



EXERCISE 2

Running a random forest model in R

Introduction

Running random forests model in RStudio is a fairly simple and straightforward process. It involves specifying a few variables (such as path to the training data), training the model, evaluating the accuracy or strength of the model, and then applying the model to the entire study area to create a map product. Most of these steps are built into the script, meaning that after setting the variables, all that's left to do is run the model.

Objectives

- Learn how to run a simple random forest analysis in RStudio using a pre-made script

Required Data

- **train_subset.shp** – shapefile of training data for the subset study area
- **NAIP_subset.tif** - layer stack of NAIP imagery with optical bands
- **NAIP_ndvi_subset.tif** – layer stack of Landsat imagery and NDVI derivative for the study area
- **RF_pixel.R** – R script for running pixel-level random forest classification model on the subset study area

Prerequisites

- **You have installed RStudio on computer**
- **You have installed ESRI ArcGIS on computer** and have basic understanding of how to use the software





Table of Contents

Part 1: Set up RStudio	3
Part 2: Review the script	4
Part 3: Running a random forests classification	5
Part 4: View the outputs	5
Part 5: Iterating the classification	11



Part 1: Set up RStudio

A. Open RStudio

1. If you do not have **RStudio** open, open it now.
 - i. Choose the **Start Menu, All Programs, RStudio**, and **RStudio** again.
2. Open Script
 - i. If your script doesn't open as a saved tab, choose **File, Open File**, and navigate to the **Scripts** folder in your course directory. Select the **RF_pixel.R** file and select **Open**.

B. Adjust the variable paths

1. Out of the seven customizable variables within the script, the following four variables need to be edited.
 - i. Go to **line 16** of the script where it says

`workspace = ""`

- (a) Copy location of the **output folder** from a Windows Explorer window (e.g., C:\RF\Data\Output) and paste it inside the quotations.
- (b) Change the slashes from backslashes (\) to forward slashes (/).

- ii. Go to **line 19** of the script where it says

`pointsshp = ""`

- (a) Copy the location of the **training data** from a Windows Explorer window (e.g., C:\RF\Data\Shapefile\train_subset.shp), and paste it inside the quotations.
- (b) Change the slashes from backslashes (\) to forward slashes (/).

- iii. Leave the next two variables, **classfield** and **predicttype** (on lines 22 and 25, respectively) as they are. These define the name of the attribute we want to predict ("Class") and the type of output we want to create (Thematic as opposed to Continuous).

- iv. Go to **line 28** of the script where it says

`imageList = c("")`

Note: in this line of code, the **c()** is used to create a vector. In R, a vector is a sequence of data elements of the same basic type. Here, our vector contains strings, each of which points to a continuous raster used as a predictor variable. In this example, we have one layer stack of NAIP imagery, however, if we had additional continuous rasters, we would separate the strings with commas (e.g., `c("path1", "path2", "path3")`)

- (a) Copy the location of the **NAIP imagery** from a Windows Explorer window (e.g., C:\RF\Data\Imagery\naip_subset.tif), and paste it inside the quotations.
- (b) Change the slashes from backslashes (\) to forward slashes (/).
- v. Leave the next variable, **thematicimagelist** (on **line 31**), as it is. This variable is used to specify any thematic rasters that may be available (e.g., a soils raster) for use as predictor variables.

vi. Go to **line 34** of the script where it says

`OutputModel = ""`

- (a) Copy the **desired name of the output file**. This should be in your Output folder (e.g., C:\RF\Data\Output\predict_NAIP_subset.tif), and paste it inside the quotations.

Part 2: Review the script

A. Read through the script

Note: You do not have to be skilled at scripting to understand what the script is doing. While helpful for deeper understanding (and for troubleshooting if or when you decide to alter the script), the text that has been commented out explains what is happening in each step. You will notice that in addition to comments peppered throughout the script, the script is also chunked into major sections with commented headings.

1. Go to **line 37**. This first section of code will install necessary packages (i.e., utilities that are not native to R) and setup the workspace. After you run through the script once, you can comment out the four installations (**lines 38-41**) for future runs, because they have already been installed.
2. Go to **line 59**. This second section of code creates a variable called **stacklist**, which stacks all continuous and thematic rasters together. It will also plot the imagery within the GUI window as a color infrared composite, so you do not have to open ArcMap to view it.

Note: In order for the script to run without error, all of your rasters need to have identical spatial information, meaning they require the same number of rows/columns, resolution, extent, and projection.

