight   = pmin,
                sd_prop_sunlight    = psd
            )
        },
        by = .(view_id, interval_id)
    ]

    if (nrow(master_ag)) {
        master_ag[, `:=`(
            max_temp = ifelse(is.finite(max_temp), max_temp, NA_real_),
            min_temp = ifelse(is.finite(min_temp), min_temp, NA_real_)
        )]
    }

    # Merge metrics back to the matched pairs
    # First select only unique columns from each dataset before merging
    wind_ag_clean <- wind_ag[, .(
        interval_id, mean_wind_speed, max_wind_speed,
        sd_wind_speed, cumulative_wind, gust_factor,
        sustained_minutes_above_2mps, gust_minutes_above_2mps
    )]
    master_ag_clean <- master_ag[, .(
        interval_id, mean_temp, max_temp, min_temp,
        sd_temp, sunlight_proportion, max_prop_sunlight,
        min_prop_sunlight, sd_prop_sunlight
    )]
    metrics <- merge(wind_ag_clean, master_ag_clean, by = "interval_id", all = TRUE)

    final_out <- final_df %>%
        left_join(intervals, by = c("view_id", "wind_meter_name", "datetime_t_minus_1", "date
        left_join(metrics, by = "interval_id") %>%
        select(-interval_id) %>%
        filter(!is.na(mean_wind_speed) & !is.na(mean_temp)) %>%
        arrange(view_id, datetime) %>%
        group_by(view_id) %>%
        mutate(time_index = row_number()) %>%
        ungroup()

    # Add GAM and LOESS smoothed predictions
```

```
cat("Computing GAM and LOESS smoothed predictions...\n")

# GAM smoothed predictions
final_out <- final_out %>%
    group_by(view_id) %>%
    do({
        if (nrow(.) >= 10) {  # Need minimum observations for smoothing
            tryCatch({
                # GAM smoothing
                gam_mod <- gam(total_butterflies ~ s(as.numeric(datetime)),
                               data = ., method = "REML")
                gam_fitted <- predict(gam_mod, newdata = .)

                # LOESS smoothing
                if (nrow(.) >= 3) {
                    loess_fitted <- predict(loess(total_butterflies ~ as.numeric(datetime
                                                 data = ., span = 0.3), newdata = .)
                } else {
                    loess_fitted <- .$total_butterflies
                }

                data.frame(.,
                          gam_fitted_count = gam_fitted,
                          loess_fitted_count = loess_fitted)
            }, error = function(e) {
                cat("Warning: Could not fit smoothers for view_id", .$view_id[1], "\n")
                data.frame(.,
                          gam_fitted_count = .$total_butterflies,
                          loess_fitted_count = .$total_butterflies)
            })
        } else {
            # Too few observations - use raw counts
            data.frame(.,
                      gam_fitted_count = .$total_butterflies,
                      loess_fitted_count = .$total_butterflies)
        }
    }) %>%
    ungroup()

# Compute differences from smoothed predictions and add t-1 fitted values
final_out <- final_out %>%
    group_by(view_id) %>%
        arrange(datetime) %>%
        mutate(
            gam_fitted_count_t_minus_1 = lag(gam_fitted_count),
            loess_fitted_count_t_minus_1 = lag(loess_fitted_count),
            gam_pred_diff = gam_fitted_count - lag(gam_fitted_count),
            loess_pred_diff = loess_fitted_count - lag(loess_fitted_count),
            gam_log_diff = log((gam_fitted_count + 0.1) / (lag(gam_fitted_count) + 0.1)),
            loess_log_diff = log((loess_fitted_count + 0.1) / (lag(loess_fitted_count) + 0.1
        ) %>%
        ungroup()

    cat("Final dataset rows:", nrow(final_out), "\n")
    if (nrow(final_out) > 0) {
        cat(
            "Time delta range:",
            round(min(final_out$time_delta_mins, na.rm = TRUE), 1), "to",
            round(max(final_out$time_delta_mins, na.rm = TRUE), 1), "minutes\n"
        )
    }

    final_out
}
```

Generate Lag Datasets

```
cat("=== GENERATING LAG DATASETS ===\n")
```

```
=== GENERATING LAG DATASETS ===
```

```
data_30m <- prepare_lag_data(master_df, wind, lag_minutes = 30)
```

```
Preparing data for 30 minute lag...
Valid pairs after filtering: 6888
Computing GAM and LOESS smoothed predictions...
Warning: Could not fit smoothers for view_id 5

Warning: There were 7 warnings in `mutate()`.
The first warning was:
i In argument: `gam_log_diff = log((gam_fitted_count +
  0.1)/(lag(gam_fitted_count) + 0.1))`.
i In group 2: `view_id = 2`.
Caused by warning in `log()`:
! NaNs produced
i Run `dplyr::last_dplyr_warnings()` to see the 6 remaining warnings.

Final dataset rows: 5601
Time delta range: 30 to 30 minutes
```

```r
data_120m <- prepare_lag_data(master_df, wind, lag_minutes = 120)
```

```
Preparing data for 120 minute lag...
Valid pairs after filtering: 6912
Computing GAM and LOESS smoothed predictions...
Warning: Could not fit smoothers for view_id 5

Warning: There were 10 warnings in `mutate()`.
The first warning was:
i In argument: `gam_log_diff = log((gam_fitted_count +
  0.1)/(lag(gam_fitted_count) + 0.1))`.
i In group 2: `view_id = 2`.
Caused by warning in `log()`:
! NaNs produced
i Run `dplyr::last_dplyr_warnings()` to see the 9 remaining warnings.

Final dataset rows: 5650
Time delta range: 120 to 120 minutes
```

```r
data_240m <- prepare_lag_data(master_df, wind, lag_minutes = 240)
```

```
Preparing data for 240 minute lag...
Valid pairs after filtering: 6894
Computing GAM and LOESS smoothed predictions...
Warning: Could not fit smoothers for view_id 5

Warning: There were 9 warnings in `mutate()`.
The first warning was:
i In argument: `gam_log_diff = log((gam_fitted_count +
  0.1)/(lag(gam_fitted_count) + 0.1))`.
i In group 2: `view_id = 2`.
```

```
Caused by warning in `log()`:
! NaNs produced
i Run `dplyr::last_dplyr_warnings()` to see the 8 remaining warnings.
```

```
Final dataset rows: 5663
Time delta range: 240 to 240 minutes
```

```r
cat("\n=== DATASET SUMMARY ===\n")
```

```
=== DATASET SUMMARY ===
```

```r
cat("30-minute lag dataset:", nrow(data_30m), "observations\n")
```

```
30-minute lag dataset: 5601 observations
```

```r
cat("2-hour   lag dataset:", nrow(data_120m), "observations\n")
```

```
2-hour   lag dataset: 5650 observations
```

```r
cat("4-hour   lag dataset:", nrow(data_240m), "observations\n")
```

```
4-hour   lag dataset: 5663 observations
```

Exploratory Analysis

```r
# Load additional libraries for visualization and correlation analysis
library(corrplot)
```

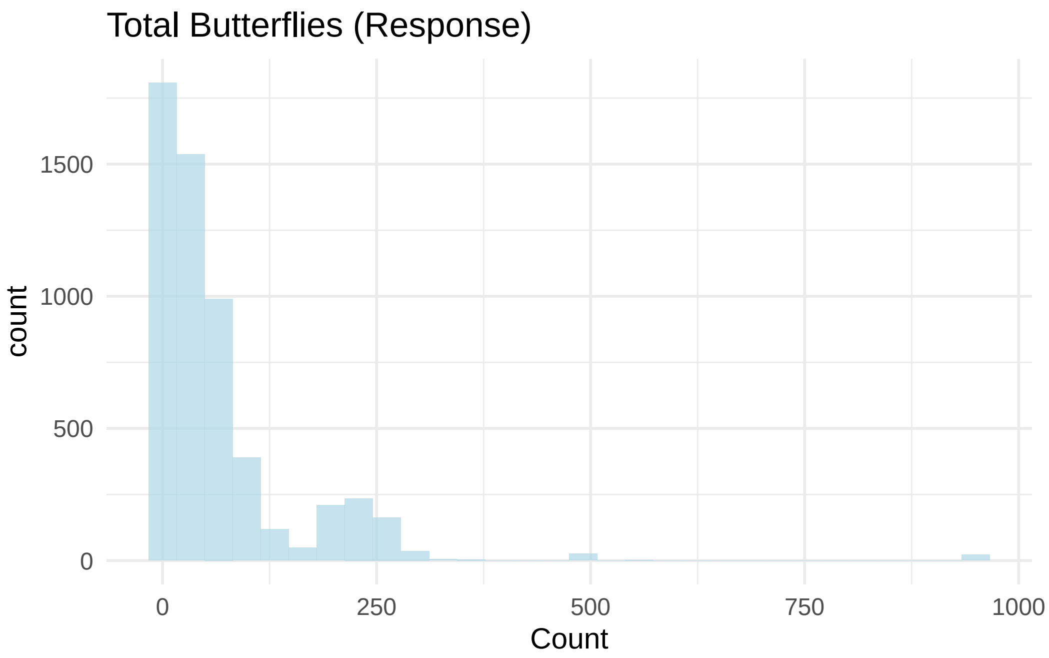
```
corrplot 0.95 loaded
```

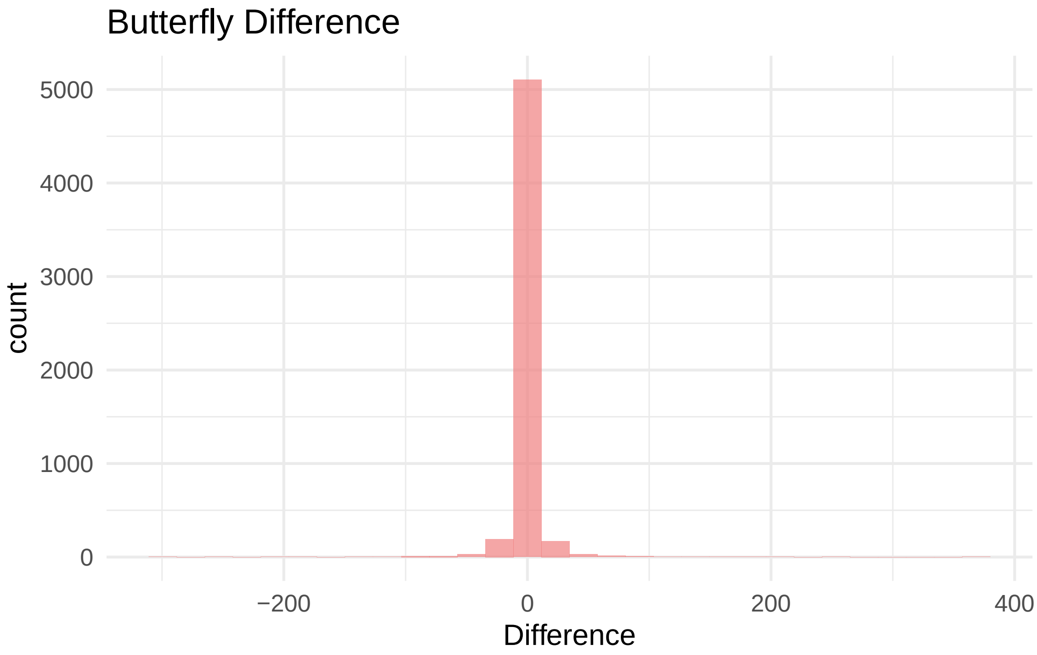
```
library(GGally)
library(patchwork)

# Function to create histograms for a dataset
create_histograms <- function(data, dataset_name) {
    cat(paste0("\n=== HISTOGRAMS FOR: ", dataset_name, " ===\n"))

    # Response variables
    p1 <- ggplot(data, aes(x = total_butterflies)) +
        geom_histogram(bins = 30, fill = "lightblue", alpha = 0.7) +
        labs(title = "Total Butterflies (Response)", x = "Count") +
        theme_minimal()

    p2 <- ggplot(data, aes(x = butterfly_diff)) +
        geom_histogram(bins = 30, fill = "lightcoral", alpha = 0.7) +
        labs(title = "Butterfly Difference", x = "Difference") +
        theme_minimal()

    p3 <- ggplot(data, aes(x = butterfly_log_diff)) +
        geom_histogram(bins = 30, fill = "lightgreen", alpha = 0.7) +
        labs(title = "Log Butterfly Difference", x = "Log Difference") +
        theme_minimal()

    # GAM fitted response variables
    p3a <- ggplot(data, aes(x = gam_fitted_count)) +
        geom_histogram(bins = 30, fill = "cyan", alpha = 0.7) +
        labs(title = "GAM Fitted Count", x = "Count") +
        theme_minimal()

    p3b <- ggplot(data, aes(x = gam_pred_diff)) +
        geom_histogram(bins = 30, fill = "darkgreen", alpha = 0.7) +
        labs(title = "GAM Predicted Difference", x = "Difference") +
        theme_minimal()

    p3c <- ggplot(data, aes(x = loess_fitted_count)) +
        geom_histogram(bins = 30, fill = "lightpink", alpha = 0.7) +
        labs(title = "LOESS Fitted Count", x = "Count") +
        theme_minimal()

    p3d <- ggplot(data, aes(x = loess_pred_diff)) +
        geom_histogram(bins = 30, fill = "darkmagenta", alpha = 0.7) +
        labs(title = "LOESS Predicted Difference", x = "Difference") +
        theme_minimal()

    # Lagged predictors
    p4 <- ggplot(data, aes(x = abundance_t_minus_1)) +
        geom_histogram(bins = 30, fill = "orange", alpha = 0.7) +
        labs(title = "Abundance (t-1)", x = "Count") +
        theme_minimal()

    p4a <- ggplot(data, aes(x = gam_fitted_count_t_minus_1)) +
        geom_histogram(bins = 30, fill = "turquoise", alpha = 0.7) +
        labs(title = "GAM Fitted Count (t-1)", x = "Count") +
        theme_minimal()

    p4b <- ggplot(data, aes(x = loess_fitted_count_t_minus_1)) +
        geom_histogram(bins = 30, fill = "plum", alpha = 0.7) +
        labs(title = "LOESS Fitted Count (t-1)", x = "Count") +
        theme_minimal()

