Wind Effects on Butterfly Abundance - GAMM Analysis

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(broom)
library(gratia)
```

```
Attaching package: 'gratia'

The following object is masked from 'package:stringr':

    boundary
```

```r
library(performance)
library(DHARMa)
```

```
This is DHARMa 0.4.7. For overview type '?DHARMa'. For recent changes, type news(package = 'I
```

```r
library(ggeffects)
library(here)
```

```
here() starts at /Users/kylenessen/Documents/Code/masters-analysis
```

```r
library(mgcv) # Load mgcv last to avoid conflicts
```

```
Loading required package: nlme

Attaching package: 'nlme'

The following object is masked from 'package:dplyr':

    collapse

This is mgcv 1.9-3. For overview type 'help("mgcv-package")'.
```

```r
theme_set(theme_minimal())
```

```r
# Load the prepared data
# Assuming df is already loaded with the structure described
# If not, load it here:
df <- read_csv(here("data", "analysis_dataset_final.csv"))
```

```
Rows: 2098 Columns: 16
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (4): deployment_id, image_filename, day_id, Observer
dbl  (10): total_butterflies, butterflies_direct_sun, temperature, view_id, ...
lgl   (1): AR_start
dttm  (1): timestamp

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Check data structure
glimpse(df)
```

```
Rows: 2,098
Columns: 16
$ deployment_id          <chr> "SC1", "SC1", "SC1", "SC1", "SC1", "SC1", "SC1"~
$ image_filename         <chr> "SC1_20231118063002.JPG", "SC1_20231118070001.J~
$ total_butterflies      <dbl> 56, 33, 44, 55, 51, 42, 48, 46, 46, 56, 40, 47,~
$ butterflies_direct_sun <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ timestamp              <dttm> 2023-11-18 06:30:02, 2023-11-18 07:00:01, 2023~
$ day_id                 <chr> "SC1-20231118", "SC1-20231118", "SC1-20231118",~
$ AR_start               <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,~
$ temperature            <dbl> 16, 17, 16, 17, 16, 17, 17, 17, 17, 18, 18, 17,~
$ Observer               <chr> "Skyler", "Skyler", "Skyler", "Skyler", "Skyler~
$ view_id                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ wind_mean              <dbl> 2.34333333, 2.34333333, 2.61333333, 2.44666667,~
$ wind_max_gust          <dbl> 4.7, 4.7, 5.1, 4.7, 4.1, 4.1, 4.3, 4.3, 4.3, 4.~
$ wind_sd                <dbl> 0.3738738, 0.3738738, 0.3093189, 0.3919301, 0.3~
$ gust_differential_mean <dbl> 0.95000000, 0.95000000, 0.96000000, 0.90333333,~
$ cumulative_wind        <dbl> 70.3, 70.3, 78.4, 73.4, 66.3, 61.3, 63.8, 74.7,~
$ time_above_threshold   <dbl> 24, 24, 29, 26, 20, 14, 18, 26, 19, 28, 23, 25,~
```

```r
# Check for missing values
summary(df)
```

```
 deployment_id      image_filename     total_butterflies butterflies_direct_sun
 Length:2098        Length:2098        Min.   :  0.0     Min.   :  0.000       
 Class :character   Class :character   1st Qu.:  5.0     1st Qu.:  0.000       
 Mode  :character   Mode  :character   Median : 26.0     Median :  0.000       
                                       Mean   : 72.5     Mean   :  4.662       
                                       3rd Qu.:105.8     3rd Qu.:  1.000       
                                       Max.   :770.0     Max.   :295.000       
   timestamp                          day_id           AR_start      
 Min.   :2023-11-18 06:30:01.00   Length:2098        Mode :logical  
 1st Qu.:2023-12-21 16:57:31.75   Class :character   FALSE:1999     
 Median :2024-01-03 15:05:01.00   Mode  :character   TRUE :99       
 Mean   :2024-01-02 22:23:31.49                                     
 3rd Qu.:2024-01-16 15:59:01.75                                     
 Max.   :2024-02-03 17:30:01.00                                     
  temperature      Observer            view_id        wind_mean      
 Min.   : 3.00   Length:2098        Min.   :1.000   Min.   :0.00000  
 1st Qu.:12.00   Class :character   1st Qu.:2.000   1st Qu.:0.05333  
 Median :14.00   Mode  :character   Median :2.000   Median :0.64333  
 Mean   :14.62                      Mean   :2.967   Mean   :0.74296  
 3rd Qu.:17.00                      3rd Qu.:4.000   3rd Qu.:1.09583  
```

```
 Max.   :30.00                      Max.   :5.000   Max.   :4.95000  
 wind_max_gust       wind_sd        gust_differential_mean cumulative_wind 
 Min.   : 0.000   Min.   :0.00000   Min.   :0.00000        Min.   :  0.00  
 1st Qu.: 0.700   1st Qu.:0.05986   1st Qu.:0.04333        1st Qu.:  1.60  
 Median : 1.300   Median :0.17162   Median :0.23667        Median : 19.25  
 Mean   : 1.635   Mean   :0.19289   Mean   :0.29865        Mean   : 22.26  
 3rd Qu.: 2.200   3rd Qu.:0.28679   3rd Qu.:0.40000        3rd Qu.: 32.88  
 Max.   :12.800   Max.   :1.37730   Max.   :3.42667        Max.   :148.50  
 time_above_threshold
 Min.   : 0.000      
 1st Qu.: 0.000      
 Median : 0.000      
 Mean   : 2.131      
 3rd Qu.: 0.000      
 Max.   :30.000      
```

```r
# Check correlations among predictors
cor_matrix <- df %>%
  select(temperature, wind_mean, butterflies_direct_sun) %>%
  cor(use = "complete.obs")
print(cor_matrix)
```

```
                       temperature    wind_mean butterflies_direct_sun
temperature             1.00000000 -0.182469624            0.046314432
wind_mean              -0.18246962  1.000000000           -0.001783819
butterflies_direct_sun  0.04631443 -0.001783819            1.000000000
```

```r
# Load the data
df_full <- df

# Prepare data for modeling
# Select variables, ensure correct types, create AR.start, and handle missing values
df_model <- df_full %>%
  select(total_butterflies, temperature, wind_mean, butterflies_direct_sun, day_id, Observer)
  mutate(
    day_id = as.factor(day_id),
    Observer = as.factor(Observer)
  ) %>%
  group_by(day_id) %>%
  mutate(AR_start = row_number() == 1) %>%
  ungroup() %>%
  na.omit()
```

```
# Define the models
k_val <- 28

# Note: After changing AR_start to factor, all models need to be rerun
# to avoid errors with ggpredict()

# Model 1: Null Model
m_null <- bam(total_butterflies ~ s(day_id, bs = "re") + s(Observer, bs = "re"),
              data = df_model,
              family = tw(),
              method = "fREML",
              AR.start = df_model$AR_start)

# Model 2: Single Predictor Models
m_temp <- bam(total_butterflies ~ s(temperature, k = k_val) + s(day_id, bs = "re") + s(Obser
              data = df_model,
              family = tw(),
              method = "fREML",
              discrete = TRUE,
              AR.start = df_model$AR_start)

m_wind <- bam(total_butterflies ~ s(wind_mean, k = k_val) + s(day_id, bs = "re") + s(Observe
              data = df_model,
              family = tw(),
              method = "fREML",
              discrete = TRUE,
              AR.start = df_model$AR_start)

m_sun <- bam(total_butterflies ~ s(butterflies_direct_sun, k = k_val) + s(day_id, bs = "re")
             data = df_model,
             family = tw(),
             method = "fREML",
             discrete = TRUE,
             AR.start = df_model$AR_start)


# Model 3: Additive Model
m_additive <- bam(total_butterflies ~ s(temperature, k = k_val) +
                                      s(wind_mean, k = k_val) +
                                      s(butterflies_direct_sun, k = k_val) +
                                      s(day_id, bs = "re") +
                                      s(Observer, bs = "re"),
```

```r
                     data = df_model,
                     family = tw(),
                     method = "fREML",
                     discrete = TRUE,
                     AR.start = df_model$AR_start)

# Model 4: Additive + Interaction Models
m_int_temp_wind <- bam(total_butterflies ~ s(temperature, k = k_val) +
                                          s(wind_mean, k = k_val) +
                                          s(butterflies_direct_sun, k = k_val) +
                                          ti(temperature, wind_mean, k = 10) +
                                          s(day_id, bs = "re") +
                                          s(Observer, bs = "re"),
                          data = df_model,
                          family = tw(),
                          method = "fREML",
                          discrete = TRUE,
                          AR.start = df_model$AR_start)

# We can now inspect these models.
```

```r
# Model Diagnostics

# 1. Check for concurvity in the more complex models
# Concurvity is the GAM equivalent of multicollinearity.
# High values (close to 1) can be problematic.
concurvity(m_additive, full = FALSE)
```

