The Effects of Climate Change on the Breeding Success and Timing of African White-backed Vultures

Mieke Deyzel

0.1 Load Data

```
vultures <- read.csv("data/93_24_cleaned.csv")
```

0.1.1 Clean Breeding Data

```
vultures.clean <- vultures |>
  clean_names() |>
  rename_with(~ gsub("_", ".", .x)) |>
  mutate(
    ringing.date = ringing.date |> as.character() |> str_trim(),
    ringing.date = if_else(ringing.date %in% c("", "NA", "?"), NA_character_,
    ↪  ringing.date),
    ringing.date = ymd(gsub("/", "-", ringing.date)),

    laying.date  = laying.date |> as.character() |> str_trim(),
    laying.date  = if_else(laying.date %in% c("", "NA", "?"), NA_character_,
    ↪  laying.date),
    laying.date  = ymd(gsub("/", "-", laying.date))
  )
```

0.1.2 Explore Data

```
glimpse(vultures.clean)
```

```
Rows: 2,657
Columns: 14
$ ringing.date            <date> 1993-09-21, 1993-09-21, 1993-09-19, 1993-09-1~
$ tree.drn                <chr> "1", "2", "3", "4", "5", "6", "8", "12", "13",~
$ southings               <chr> " 28 38 43.8", "28 38 48.2", "28 38 55.6", "28~
$ eastings                <chr> "24 47 14.5", "24 47 05.1", "24 47 15.4", "24 ~
$ code                    <int> 1, 1, 1, 1, 1, 1, 1, 15, 1, 1, 1, 1, 3, 1, 5, ~
$ ring                    <chr> "G19791", "G19792", "G19781", "G19780", "G1978~
$ colour.rings.r.top.down <chr> "M", "M", "M", "M", "M", "M", "M", "", "M", "M~
$ colour.rings.l.top.down <chr> "B, Y, Y", "B, Y, R", "B, W, G", "B, B, R", "B~
$ mass.g                  <int> 5900, 5000, 3900, 4950, 4950, 3800, 3550, NA, ~
$ wing.length.mm          <int> 475, 425, 260, 310, 290, 275, 210, NA, 245, 29~
$ age.at.ringing.days     <int> 89, 80, 57, 64, 61, 59, 49, NA, 55, 61, 5, 92,~
$ value.at.ringing        <int> 34233, 34233, 34231, 34231, 34231, 34231, 3423~
$ hatching.date           <chr> "1993/06/24", "1993/07/03", "1993/07/24", "199~
$ laying.date             <date> 1993-04-29, 1993-05-08, 1993-05-29, 1993-05-2~
```

```r
summary(vultures.clean)
```

```
  ringing.date          tree.drn           southings            eastings        
 Min.   :1993-09-19   Length:2657        Length:2657        Length:2657       
 1st Qu.:2004-06-06   Class :character   Class :character   Class :character  
 Median :2012-10-13   Mode  :character   Mode  :character   Mode  :character  
 Mean   :2011-08-31                                                           
 3rd Qu.:2019-08-21                                                           
 Max.   :2024-10-18                                                           
 NA's   :32                                                                   
      code             ring           colour.rings.r.top.down
 Min.   :  1.000   Length:2657        Length:2657            
 1st Qu.:  1.000   Class :character   Class :character       
 Median :  1.000   Mode  :character   Mode  :character       
 Mean   :  5.001                                             
 3rd Qu.:  6.000                                             
 Max.   :138.000                                             
 NA's   :4                                                   
 colour.rings.l.top.down     mass.g     wing.length.mm  age.at.ringing.days
 Length:2657             Min.   :   2   Min.   : 25.0   Min.   :  5.00     
 Class :character        1st Qu.:4200   1st Qu.:320.0   1st Qu.: 65.00     
 Mode  :character        Median :4800   Median :395.0   Median : 76.00     
                         Mean   :4610   Mean   :375.7   Mean   : 74.18     
                         3rd Qu.:5200   3rd Qu.:460.0   3rd Qu.: 86.00     
                         Max.   :6700   Max.   :570.0   Max.   :300.00     
```

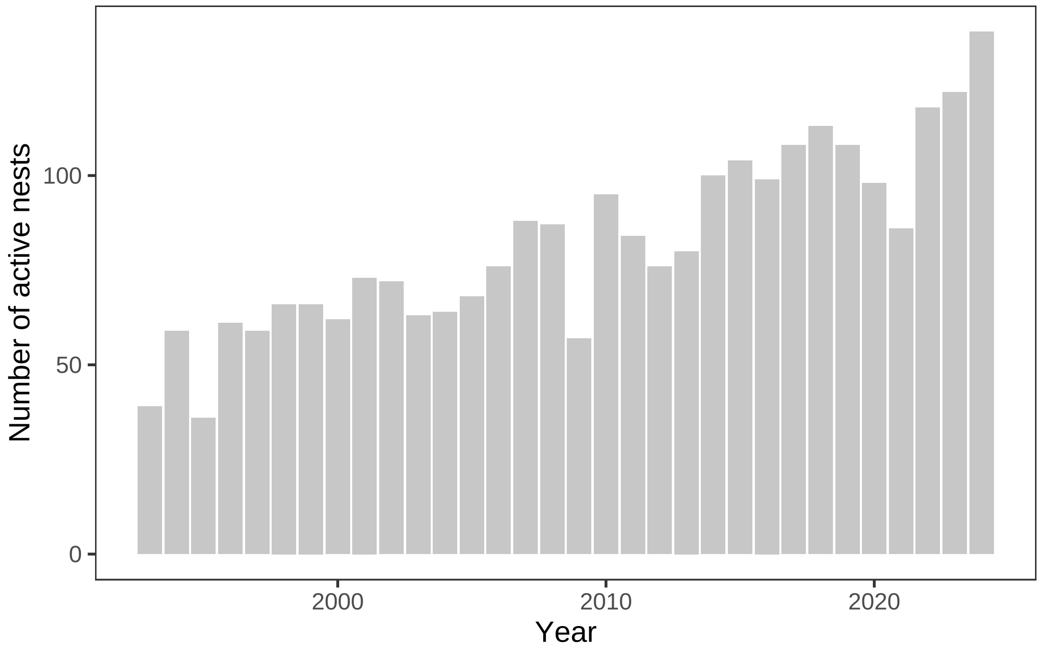
```
                         NA's   :1225   NA's   :1192    NA's   :1198
 value.at.ringing hatching.date       laying.date
 Min.   :34231    Length:2657        Min.   :1900-01-02
 1st Qu.:37541    Class :character   1st Qu.:2002-06-02
 Median :40824    Mode  :character   Median :2011-05-13
 Mean   :40489                       Mean   :2010-05-21
 3rd Qu.:43386                       3rd Qu.:2018-05-26
 Max.   :45583                       Max.   :2024-08-11
 NA's   :1231                        NA's   :1197
```

```r
colSums(is.na(vultures.clean))
```

```
          ringing.date                tree.drn              southings
                    32                       0                      0
              eastings                    code                   ring
                     0                       4                      0
colour.rings.r.top.down colour.rings.l.top.down                 mass.g
                     0                       0                   1225
        wing.length.mm      age.at.ringing.days       value.at.ringing
                  1192                    1198                   1231
         hatching.date             laying.date
                     0                    1197
```

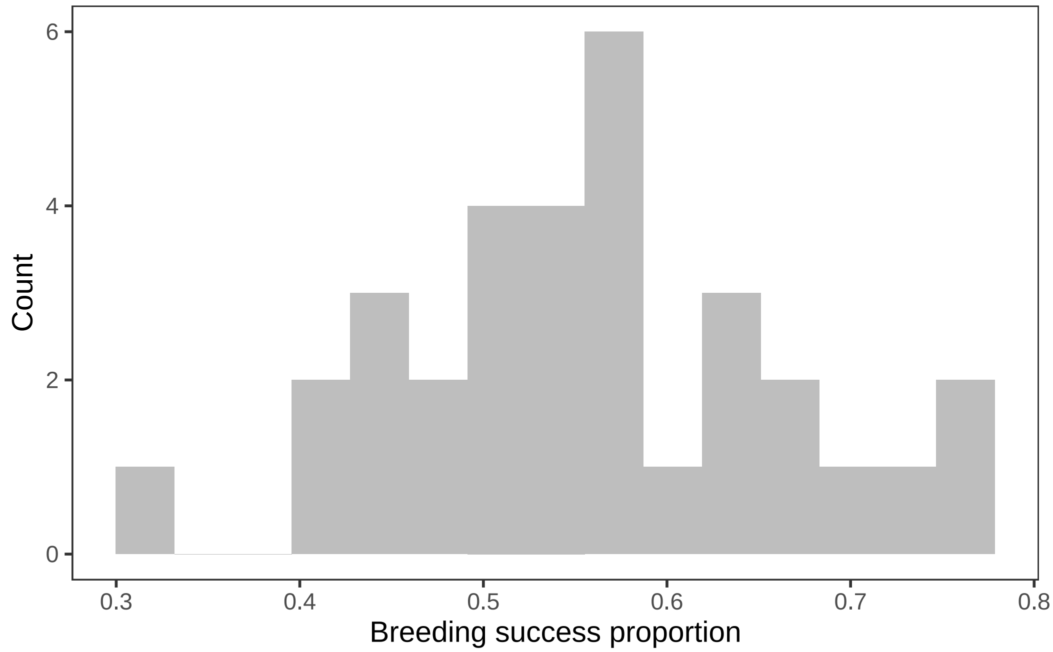
0.1.3 Number of Active Nests

```r
active_nests_yearly <- vultures.clean |>
  mutate(year = year(ringing.date)) |>
  filter(!is.na(year)) |>
  count(year, name = "active_nests")

ggplot(active_nests_yearly, aes(x = year, y = active_nests)) +
  geom_col(fill = "grey", alpha = 0.85) +
  labs(x = "Year", y = "Number of active nests") +
  theme_bw() +
  theme(panel.grid = element_blank())
```



0.1.4 Breeding success

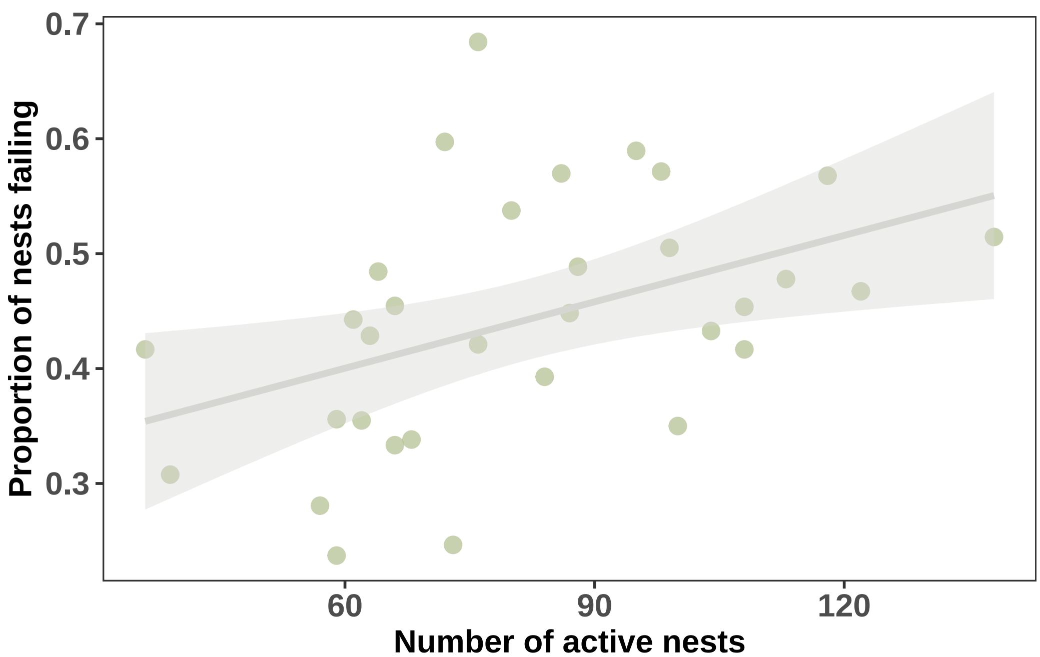
```
success_yearly <- vultures.clean |>
  mutate(year = year(ringing.date)) |>
  filter(!is.na(year)) |>
  summarise(
    active_nests  = n(),
    success_nests = sum(!is.na(laying.date)),
    .by = year
  ) |>
  mutate(
    failed_nests = active_nests - success_nests,
    success_prop = success_nests / active_nests,
    fail_prop    = 1 - success_prop
  ) |>
  filter(active_nests > 0)

ggplot(success_yearly, aes(x = success_prop)) +
  geom_histogram(bins = 15, fill = "grey") +
  labs(x = "Breeding success proportion", y = "Count") +
  theme_bw() +
  theme(panel.grid = element_blank())
```



0.1.5 Failed vs Successful nests

```
mod_fail <- lm(fail_prop ~ active_nests, data = success_yearly)

newdat <- tibble(
  active_nests = seq(
    min(success_yearly$active_nests, na.rm = TRUE),
    max(success_yearly$active_nests, na.rm = TRUE),
    length.out = 200
  )
)

pr <- predict(mod_fail, newdata = newdat, interval = "confidence") |> as.data.frame()
pred_fail <- bind_cols(newdat, pr) |>
  rename(fit = fit, lwr = lwr, upr = upr)

ggplot() +
  geom_point(
    data = success_yearly,
    aes(x = active_nests, y = fail_prop),
    colour = "#B1BD8C",
    alpha = 0.7,
    size = 2.6
  ) +
```

```r
  geom_ribbon(
    data = pred_fail,
    aes(x = active_nests, ymin = lwr, ymax = upr),
    fill = "#D5D6D2",
    alpha = 0.4
  ) +
  geom_line(
    data = pred_fail,
    aes(x = active_nests, y = fit),
    colour = "#D5D6D2",
    linewidth = 1.2
  ) +
  labs(x = "Number of active nests", y = "Proportion of nests failing") +
  theme_bw(base_family = "Times New Roman") +
  theme(
    panel.grid = element_blank(),
    axis.title = element_text(face = "bold", size = 12),
    axis.text  = element_text(face = "bold", size = 12)
  )
```



0.1.6 Weather data (GSOD)

```r
options(timeout = 600)
options(download.file.method = "libcurl")

kim_station <- "684380-99999"

weather_raw <- NULL
for (i in 1:5) {
  message("GSOD download attempt: ", i)
  weather_raw <- try(
    GSODR::get_GSOD(station = kim_station, years = 1992:2024),
    silent = TRUE
  )
  if (!inherits(weather_raw, "try-error")) break
  Sys.sleep(2 * i)
}

if (inherits(weather_raw, "try-error")) {
  stop("GSOD download failed after 5 attempts. Run this chunk interactively or use
  ↪  cached data (see below).")
}

WeatherData <- weather_raw |>
  transmute(
    YEAR, MONTH, DAY,
    TEMP  = as.numeric(TEMP),
    MAX   = as.numeric(MAX),
    MIN   = as.numeric(MIN),
    PRCP  = as.numeric(PRCP),
    WDSP  = as.numeric(WDSP),
    MXSPD = as.numeric(MXSPD),
    I_HAIL = as.numeric(I_HAIL)
  ) |>
  arrange(YEAR, MONTH, DAY) |>
  mutate(date = make_date(YEAR, MONTH, DAY))
```