    # Wind predictors
    p5 <- ggplot(data, aes(x = mean_wind_speed)) +
        geom_histogram(bins = 30, fill = "steelblue", alpha = 0.7) +
        labs(title = "Mean Wind Speed", x = "m/s") +
        theme_minimal()

    p6 <- ggplot(data, aes(x = max_wind_speed)) +
        geom_histogram(bins = 30, fill = "navy", alpha = 0.7) +
        labs(title = "Max Wind Speed (Gust)", x = "m/s") +
        theme_minimal()

    p7 <- ggplot(data, aes(x = cumulative_wind)) +
        geom_histogram(bins = 30, fill = "darkblue", alpha = 0.7) +
        labs(title = "Cumulative Wind", x = "Total m/s") +
        theme_minimal()

    p8 <- ggplot(data, aes(x = sustained_minutes_above_2mps)) +
        geom_histogram(bins = 30, fill = "purple", alpha = 0.7) +
        labs(title = "Sustained Minutes > 2 m/s", x = "Minutes") +
        theme_minimal()

    # Temperature predictors
    p9 <- ggplot(data, aes(x = mean_temp)) +
        geom_histogram(bins = 30, fill = "red", alpha = 0.7) +
        labs(title = "Mean Temperature", x = "°C") +
        theme_minimal()

    p10 <- ggplot(data, aes(x = max_temp)) +
        geom_histogram(bins = 30, fill = "darkred", alpha = 0.7) +
        labs(title = "Max Temperature", x = "°C") +
        theme_minimal()

    # Sunlight predictors
    p11 <- ggplot(data, aes(x = sunlight_proportion)) +
        geom_histogram(bins = 30, fill = "gold", alpha = 0.7) +
        labs(title = "Sunlight Proportion", x = "Proportion") +
        theme_minimal()

    p12 <- ggplot(data, aes(x = max_prop_sunlight)) +
        geom_histogram(bins = 30, fill = "orange", alpha = 0.7) +
        labs(title = "Max Sunlight Proportion", x = "Proportion") +
        theme_minimal()

    # Print individual plots (will appear on separate pages in PDF)
    print(p1)
    print(p2)
    print(p3)
    print(p3a)
    print(p3b)
    print(p3c)
    print(p3d)
    print(p4)
    print(p4a)
    print(p4b)
    print(p5)
    print(p6)
    print(p7)
    print(p8)
    print(p9)
    print(p10)
    print(p11)
    print(p12)

    return(list(p1, p2, p3, p3a, p3b, p3c, p3d, p4, p4a, p4b, p5, p6, p7, p8, p9, p10, p11, p
}
```

```r
# Function to create correlation matrix and plot
create_correlation_analysis <- function(data, dataset_name) {
    cat(paste0("\n=== CORRELATION ANALYSIS FOR: ", dataset_name, " ===\n"))

    # Select numeric variables for correlation analysis
    numeric_vars <- data %>%
        select(
            total_butterflies, butterfly_diff, butterfly_log_diff,
            abundance_t_minus_1, butterflies_sun_t_minus_1, temperature_t_minus_1,
            mean_wind_speed, max_wind_speed, sd_wind_speed, cumulative_wind,
            gust_factor, sustained_minutes_above_2mps, gust_minutes_above_2mps,
            mean_temp, max_temp, min_temp, sd_temp,
            sunlight_proportion, max_prop_sunlight, min_prop_sunlight, sd_prop_sunlight
        ) %>%
        na.omit()

    # Calculate correlation matrix
    cor_matrix <- cor(numeric_vars, use = "complete.obs")

    # Create correlation plot
    corrplot(cor_matrix,
             method = "color",
             type = "upper",
             order = "hclust",
             tl.cex = 0.8,
             tl.col = "black",
             tl.srt = 45,
             title = paste("Correlation Matrix -", dataset_name),
             mar = c(0,0,1,0))

    # Print highly correlated pairs (>0.7 or <-0.7)
    high_cor <- which(abs(cor_matrix) > 0.7 & cor_matrix != 1, arr.ind = TRUE)
    if (nrow(high_cor) > 0) {
        cat("\nHighly correlated variable pairs (|r| > 0.7):\n")
        for (i in 1:nrow(high_cor)) {
            row_var <- rownames(cor_matrix)[high_cor[i, 1]]
            col_var <- colnames(cor_matrix)[high_cor[i, 2]]
            cor_val <- cor_matrix[high_cor[i, 1], high_cor[i, 2]]
            cat(sprintf("%s <-> %s: r = %.3f\n", row_var, col_var, cor_val))
        }
    }
    return(cor_matrix)
}
```

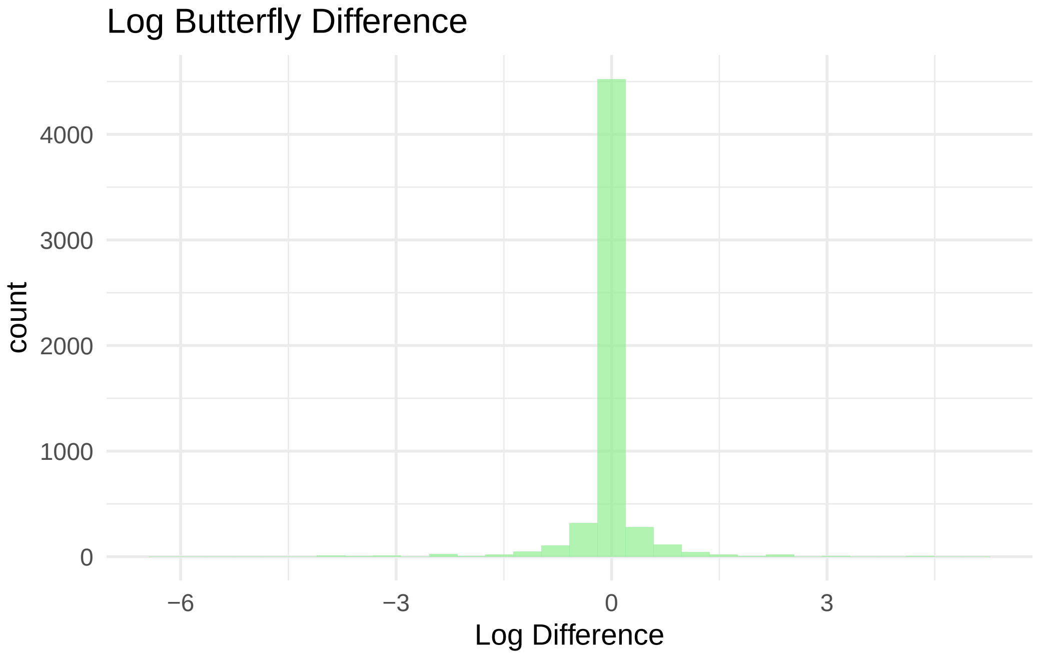
```
# Run exploratory analysis for 30-minute lag data (most observations)
if (nrow(data_30m) > 0) {
    hist_30m <- create_histograms(data_30m, "30-Minute Lag")
    cor_30m <- create_correlation_analysis(data_30m, "30-Minute Lag")
}
```

```
=== HISTOGRAMS FOR: 30-Minute Lag ===
```

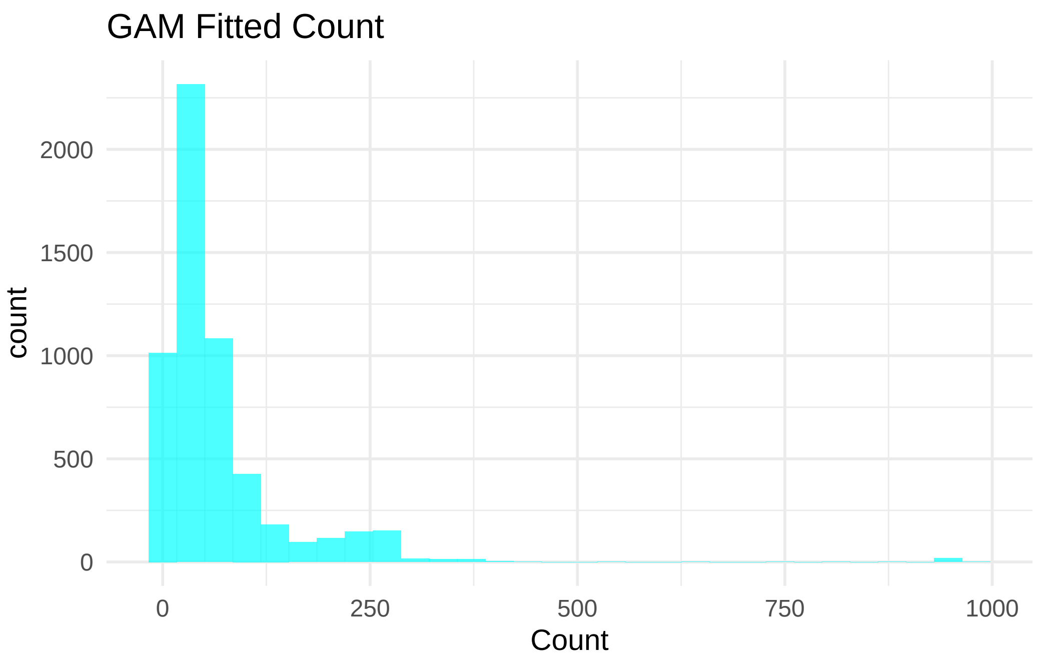




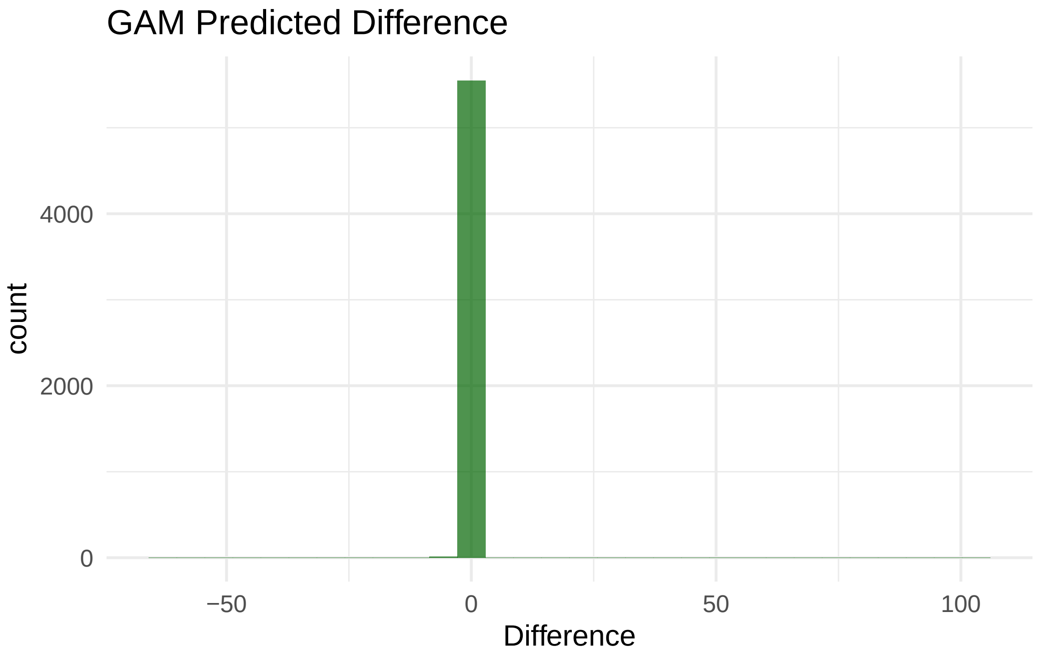
Butterfly Difference

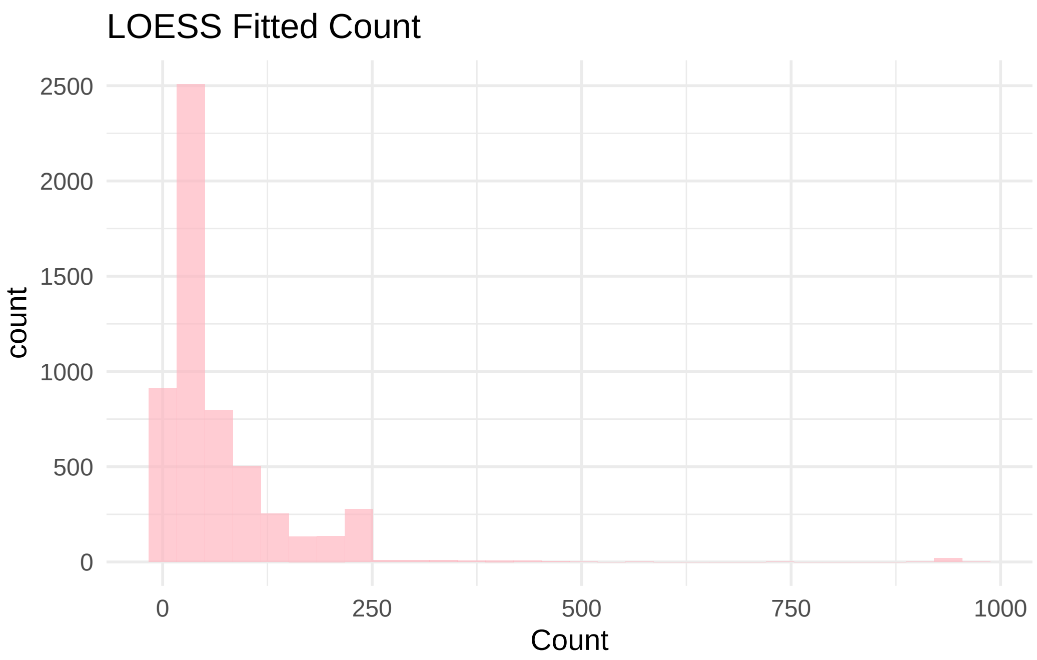


Log Butterfly Difference

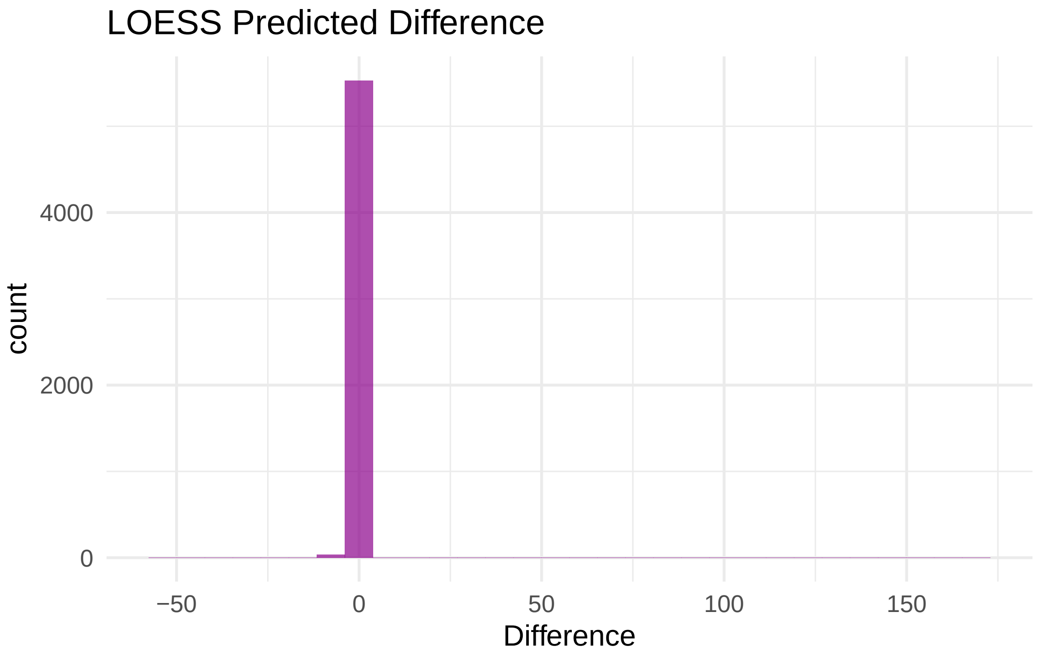


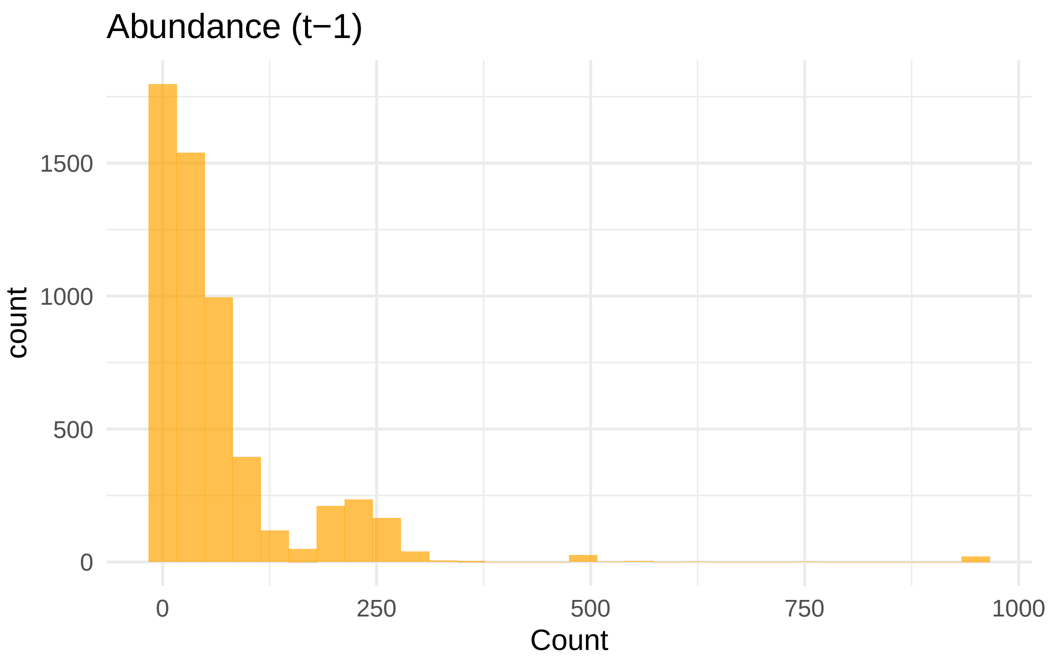
```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```