3. Go to **line 100**. This section extracts raster values from each layer in **stacklist** for every point location in **pointssh** and writes those values as a data frame.
4. Go to **line 120**. This section creates a variable, **classindex**, which pulls in the attribute value, **classfield**, which was specified in Part 1. This variable is extracting the attributes from the **Class** field (i.e., Ground, Shadow, Tree, and Shrub) in our shapefile.
5. Go to **line 131**. This section converts the thematic classes present in the training data (shapefile) to numeric categories.
6. Go to **line 162**. This section (and the next, starting at **line 188**) extracts data from the raster stack and saves them into a dataframe that is compatible with the modeling process.
7. Go to **line 251**. This section creates the random forest model – this is essentially the ruleset which will be used to classify each pixel from the input imagery. The model is set to run 1000 trees and omit no-data pixels from model-creation.
8. Go to **line 264**. This section creates some useful, non-image outputs to evaluate the strength of the model. It includes a CSV file containing the legend for the output land-cover map, a confusion matrix, and the variable importance (VI) plot. The legend file defines the numeric classes and the associated land-cover type that were mapped. The VI plot is used to assess the frequency with which each variable was used in the classification process (i.e., at the decision nodes). It summarizes how much accuracy would be sacrificed (on average) if a

variable was dropped from the analysis. The confusion matrix is written as a text file. It describes the class that was assigned to the pixel through the modeling process as opposed to the class assigned in the training data. The percent agreement and disagreement between the classes is used to assess the strength and accuracy of the mapped product.

9. Go to **line 301**. The final section writes the predicted raster and prints information regarding the script status to the console. It also plots the predicted raster in the GUI window, so you can get a sense of how well the model performed before moving to ArcMap.

Part 3: Running a random forests classification

A. Run the script

1. In the script editor window, press the **Ctrl key** and **A** on the keyboard to select all of the text. Alternatively, you can right-click within the scripting window and select **Select All**.
2. Press the **Ctrl key** and the **Enter key** to run the script.

Note: You can run individual lines by placing the cursor at the line that you want to run and pressing **Ctrl** and **Enter**. You can also run a select portion of the script instead of running the entire file – to do this, simply select the portion of the script that you want to run and press **Ctrl** and **Enter**.

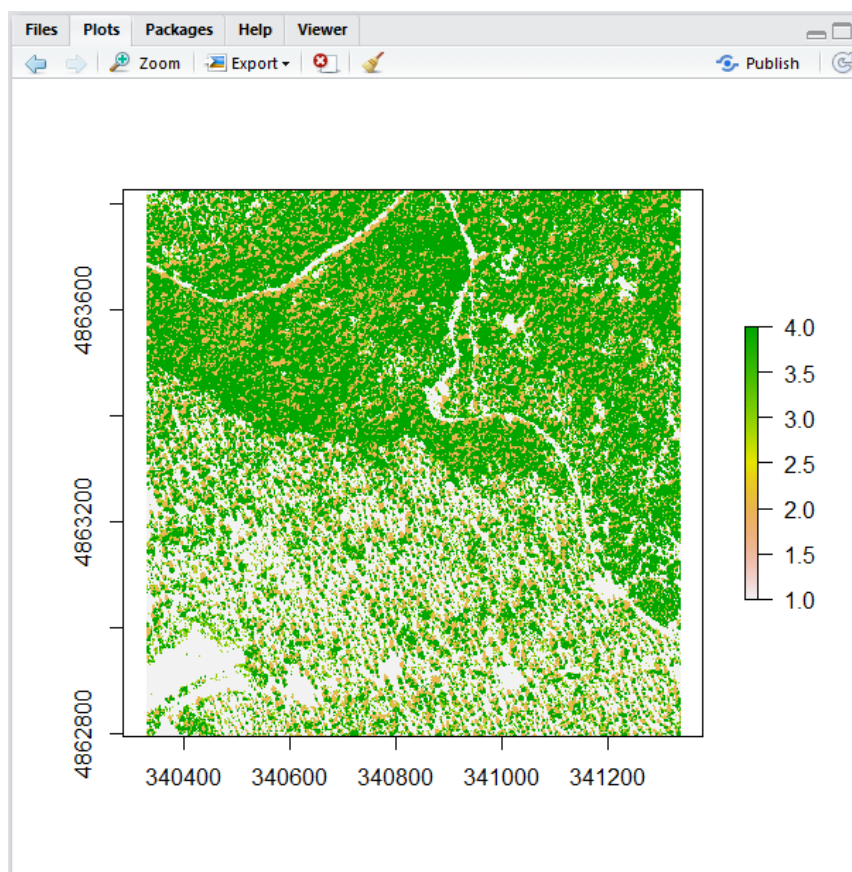
3. Monitor the progress of the process in the **console**. Once the output raster starts being written, you should see a progress bar (created with equals signs) at the bottom of the console.