0.1.7 Adding correct rainfall for 2004

```r
lonlat <- tibble(lon = 24.81, lat = -28.62)

kimberley_chirps_2004 <- get_chirps(
  object    = lonlat,
  dates     = c("2004-01-01", "2004-12-31"),
  server    = "ClimateSERV",
  as.matrix = FALSE
)
```

```r
daily_chirps_2004 <- tibble(
  date    = as.Date(kimberley_chirps_2004$date),
  rain_mm = as.numeric(kimberley_chirps_2004$chirps)
)

annual_chirps_2004 <- daily_chirps_2004 |>
  mutate(YEAR = year(date)) |>
  summarise(
    total_rain_2004 = sum(rain_mm, na.rm = TRUE),
    max_rain_2004   = max(rain_mm, na.rm = TRUE),
    rain_days_2004  = sum(rain_mm > 0, na.rm = TRUE)
  )

annual_chirps_2004
```

```
# A tibble: 1 x 3
  total_rain_2004 max_rain_2004 rain_days_2004
            <dbl>         <dbl>          <int>
1            461.          28.8             54
```

```r
WeatherData_clean <- WeatherData |>
  left_join(daily_chirps_2004, by = "date") |>
  mutate(
    PRCP = if_else(YEAR == 2004 & !is.na(rain_mm), rain_mm, PRCP)
  ) |>
  select(-rain_mm)
```

```r
library(dplyr)
library(lubridate)
library(tibble)

# Daily weather
weather_daily <- WeatherData_clean  |>
  mutate(
    date = as.Date(date),
    year = year(date)
  )

# Fixed window
weather_march_by_year <- weather_daily  |>
  filter(month(date) == 3)  |>      # March
  group_by(year)  |>
  summarise(
    mean_temp_win  = mean(TEMP,  na.rm = TRUE),
```

```
    max_temp_win   = max(MAX,    na.rm = TRUE),
    min_temp_win   = min(MIN,    na.rm = TRUE),
    total_rain_win = sum(PRCP,   na.rm = TRUE),
    max_rain_win   = max(PRCP,   na.rm = TRUE),
    rain_days_win  = sum(PRCP > 0, na.rm = TRUE),
    mean_wind_win  = mean(WDSP,  na.rm = TRUE),
    max_wind_win   = max(MXSPD,  na.rm = TRUE),
    hail_days_win  = sum(I_HAIL, na.rm = TRUE),
    n_days         = n(),
    .groups = "drop"
  )  |>
  arrange(year)

weather_march_by_year
```

```
# A tibble: 33 x 11
    year mean_temp_win max_temp_win min_temp_win total_rain_win max_rain_win
   <dbl>         <dbl>        <dbl>        <dbl>          <dbl>        <dbl>
 1  1992          23.8         37            5.3          55.1         23.1
 2  1993          21.0         31.6          5.8          12.7          7.62
 3  1994          21.7         34.5          9.7          38.4         11.4
 4  1995          22.1         35            9.4         155.          42.7
 5  1996          22.3         34.7          8.2           3.55         2.03
 6  1997          19.8         33.1          8.8         118.          33.0
 7  1998          22.2         36.7          9           100.          32
 8  1999          25.4         37.8         12.8          39.9         20.1
 9  2000          22.3         33.2         12.4          89.6         28.2
10  2001          23.2         36           11.3          57.7         24.1
# i 23 more rows
# i 5 more variables: rain_days_win <int>, mean_wind_win <dbl>,
#   max_wind_win <dbl>, hail_days_win <dbl>, n_days <int>
```

0.1.8 Lagged weather summaries

```
weather_annual_by_year <- weather_daily  |>
  group_by(year)  |>
  summarise(
    mean_temp_annual  = mean(TEMP, na.rm = TRUE),
    min_temp_annual   = mean(MIN, na.rm = TRUE),
    total_rain_annual = sum(PRCP, na.rm = TRUE),
    rain_days_annual  = sum(PRCP > 0, na.rm = TRUE),
    mean_wind_annual  = mean(WDSP, na.rm = TRUE),
```

```r
    .groups = "drop"
  )  |>
  arrange(year)

weather_annual_by_year <- weather_annual_by_year  |>
  mutate(
    lag_mean_temp_annual  = lag(mean_temp_annual, 1),
    lag_total_rain_annual = lag(total_rain_annual, 1),
    lag_rain_days_annual  = lag(rain_days_annual, 1)
  )
```

```r
summary(weather_annual_by_year)
```

```
      year      mean_temp_annual min_temp_annual  total_rain_annual
 Min.   :1992   Min.   :17.01    Min.   : 8.195   Min.   :190.5    
 1st Qu.:2000   1st Qu.:17.79    1st Qu.: 9.257   1st Qu.:347.7    
 Median :2008   Median :18.54    Median : 9.944   Median :461.1    
 Mean   :2008   Mean   :18.51    Mean   : 9.824   Mean   :441.9    
 3rd Qu.:2016   3rd Qu.:19.18    3rd Qu.:10.277   3rd Qu.:567.2    
 Max.   :2024   Max.   :20.40    Max.   :11.401   Max.   :807.3    
                                                                   
 rain_days_annual mean_wind_annual lag_mean_temp_annual lag_total_rain_annual
 Min.   : 39.00   Min.   :2.951    Min.   :17.01        Min.   :190.5        
 1st Qu.: 65.00   1st Qu.:3.381    1st Qu.:17.78        1st Qu.:346.9        
 Median : 76.00   Median :3.714    Median :18.53        Median :463.5        
 Mean   : 74.18   Mean   :3.670    Mean   :18.49        Mean   :444.9        
 3rd Qu.: 84.00   3rd Qu.:3.876    3rd Qu.:19.06        3rd Qu.:567.5        
 Max.   :107.00   Max.   :4.537    Max.   :20.40        Max.   :807.3        
                                   NA's   :1            NA's   :1            
 lag_rain_days_annual
 Min.   : 39.00      
 1st Qu.: 65.75      
 Median : 77.00      
 Mean   : 74.69      
 3rd Qu.: 84.25      
 Max.   :107.00      
 NA's   :1           
```

```r
timing_df <- vultures.clean |>
  dplyr::filter(!is.na(laying.date)) |>
  dplyr::mutate(
    year     = lubridate::year(laying.date),
```

```
    lay_DOY = lubridate::yday(laying.date)
  )
```

0.1.9 Creating 30-day window data

```
timing_by_year <- timing_df |>
  dplyr::mutate(year = as.integer(year)) |>
  dplyr::filter(!is.na(year), year != 1900) |>
  dplyr::group_by(year) |>
  dplyr::summarise(
    lay_DOY_median = median(lay_DOY, na.rm = TRUE),
    lay_DOY_mean   = mean(lay_DOY, na.rm = TRUE),
    n_nests        = sum(!is.na(lay_DOY)),
    first_DOY      = min(lay_DOY, na.rm = TRUE),
    last_DOY       = max(lay_DOY, na.rm = TRUE),
    .groups = "drop"
  ) |>
  dplyr::arrange(year)

timing_by_year
```

```
# A tibble: 32 x 6
    year lay_DOY_median lay_DOY_mean n_nests first_DOY last_DOY
   <int>          <dbl>        <dbl>   <int>     <dbl>    <dbl>
 1  1993           148.         148.      28       103      203
 2  1994           159          162.      39       128      212
 3  1995           156.         154.      22       132      171
 4  1996           158          158.      35       129      209
 5  1997           151          152.      46       132      190
 6  1998           148          151.      45       122      188
 7  1999           152          153.      37       136      192
 8  2000           153          155.      41       124      201
 9  2001           156.         158.      56       129      208
10  2002           154.         155.      30        20      185
# i 22 more rows
```

```
nrow(timing_by_year)
```

```
[1] 32
```

```r
n_distinct(timing_by_year$year)
```

```
[1] 32
```

```r
summary(timing_by_year$n_nests)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  22.00   37.75   45.50   45.59   53.00   67.00 
```

```r
obj3_year_df <- timing_by_year  |>
  left_join(weather_march_by_year, by = "year")  |>
  arrange(year)

obj3_year_df
```

```
# A tibble: 32 x 16
    year lay_DOY_median lay_DOY_mean n_nests first_DOY last_DOY mean_temp_win
   <dbl>          <dbl>        <dbl>   <int>     <dbl>    <dbl>         <dbl>
 1  1993           148.         148.      28       103      203          21.0
 2  1994           159          162.      39       128      212          21.7
 3  1995           156.         154.      22       132      171          22.1
 4  1996           158          158.      35       129      209          22.3
 5  1997           151          152.      46       132      190          19.8
 6  1998           148          151.      45       122      188          22.2
 7  1999           152          153.      37       136      192          25.4
 8  2000           153          155.      41       124      201          22.3
 9  2001           156.         158.      56       129      208          23.2
10  2002           154.         155.      30        20      185          23.5
# i 22 more rows
# i 9 more variables: max_temp_win <dbl>, min_temp_win <dbl>,
#   total_rain_win <dbl>, max_rain_win <dbl>, rain_days_win <int>,
#   mean_wind_win <dbl>, max_wind_win <dbl>, hail_days_win <dbl>, n_days <int>
```

0.1.10 Annual Weather Summaries

```r
weather_yearly <- WeatherData_clean |>
  group_by(YEAR) |>
  summarise(
```

```r
    mean_temp  = mean(TEMP, na.rm = TRUE),
    max_temp   = max(MAX,  na.rm = TRUE),
    min_temp   = min(MIN,  na.rm = TRUE),
    total_rain = sum(PRCP, na.rm = TRUE),
    max_rain   = max(PRCP, na.rm = TRUE),
    rain_days  = sum(PRCP > 0, na.rm = TRUE),
    mean_wind  = mean(WDSP, na.rm = TRUE),
    max_wind   = max(MXSPD, na.rm = TRUE),
    hail_days  = sum(I_HAIL, na.rm = TRUE),
    .groups    = "drop"
  )

weather_yearly |>
  filter(YEAR == 2004) |>
  select(YEAR, total_rain, max_rain, rain_days)
```

```
# A tibble: 1 x 4
   YEAR total_rain max_rain rain_days
  <int>      <dbl>    <dbl>     <int>
1  2004       461.     28.8        54
```

0.1.11 Weather trends over time

```r
# Temperature trends
summary(lm(mean_temp ~ YEAR, data = weather_yearly))
```

```

Call:
lm(formula = mean_temp ~ YEAR, data = weather_yearly)

Residuals:
     Min       1Q   Median       3Q      Max
-1.49859 -0.73263  0.01794  0.68011  1.91081

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.67507   31.36043   0.755    0.456
YEAR        -0.00257    0.01562  -0.165    0.870

Residual standard error: 0.8543 on 31 degrees of freedom
Multiple R-squared:  0.0008729, Adjusted R-squared:  -0.03136
F-statistic: 0.02708 on 1 and 31 DF,  p-value: 0.8704
```

```r
summary(lm(max_temp  ~ YEAR, data = weather_yearly))
```

```
Call:
lm(formula = max_temp ~ YEAR, data = weather_yearly)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4194 -1.4481 -0.1481  0.7663  5.1377 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) -103.42807   74.27533  -1.392   0.1737  
YEAR           0.07142    0.03699   1.931   0.0627 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 31 degrees of freedom
Multiple R-squared:  0.1074,	Adjusted R-squared:  0.07857 
F-statistic: 3.728 on 1 and 31 DF,  p-value: 0.06268
```

```r
summary(lm(min_temp  ~ YEAR, data = weather_yearly))
```

```
Call:
lm(formula = min_temp ~ YEAR, data = weather_yearly)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.6621 -1.1999  0.0613  0.8673  3.2483 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -32.37852   55.87587  -0.579    0.566
YEAR          0.01293    0.02783   0.465    0.645

Residual standard error: 1.522 on 31 degrees of freedom
Multiple R-squared:  0.006922,	Adjusted R-squared:  -0.02511 
F-statistic: 0.2161 on 1 and 31 DF,  p-value: 0.6453
```

```r
# Rainfall trends
summary(lm(total_rain ~ YEAR, data = weather_yearly))
```

```
Call:
lm(formula = total_rain ~ YEAR, data = weather_yearly)

Residuals:
    Min      1Q  Median      3Q     Max 
-296.19  -95.76    8.01   86.31  351.34 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 6056.800   5445.968   1.112    0.275
YEAR          -2.796      2.712  -1.031    0.311

Residual standard error: 148.3 on 31 degrees of freedom
Multiple R-squared:  0.03315,   Adjusted R-squared:  0.001965 
F-statistic: 1.063 on 1 and 31 DF,  p-value: 0.3105
```

```r
summary(lm(rain_days  ~ YEAR, data = weather_yearly))
```

```
Call:
lm(formula = rain_days ~ YEAR, data = weather_yearly)

Residuals:
    Min      1Q  Median      3Q     Max 
-32.538  -9.126  -0.448  11.707  32.252 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -305.0027   613.8635  -0.497    0.623
YEAR           0.1888     0.3057   0.618    0.541

Residual standard error: 16.72 on 31 degrees of freedom
Multiple R-squared:  0.01216,   Adjusted R-squared:  -0.01971 
F-statistic: 0.3816 on 1 and 31 DF,  p-value: 0.5413
```

```r
summary(lm(max_rain   ~ YEAR, data = weather_yearly))
```

```
Call:
lm(formula = max_rain ~ YEAR, data = weather_yearly)

Residuals:
    Min      1Q  Median      3Q     Max 
-25.448 -15.957  -6.285  14.454  60.935 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 592.4045   792.3973   0.748    0.460
YEAR         -0.2721     0.3946  -0.690    0.496

Residual standard error: 21.59 on 31 degrees of freedom
Multiple R-squared:  0.01511,   Adjusted R-squared:  -0.01666 
F-statistic: 0.4755 on 1 and 31 DF,  p-value: 0.4956
```

```r
# Wind trends
summary(lm(mean_wind ~ YEAR, data = weather_yearly))
```

```
Call:
lm(formula = mean_wind ~ YEAR, data = weather_yearly)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.73374 -0.18419 -0.03271  0.14879  0.92830 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)  
(Intercept) 34.014073  14.135469   2.406   0.0223 *
YEAR        -0.015112   0.007039  -2.147   0.0398 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3851 on 31 degrees of freedom
Multiple R-squared:  0.1294,    Adjusted R-squared:  0.1013 
F-statistic: 4.608 on 1 and 31 DF,  p-value: 0.03975
```

```r
summary(lm(max_wind  ~ YEAR, data = weather_yearly))
```

```

Call:
lm(formula = max_wind ~ YEAR, data = weather_yearly)

Residuals:
   Min     1Q Median     3Q    Max 
-6.003 -3.491 -1.363  2.440  9.249 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 194.80838  176.22563   1.105    0.277
YEAR         -0.08790    0.08776  -1.002    0.324

Residual standard error: 4.8 on 31 degrees of freedom
Multiple R-squared:  0.03135,	Adjusted R-squared:  9.998e-05 
F-statistic: 1.003 on 1 and 31 DF,  p-value: 0.3243
```

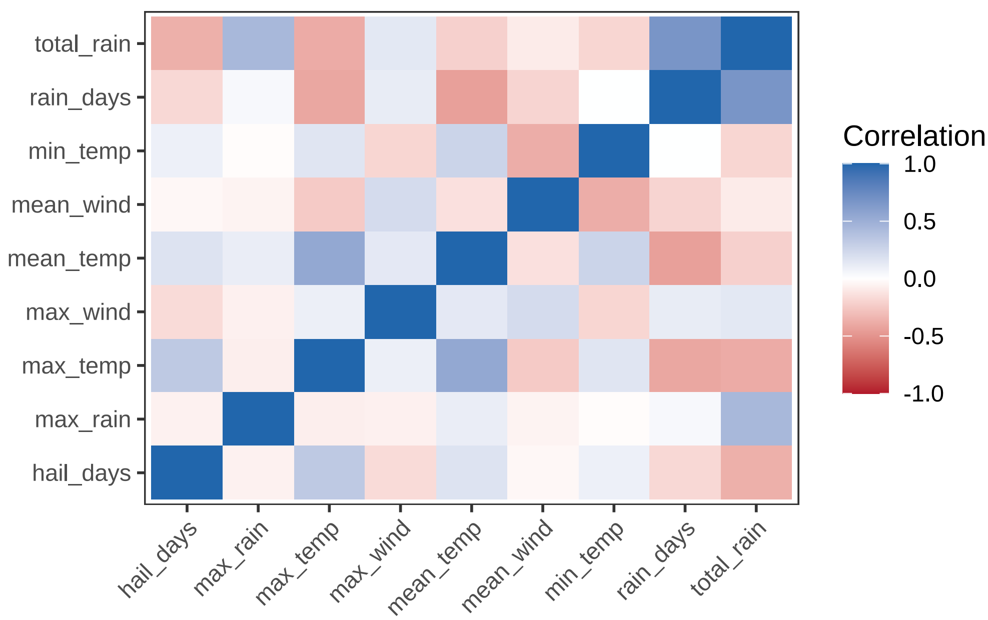
0.1.12 Lagged Weather summaries

```r
weather_yearly_lag <- weather_yearly |>
  arrange(YEAR) |>
  mutate(across(
    c(mean_temp, max_temp, min_temp, total_rain, max_rain, rain_days,
      mean_wind, max_wind, hail_days),
    ~ lag(.x, 1),
    .names = "lag_{.col}"
  ))
```