```
$worst
                              para s(temperature) s(wind_mean)
para                     1.0000000      0.8487645    0.2941558
s(temperature)           0.8487645      1.0000000    0.2860057
s(wind_mean)             0.2941558      0.2860057    1.0000000
s(butterflies_direct_sun) 0.9217986      0.7758683    0.2835593
s(day_id)                1.0000000      0.8989502    0.7511680
s(Observer)              1.0000000      0.8547673    0.3980913
                          s(butterflies_direct_sun) s(day_id) s(Observer)
para                                      0.9217986 1.0000000   1.0000000
s(temperature)                            0.7758683 0.8989502   0.8547673
s(wind_mean)                              0.2835593 0.7511680   0.3980913
s(butterflies_direct_sun)                 1.0000000 0.9419389   0.9258392
s(day_id)                                 0.9419389 1.0000000   1.0000000
```

```
s(Observer)                                 0.9258392 1.0000000   1.0000000

$observed
                                para s(temperature) s(wind_mean)
para                       1.0000000      0.6677120    0.1353571
s(temperature)             0.8487645      1.0000000    0.1539680
s(wind_mean)               0.2941558      0.1645570    1.0000000
s(butterflies_direct_sun)  0.9217986      0.6313431    0.1440890
s(day_id)                  1.0000000      0.8202542    0.6340910
s(Observer)                1.0000000      0.7083196    0.1934230
                           s(butterflies_direct_sun)    s(day_id) s(Observer)
para                                      0.02088505 0.0000293885  0.01396555
s(temperature)                            0.02737366 0.0391515119  0.12345663
s(wind_mean)                              0.01571380 0.0455859593  0.11997489
s(butterflies_direct_sun)                 1.00000000 0.0124749326  0.02987560
s(day_id)                                 0.13251302 1.0000000000  1.00000000
s(Observer)                               0.02796066 0.0222393877  1.00000000

$estimate
                                para s(temperature) s(wind_mean)
para                       1.0000000     0.31390969   0.09596274
s(temperature)             0.8487645     1.00000000   0.12405118
s(wind_mean)               0.2941558     0.08949011   1.00000000
s(butterflies_direct_sun)  0.9217986     0.32973799   0.10324313
s(day_id)                  1.0000000     0.55819396   0.54702541
s(Observer)                1.0000000     0.37620567   0.18426220
                           s(butterflies_direct_sun)  s(day_id) s(Observer)
para                                       0.8610479 0.01011404   0.2564329
s(temperature)                             0.7245523 0.03333273   0.2674927
s(wind_mean)                               0.2521644 0.03641063   0.1522140
s(butterflies_direct_sun)                  1.0000000 0.02555387   0.2607489
s(day_id)                                  0.8886026 1.00000000   1.0000000
s(Observer)                                0.8663247 0.04055501   1.0000000
```

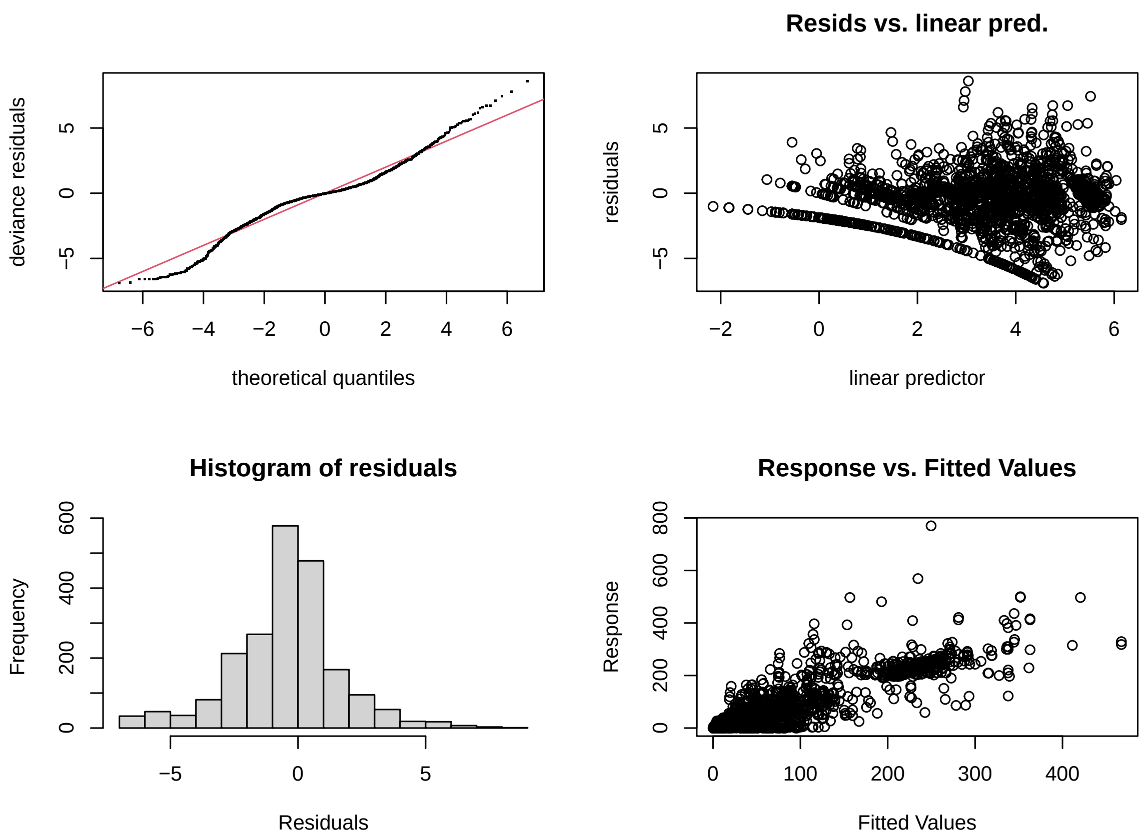
```r
concurvity(m_int_temp_wind, full = FALSE)
```

```
$worst
                                para s(temperature) s(wind_mean)
para                       1.0000000      0.8487645    0.2941558
s(temperature)             0.8487645      1.0000000    0.2860057
s(wind_mean)               0.2941558      0.2860057    1.0000000
s(butterflies_direct_sun)  0.9217986      0.7758683    0.2835593
```

```
ti(temperature,wind_mean) 0.4873051      1.0000000    1.0000000
s(day_id)                 1.0000000      0.8989502    0.7511680
s(Observer)               1.0000000      0.8547673    0.3980913
                          s(butterflies_direct_sun) ti(temperature,wind_mean)
para                                      0.9217986                 0.4873051
s(temperature)                            0.7758683                 1.0000000
s(wind_mean)                              0.2835593                 1.0000000
s(butterflies_direct_sun)                 1.0000000                 0.4693306
ti(temperature,wind_mean)                 0.4693306                 1.0000000
s(day_id)                                 0.9419389                 0.7540740
s(Observer)                               0.9258392                 0.5479609
                          s(day_id) s(Observer)
para                      1.0000000   1.0000000
s(temperature)            0.8989502   0.8547673
s(wind_mean)              0.7511680   0.3980913
s(butterflies_direct_sun) 0.9419389   0.9258392
ti(temperature,wind_mean) 0.7540740   0.5479609
s(day_id)                 1.0000000   1.0000000
s(Observer)               1.0000000   1.0000000

$observed
                               para s(temperature) s(wind_mean)
para                      1.0000000      0.6920948    0.1608985
s(temperature)            0.8487645      1.0000000    0.1834593
s(wind_mean)              0.2941558      0.1742777    1.0000000
s(butterflies_direct_sun) 0.9217986      0.6503154    0.1657409
ti(temperature,wind_mean) 0.4873051      0.4434042    0.9747309
s(day_id)                 1.0000000      0.8323277    0.7057801
s(Observer)               1.0000000      0.7281089    0.2914102
                          s(butterflies_direct_sun) ti(temperature,wind_mean)
para                                     0.01469171               0.001428778
s(temperature)                           0.02260964               0.231990596
s(wind_mean)                             0.01405912               0.403981460
s(butterflies_direct_sun)                1.00000000               0.030315977
ti(temperature,wind_mean)                0.03407875               1.000000000
s(day_id)                                0.13007452               0.330746451
s(Observer)                              0.02211465               0.017672925
                             s(day_id) s(Observer)
para                      2.912206e-05  0.01400933
s(temperature)            4.073523e-02  0.12337460
s(wind_mean)              4.622096e-02  0.11966413
s(butterflies_direct_sun) 1.258683e-02  0.02995219
ti(temperature,wind_mean) 9.072355e-02  0.22481951
```

```
s(day_id)                 1.000000e+00  1.00000000
s(Observer)               2.264774e-02  1.00000000

