

Lidar-derived Individual Tree Inventory using ArcGIS Pro

Summary: These instructions demonstrate how to generate an individual tree inventory for trees in the dominant/co-dominant crown position. The instructions use 2018 lidar data from the PA Sullivan County Quality Level 1 (QL1) in the Loyalsock State Forest.

System Requirements: ArcGIS Pro 2.4.1, R, R Studio

Contact: PA Department of Conservation & Natural Resources, Bureau of Forestry

References: <https://github.com/R-ArcGIS/r-bridge-install>

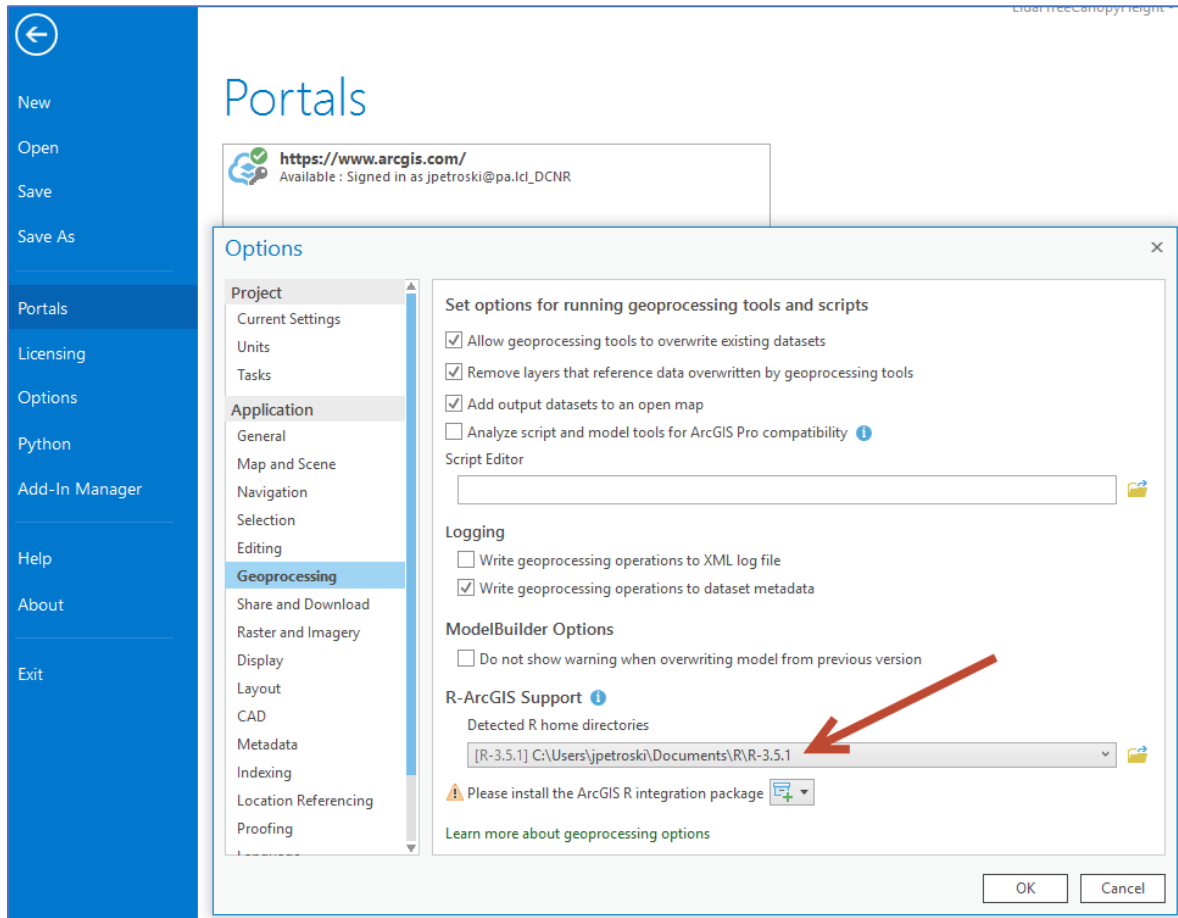
<https://cran.r-project.org/web/packages/ForestTools/vignettes/treetopAnalysis.html>

Disclaimer: Users are advised to use these instructions strictly at their own risk. No warranties are made about the reliability or security of these instructions. No parties shall be held liable for any losses or damages of any kind in connection with the use of these instructions.