0.1.13 Correlations between weather variables

```r
vars <- weather_yearly |>
  select(mean_temp, max_temp, min_temp,
         total_rain, max_rain, rain_days,
         mean_wind, max_wind, hail_days)

res <- rcorr(as.matrix(vars), type = "pearson")
```

```r
cor_df <- as.data.frame(res$r) |>
  rownames_to_column("var1") |>
  pivot_longer(-var1, names_to = "var2", values_to = "cor") |>
  mutate(p = as.vector(res$P))

ggplot(cor_df, aes(x = var1, y = var2, fill = cor)) +
  geom_tile() +
  scale_fill_gradient2(low = "#b2182b", mid = "white", high = "#2166ac",
                       midpoint = 0, limits = c(-1, 1)) +
  labs(x = "", y = "", fill = "Correlation") +
  theme_bw() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid  = element_blank()
  )
```



0.1.14 Trends over time for breeding variables

0.1.15 Number of Active nests

```r
summary(lm(active_nests ~ year, data = active_nests_yearly))
```

```
Call:
lm(formula = active_nests ~ year, data = active_nests_yearly)

Residuals:
    Min      1Q  Median      3Q     Max 
-26.189  -6.508   4.153   7.167  20.079 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4568.5150   413.0744  -11.06 4.17e-12 ***
year            2.3154     0.2057   11.26 2.70e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.74 on 30 degrees of freedom
Multiple R-squared:  0.8086,	Adjusted R-squared:  0.8022 
F-statistic: 126.8 on 1 and 30 DF,  p-value: 2.699e-12
```

0.1.16 Breeding success

```r
success_year <- glm(
  cbind(success_nests, failed_nests) ~ year,
  family = binomial,
  data   = success_yearly
)

summary(success_year)
```

```
Call:
glm(formula = cbind(success_nests, failed_nests) ~ year, family = binomial, 
    data = success_yearly)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) 45.434641   8.858715   5.129 2.92e-07 ***
year        -0.022505   0.004405  -5.109 3.24e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 107.689  on 31  degrees of freedom
Residual deviance:  81.283  on 30  degrees of freedom
AIC: 237.58

Number of Fisher Scoring iterations: 3
```

```
check_overdispersion(success_year)
```

```
[1] 2.670817
```

0.1.17 Breeding timing of succesful nests

```
timing_yearly <- vultures.clean |>
  mutate(
    year    = year(ringing.date),
    lay_DOY = yday(laying.date)
  ) |>
  filter(!is.na(year), !is.na(lay_DOY)) |>
  group_by(year) |>
  summarise(
    mean_lay_DOY = mean(lay_DOY, na.rm = TRUE),
    .groups = "drop"
  )

timing_yearly_clean <- timing_yearly |>
  filter(year >= 1992)

summary(lm(mean_lay_DOY ~ year, data = timing_yearly_clean))
```

```
Call:
lm(formula = mean_lay_DOY ~ year, data = timing_yearly_clean)

Residuals:
    Min      1Q  Median      3Q     Max
-9.3676 -2.6114 -0.0087  3.5078  7.7270

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 312.43769  165.46254   1.888   0.0687 .
year         -0.07915    0.08238  -0.961   0.3443
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.303 on 30 degrees of freedom
Multiple R-squared:  0.02985,   Adjusted R-squared:  -0.002487
F-statistic: 0.9231 on 1 and 30 DF,  p-value: 0.3443
```

0.2 Objective 1: Number of Active nests

0.2.1 DAGS

0.2.2 Scenario A

```
dir.create("figures", showWarnings = FALSE)

export_dag_pdf <- function(dag, filename, w = 8, h = 4) {
  svg_txt <- DiagrammeRsvg::export_svg(dag)
  rsvg::rsvg_pdf(
    charToRaw(svg_txt),
    file = file.path("figures", filename),
    width = w, height = h
  )
  file.info(file.path("figures", filename))[, "size"]
}

trend_confounded_dag <- DiagrammeR::grViz("
digraph trend_confounded {
  graph [
    layout = neato,
    splines = false,
    overlap = false
  ]

  node [
    fontname = Helvetica,
    fontsize = 22,
    penwidth = 2,
    shape = box,
    margin = 0.25
  ]

  edge [
    penwidth = 1.8,
```

```
    arrowsize = 0.55
  ]

  Y [label = 'Y', pos = '2.5,2!']
  W [label = 'W', pos = '1.2,0!']
  N [label = 'N', pos = '4.0,0!']

  Y -> W
  Y -> N
  W -> N
}
")

export_dag_pdf(trend_confounded_dag, "trend_confounded_dag.pdf", 8, 4)
```

```
[1] 5982
```

0.2.3 Scenario B

```
trend_W_B_dag <- DiagrammeR::grViz("
digraph trend_W_B {
  graph [
    layout = neato,
    splines = false,
    overlap = false
  ]

  node [
    fontname = Helvetica,
    fontsize = 22,
    penwidth = 2,
    shape = box,
    margin = 0.25
  ]

  edge [
    penwidth = 1.8,
    arrowsize = 0.55
  ]

  Y [label = 'Y', pos = '0,0!']
  W [label = 'W', pos = '3,0!']
  N [label = 'N', pos = '6,0!']

  Y -> W
```

```
  W -> N
}
")

export_dag_pdf(trend_W_B_dag, "trend_W_B_dag.pdf", 8, 4)
```

```
[1] 5915
```

0.2.4 Scenario C

```
trend_mediator_dag <- DiagrammeR::grViz("
digraph trend_mediator {
  graph [
    layout = neato,
    splines = false,
    overlap = false
  ]

  node [
    fontname = Helvetica,
    fontsize = 22,
    penwidth = 2,
    shape = box,
    margin = 0.25
  ]

  edge [
    penwidth = 1.8,
    arrowsize = 0.55
  ]

  W [label = 'W', pos = '0,0!']
  Y [label = 'Y', pos = '3,0!']
  N [label = 'N', pos = '6,0!']

  W -> Y
  Y -> N
}
")

export_dag_pdf(trend_mediator_dag, "trend_mediator_dag.pdf", 8, 4)
```

```
[1] 5910
```

0.2.5 Models

```
objective1_table <- tribble(
  ~`Hypothesis / justification`, ~Model,

  "**Temp**", NA,
  "Warmer years increase energetic cost of adult birds, reducing likelihood to
   ↪  breed.",
  "active_nests ~ mean_temp",
  "Extreme heat events cause thermal stress, reducing breeding effort (active cooling
   ↪  is energetically costly and may reduce adult condition).",
  "active_nests ~ max_temp",
  "Cold extremes increase thermoregulatory demand (often increasing time spent
   ↪  foraging), reducing effort put into breeding",
  "active_nests ~ min_temp",

  "**Rain**", NA,
  "High rainfall reduces foraging efficiency and adult condition, reducing breeding
   ↪  effort.",
  "active_nests ~ total_rain",
  "Prolonged rainfall conditions reduce breeding effort by limiting foraging
   ↪  opportunity (fewer or weaker thermals for soaring)",
  "active_nests ~ rain_days",
  "Intense rainfall events decrease foraging efficiency and nest attendance, reducing
   ↪  breeding effort.",
  "active_nests ~ max_rain",
  "Previous year's rainfall influences food availability and adult condition,
   ↪  affecting breeding effort in the following year.",
  "active_nests ~ lag_total_rain",

  "**Wind**", NA,
  "Persistent windy conditions increase the cost of flying, limiting foraging and the
   ↪  energy available for reproduction, reducing breeding effort",
  "active_nests ~ mean_wind",
  "Extreme winds increase nest damage risk, reducing the likelihood that adults will
   ↪  attempt breeding (WBV often reuse nests from previous years) ",
  "active_nests ~ max_wind",

  "**Hail**", NA,
  "Severe storm events increase disturbance and nest damage risk, reducing breeding
   ↪  effort",
  "active_nests ~ hail_days",

  "**Joint effects**", NA,
  "Cold stress combined with prolonged wet conditions jointly limit breeding effort",
  "active_nests ~ min_temp + total_rain",
  "Cold stress combined with intense rainfall events reduces breeding effort",
```

```
  "active_nests ~ min_temp + max_rain",
  "Cold extremes combined with extreme wind increase energetic costs and breeding
   ↪  risk, reducing breeding effort",
  "active_nests ~ min_temp + max_wind",
  "Severe storm exposure combined with extreme wind increases nest disturbance,
   ↪  reducing breeding effort",
  "active_nests ~ hail_days + max_wind",
  "Reduced thermals for soaring combined with high flight costs limit breeding
   ↪  effort",
  "active_nests ~ max_rain + max_wind",
  "Heat stress combined with prolonged wet conditions jointly reduce breeding effort
   ↪  by increasing energetic costs while limiting foraging",
  "active_nests ~ max_temp + rain_days",
  "High annual rainfall and frequent hail events each impose energetic and
   ↪  disturbance costs that reduce breeding effort.",
  "active_nests ~ hail_days + rain_days",

  "**Interactive effects**", NA,
  "The negative effect of cold extremes on breeding effort is increased in
   ↪  persistently wet years.",
  "active_nests ~ min_temp * rain_days",
  "Cold-related energetic stress is amplified in windy years due to increased heat
   ↪  loss and flight costs",
  "active_nests ~ min_temp * mean_wind",
  "Adult condition resulting from the previous year interacts with current-year heat
   ↪  stress to influence breeding effort",
  "active_nests ~ lag_total_rain * max_temp"
) |>
  mutate(Model = if_else(is.na(Model), "", Model))

knitr::kable(
  objective1_table,
  align = c("l", "l"),
  caption = "Objective 1 candidate model set for breeding effort (active nests)"
)
```

Table 1: Objective 1 candidate model set for breeding effort (active nests)

Hypothesis / justification	Model
Temp	
Warmer years increase energetic cost of adult birds, reducing likelihood to breed.	active_nests ~ mean_temp
Extreme heat events cause thermal stress, reducing breeding effort (active cooling is energetically costly and may reduce adult condition).	active_nests ~ max_temp
Cold extremes increase thermoregulatory demand (often increasing time spent foraging), reducing effort put into breeding	active_nests ~ min_temp

Hypothesis / justification	Model
Rain	
High rainfall reduces foraging efficiency and adult condition, reducing breeding effort.	active_nests ~ total_rain
Prolonged rainfall conditions reduce breeding effort by limiting foraging opportunity (fewer or weaker thermals for soaring)	active_nests ~ rain_days
Intense rainfall events decrease foraging efficiency and nest attendance, reducing breeding effort.	active_nests ~ max_rain
Previous year's rainfall influences food availability and adult condition, affecting breeding effort in the following year.	active_nests ~ lag_total_rain
Wind	
Persistent windy conditions increase the cost of flying, limiting foraging and the energy available for reproduction, reducing breeding effort	active_nests ~ mean_wind
Extreme winds increase nest damage risk, reducing the likelihood that adults will attempt breeding (WBV often reuse nests from previous years)	active_nests ~ max_wind
Hail	
Severe storm events increase disturbance and nest damage risk, reducing breeding effort	active_nests ~ hail_days
Joint effects	
Cold stress combined with prolonged wet conditions jointly limit breeding effort	active_nests ~ min_temp + total_rain
Cold stress combined with intense rainfall events reduces breeding effort	active_nests ~ min_temp + max_rain
Cold extremes combined with extreme wind increase energetic costs and breeding risk, reducing breeding effort	active_nests ~ min_temp + max_wind
Severe storm exposure combined with extreme wind increases nest disturbance, reducing breeding effort	active_nests ~ hail_days + max_wind
Reduced thermals for soaring combined with high flight costs limit breeding effort	active_nests ~ max_rain + max_wind
Heat stress combined with prolonged wet conditions jointly reduce breeding effort by increasing energetic costs while limiting foraging	active_nests ~ max_temp + rain_days
High annual rainfall and frequent hail events each impose energetic and disturbance costs that reduce breeding effort.	active_nests ~ hail_days + rain_days

Hypothesis / justification	Model
Interactive effects	
The negative effect of cold extremes on breeding effort is increased in persistently wet years.	active_nests ~ min_temp * rain_days
Cold-related energetic stress is amplified in windy years due to increased heat loss and flight costs	active_nests ~ min_temp * mean_wind
Adult condition resulting from the previous year interacts with current-year heat stress to influence breeding effort	active_nests ~ lag_total_rain * max_temp

0.2.6 Data for number of active nest models

```
# number of active nests per year
active.nests <- vultures.clean |>
  mutate(year = year(ringing.date)) |>
  group_by(year) |>
  summarise(active_nests = n(), .groups = "drop")


# Join to weather, create log effort response
effort_weather <- active.nests |>
  left_join(weather_yearly_lag, by = c("year" = "YEAR")) |>
  filter(!is.na(lag_total_rain)) |>
  mutate(
    log_active_nests = log(active_nests)
  )
```

0.2.7 Fitting the no-year models

```
# Baseline
m_null <- lm(log_active_nests ~ 1, data = effort_weather)

# Single predictor models
m_mean_temp      <- lm(log_active_nests ~ mean_temp,      data = effort_weather)
m_max_temp       <- lm(log_active_nests ~ max_temp,       data = effort_weather)
m_min_temp       <- lm(log_active_nests ~ min_temp,       data = effort_weather)

m_total_rain     <- lm(log_active_nests ~ total_rain,     data = effort_weather)
```

```r
m_rain_days      <- lm(log_active_nests ~ rain_days,      data = effort_weather)
m_max_rain       <- lm(log_active_nests ~ max_rain,       data = effort_weather)
m_lag_total_rain <- lm(log_active_nests ~ lag_total_rain, data = effort_weather)

m_mean_wind      <- lm(log_active_nests ~ mean_wind,      data = effort_weather)
m_max_wind       <- lm(log_active_nests ~ max_wind,       data = effort_weather)
m_hail_days      <- lm(log_active_nests ~ hail_days,      data = effort_weather)

# Joint effects
m_minT_totalR    <- lm(log_active_nests ~ min_temp + total_rain,  data =
↪   effort_weather)
m_minT_maxR      <- lm(log_active_nests ~ min_temp + max_rain,    data =
↪   effort_weather)
m_minT_maxW      <- lm(log_active_nests ~ min_temp + max_wind,    data =
↪   effort_weather)
m_hail_maxW      <- lm(log_active_nests ~ hail_days + max_wind,   data =
↪   effort_weather)
m_maxR_maxW      <- lm(log_active_nests ~ max_rain + max_wind,    data =
↪   effort_weather)
m_maxT_rainDays  <- lm(log_active_nests ~ max_temp + rain_days,   data =
↪   effort_weather)
m_hail_rainDays  <- lm(log_active_nests ~ hail_days + rain_days,  data =
↪   effort_weather)

# Interactions
m_minT_x_rainDays <- lm(log_active_nests ~ min_temp * rain_days,      data =
↪   effort_weather)
m_minT_x_meanWind <- lm(log_active_nests ~ min_temp * mean_wind,      data =
↪   effort_weather)
m_lagR_x_maxT     <- lm(log_active_nests ~ lag_total_rain * max_temp, data =
↪   effort_weather)
```

0.2.8 Fit with year models

```r
# Baseline
m_year <- lm(log_active_nests ~ year, data = effort_weather)

# Single predictor models
m_mean_temp_year      <- lm(log_active_nests ~ year + mean_temp,      data =
↪   effort_weather)
m_max_temp_year       <- lm(log_active_nests ~ year + max_temp,       data =
↪   effort_weather)
m_min_temp_year       <- lm(log_active_nests ~ year + min_temp,       data =
↪   effort_weather)
```

```r
m_total_rain_year     <- lm(log_active_nests ~ year + total_rain,     data =
  ↪ effort_weather)
m_rain_days_year      <- lm(log_active_nests ~ year + rain_days,      data =
  ↪ effort_weather)
m_max_rain_year       <- lm(log_active_nests ~ year + max_rain,       data =
  ↪ effort_weather)
m_lag_total_rain_year <- lm(log_active_nests ~ year + lag_total_rain, data =
  ↪ effort_weather)

m_mean_wind_year      <- lm(log_active_nests ~ year + mean_wind,      data =
  ↪ effort_weather)
m_max_wind_year       <- lm(log_active_nests ~ year + max_wind,       data =
  ↪ effort_weather)
m_hail_days_year      <- lm(log_active_nests ~ year + hail_days,      data =
  ↪ effort_weather)

# Joint effects
m_minT_totalR_year    <- lm(log_active_nests ~ year + min_temp + total_rain, data =
  ↪ effort_weather)
m_minT_maxR_year      <- lm(log_active_nests ~ year + min_temp + max_rain,   data =
  ↪ effort_weather)
m_minT_maxW_year      <- lm(log_active_nests ~ year + min_temp + max_wind,   data =
  ↪ effort_weather)
m_hail_maxW_year      <- lm(log_active_nests ~ year + hail_days + max_wind,  data =
  ↪ effort_weather)
m_maxR_maxW_year      <- lm(log_active_nests ~ year + max_rain + max_wind,   data =
  ↪ effort_weather)
m_maxT_rainDays_year  <- lm(log_active_nests ~ year + max_temp + rain_days,  data =
  ↪ effort_weather)
m_hail_rainDays_year  <- lm(log_active_nests ~ year + hail_days + rain_days, data =
  ↪ effort_weather)

# Interactions
m_minT_x_rainDays_year <- lm(log_active_nests ~ year + min_temp * rain_days,
  ↪ data = effort_weather)
m_minT_x_meanWind_year <- lm(log_active_nests ~ year + min_temp * mean_wind,
  ↪ data = effort_weather)
m_lagR_x_maxT_year     <- lm(log_active_nests ~ year + lag_total_rain * max_temp,
  ↪ data = effort_weather)
```

