STAT 5214G: Methods of Regression: Final Individual Mini-Project

Cheyenne Erickson

11/28/25

Summary/Introduction

Introduction

Alpine butterfly (Parnassius smintheus) populations fluctuate substantially from year to year, and previous ecological work suggests that both winter conditions and summer temperature extremes may influence survival and reproduction. The goal of this analysis is to identify which weather variables most strongly affect annual population size and to develop a model capable of predicting population abundance across meadows.

The data come from long-term monitoring of 11 meadows in the Canadian Rockies from 1995–2015, collected using mark–release–recapture methods (Roland et al.). Population counts (Nt) were paired with monthly weather summaries, allowing exploration of how winter snow, temperature extremes, and past population size contribute to future dynamics.

The guiding questions are:

- Which weather variables best explain variation in butterfly population size?
- How strongly does previous population size (Nt–1) influence current population abundance?
- Can a statistical model using weather and lagged population size accurately forecast Nt in an unseen meadow or future year?

To address these questions, I conducted exploratory data analysis, built and compared at least four competing regression models (including hypothesis-driven models and stepwise-selected models), evaluated model diagnostics and goodness of fit, and assessed predictive performance using both leave-one-meadow-out validation and a 2015 out-of-sample forecast.

Summary

This project used 21 years of population and weather data from 11 alpine meadows to model the annual abundance of Parnassius smintheus. Four regression models were constructed and compared, including both stepwise-selected models and hypothesis-driven models emphasizing overwinter snow, spring snowmelt, previous population size, and summer temperature extremes. Exploratory analysis revealed strong temporal autocorrelation and heteroscedasticity, motivating the use of an AR(1) term and a square-root transformation of Nt.

Model comparison using partial F-tests, BIC, and leave-one-meadow-out validation identified a square-root linear model containing Nt–1, marsnow, aprsnowfall, julextmax, and their interaction as the best-performing model. Diagnostic checks showed no major violations of linear model assumptions. Using this final model, forecasts for 2015 across all meadows achieved an RMSE of approximately 111 butterflies, indicating moderate predictive accuracy given the wide range of observed population sizes.

Overall, the analysis suggests that previous population size, winter and early-spring snow conditions, and extreme July temperatures jointly shape butterfly population dynamics, and that the final model provides a reasonable basis for forecasting abundance in future years.

Exploratory Analysis

Working Universe of Variables:

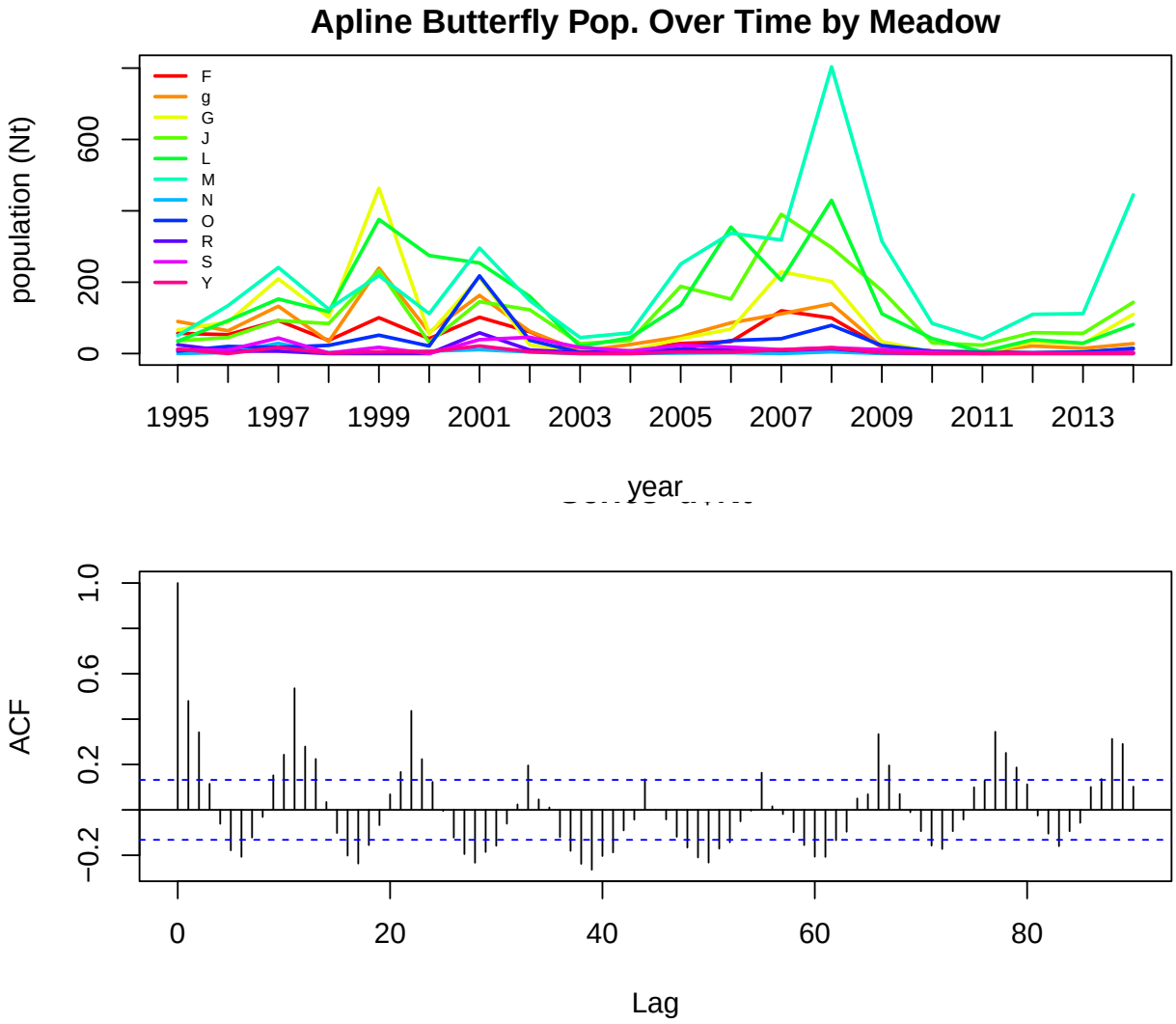
- meada (meadows)
- AR(1)
- AR(2)
- maxsnow
- Datesnowmelt
- novsnow
- decsnow
- jansnow
- marsnow
- janextmin
- febextmin
- marextmin
- aprsnow
- aprsnowfall
- maymean
- mayextmin
- junmean
- junextmax
- julsnow
- julmeanmax
- julextmax
- augmean
- augextmax
- sepmean
- sepextmin

With research (see sources), I learned several key characteristics of the alpine butterfly Parnassius smintheus that help guide my selection of weather predictors and inform how weather should influence the population outcomes in this system:

- Unlike many butterflies, Adult Apline butterflies emerge only in mid-summer and live a 2-3 week life. During July–August they mate and lay eggs near their larval host plant. All adults die before winter, so any weather that happens after August affects the next generation, not the current adults.
- After adults lay eggs in August, the eggs hatch within a few weeks. The tiny larvae enter diapause early (usually by late September) and remain dormant beneath the snow until the following spring. When spring arrives and snow begins to melt, larvae resume feeding. They continue feeding through April–June until it is time to pupate in June/July. Adults emerge again in late July–August, completing the life cycle.
- Because larvae overwinter and must complete development in spring and early summer, the population is extremely sensitive to winter, spring, and summer climate conditions.
- Winter (Dec-Mar): Snowpack provides insulation for overwintering larvea.
  - Too little snow: larvae are exposed to freezing temperatures -> increased mortality (negative effect).
  - Too much snow combined with cold spring: delays snowmelt -> shortens spring feeding window (negative).
- Spring (Mar-May): Feeding window and Growth period
  - Warm/early springs: earlier melt -> longer feeding period -> stronger adults (positive)
  - Cold/late springs: delayed melt -> rushed feeding period -> weak or smaller butterflies (negative).
  - Late frosts: can kill exposed larvae (negative).
- Early Summer (Jun-Jul): Larvae pupate in June and July, and pupae cannot move or regulate temperature.
  - Extreme heat: can kill late larvae or pupae (negative).
  - Dry summers: reduce host plant quality -> weaker larvae -> reduced adult survival (negative).
  - Heavy storms: can physically damage pupae or disrupt emergence.
- I noted that weather can differ substantially between meadows which will motivate my meadow grouping during the EDA phase of this project. Although, I do understand that the second half of this project will be aimed at making a general model that can be applied to any meadow.

Based on the nature of the Apline butterfly I suspect that the number of butterfies in year k will depend on the number of butterflies in the year k-1. I will first start my EDA to explore this. Simultaneously, I will explore how the number of butterflies over time changes for each meadow.

```
par(mfrow = c(2, 1), mar = c(4, 4, 2, 1))

meadows <- levels(d$meada)
n_meadows <- length(meadows)


cols <- rainbow(n_meadows)

# Empty plot
plot(d$year, d$Nt,
     type = "n",
     xlab = "year",
     ylab = "population (Nt)",
     xaxt = "n",
     main = 'Apline Butterfly Pop. Over Time by Meadow')

# Custom x-axis
axis(1,
     at = sort(unique(d$year)),
     labels = sort(unique(d$year)))

# each meadow as a colored line
for (i in seq_along(meadows)) {
  dat <- subset(d, meada == meadows[i])
  lines(dat$year, dat$Nt,
        col = cols[i],
        lwd = 2)
}

legend("topleft",
       legend = meadows,
       col = cols,
       lwd = 2,
       cex = 0.6,
       bty = "n")
acf(d$Nt, lag.max = 90)
```



Here I notice that some meadows (Y and N) have an extremely low population (close to zero) over the whole experiment. It is possible that these meadows do not have sufficient host-plants growing there to support the Apline butterfly life-cycle, or that the terrain/weather of that meadow does not support Apline butterfly life. Additionally, It is notable that the number of butterflies differs greatly depending on the meadow. This brings up the question, "Is this difference significant?". To answer this I can do a simple ANOVA test. If so, I will add the factor 'Meadow' into my universe of variables. Most meadows have a fluctuating Population of butterflies over time (up <-> down) but I do not see an overall up or downward trend. Another highlight is that the meadows share peaks and troughs in many instances (2008, 1999, 1997, 2001, 2008, 1998, 2000, etc.). This hints at possibly periodicity, common environmental drivers, and shared climate signals.