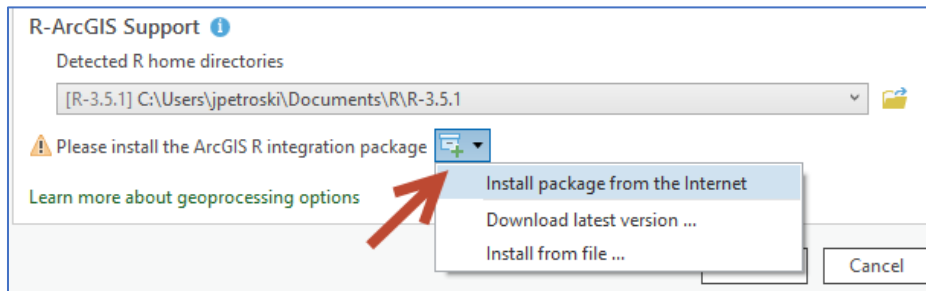
1. First, generate a high-resolution tree canopy height raster. There are separate instructions on the PA Lidar Working Group website (Vegetation Page) for how to generate this. QL1 lidar or the highest resolution available is preferred for this analysis. The resolution of the tree canopy height raster should ideally be 1-3 foot. Leaf-on lidar would also be preferred as it would likely produce higher quality results than leaf-off lidar. Below is example of a high resolution (1-meter) tree canopy height raster:



2. Create the bridge between R and ArcGIS Pro. In ArcGIS Pro, go to the **Projects Menu → Options → Geoprocessing → R-ArcGIS Support**. Enter the file path where you have R installed on your machine.



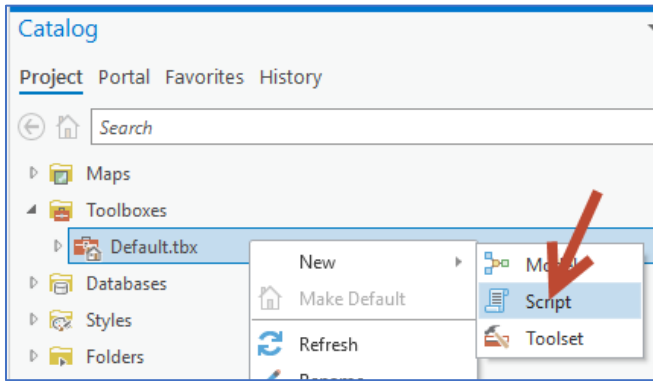
3. If you have not previously installed the **ArcGIS R integration package**, wait for the icon to appear and do so.



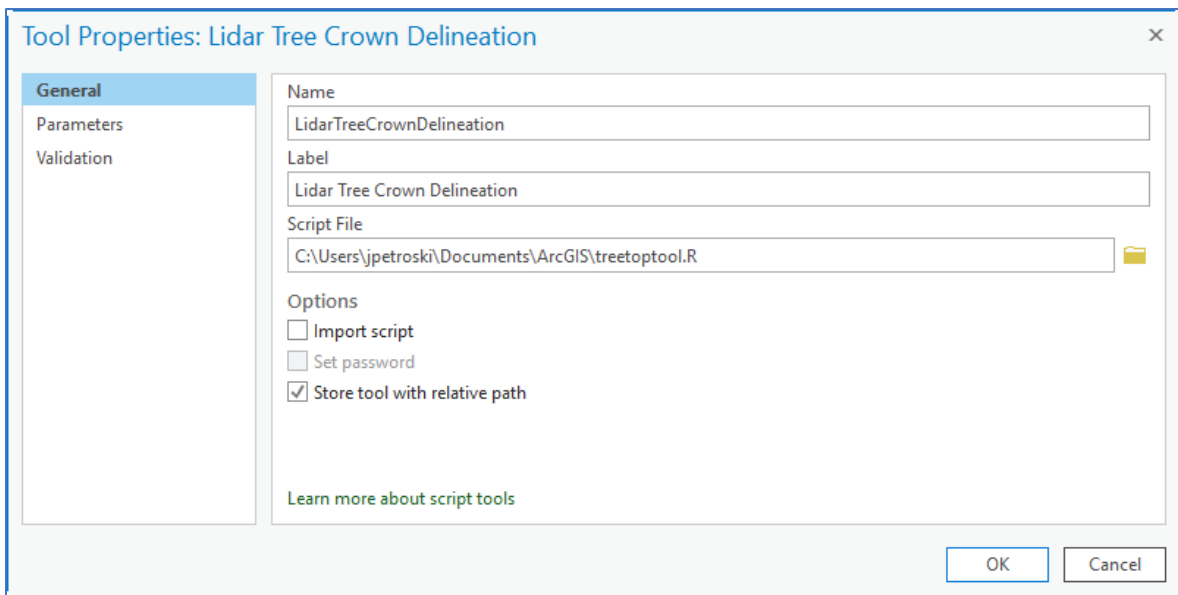
4. The source R code (see <https://cran.r-project.org/web/packages/ForestTools/vignettes/treetopAnalysis.html>) implements the *variable window filter* (vwf) algorithm, which creates a moving window that scans the canopy heights, and if a given cell is found to be the highest within the window, its tagged as a treetop. The size of this window should be adjusted for your specific site and crown sizes of the trees at that site. Oak trees typically have larger crowns than northern hardwood species, and mature trees have larger crowns than immature trees. The default source code uses a function to define the size of the search window, where x represents the height of the canopy above the ground at that location. Modify the lin variable to fit the trees at your specific site.

```
lin <- function(x){x * .05 + 3.5}
```

