Lab 2: Degree-Day Modeling

Your name

2024-01-23

### **Learning Objectives**

* Produce, visualize, and interpret phenology model predictions
* Describe how and why phenology varies both within and among years
* Practice using base R and tidyverse functions

### **Grading Rubric**

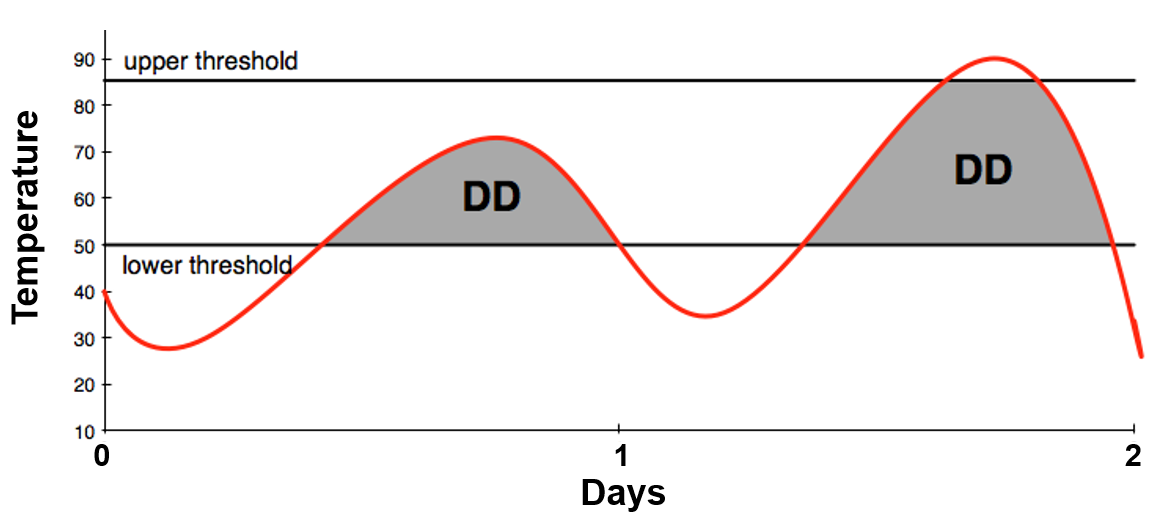
Most of your grade will be for effort - i.e., that you did your best to follow instructions and to answer questions. Each lab is worth 30 pts, so divide this number by the total number of questions to estimate the approximate number of points per question. Points may be deducted for errors in code, not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

### **Instructions**

There are a set of questions associated with each exercise below that you must 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.

### **Review: Calculating Degree-Days**

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 gray areas in the curves represent the number of DDs that fall between a and , for each 24-hour period.



Assumptions of DD models include:

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

#### **Calculation Method: 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 simple average, single triangle, double triangle, single sine, and double sine. More complex DD equations may result in more precise DD estimates.

### **Exercise 1: Modeling Phenology of the Asian Longhorned Beetle**

In the following exercise, we will use the simple formula to model the phenology of [Asian longhorned beetle](https://en.wikipedia.org/wiki/Asian_long-horned_beetle), *Anoplophora glabripennis*. The Asian loghorned beetle is an invasive insect in North America that poses a serious threat to hardwood trees in natural and urban areas. It currently infests areas in Massachusetts, New York and Ohio. USDA APHIS estimates that the beetle has the potential to cause more damage than Dutch elm disease, chestnut blight and gypsy moths combined.



#### **Degree-Day Model**

First, we import weather data for Corvallis in 2021 derived from an Agrimet station (station ID = CRVO).

The code chunk below contains code presented in Lab 2. However, notice that some base R functions were replaced with tidyverse functions.

# Import and format weather data  
# First line is skipped because it's descriptive info (not data)  
weath\_data <- read\_table(file = here("Lab3\_phenology\_models", "weather\_data", "CRVO21.txt"),  
 skip = 1, col\_names = FALSE,  
 col\_types = cols(X7 = col\_skip())) %>%  
 data.frame() %>%  
 rename(c("month" = 1, "day" = 2, "tmax" = 3, "tmin" = 4, "prec" = 5, "dd50" = 6))

Below, we create an empty data frame to store results of the DD model and define the number of days, as we did in Lab 2.