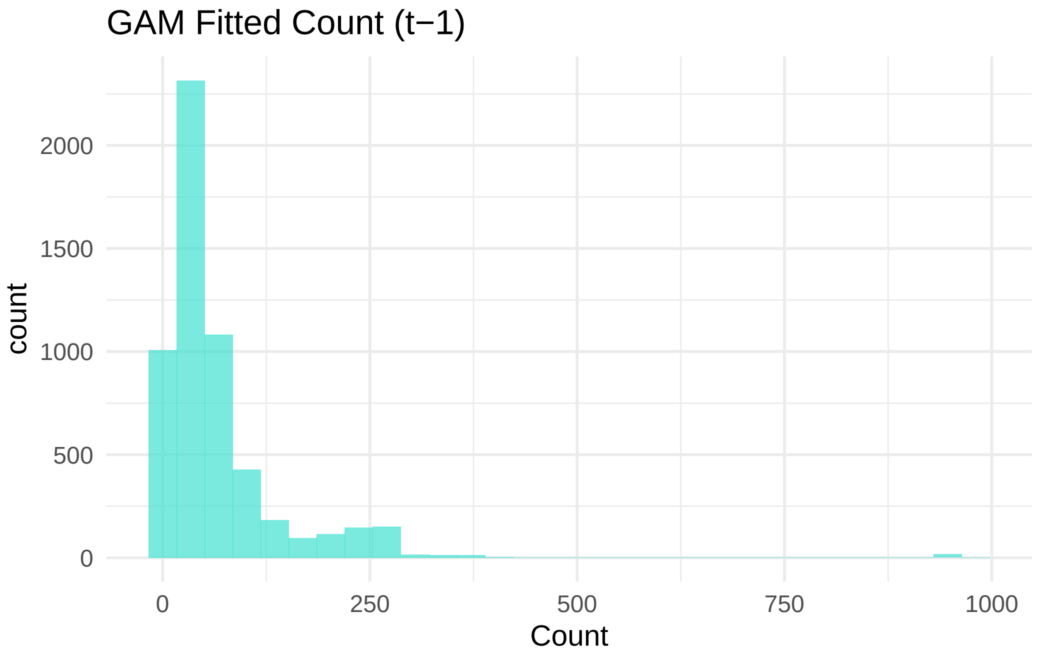


```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```

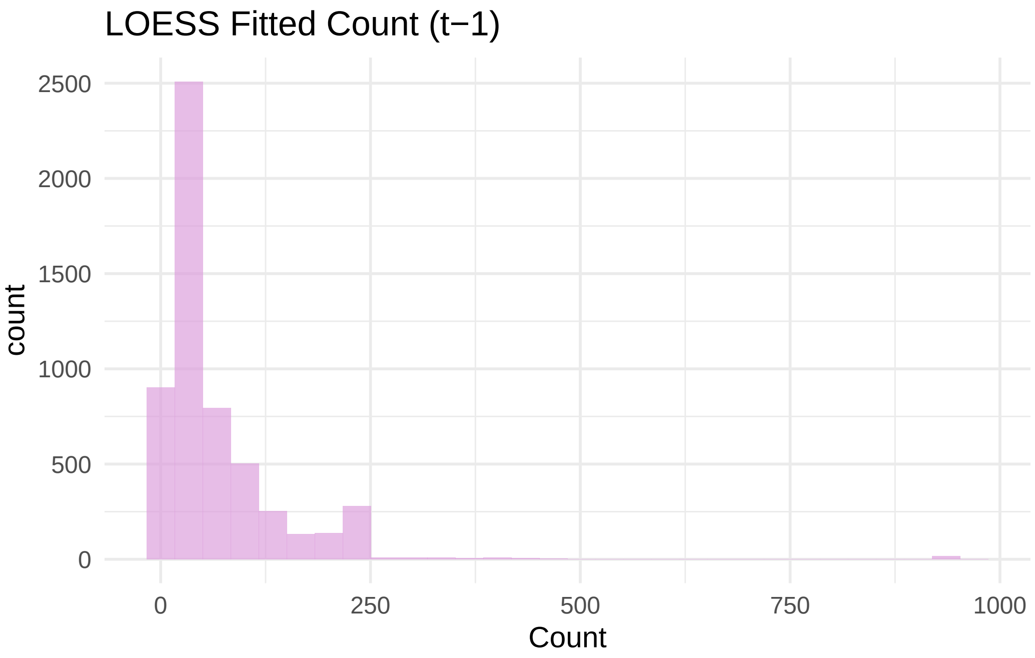


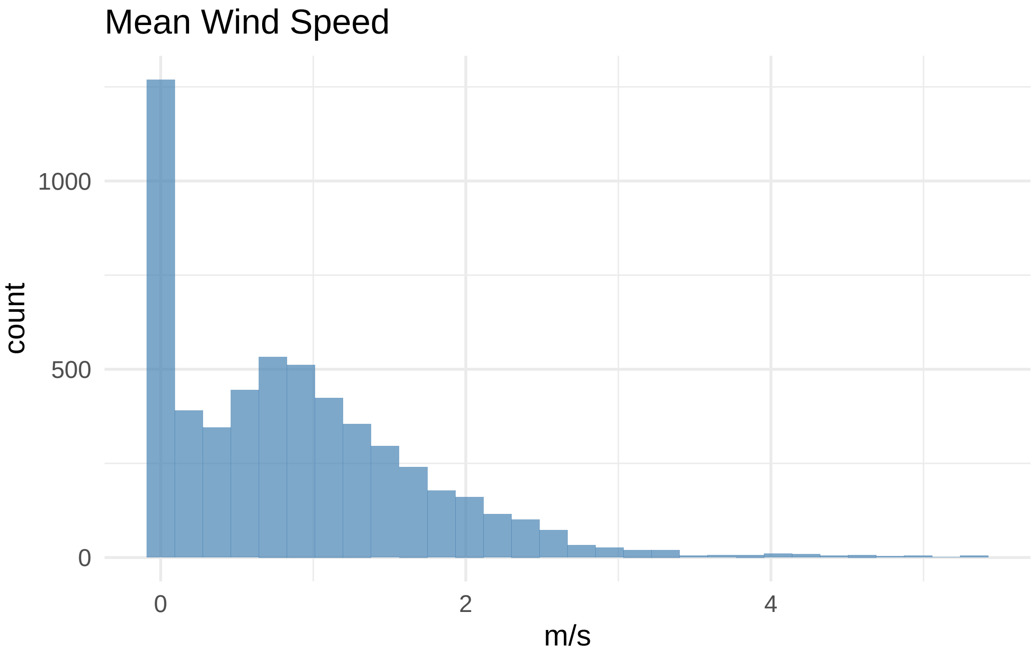


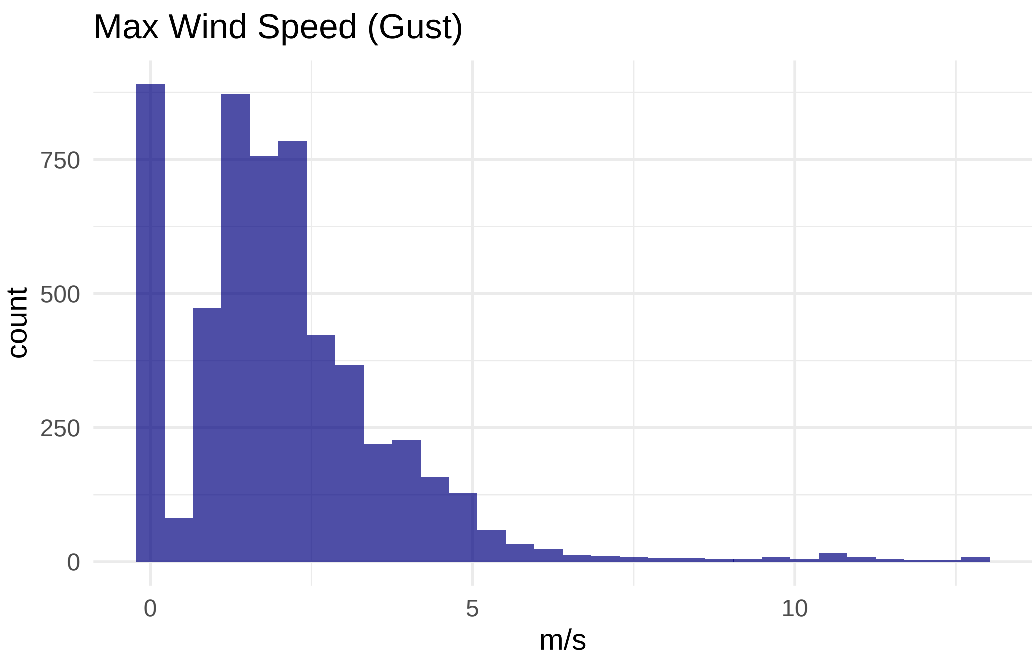
```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```