$estimate
                               para s(temperature) s(wind_mean)
para                      1.0000000     0.31390969   0.09596274
s(temperature)            0.8487645     1.00000000   0.12405118
s(wind_mean)              0.2941558     0.08949011   1.00000000
s(butterflies_direct_sun) 0.9217986     0.32973799   0.10324313
ti(temperature,wind_mean) 0.4873051     0.51550632   0.96843462
s(day_id)                 1.0000000     0.55819396   0.54702541
s(Observer)               1.0000000     0.37620567   0.18426220
                          s(butterflies_direct_sun) ti(temperature,wind_mean)
para                                      0.8610479               0.009949196
s(temperature)                            0.7245523               0.098683465
s(wind_mean)                              0.2521644               0.199532461
s(butterflies_direct_sun)                 1.0000000               0.024698667
ti(temperature,wind_mean)                 0.4245769               1.000000000
s(day_id)                                 0.8886026               0.166621345
s(Observer)                               0.8663247               0.021069432
                           s(day_id) s(Observer)
para                      0.01011404   0.2564329
s(temperature)            0.03333273   0.2674927
s(wind_mean)              0.03641063   0.1522140
s(butterflies_direct_sun) 0.02555387   0.2607489
ti(temperature,wind_mean) 0.09493258   0.2504391
s(day_id)                 1.00000000   1.0000000
s(Observer)               0.04055501   1.0000000
```

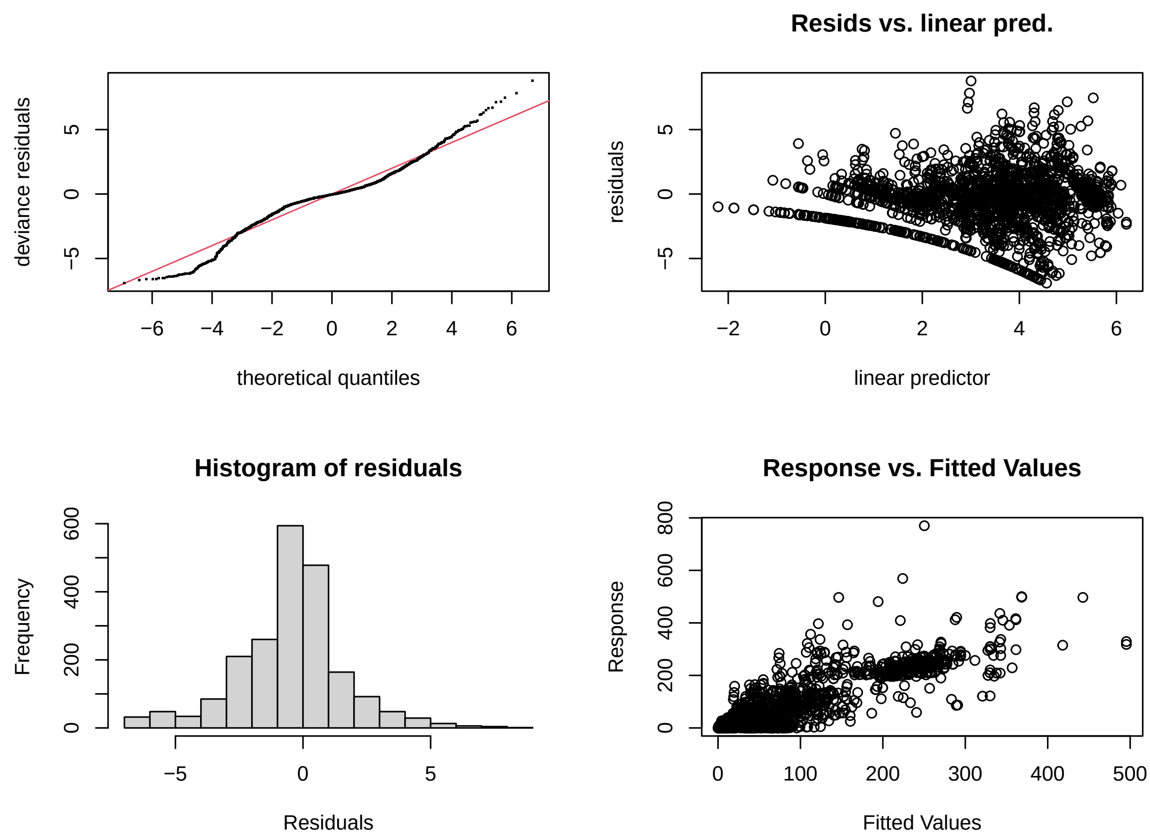
```
# 2. Use gam.check() for standard diagnostics - Additive Model
# This provides k-checks (are basis dimensions adequate?) and residual plots.
gam.check(m_additive)
```




```
Method: fREML   Optimizer: perf chol
$grad
[1] 5.424758e-07 3.475534e-08 1.159222e-07 3.784388e-07 1.885131e-09

$hess
             [,1]        [,2]         [,3]        [,4]         [,5]
[1,]  1.426177749 -0.03251365 -0.035615178  0.07525999  0.005267917
[2,] -0.032513645  0.39563695  0.016493337  0.09199756  0.002004900
[3,] -0.035615178  0.01649334  0.143833969 -0.03953295 -0.000258868
[4,]  0.075259990  0.09199756 -0.039532950 44.28564175  0.248071556
[5,]  0.005267917  0.00200490 -0.000258868  0.24807156  0.118863409

Model rank =  185 / 185

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.
```

```
                              k'    edf k-index p-value
s(temperature)            27.000 11.744    0.75  <2e-16 ***
s(wind_mean)              27.000  2.555    0.76  <2e-16 ***
s(butterflies_direct_sun) 27.000  1.916    0.40  <2e-16 ***
s(day_id)                 99.000 95.323      NA      NA
s(Observer)                4.000  0.823      NA      NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# gam.check() for interaction model
gam.check(m_int_temp_wind)
```




```
Method: fREML   Optimizer: perf chol
$grad
```

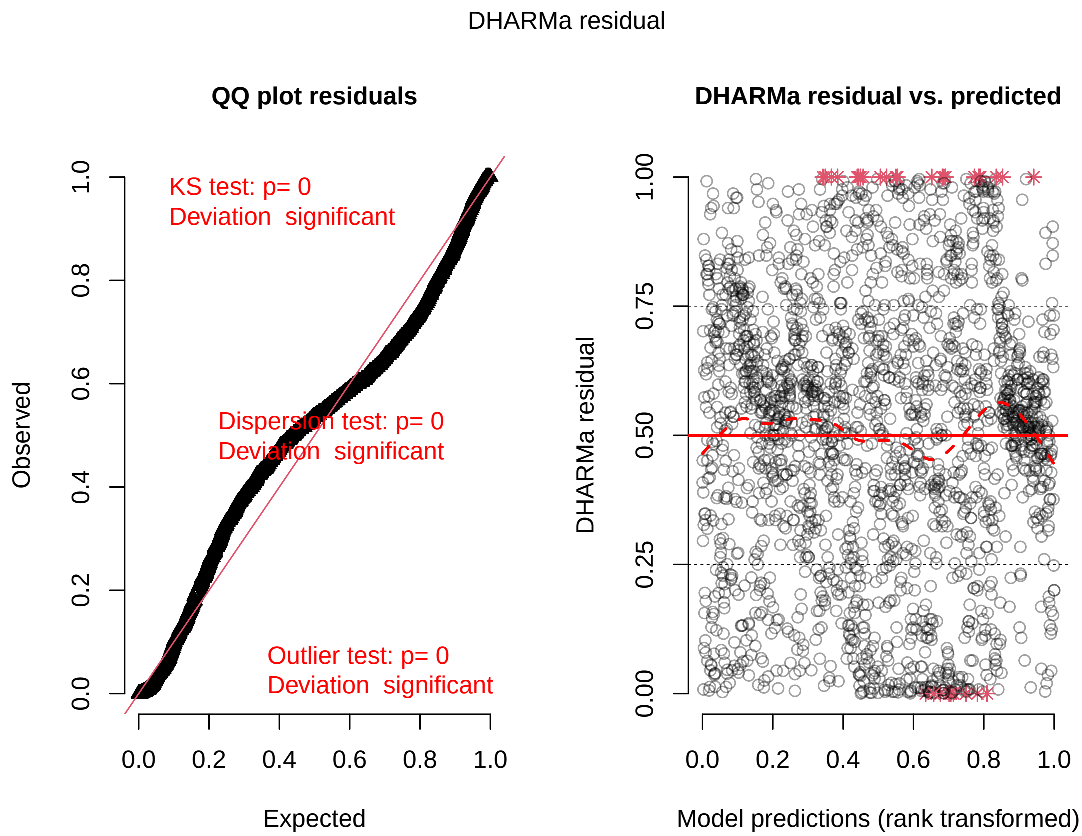
```
[1]  1.993727e-07 -1.584036e-05  5.544573e-08  4.436391e-08  4.164144e-08
[6] -1.488562e-07 -2.027157e-09

$hess
              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.437138e+00 -3.361283e-06 -3.625087e-02  6.778250e-02 -2.693136e-02
[2,] -3.361283e-06  1.584708e-05  3.026674e-06  9.434466e-06  2.784738e-05
[3,] -3.625087e-02  3.026674e-06  1.443905e-01 -1.462153e-03  1.068614e-02
[4,]  6.778250e-02  9.434466e-06 -1.462153e-03  3.528525e-01  3.411160e-02
[5,] -2.693136e-02  2.784738e-05  1.068614e-02  3.411160e-02  2.637222e-01
[6,]  6.656769e-02  1.579149e-05 -4.151280e-02  5.760130e-02  8.927296e-02
[7,]  4.873750e-03  5.482175e-07 -1.126457e-04  2.789970e-03  2.026705e-03
              [,6]          [,7]
[1,]  6.656769e-02  4.873750e-03
[2,]  1.579149e-05  5.482175e-07
[3,] -4.151280e-02 -1.126457e-04
[4,]  5.760130e-02  2.789970e-03
[5,]  8.927296e-02  2.026705e-03
[6,]  4.432225e+01  2.420525e-01
[7,]  2.420525e-01  1.077232e-01