0.2.9 Two separate AICc model selection runs

```r
## AIC 1: without year
cand_effort_no_year <- list(
  null            = m_null,
```

```r
  mean_temp       = m_mean_temp,
  max_temp        = m_max_temp,
  min_temp        = m_min_temp,
  total_rain      = m_total_rain,
  rain_days       = m_rain_days,
  max_rain        = m_max_rain,
  lag_total_rain  = m_lag_total_rain,
  mean_wind       = m_mean_wind,
  max_wind        = m_max_wind,
  hail_days       = m_hail_days,

  minT_totalR     = m_minT_totalR,
  minT_maxR       = m_minT_maxR,
  minT_maxW       = m_minT_maxW,
  hail_maxW       = m_hail_maxW,
  maxR_maxW       = m_maxR_maxW,
  maxT_rainDays   = m_maxT_rainDays,
  hail_rainDays   = m_hail_rainDays,

  minT_x_rainDays = m_minT_x_rainDays,
  minT_x_meanWind = m_minT_x_meanWind,
  lagR_x_maxT     = m_lagR_x_maxT
)

aic_effort_no_year <- aictab(cand.set = cand_effort_no_year, modnames =
 ↪  names(cand_effort_no_year))
aic_effort_no_year
```

```
Model selection based on AICc:

                 K  AICc Delta_AICc AICcWt Cum.Wt    LL
total_rain       3 17.13       0.00   0.14   0.14 -5.14
max_rain         3 17.45       0.32   0.12   0.27 -5.30
max_temp         3 17.80       0.67   0.10   0.37 -5.47
hail_days        3 17.96       0.83   0.10   0.46 -5.55
maxR_maxW        4 18.03       0.90   0.09   0.56 -4.27
maxT_rainDays    4 18.78       1.65   0.06   0.62 -4.65
mean_wind        3 19.00       1.86   0.06   0.68 -6.07
null             2 19.44       2.31   0.05   0.72 -7.51
minT_maxR        4 19.66       2.53   0.04   0.76 -5.09
minT_totalR      4 19.73       2.60   0.04   0.80 -5.12
hail_maxW        4 19.90       2.77   0.04   0.84 -5.21
hail_rainDays    4 20.11       2.98   0.03   0.87 -5.32
```

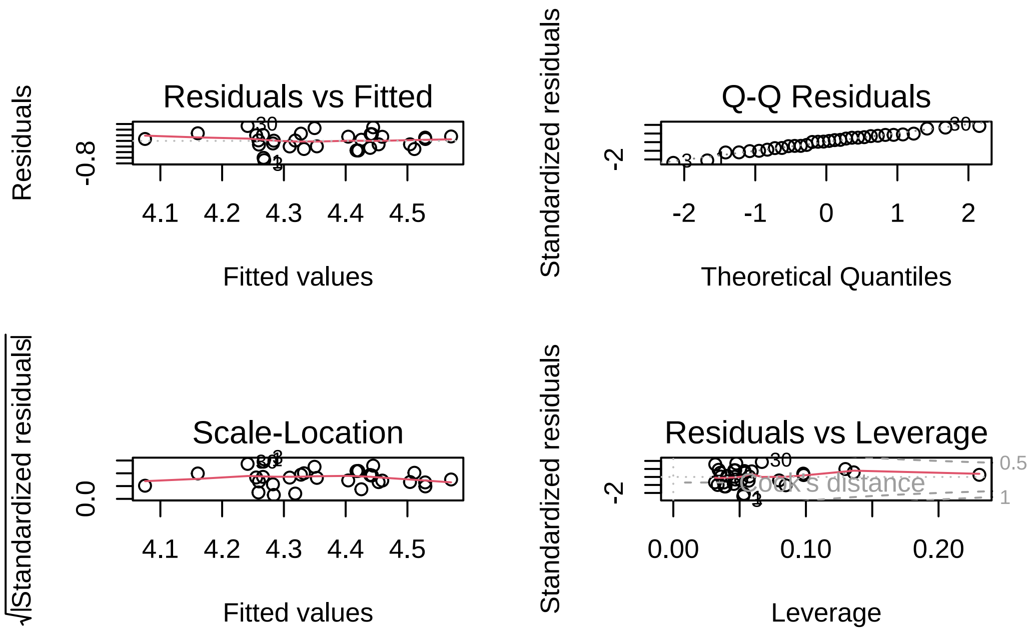
```
lagR_x_maxT     5 20.18       3.04   0.03   0.90 -3.93
max_wind        3 20.61       3.48   0.03   0.93 -6.88
min_temp        3 21.38       4.25   0.02   0.94 -7.26
mean_temp       3 21.61       4.48   0.02   0.96 -7.38
lag_total_rain  3 21.73       4.60   0.01   0.97 -7.43
rain_days       3 21.85       4.72   0.01   0.99 -7.50
minT_maxW       4 23.01       5.88   0.01   0.99 -6.76
minT_x_meanWind 5 24.01       6.88   0.00   1.00 -5.85
minT_x_rainDays 5 26.33       9.19   0.00   1.00 -7.01
```

```r
## AIC 2: with year
cand_effort_with_year <- list(
  year                = m_year,

  mean_temp           = m_mean_temp_year,
  max_temp            = m_max_temp_year,
  min_temp            = m_min_temp_year,
  total_rain          = m_total_rain_year,
  rain_days           = m_rain_days_year,
  max_rain            = m_max_rain_year,
  lag_total_rain      = m_lag_total_rain_year,
  mean_wind           = m_mean_wind_year,
  max_wind            = m_max_wind_year,
  hail_days           = m_hail_days_year,

  minT_totalR         = m_minT_totalR_year,
  minT_maxR           = m_minT_maxR_year,
  minT_maxW           = m_minT_maxW_year,
  hail_maxW           = m_hail_maxW_year,
  maxR_maxW           = m_maxR_maxW_year,
  maxT_rainDays       = m_maxT_rainDays_year,
  hail_rainDays       = m_hail_rainDays_year,

  minT_x_rainDays     = m_minT_x_rainDays_year,
  minT_x_meanWind     = m_minT_x_meanWind_year,
  lagR_x_maxT         = m_lagR_x_maxT_year
)

aic_effort_with_year <- aictab(cand.set = cand_effort_with_year, modnames =
↪  names(cand_effort_with_year))
aic_effort_with_year
```

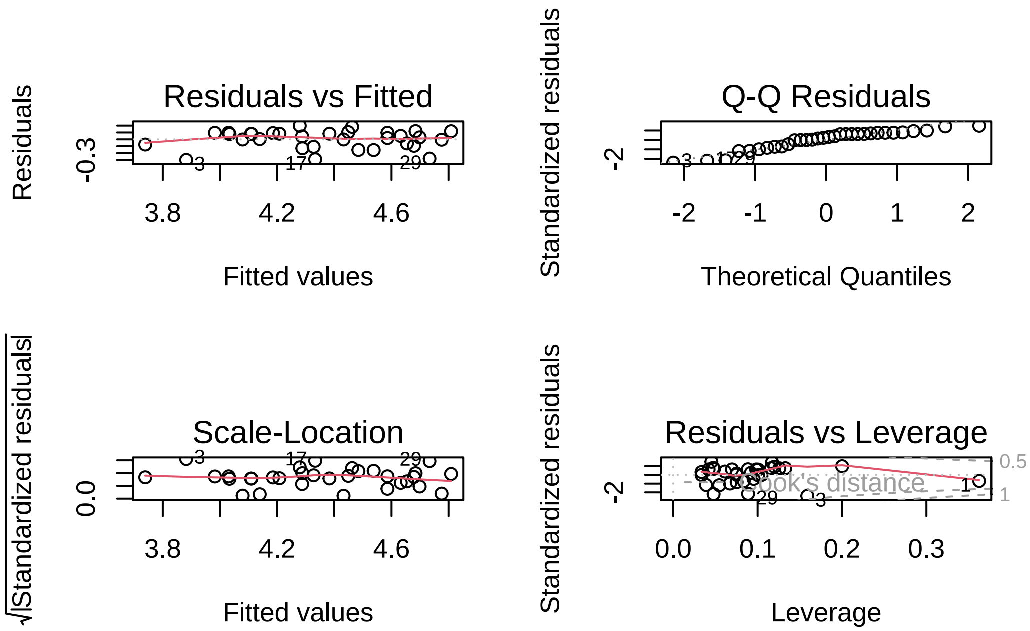
```
Model selection based on AICc:

                 K   AICc Delta_AICc AICcWt Cum.Wt    LL
```

```
max_rain        4 -30.85       0.00    0.39    0.39 20.17
minT_maxR       5 -29.55       1.31    0.20    0.59 20.93
maxR_maxW       5 -28.14       2.71    0.10    0.69 20.22
year            3 -27.09       3.76    0.06    0.75 16.97
total_rain      4 -26.37       4.49    0.04    0.79 17.92
min_temp        4 -25.97       4.88    0.03    0.82 17.73
mean_temp       4 -25.27       5.58    0.02    0.84 17.38
lag_total_rain  4 -25.17       5.68    0.02    0.87 17.32
max_temp        4 -24.99       5.86    0.02    0.89 17.24
mean_wind       4 -24.70       6.16    0.02    0.91 17.09
rain_days       4 -24.50       6.35    0.02    0.92 16.99
hail_days       4 -24.47       6.38    0.02    0.94 16.98
max_wind        4 -24.47       6.38    0.02    0.95 16.97
minT_totalR     5 -24.37       6.49    0.02    0.97 18.34
minT_maxW       5 -23.25       7.60    0.01    0.98 17.78
maxT_rainDays   5 -22.55       8.30    0.01    0.98 17.43
hail_rainDays   5 -21.69       9.16    0.00    0.99 17.00
hail_maxW       5 -21.65       9.21    0.00    0.99 16.98
lagR_x_maxT     6 -21.46       9.39    0.00    0.99 18.41
minT_x_meanWind 6 -21.16       9.69    0.00    1.00 18.26
minT_x_rainDays 6 -20.32      10.53    0.00    1.00 17.84
```

```
## Best model (no year)
best_no_year_name <- aic_effort_no_year$Modnames[1]
best_no_year_model <- cand_effort_no_year[[best_no_year_name]]

par(mfrow = c(2,2))
plot(best_no_year_model)
```




```r
## Best model (with year)
best_with_year_name <- aic_effort_with_year$Modnames[1]
best_with_year_model <- cand_effort_with_year[[best_with_year_name]]

par(mfrow = c(2,2))
plot(best_with_year_model)
```




```r
## Adjusted R² summaries (as you had)
summary(m_year)$adj.r.squared
```

```
[1] 0.776371
```

```r
summary(best_with_year_model)$adj.r.squared
```

```
[1] 0.8105132
```

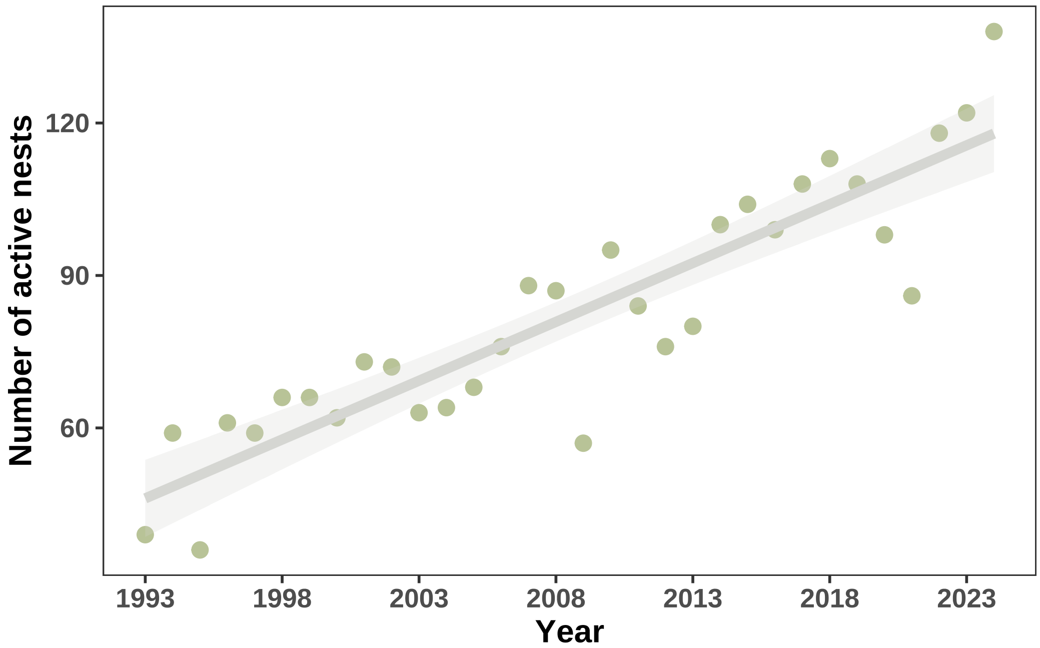
```r
summary(best_no_year_model)$adj.r.squared
```

```
[1] 0.1093628
```

0.2.10 Annual breeding effort over time

```r
ggplot(effort_weather, aes(x = year, y = active_nests)) +
  geom_point(
    size = 2.4,
    colour = "#B1BD8C",
```

```r
    alpha = 0.9
  ) +
  geom_smooth(
    method = "lm",
    se = TRUE,
    colour = "#D5D6D2",
    fill   = "#D5D6D2",
    alpha  = 0.25,
    linewidth = 1.8
  ) +
  scale_x_continuous(
    breaks = seq(
      min(effort_weather$year),
      max(effort_weather$year),
      by = 5
    )
  ) +
  labs(
    x = "Year",
    y = "Number of active nests"
  ) +
  theme_bw(base_family = "Times New Roman") +
  theme(
    panel.grid = element_blank(),
    axis.title = element_text(face = "bold", size = 12),
    axis.text  = element_text(face = "bold", size = 10)
  )
```