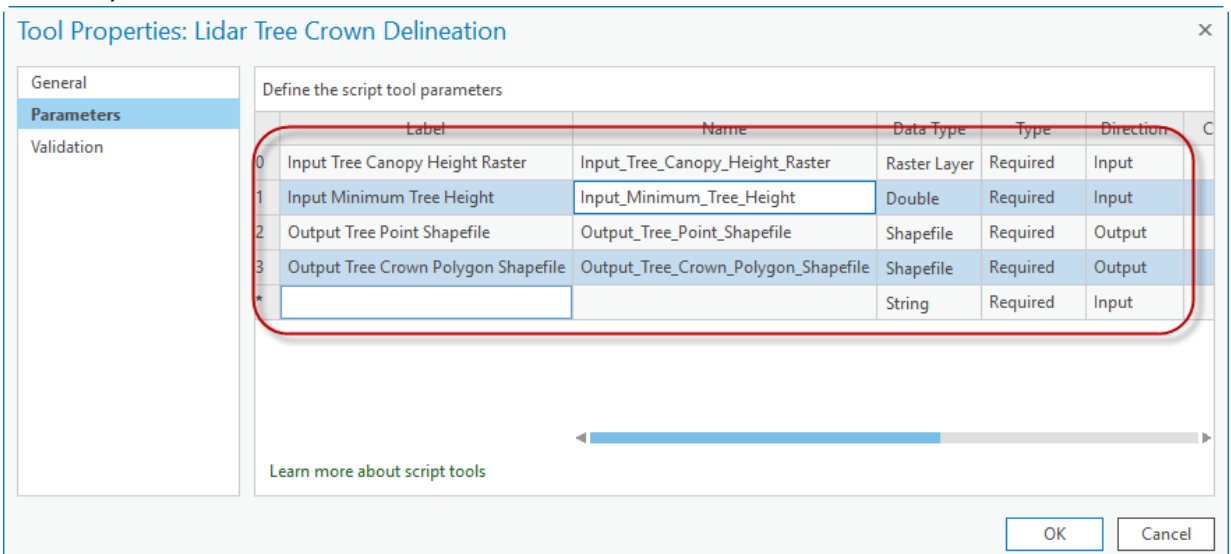
- In the **Catalog Pane**, right-click on a custom toolbox and choose **New → Script**.



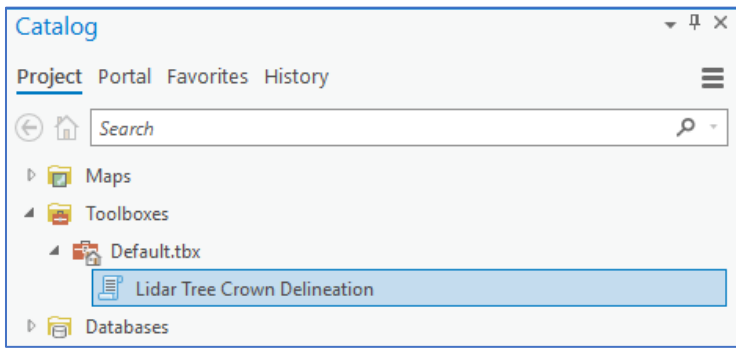
- In the **GENERAL** section, give the new tool a **name**, **label**, and specify the path to the R file that contains the code (see <https://cran.r-project.org/web/packages/ForestTools/vignettes/treetopAnalysis.html>). The source R code is also included at the end of this document.