Note: Scrolling through the console, you should see the comments and lines of code printed in blue, the actual work and/or outputs printed in black, and any errors or exceptions printed in red.

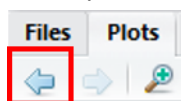
Part 4: View the outputs

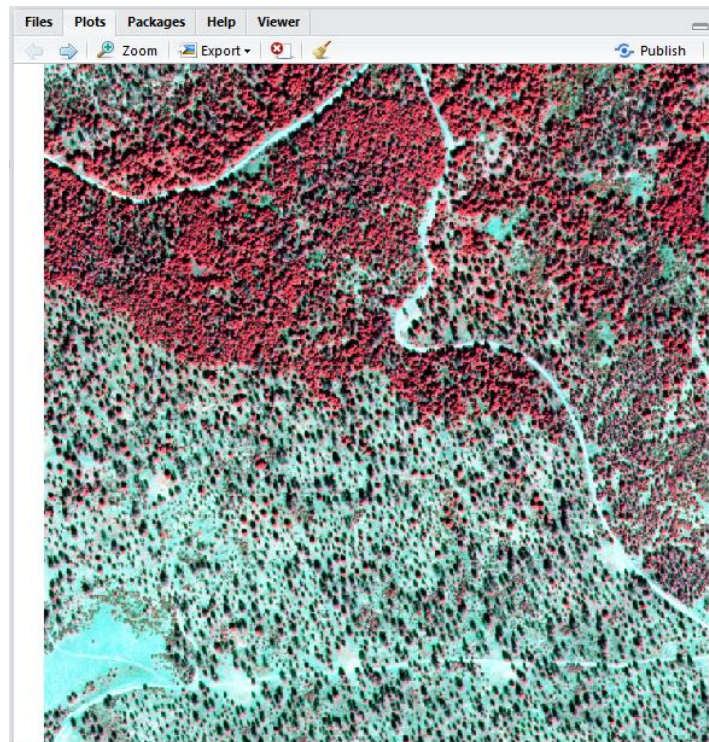
A. View the R plots

1. In the GUI window, select the Plots tab. You should see simple representation of the output classification, where the X and Y axes represent the longitude and latitude coordinates, respectively, and each pixel is plotted according to its geographic coordinates.



2. Click the back button in the Plots tab (shown below), and you can view the previously plotted color infrared RGB composite of the study area.









Note: If you have the Plots tabs active before running your script, the plots will populate in real time.

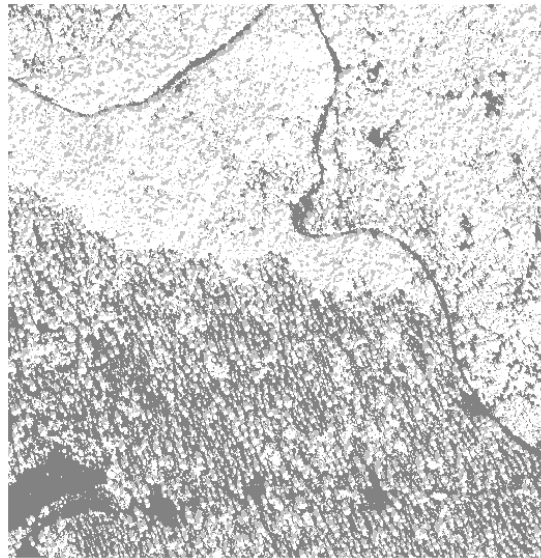
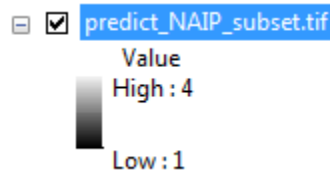
B. Examine the Output folder

1. Open a Windows explorer window (**Windows key + E**) and navigate to your **course directory**.
 - i. In the **Data** folder, open the **Output** subfolder.
 - ii. You should see four files: the output TIFF file, a confusion matrix (text file), a CSV file containing the legend for the output, and a variable importance plot (JPEG).

Name	Date modified	Type	Size
 predict_NAIP_subset.tif	6/3/2016 11:41 AM	TIFF image	180 KB
 predict_NAIP_subset_confusion_matrix.txt	6/3/2016 11:41 AM	Text Document	1 KB
 predict_NAIP_subset_legend.csv	6/3/2016 11:41 AM	Microsoft Excel C...	1 KB
 predict_NAIP_subset_VarImpPlot.jpg	6/3/2016 11:41 AM	JPEG image	4 KB

C. Display raster in ArcMap

1. Open **ArcMap**, and drag and drop the **TIFF** from the Windows explorer window into the viewer. You will see the classification displayed with a continuous, greyscale symbology as shown in the graphics below.



2. Edit the symbology to include more intuitive colors and labels.
 - i. Double-click on **predict_NAIP_subset.tif** in the **Table of Contents**.
 - ii. Select the **Symbology** tab, and on the left under **Show**: click on **Unique Values**. You will be prompted to create an attribute table – select **Yes**.
 - iii. With just numerical identifiers, it's hard to tell which class is which. Go back to the Outputs folder, and double-click on **predict_NAIP_subset_legend.csv** – it should open in Excel, shown below.

NumValue	TextValue
1	Ground
2	Shadow
3	Shrub
4	Tree

- iv. Back in the **Layer Properties** window in ArcMap, select the cell that reads **1** from the **Label** column, and replace this text with the corresponding label from the CSV file, **Ground**, and then press the **Enter** key. Repeat the process for each class.
- v. Apply more intuitive colors for each class. In the **Symbol** column, double click on the **colored rectangles** to choose new colors. When finished, click **OK**. Use the following colors:
 - (a) Ground: light gray
 - (b) Shadow: black
 - (c) Shrub: brown
 - (d) Tree: green



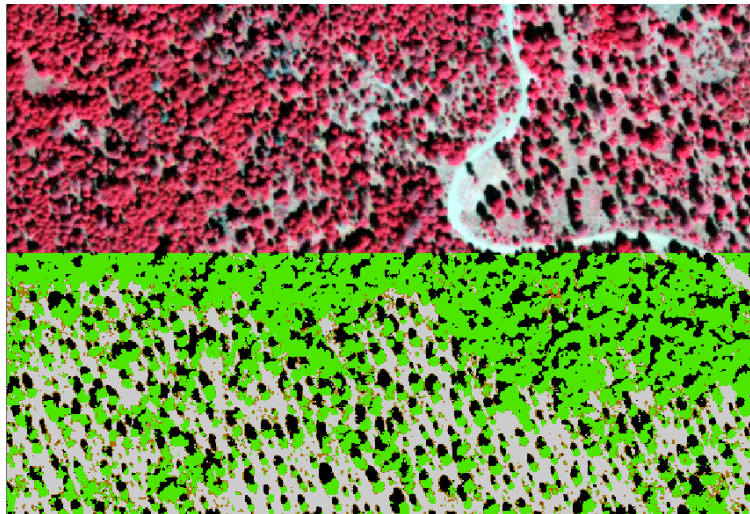
3. In your **Windows explorer** window, navigate to the **Imagery folder** and drag and drop the imagery into the ArcMap viewer. Adjust the composite if you so desire.
4. Swipe between the output raster and the input imagery.
 - i. In the grey space at the top of the ArcMap window, right-click and ensure that there is a checkmark next to the Effects option – this will load the Effects toolbar. If there is no checkmark next to it, select Effects to load the toolbar.



- ii. Click on the Swipe tool from the Effects toolbar, and from the dropdown that includes the available layers, make sure that the layer found at the top of your map composition is the one that is selected.

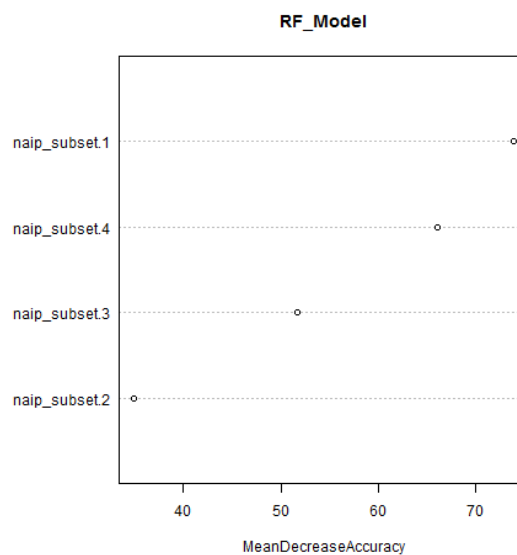


- iii. In your map window, click and drag the cursor to transition back and forth from imagery to classification. Zoom in and pan around to investigate how well the random forest model performed.



D. View the variable importance plot

1. Navigate back to the **Output** folder in your Windows Explorer window.
2. Double-click on the file **predict_NAIP_subset_VarImpPlot.jpg**, shown below.



Note: This plot shows which variables contributed the most in developing the predictive ruleset. The individual predictor variables are on the y-axis, while the x-axis represents the mean decrease in accuracy. The current plot shows band 1 (red) and band 4 (near-infrared) from the image stack were most important in the current classification. Observe how much accuracy might change based on this VI plot if you were to remove any given variable from the analysis.

E. View the confusion matrix

1. Double-click on the file **predict_NAIP_subset_confusion_matrix.txt**

```
"type" "errorRate"
"1" "classification" 0.207801279252403
"confusion.1" "confusion.2" "confusion.3" "confusion.4" "confusion.class.error"
"1" 47 3 5 0 0.145454545454546
"2" 3 29 0 3 0.171428571428571
"3" 8 1 5 9 0.782608695652174
"4" 2 5 2 77 0.104651162790698
```

Note: This is a text file, so by default it will open in a text editor. As you can see, it is very hard to read or work with in this format! We need to open it in Excel in order to organize the file properly.

2. Open the file in Excel.
 - i. Open **Excel** by selecting the **Start Menu, All Programs, Microsoft Office 2013**, and then **Excel 2013**.
 - ii. Select **File, Open**, and then double-click on **Computer** to navigate to your output folder.
 - iii. From the drop-down menu next to the File name field, select **Text Files (*.prn, *.txt, *.csv)** and then choose **predict_NAIP_subset_confusion_matrix.txt**
 - iv. Click **Open**.
3. You will need to run through the **Text Import Wizard** in order to load the file in Excel.
 - i. Make sure that the radio button next to **Delimited** is selected and choose **Next**.
 - ii. Check the box next to **Space** under **Delimiters**. You should see the data preview portion of the window snap your data into columns. Click **Next** and then **Finish**.

Note: Though we won't be doing it in this course, we can now easily read the confusion matrix and use it to calculate error metrics for our map. The error rate for the classification is also appended to this file, making it easy to get a quick sense of model performance. Since random forests make use of bagging, the model has a built-in dataset for validation. From this text file, we can see that our classification has an out-of-bag (OOB) error of ~20%.

Part 5: Iterating the classification

With scripts, it's easy to make small tweaks, changing variables or parameters, to iterate the process. In this next part, we will rerun the classification to run on a layer stack that includes the NDVI layer to see how it affects the output.

A. Adjust the script and run again

1. Go back to **RStudio** and press **Ctrl** and **L** – this will clear the console.
2. Scroll to the top of the code, and adjust the variable **imageList** at **line 28** such that it points to **NAIP_ndvi_subset.tif**
3. On **line 34**, change the variable **OutputModel** to write a file called **predict_NAIP_ndvi_subset.tif**
4. Comment out **lines 38-41** by placing a pound sign/hashtag (**#**) at the front of each. There is no need to install these packages, because you installed them the first time you ran the script.
5. Run the script by pressing **Ctrl + A** and then **Ctrl + Enter**.

B. Compare NDVI and no-NDVI outputs



1. Navigate to your **Output** folder, and drag and drop the new TIFF file, **predict_NAIP_ndvi_subset.tif**, into the ArcMap viewer.
2. Adjust the display of **predict_NAIP_ndvi_subset.tif** so that it matches **predict_NAIP_subset.tif**
3. In the **Effects** toolbar, select **predict_NAIP_subset.tif** from the dropdown menu.
4. Use the **Swipe** tool to compare outputs.

Note: You should see that most of the differences are subtle and manifest as changes between the shrub and tree classes. The shadow and ground classes remain almost entirely unchanged. Why do you think this is?

Congratulations! You have successfully completed this exercise. You now know how to run a random forest model in RStudio.