# Create an empty data frame to store accumulated DDs  
out\_all <- data.frame(matrix(ncol = 4, nrow = 0))  
colnames(out\_all) <- c("month", "month\_day", "dd\_today", "dd\_accum")  
  
# Number of days for daily time step  
number\_of\_days <- nrow(weath\_data)

**(1)** In the code chunk below, use code from Lab 2 to create a simple DD model for Asian long-horned beetle. The model for this species at [USPest.org](http://uspest.org) uses an LDT of 50F. Remember to:  
(1) define the LDT (name it LDT)  
(2) initialize an object for DD accumulation (dd\_accum)  
(3) use a for loop for the daily time step  
(4) include informative comments (as in Lab 2)

**IMPORTANT NOTE**: there was a typo in Lab2 (sorry)! The formula for the simple equation should be: dd\_today <- (tmax + tmin)/2 - LDT

Run the model when you’re done.

# Insert your code here  
LDT <- 50  
  
# 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 - LDT  
   
 # 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 <- bind\_rows(out\_all, out\_i)   
}

#### **Get Dates of Phenological Events**

We’d like to know when to expected specific phenological events for Asian longhorned beetle, particularly those that are relevant to its surveillance and management. According the USPest.org model, 1% adult emergence occurs at 806 DDs and first egg hatch occurs at 1438 DDs. On which dates would these events occur in 2021 according to our simple model? To address this question, we can use the [filter()](https://dplyr.tidyverse.org/reference/filter.html) and [slice\_min()](https://dplyr.tidyverse.org/reference/slice.html) functions in the dplyr package to subset the rows of the out\_all data frame in which dd\_accum is closest to these values. The date (day of year) of the event is saved as a new object, adult\_emerg\_doy.

**(2)** Below is the code for 1% adult emergence. Repeat this step for egg hatch, naming the two new objects egg\_hatch and egg\_hatch\_doy.

# Date of adult emergence  
adult\_emerg <- out\_all %>%  
 filter(dd\_accum >= 806) %>% # subset rows where dd\_accum >= 806  
 slice\_min(dd\_accum) # keep row with values closest to 806  
# View output  
adult\_emerg

## month month\_day dd\_today dd\_accum doy  
## 1 7 1 19.75 817.25 182

# Day of year of event  
adult\_emerg\_doy <- adult\_emerg$doy  
  
# Date of egg hatch  
egg\_hatch <- out\_all %>%  
 filter(dd\_accum >= 1438) %>% # subset rows where dd\_accum >= 1438  
 slice\_min(dd\_accum) # keep row with values closest to 1438  
egg\_hatch

## month month\_day dd\_today dd\_accum doy  
## 1 8 1 24.05 1443.4 213

egg\_hatch\_doy <- egg\_hatch$doy

#### **Plotting and Analysis**

We will create and save a plot of the results using the [ggplot()](https://www.rdocumentation.org/packages/ggplot2/versions/3.4.0/topics/ggplot) function of the ggplot2 package. A line plot is created using [geom\_line()](https://www.rdocumentation.org/packages/ggplot2/versions/1.0.1/topics/geom_line) to depict DD accumulation over the entire year.

First, we create a custom x-axis in which axis breaks correspond approximately with the first day of each month of the year (e.g., 1 = Jan 1, 32 = Feb 1, etc.). The custom x-axis labels x\_labels present each break as the first of each month (e.g., 1-1, 2-1, etc.). This is created by using the [paste()](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/paste) function to literally paste together all unique months of the year present in the data (in the out\_all$month column) to the number 1.

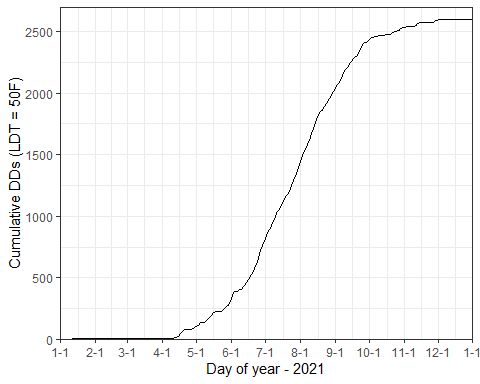
# Create and save a simple plot  
# New set of x-axis labels (dates vs. day of year)  
days <- c(1, 32, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 365)  
x\_labels <- paste(c(unique(out\_all$month), 1), 1, sep = "-")