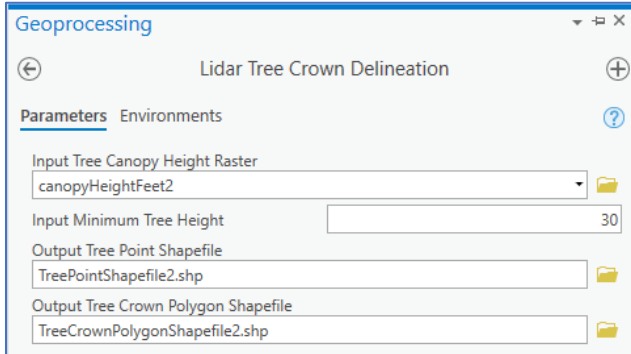
- In the **PARAMETERS** section, define the input and output parameters of the tool. This script has a total of 4 parameters that you must define in the blank table. The table entries should be entered like this:



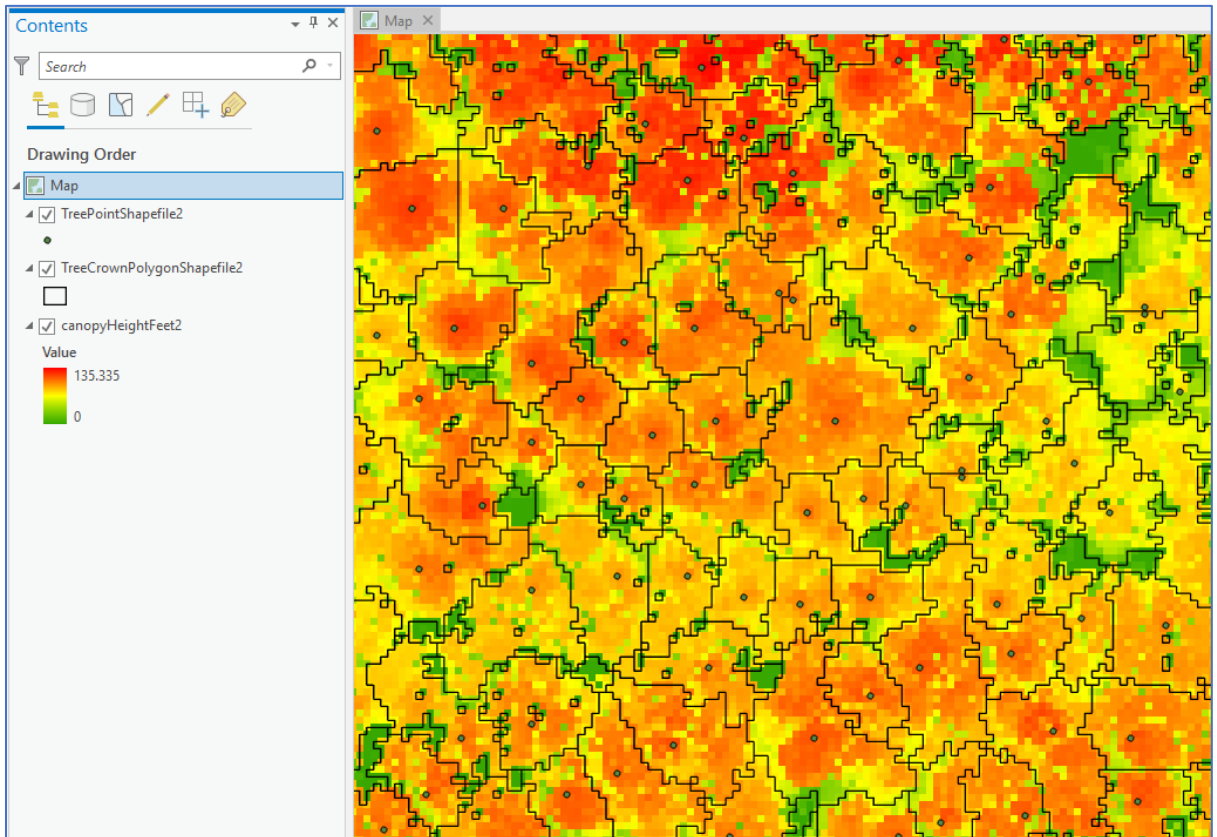
8. Your new geoprocessing tool should look like this:



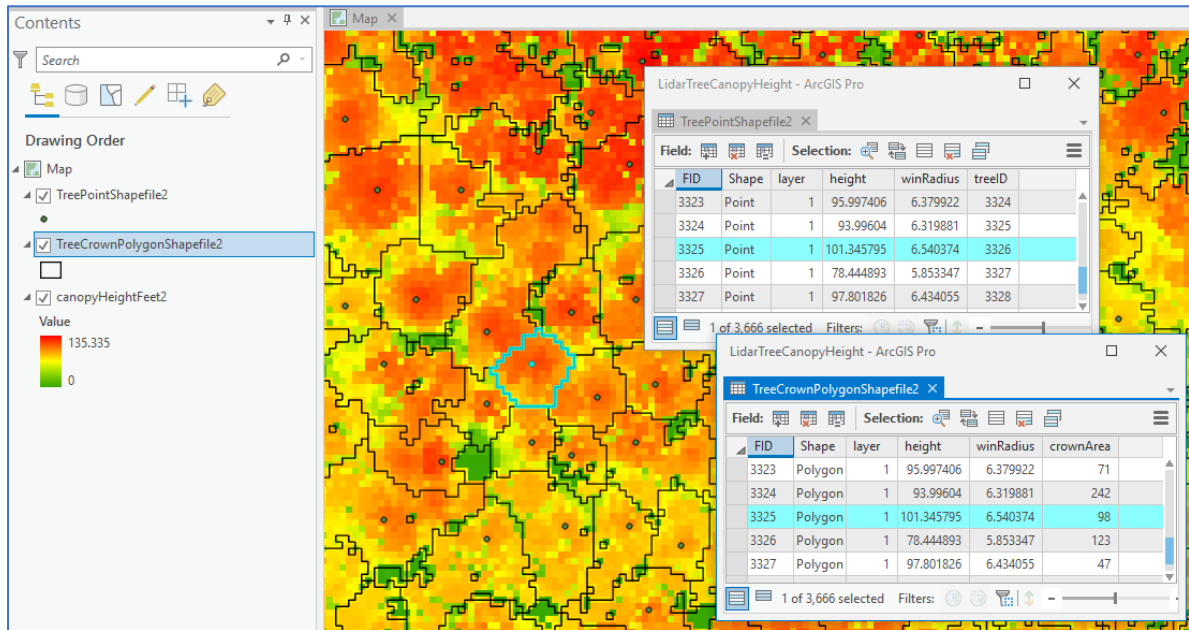
9. Open the new tool, specify values for the parameters, and run it.



10. When the tool is finished, the output will look something like this:



The attribute tables of the output shapefile will also contain the height of the trees and radius of the tree crowns:



Source R Code (<https://cran.r-project.org/web/packages/ForestTools/vignettes/treetopAnalysis.html>)

```

tool_exec <- function(in_params, out_params) {
  arc.progress_label("Loading packages...")
  arc.progress_pos(20)
  install.packages("ForestTools")
  library(ForestTools)
  library(raster)
  arc.progress_label("Reading Data...")
  arc.progress_pos(40)
  #Tree Tops
  treetop <- out_params[[1]]
  input_raster <- in_params[[1]]
  canopy <- arc.data2sp(arc.raster(arc.open(input_raster)))
  canopyht <- raster(canopy)
  arc.progress_label("Generating Tree Tops...")
  arc.progress_pos(60)
  lin <- function(x){x * .05 + 3.5}
  minht <- in_params[[2]]
  ttops <- vwf(CHM = canopyht, minHeight = minht, minWinNeib = "rook", winFun = lin)
  arc.write(treetop, ttops)
  arc.progress_label("Generating Crown Polygons...")
  arc.progress_pos(80)
  #Crowns
  crownpoly <- out_params[[2]]
  crowns <- mcws(treetops = ttops, format = "polygons", CHM = canopyht, minHeight = 15, verbose = FALSE)
  arc.write(crownpoly, crowns)
  return(out_params)
}

```