Lab 2: Introduction to modeling in R

Your name

2024-01-22

### **Learning Objectives**

* Understand how to construct a simple process-based model in R
* Import external data as a data frame
* Summarize the contents of a data frame
* Produce and interpret degree-day model predictions

### **Grading Rubric**

Most of your grade will be for effort - i.e., that you did your best to follow instructions and to answer questions. Each lab is worth 30 pts, so divide this number by the total number of questions to estimate the approximate number of points per question. Points may be deducted for errors in code, not following instructions, or answering questions incorrectly.

### **Instructions**

There are questions associated with each exercise below to complete either during class or on your own time. Record answers to the questions **within** the Markdown (.Rmd) file and record your name on line 2. Once you’re done with the lab, knit the R markdown file as a Word document (.docx) within RStudio by clicking the Knit button found in the toolbar (look for a ball of yarn with a needle sticking out of it) and selecting Knit to Word. Ignore any warning messages that pop up in the Console during knitting. Submit the .Rmd **and** the .docx files on Canvas.

### **Introduction**

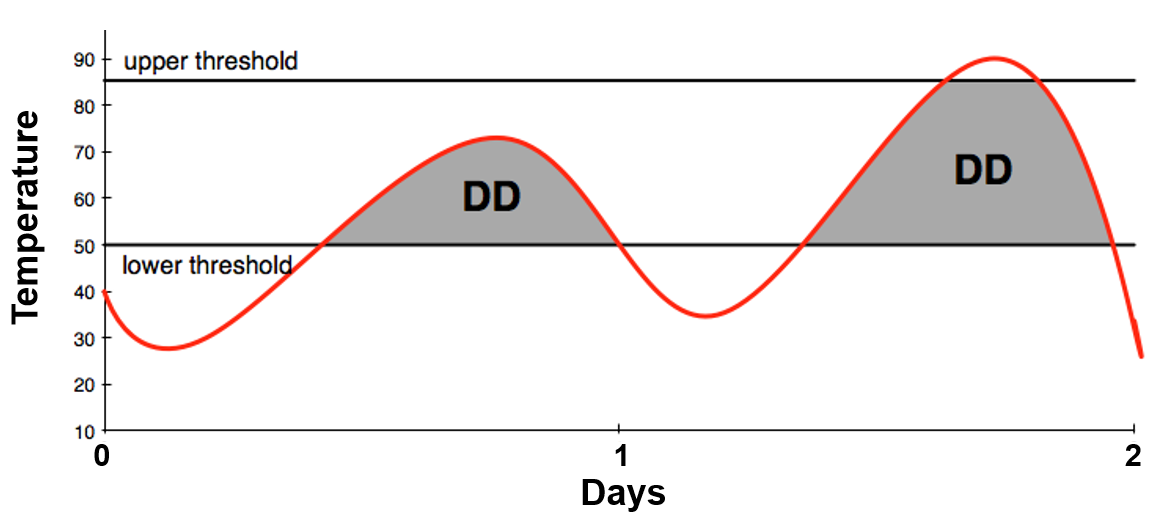
We will develop several **process-based** models in this course. In ecology, process-based models are based on a theoretical understanding of relevant ecological processes. Relationships are described in terms of explicitly stated processes or mechanisms, and model parameters therefore have a clear ecological interpretation. An example of a process-based model in ecology includes the process of how radiation, water, and temperature affect the rate of photosynthesis. Process-based models can have a spatial and/or temporal component. For example, you could use a process-based model for photosynthesis to study how rates differ across different latitudes or how they change over the course of a year.

For this lab, you will learn how to develop a process-based model that predicts degree-day accumulation over time. **Degree-day** models are often used to predict the phenology of ectothermic organisms (poikilothermic) such as insects, plants, and fungi. These organisms require a consistent amount of heat accumulation to reach certain life stages, such as egg hatch or adult flight for insects, or leaf out and flowering in plants. A **degree-day** is a measurement of heat units over time (i.e., heat accumulation), calculated from daily maximum and minimum temperatures ( and , respectively). DD models typically use a daily or hourly **time step**, meaning that DDs are calculated for each day or hour, respectively. The model tracks the accumulation of these DDs over a specific time period, such as a season or an entire year.

#### **Calculating Degree-Days**

The at which an ectothermic organism begins to develop is called the “lower developmental threshold” (), or baseline. The at which development stops is called the “upper developmental threshold (),” or cutoff. The and vary among species, and have been determined for many economically important pests.

Threshold values are used in calculating species-specific DDs. DD calculations are based on the area under the diurnal temperature curve and between the thresholds. A visual representation of DDs for two 24-hour periods is below. The area in black under the curve represents the number of DDs that fall between a and , for each 24-hour period.