Next, we pass out\_all to the data argument of the ggplot function, and the x and y variables to be plotted (day of year and DD accumulation, respectively) within the aesthetic ([aes()](https://www.rdocumentation.org/packages/ggplot2/versions/1.0.1/topics/aes)) function. A simple [plot theme](https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html#ggplot2_themes) (theme\_bw) is applied, titles for the x- and y-axis are added, and specifications are made for both axes. The + operator is used in ggplot2 to add layers to plots.

# Plot  
dd\_plot <- ggplot(data = out\_all, aes(x = doy, y = dd\_accum)) +  
 geom\_line() + # for trend lines, time series, etc.  
 theme\_bw() + # simple black and white plot theme  
 ylab("Cumulative DDs (LDT = 50F)") + # y-axis title  
 xlab("Day of year - 2021") + # x-axis title  
 # Both axes start at 0, x-axis uses custom breaks and labels  
 scale\_x\_continuous(expand = c(0, 0), breaks = days, labels = x\_labels) +  
 # Make the y-axis 100 units larger than the max value in data  
 scale\_y\_continuous(expand = c(0, 0),   
 limits = c(0, max(out\_all$dd\_accum) + 100))

Now let’s view the plot, and save it as a PNG file using ggsave().

# View plot  
dd\_plot

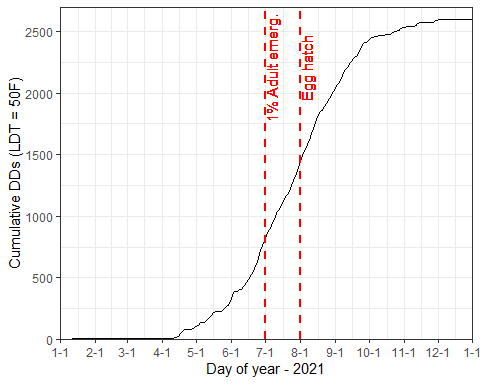


# Save plot  
ggsave(dd\_plot,   
 filename = here("Lab3\_phenology\_models", "plots", "DDaccum\_R.png"),  
 height = 3, width = 5, units = c('in'), device = "png", dpi = 300)

We can depict the dates of phenological events on our plot with dashed red lines using the [geom\_vline()](https://www.rdocumentation.org/packages/ggplot2/versions/0.9.1/topics/geom_vline) function and label them using [geom\_text()](https://www.rdocumentation.org/packages/ggplot2/versions/0.9.1/topics/geom_text). The day of year for each phenological event is used to define x-intercepts for these geoms (the label is shifted 5 days later so it doesn’t overlap with the line).

**(3)** The plot below shows the date for 1% adult emergence. Add the appropriate code to show the date for egg hatch.

dd\_plot +  
 geom\_vline(xintercept = adult\_emerg\_doy, color ="red", linetype = "dashed", size = 0.75) +  
 geom\_text(aes(x = adult\_emerg\_doy + 5, label="1% Adult emerg.", y = 2200), colour="red", size = 4, angle=90) +  
 geom\_vline(xintercept = egg\_hatch\_doy, color ="red", linetype = "dashed", size = 0.75) +  
 geom\_text(aes(x = egg\_hatch\_doy + 5, label="Egg hatch", y = 2200), colour="red", size = 4, angle=90)



### **Exercise 2: Annual Variation in Phenology of Asian Longhorned Beetle**

In the next exercise, we’ll expand on the example above by modeling phenology of Asian longhorned beetle across multiple years. The only differences in the model is that (1) weather data for multiple years are needed, and (2) a second for loop is needed to model phenology over multiple years **and** days.