The ACF plot shows that there is autocorrelation in the data. This supports the idea that population count of the previous year effects the population count of the current year. Therefore, I will add an AR variable into my universe of variables.

ANOVA testing for significant difference in butterfly counts grouped by meadow:

```
anova_result <- aov(Nt ~ meada, data = d)
summary(anova_result)
```

```
##              Df  Sum Sq Mean Sq F value Pr(>F)
## meada        10  944911   94491   12.91 <2e-16 ***
```

```
## Residuals   209 1530197    7322
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on this ANOVA test the answer to my question, "Is the difference in butterfly counts between meadows significant?", is yes! the factor variable 'meada' will be added to my universe of variables. Because meadow identity captures baseline differences among sites (e.g., habitat quality), including meadow as a factor improves the mechanism-based model structure. Later model comparisons will determine whether it is ultimately retained.

Note to add: ANOVA assumes independence, which is technically violated in time-series data, the test here is exploratory rather than inferential. The strong significance still suggests meaningful differences among meadows

Possible Transformations: Because meadows differ dramatically in population scale, transformations (e.g., Sqrt(Nt)) may improve model stability if the AR factor does not already.

Now I will work on expanding my Universe of variables starting by verifying the weather conditions I suspect to be highly linearly predictive in the number of butterflies based on the key characteristics highlighted in the beginning of this section.

The variables with associated biology reason in question are:

"big picture" snow varibales.

- maxsnow : insulation vs. deep snow delaying melt
- Datesnowmelt : directly tied to length of the spring feeding season

Winter (Nov-Mar): overwintering larvae under snow

- novsnow
- decsnow
- jansnow
- marsnow

I chose these four months in hopes that they give early winter snow onset and mid/late winter snow cover without taking every possible snow/snowfall variable. Together with maxsnow, they can cover both typical and extreme snow conditions.

- janextmin (extreme minimum in mid-winter)
- febextmin (potential late winter cold front)
- marextmin (early spring cold)

These three months are used to detect extreme colds that could be lethal to larvae despite the snowpack insulation. I am not considering mean/meanmax/meanmin yet for winter months because subtle shifts in average temperature do not affect the larvae probability of survival.

Spring (Mar-May): active feeding & growth

- aprsnow
- aprsnowfall

Snow in the early spring (late snow) can compress feeding time and/or refreeze things possibly leading to the death of the soon-to-be butterfly.

- maymean
- mayextmin (extreme may minimum)

An overall warmer may creates earlier/longer growing conditions resulting in better growth of larvae. Additionally a late frost when larvae are out feeding can be fatal.

Early summer (Jun-Jul): pupation & heat stress

- junmean
- junextmax (extreme maximum in June)
- julmeanmax (average of hottest daily temps in July)
- julextmax (absolute extreme in july)

This variables were chosen to capture chronic heat and acute heat waves that could kill larvae/pupae or reduce host plant quality. Additionaly, the general thermal environment as larvae finish developing (junmean)

- julsnow

higher moisture leads to better quality plants. Very low moisture might signal hot, dry summers stressing the larvea

Late summer & early fall (Aug-Sep): adults, egg laying, early larvae

- augmean
- augextmax

This time period affects adult mating, egg laying, and egg survival during potential heat waves.

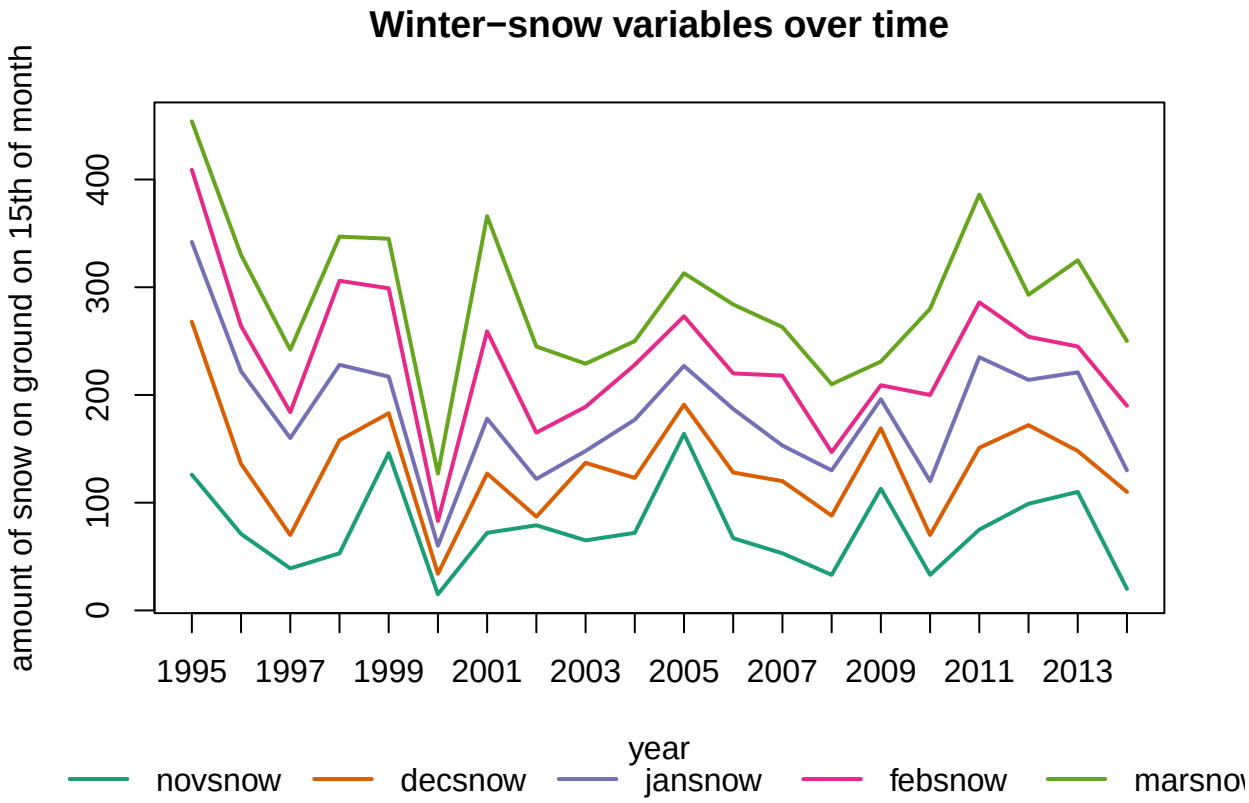
- sepmean
- sepextmin

warmer septembers extend feeding, but early frosts can kill exposed larvae before they go under snow.

Checking Winter predictors for possible transformations and collinearity

Winter-snow predictors:

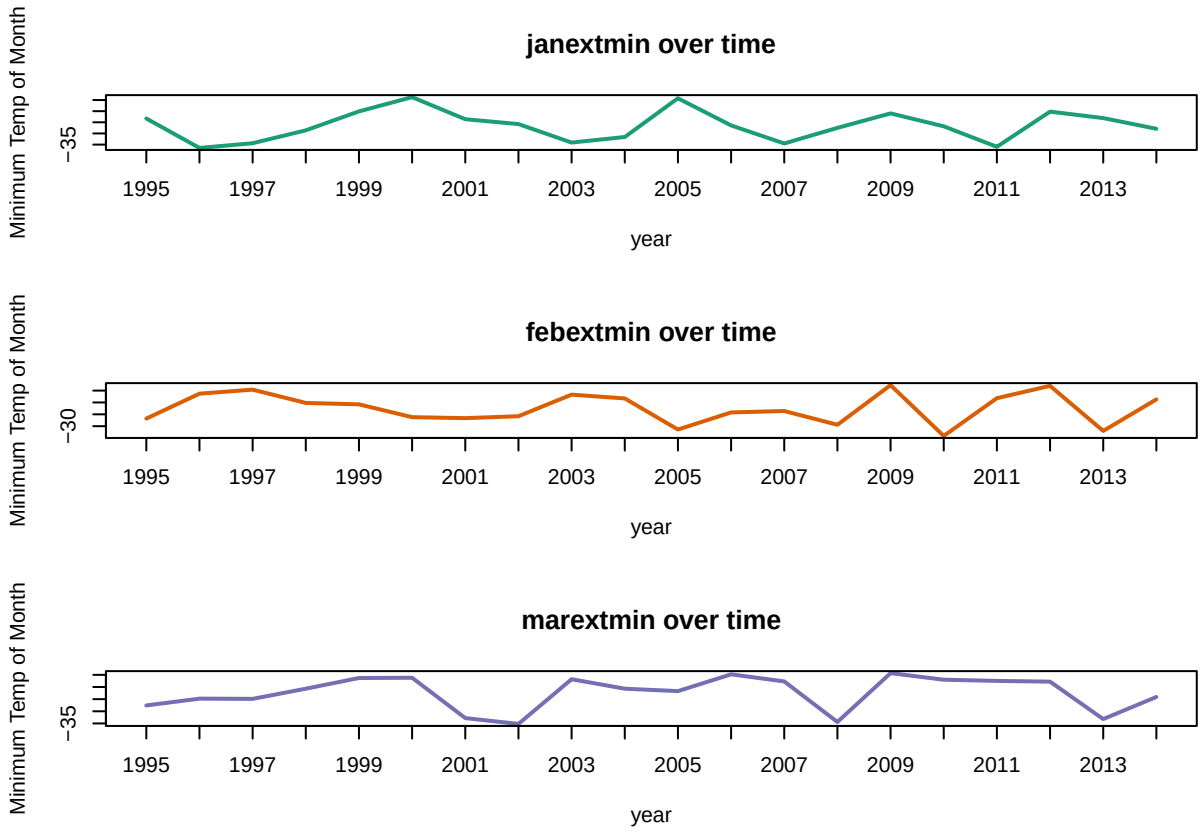
```
#winter-snow variables
winter_vars <- c("novsnow", "decsnow", "jansnow", "febsnow", "marsnow")

