

Faculty notes - Working with plant phenology data and fitting a nonlinear model using least squares in R

Madeleine Bonsma-Fisher and Ahmed Hasan

Overview of module

This module was presented as part of a participatory live-coding class for third and fourth-year undergraduate ecology students. In this lesson, we plot a subset of NEON phenology data and fit an oscillatory model to determine when different species get and lose their leaves. The module contains an optional section that was not presented in class on calculating growing degree days (GDD) and plotting leaf cover vs. growing degree days. The student notes are designed to be presented through participatory live-coding: the instructor's live coding is projected to the class while they follow along on their own computers.

- **Estimated time:** 1.5 hours (participatory live-coding, not including GDD extension)
- **Target audience:** 3rd to 4th year ecology or biology students
- **Student prerequisites:**
 - Ecological concepts:
 - * Basics of plant phenology
 - * Basics of population cycling
 - Quantitative skills:
 - * Familiarity with R and the [tidyverse](#).
 - * Familiarity with the concept of least squares fitting and minimizing the sum of squares of residuals

Learning objectives

Upon completion of the module, students will be able to:

- Import and manipulate discrete time series data in R:
 - Join data frames together
 - Convert dates to the `Date` class in R
 - Use `dplyr` functions such as `group_by`, `mutate`, `case_when` to restructure data
- Plot time series data using the `ggplot2` package in R
- Define custom functions in R and apply them to vectors with `mapply` and `sapply`
- Fit a sine wave model to ecological time series data using least squares
- Explain the following ecological concepts:
 - Phenology
 - Growing Degree Days (if optional extension completed)

Data description

The data used in this module is a subset of NEON plant phenology data, downloaded from the NEON data tutorial “[Work With NEON’s Plant Phenology Data](#)”. The data download link is [here](https://ndownloader.figshare.com/files/9012085) (<https://ndownloader.figshare.com/files/9012085>). The three files used are `pheno/phe_perindividual.csv`, `pheno/phe_statusintensity.csv`, and `temp/SAAT_30min.csv`.

These data are from the three terrestrial field sites in NEON’s Domain 02, the Mid Atlantic, and include BLAN, SCBI, and SERC. For field site descriptions and full names see the NEON field sites webpages (<https://www.neonscience.org/field-sites/field-sites-map>). The data are from June 2015 to December 2016.

Faculty who want to use a more recent data or data from another location are encouraged to use the NEON data portal (data.neonscience.org) to create a customized data set.

Teaching challenges to address

When this module was implemented as a participatory live-coding lesson, we didn't include the data processing steps (everything before *Plotting and fitting the data*). We provided students with a direct link to the joined and filtered dataframe. If the processing steps are covered in a class setting, they may be confusing or slow for students to get through.

Students struggled to follow along during the date conversion steps, which caused some downstream issues when the date columns hadn't been properly converted to use as the x-axis for fitting a function.

If presenting this material or a version of it as a live-coding exercise, be cautious about the speed at which the material can be covered and be prepared for it to take longer than expected.

Assessment strategy

We assessed students' ability to fit an oscillatory model to data with the following assignment question. Advanced classes could also consider using a form of spectral decomposition to analyze periodic data.

The Canadian lynx population cycle

The Canadian lynx experiences large periodic changes in its population size over a timescale of several years. This is thought to be driven by oscillations in the population size of the snowshoe hare, the primary food source for the lynx. Read more about the lynx population cycle on this [Northwest Territories website] (<https://www.enr.gov.nt.ca/en/services/lynx/lynx-snowshoe-hare-cycle>).

R has a built-in dataset called `lynx` which contains annual population measurements for the Canadian lynx as a time series.`

- (a) Plot `lynx` vs. time in years using either `ggplot` or `qplot`. Plot points (`geom_point`) and a connecting line (`geom_line`). Create a time series that starts at 0 and ends at the total number of years in the dataset (total years = 1934 - 1821). By eye, estimate the time between peaks in the population.

SOLUTION

```
years = seq(0, 1934-1821, by = 1)
qplot(x = years, y = lynx) +
  geom_line()
```

```
# optional: make data frame
lynx_df = data.frame(lynx)
lynx_df["time"] = years
```

the time between peaks is about 10 years.

- (b) Define a function called `sine_model` that takes 5 arguments: a vector of years for the x-axis and four parameters (amplitude, period, phase, and offset). Recall the general formula for a sine wave:

$$y = A\sin(kx - b) + c$$

where $k = 2\pi/T$, T is the period or length of time between peaks, A is the amplitude, b is the phase, and c is the offset. Using a value of $A = c = 1700$ for both the amplitude and offset and a value of $b = 2.75$ for the phase, plot the lynx data as before and add a sine curve using your guess of the

timescale from part (a) for the period. Use a colour other than black to plot the sine wave. Note that the x axis must start at 0 in order for the offset of 2.75 to match the data.

SOLUTION

```
sine_model <- function(x, amplitude, period, phase, offset) {  
  return(amplitude*sin(2*pi/period*x + phase) + offset)  
}
```

```
guess_curve <- sine_model(years, 1700, 10, -2.75, 1700)
```

```
qplot(x = years, y = lynx) +  
  geom_line() +  
  geom_line(aes(x = years, y = guess_curve, colour = 'red'))
```

(c) Use least-squares fitting to refine your estimate of the lynx cycle length. (1.5 marks)

- Make a range of values for the period that span your guess from part (a).
- Use the `sapply` function to calculate a predicted dataset using a sine model. Calculate the sum of the difference (*residuals*) between the lynx data and your prediction, then return the sum of the residuals squared.
- Plot the sum of the residuals squared vs. the range of period values. By eye, what is the minimum of this curve? What value of the period gives the best fit?
- Use the function `which` to extract the period value that gives the best fit. What is your calculated length of the lynx population cycle?

SOLUTION

```
# Make a range of b parameter values to try
```

```
lambda_vals <- seq(9, 10, by = 0.01)
```

```
amplitude <- 1700
```

```
offset <- 1700
```

```
b <- -2.75
```

```
# use the function 'sapply' to loop over b_vals list
```

```
resids_sq <- sapply(lambda_vals, function(lambda) {  
  prediction <- amplitude*sin(2*pi/lambda*years + b) + offset  
  residuals <- prediction - lynx  
  sum(residuals^2)  
})
```

```
qplot(lambda_vals, resids_sq)
```

```
# by eye, the best fit is around 9.62
```

```
best_fit <- which(resids_sq == min(resids_sq))
```

```
l_fit <- lambda_vals[best_fit]
```

```
l_fit
```

(d) Plot the lynx data again and plot your best fit curve on top. Does your estimate of the cycle length match the literature? Why or why not? (Find and cite a resource that gives an estimate of the lynx population cycle.)

SOLUTION

```
fit_curve <- sine_model(years, 1700, l_fit, -2.75, 1700)
```

```
qplot(x = years, y = lynx) +
```

```
geom_line() +  
geom_line(aes(x = years, y = fit_curve, colour = 'red'))
```