#### **Setup**

First, we create a vector of years between 2012 and 2021.

years <- c(2012:2021)  
years

## [1] 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021

Next we create a vector that contains the name of weather data files for each year using the [list.files()](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/list.files) function. Weather data are from the same Corvallis weather station as above.

all\_files <- list.files(here("Lab3\_phenology\_models", "weather\_data"))  
all\_files

## [1] "CRVO12.txt" "CRVO13.txt" "CRVO14.txt" "CRVO15.txt" "CRVO16.txt"  
## [6] "CRVO17.txt" "CRVO18.txt" "CRVO19.txt" "CRVO20.txt" "CRVO21.txt"

Following the first example, we’ll create an empty data frame to store the model results. However, this time the data frame will include a column named year so we can delineate which predictions correspond with which year.

out\_all2 <- data.frame(matrix(ncol = 5, nrow = 0))  
colnames(out\_all2) <- c("month", "day", "year", "dd\_today", "dd\_accum")

The loop for years starts with 1 and runs through all integer values up to the total number of years (10). A variable named j is used as an index that is iteratively replaced by each value in the vector 1:length(years). The [length()](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/length) function returns the length of vectors (or other object types), which in our case is 10. So, 1:length(years) returns all values between 1 and 10.

# Length of years  
length(years)

## [1] 10

#1:length of years  
1:length(years)

## [1] 1 2 3 4 5 6 7 8 9 10

#### **Multi-Year DD Model**

The entire multi-year DD model is below.

# Initialize an object for degree-day accumulation  
dd\_accum <- 0  
  
# Step through years  
for (j in 1:length(years)) {  
  
 # Weather data for year plus column names (skip column 7 with comments)  
 weath\_data <- read\_table(file = here("Lab3\_phenology\_models", "weather\_data", all\_files[j]),  
 skip = 1, col\_names = FALSE,  
 col\_types = cols(X7 = col\_skip())) %>%  
 data.frame() %>%  
 rename(c("month" = 1, "day" = 2, "tmax" = 3, "tmin" = 4, "prec" = 5, "dd50" = 6))  
   
 # Days of year in data (365 for non-leap years, 366 for leap years)  
 number\_of\_days <- nrow(weath\_data)  
   
 # Initialize an object for degree-day accumulation  
 dd\_accum <- 0  
   
 # Step through days (same daily loop as above)  
 for (i in 1:number\_of\_days) {  
   
 # Tmin and Tmax for day, but convert from degrees F to C  
 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 - LDT  
   
 # 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  
  
 # Add results to output data frame  
 out\_all2 <- rbind(  
 out\_all2, data.frame("doy" = i,  
 "month" = weath\_data$month[i],  
 "day" = weath\_data$day[i],  
 "year" = years[j],  
 "dd\_today" = dd\_today,  
 "dd\_accum" = dd\_accum)  
 )  
 }  
   
}

#### **Plotting and Analysis**

We’ll use ggplot2 again to plot and save the results. The only difference from above is that a group variable is defined as year within the aesthetic (aes) function. This tells ggplot that it needs to combine results for all years into a single plot. Additionally, the color variable in the aesthetic function tells ggplot to color lines by year. A custom color palette for the lines is applied using [scale\_color\_brewer()](https://ggplot2.tidyverse.org/reference/scale_brewer.html).

One small edit is needed before plotting the result. Currently, the year column is in integer format, which causes ggplot to use a continuous color palette rather than a categorical color palette. We can solve this problem by converting year to a factor.

str(out\_all2)

## 'data.frame': 3653 obs. of 6 variables:  
## $ doy : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ month : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ day : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ year : int 2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...  
## $ dd\_today: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ dd\_accum: num 0 0 0 0 0 0 0 0 0 0 ...

out\_all2$year <- factor(out\_all2$year)  
str(out\_all2)

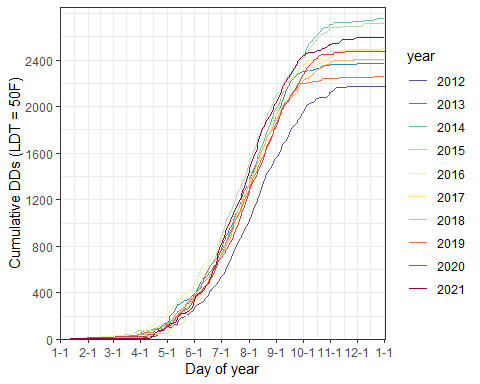
## 'data.frame': 3653 obs. of 6 variables:  
## $ doy : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ month : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ day : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ year : Factor w/ 10 levels "2012","2013",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ dd\_today: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ dd\_accum: num 0 0 0 0 0 0 0 0 0 0 ...