#### Degree-Day Model Assumptions

1. Temperatures of a 24-hour day follow a defined curve
2. The rate of development is presumed to be a straight line directly related to temperature

#### Simple Average

DDs can be calculated using a simple formula for the average daily temperature, calculated from the daily and , minus the

[(daily + )/2] –

For example, a day where the high is 72F and the low is 44F would accumulate 8 DDs using 50F as the :

For a site with and data available, the different methods for calculating DDs are single sine, double sine, single triangle, and double triangle.

### **A Simple Degree-Day Model**

#### **Lower Developmental Threshold (LDT)**

In the following exercise, we will use the simple formula to calculate DDs using a of 50F. This is commonly used in DD models for plants and insect pests. The weather data used in our model is in degrees Fahrenheit.

Below we create objects called LDT50F that defines the lower developmental threshold as 50F. Additionally, we get information on its class and specific data type. The is defined as 50 using the assignment operator <-.

# Lower development threshold  
LDT50F <- 50

**(1)** What is the class and specific data type of LDT50F object? Show your work in the code chunk below.

*Response*:

**(2)** Say you’re working on a species with an of 54F. In the code chunk below, create an object named LDT54F that defines this threshold.

# LDT54F   
#   
#

**(3)** In the code chunk below, calculate DDs using your two thresholds, LDT50F and LDT54F. Use 80.1 and 47.5 as and , respectively. Which threshold resulted in a larger DD value and why?

# tmax = 80.1  
# tmin = 47.5

*Response*:

#### **Importing and Formatting Weather Data**

Here we import weather data derived from an Agrimet station in Corvallis in 2021 (station ID = CRVO). The here() function is used to define project-relative paths for files. The weather data file (CRVO21.txt) is in the weather\_data subfolder of Lab2\_intro\_R\_modeling.

# Path of weather data file  
weath\_fl <- here("Lab2\_model\_building\_R", "weather\_data", "CRVO21.txt")

The weather data file is a space-delimited text file. We will import it using the [read\_table()](https://www.rdocumentation.org/packages/readr/versions/2.1.3/topics/read_table) function from the [readr](https://readr.tidyverse.org/) package (part of Tidyverse). We will talk more about the Tidyverse soon. The first row is skipped using skip and the seventh column is skipped using col\_skip because they contain comments rather than data. The column names are defined using the colnames() function in base R.

# Import weather data  
weath\_tbl <- read\_table(weath\_fl, skip = 1, col\_names = FALSE,   
 col\_types = cols(X7 = col\_skip()))

Next we convert the table to a data frame using the data.frame() function. Additionally, the colnames() function is used to define the column names.

# Convert the table to a data frame and name the columns  
weath\_data <- data.frame(weath\_tbl)  
colnames(weath\_data) <- c("month", "day", "tmax", "tmin", "prec", "dd50")

**(4)** In the code chunk below, use recently learned functions to get the dimensions and data structure of the weather data. Next, write code to peek at the first and last 6 rows of data. What are two things that you learned about the data set?

# Dimensions  
  
# Structure  
  
# Head  
  
# Tail

*Response*:

#### **The Daily Time Step**

The simple DD model will predict the development of a hypothetical organism in 2021. To do this, it will use a daily time step to calculate DDs for each day and accumulate these DDs over the entire year. In R, this can be accomplished using a for loop. First, we will specify the time period as the number of rows in the weather data.

**(5)** In the code chunk below, create an object named number\_of\_days that uses the nrow() function to get this information.

# How many rows in the data frame?  
number\_of\_days <- nrow(weath\_data)

#### **Model Components**

The first part of the loop is below. Notice that we’re using a variable named i as an index that is iteratively replaced by each value in the vector 1:number\_of\_days. The loop starts with 1 and stops once it reaches the last value in number\_of\_days. An if statement is used to change negative DD values to 0.

# Step through days in year  
for (i in 1:number\_of\_days) {  
   
 # Tmin and Tmax for day of year  
 tmax <- weath\_data$tmax[i]  
 tmin <- weath\_data$tmin[i]  
   
 # Use simple average DDs, more complex formulas are available and preferable  
 dd\_today <- (tmax + tmin/2) - LDT50F  
   
 # Can't have negative degree-days (change to 0)  
 if (dd\_today < 0) {  
 dd\_today <- 0  
 }  
}

The model is incomplete because it’s only calculating DDs for a single day. DD models work by accumulating DDs over time to predict phenology - i.e., predicting and tracking the development of an organism over a given time frame. Thus, an object is needed to track DD accumulation, which we’ll call dd\_accum.

