An example of creating modular code in R - Efficient scientific programming

Learning Objectives

After completing this tutorial, you will be able to:

- Identify chunks of code that are well suited to becoming functions.

What you need

You will need a computer with internet access to complete this lesson and the data that we already downloaded for week 6 of the course.

{% include/data_subsets/course_earth_analytics/_data-week6-7.md %}

In this lesson, we will practice identifying modular or repeated tasks in your code and will work through the exercise of turning code written as a linear script into modular code that utilizes functions.

Have a close look at the code below. Are there components of the code that are repeated with slightly different argument values?

Setup R

```
knitr::opts_chunk$set(echo = TRUE, eval=F)
# set working dir
setwd("~/Documents/earth-analytics")

# load spatial packages
library(raster)
library(rgdal)
# turn off factors
options(stringsAsFactors = F)

# set colors for plotting
nbr_colors = c("seagreen4", "seagreen1",  "ivory1", "palevioletred1", "palevioletred4")
ndvi_colors = c("brown","ivory1","seagreen1","seagreen4")
```

Import Landsat data - Julian day 189 - pre fire

```
# get list of tif files
all_landsat_bands_pre <- list.files("data/week6/Landsat/LC80340322016189-SC20170128091153/crop",
                                    pattern=glob2rx("*band*.tif$"),
                                    full.names = T)

# stack landsat bands
landsat_stack_csf_pre <- stack(all_landsat_bands_pre)
```

Calculate NDVI - pre-fire

```
# calculate normalized index - NDVI
landsat_ndvi_pre <- (landsat_stack_csf_pre[[5]] - landsat_stack_csf_pre[[4]]) / (landsat_stack_csf_pre[

# create classification matrix
reclass <- c(-1, -.2, 1,
             -.2, .2, 2,
             .2, .5, 3,
             .5, 1, 4)
# reshape the object into a matrix with columns and rows
reclass_m <- matrix(reclass,
                    ncol=3,
                    byrow=TRUE)

ndvi_classified_pre <- reclassify(landsat_ndvi_pre,
                              reclass_m)

# plot classified data
plot(ndvi_classified_pre,
     box=F, axes=F, legend=F,
     col=ndvi_colors,
     main="NDVI - Pre fire")
legend(ndvi_classified_pre@extent@xmax, ndvi_classified_pre@extent@ymax,
        legend=c("No Vegetation", "Low Greenness", "Medium Greenness", "High Greeness"),
        fill = ndvi_colors, bty="n", xpd=T)


### export NDVI raster with unique name
writeRaster(x = ndvi_classified_pre,
            filename="data/week6/outputs/landsat_ndvi_pre.tif",
            format = "GTiff", # save as tif
            datatype='INT2S', # save as a INTEGER
            overwrite = T)  # overwrite previous file
```

Calculate Normalized Burn Ratio (NBR) - Pre fire

```
# calculate normalized index = NBR
landsat_nbr_pre <- (landsat_stack_csf_pre[[4]] - landsat_stack_csf_pre[[7]]) / (landsat_stack_csf_pre[[

# plot classified data
plot(landsat_nbr_pre,
     box=F, axes=F,
     main="Landsat NBR - Pre Fire \n Julian Day 189")
```

Open & Process Post-fire data

```
# get list of tif files
all_landsat_bands_post <- list.files("data/week6/Landsat/LC80340322016205-SC20170127160728/crop",
                                pattern=glob2rx("*band*.tif$"),
```

```
                                full.names = T)

# stack the data (create spatial object)
landsat_stack_csf_post <- stack(all_landsat_bands_post)
```

Calculate NDVI - post-fire

```
# calculate NDVI
landsat_ndvi_post <- (landsat_stack_csf_post[[5]] - landsat_stack_csf_post[[4]]) / (landsat_stack_csf_po

# create classification matrix
reclass <- c(-1, -.2, 1,
             -.2, .2, 2,
             .2, .5, 3,
             .5, 1, 4)
# reshape the object into a matrix with columns and rows
reclass_m <- matrix(reclass,
                    ncol=3,
                    byrow=TRUE)

ndvi_classified_post <- reclassify(landsat_ndvi_post,
                                   reclass_m)

#### Plot with legend
plot(ndvi_classified_post,
     box=F, axes=F, legend=F,
     main="NDVI - Post Fire",
     col=ndvi_colors)
legend(ndvi_classified_post@extent@xmax, ndvi_classified_post@extent@ymax,
       legend=c("No Vegetation", "Low Greenness", "Medium Greenness", "High Greeness"),
       fill = ndvi_colors, bty="n", xpd=T)

### Optional -- export NDVI raster with unique name
writeRaster(x = ndvi_classified_post,
            filename="data/week6/outputs/landsat_ndvi_post.tif",
            format = "GTiff", # save as a tif
            datatype='INT2S', # save as a INT
            overwrite = T)
```

Calculate NBR post fire

Next, calculate Normalized Burn Ratio (NBR).

```
# calculate normalized index = NBR
landsat_nbr_post <- (landsat_stack_csf_post[[5]] - landsat_stack_csf_post[[7]]) / (landsat_stack_csf_pos

# calculate difference NBR (pre - post)
landsat_nbr_diff <- landsat_nbr_pre - landsat_nbr_post

# create classification matrix
reclass <- c(-1.0, -.1, 1,
```

```
             -.1, .1, 2,
             .1, .27, 3,
             .27, .66, 4,
             .66, 1.3, 5)
# reshape the object into a matrix with columns and rows
reclass_m <- matrix(reclass,
                ncol=3,
                byrow=TRUE)

landsat_nbr_diff_class <- reclassify(landsat_nbr_diff,
                     reclass_m)

# plot classified data
plot(landsat_nbr_diff_class,
     box=F, axes=F, legend=F,
     col=nbr_colors,
     main="Landsat difference NBR - Post Fire \n Julian Day 205")
legend(landsat_nbr_diff_class@extent@xmax-100, landsat_nbr_diff_class@extent@ymax,
       c("Enhanced Regrowth", "Unburned", "Low Severity", "Moderate Severity", "High Severity"),
       fill=nbr_colors,
       cex=.9, bty="n", xpd=T)


writeRaster(x = landsat_nbr_diff_class,
              filename="data/week6/outputs/landsat_nbr_diff_class.tif",
              format = "GTiff", # save as a tif
              datatype='INT2S', # save as a INTEGER rather than a float
              overwrite = T)
```

Compare pre and post fire.

```
par(mfrow=c(2,1))
plot(landsat_nbr_pre, zlim=c(-1,1),
     main="pre-fire NBR")
plot(landsat_nbr_post, zlim=c(-1,1),
     main="post-fire NBR")
```