# EDS 223: week 8 lab notes

Ruth Oliver

2022-11-16

## Overview

Phenology is the timing of life history events. Important phenological events for plants involve the growth of leaves, flowering, and senescence (death of leaves). Plants species adapt the timing of these events to local climate conditions to ensure successful reproduction. Subsequently, animal species often adapt their phenology to take advantage of food availability. As the climate shifts this synchronization is being thrown out of whack. Shifts in phenology are therefore a common yardstick of understanding how and if ecosystems are adjusting to climate change.

Plant species may employ the following phenological strategies:

- winter deciduous: lose leaves in the winter, grow new leaves in the spring
- drought deciduous: lose leaves in the summer when water is limited
- evergreen: maintain leaves yearround

credit: this lab is based on a materials developed by Chris Kibler.

## Task

In this lab we are analyzing plant phenology near the Santa Clara River which flows from Santa Clarita to Ventura. We will investigate the phenology of the following plant communities:

- riparian forests: grow along the river, dominated by winter deciduous cottonwood and willow trees
- grasslands: grow in openspaces, dominated by drought deciduous grasses
- chaparral shrublands: grow in more arid habitats, dominated by evergreen shrubs

To investigate the phenology of these plant communities we will a time series of Landsat imagery and polygons identifying the locations of study sites within each plant community.

## Summary

- load Landsat scenes
- rename Landsat layers
- compute NDVI for each scene
- extract NDVI in study sites
- tidy data and plot results

## Data

### Landsat Operational Land Imager (OLI sensor)

- 8 pre-processed scenes
  - Level 2 surface reflectance products
  - erroneous values were set to NA
  - scale factor set to 100
  - bands 2-7
  - dates in filenname

### Study sites

- polygons representing sites
  - study_site: character string with plant type

# Workflow

```r
library(terra)
library(sf)
library(dplyr)
library(tidyr)
library(stringr)
library(ggplot2)
library(here)
library(tmap)

rm(list = ls())
here::i_am("labs/week8-notes.Rmd")
```

**load Landsat data** Here we load 8 scenes collected by the Landsat OLI sensor.

```
landsat_20180612 <-rast(here("labs", "data", "week8", "landsat_20180612.tif"))
landsat_20180815 <- rast(here("labs", "data", "week8", "landsat_20180815.tif"))
landsat_20181018 <- rast(here("labs", "data", "week8", "landsat_20181018.tif"))
landsat_20181103 <- rast(here("labs", "data", "week8", "landsat_20181103.tif"))
landsat_20190122 <- rast(here("labs", "data", "week8", "landsat_20190122.tif"))
landsat_20190223 <- rast(here("labs", "data", "week8", "landsat_20190223.tif"))
landsat_20190412 <- rast(here("labs", "data", "week8", "landsat_20190412.tif"))
landsat_20190701 <- rast(here("labs", "data", "week8", "landsat_20190701.tif"))
```

**rename layers** Update the layer names to match the spectral bands they correspond to.

```
names(landsat_20180612) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
names(landsat_20180815) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
names(landsat_20181018) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
names(landsat_20181103) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
names(landsat_20190122) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
names(landsat_20190223) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
names(landsat_20190412) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
names(landsat_20190701) <- c("blue", "green", "red", "NIR", "SWIR1", "SWIR2")
```

**create NDVI function** Here we define a function to compute the Normalized Difference Vegetation Index (NDVI). NDVI computes the difference in reflectance in the near infrared and red bands, normalized by their sum.

```
ndvi_fun = function(nir, red){
  (nir - red) / (nir + red)
}
```

**compute NDVI** Here we apply the NDVI function we created to compute NDVI for each image using the `lapp()` function. The `lapp()` function applies a function to each cell using layers as arguments. Therefore, we need to tell `lapp()` which layers (or bands) to pass into the function. The NIR band is the 4th layer and the red band is the 3rd layer in our raster. In this case, because we defined the NIR band as the first argument and the red band as the second argument in our function, we tell `lapp()` to use the 4th layer first and 3rd layer second.

```
ndvi_20180612 <- lapp(landsat_20180612[[c(4, 3)]], fun = ndvi_fun)
ndvi_20180815 <- lapp(landsat_20180815[[c(4, 3)]], fun = ndvi_fun)
ndvi_20181018 <- lapp(landsat_20181018[[c(4, 3)]], fun = ndvi_fun)
ndvi_20181103 <- lapp(landsat_20181103[[c(4, 3)]], fun = ndvi_fun)
ndvi_20190122 <- lapp(landsat_20190122[[c(4, 3)]], fun = ndvi_fun)
ndvi_20190223 <- lapp(landsat_20190223[[c(4, 3)]], fun = ndvi_fun)
ndvi_20190412 <- lapp(landsat_20190412[[c(4, 3)]], fun = ndvi_fun)
ndvi_20190701 <- lapp(landsat_20190701[[c(4, 3)]], fun = ndvi_fun)
```

**combine NDVI layers** Now we can combine the NDVI layers we created into a single raster stack.

```r
all_ndvi <- c(ndvi_20180612,
              ndvi_20180815,
              ndvi_20181018,
              ndvi_20181103,
              ndvi_20190122,
              ndvi_20190223,
              ndvi_20190412,
              ndvi_20190701)
```

Let's update the names of each layer to match the date of each image.

```r
names(all_ndvi) <- c("2018-06-12",
                     "2018-08-15",
                     "2018-10-18",
                     "2018-11-03",
                     "2019-01-22",
                     "2019-02-23",
                     "2019-04-12",
                     "2019-07-01")
```

```r


sites <- st_read(here("labs","data","week8","study_sites.shp"))

tm_shape(ndvi_20180612) +
  tm_raster() +
  tm_shape(sites) +
  tm_polygons()
```

**read in study sites**

**extract NDVI at study sites** Here we find the average NDVI within each study site. The output of extract is a data frame with rows that match the study site dataset, so we bind the results to the original dataset.

```r
sites_ndvi <- terra::extract(all_ndvi, sites, fun = "mean")

sites_annotated <- cbind(sites, sites_ndvi)
```

**clean results** We're done! Except our data is very untidy... Let's tidy it up!

- covert to data frame
- turn from wide to long format
- turn layer names into date format

```r
sites_clean <- sites_annotated %>%
  st_drop_geometry() %>%
  select(-ID) %>%
  pivot_longer(!study_site) %>%
  rename("NDVI" = value) %>%
  mutate("year" = str_sub(name, 2, 5),
         "month" = str_sub(name, 7, 8),
         "day" = str_sub(name, -2, -1)) %>%
  unite("date", 4:6, sep = "-") %>%
  mutate("date" = lubridate::as_date(date))
```

```
ggplot(sites_clean,
       aes(x = date, y = NDVI,
           group = study_site, col = study_site)) +
  geom_line()
```

**plot results**