# Initialize an object for degree-day accumulation  
dd\_accum <- 0  
  
# Step through days of year  
for (i in 1:number\_of\_days) {  
  
 # Tmin and Tmax for day of year  
 tmax <- weath\_data$tmax[i]  
 tmin <- weath\_data$tmin[i]  
   
 # Use simple average DDs, more complex formulas are available and preferable  
 dd\_today <- (tmax + tmin/2) - LDT50F  
   
 # Can't have negative degree-days (change to 0)  
 if (dd\_today < 0) {  
 dd\_today <- 0  
 }  
   
 # Accumulate degree-days  
 dd\_accum <- dd\_accum + dd\_today  
}

It’s often helpful to store results of a model in a data frame that includes other information, such as the names of your study sites, sampling dates, etc. Below, we create an empty data frame to store results of the DD model. The first two columns simply copy the month and day from the input weather data (month and month\_day), whereas the remaining columns store the DDs for each day (dd\_today) and the total number of accumulated DDs (dd\_accum). One way to [create an empty data frame](https://www.statology.org/create-empty-data-frame-in-r/) is to first make a matrix with a specified number of columns and rows, and then name the columns.

# Create an empty data frame to store accumulated DDs  
out\_all\_LDT50 <- data.frame(matrix(ncol = 4, nrow = 0))  
colnames(out\_all\_LDT50) <- c("month", "month\_day", "dd\_today", "dd\_accum")

Over the daily time step, results of the model for each day of year (i) will be saved in a single row. Two columns extract month (month) and day of the month (month\_day) information from the weather data using i as an index. The $ symbols signifies a column.

# Put results for day of year in a data frame  
out\_i <- data.frame("doy" = i,   
 "month" = weath\_data$month[i],  
 "month\_day" = weath\_data$day[i],  
 "dd\_today" = dd\_today,  
 "dd\_accum" = dd\_accum)

Results for each day are tacked onto the out\_all\_LTD50 data frame using the [bind\_rows()](https://dplyr.tidyverse.org/reference/bind.html) function in the [dplyr](https://dplyr.tidyverse.org/) package (part of Tidyverse). We’ll learn more about this next week.

# Attach results for day to all results   
out\_all\_LDT50 <- bind\_rows(out\_all\_LDT50, out\_i)

#### **Run the DD Model Using a LDT of 50F**

The entire DD model is below.

# Initialize an object for degree-day accumulation  
dd\_accum <- 0  
  
# Step through days of year  
for (i in 1:number\_of\_days) {  
  
 # Tmin and Tmax for day of year  
 tmax <- weath\_data$tmax[i]  
 tmin <- weath\_data$tmin[i]  
   
 # Use simple average DDs, more complex formulas are available and preferable  
 dd\_today <- (tmax + tmin/2) - LDT50F  
   
 # Can't have negative degree-days (change to 0)  
 if (dd\_today < 0) {  
 dd\_today <- 0  
 }  
   
 # Accumulate degree-days  
 dd\_accum <- dd\_accum + dd\_today  
   
 # Put results for day in a data frame  
 out\_i <- data.frame("doy" = i,  
 "month" = weath\_data$month[i],  
 "month\_day" = weath\_data$day[i],  
 "dd\_today" = dd\_today,  
 "dd\_accum" = dd\_accum)  
   
 # Attach results for day to all results   
 out\_all\_LDT50 <- bind\_rows(out\_all\_LDT50, out\_i)   
  
}

The out\_all\_LDT50 data frame now contains DD results for each day of the year, as we can see by peeking at both the top and end of the data frame using head() and tail(), respectively. You can explore the data frame more thoroughly by clicking on it in the Environment tab of the Environment/History/etc. pane.

**(6)** What was the minimum and maximum amount of DD accumulation for 2021? Hint: Use functions such as min() on max() for the dd\_accum column of the out\_all data frame, or the summary() function for the entire data frame.

*Response*:

#### **Run the DD Model Using a LDT of 54F**

**(5)** Compare model outputs produced using an LDT of 50F vs. 54F. The easiest way to do this is to simply copy/paste the previous code chunk and then replace the threshold value (hint: you defined it above, in **Q2**). However, you’ll need a new data frame to store your results to avoid overwriting your previous results (i.e., out\_all\_LDT50). I did this part for you below.

# Create an empty data frame to store accumulated DDs based on a LDT of 54F  
out\_all\_LDT54 <- data.frame(matrix(ncol = 4, nrow = 0))  
colnames(out\_all\_LDT54) <- c("month", "month\_day", "dd\_today", "dd\_accum")  
  
# Now create and run your loop below

**(7)** What was the minimum and maximum amount of DD accumulation based on a LDT of 54F?

*Response*:

**(8)** Which LDT resulted in higher DD accumulation? Explain the biological significance of this result. Specifically, explain what it signifies in terms of the organism’s development for 2021.

*Response*:

**(9)** Discuss your overall results from this lab exercise and explain how the DD model is an example of a process-based model.