Model rank =  266 / 266

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

                              k'    edf k-index p-value
s(temperature)            27.000 11.929    0.76  <2e-16 ***
s(wind_mean)              27.000  1.000    0.77  <2e-16 ***
s(butterflies_direct_sun) 27.000  2.122    0.40  <2e-16 ***
ti(temperature,wind_mean) 81.000  5.950    0.90  <2e-16 ***
s(day_id)                 99.000 95.366      NA      NA
s(Observer)                4.000  0.784      NA      NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 3. Use DHARMa for more advanced residual diagnostics - Additive Model
# This simulates residuals from the fitted model and compares them to the observed residuals
sim_res_additive <- simulateResiduals(fittedModel = m_additive, n = 250)
```

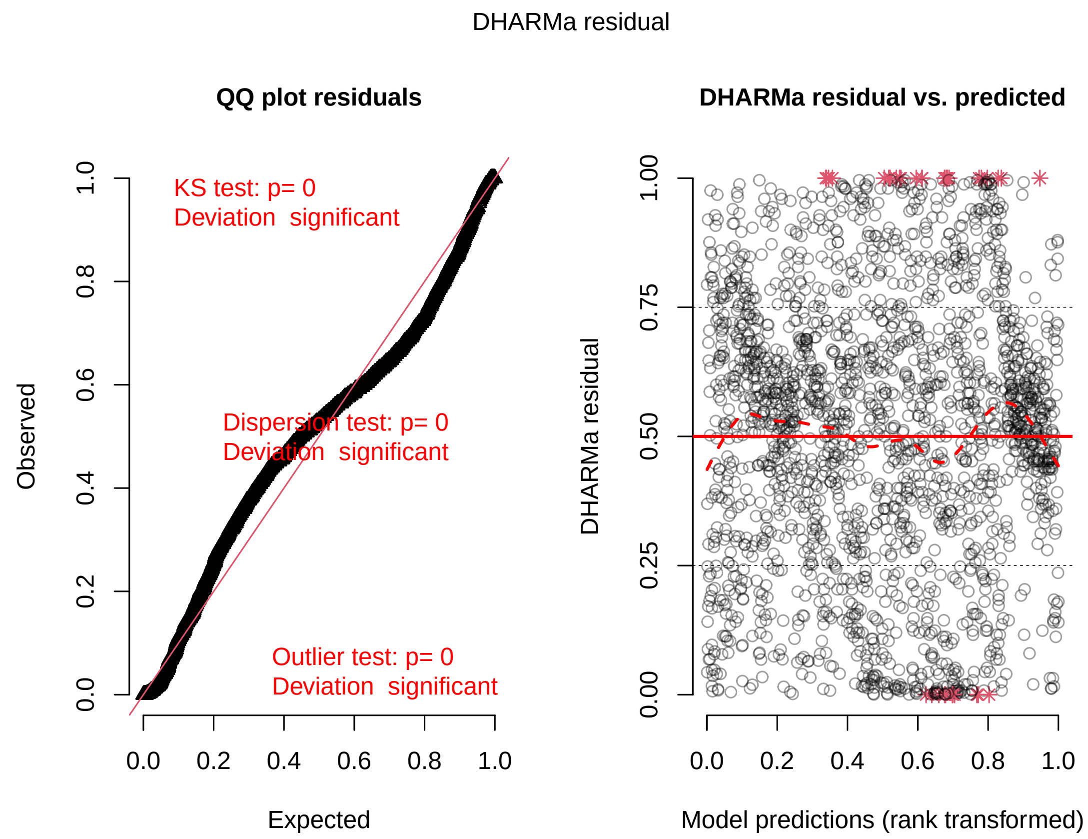
```
Registered S3 methods overwritten by 'mgcViz':
  method       from
```

```
  +.gg          ggplot2
  simulate.gam gratia
```

```r
plot(sim_res_additive)
```




```r
# DHARMa diagnostics for interaction model
sim_res_int <- simulateResiduals(fittedModel = m_int_temp_wind, n = 250)
plot(sim_res_int)
```

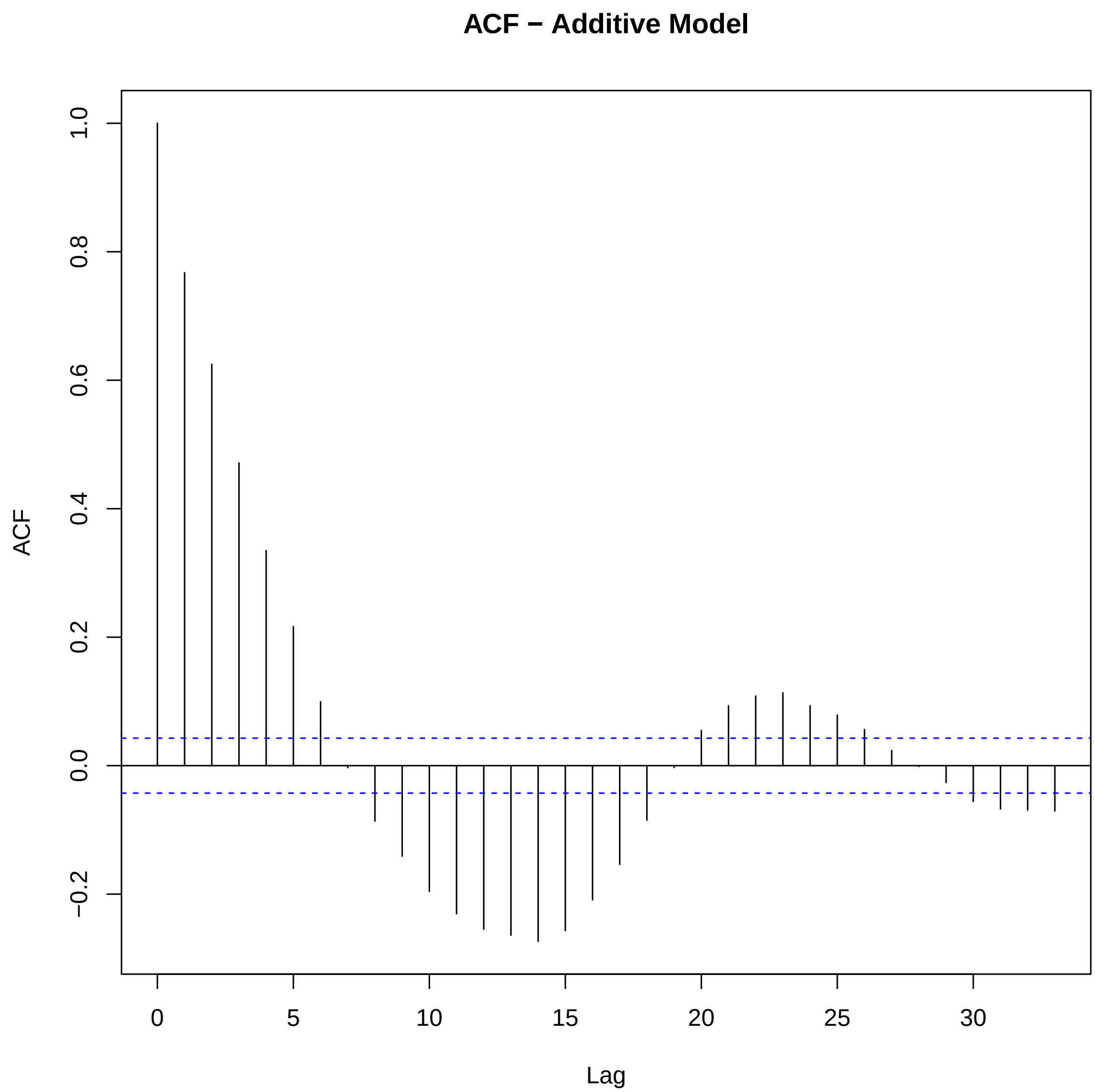



```
# 4. Check for temporal autocorrelation using ACF plots
# This is crucial for time series data to ensure residuals are not autocorrelated

library(gridExtra)
```

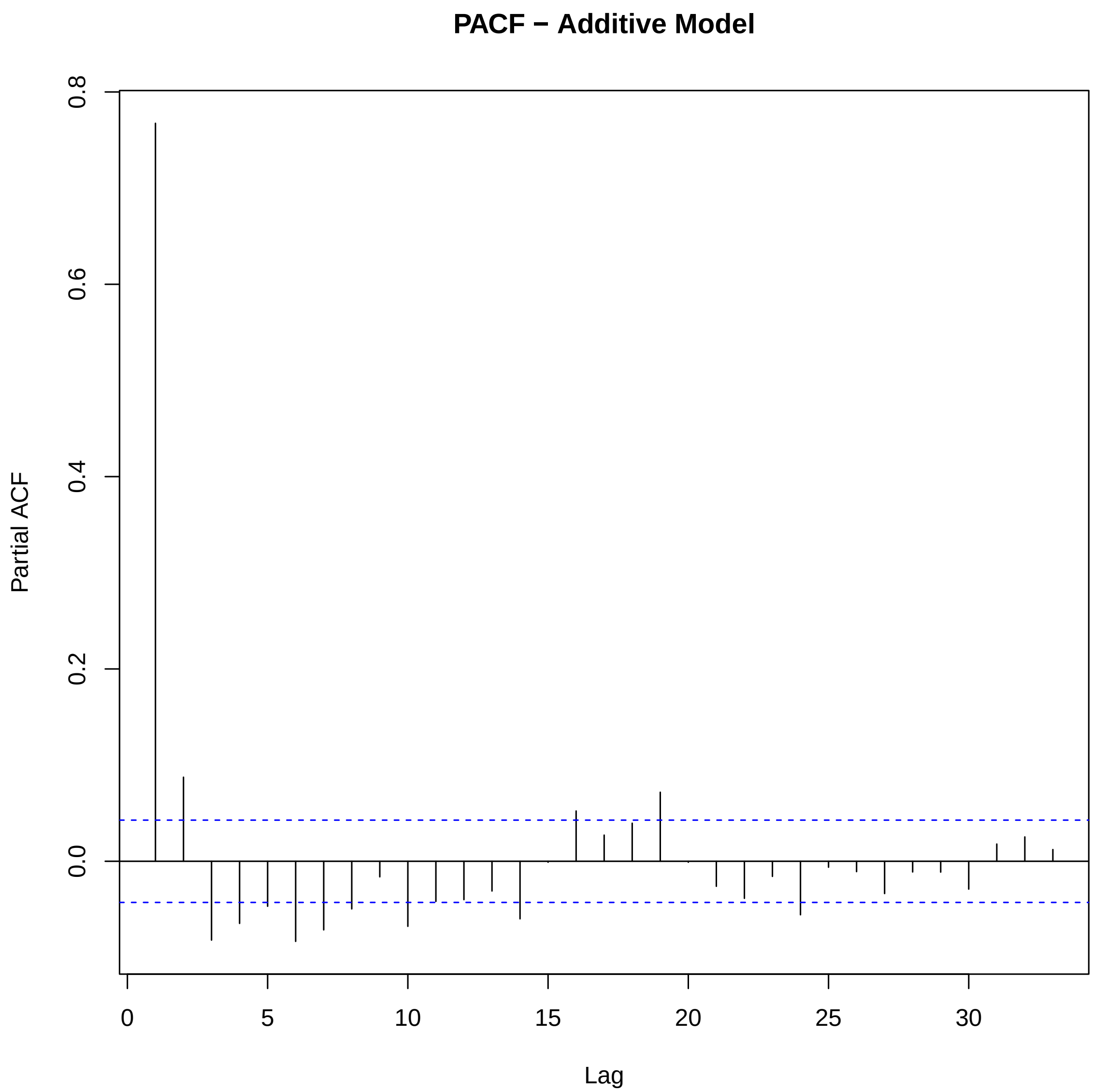
```
Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':

    combine
```

```
library(grid)

# Create ACF plots using base R but arrange with grid
# ACF for additive model
```

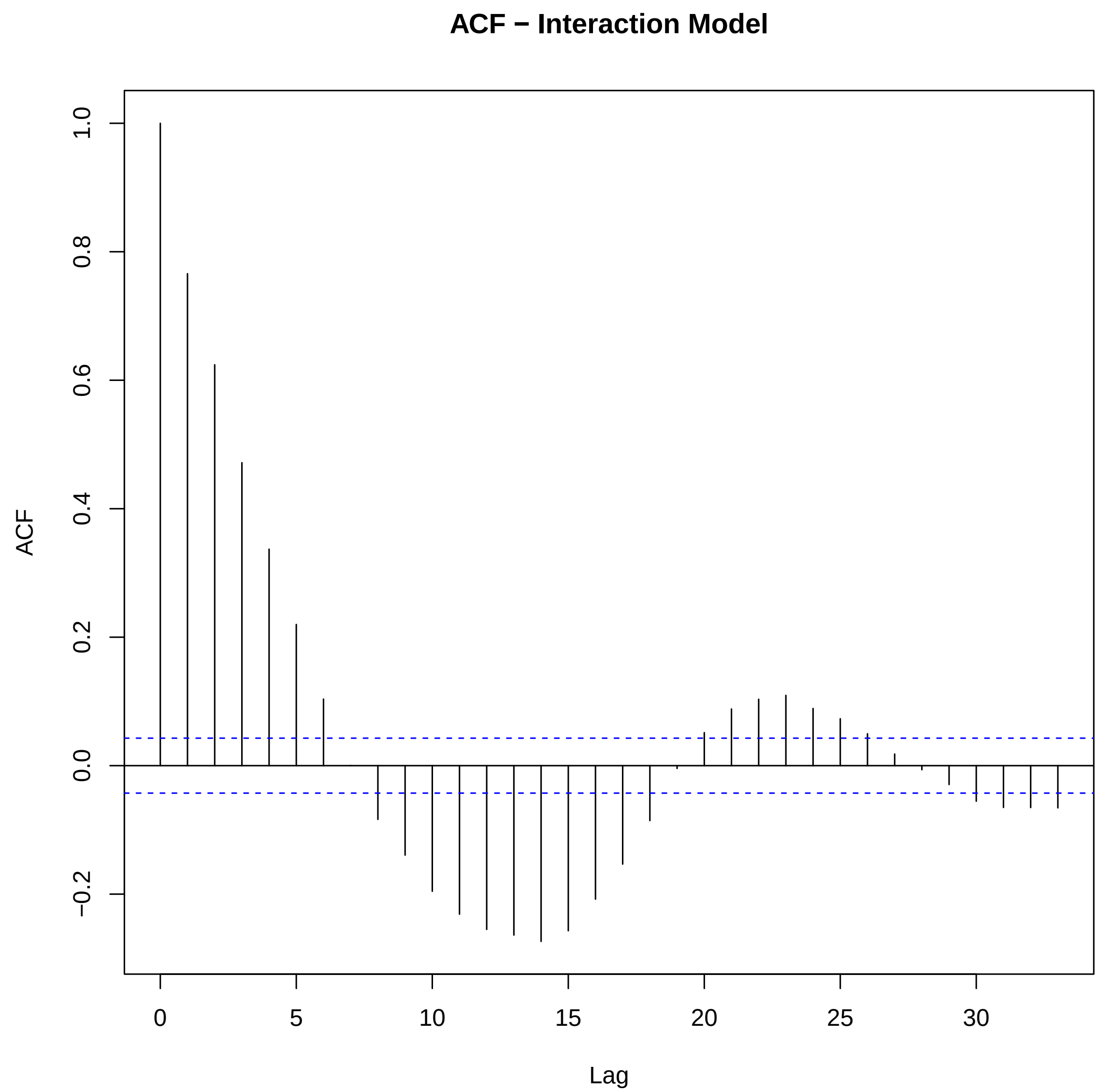
```r
acf_add <- acf(residuals(m_additive), plot = FALSE, main = "ACF - Additive Model")
pacf_add <- pacf(residuals(m_additive), plot = FALSE, main = "PACF - Additive Model")

# ACF for interaction model
acf_int <- acf(residuals(m_int_temp_wind), plot = FALSE, main = "ACF - Interaction Model")
pacf_int <- pacf(residuals(m_int_temp_wind), plot = FALSE, main = "PACF - Interaction Model")

# Plot ACF results
plot(acf_add, main = "ACF - Additive Model")
```



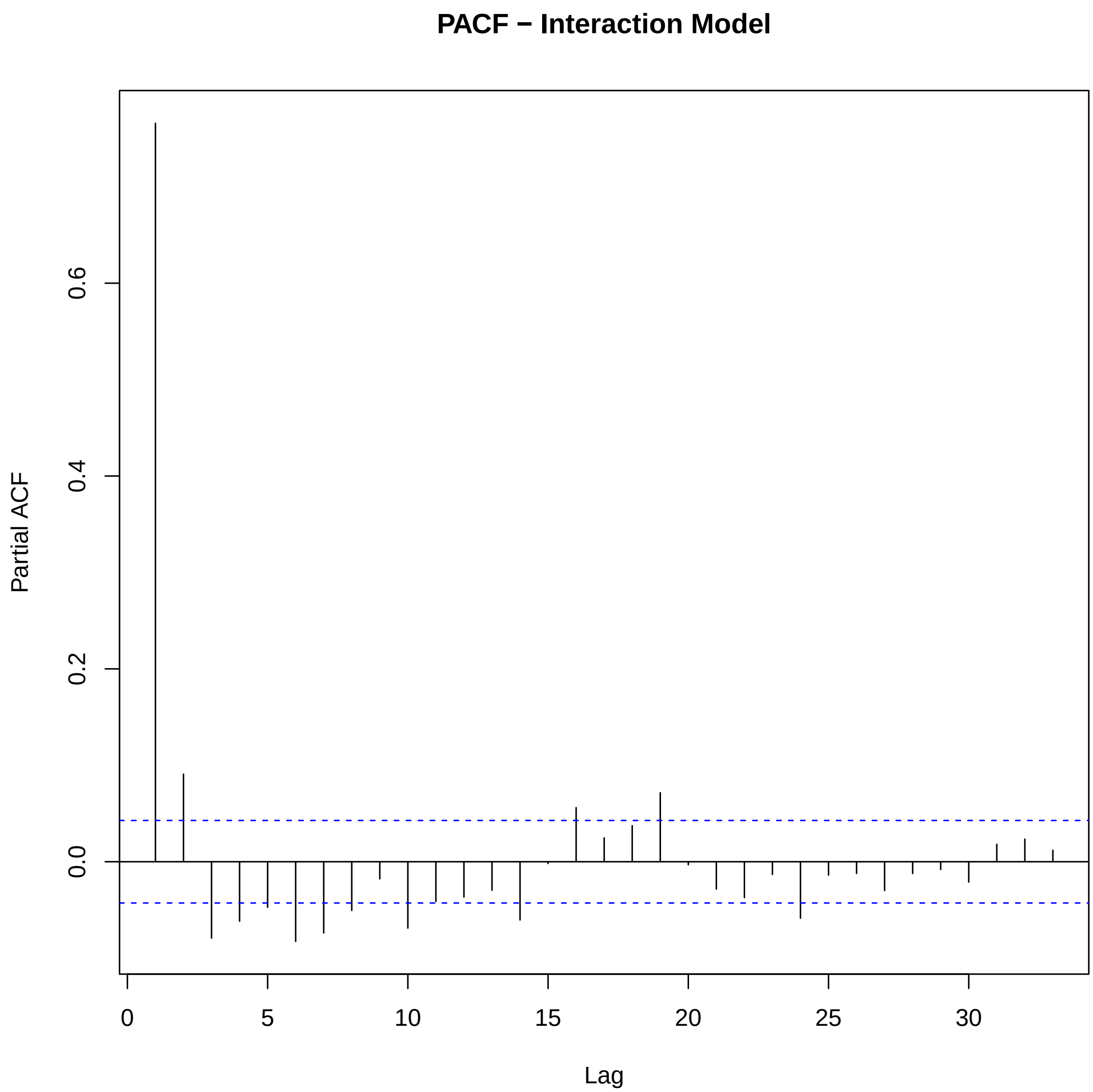

```r
plot(pacf_add, main = "PACF - Additive Model")
```




```
plot(acf_int, main = "ACF - Interaction Model")
```




```r
plot(pacf_int, main = "PACF - Interaction Model")
```

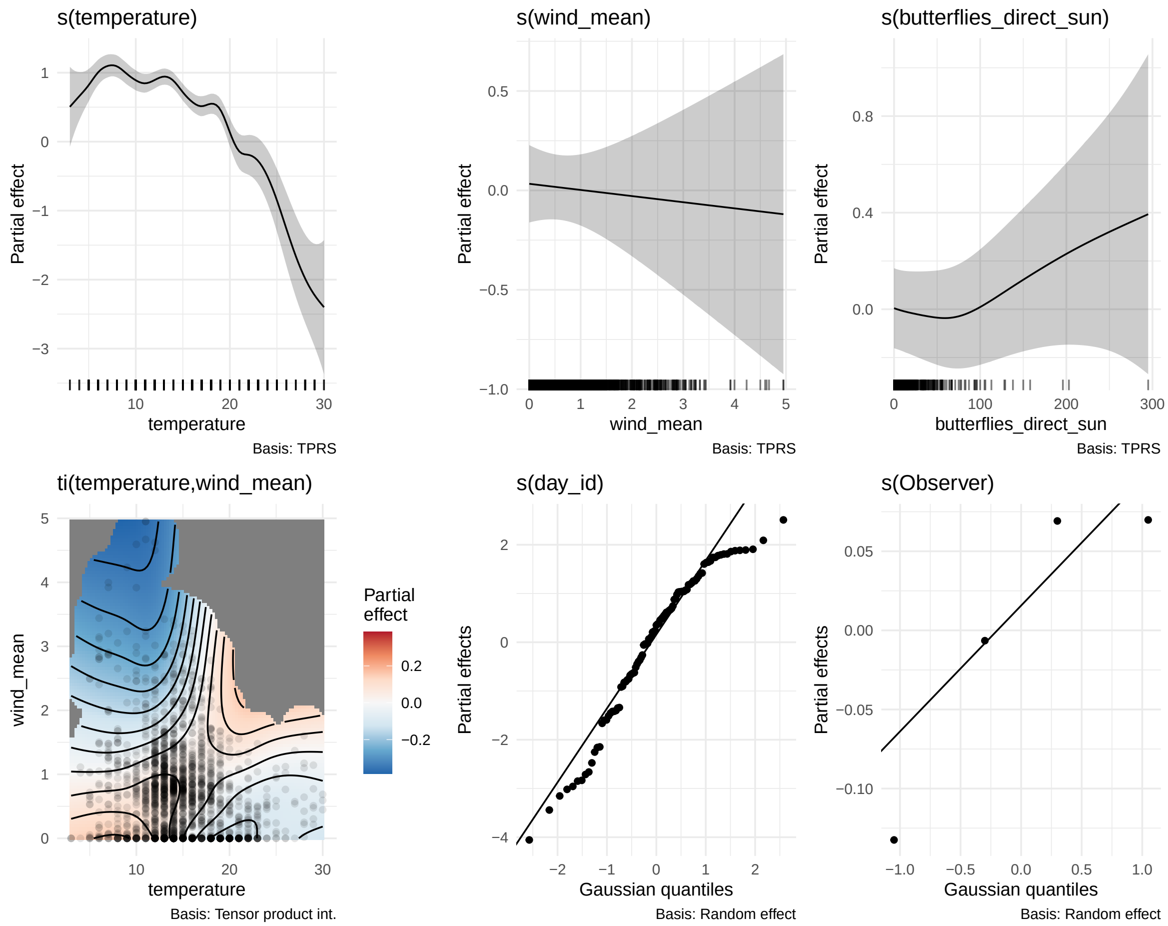



```
# 5. Compare models using AIC
# Create a list of models
model_list <- list(
  null = m_null,
  temp = m_temp,
  wind = m_wind,
  sun = m_sun,
  additive = m_additive,
  interaction = m_int_temp_wind
```

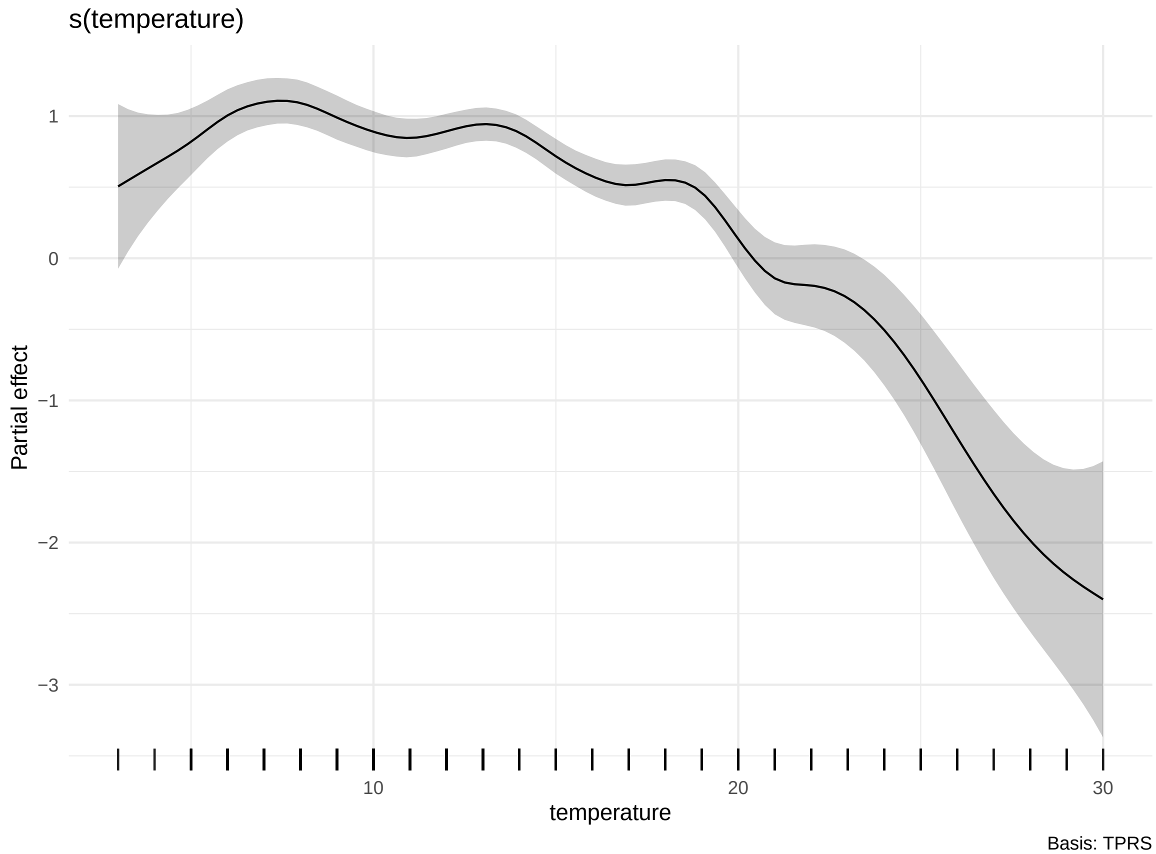
```r
)

# Get AIC for each model
aic_values <- sapply(model_list, AIC)

# Create a summary table
aic_table <- tibble(
  model = names(aic_values),
  AIC = aic_values
) %>%
  arrange(AIC)

print(aic_table)
```

```
# A tibble: 6 x 2
  model          AIC
  <chr>        <dbl>
1 interaction 18094.
2 additive    18098.
3 temp        18108.
4 sun         18378.
5 wind        18438.
6 null        18438.
```

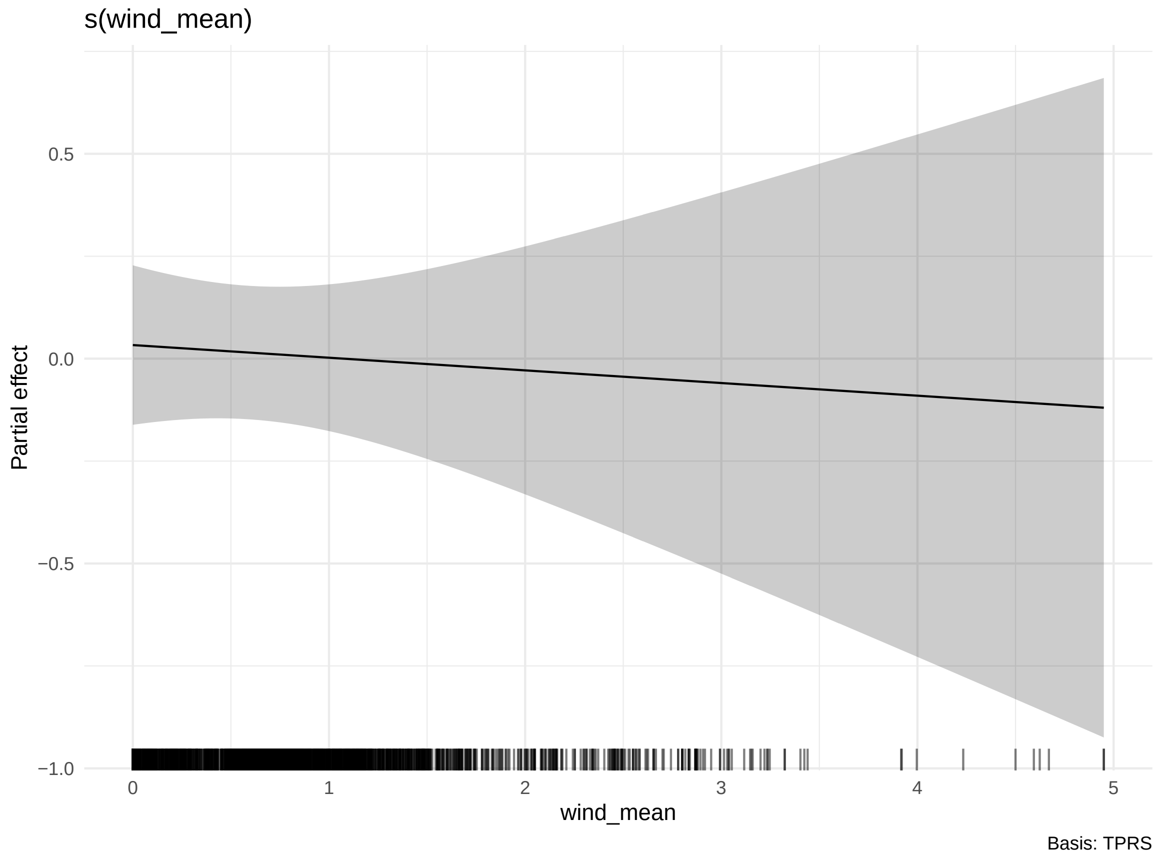
```r
# --- Plotting the Best Model ---

# The model with the lowest AIC is `m_int_temp_wind`.
# Let's visualize the effects from this model.

# 1. Plot all smooth terms (main effects and interactions) together
# `gratia::draw()` is excellent for this. `scales = "free"` allows each plot
# to have its own y-axis scale.
draw(m_int_temp_wind, scales = "free")
```



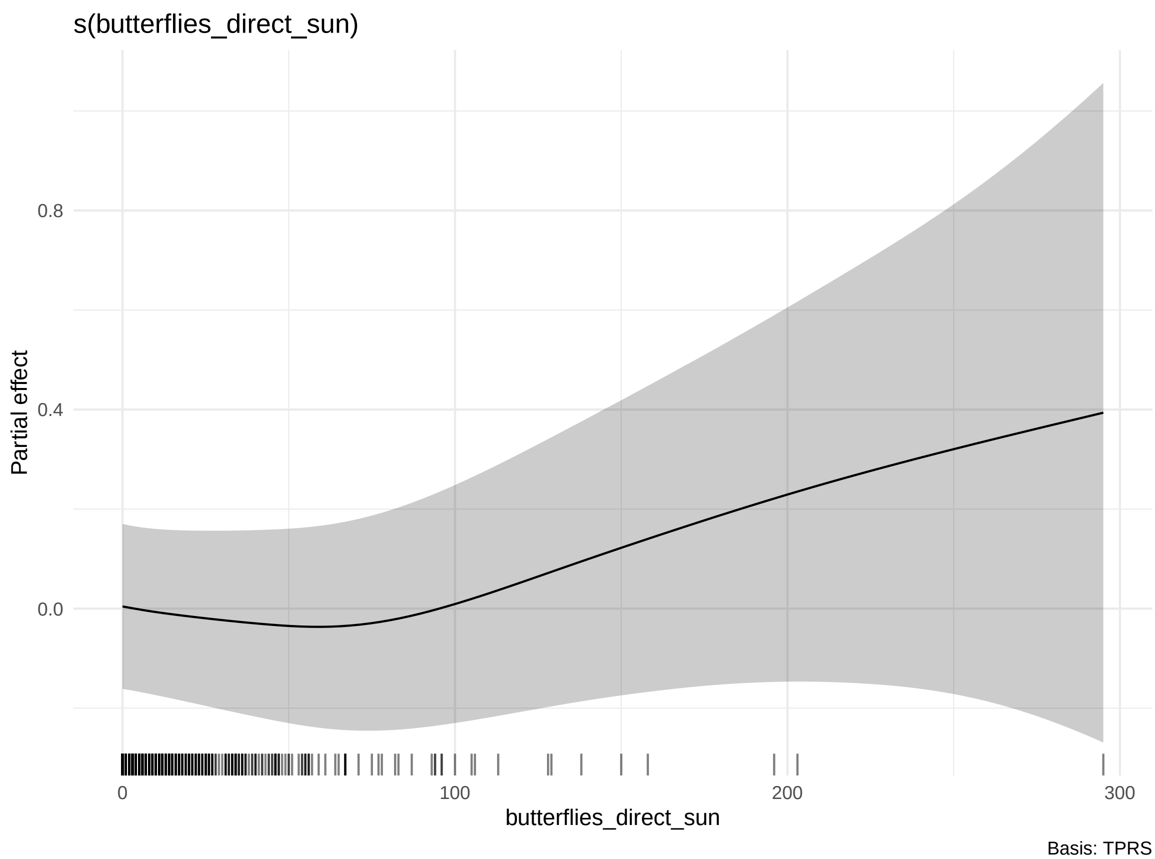

```r
# 2. Plot individual effects for more detail

# Main effect of Temperature
draw(m_int_temp_wind, select = "s(temperature)")
```



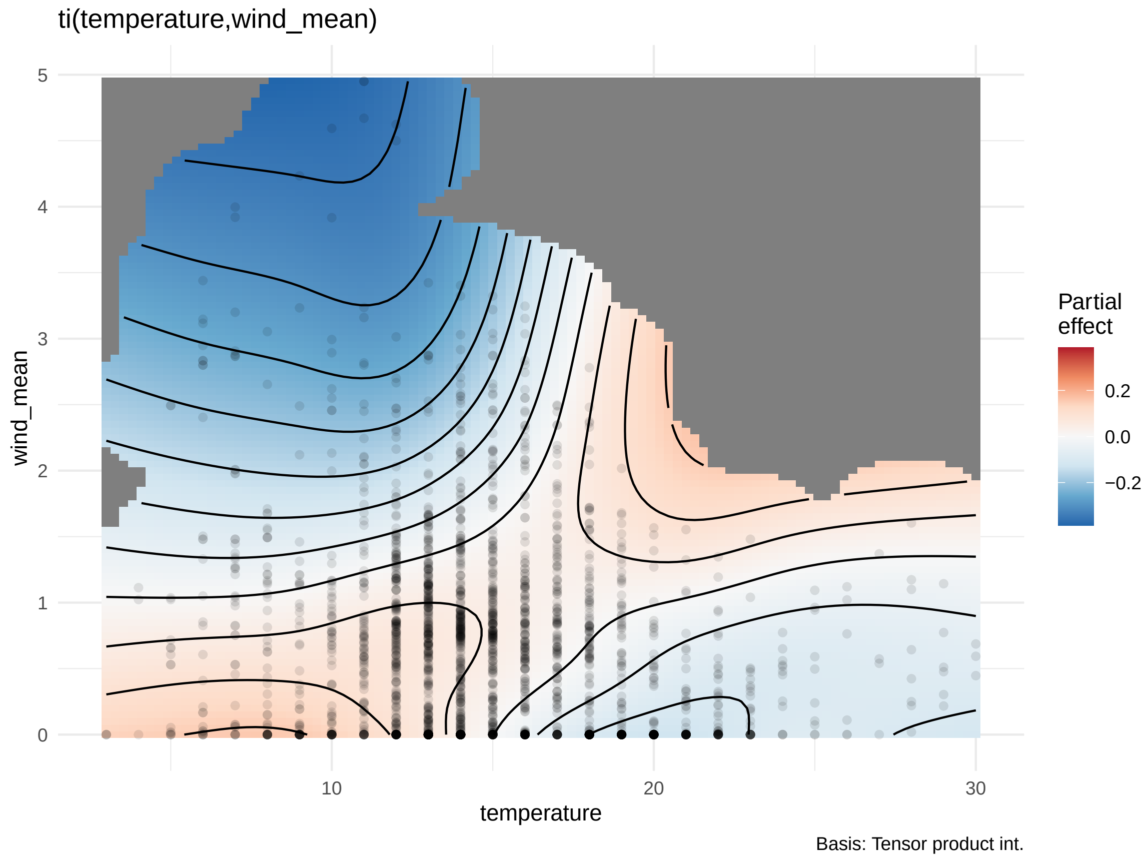

```
# Main effect of Wind
draw(m_int_temp_wind, select = "s(wind_mean)")
```




```
# Main effect of Sun
draw(m_int_temp_wind, select = "s(butterflies_direct_sun)")
```

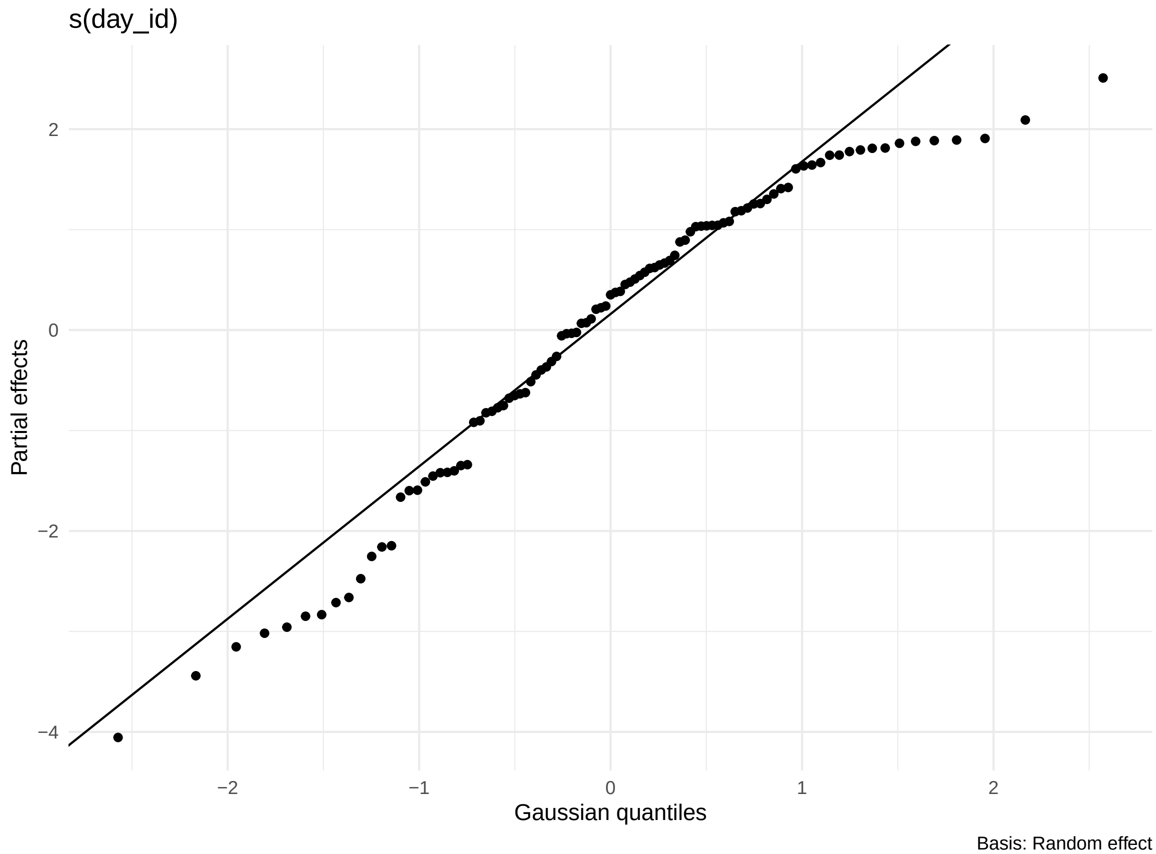



```r
# 3. Visualize the interaction between Temperature and Wind

# Option A: A 2D heatmap of the interaction surface using gratia
draw(m_int_temp_wind, select = "ti(temperature,wind_mean)")
```

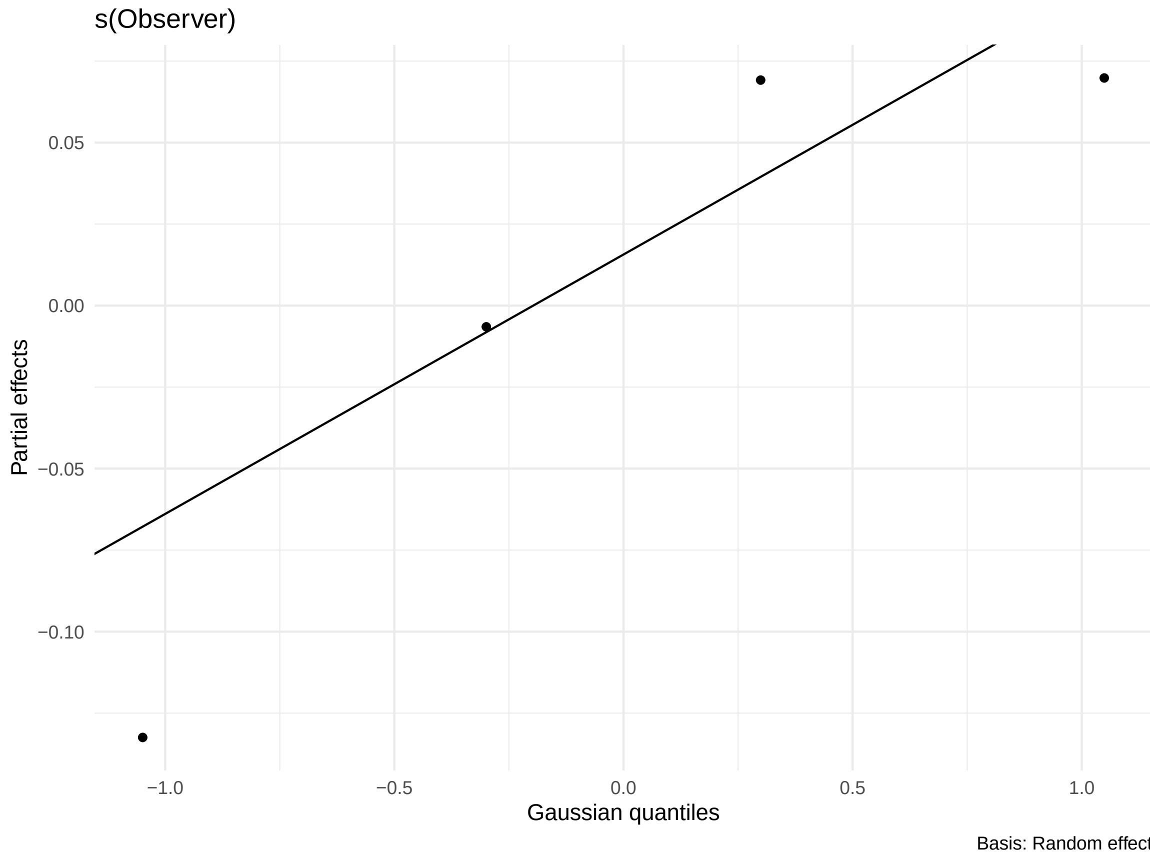



```r
# Option B: Using ggeffects to plot conditional effects.
# This shows the effect of temperature at different levels of wind speed.
# It can sometimes be easier to interpret.
library(ggeffects)
# Note: ggpredict has issues with logical AR_start, so we use typical values
#ggpredict(m_int_temp_wind, terms = c("temperature", "wind_mean"),
#          typical = "mean") %>% plot()
```

```r
# 4. Visualize the random effects
# This can help understand the variation among days and observers.
draw(m_int_temp_wind, select = "s(day_id)")
```




```
draw(m_int_temp_wind, select = "s(Observer)")
```


s(Observer)
0.05
0.00
−0.05
−0.10
Partial effects
−1.0
−0.5
0.0
0.5
1.0
Gaussian quantiles
Basis: Random effect