#distinguishable colors
cols <- brewer.pal(5, "Dark2")

# y-axis bounds
ymin <- min(d[winter_vars], na.rm = TRUE)
ymax <- max(d[winter_vars], na.rm = TRUE)

plot(d$year, d$novsnow,
     type = "n",
     xlab = "year",
     ylab = "amount of snow on ground on 15th of month",
     xaxt = "n",
     main = 'Winter-snow variables over time',
      ylim = c(ymin, ymax))

# Custom x-axis
axis(1,
     at = sort(unique(d$year)),
     labels = sort(unique(d$year)))

for(i in seq_along(winter_vars)){
  lines(d$year, d[[winter_vars[i]]], col = cols[i], lwd = 2)
}
legend("bottom",
       legend = winter_vars,
       col = cols,
       lwd = 2,
       horiz = TRUE,
       bty = "n",
       xpd = TRUE,
       inset = c(0, -0.4))
```



Winter-temp predictors:

```
winter_vars <- c("janextmin", "febextmin", "marextmin")

# colors for each line
cols <- brewer.pal(5, "Dark2")[1:3]

# set up 3 rows, 1 column of plots
par(mfrow = c(3, 1))

for (i in seq_along(winter_vars)) {

  v <- winter_vars[i]

  # compute axis limits for THIS variable only
  ymin <- min(d[[v]], na.rm = TRUE)
  ymax <- max(d[[v]], na.rm = TRUE)

  # individual plot
  plot(d$year, d[[v]],
       type = "l",
       col = cols[i],
       lwd = 2,
       xlab = "year",
       ylab = "Minimum Temp of Month",
       main = paste(v, "over time"),
       ylim = c(ymin, ymax),
```

```
       xaxt = "n")

  axis(1,
       at = sort(unique(d$year)),
       labels = sort(unique(d$year)))
}
```



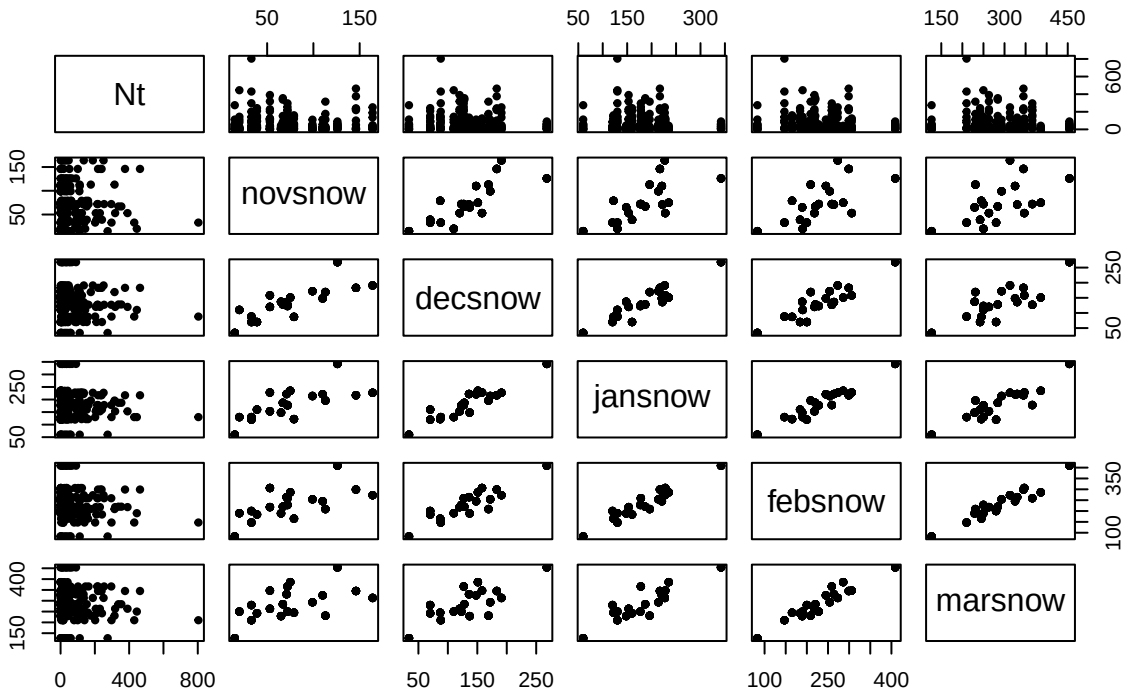
I examined the winter extreme minimum temperature variables (janextmin, febextmin, marextmin) individually over time to check for patterns that might suggest a need for transformation. janextmin showed fairly consistent variation across years, indicating no clear issue. febextmin and especially marextmin showed mild non-constant variance, with some periods having larger fluctuations than others. However, since these are predictors—not the response—and the variance differences were not severe, I decided to keep them on their original scale. I will revisit whether transformation is needed when evaluating their relationship with population size in the modeling stage/pairsplot.

The snow predictors did not exhibit obvious non-constant variance or skew, so I will keep them on their natural scale.

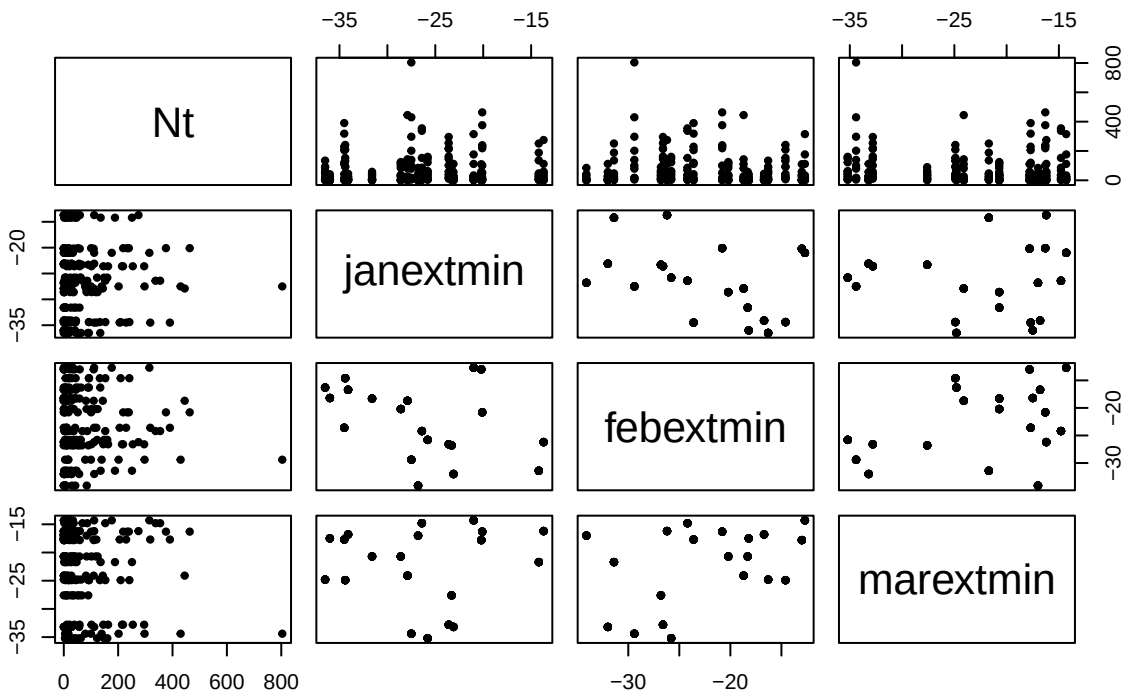
pairsplot of winter variables:

```
winter_snow  <- c("novsnow", "decsnow", "jansnow", "febsnow", "marsnow")
winter_temp  <- c("janextmin", "febextmin", "marextmin")


pair_vars <- c("Nt", winter_snow, winter_temp)
```

```r
winter_dat <- d[, pair_vars]
winter_dat <- winter_dat[complete.cases(winter_dat), ]


snow_dat <- winter_dat[, c("Nt", winter_snow)]
temp_dat <- winter_dat[, c("Nt", winter_temp)]

# plot
par(mfrow = c(1, 2))    # 1 row, 2 columns

pairs(snow_dat,
      main = "Pair plot: Nt and Winter Snow Predictors",
      pch = 20)
```




```r
pairs(temp_dat,
      main = "Pair plot: Nt and Winter Temperature Predictors",
      pch = 20)
```



The pair plot of Nt and the winter predictors revealed several useful patterns. First, the five winter snow variables were extremely highly correlated with each other, indicating strong collinearity. This suggests that only a subset of these variables can be included together in the same model. In contrast, the winter temperature variables were less correlated with the snow variables and with each other, indicating that they provide different information.

When comparing Nt with the predictors, the relationships with the snow variables showed curvature and non-constant variance, which supports considering a transformation of Nt (e.g., Sqrt(Nt)) in later modeling. The winter temperature predictors did not show strong linear relationships with Nt, suggesting they may not be strong predictors by themselves. Overall, the pair plot helps narrow the universe of variables and highlights issues of collinearity and potential response transformations.

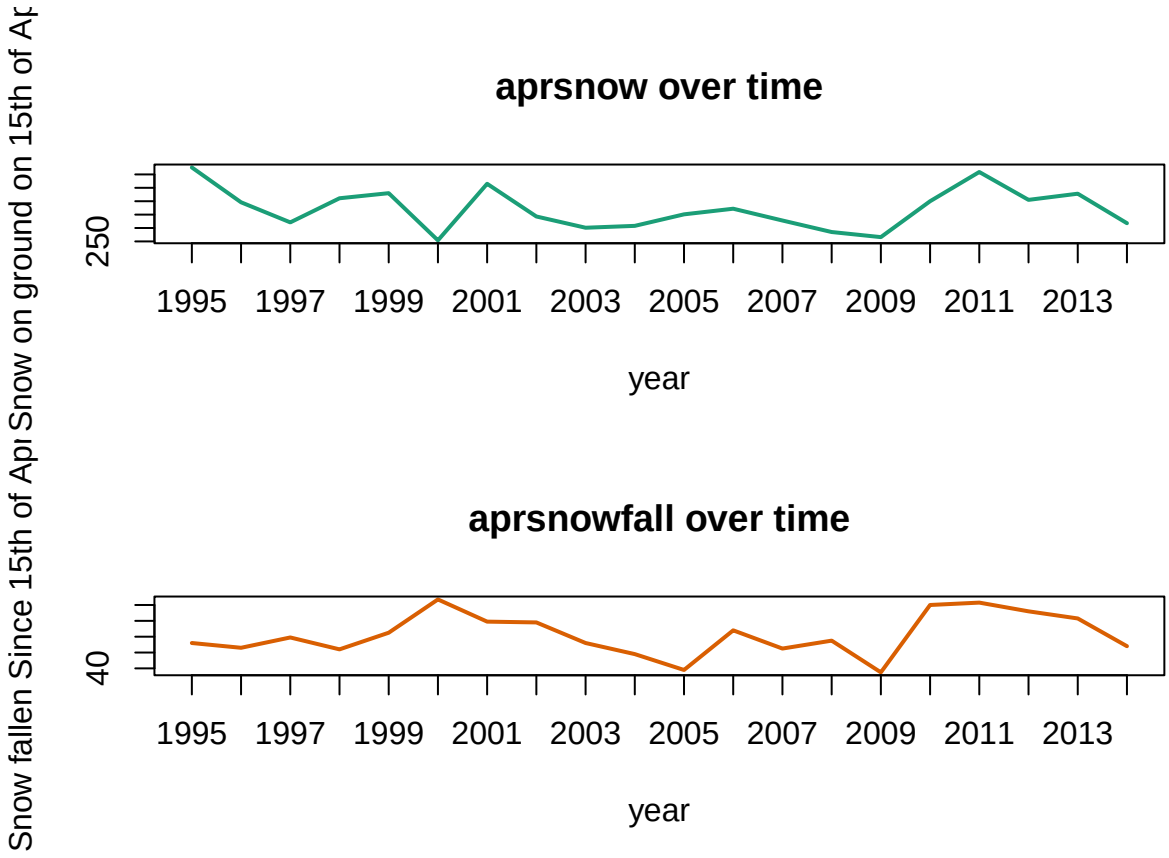
Note: If in later modeling I find that the R-squared value is low for my competing models than I will consider adding other predictors not in the initial Universe of variables. Perhaps I am missing new information and/or an interaction that explains Nt.

Checking Spring-predictors for possible transformations and collinearity.

Spring snow predictors:

```
par(mfrow = c(2,1))
spring_snow_var <- c("aprsnow", "aprsnowfall")
y_axis <- c("Snow on ground on 15th of April", "Snow fallen Since 15th of April")

for (i in seq_along(spring_snow_var)) {

  v <- spring_snow_var[i]
  label <- y_axis[i]
```

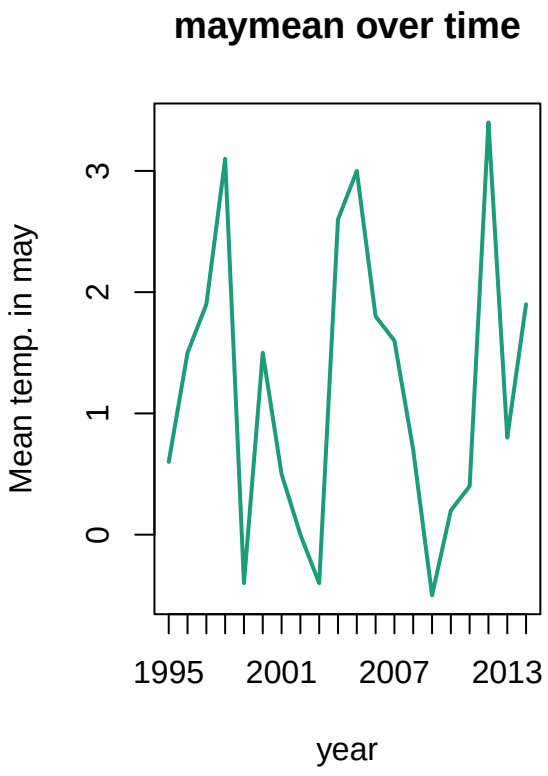
```
  # compute axis limits for THIS variable only
  ymin <- min(d[[v]], na.rm = TRUE)
  ymax <- max(d[[v]], na.rm = TRUE)

  # individual plot
  plot(d$year, d[[v]],
       type = "l",
       col = cols[i],
       lwd = 2,
       xlab = "year",
       ylab = label,
       main = paste(v, "over time"),
       ylim = c(ymin, ymax),
       xaxt = "n")

  axis(1,
       at = sort(unique(d$year)),
       labels = sort(unique(d$year)))
}
```

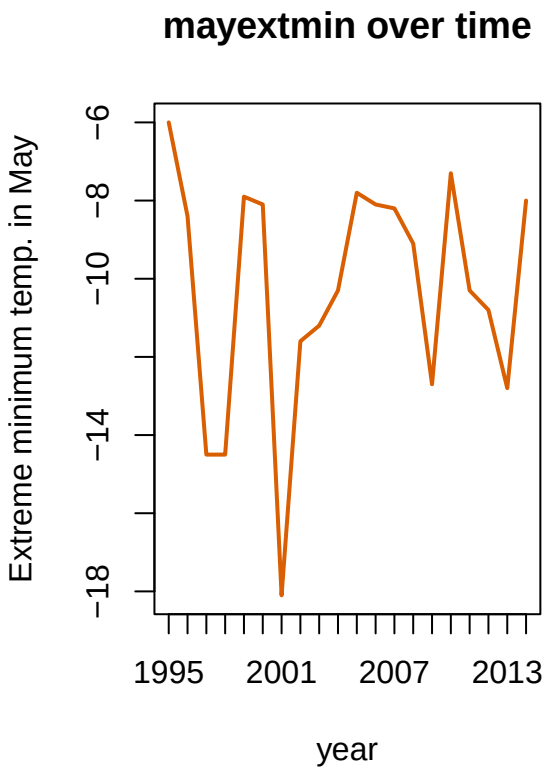


Spring temperature predictors:

```
par(mfrow = c(1,2))
spring_temp_var <- c("maymean", "mayextmin")
y_axis <- c("Mean temp. in may", "Extreme minimum temp. in May")
```

```r
for (i in seq_along(spring_temp_var)) {

  v <- spring_temp_var[i]
  label <- y_axis[i]

  # compute axis limits for THIS variable only
  ymin <- min(d[[v]], na.rm = TRUE)
  ymax <- max(d[[v]], na.rm = TRUE)

  # individual plot
  plot(d$year, d[[v]],
       type = "l",
       col = cols[i],
       lwd = 2,
       xlab = "year",
       ylab = label,
       main = paste(v, "over time"),
       ylim = c(ymin, ymax),
       xaxt = "n")

  axis(1,
       at = sort(unique(d$year)),
       labels = sort(unique(d$year)))
}
```



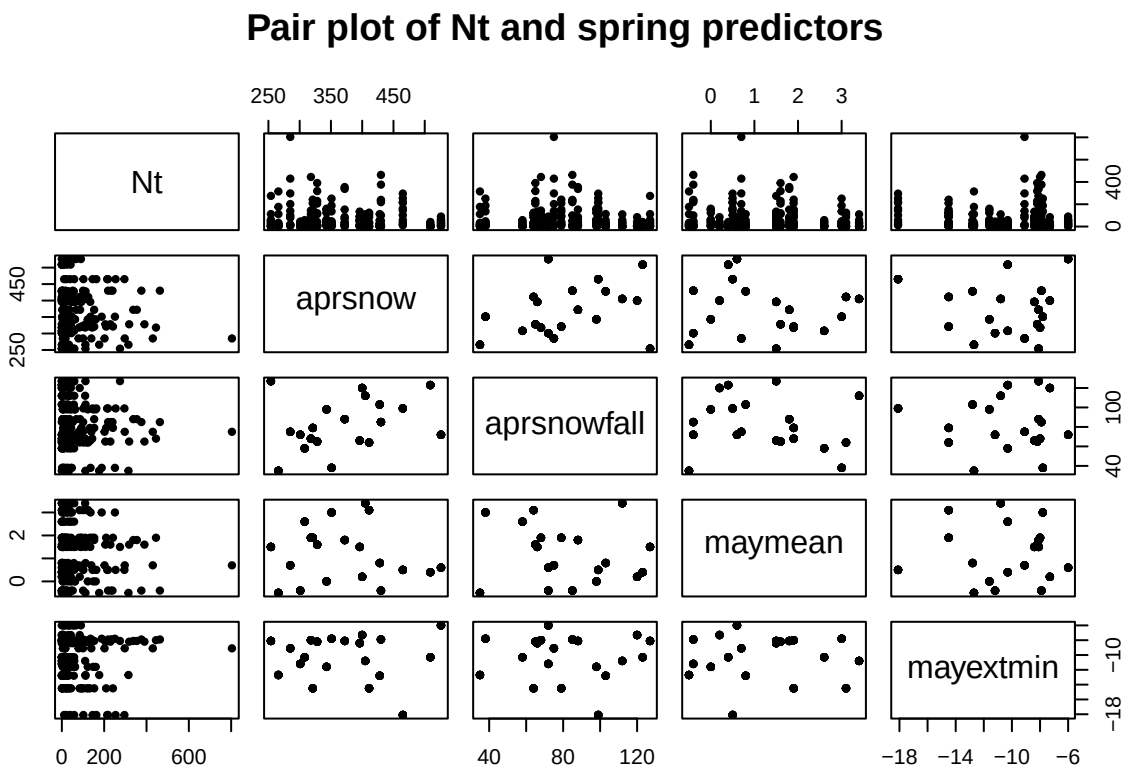


For the spring predictors, the two snow variables (aprsnow and aprsnowfall) showed mild non-constant variance over time, with some years having much larger fluctuations than others. This is similar to the pattern seen in the winter snow variables. However, because these are predictors rather than the response, I kept them on their original scale for now and will reassess whether a transformation is necessary after fitting models by examining residual plots. If the residuals display non-constant variance or strong nonlinearity with respect to these predictors, then a transformation may be appropriate.

In contrast, the spring temperature variables (maymean and mayextmin) displayed very consistent variability across years and did not suggest any need for transformation. I do notice an outlier year 2001 in the mayextmin over time plot, however, the majoirty of the values are between -15 and -8 degrees celcius.

Pairs plot of Spring variables:

```
spring_snow  <- c("aprsnow", "aprsnowfall")
spring_temp  <- c("maymean", "mayextmin")


spring_vars <- c("Nt", spring_snow, spring_temp)

# subset data and drop any incomplete rows
spring_dat <- d[, spring_vars]
spring_dat <- spring_dat[complete.cases(spring_dat), ]

# pair plot
pairs(spring_dat,
      main = "Pair plot of Nt and spring predictors",
      pch = 20)
```



The spring pair plot revealed strong collinearity between the two spring snow variables (aprsnow and aprsnowfall), indicating that they represent essentially the same information about spring snowpack. In contrast, the spring temperature predictors (maymean and mayextmin) were not strongly correlated with each other or with the snow variables, suggesting they capture a distinct seasonal signal.

The relationship between Nt and the spring predictors showed considerable scatter and some curvature, with Nt displaying non-constant variance across predictor values. This provides additional support for considering a transformation of the response (e.g., sqrt(Nt)) during model building.

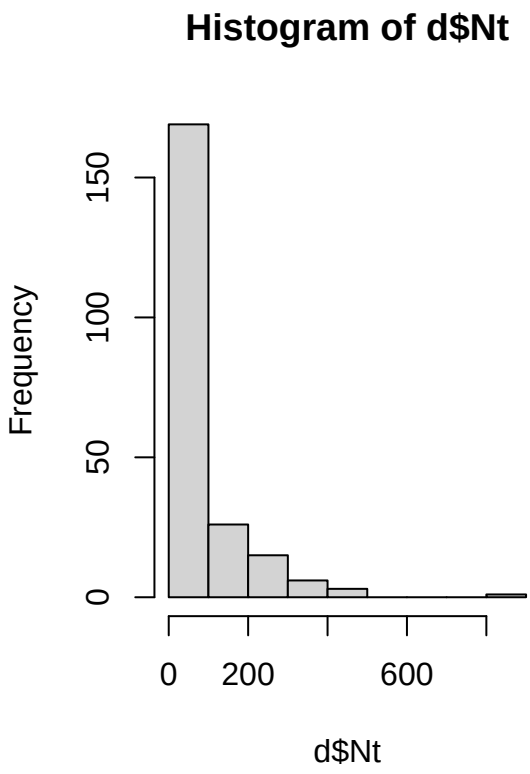
The exploratory checks for the early-summer (Jun–Jul), late-summer/early-fall (Aug–Sep), and big-picture predictors (maxsnow and datesnowmelt) showed very similar behavior to the winter and spring variables. Predictors within each season were often strongly correlated with each other (e.g., mean vs. extreme temperatures), while their relationships with Nt showed the same curvature and non-constant variance seen earlier. These patterns reinforced two conclusions already apparent from the winter and spring EDA:

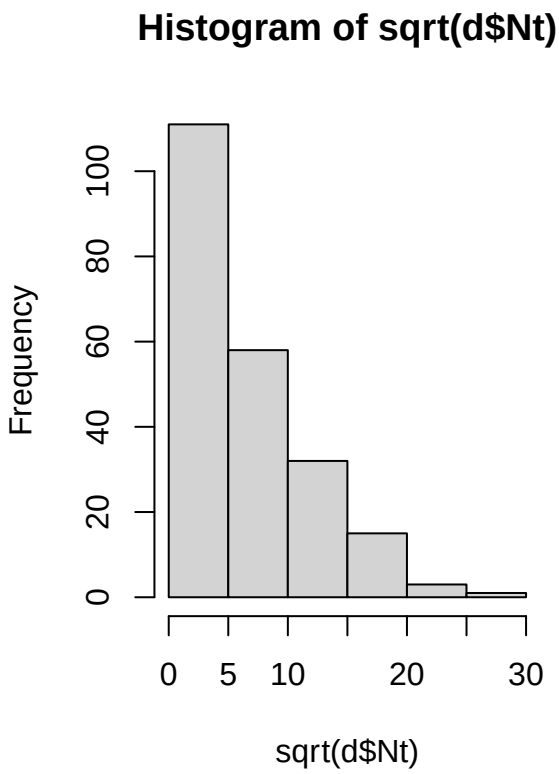
- Nt should be modeled using a variance-stabilizing transformation such as sqrt(Nt), and
- highly correlated predictors should be treated as interchangeable candidates rather than used simultaneously in a single model. Because these results were qualitatively the same across all remaining seasons, I omit the redundant plots here and summarize only the key takeaways that inform the modeling stage.

Now I will apply the sqrt() transformation to the response variable Nt, and re access the predictors that showed non-constant variance or a nonlinear relationship with Nt in the pair plots. I will use mini pairs plots to look at the relationship between predictors and Nt.

Before doing the sqrt(Nt) transform we can look at the distribution of Nt before and after the transform:

```
par(mfrow = c(1,2))
hist(d$Nt)
hist(sqrt(d$Nt))
```





This shows the heavy skew in Nt being levitated from the transform, supporting the need for a transformation. Additionally, we cannot do a log transformation because the data has zeros.

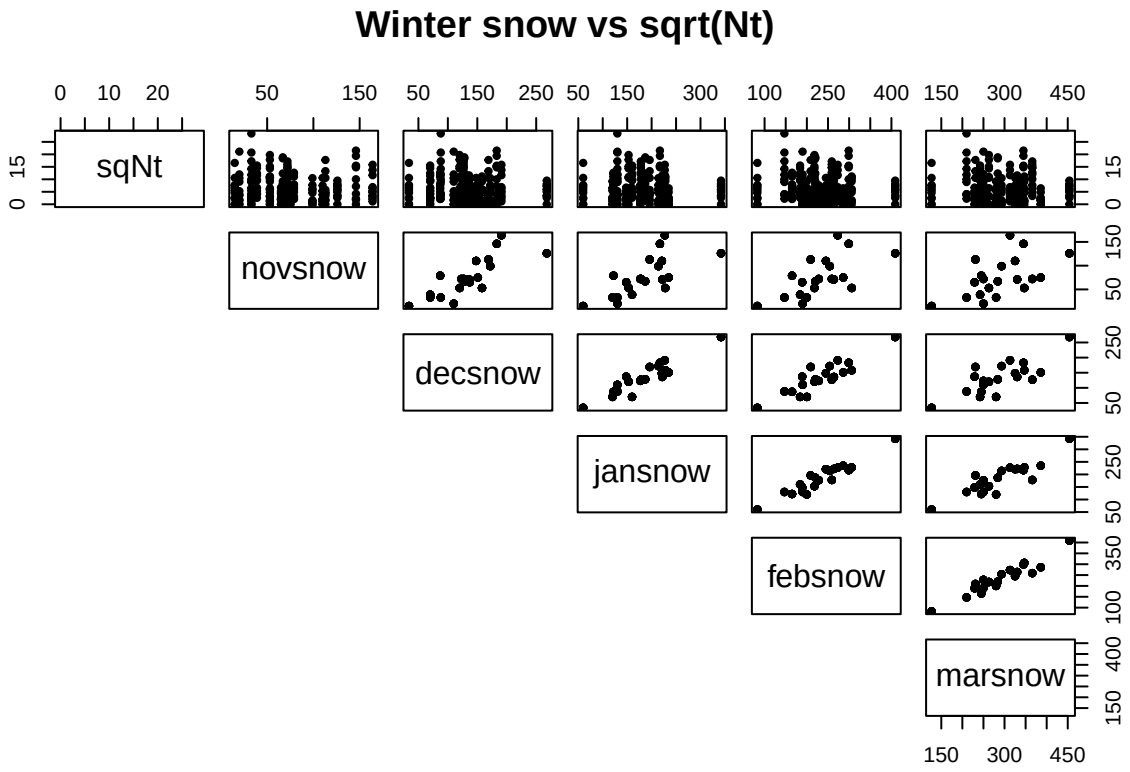
adding transform to data set:

```
d$sqNt <- sqrt(d$Nt)
```

Predcitors re-accessment after transform

mini pairs plot:

```
pairs(~ sqNt + novsnow + decsnow + jansnow + febsnow + marsnow,
      data = d,
      main = "Winter snow vs sqrt(Nt)",
      pch = 20,
      lower.panel = NULL)
```



After applying the sqrt(Nt) transformation, the relationships between Nt and the winter snow predictors improved substantially. The strong curvature and heteroscedasticity evident in the raw Nt plots were largely removed, and the transformed response displayed much more linear and homoscedastic patterns. Although some variability remains, sqrt(Nt) shows a far more stable spread across the full range of snow values, suggesting that this transformation is appropriate for modeling.

The winter snow variables remained highly collinear with each other after transformation, confirming that only one or two should be included in a given model. Overall, the sqrt(Nt) transformation effectively linearized the predictor–response relationships for winter snow variables, supporting its use in subsequent modeling.

Although several predictors exhibited strong collinearity (particularly the winter and spring snow variables), I did not remove them at the EDA stage. Instead, I will address collinearity, variable selection, and response transformation formally during the modeling phase.

The remaining seasonal predictors (spring, early summer, late summer, early fall, and "big picture" vars) showed the same qualitative improvements after applying sqrt(Nt): variance became more stable, curvature was reduced, and the predictor–response relationships were visibly more linear. A small amount of non-linearity remained in some cases, but not to a degree that would motivate polynomial terms at this stage.

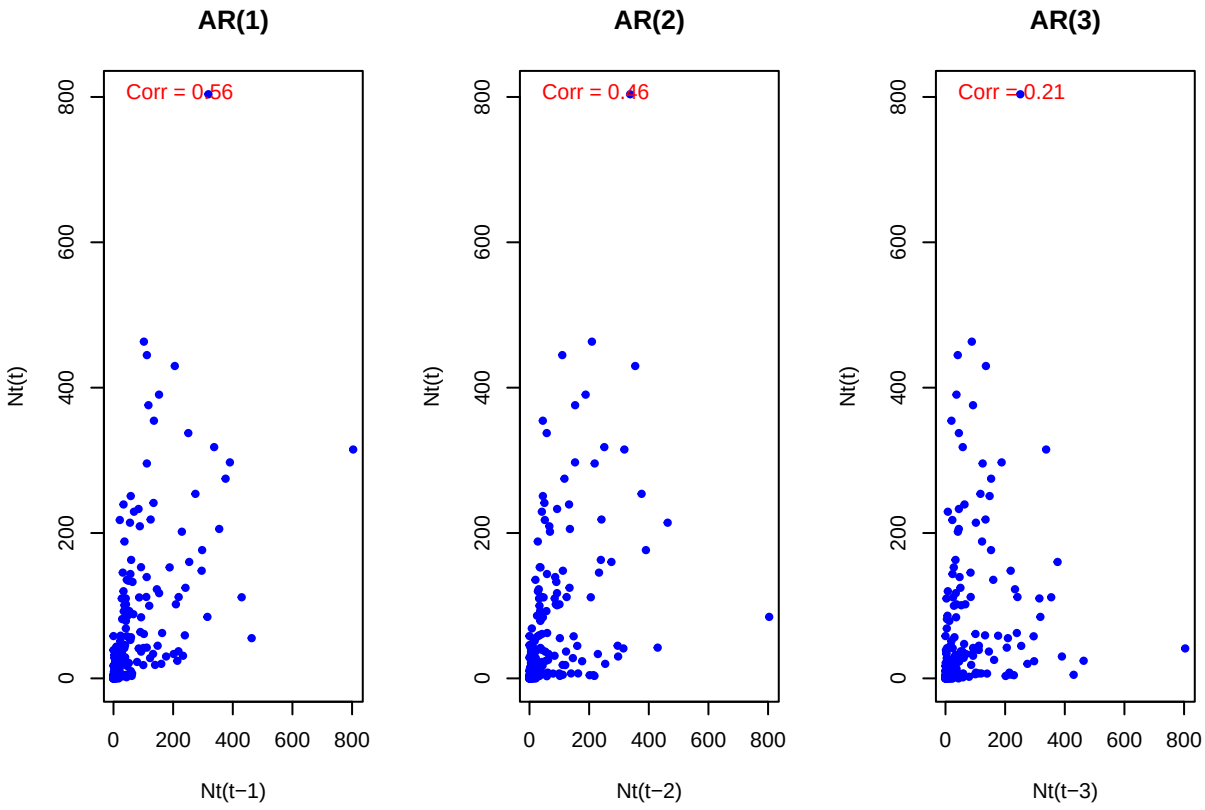
Notably, the summer and early-fall predictors continued to show weak associations with sqrt(Nt), suggesting that these variables may play a smaller direct role in determining population size compared to winter and spring conditions. Overall, the consistency of these improvements across all predictor groups further supports the decision to model sqrt(Nt) rather than raw Nt, and highlights that the main concerns moving forward are collinearity and variable selection rather than additional transformations.

Adding Nt-1 (AR(1)) variable to Data

```
d <- d %>%
  arrange(meada, year) %>%
  group_by(meada) %>%
  mutate(
    Nt_lag1 = lag(Nt, 1),
    Nt_lag2 = lag(Nt, 2),
    Nt_lag3 = lag(Nt, 3)
  ) %>%
  ungroup()
```

Now I will make sure that lag1 is effecient and we do not need a lag2, lag3, etc.

```
par(mfrow = c(1, 3))

r1 <- cor(d$Nt, d$Nt_lag1, use = "complete.obs")
r2 <- cor(d$Nt, d$Nt_lag2, use = "complete.obs")
r3 <- cor(d$Nt, d$Nt_lag3, use = "complete.obs")

# AR(1)
plot(d$Nt_lag1, d$Nt, pch = 20, col = "blue",
     xlab = "Nt(t-1)", ylab = "Nt(t)",
     main = "AR(1)")

text(min(d$Nt_lag1, na.rm=TRUE),
     max(d$Nt, na.rm=TRUE),
     labels = paste("Corr =", round(r1, 2)), pos = 4, col="red")

# AR(2)
plot(d$Nt_lag2, d$Nt, pch = 20, col = "blue",
     xlab = "Nt(t-2)", ylab = "Nt(t)",
     main = "AR(2)")
text(min(d$Nt_lag2, na.rm=TRUE),
     max(d$Nt, na.rm=TRUE),
     labels = paste("Corr =", round(r2, 2)), pos = 4, col="red")

# AR(3)
plot(d$Nt_lag3, d$Nt, pch = 20, col = "blue",
     xlab = "Nt(t-3)", ylab = "Nt(t)",
     main = "AR(3)")
text(min(d$Nt_lag3, na.rm=TRUE),
     max(d$Nt, na.rm=TRUE),
     labels = paste("Corr =", round(r3, 2)), pos = 4, col="red")
```



As I increase the lag, the correlation with the response Nt, reduces significantly. For now, I will stick with only AR(1) in my universe of variables, but I will keep in mind that AR(2) could potentially be an option if I see left over autocorrelation in the residuals.

In summary, the exploratory analysis clarified the structure and behavior of the dataset and guided the construction of my final universe of variables. Meadow identity and an AR(1) term were retained as key structural predictors based on clear differences among meadows and strong temporal autocorrelation. Most weather predictors showed stable variance across years, while Nt exhibited strong heteroscedasticity and skew, motivating the use of sqrt(Nt) as the response in all subsequent modeling. The seasonal pair plots revealed substantial collinearity within groups of similar predictors (especially among snow variables and among mean vs. extreme temperatures), so only one or two representatives from each correlated group will be considered together in any single model. After transformation, the predictor–response relationships became more linear and homoscedastic, and no additional transformations for predictors were needed. Some predictors—particularly late-summer and early-fall variables—showed weak associations with sqrt(Nt), suggesting they may play a limited role, but they will remain in the candidate set. With the data cleaned, transformed, and reduced to a workable universe of variables, I now proceed to the modeling stage where I formally evaluate model structure, assess collinearity, and compare competing models.

My full list of variables in my universe of varibles is listed at the top of the EDA section.

Data Anlysis

Firstly, I would like to use a step function to see what the highest R-squared value I can achieve is while still rewarding simplicity by using BIC. This is motivated by my suspicion of the explanatory power of predictors, and my curiosity as to what interactions the step() function will return. Potentially, I could "reverse engineer" these interactions to understand them.

Easy Access to variables set-up:

```
winter_var <- c("novsnow", "decsnow", "jansnow", "marsnow", "janextmin",
                "febextmin", "marextmin")
spring_var <- c("aprsnow", "aprsnowfall", "maymean", "mayextmin")
early_summer_var <- c("junmean", "junextmax", "julmeanmax", "julextmax",
                      "julsnow")
summer_to_fall_var <- c("augmean", "augextmax", "sepmean", "sepextmin")

forced_var <- c("maxsnow", "Datesnowmelt", "meada", "Nt_lag1")

weather_var <- c(winter_var, spring_var, early_summer_var, summer_to_fall_var)

all_pred <- c(forced_var, weather_var)

# All variables that matter for step (response + all predictors)
vars_for_step <- c("Nt", all_pred)

# Keep only rows that are complete across these
d_step <- d[complete.cases(d[, vars_for_step]), ]
```

First I will run a step() function without considering interactions:

```
form_full <- as.formula(
  paste("Nt ~", paste(all_pred, collapse = " + "))
)

# Upper model includes everything
upper_mod <- lm(form_full, data = d_step)

# Lower model = NULL model (intercept only)
lower_mod <- lm(Nt ~ 1, data = d_step)

n <- nrow(d_step)

step_main <- step(
  object = lower_mod,
  scope  = list(lower = lower_mod, upper = upper_mod),
  direction = "both",
  k = log(n),    # BIC penalty
  trace = FALSE
)
```

```r
summary(step_main)
```

```
##
## Call:
## lm(formula = Nt ~ Nt_lag1 + julextmax, data = d_step)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -252.47  -37.35  -18.29    7.04  592.56
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -185.41359   60.67016  -3.056 0.002539 **
## Nt_lag1        0.61191    0.05873  10.419  < 2e-16 ***
## julextmax      9.84624    2.74133   3.592 0.000411 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87.84 on 206 degrees of freedom
## Multiple R-squared:  0.3525, Adjusted R-squared:  0.3462
## F-statistic: 56.06 on 2 and 206 DF,  p-value: < 2.2e-16
```

After running the stepwise BIC procedure on my full set of predictors, the selected model retained only Nt_lag1 and julextmax. Interestingly, the meadow factor (meada), which was intended to account for differences between locations, was not selected. The resulting model explains a modest portion of the variation ($R^2 \approx 0.35$), and both predictors are highly statistically significant. I included this model as one of my competing models because a structure that does not rely on meadow may generalize better to new locations.

Because Nt_lag1 was strongly significant and the correlation between Nt and Nt(t−2) was moderate (0.46), I ran the stepwise procedure again while allowing an AR(2) term. This second model kept the same predictors as before but additionally selected Nt_lag2, which was moderately statistically significant (p 0.001). However, the increase in R^2 was very small (from 0.35 to 0.38), indicating that AR(2) adds little explanatory power. Given the minimal improvement in fit and the added complexity of an extra lag, I decided not to move forward with the AR(2) model.

adding model to competing models:

```r
model1 <- lm(Nt ~ Nt_lag1 + julextmax, data = d)

competeting_models <- c(model1)
```

Now I will run a step() function with considering interactions:

```r
form_int_all <- as.formula(
  paste(
    "Nt ~ (",
    paste(all_pred, collapse = " + "),
    ")^2"
  )
)

# upper and lower models
upper_int <- lm(form_int_all, data = d_step)
lower_null <- lm(Nt ~ 1, data = d_step)
```

```r
n <- nrow(d_step)

step_int <- step(
  object    = lower_null,
  scope     = list(lower = lower_null, upper = upper_int),
  direction = "both",
  k         = log(n),   # BIC
  trace     = FALSE
)

summary(step_int)
```

```
##
## Call:
## lm(formula = Nt ~ Nt_lag1 + julextmax + aprsnowfall + marsnow +
##     Nt_lag1:julextmax, data = d_step)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -232.50  -34.39  -16.51   15.12  596.53
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -92.19003   73.05973  -1.262  0.20845
## Nt_lag1            -1.29969    0.62545  -2.078  0.03897 *
## julextmax           4.30501    3.28687   1.310  0.19176
## aprsnowfall        -0.57107    0.23156  -2.466  0.01448 *
## marsnow             0.24547    0.10238   2.398  0.01741 *
## Nt_lag1:julextmax   0.09443    0.03031   3.115  0.00211 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 84.91 on 203 degrees of freedom
## Multiple R-squared:  0.4038, Adjusted R-squared:  0.3891
## F-statistic:  27.5 on 5 and 203 DF,  p-value: < 2.2e-16
```

Given that the R-squared did not increase much with interactions I feel that I am missing a key structure to this data. Currently weather is year-specific instead of month-specific meaning weather variables are as if they happen at the "time" when in reality they do not. Because of this issue I believe adding a time component such as year as a continuous variable will help increase R squared in the models. Adding a time component for month would be optimal but that is not possible with the structure of the data.

After allowing a linear time trend (year as a continuous predictor) in the stepwise search, both with and without interactions, the stepwise procedure did not retain the time term in either search. This indicates that there is no strong remaining linear trend in population size once autocorrelation and weather covariates are accounted for; EDA also supported no time trend by year.

I will add the model outputted from the stepwise search with interactions as a starting point:

```r
model2 <- lm(Nt ~ Nt_lag1 + julextmax + aprsnowfall + marsnow + Nt_lag1:julextmax, data = d)
competeting_models <- c(model1, model2)
```

Butterfly population size should be strongly influenced by the previous year's population, overwinter conditions affecting larval survival, and extreme summer temperatures impacting adult activity and reproduction. I therefore propose a model combining AR(1), a winter severity measure, a snow melt timing variable, and a summertime temperature variable:

- note: I am using collinearity finding from my EDA to avoid picking variables that explain the same thing

```
model3 <- lm(Nt ~ Nt_lag1 + marsnow + Datesnowmelt + julextmax , data=d)
model4 <- lm(Nt ~ Nt_lag1 + marsnow + Datesnowmelt + julextmax + Nt_lag1:julextmax, data=d)

competeting_models <- c(model1, model2, model3, model4)
```

Models 1 and 2 are nested and models 3 and 4 are nested. Given this, I will use Partial-F tests to compare model1 vs. model2 and model3 vs. model4. Then I will use BIC to determine the top model of these competing models:

```
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: Nt ~ Nt_lag1 + julextmax
## Model 2: Nt ~ Nt_lag1 + julextmax + aprsnowfall + marsnow + Nt_lag1:julextmax
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    206 1589389
## 2    203 1463401  3    125988 5.8256 0.0007747 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model3, model4)
```

```
## Analysis of Variance Table
##
## Model 1: Nt ~ Nt_lag1 + marsnow + Datesnowmelt + julextmax
## Model 2: Nt ~ Nt_lag1 + marsnow + Datesnowmelt + julextmax + Nt_lag1:julextmax
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1    204 1560845
## 2    203 1507092  1     53753 7.2404 0.007722 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I compared the nested pairs of models using partial F-tests. For the two models generated by stepwise selection, Model 2 (which included additional weather terms and the interaction Nt_lag1 : julextmax) fit significantly better than the simpler Model 1 ($F = 5.83$, $p < 0.001$). This indicates that the extra predictors in Model 2 explain a meaningful amount of variation in Nt beyond the AR(1) term and julextmax alone.

For my two hypothesis-driven models, I similarly compared the main-effects model (Model 3) with the version including the biologically motivated interaction Nt_lag1 : julextmax (Model 4). The interaction model provided a significantly improved fit ($F = 7.24$, $p = 0.008$), supporting the idea that the effect of summer heat on population size depends on previous population size.

Thus, in both case (the stepwise models and the hypothesis models), the full model (the model including the additional terms or interaction) was preferred over the reduced model.

BIC comparison of Model 2 and Model 4:

```
BIC_model2 <- BIC(model2)
BIC_model4 <- BIC(model4)
BIC <- c(BIC_model2, BIC_model4)
BIC
```

```
## [1] 2480.986 2487.135
```

The BIC comparison shows that Model 2 is preferred over Model 4, as it has the lower BIC value (2481 vs. 2487). Since both models contain the same number of predictors, this difference reflects a better overall

model fit for Model 2 rather than a penalty for complexity. Based on this, I selected Model 2 as my top linear model. I then fit a Negative Binomial GLM using the same predictors to compare its performance against the linear model version.

```
d$Nt_int <- as.integer(round(d$Nt)) #in order to avoid long warning message
# Fit Negative Binomial GLM
glm_nb2 <- glm.nb(Nt_int ~ Nt_lag1 + julextmax + aprsnowfall + marsnow + Nt_lag1:julextmax,
                  data = d)
```

BIC of Neg Binomial GLM:

```
n <- nrow(d)
BIC_nb2 <- extractAIC(glm_nb2, k = log(n))[2]
BIC_nb2
```

```
## [1] 2000.549
```

The Negative Binomial GLM had a BIC value of approximately 2000.5, which is substantially higher than the BIC values for both linear models (Model 2 and Model 4). Since lower BIC values indicate better model performance after accounting for model complexity, this result suggests that the Negative Binomial GLM does not provide an improvement over the linear version of Model 2. Thus, Model 2 remains the preferred model among all candidates.

As a supplementary check, I evaluated whether Model 2 improved when using a square-root transformation of the response. Although this transformation is often considered earlier in the analysis, performing it here allows a direct comparison with the best-performing linear model:

```
model2_sqrt <- lm(sqNt ~ Nt_lag1 + julextmax + aprsnowfall + marsnow +
                    Nt_lag1:julextmax, data = d)
BIC_model2_sqrt <- BIC(model2_sqrt)
BIC <- c(BIC_model2, BIC_model2_sqrt)
BIC
```

```
## [1] 2480.986 1198.400
```

Because a square-root transformation had been considered during the exploratory analysis, I fit a sqrt(Nt) version of Model 2 to determine whether stabilizing the variance improved model performance. The BIC for the transformed model (1198.4) was dramatically lower than the BIC of the untransformed model (2481.0), a difference exceeding 1,200 points. This represents overwhelming evidence in favor of the transformed response. I therefore will consider the response transformation, sqrt(), for all competing models when doing out-of-sample prediction model comparison.

Out-of sample prediction model comparison:

```
formula1_sqrt <- sqNt ~ Nt_lag1 + julextmax
formula2_sqrt <- sqNt ~ Nt_lag1 + julextmax + aprsnowfall + marsnow + Nt_lag1:julextmax
formula3_sqrt <- sqNt ~ Nt_lag1 + marsnow + Datesnowmelt + julextmax
formula4_sqrt <- sqNt ~ Nt_lag1 + marsnow + Datesnowmelt + julextmax + Nt_lag1:julextmax

meadow_out <- "G"

train_lomo <- subset(d, meada != meadow_out)
test_lomo  <- subset(d, meada == meadow_out)

# Fit sqrt models
model1_lomo <- lm(formula1_sqrt, data = train_lomo)
model2_lomo <- lm(formula2_sqrt, data = train_lomo)
model3_lomo <- lm(formula3_sqrt, data = train_lomo)
model4_lomo <- lm(formula4_sqrt, data = train_lomo)
```

```r
# Predict sqrt(Nt)
pred1_sqrt <- predict(model1_lomo, newdata = test_lomo)
pred2_sqrt <- predict(model2_lomo, newdata = test_lomo)
pred3_sqrt <- predict(model3_lomo, newdata = test_lomo)
pred4_sqrt <- predict(model4_lomo, newdata = test_lomo)

# back-transform
pred1 <- pred1_sqrt^2
pred2 <- pred2_sqrt^2
pred3 <- pred3_sqrt^2
pred4 <- pred4_sqrt^2

# RMSE function (drop unavoidable NA from first lag)
rmse_fun <- function(obs, pred){
  ok <- !is.na(pred)
  sqrt(mean((obs[ok] - pred[ok])^2))
}

rmse1 <- rmse_fun(test_lomo$Nt, pred1)
rmse2 <- rmse_fun(test_lomo$Nt, pred2)
rmse3 <- rmse_fun(test_lomo$Nt, pred3)
rmse4 <- rmse_fun(test_lomo$Nt, pred4)

c(rmse1, rmse2, rmse3, rmse4)
```

```
## [1] 146.0052 140.2399 142.8916 149.9105
```

To compare the predictive performance of my four competing models, I conducted a leave-one-meadow-out (LOMO) validation using meadow G as the held-out series. Each model was refit using all meadows except G, and predictions were generated for every available year of meadow G (excluding the first year, for which lagged predictors are unavailable). The resulting RMSE values were:

Model 1: 146.0
Model 2: 140.2
Model 3: 142.9
Model 4: 149.9

Model 2 achieved the lowest RMSE, indicating that it provides the best out-of-sample predictive accuracy among the four candidate models when forecasting an entirely unseen meadow. Although the RMSE differences are moderate, Model 2 consistently performed slightly better, justifying its selection as the top model for subsequent interpretation.

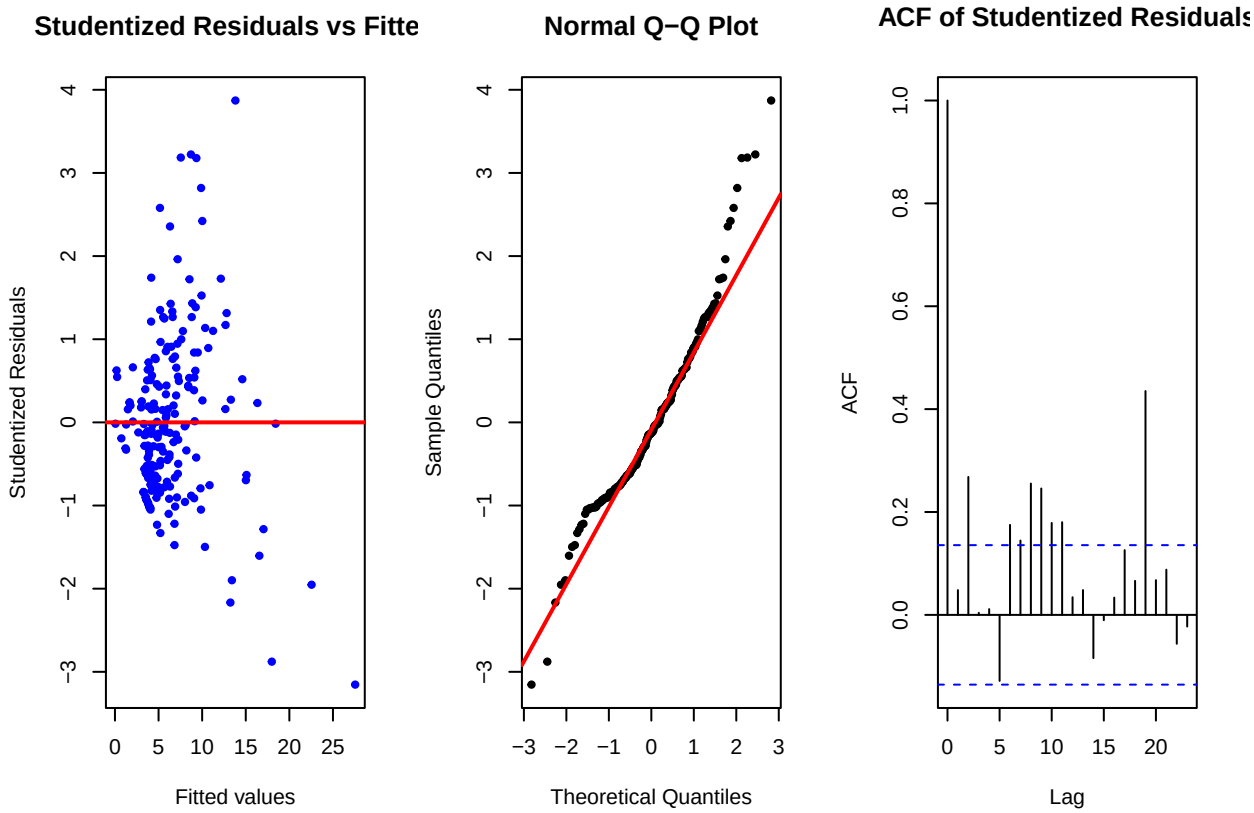
Based on these results, Model 2 (displayed below), was selected as the final model for interpretation and 2015 forecast.

$$\sqrt{N_t} = \beta_0 + \beta_1\,\text{Nt\_lag1} + \beta_2\,\text{julextmax} + \beta_3\,\text{aprsnowfall} + \beta_4\,\text{marsnow} + \beta_5\,(\text{Nt\_lag1} \cdot \text{julextmax})$$

Now I perform residual diagnostics to access Model 2:

```r
# Diagnostic plots for final model (sqrt-transformed response)
par(mfrow = c(1, 3))
```

```r
# Compute studentized residuals
stud_res <- rstudent(model2_sqrt)

# Fitted values
fit <- fitted(model2_sqrt)

# Studentized Residuals vs Fitted
plot(fit, stud_res,
     pch = 20, col = "blue",
     xlab = "Fitted values",
     ylab = "Studentized Residuals",
     main = "Studentized Residuals vs Fitted")
abline(h = 0, col = "red", lwd = 2)

#  Normal Q-Q plot (using studentized residuals)
qqnorm(stud_res,
       pch = 20,
       main = "Normal Q-Q Plot")
qqline(stud_res, col = "red", lwd = 2)

# ACF of studentized residuals
acf(stud_res,
    main = "ACF of Studentized Residuals")
```



Diagnostic plots for the final sqrt(Nt) model suggest that the model is reasonably adequate, although some assumptions are only approximately met. The studentized residuals versus fitted values show no strong nonlinear pattern, but there is still some variability in spread across fitted values, indicating mild

heteroscedasticity. The normal Q–Q plot shows moderate deviations from normality in the tails, which is common in ecological count data even after transformation. The ACF of studentized residuals shows several small but noticeable spikes at lags beyond 1, suggesting that some residual temporal autocorrelation remains, though the dominant AR(1) structure has been largely captured by the Nt_lag1 predictor. Overall, these diagnostics indicate that the model is not perfect but is sufficiently well-behaved for inference and prediction within the scope of this project.

Now I evaluate the predictive performance of the final model using an out-of-sample prediction for Nt in 2015:

```
# Add required predictors for 2015 predictions
testing$Nt_lag1 <- NA
testing$julextmax <- NA
testing$aprsnowfall <- NA
testing$marsnow <- NA

for (m in testing$meada) {
  row2014 <- d[d$meada == m & d$year == 2014, ]

  testing$Nt_lag1[testing$meada == m]      <- row2014$Nt
  testing$julextmax[testing$meada == m]    <- row2014$julextmax
  testing$aprsnowfall[testing$meada == m]  <- row2014$aprsnowfall
  testing$marsnow[testing$meada == m]      <- row2014$marsnow
}
pred_sqrt <- predict(model2_sqrt, newdata = testing)
pred_Nt <- pred_sqrt^2

rmse_2015 <- sqrt(mean((testing$Nt - pred_Nt)^2))
rmse_2015
```

```
## [1] 111.4248
```

Using the final transformed Model 2, the out-of-sample forecast for all meadows in 2015 produced an RMSE of approximately 111.4 butterflies. Given that meadow populations range from about 0 to 803 individuals, this RMSE represents moderate predictive accuracy. The model is able to capture broad population trends across meadows, but substantial year-to-year variability and extreme population sizes (such as in meadows M or G) contribute to forecast error.

Overall, the 2015 prediction performance is consistent with the accuracy observed in the leave-one-meadow-out analysis, supporting Model 2 as a reasonable, though not perfect, forecasting tool for this system

Conclusion

The objective of this project was to identify weather factors influencing alpine butterfly (Parnassius smintheus) population dynamics and to develop a model capable of predicting population size across meadows and years. Exploratory analysis showed strong temporal autocorrelation and substantial variability among meadows, as well as non-constant variance in Nt, motivating the use of an AR(1) term and a square-root transformation of the response.

Four primary linear models were constructed and compared: two selected through BIC-based stepwise regression and two built from a biological hypothesis relating overwinter conditions, snowmelt timing, and summer heat to population size. Within-sample comparisons using partial F-tests and BIC favored models including an interaction between lagged population size and extreme July temperature. A Negative Binomial GLM was also examined but performed far worse according to BIC, reinforcing the suitability of the transformed Gaussian approach.

Out-of-sample evaluation using a leave-one-meadow-out design (excluding meadow G) showed that all models achieved similar predictive accuracy, but the square-root version of Model 2 consistently produced the lowest RMSE. This model included Nt_lag1, marsnow, aprsnowfall, julextmax, and the Nt_lag1 x julextmax interaction, capturing both density dependence and weather-driven modification of survival and reproduction.

Diagnostic checks for the final model revealed only mild heteroscedasticity, moderate tail deviations in the Q–Q plot, and minimal remaining autocorrelation, indicating that model assumptions were reasonably satisfied. A final forecast for 2015 across all meadows produced an RMSE of about 111 butterflies, reflecting moderate accuracy given the wide range of observed population sizes.

Overall, the analysis indicates that previous population size, winter and early-spring snow conditions, and summer heat jointly shape butterfly population fluctuations. The final square-root linear model with an AR(1) structure provides a interpretable and reasonably accurate tool for understanding and forecasting P. smintheus population dynamics.

Sources

- Population Dynamics of Alpine Parnassius smintheus Butterflies - Prof Jens Roland (YouTube Video)