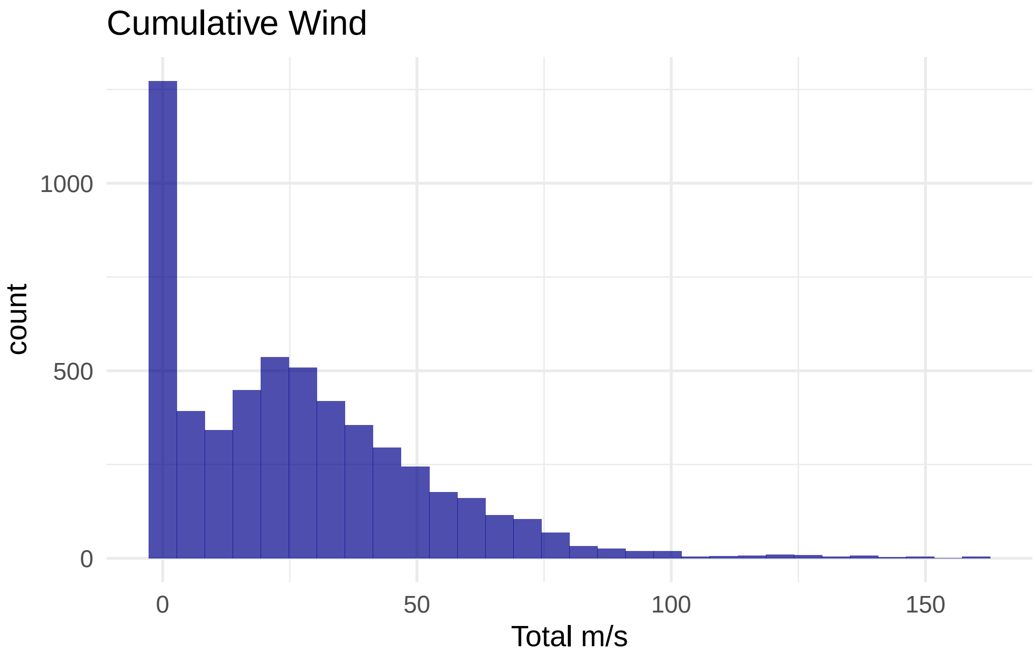


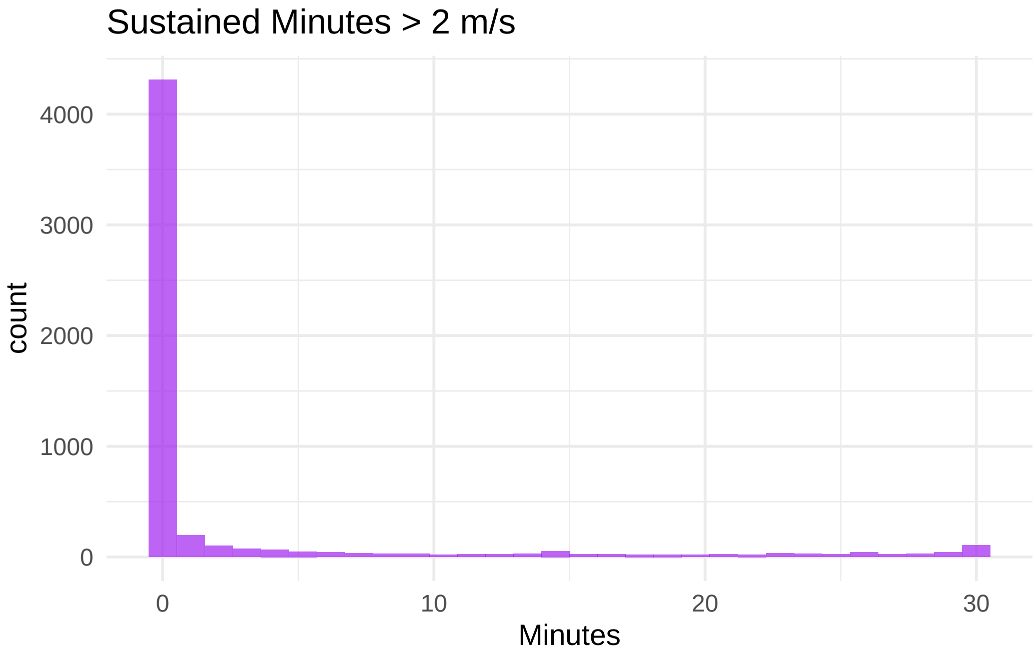
```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```

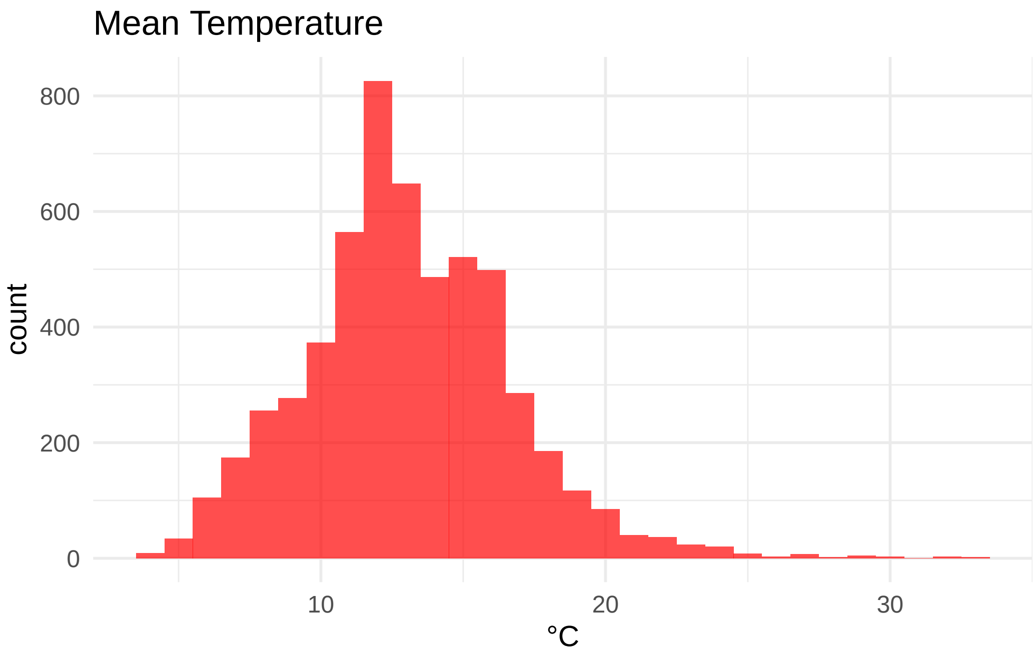


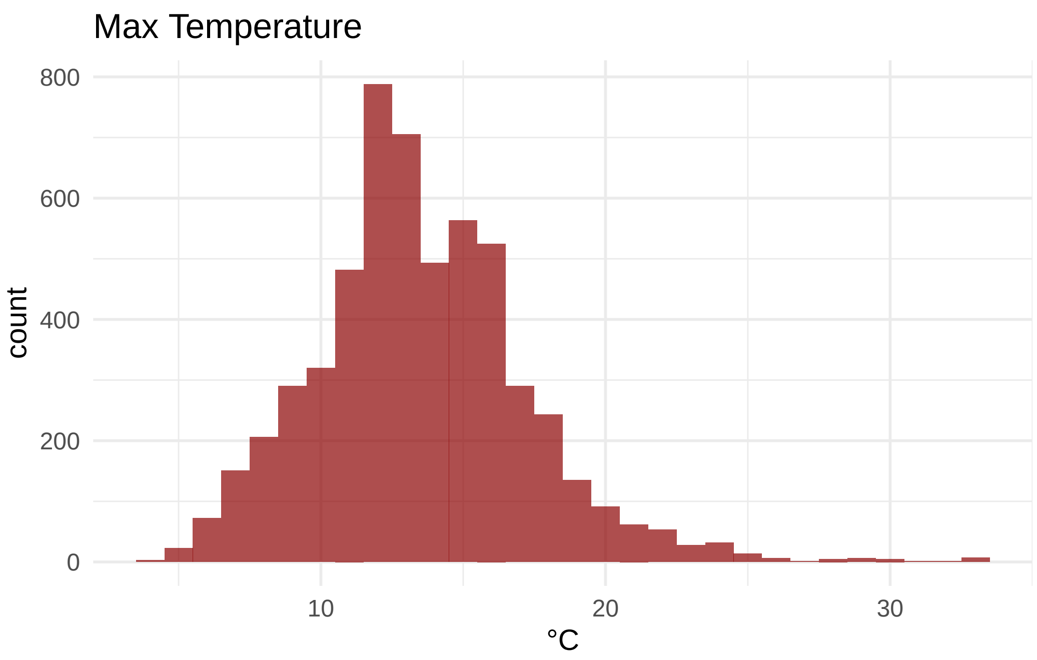


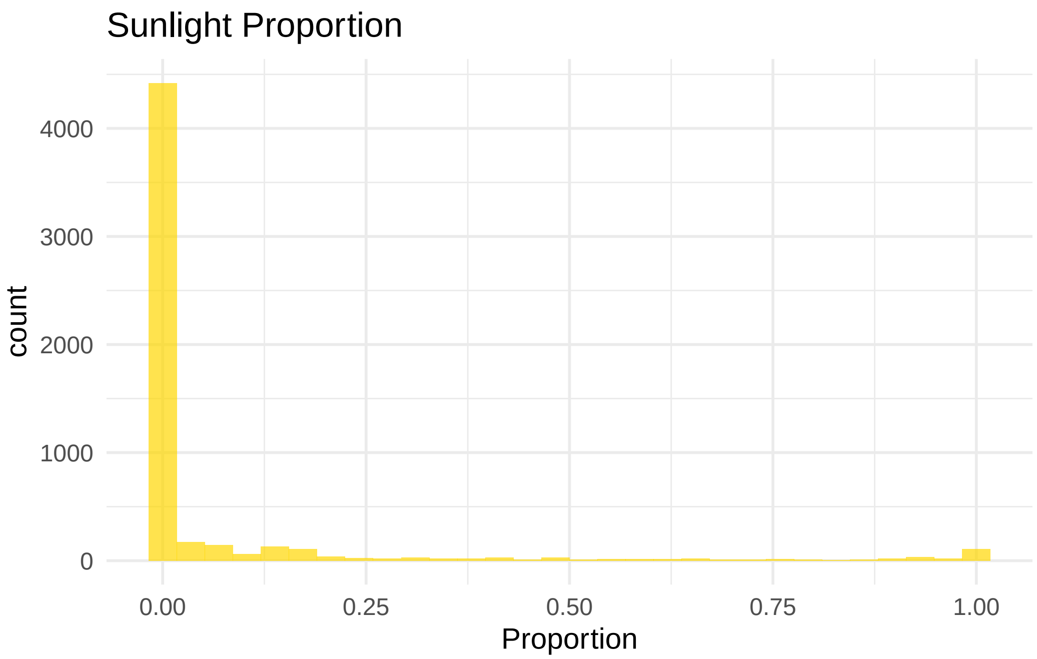

Max Wind Speed (Gust)
count
750
500
250
0
0
5
10
m/s

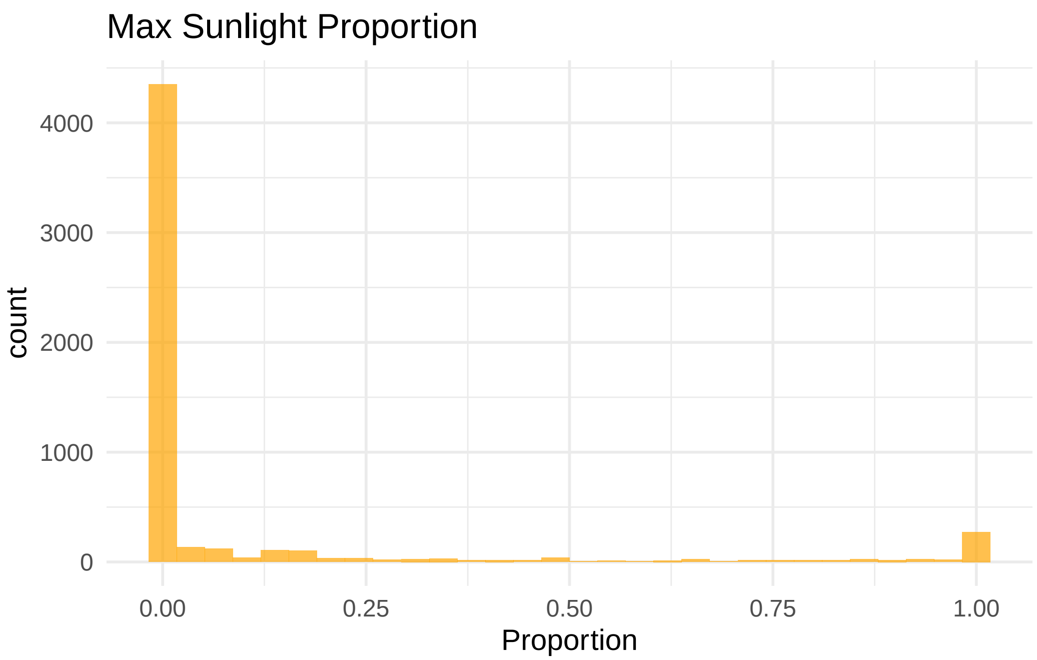

Cumulative Wind
count
1000
500
0
0
50
100
150
Total m/s


Sustained Minutes > 2 m/s
4000
3000
2000
1000
0
count
0
10
20
30
Minutes

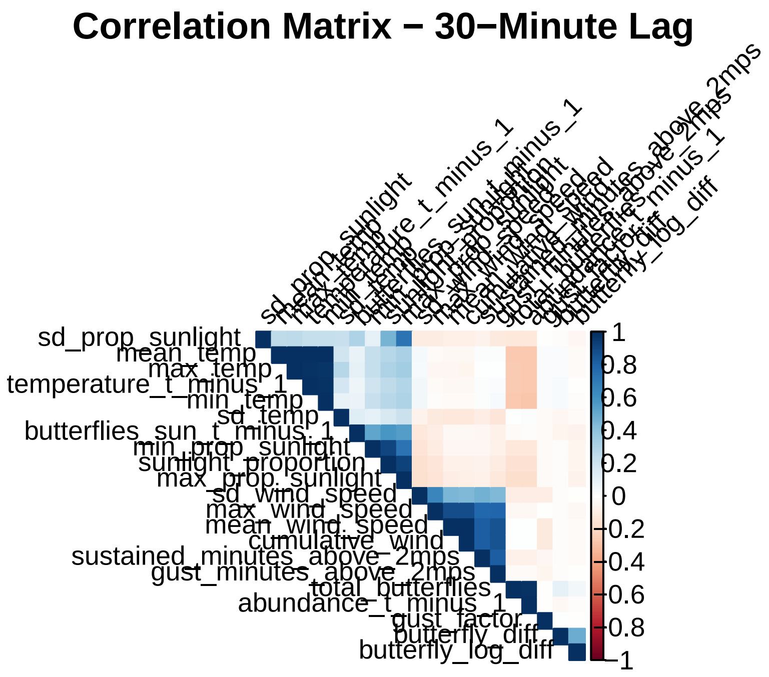

Mean Temperature
800
600
400
200
0
count
10
20
30
°C


Max Temperature
count
800
600
400
200
0
10
20
30
°C


Sunlight Proportion
count
4000
3000
2000
1000
0
0.00
0.25
0.50
0.75
1.00
Proportion



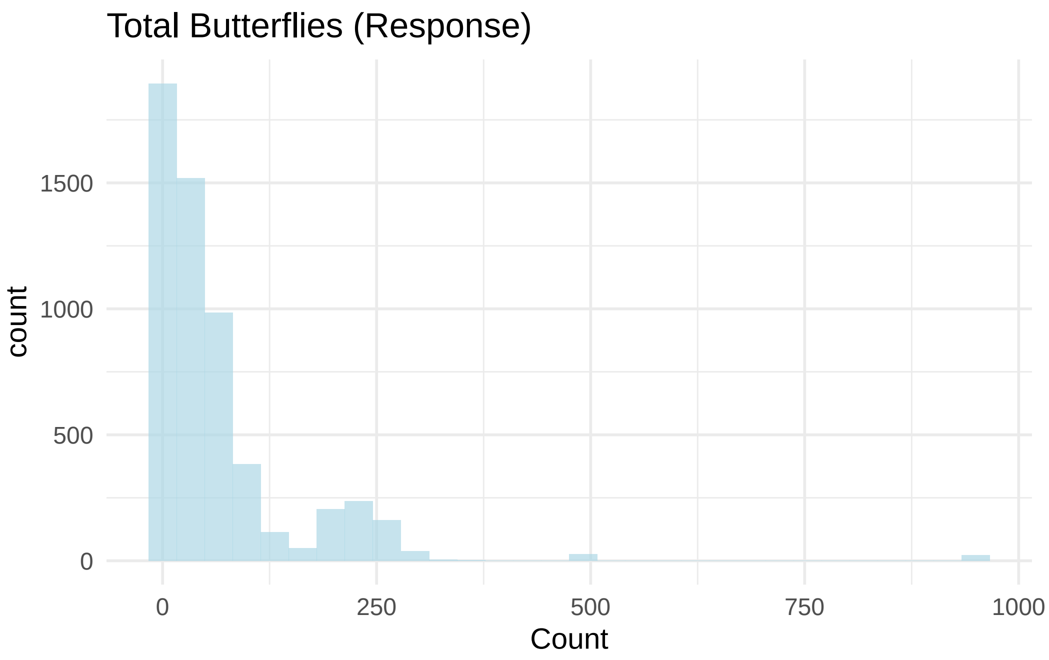
```
=== CORRELATION ANALYSIS FOR: 30-Minute Lag ===
```

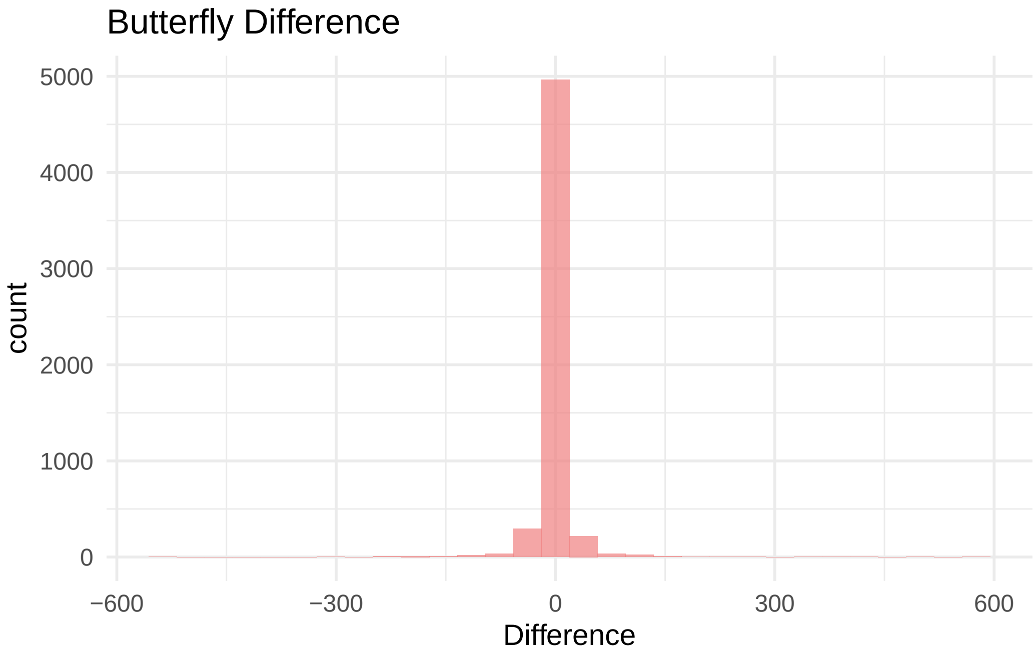


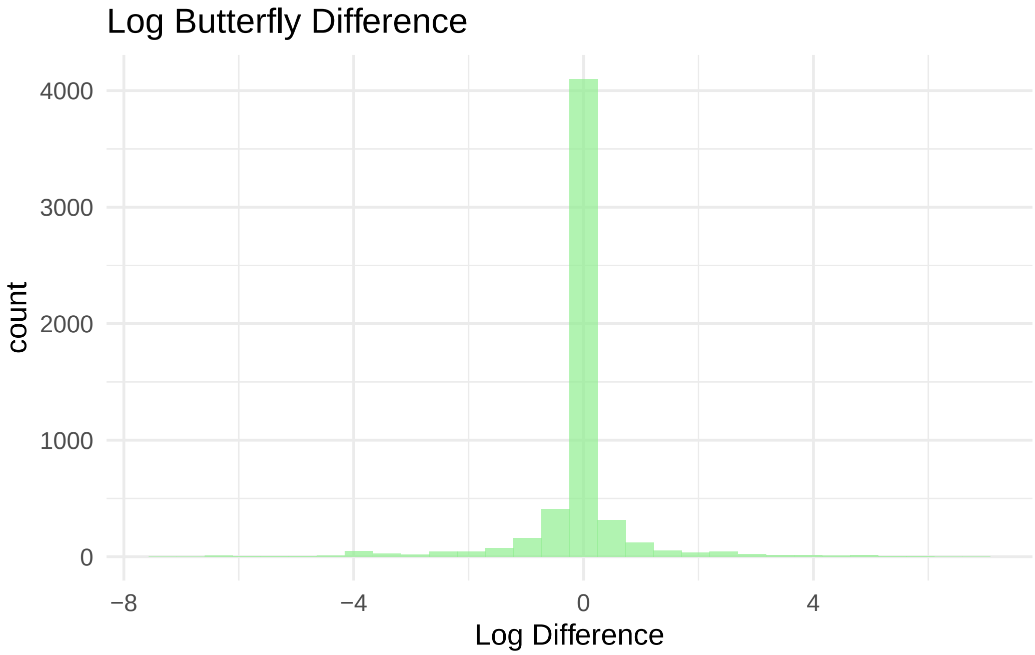
```
Highly correlated variable pairs (|r| > 0.7):
abundance_t_minus_1 <-> total_butterflies: r = 0.989
total_butterflies <-> abundance_t_minus_1: r = 0.989
mean_temp <-> temperature_t_minus_1: r = 0.991
max_temp <-> temperature_t_minus_1: r = 0.986
min_temp <-> temperature_t_minus_1: r = 0.987
max_wind_speed <-> mean_wind_speed: r = 0.885
cumulative_wind <-> mean_wind_speed: r = 1.000
sustained_minutes_above_2mps <-> mean_wind_speed: r = 0.824
gust_minutes_above_2mps <-> mean_wind_speed: r = 0.863
mean_wind_speed <-> max_wind_speed: r = 0.885
cumulative_wind <-> max_wind_speed: r = 0.885
sustained_minutes_above_2mps <-> max_wind_speed: r = 0.787
gust_minutes_above_2mps <-> max_wind_speed: r = 0.793
mean_wind_speed <-> cumulative_wind: r = 1.000
max_wind_speed <-> cumulative_wind: r = 0.885
sustained_minutes_above_2mps <-> cumulative_wind: r = 0.824
gust_minutes_above_2mps <-> cumulative_wind: r = 0.863
mean_wind_speed <-> sustained_minutes_above_2mps: r = 0.824
max_wind_speed <-> sustained_minutes_above_2mps: r = 0.787
cumulative_wind <-> sustained_minutes_above_2mps: r = 0.824
gust_minutes_above_2mps <-> sustained_minutes_above_2mps: r = 0.823
mean_wind_speed <-> gust_minutes_above_2mps: r = 0.863
max_wind_speed <-> gust_minutes_above_2mps: r = 0.793
cumulative_wind <-> gust_minutes_above_2mps: r = 0.863
sustained_minutes_above_2mps <-> gust_minutes_above_2mps: r = 0.823
temperature_t_minus_1 <-> mean_temp: r = 0.991
max_temp <-> mean_temp: r = 0.995
min_temp <-> mean_temp: r = 0.994
temperature_t_minus_1 <-> max_temp: r = 0.986
mean_temp <-> max_temp: r = 0.995
min_temp <-> max_temp: r = 0.980
temperature_t_minus_1 <-> min_temp: r = 0.987
mean_temp <-> min_temp: r = 0.994
max_temp <-> min_temp: r = 0.980
max_prop_sunlight <-> sunlight_proportion: r = 0.925
min_prop_sunlight <-> sunlight_proportion: r = 0.911
sunlight_proportion <-> max_prop_sunlight: r = 0.925
min_prop_sunlight <-> max_prop_sunlight: r = 0.740
sd_prop_sunlight <-> max_prop_sunlight: r = 0.733
sunlight_proportion <-> min_prop_sunlight: r = 0.911
max_prop_sunlight <-> min_prop_sunlight: r = 0.740
max_prop_sunlight <-> sd_prop_sunlight: r = 0.733
```

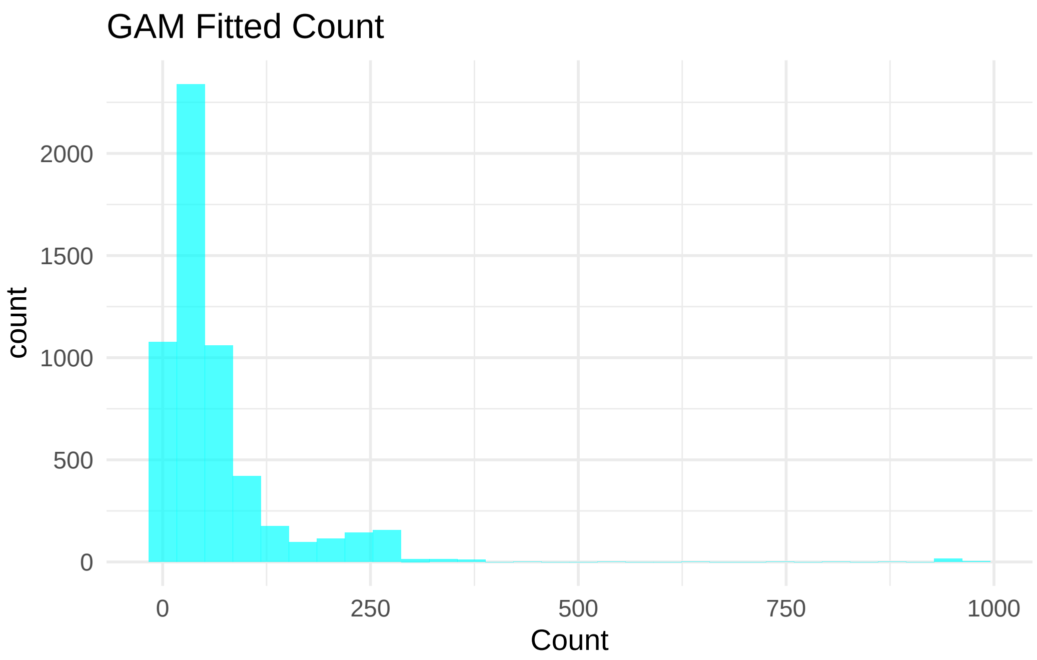
```
# Run exploratory analysis for 2-hour lag data
if (nrow(data_120m) > 0) {
    hist_120m <- create_histograms(data_120m, "2-Hour Lag")
    cor_120m <- create_correlation_analysis(data_120m, "2-Hour Lag")
}
```

```
=== HISTOGRAMS FOR: 2-Hour Lag ===
```

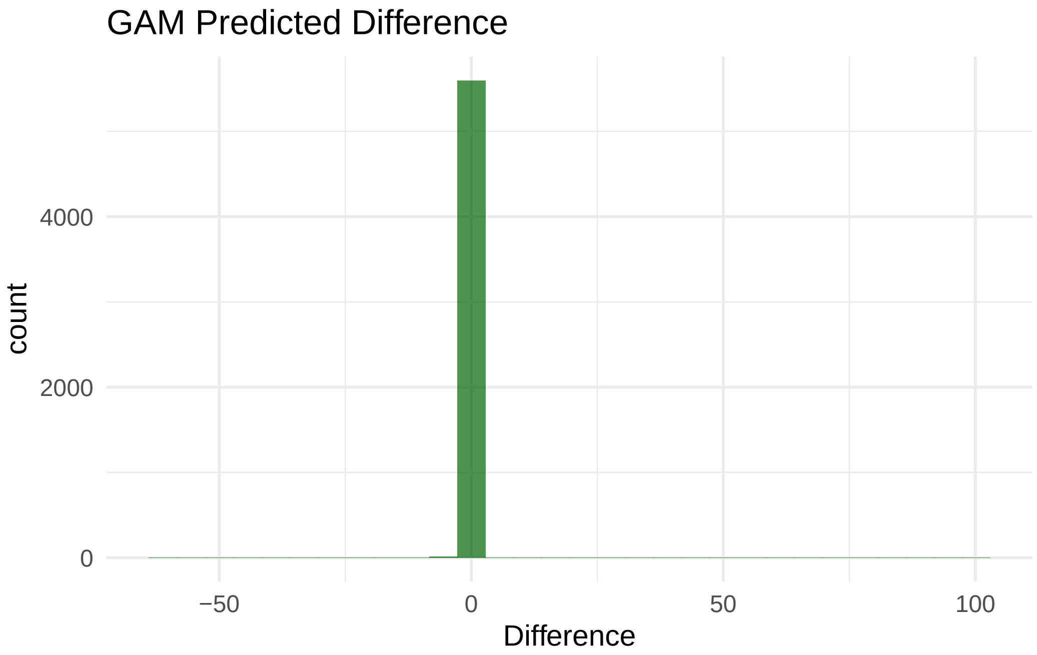


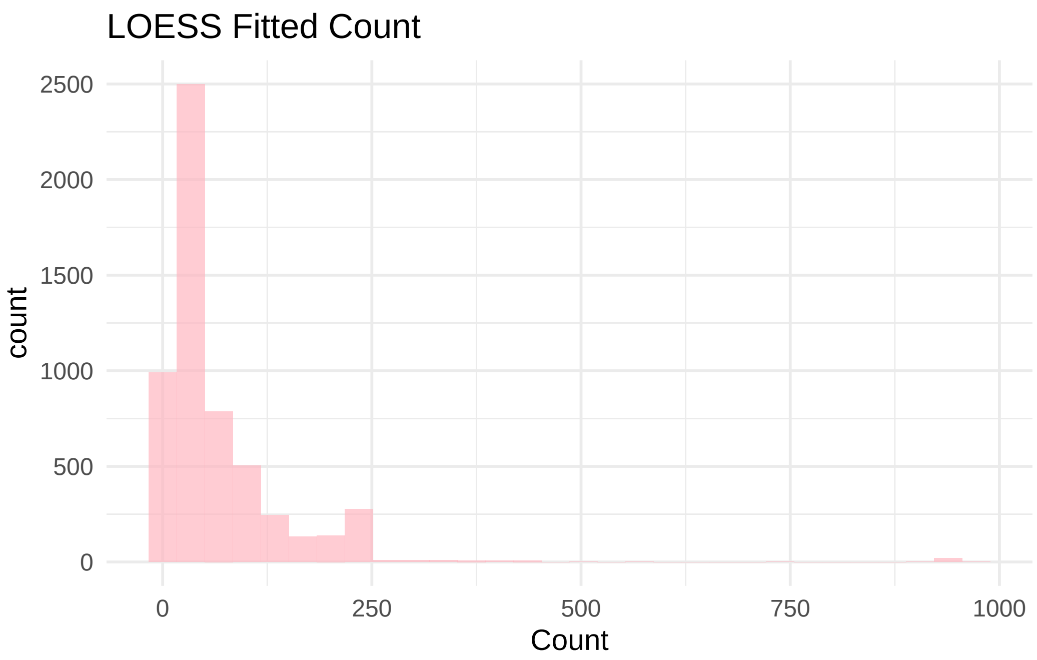

Butterfly Difference
5000
4000
3000
2000
1000
0
count
−600
−300
0
300
600
Difference


Log Butterfly Difference
4000
3000
2000
1000
0
count
−8
−4
0
4
Log Difference

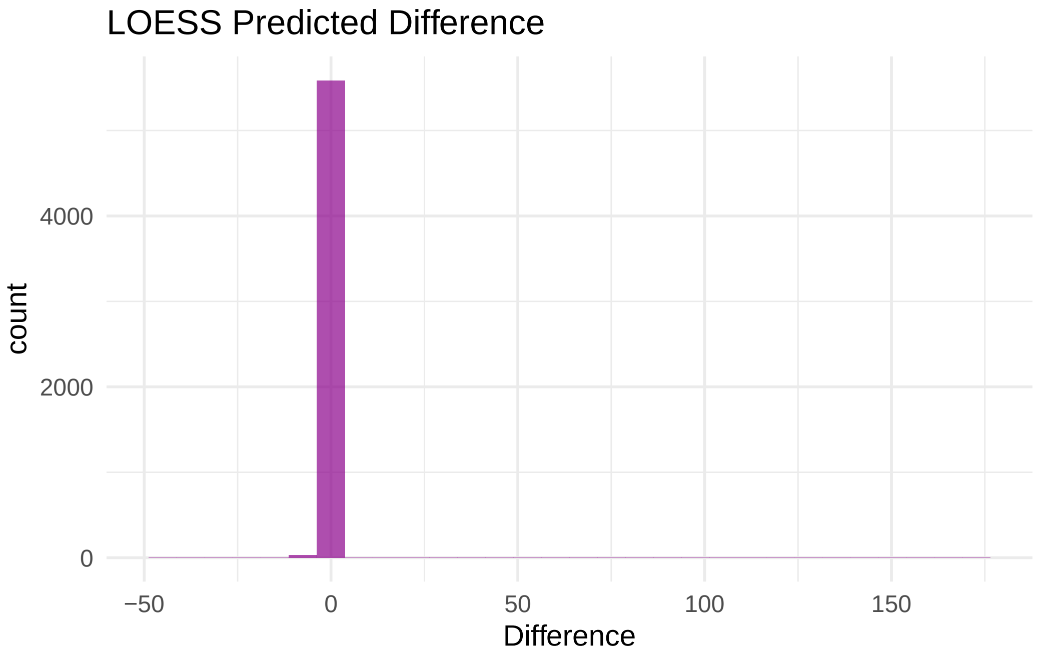


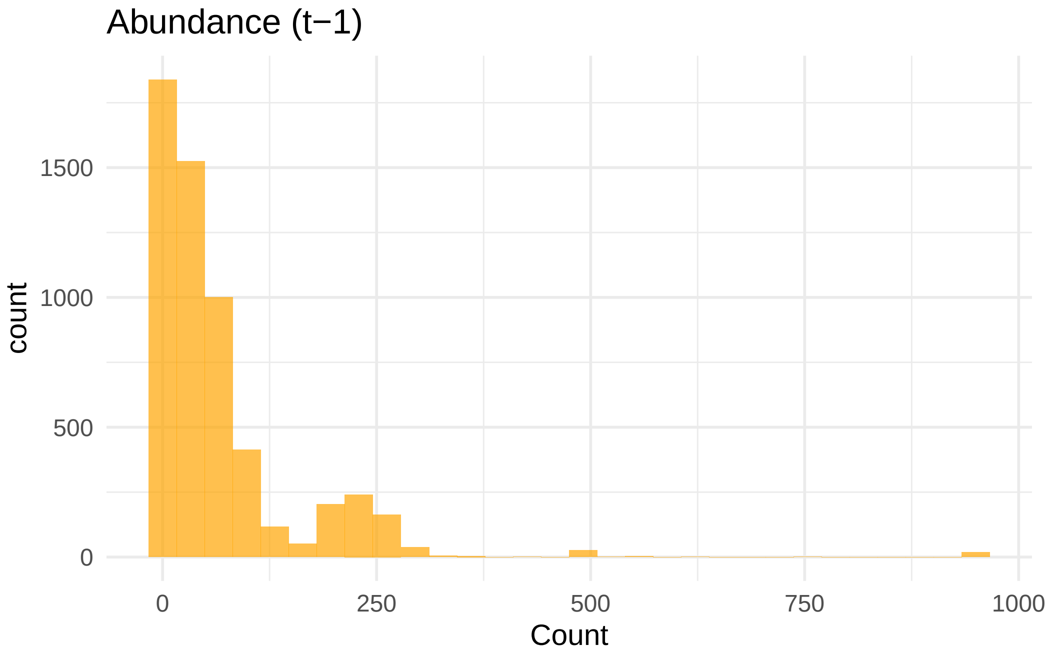
```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```