Create the plot.

# Plot  
dd\_plot2 <- ggplot(data = out\_all2, aes(x = doy, y = dd\_accum, group = year, color = year)) +  
 geom\_line() + # for trend lines, time series, etc.  
 theme\_bw() + # simple black and white plot theme  
 ylab("Cumulative DDs (LDT = 50F)") + # y-axis title  
 xlab("Day of year") + # x-axis title  
 # Custon color palette  
 scale\_color\_brewer(palette="Spectral", direction = -1) +  
 # Both axes start at 0, x-axis uses custom breaks and labels  
 scale\_x\_continuous(expand = c(0, 0), breaks = days, labels = x\_labels) +  
 # Make the y-axis 100 units larger than the max value in data  
 scale\_y\_continuous(expand = c(0, 0),   
 breaks = seq(0, max(out\_all2$dd\_accum), 400),  
 limits = c(0, max(out\_all2$dd\_accum) + 100))

View the plot and save it as a PNG file.

dd\_plot2



# Save plot  
ggsave(dd\_plot2,   
 filename = here("Lab3\_phenology\_models", "plots", "DDaccum\_10yr\_R.png"),  
 height = 3, width = 5, units = c('in'), device = "png", dpi = 300)

To determine the predicted dates of egg hatch by Asian longhorn beetle over the 10 year period, the filter() function is again used to subset the rows of the out\_all2 data frame in which dd\_accum is closest to 1438. However, the group\_by() function in the dplyr package is used to group the output by the year column before filtering because we’re interested in knowing dates for every single year.

egg\_hatch2 <- out\_all2 %>%  
 group\_by(year) %>% # group by year  
 filter(dd\_accum >= 1438) %>% # subset rows where dd\_accum >= 1438  
 slice\_min(dd\_accum) # keep row with values closest to 1438  
egg\_hatch2

## # A tibble: 10 × 6  
## # Groups: year [10]  
## doy month day year dd\_today dd\_accum  
## <int> <dbl> <dbl> <fct> <dbl> <dbl>  
## 1 235 8 22 2012 15.1 1446.  
## 2 218 8 6 2013 22.9 1449.  
## 3 217 8 5 2014 17.3 1444.  
## 4 211 7 30 2015 33.0 1455.  
## 5 219 8 6 2016 12.5 1446.  
## 6 219 8 7 2017 25.2 1443.  
## 7 219 8 7 2018 24.2 1448.  
## 8 221 8 9 2019 18.5 1438.  
## 9 223 8 10 2020 25.1 1440.  
## 10 213 8 1 2021 24.0 1443.

**(4)** On which years was egg hatch predicted to be earliest vs. latest? What does this tell you about the temperatures on those two years?

*Response*:

**(5)** Compare your results for *3 years* (e.g., 2019, 2020, and 2021) to those produced by the USPest.org model. Follow these instructions:

1. Visit the model app at [USPest.org](https://uspest.org/dd/model_app).
2. In the Station tab, type in CRVO and select this station.
3. In the Model tab, select “asian longhorned beetle” and enter a Start Date of January 1 for each of your chosen years. Keep default values for other options.
4. Click on the Output tab and check the “show model inputs table” in the Model Inputs section.
5. Look at the Model Output section (you can also look at the graph) and document the date of 1st egg hatch.

According to the USPest.org model, when did egg hatch occur for each of your 3 chosen years? Are these dates earlier or later than dates predicted by your model (i.e., dates in egg\_hatch2)? Optional: enter a code chunk below if it helps in making comparisons.

*Response*:

**(6)** The model at USPest.org uses the single sine formula to calculate degree-days. This method uses the day’s and to produce a sine curve over a 24-hour period, and then estimates DDs for that day by calculating the area above the threshold and below the curve. Why is this method likely to be more accurate than the simple average method used in our model?

*Response*:

**(7)** Name at least one assumption made by both models that could be a source of error in predicting phenology.

*Response*:

**(8)** As this exercise demonstrated, the timing of phenological events can vary across years. Think of your favorite ecological system and describe how a phenology model might help you make better management decisions. For example, you might think about applications for invasive species, species of conservation concern, or farming.

*Response*:

**(9)** Discuss your overall results from the phenology model exercises.

*Response*: