```
#=========================================================================================
# R class exercise 2: nested loops with lidar data
#=========================================================================================

# << INFORMATION >>
# LIDAR data: There are eight LIDAR datasets associated with this exercise, four from each
# of two surveys, named "lidar_s1_t1.csv" through "..s2_t4.csv". The Survey 2 data is
# actually just a reshuffled version of Survey 1 data, created for coding practice only.

# LIDAR stands for Light Detection and Ranging. It is like RADAR, but with lasers. For
# these datasets, a ground-based LIDAR was used to walk four forest transects and
# determine the vertical distribution of leaves and wood. Each file represents a vertical
# slice through the forest, one for each transect. Each cell contains calculated leaf area
# density (LAD), or the area of leaves in a 1x1.8m cube of space. Each file has exactly
# the same dimensions: 60 x 556, which corresponds to 60 m height and 1000 m length.

# << PLAN >>
# Nested loops to pull in all the data and clean it:
# - Make a nested for loop to work through two survey folders and each lidar dataset.
# - Use the loop to pull in the lidar datasets, concatenate them into a nested list.
# - The list should contain two nested lists, one for each survey.
# - Each sub-list should contain four matrices, one for each transect.
# - Loop through each matrix and turn all -9999 values into NA.

# Nested loops to analyze the data:
# - Create and export image plots from each matrix into a "Survey i_Results" folder.
# - Make a table of summary results with max.lad, sum.lad, and mean.lad for each height.
# - Have those summaries occur as data frames in each survey list.
# - Export the summary tables to .csv in the corresponding Results folder.

# << PACKAGES >>
#--Load the package 'fields' (if you run into errors later with image.plot(), you might
# have to re-install 'fields' to get the latest version).
library (fields)

# << DIRECTORIES >>
#--Directory path to the folder containing the lidar data folders.
dat.dir <- "/Users/ttaylor/Documents/R course/Tutorial datasets/"
#--Directory path to a results folder, which should contain empty folders named
# "Survey 1_Results" and "Survey 2_Results".
res.dir <- "/Users/ttaylor/Documents/R course/Class Exercises/Class Exercise 2/"

#-----------------------------------------------------------------------------------------
# Loops to get the data into a list, and clean it up.
#-----------------------------------------------------------------------------------------

# (1) Make an empty list called 'lidar'. We'll deposit data into it next.
lidar <- list()

# 'lidar' is going to be a list of lists of matrices. The first list in 'lidar' will
# contain all four transects from Survey 1 as matrices; the second list inside of the
# 'lidar' list will contain all four transects from Survey 2 as matrices.

# (2) Make a nested for loop that looks into each survey folder, and calls in each
# transect file, converting it to a matrix, and depositing the matrix in the appropriate
# list in 'lidar'.
# - Don't bother naming the sub-lists yet, just identify them by numbers.
# - TIP: Before entering the inner loop, you'll have to make the 'ith' element in 'lidar'
# an empty list as well.
for (s in 1:2) {
  lidar [[s]] <- list()
  for (t in 1:4) {
    lidar [[s]] [[t]] <- as.matrix (
      read.csv (paste0 (dat.dir, "Lidar_Survey ", s, "/lidar_s", s, "_t", t, ".csv"))
    )
  }
}

# (3) Now, copy the above for-loop below, and modify it so instead of identifying each new
# list element by number, allow them to be dynamically named "survey.1", "survey.2", and
# for the transects, "lidar.s1.t1" through "lidar.s2.t4".
# FIRST: make 'lidar' an empty list again. Otherwise this next loop will just add surveys.
lidar <- list()

for (s in 1:2) {
  lidar [[ paste0 ("survey.", s) ]] <- list()
  for (t in 1:4) {
    lidar [[s]] [[ paste0 ("lidar.s", s, ".t", t) ]] <- as.matrix (
      read.csv (paste0 (dat.dir, "Lidar_Survey ", s, "/lidar_s", s, "_t", t, ".csv"))
    )
  }
}


# (4) Make another nested loop to go through each matrix and convert -9999 values to NA.
# Note that this could have been added into the loop above.

for (s in 1:2) {
  for (t in 1:4) {
    lidar [[s]] [[t]] [lidar [[s]] [[t]] == -9999] = NA
  }
}

#------------------------------------------------------------------------------------------
# Loop to plot the data with image.plot() and export plots to file.
#------------------------------------------------------------------------------------------

# (5) Now, loop through to make a heat map of leaf area density from each lidar matrix
# using the image.plot() function from the 'fields' package.
# - Note that each matrix must be transposed by t() when it's plotted, or the forest will
# end up on its side.
# - Have each plot exported automatically to a ".png" image file in the correct results
# folder corresponding to each survey. Make the width=1000 and height=600. See the png()
# example below.
# - Make the filename of each plot reflect the survey and transect numbers.
# - Make the title at the top of each plot reflect survey and transect numbers. Use the
# 'main' argument in image.plot() the same way you would with base-R's plot().
# - TIP: It can be less confusing if you make dynamic filenames and titles as separate
# objects in the loop, and drop the objects into filname= and main= arguments.

for (s in 1:2) {
  for (t in 1:4) {
    #--Paste together a plot title indicating the survey and transect numbers.
    plot.title="***paste a title together here***"
    #--Paste together a directory path and plot image file name for export.
    #***
    #--Open a graphics device
    #***
    #--Make the plot of leaf area density by height and length on the transect.
    image.plot (t (lidar [[s]] [[t]]), main=plot.title)
    #--Close the graphics device
    #***
  }
}

# >> Using a graphics device to export plots:
# To export a plot directly to an image file, use one of the "graphics device" functions,
# such as png(), jpeg(), etc., as follows:
```

```
png (filename="directory and filename", width=1000, height=600)
# Put your plotting code here, i.e. the code that would normally make a plot in your
# RStudio plotting window.
# Now close the graphics device with dev.off(), no arguments.
dev.off()


#-----------------------------------------------------------------------------------------
# Loop to add summary stats data frames to each survey sub-list
#-----------------------------------------------------------------------------------------

# (6) Stats table:
# - We want a summary table that contains statistics for each height (1m - 60m) from each
# transect matrix.
# - In each sub-list, create a stats data frame with one column for height. Name the data
# frames 's1.stats' and 's2.stats'.
# - Use an inner loop to get the stats from each matrix into the stats data frame.
# -- For each matrix, there should be three stats columns for the sums, means, and
# maximums for each row (height): e.g., 't1.max.lad', 't1.sum.lad', 't1.mean.lad'. So
# you'll also get stats for t2, t3, and t4.
# -- You can even create the named columns using this loop by df[[paste0(t,"stat.col")]].
# -- For sums and means, look up rowSums(). These are vectorized, so they'll give you all
# the row sums from a matrix all at once.
# -- For max, well, there's no vectorized function for that! So, you know what to do...
# -- IMPORTANT NOTE: Meter 1 is in row 1 of each lidar matrix, so height matches row num.
```