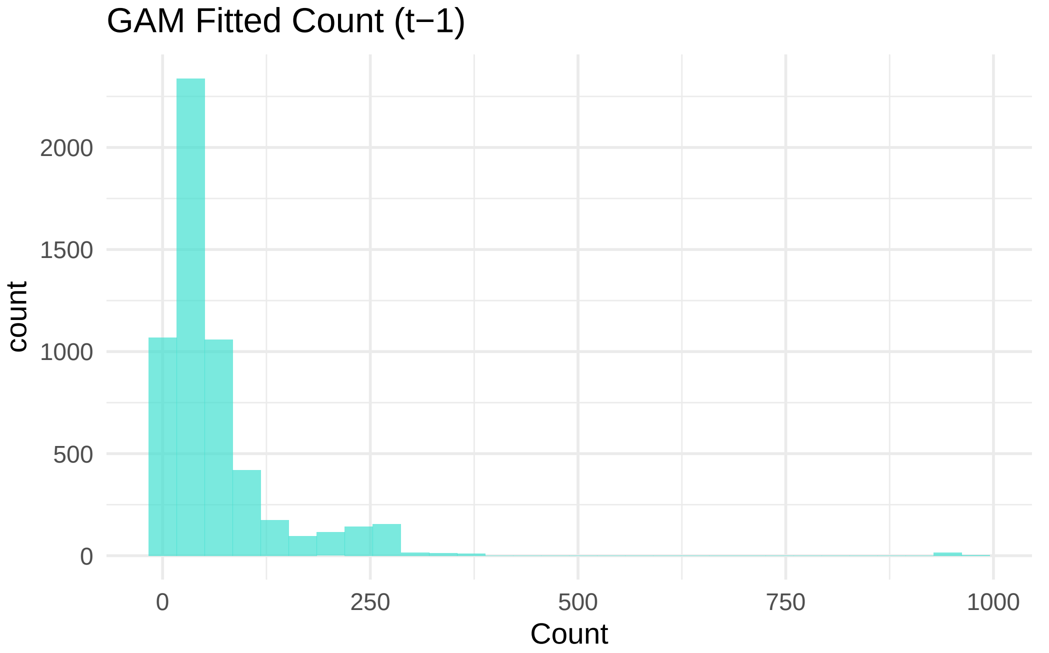


```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```

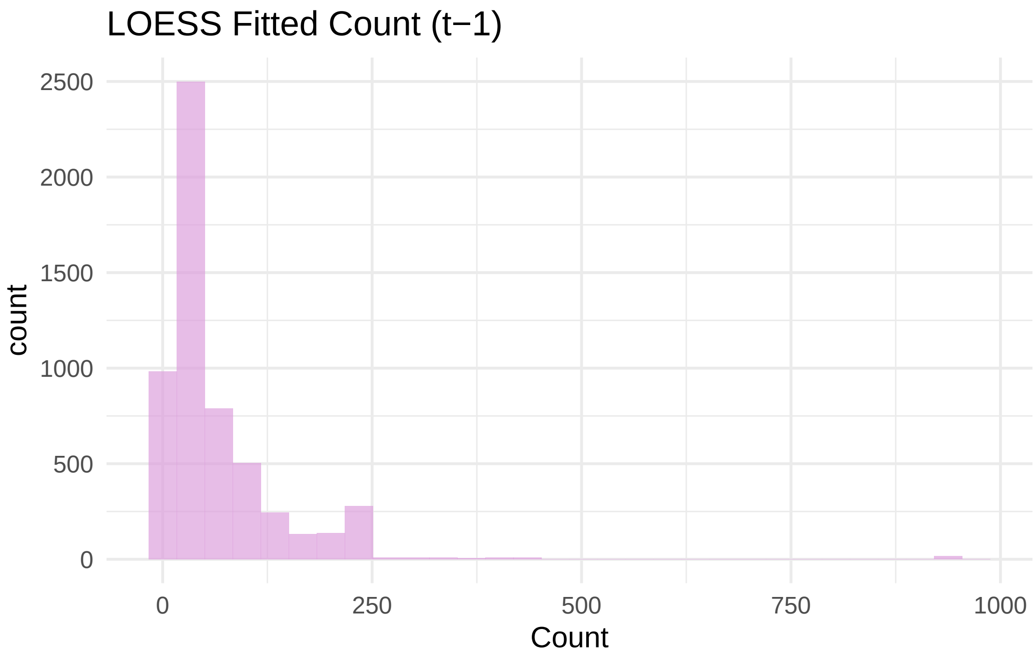


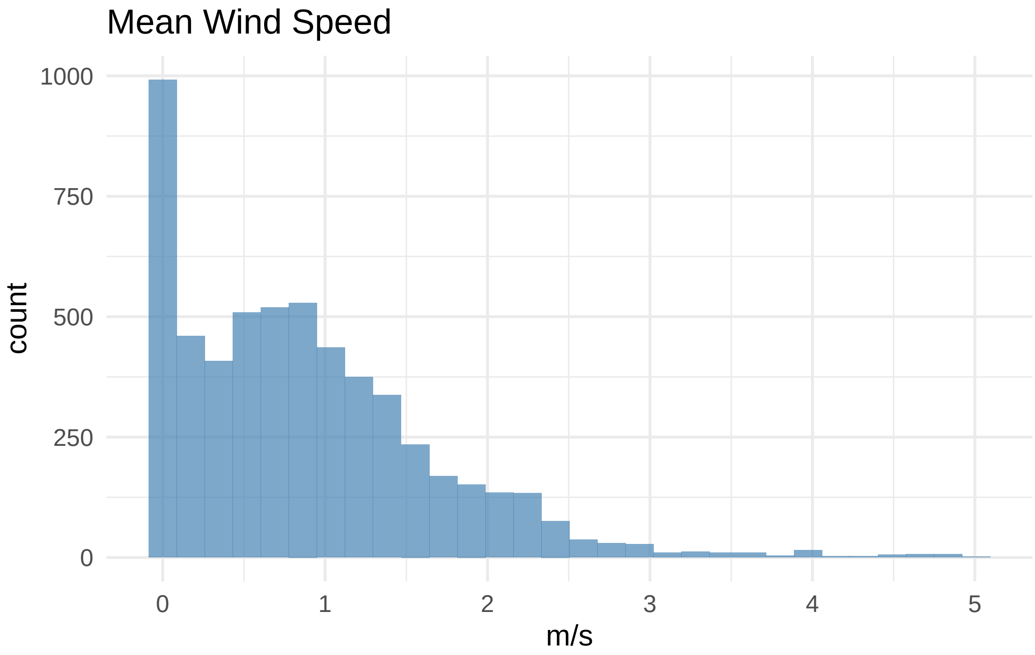


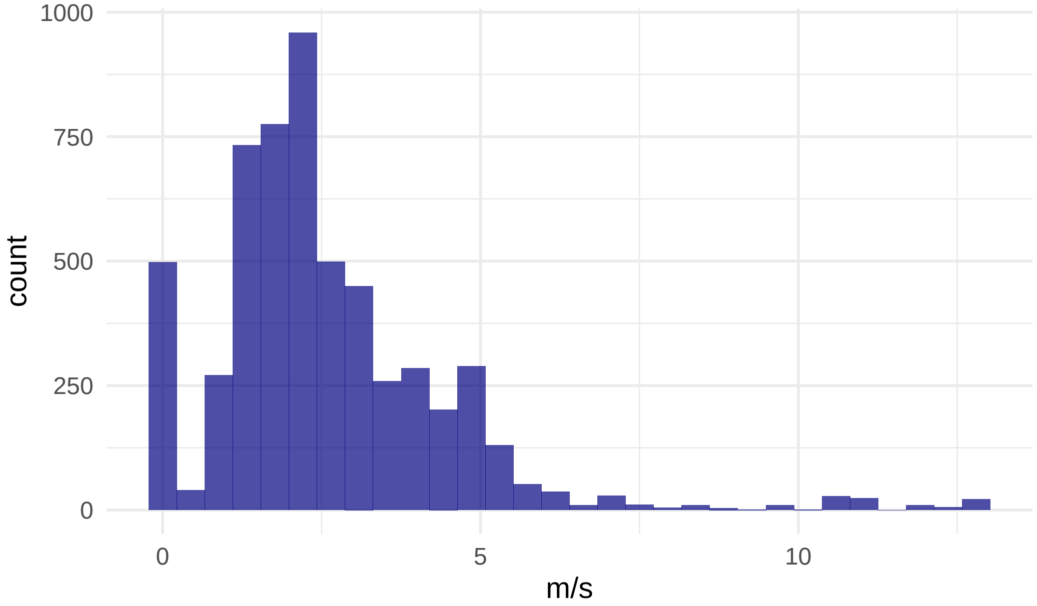
```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```



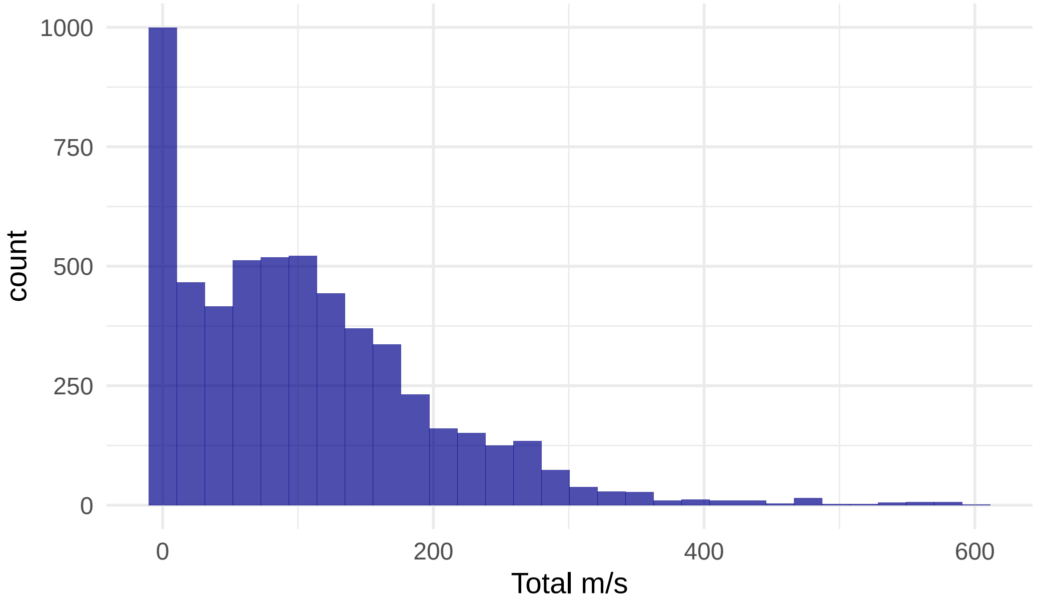
```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_bin()`).
```



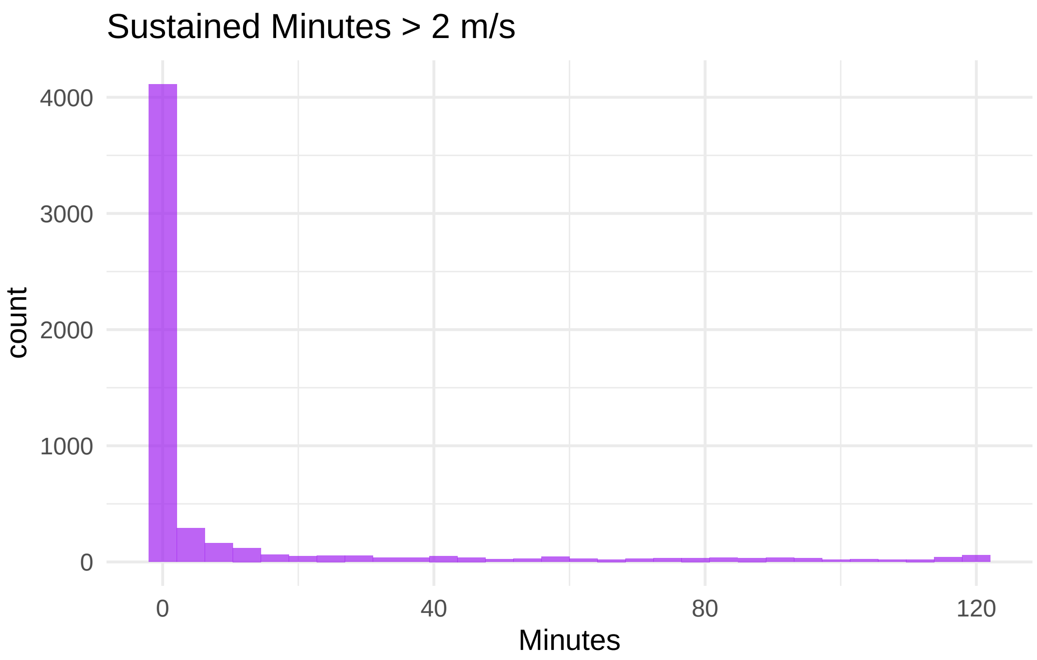


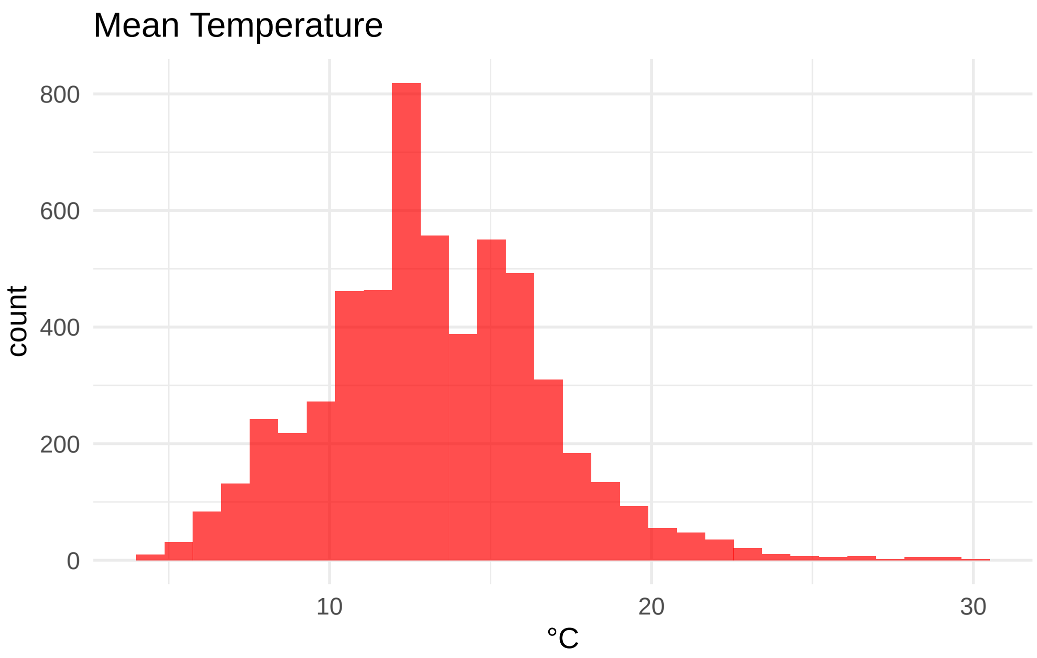


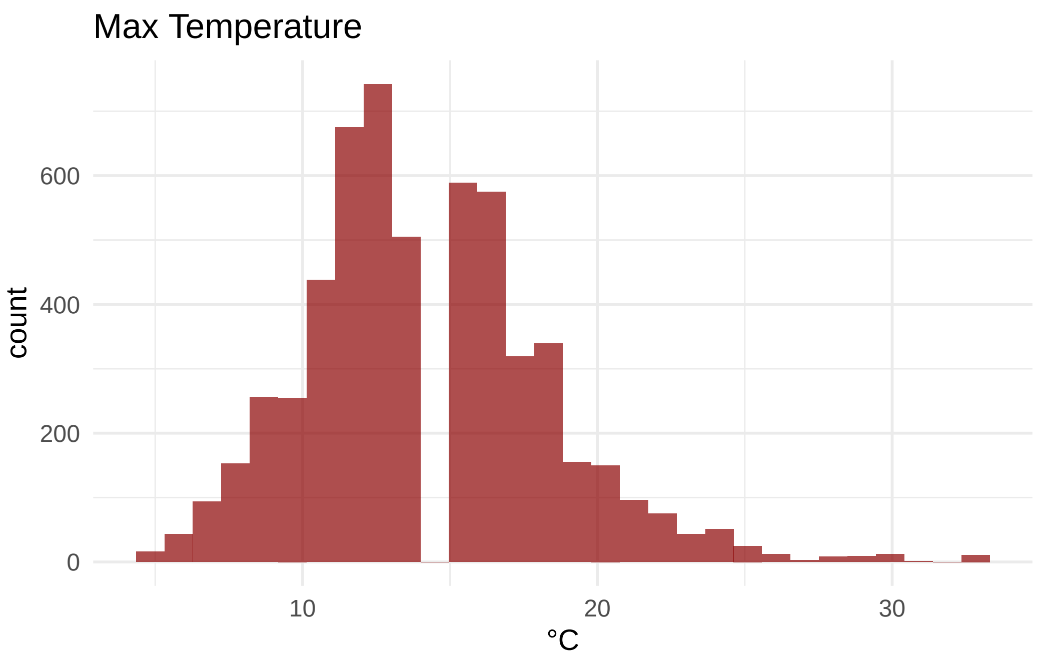
Max Wind Speed (Gust)

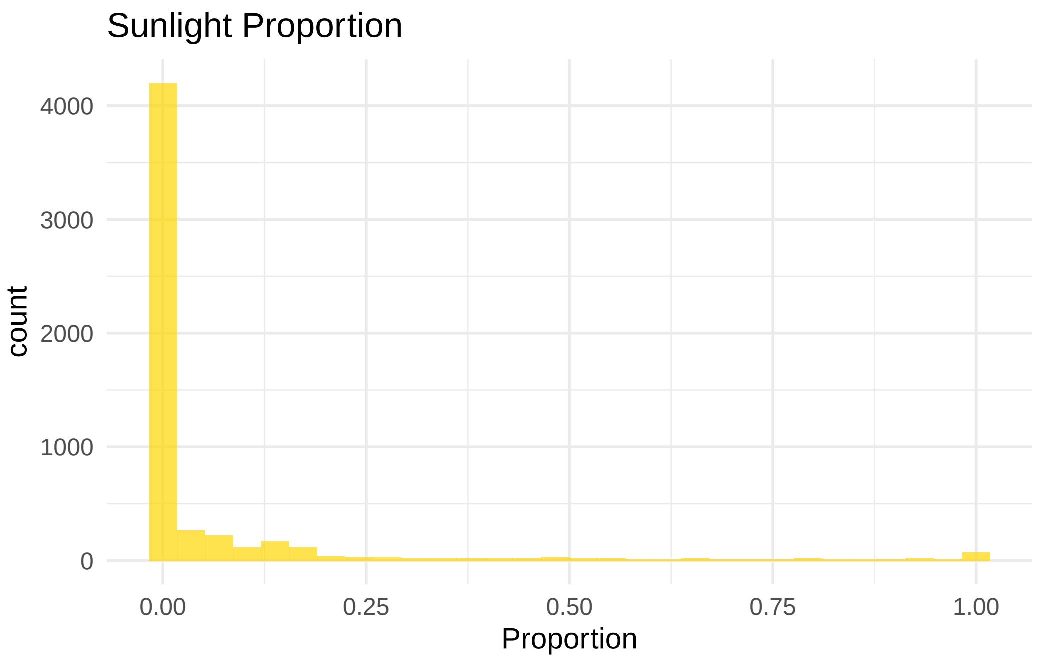


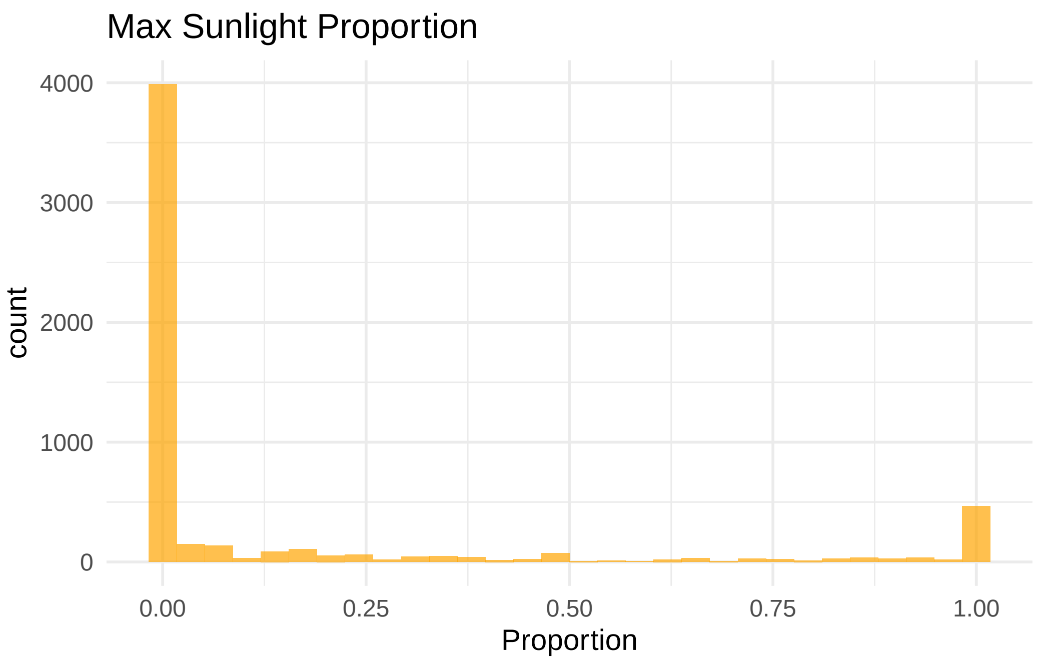
Cumulative Wind


Sustained Minutes > 2 m/s
count
4000
3000
2000
1000
0
0
40
80
120
Minutes

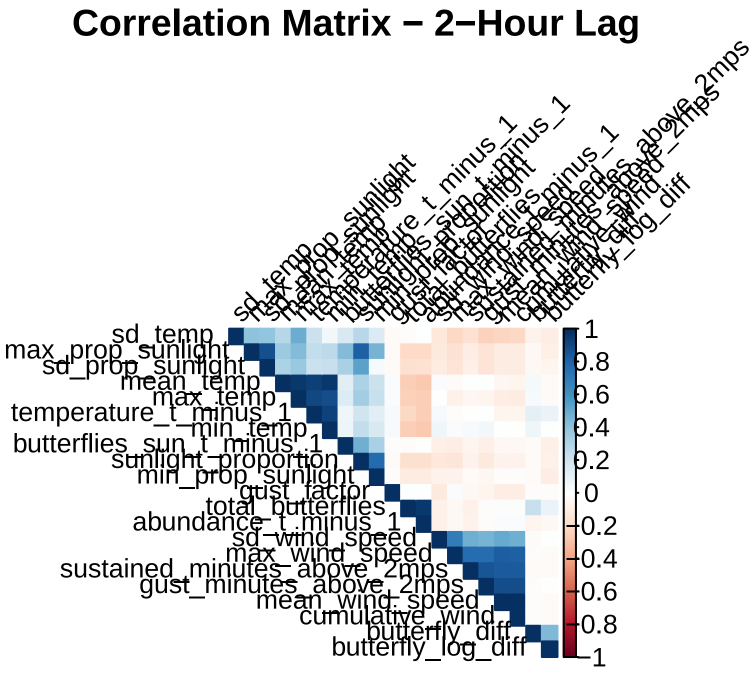

Mean Temperature
count
800
600
400
200
0
10
20
30
°C


Max Temperature


Sunlight Proportion



```
=== CORRELATION ANALYSIS FOR: 2-Hour Lag ===
```



```
Highly correlated variable pairs (|r| > 0.7):
abundance_t_minus_1 <-> total_butterflies: r = 0.961
total_butterflies <-> abundance_t_minus_1: r = 0.961
mean_temp <-> temperature_t_minus_1: r = 0.939
max_temp <-> temperature_t_minus_1: r = 0.906
min_temp <-> temperature_t_minus_1: r = 0.928
max_wind_speed <-> mean_wind_speed: r = 0.823
cumulative_wind <-> mean_wind_speed: r = 0.996
sustained_minutes_above_2mps <-> mean_wind_speed: r = 0.838
gust_minutes_above_2mps <-> mean_wind_speed: r = 0.883
mean_wind_speed <-> max_wind_speed: r = 0.823
cumulative_wind <-> max_wind_speed: r = 0.819
sustained_minutes_above_2mps <-> max_wind_speed: r = 0.761
gust_minutes_above_2mps <-> max_wind_speed: r = 0.765
mean_wind_speed <-> cumulative_wind: r = 0.996
max_wind_speed <-> cumulative_wind: r = 0.819
sustained_minutes_above_2mps <-> cumulative_wind: r = 0.839
gust_minutes_above_2mps <-> cumulative_wind: r = 0.882
mean_wind_speed <-> sustained_minutes_above_2mps: r = 0.838
max_wind_speed <-> sustained_minutes_above_2mps: r = 0.761
cumulative_wind <-> sustained_minutes_above_2mps: r = 0.839
gust_minutes_above_2mps <-> sustained_minutes_above_2mps: r = 0.858
mean_wind_speed <-> gust_minutes_above_2mps: r = 0.883
max_wind_speed <-> gust_minutes_above_2mps: r = 0.765
cumulative_wind <-> gust_minutes_above_2mps: r = 0.882
sustained_minutes_above_2mps <-> gust_minutes_above_2mps: r = 0.858
temperature_t_minus_1 <-> mean_temp: r = 0.939
max_temp <-> mean_temp: r = 0.969
min_temp <-> mean_temp: r = 0.967
temperature_t_minus_1 <-> max_temp: r = 0.906
mean_temp <-> max_temp: r = 0.969
min_temp <-> max_temp: r = 0.888
temperature_t_minus_1 <-> min_temp: r = 0.928
mean_temp <-> min_temp: r = 0.967
max_temp <-> min_temp: r = 0.888
max_prop_sunlight <-> sunlight_proportion: r = 0.810
min_prop_sunlight <-> sunlight_proportion: r = 0.775
sunlight_proportion <-> max_prop_sunlight: r = 0.810
sd_prop_sunlight <-> max_prop_sunlight: r = 0.879
sunlight_proportion <-> min_prop_sunlight: r = 0.775
max_prop_sunlight <-> sd_prop_sunlight: r = 0.879
```

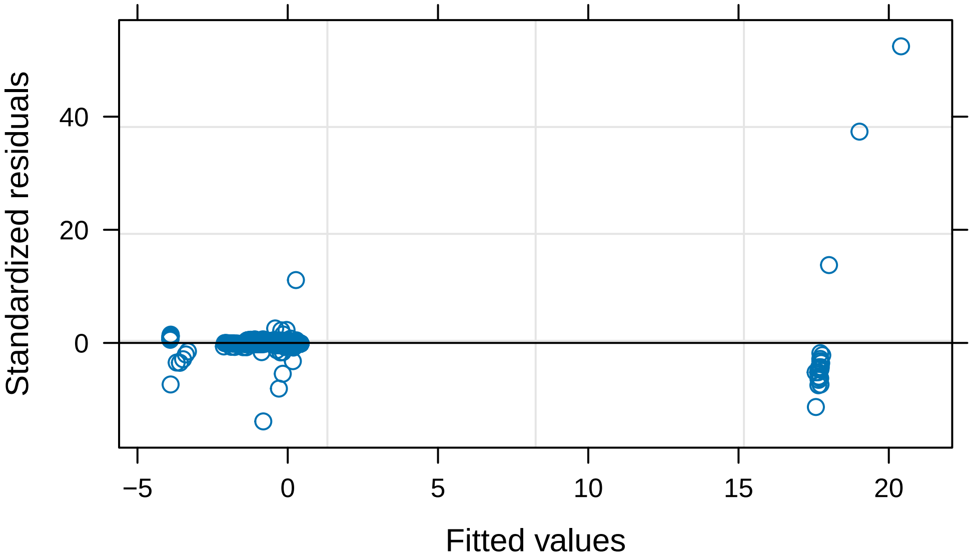
Models

30 min

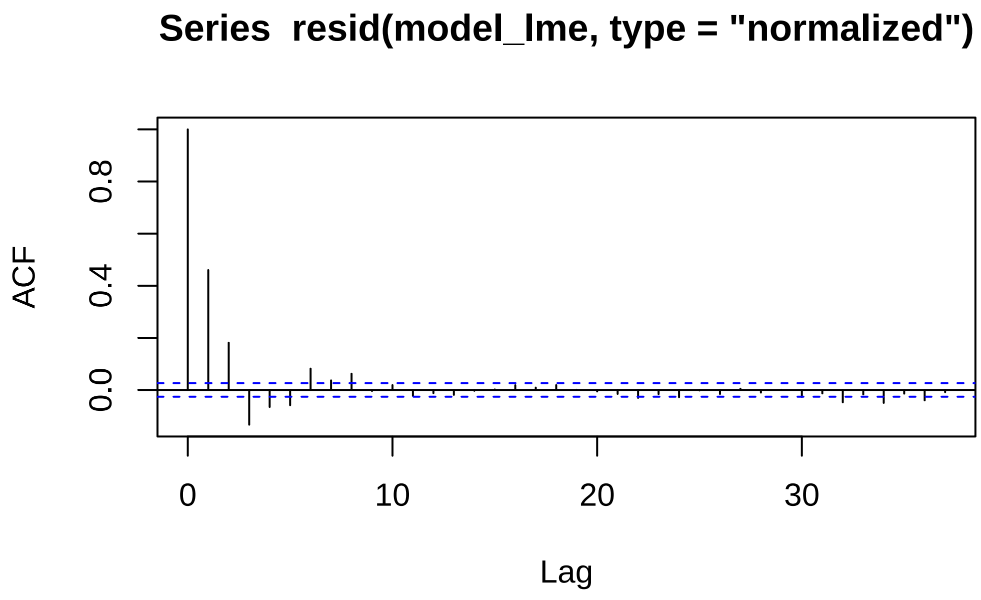
```r
library(nlme)

# Mixed model with AR(1) correlation structure within view_id
model_lme <- lme(
  gam_pred_diff ~ gam_fitted_count_t_minus_1,
  random = ~1 | view_id,
  #correlation = corAR1(form = ~ time_index | view_id), # time_index = observation order wit
  na.action = na.omit,
  data = data_240m,
  method = "REML"
)

summary(model_lme)
```

```
Linear mixed-effects model fit by REML
  Data: data_240m
       AIC      BIC    logLik
  30707.16 30733.71 -15349.58

Random effects:
 Formula: ~1 | view_id
        (Intercept) Residual
StdDev:    8.169127 3.632139

Fixed effects:  gam_pred_diff ~ gam_fitted_count_t_minus_1
                                Value Std.Error   DF   t-value p-value
(Intercept)                 2.4102401 2.7307281 5644  0.882636  0.3775
gam_fitted_count_t_minus_1 -0.0065568 0.0007759 5644 -8.450404  0.0000
 Correlation:
                           (Intr)
gam_fitted_count_t_minus_1 -0.041

Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-13.87339948  -0.08279981  -0.03131905   0.06800370  52.43190662

Number of Observations: 5654
Number of Groups: 9
```

```
# Diagnostics
plot(model_lme)                        # residuals vs fitted, QQ plot
```



```
acf(resid(model_lme, type="normalized")) # residual autocorrelation check
```

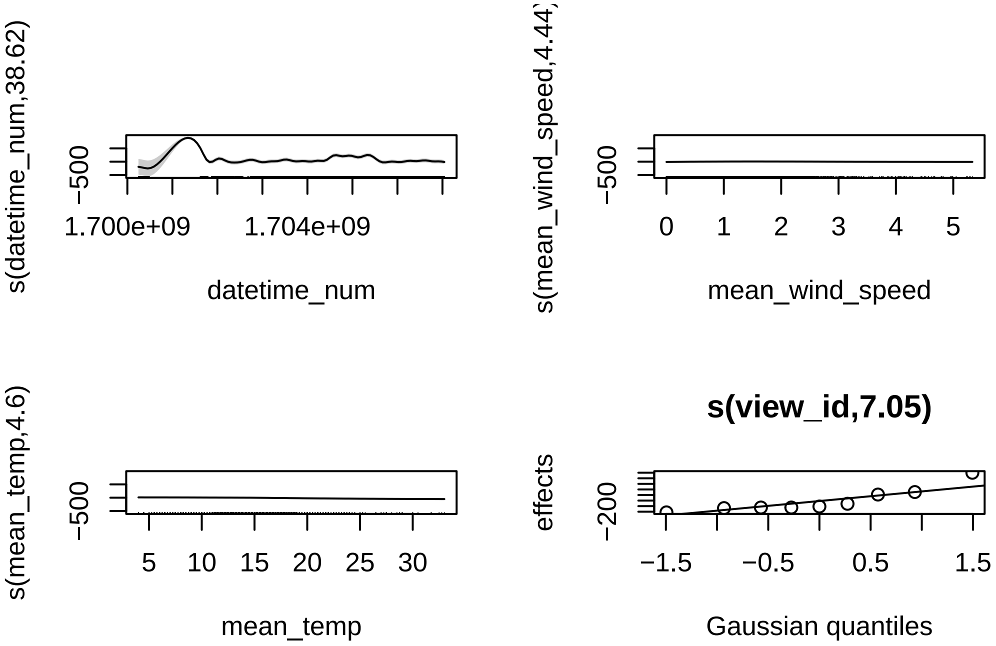


```
library(mgcv)

data_30m$datetime_num <- as.numeric(data_30m$datetime)

gam_model <- gam(
  total_butterflies ~
    s(datetime_num, k = 40) +             # smooth time trend
    s(mean_wind_speed, k = 10) +               # smooth wind effect
    s(mean_temp, k = 10) +                     # smooth temp effect
    sunlight_proportion +                      # linear or smooth
    s(view_id, bs = "re"),                     # random intercepts
  data = data_30m,
  method = "REML"
)

summary(gam_model)
```

```
Family: gaussian
Link function: identity

Formula:
```

```
total_butterflies ~ s(datetime_num, k = 40) + s(mean_wind_speed,
    k = 10) + s(mean_temp, k = 10) + sunlight_proportion + s(view_id,
    bs = "re")

Parametric coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          143.567     78.942   1.819    0.069 .
sunlight_proportion  -33.296      3.014 -11.047   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                       edf Ref.df      F p-value
s(datetime_num)     38.615 38.964 448.02  <2e-16 ***
s(mean_wind_speed)   4.437  5.467  17.36  <2e-16 ***
s(mean_temp)         4.599  5.684  31.20  <2e-16 ***
s(view_id)           7.050  8.000 928.28  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.854   Deviance explained = 85.5%
-REML =  28331  Scale est. = 1367      n = 5601
```

```r
plot(gam_model, pages = 1, shade = TRUE)
```



s(datetime_num,38.62)
−500
1.700e+09
1.704e+09
datetime_num
s(mean_wind_speed,4.44)
−500
0
1
2
3
4
5
mean_wind_speed
s(mean_temp,4.6)
−500
5
10
15
20
25
30
mean_temp
s(view_id,7.05)
effects
−200
−1.5
−0.5
0.5
1.5
Gaussian quantiles