0.2.11 AIC tables: with and without year

```
library(dplyr)
library(tibble)
library(officer)
library(flextable)

if (requireNamespace("conflicted", quietly = TRUE)) {
  conflicted::conflicts_prefer(flextable::compose)
}

stopifnot(exists("aic_effort_with_year"))
stopifnot(exists("cand_effort_with_year"))
stopifnot(exists("aic_effort_no_year"))
stopifnot(exists("cand_effort_no_year"))


group_order <- c("Baseline","Rain","Temperature","Wind","Hail","Joint
↪  effects","Interactions")

model_group <- function(m){
  dplyr::case_when(
    m %in% c("null", "year") ~ "Baseline",
    m %in% c("total_rain","rain_days","max_rain","lag_total_rain") ~ "Rain",
```

```r
    m %in% c("mean_temp","max_temp","min_temp") ~ "Temperature",
    m %in% c("mean_wind","max_wind") ~ "Wind",
    m %in% c("hail_days") ~ "Hail",
    m %in% c("minT_totalR","minT_maxR","minT_maxW","hail_maxW","maxR_maxW",
    ↪ "maxT_rainDays","hail_rainDays") ~ "Joint effects",
    m %in% c("minT_x_rainDays","minT_x_meanWind","lagR_x_maxT") ~ "Interactions",
    TRUE ~ "Joint effects"
  )
}

label_model <- function(x){
  dplyr::recode(
    x,
    "null"             = "Intercept only",
    "year"             = "Year (baseline)",
    "total_rain"       = "Total rainfall",
    "rain_days"        = "Rainy days",
    "max_rain"         = "Maximum daily rainfall",
    "lag_total_rain"   = "Previous year's total rainfall",
    "mean_temp"        = "Mean temperature",
    "max_temp"         = "Maximum temperature",
    "min_temp"         = "Minimum temperature",
    "mean_wind"        = "Mean wind speed",
    "max_wind"         = "Maximum wind speed",
    "hail_days"        = "Hail days",
    "minT_totalR"      = "Min temperature + total rainfall",
    "minT_maxR"        = "Min temperature + max daily rainfall",
    "minT_maxW"        = "Min temperature + max wind speed",
    "hail_maxW"        = "Hail days + max wind speed",
    "maxR_maxW"        = "Max daily rainfall + max wind speed",
    "maxT_rainDays"    = "Max temperature + rainy days",
    "hail_rainDays"    = "Hail days + rainy days",
    "minT_x_rainDays" = "Min temperature × rainy days",
    "minT_x_meanWind" = "Min temperature × mean wind speed",
    "lagR_x_maxT"      = "Prev. rainfall × max temperature",
    .default = x
  )
}

single_weather_models <- c(
  "total_rain","rain_days","max_rain","lag_total_rain",
  "mean_temp","max_temp","min_temp",
  "mean_wind","max_wind",
  "hail_days"
)

get_slope_se_single_weather <- function(mod, model_id){
  slope_se <- ""
```

```r
  if (model_id %in% single_weather_models) {
    cn <- names(coef(mod))
    cn <- setdiff(cn, "(Intercept)")
    cn_weather <- cn[!grepl("^year$|^year\\b|\\byear\\b", cn)]
    if (length(cn_weather) == 1) {
      term <- cn_weather[1]
      est <- coef(mod)[term]
      se  <- sqrt(diag(vcov(mod)))[term]
      slope_se <- paste0(round(est, 3), " (", round(se, 3), ")")
    }
  }
  slope_se
}

prep_aic_set <- function(aic_obj){
  aic_df <- as.data.frame(aic_obj)

  if ("Modnames" %in% names(aic_df)) {
    aic_df <- aic_df  |>  mutate(model_id = Modnames)
  } else {
    aic_df <- tibble::rownames_to_column(aic_df, "model_id")
  }

  if (!("AICc" %in% names(aic_df))) stop("Could not find an AICc column in the AIC
   ↪  table.")

  if (!("K" %in% names(aic_df))) {
    if ("k" %in% names(aic_df)) aic_df$K <- aic_df$k
    if ("df" %in% names(aic_df)) aic_df$K <- aic_df$df
  }
  if (!("K" %in% names(aic_df))) stop("Could not find K (number of parameters) in the
   ↪  AIC table.")

  aic_df  |>
    mutate(
      AICc = as.numeric(AICc),
      `ΔAICc` = AICc - min(AICc, na.rm = TRUE),
      w = {
        rel <- exp(-0.5 * (AICc - min(AICc, na.rm = TRUE)))
        rel / sum(rel, na.rm = TRUE)
      }
    )  |>
    select(model_id, K, AICc, `ΔAICc`, w)  |>
    arrange(`ΔAICc`)
}

get_deviance_original <- function(model_id, aic_obj, cand_list){
  aic_df <- as.data.frame(aic_obj)
```

```r
  if ("Modnames" %in% names(aic_df)) {
    aic_df$model_id <- aic_df$Modnames
  } else if (!"model_id" %in% names(aic_df)) {
    aic_df <- tibble::rownames_to_column(aic_df, "model_id")
  }

  row <- aic_df[aic_df$model_id == model_id, , drop = FALSE]

  if (nrow(row) == 1) {
    if ("Deviance" %in% names(row) && is.finite(row$Deviance))
     ↪  return(as.numeric(row$Deviance))
    if ("LL" %in% names(row) && is.finite(row$LL)) return(as.numeric(row$LL))
  }

  as.numeric(-2 * stats::logLik(cand_list[[model_id]]))
}

build_df <- function(tab, cand_list, aic_obj){
  tab  |>
    mutate(
      Deviance = sapply(model_id, function(mn) get_deviance_original(mn, aic_obj,
       ↪  cand_list)),
      `R²`     = sapply(model_id, function(mn) summary(cand_list[[mn]])$r.squared),
      `slope (SE)` = sapply(model_id, function(mn)
       ↪  get_slope_se_single_weather(cand_list[[mn]], mn)),
      Group = factor(model_group(model_id), levels = group_order),
      Model = label_model(model_id),
      K = as.integer(K)
    )  |>
    mutate(
      Deviance = round(Deviance, 2),
      AICc     = round(AICc, 2),
      `ΔAICc`  = round(`ΔAICc`, 2),
      w        = round(w, 3),
      `R²`     = round(`R²`, 3)
    )  |>
    arrange(Group, `ΔAICc`)  |>
    select(Group, Model, K, Deviance, AICc, `ΔAICc`, w, `R²`, `slope (SE)`)
}

add_headers <- function(df, trend_label){
  df <- df  |>  mutate(Group = factor(Group, levels = group_order))

  groups_here <- levels(droplevels(df$Group))
  groups_here <- groups_here[groups_here %in% unique(as.character(df$Group))]

  empty_row <- df[0, ]
```

```r
  trend_row <- empty_row
  trend_row[1, ] <- NA
  trend_row$Group <- NA
  trend_row$Model <- trend_label

  out <- trend_row
  is_trend <- TRUE
  is_group <- FALSE

  for (g in groups_here) {
    gh <- empty_row
    gh[1, ] <- NA
    gh$Group <- NA
    gh$Model <- g

    rows_g <- df  |>  filter(as.character(Group) == g)  |>  mutate(Group = NA)

    out <- bind_rows(out, gh, rows_g)
    is_trend <- c(is_trend, FALSE, rep(FALSE, nrow(rows_g)))
    is_group <- c(is_group, TRUE,  rep(FALSE, nrow(rows_g)))
  }

  df_print <- out  |>  select(-Group)
  df_print[is.na(df_print)] <- ""

  list(
    df_print = df_print,
    trend_rows = which(is_trend),
    group_rows = which(is_group)
  )
}

tab_wy <- prep_aic_set(aic_effort_with_year)
tab_ny <- prep_aic_set(aic_effort_no_year)

df_wy <- build_df(tab_wy, cand_effort_with_year, aic_effort_with_year)
df_ny <- build_df(tab_ny, cand_effort_no_year,  aic_effort_no_year)

hdr_wy <- add_headers(df_wy, "a) Models including year trend")
hdr_ny <- add_headers(df_ny, "b) Models without year trend")

df_print_1a <- hdr_wy$df_print
df_print_1b <- hdr_ny$df_print


make_ft <- function(df_print, trend_rows, group_rows){
```

```r
  ft <- flextable(df_print)  |>
    font(fontname = "Times New Roman", part = "all")  |>
    fontsize(size = 12, part = "all")  |>
    bold(part = "header")  |>
    bg(part = "header", bg = "#f2f2f2")  |>
    align(j = 1, align = "left", part = "all")  |>
    align(j = 2:ncol(df_print), align = "center", part = "all")  |>
    bg(i = trend_rows, bg = "#d9d9d9", part = "body")  |>
    bold(i = trend_rows, part = "body")  |>
    bg(i = group_rows, bg = "#e6e6e6", part = "body")  |>
    bold(i = group_rows, part = "body")  |>
    autofit()

  ft <- flextable::compose(
    ft,
    i = trend_rows,
    j = 1,
    part = "body",
    value = as_paragraph(
      as_chunk(
        df_print[trend_rows, 1],
        props = fp_text(underline = TRUE, bold = TRUE)
      )
    )
  )

  ft
}

ft_1a <- make_ft(df_print_1a, hdr_wy$trend_rows, hdr_wy$group_rows)
ft_1b <- make_ft(df_print_1b, hdr_ny$trend_rows, hdr_ny$group_rows)

caption_1a <- paste0(
  "Table 1a. Model selection results relating to annual weather variation and the
  ↪  number of active nests at Dronfield (1993-2024) for models including a linear
  ↪  year effect (corresponding to Fig. 1a). ",
  "Models were fitted using linear regressions. A '+' indicates additive effects and
  ↪  '×' indicates an interaction. ",
  "K is the number of parameters; AICc is Akaike's Information Criterion; ΔAICc is
  ↪  relative to the best model within this model set; w is the Akaike weight; R² is
  ↪  variance explained. ",
  "Slopes (SE) are reported for simple predictor models only."
)

caption_1b <- paste0(
  "Table 1b. Model selection results relating to annual weather variation and the
  ↪  number of active nests at Dronfield (1993-2024) for models excluding a linear
  ↪  year effect (corresponding to Figs. 1b-c). ",
```

```
  "Models were fitted using linear regressions. A '+' indicates additive effects and
   ↪  '×' indicates an interaction. ",
  "K is the number of parameters; AICc is Akaike's Information Criterion; ΔAICc is
   ↪  relative to the best model within this model set; w is the Akaike weight; R² is
   ↪  variance explained. ",
  "Slopes (SE) are reported for simple predictor models only."
)

doc <- read_docx()  |>
  body_add_par(caption_1a, style = "Normal")  |>
  body_add_par("", style = "Normal")  |>
  body_add_flextable(ft_1a)  |>
  body_add_par("", style = "Normal")  |>
  body_add_par(caption_1b, style = "Normal")  |>
  body_add_par("", style = "Normal")  |>
  body_add_flextable(ft_1b)

print(doc, target = "Table_1a_1b_Active_Nests.docx")
```

0.3 Objective 2: Breeding success

0.3.1 Candidate model set

```
objective2_table <- tribble(
  ~`Hypothesis / justification`, ~Model,

  "**Temp**", "",
  "Extreme heat events cause chick heat stress and dehydration, reducing fledging
   ↪  success",
  "success ~ max_temp + (1 | year)",
  "Cold extremes increase thermoregulatory demand in chicks, increasing mortality
   ↪  risk",
  "success ~ min_temp + (1 | year)",

  "**Rain**", "",
  "Prolonged rainfall reduces adult foraging efficiency, limiting food delivery to
   ↪  chicks and reducing breeding success",
  "success ~ rain_days + (1 | year)",
  "Intense rainfall events cause acute nest disturbance and chick exposure, reducing
   ↪  breeding success",
  "success ~ max_rain + (1 | year)",
  "Overall wet years reduce provisioning efficiency across the breeding season,
   ↪  lowering breeding success",
  "success ~ total_rain + (1 | year)",
```

```r
  "**Wind**", "",
  "Extreme winds increase nest exposure and disrupt provisioning, reducing chick
   ↪  survival",
  "success ~ max_wind + (1 | year)",

  "**Hail**", "",
  "Severe hail events increase nest destruction and chick mortality, reducing
   ↪  breeding success",
  "success ~ hail_days + (1 | year)",

  "**Joint effects**", "",
  "Severe storm exposure increases nest destruction and chick mortality, reducing
   ↪  breeding success",
  "success ~ hail_days + max_wind + (1 | year)",
  "Prolonged wet conditions combined with storm events increase chick exposure and
   ↪  limit provisioning, reducing breeding success",
  "success ~ rain_days + hail_days + (1 | year)",
  "Heat stress combined with prolonged wet conditions reduces chick thermoregulation
   ↪  and food delivery, lowering breeding success",
  "success ~ max_temp + rain_days + (1 | year)",

  "**Interactive effects**", "",
  "The negative effect of heat stress on breeding success is amplified during
   ↪  persistently wet conditions that limit adult foraging",
  "success ~ max_temp * rain_days + (1 | year)",
  "Cold stress on chicks is amplified during extreme wind events due to increased
   ↪  exposure and heat loss, reducing breeding success",
  "success ~ min_temp * max_wind + (1 | year)"
)

knitr::kable(
  objective2_table,
  align = c("l", "l"),
  caption = "Objective 2 candidate model set for breeding success. All models are
   ↪  fitted as binomial GLMMs: cbind(success_nests, failed_nests) with a random
   ↪  intercept for year (1 | year)."
)
```

Table 2: Objective 2 candidate model set for breeding success. All models are fitted as binomial GLMMs: cbind(success_nests, failed_nests) with a random intercept for year (1 | year).

Hypothesis / justification	Model
Temp	
Extreme heat events cause chick heat stress and dehydration, reducing fledging success	success ~ max_temp + (1 | year)

Hypothesis / justification	Model
Cold extremes increase thermoregulatory demand in chicks, increasing mortality risk	success ~ min_temp + (1 | year)
Rain	
Prolonged rainfall reduces adult foraging efficiency, limiting food delivery to chicks and reducing breeding success	success ~ rain_days + (1 | year)
Intense rainfall events cause acute nest disturbance and chick exposure, reducing breeding success	success ~ max_rain + (1 | year)
Overall wet years reduce provisioning efficiency across the breeding season, lowering breeding success	success ~ total_rain + (1 | year)
Wind	
Extreme winds increase nest exposure and disrupt provisioning, reducing chick survival	success ~ max_wind + (1 | year)
Hail	
Severe hail events increase nest destruction and chick mortality, reducing breeding success	success ~ hail_days + (1 | year)
Joint effects	
Severe storm exposure increases nest destruction and chick mortality, reducing breeding success	success ~ hail_days + max_wind + (1 | year)
Prolonged wet conditions combined with storm events increase chick exposure and limit provisioning, reducing breeding success	success ~ rain_days + hail_days + (1 | year)
Heat stress combined with prolonged wet conditions reduces chick thermoregulation and food delivery, lowering breeding success	success ~ max_temp + rain_days + (1 | year)
Interactive effects	
The negative effect of heat stress on breeding success is amplified during persistently wet conditions that limit adult foraging	success ~ max_temp * rain_days + (1 | year)
Cold stress on chicks is amplified during extreme wind events due to increased exposure and heat loss, reducing breeding success	success ~ min_temp * max_wind + (1 | year)

0.3.2 Data used

```r
success_df <- vultures.clean |>
  dplyr::mutate(year = lubridate::year(ringing.date)) |>
  dplyr::filter(!is.na(year)) |>
  dplyr::group_by(year) |>
  dplyr::summarise(
    active_nests  = dplyr::n(),
    success_nests = sum(!is.na(laying.date)),
    failed_nests  = active_nests - success_nests,
    success_prop  = success_nests / active_nests,
    .groups       = "drop"
```

```r
  ) |>
  dplyr::filter(active_nests > 0) |>
  dplyr::left_join(weather_yearly, by = c("year" = "YEAR")) |>
  dplyr::mutate(
    year = factor(year),

    z_max_temp    = as.numeric(scale(max_temp)),
    z_min_temp    = as.numeric(scale(min_temp)),
    z_rain_days   = as.numeric(scale(rain_days)),
    z_max_rain    = as.numeric(scale(max_rain)),
    z_total_rain  = as.numeric(scale(total_rain)),
    z_max_wind    = as.numeric(scale(max_wind)),
    z_hail_days   = as.numeric(scale(hail_days))
  )

success_cc <- success_df |>
  dplyr::select(
    year, success_nests, failed_nests,
    z_max_temp, z_min_temp,
    z_rain_days, z_max_rain, z_total_rain,
    z_max_wind, z_hail_days
  ) |>
  tidyr::drop_na()

nrow(success_df)
```

```
[1] 32
```

```r
nrow(success_cc)
```

```
[1] 32
```

0.3.3 Checking overdispersion

```r
success_year_glm <- glm(
  cbind(success_nests, failed_nests) ~ year,
  family = binomial,
  data   = success_cc
)

summary(success_year_glm)
```

```
Call:
glm(formula = cbind(success_nests, failed_nests) ~ year, family = binomial,
    data = success_cc)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.8109     0.3469   2.337 0.019421 *
year1994     -0.2179     0.4408  -0.494 0.621128
year1995     -0.4745     0.4844  -0.979 0.327358
year1996     -0.5804     0.4322  -1.343 0.179326
year1997      0.3567     0.4626   0.771 0.440718
year1998     -0.1178     0.4342  -0.271 0.786200
year1999     -0.6286     0.4260  -1.476 0.140055
year2000     -0.2131     0.4368  -0.488 0.625683
year2001      0.3060     0.4406   0.695 0.487296
year2002     -1.2048     0.4220  -2.855 0.004306 **
year2003     -0.5232     0.4303  -1.216 0.224016
year2004     -0.7484     0.4277  -1.750 0.080148 .
year2005     -0.1398     0.4314  -0.324 0.745935
year2006     -0.4925     0.4175  -1.179 0.238221
year2007     -0.7655     0.4072  -1.880 0.060159 .
year2008     -0.6033     0.4085  -1.477 0.139685
year2009      0.1301     0.4553   0.286 0.775132
year2010     -1.1727     0.4048  -2.897 0.003768 **
year2011     -0.3756     0.4127  -0.910 0.362696
year2012     -1.5841     0.4258  -3.721 0.000199 ***
year2013     -0.9612     0.4131  -2.327 0.019975 *
year2014     -0.1919     0.4054  -0.473 0.635949
year2015     -0.5401     0.3994  -1.352 0.176351
year2016     -0.8311     0.4010  -2.073 0.038191 *
year2017     -0.6252     0.3971  -1.574 0.115429
year2018     -0.7224     0.3948  -1.830 0.067265 .
year2019     -0.4745     0.3981  -1.192 0.233311
year2020     -1.0986     0.4025  -2.729 0.006349 **
year2021     -1.0918     0.4096  -2.665 0.007691 **
year2022     -1.0838     0.3936  -2.754 0.005892 **
year2023     -0.6796     0.3915  -1.736 0.082613 .
year2024     -0.8689     0.3865  -2.248 0.024564 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance:  1.0769e+02  on 31  degrees of freedom
Residual deviance: -5.0404e-14  on  0  degrees of freedom
AIC: 216.3

Number of Fisher Scoring iterations: 3
```

```r
performance::check_overdispersion(success_year_glm)
```

```
# Overdispersion test

 dispersion ratio = 0.877
          p-value = 0.304
```

0.3.4 Fitting the models

```r
library(lme4)

success_df <- success_df |> dplyr::mutate(year = factor(year))

# Baseline (null) model

m0_success <- glmer(
  cbind(success_nests, failed_nests) ~ 1 + (1 | year),
  family = binomial,
  data = success_df
)

# Single predictor models

m_succ_maxT <- glmer(
  cbind(success_nests, failed_nests) ~ z_max_temp + (1 | year),
  family = binomial, data = success_df
)

m_succ_minT <- glmer(
  cbind(success_nests, failed_nests) ~ z_min_temp + (1 | year),
  family = binomial, data = success_df
)

m_succ_rainD <- glmer(
  cbind(success_nests, failed_nests) ~ z_rain_days + (1 | year),
  family = binomial, data = success_df
```

```r
)

m_succ_maxR <- glmer(
  cbind(success_nests, failed_nests) ~ z_max_rain + (1 | year),
  family = binomial, data = success_df
)

m_succ_totR <- glmer(
  cbind(success_nests, failed_nests) ~ z_total_rain + (1 | year),
  family = binomial, data = success_df
)

m_succ_maxW <- glmer(
  cbind(success_nests, failed_nests) ~ z_max_wind + (1 | year),
  family = binomial, data = success_df
)

m_succ_hail <- glmer(
  cbind(success_nests, failed_nests) ~ z_hail_days + (1 | year),
  family = binomial, data = success_df
)

# Joint effects

m_succ_hailW <- glmer(
  cbind(success_nests, failed_nests) ~ z_hail_days + z_max_wind + (1 | year),
  family = binomial, data = success_df
)

m_succ_rainH <- glmer(
  cbind(success_nests, failed_nests) ~ z_rain_days + z_hail_days + (1 | year),
  family = binomial, data = success_df
)

m_succ_heatR <- glmer(
  cbind(success_nests, failed_nests) ~ z_max_temp + z_rain_days + (1 | year),
  family = binomial, data = success_df
)

# Interactions

m_succ_int_heat_rain <- glmer(
  cbind(success_nests, failed_nests) ~ z_max_temp * z_rain_days + (1 | year),
  family = binomial, data = success_df
)

m_succ_int_cold_wind <- glmer(
  cbind(success_nests, failed_nests) ~ z_min_temp * z_max_wind + (1 | year),
```

```r
  family = binomial, data = success_df
)
```

0.3.5 AIC model selection

```r
cand_success <- list(
  null_re      = m0_success,

  max_temp     = m_succ_maxT,
  min_temp     = m_succ_minT,
  rain_days    = m_succ_rainD,
  max_rain     = m_succ_maxR,
  total_rain   = m_succ_totR,
  max_wind     = m_succ_maxW,
  hail_days    = m_succ_hail,

  hail_maxW    = m_succ_hailW,
  rain_hail    = m_succ_rainH,
  maxT_rainD   = m_succ_heatR,

  int_heatRain = m_succ_int_heat_rain,
  int_coldWind = m_succ_int_cold_wind
)

aic_success <- aictab(
  cand.set  = cand_success,
  modnames  = names(cand_success)
)

aic_success
```

```
Model selection based on AICc:

              K   AICc Delta_AICc AICcWt Cum.Wt      LL
min_temp      3 226.93       0.00   0.43   0.43 -110.03
null_re       2 229.45       2.52   0.12   0.56 -112.52
hail_days     3 230.45       3.53   0.07   0.63 -111.80
max_temp      3 230.85       3.92   0.06   0.69 -111.99
max_rain      3 231.01       4.09   0.06   0.75 -112.08
rain_days     3 231.61       4.68   0.04   0.79 -112.38
total_rain    3 231.65       4.72   0.04   0.83 -112.40
max_wind      3 231.73       4.81   0.04   0.87 -112.44
```

```
int_coldWind 5 232.04       5.11   0.03   0.91 -109.87
maxT_rainD   4 232.10       5.18   0.03   0.94 -111.31
rain_hail    4 232.30       5.38   0.03   0.97 -111.41
hail_maxW    4 232.67       5.74   0.02   0.99 -111.59
int_heatRain 5 234.93       8.01   0.01   1.00 -111.31
```

```r
vc_year <- function(mod) {
  vc <- as.data.frame(VarCorr(mod))
  vc$vcov[vc$grp == "year"][1]
}

v0     <- vc_year(m0_success)
v_minT <- vc_year(m_succ_minT)

prop_between_year_explained <- (v0 - v_minT) / v0
prop_between_year_explained
```

```
[1] 0.2098479
```

0.3.6 Best models

```r
aic_df_success <- as.data.frame(aic_success)

best_success_id <- if ("Modnames" %in% names(aic_df_success)) {
  aic_df_success$Modnames[1]
} else {
  rownames(aic_df_success)[1]
}

best_success_mod <- cand_success[[best_success_id]]

best_success_id
```

```
[1] "min_temp"
```

```r
summary(best_success_mod)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(success_nests, failed_nests) ~ z_min_temp + (1 | year)
```

```
   Data: success_df

      AIC       BIC    logLik -2*log(L)  df.resid
    226.1     230.5    -110.0     220.1        29

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.2181 -0.3386 -0.1177  0.5078  1.3408

Random effects:
 Groups Name        Variance Std.Dev.
 year   (Intercept) 0.1021   0.3195
Number of obs: 32, groups:  year, 32

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.21436    0.07017   3.055  0.00225 **
z_min_temp   0.16711    0.07222   2.314  0.02066 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
           (Intr)
z_min_temp -0.005
```

0.3.7 Adding population size to the best supported models:

```r
success_df <- success_df |>
  mutate(z_active_nests = scale(active_nests))
```

```r
m_succ_pop <- glmer(
  cbind(success_nests, failed_nests) ~ z_active_nests + (1 | year),
  family = binomial,
  data = success_df
)
```

```r
m_succ_minT_pop <- glmer(
  cbind(success_nests, failed_nests) ~ z_min_temp + z_active_nests + (1 | year),
  family = binomial,
  data = success_df
)
```

```r
summary(m_succ_pop)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(success_nests, failed_nests) ~ z_active_nests + (1 | year)
   Data: success_df

      AIC       BIC    logLik -2*log(L)  df.resid
    224.6     229.0    -109.3     218.6        29

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.5918 -0.3451  0.0927  0.3398  1.2347

Random effects:
 Groups Name        Variance Std.Dev.
 year   (Intercept) 0.09762  0.3124
Number of obs: 32, groups:  year, 32

Fixed effects:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.23572    0.06936   3.399 0.000677 ***
z_active_nests -0.18646    0.07062  -2.640 0.008286 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
z_actv_nsts -0.117
```

```r
summary(m_succ_minT)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(success_nests, failed_nests) ~ z_min_temp + (1 | year)
   Data: success_df

      AIC       BIC    logLik -2*log(L)  df.resid
    226.1     230.5    -110.0     220.1        29
```

```
Scaled residuals:
    Min      1Q  Median      3Q     Max 
-1.2181 -0.3386 -0.1177  0.5078  1.3408 

Random effects:
 Groups Name        Variance Std.Dev.
 year   (Intercept) 0.1021   0.3195  
Number of obs: 32, groups:  year, 32

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)   
(Intercept)  0.21436    0.07017   3.055  0.00225 **
z_min_temp   0.16711    0.07222   2.314  0.02066 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
           (Intr)
z_min_temp -0.005
```

```r
summary(m_succ_minT_pop)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(success_nests, failed_nests) ~ z_min_temp + z_active_nests +  
    (1 | year)
   Data: success_df

      AIC       BIC    logLik -2*log(L)  df.resid 
    218.5     224.4    -105.3     210.5        28 

Scaled residuals:
     Min       1Q   Median       3Q      Max 
-1.67994 -0.44118  0.07867  0.37818  1.53542 

Random effects:
 Groups Name        Variance Std.Dev.
 year   (Intercept) 0.0643   0.2536  
Number of obs: 32, groups:  year, 32
```

```
Fixed effects:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.23483    0.06137   3.826  0.00013 ***
z_min_temp      0.19152    0.06372   3.006  0.00265 **
z_active_nests -0.20533    0.06279  -3.270  0.00108 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) z_mn_t
z_min_temp   0.012
z_actv_nsts -0.147 -0.125
```

```r
aictab(list(
  min_temp      = m_succ_minT,
  min_temp_pop  = m_succ_minT_pop,
  pop_only      = m_succ_pop
))
```

```
Model selection based on AICc:

             K   AICc Delta_AICc AICcWt Cum.Wt      LL
min_temp_pop 4 220.02       0.00   0.91   0.91 -105.27
pop_only     3 225.45       5.44   0.06   0.97 -109.30
min_temp     3 226.93       6.91   0.03   1.00 -110.03
```

```r
summary(m_succ_minT_pop)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(success_nests, failed_nests) ~ z_min_temp + z_active_nests +
    (1 | year)
   Data: success_df

      AIC       BIC    logLik -2*log(L)  df.resid
    218.5     224.4    -105.3     210.5        28

Scaled residuals:
     Min       1Q   Median       3Q      Max
```

```
-1.67994 -0.44118  0.07867  0.37818  1.53542

Random effects:
 Groups Name        Variance Std.Dev.
 year   (Intercept) 0.0643   0.2536
Number of obs: 32, groups:  year, 32

Fixed effects:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.23483    0.06137   3.826  0.00013 ***
z_min_temp      0.19152    0.06372   3.006  0.00265 **
z_active_nests -0.20533    0.06279  -3.270  0.00108 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) z_mn_t
z_min_temp   0.012
z_actv_nsts -0.147 -0.125
```

```r
vc_year <- function(mod) {
  vc <- as.data.frame(VarCorr(mod))
  vc$vcov[vc$grp == "year"][1]
}

v0 <- vc_year(m0_success)
v1 <- vc_year(m_succ_minT_pop)

prop_between_year_explained <- (v0 - v1) / v0
prop_between_year_explained
```

```
[1] 0.5023207
```

```r
v_minT <- vc_year(m_succ_minT)

prop_minT <- (v0 - v_minT) / v0
prop_minT
```

```
[1] 0.2098479
```

```r
coef(summary(m_succ_minT_pop))["z_min_temp", c("Estimate", "Std. Error")]
```

```
  Estimate Std. Error 
0.19151634 0.06371546 
```

```r
v_pop <- vc_year(m_succ_pop)

prop_pop <- (v0 - v_pop) / v0
prop_pop
```

```
[1] 0.2443964
```

```r
vc_year <- function(mod) {
  vc <- as.data.frame(VarCorr(mod))
  out <- vc$vcov[vc$grp == "year"][1]
  if (length(out) == 0) NA_real_ else out
}

slope_se_single <- function(mod, digits = 3){
  fe <- lme4::fixef(mod)

  if (length(fe) == 2) {
    cf <- summary(mod)$coefficients
    paste0(
      round(cf[2, 1], digits),
      " (", round(cf[2, 2], digits), ")"
    )
  } else {
    ""
  }
}

annual_var_expl_factory <- function(null_mod){
  v0 <- vc_year(null_mod)
  function(mod){
    v_mod <- vc_year(mod)
    if (is.finite(v_mod) && is.finite(v0) && v0 > 0) max(0, (v0 - v_mod)/v0) else
     ↪  NA_real_
  }
}

aic_df <- as.data.frame(aic_success)

if ("Modnames" %in% names(aic_df)) {
  aic_df <- aic_df  |>  mutate(model_id = Modnames)
```

```
} else {
  aic_df <- tibble::rownames_to_column(aic_df, "model_id")
}

table_success <- aic_df  |>
  rename(`ΔAICc` = Delta_AICc, w = AICcWt)  |>
  select(model_id, K, AICc, `ΔAICc`, w)  |>
  arrange(`ΔAICc`)

var_expl_fun <- annual_var_expl_factory(m0_success)

stats_df <- table_success  |>
  mutate(
    `Between-year variance explained` =
      sapply(model_id, function(mn) var_expl_fun(cand_success[[mn]])),
    `slope (SE)` =
      sapply(model_id, function(mn) slope_se_single(cand_success[[mn]]))
  )

table_success2 <- table_success  |>
  left_join(stats_df  |>  select(model_id, `Between-year variance explained`, `slope
   ↪  (SE)`),
            by = "model_id")  |>
  mutate(
    AICc = round(AICc, 2),
    `ΔAICc` = round(`ΔAICc`, 2),
    w = round(w, 3),
    `Between-year variance explained` = round(`Between-year variance explained`, 3)
  )

group_order <- c("Baseline", "Temperature", "Rain", "Wind", "Hail", "Joint effects",
 ↪  "Interactions")

model_group_success <- function(m){
  dplyr::case_when(
    m %in% c("null_re") ~ "Baseline",
    m %in% c("max_temp", "min_temp") ~ "Temperature",
    m %in% c("rain_days", "max_rain", "total_rain") ~ "Rain",
    m %in% c("max_wind") ~ "Wind",
    m %in% c("hail_days") ~ "Hail",
    m %in% c("hail_maxW", "rain_hail", "maxT_rainD") ~ "Joint effects",
    m %in% c("int_heatRain", "int_coldWind") ~ "Interactions",
    TRUE ~ "Joint effects"
  )
}

pretty_model_success <- function(x){
  dplyr::recode(
```

```r
    x,
    "null_re"      = "Intercept only",
    "max_temp"     = "Maximum temperature",
    "min_temp"     = "Minimum temperature",
    "rain_days"    = "Rainy days",
    "max_rain"     = "Maximum daily rainfall",
    "total_rain"   = "Total rainfall",
    "max_wind"     = "Maximum wind speed",
    "hail_days"    = "Hail days",
    "hail_maxW"    = "Hail days + max wind speed",
    "rain_hail"    = "Rainy days + hail days",
    "maxT_rainD"   = "Max temperature + rainy days",
    "int_heatRain" = "Max temperature × rainy days",
    "int_coldWind" = "Min temperature × max wind speed",
    .default = x
  )
}

df <- table_success2  |>
  mutate(
    Group = factor(model_group_success(model_id), levels = group_order),
    Model = pretty_model_success(model_id)
  )  |>
  arrange(Group, `ΔAICc`)  |>
  select(Group, Model, K, AICc, `ΔAICc`, w, `Between-year variance explained`, `slope
   ↪  (SE)`)

groups <- levels(droplevels(df$Group))
groups <- groups[groups %in% unique(as.character(df$Group))]

make_group_block <- function(g) {
  header <- df[0, ]
  header[1, ] <- NA
  header$Group <- g
  header$Model <- g
  header$is_group <- TRUE

  rows <- df  |>  filter(as.character(Group) == g)
  rows$is_group <- FALSE

  bind_rows(header, rows)
}

df2 <- bind_rows(lapply(groups, make_group_block))
group_rows <- which(df2$is_group)

df_print <- df2  |>  select(-Group, -is_group)
```

```r
ft <- flextable(df_print)  |>
  font(fontname = "Times New Roman", part = "all")  |>
  fontsize(size = 12, part = "all")  |>
  bold(part = "header")  |>
  bg(part = "header", bg = "#f2f2f2")  |>
  align(j = 1, align = "left", part = "all")  |>
  align(j = 2:ncol(df_print), align = "center", part = "all")  |>
  bg(i = group_rows, bg = "#e6e6e6", part = "body")  |>
  bold(i = group_rows, part = "body")  |>
  autofit()

doc <- read_docx()  |>  body_add_flextable(ft)
print(doc, target = "Table_2_Breeding_Success_REBUILT.docx")
```

```r
sens_tbl <- tibble(
  Model = c(
    "Minimum temperature",
    "Population size",
    "Minimum temperature + population size"
  ),
  `ΔAICc` = c(6.91, 5.44, 0.00),
  `AIC weight` = c(0.03, 0.06, 0.91),
  `Between-year variance explained` = c(0.21, 0.24, 0.50)
)


ft <- flextable(sens_tbl)

ft <- ft |>
  font(fontname = "Times New Roman", part = "all") |>
  fontsize(size = 12, part = "all") |>
  bold(part = "header") |>
  bg(part = "header", bg = "#f2f2f2") |>
  align(j = 1, align = "left", part = "all") |>
  align(j = 2:ncol(sens_tbl), align = "center", part = "all") |>
  autofit()


doc <- read_docx() |>
  body_add_flextable(ft)

print(doc, target = "Sensitivity_Table.docx")
```

0.3.8 Combines table

```r
sens_tbl <- tibble(
  Model = c(
    "Minimum temperature",
    "Population size",
    "Minimum temperature + population size"
  ),
  `ΔAICc` = c(6.91, 5.44, 0.00),
  w = c(0.03, 0.06, 0.91),
  `Between-year variance explained` = c(0.21, 0.24, 0.50)
)

align_to_template <- function(x, template_names) {
  missing <- setdiff(template_names, names(x))
  if (length(missing) > 0) x[missing] <- NA
  x <- x[, template_names, drop = FALSE]
  x
}

variance_targets <- c(
  "Between-year variance explained",
  "Annual variance explained",
  "Variance explained",
  "Deviance explained"
)
target_var_col <- intersect(variance_targets, names(df_print))[1]

sens_tbl2 <- sens_tbl

if (!is.na(target_var_col) && target_var_col != "Between-year variance explained") {
  sens_tbl2 <- sens_tbl2  |>
    rename(!!target_var_col := `Between-year variance explained`)  |>
    select(-`Between-year variance explained`)
}

sens_tbl2 <- align_to_template(sens_tbl2, names(df_print))

sens_header <- df_print[0, ]
sens_header[1, ] <- NA
sens_header$Model <- "A posteriori models: population size"
sens_header$is_group <- TRUE

main_tbl <- df_print  |>  mutate(is_group = FALSE)
sens_tbl2 <- sens_tbl2  |>  mutate(is_group = FALSE)

df_final_print <- bind_rows(
  main_tbl,
```

```
  sens_header,
  sens_tbl2
) |>
  select(-is_group)

group_rows_combined <- c(group_rows, nrow(main_tbl) + 1)


ft <- flextable(df_final_print) |>
  font(fontname = "Times New Roman", part = "all") |>
  fontsize(size = 12, part = "all") |>
  bold(part = "header") |>
  bg(part = "header", bg = "#f2f2f2") |>
  align(j = 1, align = "left", part = "all") |>
  align(j = 2:ncol(df_final_print), align = "center", part = "all") |>
  bg(i = group_rows_combined, bg = "#e6e6e6", part = "body") |>
  bold(i = group_rows_combined, part = "body") |>
  autofit()

doc <- read_docx() |> body_add_flextable(ft)
print(doc, target = "Table_2_Breeding_Success_WITH_PostHoc.docx")
```

0.4 Objective 3

0.4.1 DAGS

0.4.2 Scenario A

```
dir.create("figures", showWarnings = FALSE)

export_dag_pdf <- function(dag, filename, w = 8, h = 4) {
  svg_txt <- DiagrammeRsvg::export_svg(dag)
  rsvg::rsvg_pdf(
    charToRaw(svg_txt),
    file = file.path("figures", filename),
    width = w, height = h
  )
  file.info(file.path("figures", filename))[, "size"]
}

timing_simple_dag <- DiagrammeR::grViz("
digraph timing_simple {
```

```
  graph [
    layout = neato,
    splines = false,
    overlap = false
  ]

  node [
    fontname = Helvetica,
    fontsize = 22,
    penwidth = 2,
    margin = 0.25
  ]

  edge [
    penwidth = 1.8,
    arrowsize = 0.55
  ]

  W     [label = 'W',  shape = box,    pos = '1.4,1.5!']

  T [label = 'T',
     shape = circle,
     fixedsize = true,
     width = 1,
     pos = '2.8,1.5!'
  ]

  S     [label = 'S',  shape = box,    pos = '1.4,0!']
  Tstar [label = 'T*', shape = box,    pos = '2.8,0!']

  W -> T
  T -> Tstar
  S -> Tstar
}
")

export_dag_pdf(timing_simple_dag, "timing_simple_dag.pdf", 8, 4)
```

```
[1] 6396
```

0.4.3 Scenario B

```
timing_B_dag <- DiagrammeR::grViz("
digraph timing_B {
  graph [
```

```
    layout = neato,
    splines = false,
    overlap = false
  ]

  node [
    fontname = Helvetica,
    fontsize = 22,
    penwidth = 2,
    margin = 0.25
  ]

  edge [
    penwidth = 1.8,
    arrowsize = 0.55
  ]

  W     [label = 'W',  shape = box,    pos = '1.0,1.5!']

  T [label = 'T',
     shape = circle,
     fixedsize = true,
     width = 1,
     pos = '2.8,1.5!'
  ]

  S     [label = 'S',  shape = box,    pos = '1.0,0!']
  Tstar [label = 'T*', shape = box,    pos = '2.8,0!']

  W -> T
  W -> S
  T -> Tstar
  S -> Tstar
}
")

export_dag_pdf(timing_B_dag, "timing_B_dag.pdf", 8, 4)
```

```
[1] 6445
```

0.4.4 Scenario C

```
timing_C_dag <- DiagrammeR::grViz("
digraph timing_C {
  graph [
```

```
    layout = neato,
    splines = false,
    overlap = false
  ]

  node [
    fontname = Helvetica,
    fontsize = 22,
    penwidth = 2,
    margin = 0.25
  ]

  edge [
    penwidth = 1.8,
    arrowsize = 0.55
  ]

  W     [label = 'W',  shape = box,    pos = '1.0,1.5!']

  T [label = 'T',
     shape = circle,
     fixedsize = true,
     width = 1,
     pos = '2.8,1.5!'
  ]

  S     [label = 'S',  shape = box,    pos = '1.0,0!']
  Tstar [label = 'T*', shape = box,    pos = '2.8,0!']

  W -> T
  T -> Tstar
  S -> Tstar
  T -> S
}
")

export_dag_pdf(timing_C_dag, "timing_C_dag.pdf", 8, 4)
```

```
[1] 6456
```

0.4.5 Scenario D

```
timing_D_dag <- DiagrammeR::grViz("
digraph timing_D {
  graph [
```

```
    layout = neato,
    splines = false,
    overlap = false
  ]

  node [
    fontname = Helvetica,
    fontsize = 22,
    penwidth = 2,
    margin = 0.25
  ]

  edge [
    penwidth = 1.8,
    arrowsize = 0.55
  ]

  W     [label = 'W',  shape = box,    pos = '1.0,1.5!']

  T [label = 'T',
     shape = circle,
     fixedsize = true,
     width = 1,
     pos = '2.8,1.5!'
  ]

  S     [label = 'S',  shape = box,    pos = '1.0,0!']
  Tstar [label = 'T*', shape = box,    pos = '2.8,0!']

  X [label = 'X',
     shape = circle,
     fixedsize = true,
     width = 0.9,
     pos = '2.0,0.75!'
  ]

  W -> T
  T -> Tstar
  S -> Tstar
  X -> T
  X -> S
}
")

export_dag_pdf(timing_D_dag, "timing_D_dag.pdf", 8, 4)
```

```
[1] 6804
```

0.5 Analysis

0.5.1 Candidate model set

```r
objective3_table <- tribble(
  ~`Hypothesis / justification`, ~Model,

  # Temperature (pre-laying cues)
  "**Temp (pre-laying)**", "",
  "Short-term warm conditions prior to laying may act as a cue for the initiation of
   ↪  breeding.",
  "lay_DOY ~ mean_temp_30",

  "Cold pre-laying conditions delay breeding due to increased energetic costs",
  "lay_DOY ~ min_temp_30",


  # Rain (pre-laying cues)
  "**Rain (pre-laying)**", "",
  "Prolonged wet conditions prior to laying delay breeding by limiting foraging
   ↪  efficiency",
  "lay_DOY ~ rain_days_30",

  "Higher rainfall prior to laying signals improved food availability and advances
   ↪  breeding timing",
  "lay_DOY ~ total_rain_30",

  # Wind (pre-laying flight conditions)
  "**Wind (pre-laying)**", "",
  "Persistent windy conditions prior to laying increase flight costs and delay
   ↪  breeding timing",
  "lay_DOY ~ mean_wind_30",

  # Lagged effects (carry-over cues)
  "**Lagged effects (previous year)**", "",
  "Higher rainfall in the previous year improves food availability and adult
   ↪  condition entering the breeding season, allowing vultures to initiate breeding
   ↪  earlier",
  "lay_DOY ~ lag_total_rain",

  "Prolonged wet conditions in the previous year may influence adult condition and
   ↪  carry over to affect laying timing",
  "lay_DOY ~ lag_rain_days",

  "Thermal conditions in the previous year influence physiological state entering the
   ↪  breeding season",
  "lay_DOY ~ lag_mean_temp",
```

```
  #Joint effects
  "**Joint effects**", "",
  "Temperature and rainfall cues prior to laying jointly influence breeding timing",
  "lay_DOY ~ mean_temp_30 + rain_days_30",

  "Carry-over effects from the previous year interact with current pre-laying
   ↪  rainfall cues",
  "lay_DOY ~ lag_total_rain + rain_days_30",

  "Previous-year rainfall and current temperature jointly influence breeding timing",
  "lay_DOY ~ lag_total_rain + mean_temp_30",

  # Interactive effects
  "**Interactive effects**", "",
  "The effect of temperature cues on breeding timing depends on pre-laying rainfall
   ↪  conditions",
  "lay_DOY ~ mean_temp_30 * rain_days_30",

  "Carry-over effects from the previous year modify responses to current pre-laying
   ↪  temperature cues",
  "lay_DOY ~ lag_total_rain * mean_temp_30"
)

kable(
  objective3_table,
  align = c("l", "l"),
  caption = "Objective 3 candidate model set for breeding timing (laying date, day of
   ↪  year)."
)
```

Table 3: Objective 3 candidate model set for breeding timing (laying date, day of year).

Hypothesis / justification	Model
Temp (pre-laying)	
Short-term warm conditions prior to laying may act as a cue for the initiation of breeding.	lay_DOY ~ mean_temp_30
Cold pre-laying conditions delay breeding due to increased energetic costs	lay_DOY ~ min_temp_30
Rain (pre-laying)	
Prolonged wet conditions prior to laying delay breeding by limiting foraging efficiency	lay_DOY ~ rain_days_30
Higher rainfall prior to laying signals improved food availability and advances breeding timing	lay_DOY ~ total_rain_30
Wind (pre-laying)	

Hypothesis / justification	Model
Persistent windy conditions prior to laying increase flight costs and delay breeding timing	lay_DOY ~ mean_wind_30
Lagged effects (previous year)	
Higher rainfall in the previous year improves food availability and adult condition entering the breeding season, allowing vultures to initiate breeding earlier	lay_DOY ~ lag_total_rain
Prolonged wet conditions in the previous year may influence adult condition and carry over to affect laying timing	lay_DOY ~ lag_rain_days
Thermal conditions in the previous year influence physiological state entering the breeding season	lay_DOY ~ lag_mean_temp
Joint effects	
Temperature and rainfall cues prior to laying jointly influence breeding timing	lay_DOY ~ mean_temp_30 + rain_days_30
Carry-over effects from the previous year interact with current pre-laying rainfall cues	lay_DOY ~ lag_total_rain + rain_days_30
Previous-year rainfall and current temperature jointly influence breeding timing	lay_DOY ~ lag_total_rain + mean_temp_30
Interactive effects	
The effect of temperature cues on breeding timing depends on pre-laying rainfall conditions	lay_DOY ~ mean_temp_30 * rain_days_30
Carry-over effects from the previous year modify responses to current pre-laying temperature cues	lay_DOY ~ lag_total_rain * mean_temp_30

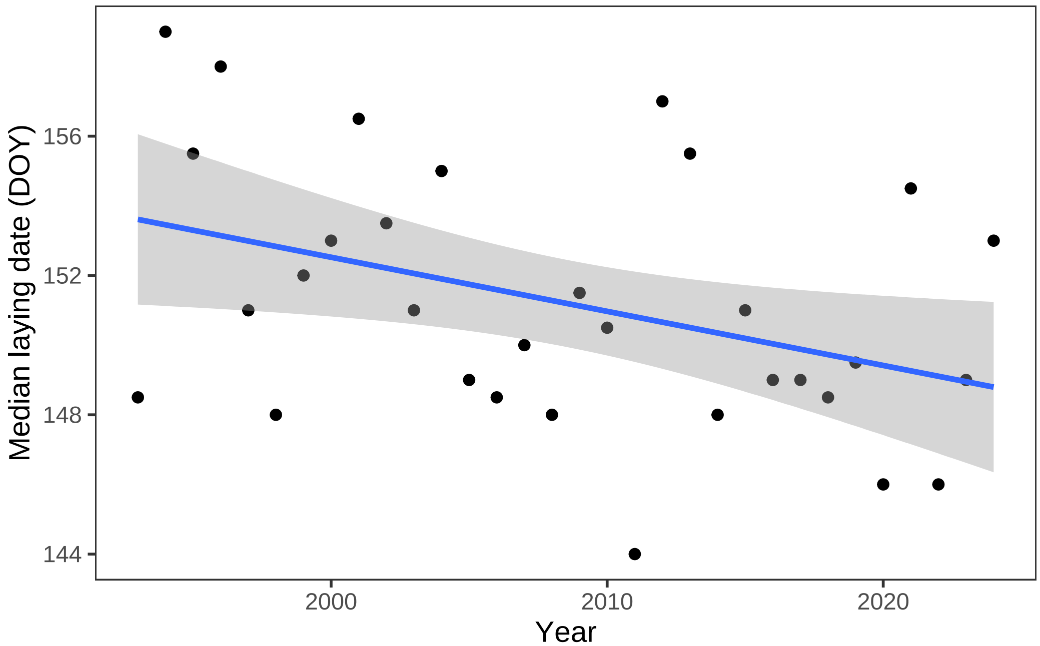
0.5.2 Timing data used

```
timing_df <- vultures.clean |>
  dplyr::filter(!is.na(laying.date)) |>
  dplyr::mutate(
    year    = as.integer(lubridate::year(laying.date)),
    lay_DOY = lubridate::yday(laying.date)
  ) |>
  dplyr::filter(!is.na(year), year != 1900)

# One value per year
timing_by_year <- timing_df |>
```

```r
  dplyr::group_by(year) |>
  dplyr::summarise(
    lay_DOY_median = median(lay_DOY, na.rm = TRUE),
    lay_DOY_mean   = mean(lay_DOY, na.rm = TRUE),
    n_nests        = sum(!is.na(lay_DOY)),
    first_DOY      = min(lay_DOY, na.rm = TRUE),
    last_DOY       = max(lay_DOY, na.rm = TRUE),
    .groups = "drop"
  ) |>
  dplyr::arrange(year)

timing_by_year
```

```
# A tibble: 32 x 6
    year lay_DOY_median lay_DOY_mean n_nests first_DOY last_DOY
   <int>          <dbl>        <dbl>   <int>     <dbl>    <dbl>
 1  1993           148.         148.      28       103      203
 2  1994           159          162.      39       128      212
 3  1995           156.         154.      22       132      171
 4  1996           158          158.      35       129      209
 5  1997           151          152.      46       132      190
 6  1998           148          151.      45       122      188
 7  1999           152          153.      37       136      192
 8  2000           153          155.      41       124      201
 9  2001           156.         158.      56       129      208
10  2002           154.         155.      30        20      185
# i 22 more rows
```

```r
ggplot(timing_by_year, aes(x = year, y = lay_DOY_median)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(x = "Year", y = "Median laying date (DOY)") +
  theme_bw() +
  theme(panel.grid = element_blank())
```




```
obj3_year_df <- timing_by_year |>
  left_join(weather_march_by_year, by = "year") |>
  left_join(
    weather_annual_by_year |>
      dplyr::select(year,
                    lag_mean_temp_annual,
                    lag_total_rain_annual,
                    lag_rain_days_annual),
    by = "year"
  ) |>
  arrange(year)

obj3_year_df
```

```
# A tibble: 32 x 19
    year lay_DOY_median lay_DOY_mean n_nests first_DOY last_DOY mean_temp_win
   <dbl>          <dbl>        <dbl>   <int>     <dbl>    <dbl>         <dbl>
 1  1993           148.         148.      28       103      203          21.0
 2  1994           159          162.      39       128      212          21.7
 3  1995           156.         154.      22       132      171          22.1
 4  1996           158          158.      35       129      209          22.3
 5  1997           151          152.      46       132      190          19.8
 6  1998           148          151.      45       122      188          22.2
 7  1999           152          153.      37       136      192          25.4
```

```
 8  2000            153           155.      41       124      201           22.3
 9  2001            156.          158.      56       129      208           23.2
10  2002            154.          155.      30        20      185           23.5
# i 22 more rows
# i 12 more variables: max_temp_win <dbl>, min_temp_win <dbl>,
#   total_rain_win <dbl>, max_rain_win <dbl>, rain_days_win <int>,
#   mean_wind_win <dbl>, max_wind_win <dbl>, hail_days_win <dbl>, n_days <int>,
#   lag_mean_temp_annual <dbl>, lag_total_rain_annual <dbl>,
#   lag_rain_days_annual <int>
```

v ‘

```
timing_by_year <- timing_df |>
  dplyr::filter(!is.na(year), year != 1900) |>
  dplyr::mutate(year = as.integer(year)) |>
  dplyr::group_by(year) |>
  dplyr::summarise(
    lay_DOY_median = median(lay_DOY, na.rm = TRUE),
    lay_DOY_mean   = mean(lay_DOY, na.rm = TRUE),
    n_nests        = sum(!is.na(lay_DOY)),
    first_DOY      = min(lay_DOY, na.rm = TRUE),
    last_DOY       = max(lay_DOY, na.rm = TRUE),
    .groups = "drop"
  ) |>
  dplyr::arrange(year)
```

```
vars_obj3 <- c(
  "year",
  "lay_DOY_median",
  "mean_temp_win", "min_temp_win",
  "rain_days_win", "total_rain_win",
  "mean_wind_win",
  "lag_mean_temp_annual",
  "lag_total_rain_annual",
  "lag_rain_days_annual"
)

obj3_cc <- obj3_year_df |>
  dplyr::select(all_of(vars_obj3)) |>
  tidyr::drop_na()
```

```
nrow(obj3_year_df)
```

```
[1] 32
```

```r
summary(obj3_year_df$lay_DOY_median)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  144.0   148.5   150.8   151.2   153.8   159.0 
```

```r
summary(obj3_year_df$mean_temp_win)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  18.70   21.40   22.03   22.10   22.94   25.41 
```

```r
cand_obj3 <- list(
  null = lm(lay_DOY_median ~ 1, data = obj3_cc),

  meanT = lm(lay_DOY_median ~ mean_temp_win, data = obj3_cc),
  minT  = lm(lay_DOY_median ~ min_temp_win,  data = obj3_cc),

  rainD = lm(lay_DOY_median ~ rain_days_win,  data = obj3_cc),
  totR  = lm(lay_DOY_median ~ total_rain_win, data = obj3_cc),

  wind  = lm(lay_DOY_median ~ mean_wind_win, data = obj3_cc),

  lagMeanT = lm(lay_DOY_median ~ lag_mean_temp_annual,  data = obj3_cc),
  lagTotR  = lm(lay_DOY_median ~ lag_total_rain_annual, data = obj3_cc),
  lagRainD = lm(lay_DOY_median ~ lag_rain_days_annual,  data = obj3_cc),

  cue_plus_lagR =
    lm(lay_DOY_median ~ mean_temp_win + rain_days_win + lag_total_rain_annual,
       data = obj3_cc),

  lagR_plus_cueRain =
    lm(lay_DOY_median ~ lag_total_rain_annual + rain_days_win,
       data = obj3_cc),

  int_TR =
    lm(lay_DOY_median ~ mean_temp_win * rain_days_win,
       data = obj3_cc),

  lagR_x_meanT =
    lm(lay_DOY_median ~ lag_total_rain_annual * mean_temp_win,
       data = obj3_cc)
)

aictab(cand.set = cand_obj3, modnames = names(cand_obj3))
```

```
Model selection based on AICc:

                  K   AICc Delta_AICc AICcWt Cum.Wt     LL
lagTotR           3 173.52       0.00   0.36   0.36 -83.33
lagR_plus_cueRain 4 174.38       0.86   0.23   0.59 -82.45
cue_plus_lagR     5 175.82       2.30   0.11   0.70 -81.76
lagR_x_meanT      5 176.21       2.69   0.09   0.79 -81.95
wind              3 178.02       4.50   0.04   0.83 -85.58
minT              3 178.06       4.54   0.04   0.87 -85.60
null              2 178.10       4.58   0.04   0.91 -86.84
meanT             3 178.49       4.97   0.03   0.93 -85.82
rainD             3 179.61       6.09   0.02   0.95 -86.38
lagRainD          3 179.77       6.25   0.02   0.97 -86.46
lagMeanT          3 180.01       6.49   0.01   0.98 -86.58
totR              3 180.10       6.58   0.01   0.99 -86.62
int_TR            5 181.87       8.35   0.01   1.00 -84.78
```

0.5.3 Adding population size to best fit model

```
obj3_pop_df <- obj3_cc |>
  left_join(
    success_df |>
      mutate(year = as.integer(as.character(year))) |>
      group_by(year) |>
      summarise(active_nests = first(active_nests), .groups = "drop"),
    by = "year"
  ) |>
  mutate(z_active_nests = as.numeric(scale(active_nests)))
```

```
summary(obj3_pop_df$active_nests)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  36.00   63.75   78.00   82.03   99.25  138.00 
```

```
summary(obj3_pop_df$z_active_nests)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-1.9057 -0.7568 -0.1669  0.0000  0.7129  2.3171 
```

```r
m_lagR     <- lm(lay_DOY_median ~ lag_total_rain_annual, data = obj3_pop_df)
m_pop      <- lm(lay_DOY_median ~ z_active_nests, data = obj3_pop_df)
m_lagR_pop <- lm(lay_DOY_median ~ lag_total_rain_annual + z_active_nests,
                 data = obj3_pop_df)

aictab(cand.set = list(
  lagR = m_lagR,
  pop = m_pop,
  lagR_plus_pop = m_lagR_pop
))
```

```
Model selection based on AICc:

              K   AICc Delta_AICc AICcWt Cum.Wt     LL
lagR_plus_pop 4 170.81       0.00   0.72   0.72 -80.67
lagR          3 173.52       2.71   0.19   0.91 -83.33
pop           3 174.95       4.14   0.09   1.00 -84.05
```

```r
dev_m2ll <- function(mod) -2 * as.numeric(logLik(mod))
get_K    <- function(mod) attr(logLik(mod), "df")
r2_fun   <- function(mod) summary(mod)$r.squared

slope_se_single <- function(mod, digits = 3){
  cf <- summary(mod)$coefficients
  if (nrow(cf) == 2) paste0(round(cf[2,1], digits), " (", round(cf[2,2], digits),
   ↪  ")") else ""
}

build_table <- function(model_list){
  bind_rows(lapply(names(model_list), function(nm){
    mod <- model_list[[nm]]
    tibble(
      model_id = nm,
      K = get_K(mod),
      Deviance = dev_m2ll(mod),
      AICc = AICc(mod),
      `R²` = r2_fun(mod),
      `slope (SE)` = slope_se_single(mod)
    )
  })) |>
    arrange(AICc) |>
    mutate(
      `ΔAICc` = AICc - min(AICc, na.rm = TRUE),
      w = exp(-0.5 * `ΔAICc`) / sum(exp(-0.5 * `ΔAICc`)),
      Deviance = round(Deviance, 2),
```

```r
      AICc = round(AICc, 2),
      `ΔAICc` = round(`ΔAICc`, 2),
      w = round(w, 3),
      `R²` = round(`R²`, 3)
    )
}

model_labels <- tribble(
  ~model_id, ~Group, ~Model,

  "null", "Baseline", "Intercept only",

  "meanT", "Temperature", "Mean temperature (March)",
  "minT",  "Temperature", "Minimum temperature (March)",

  "rainD", "Rain", "Rainy days (March)",
  "totR",  "Rain", "Total rainfall (March)",

  "wind",  "Wind", "Mean wind speed (March)",

  "lagMeanT", "Lagged effects", "Previous year mean temperature",
  "lagTotR",  "Lagged effects", "Previous year total rainfall",
  "lagRainD", "Lagged effects", "Previous year rainy days",

  "cue_plus_lagR",     "Joint effects", "March cues + previous year rainfall",
  "lagR_plus_cueRain", "Joint effects", "Previous year rainfall + March rainy days",

  "int_TR",       "Interactions", "Mean temperature × rainy days (March)",
  "lagR_x_meanT", "Interactions", "Lag rainfall × March mean temperature"
)

group_order <- c(
  "Baseline",
  "Temperature",
  "Rain",
  "Wind",
  "Lagged effects",
  "Joint effects",
  "Interactions",
  "A posteriori models: population size",
  "Other"
)

main_df <- build_table(cand_obj3)

df <- main_df |>
  left_join(model_labels, by = "model_id") |>
  mutate(
```

```r
    Group = coalesce(Group, "Other"),
    Model = coalesce(Model, model_id),
    Group = factor(Group, levels = group_order)
  ) |>
  arrange(Group, `ΔAICc`) |>
  select(Group, Model, K, Deviance, AICc, `ΔAICc`, w, `R²`, `slope (SE)`)

if (exists("m_lagR") && exists("m_pop") && exists("m_lagR_pop")) {

  posthoc_models <- list(
    lagR = m_lagR,
    pop  = m_pop,
    lagR_plus_pop = m_lagR_pop
  )

  posthoc_df <- build_table(posthoc_models) |>
    mutate(
      Group = "A posteriori models: population size",
      Model = dplyr::recode(
        model_id,
        "lagR" = "Previous year total rainfall",
        "pop"  = "Number of active nests",
        "lagR_plus_pop" = "Previous year rainfall + number of active nests",
        .default = model_id
      ),
      Group = factor(Group, levels = group_order)
    ) |>
    arrange(`ΔAICc`) |>
    select(Group, Model, K, Deviance, AICc, `ΔAICc`, w, `R²`, `slope (SE)`)

  df <- bind_rows(df, posthoc_df) |>
    arrange(Group, `ΔAICc`)
}

# group header rows
groups_present <- df |>
  mutate(Group_chr = as.character(Group)) |>
  pull(Group_chr) |>
  unique()

make_group_block <- function(g) {
  header <- df[0, ]
  header[1, ] <- NA
  header$Group <- g
  header$Model <- g
  header$is_group <- TRUE

  rows <- df |> filter(as.character(Group) == g)
```

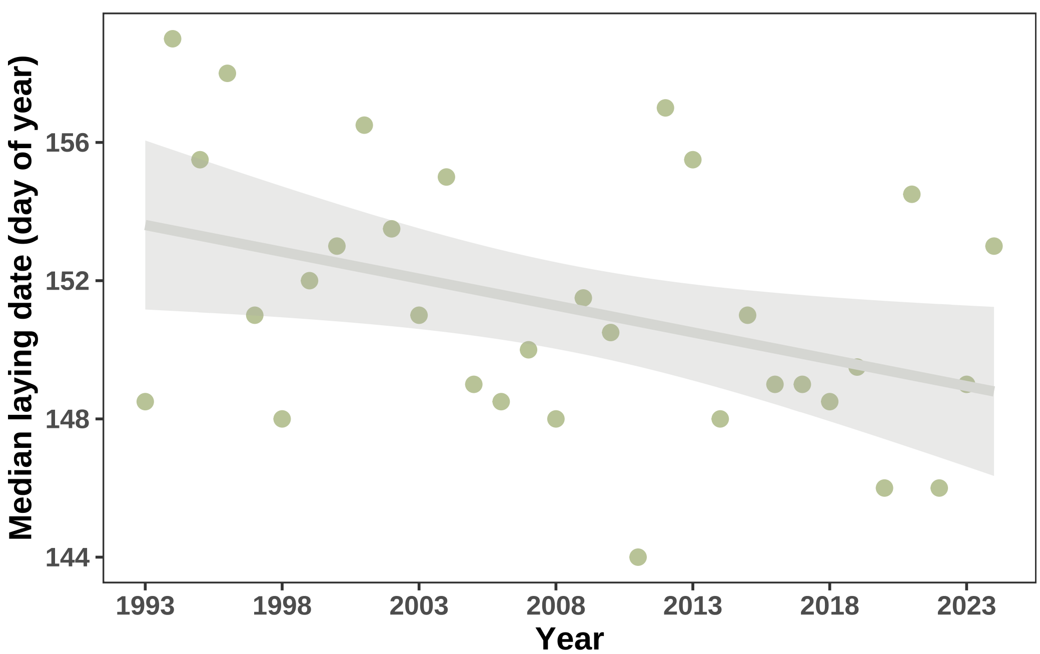
```r
  rows$is_group <- FALSE

  bind_rows(header, rows)
}

df2 <- bind_rows(lapply(groups_present, make_group_block))
group_rows <- which(df2$is_group)
df_print <- df2 |> select(-Group, -is_group)

ft <- flextable(df_print) |>
  font(fontname = "Times New Roman", part = "all") |>
  fontsize(size = 12, part = "all") |>
  bold(part = "header") |>
  bg(part = "header", bg = "#f2f2f2") |>
  align(j = 1, align = "left", part = "all") |>
  align(j = 2:ncol(df_print), align = "center", part = "all") |>
  bg(i = group_rows, bg = "#e6e6e6", part = "body") |>
  bold(i = group_rows, part = "body") |>
  autofit()

for (r in group_rows) {
  ft <- merge_at(ft, i = r, j = 1:ncol(df_print), part = "body")
  ft <- align(ft, i = r, j = 1, align = "left", part = "body")
}

doc <- read_docx() |>
  body_add_par(
    paste0(
    ),
    style = "Normal"
  ) |>
  body_add_flextable(ft)

print(doc, target = "Table_Obj3_Timing_OrderedLikeObj2.docx")
cat("Saved: Table_Obj3_Timing_OrderedLikeObj2.docx\n")
```

```
Saved: Table_Obj3_Timing_OrderedLikeObj2.docx
```

0.5.4 Figures

```r
ggplot(obj3_cc, aes(x = year, y = lay_DOY_median)) +
  geom_point(
    size = 2.4,
    colour = "#B1BD8C",
    alpha = 0.9
```

```r
) +
geom_smooth(
  method = "lm",
  se = TRUE,
  colour = "#D5D6D2",
  fill   = "#A9AAA7",
  alpha  = 0.25,
  linewidth = 1.8
) +
scale_x_continuous(
  breaks = seq(
    min(obj3_cc$year),
    max(obj3_cc$year),
    by = 5
  )
) +
labs(
  x = "Year",
  y = "Median laying date (day of year)"
) +
theme_bw(base_family = "Times New Roman") +
theme(
  panel.grid = element_blank(),
  axis.title = element_text(face = "bold", size = 12),
  axis.text  = element_text(face = "bold", size = 10)
)
```



```
ggplot(obj3_cc, aes(x = lag_total_rain_annual, y = lay_DOY_median)) +
  geom_point(
    size = 2.4,
    colour = "#B1BD8C",
    alpha = 0.9
  ) +
  geom_smooth(
    method = "lm",
    se = TRUE,
    colour = "#D5D6D2",
    fill   = "#A9AAA7",
    alpha  = 0.25,
    linewidth = 1.8
  ) +
  labs(
    x = "Previous-year total rainfall (mm)",
    y = "Median laying date (day of year)"
  ) +
  theme_bw(base_family = "Times New Roman") +
  theme(
    panel.grid = element_blank(),
    axis.title = element_text(face = "bold", size = 12),
    axis.text  = element_text(face = "bold", size = 10)
  )
```