## Lidar-derived Individual Tree Inventory using ArcGIS Pro

<u>Summary</u>: These instructions demonstrate how to generate an individual tree inventory for trees in the dominant/codominant 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

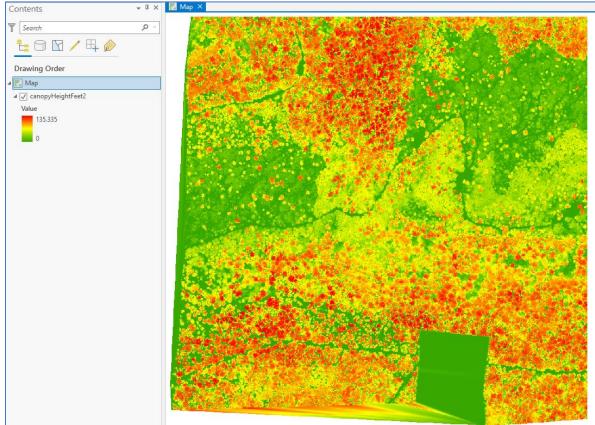
Contact: PA Department of Conservation & Natural Resources, Bureau of Forestry

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

https://cran.r-project.org/web/packages/ForestTools/vignettes/treetop\_analysis.html

<u>Disclaimer</u>: 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.

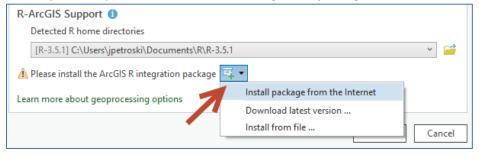
 First, generate a high-resolution tree canopy height raster. There are separate instructions on the PA Lidar Working Group website (Vegetation Page) for how to generate this. QL1 lidar or the highest resolution available is preferred for this analysis. The resolution of the tree canopy height raster 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:



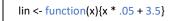
2. Create the bridge between R and ArcGIS Pro. In ArcGIS Pro, go to the **Projects Menu**  $\rightarrow$  **Options**  $\rightarrow$  **Geoprocessing**  $\rightarrow$  **R-ArcGIS Support**. Enter the file path where you have R installed on your machine.

E		
	Portals	
New	POILAIS	
Open	C https://www.arco	ais.com/
Save	Available : Signed in	gis.com/ a as jpetroski@pa.lcl_DCNR
Save As	Options	x
	Project	A
Portals	Current Settings	Set options for running geoprocessing tools and scripts
Licensing	Units	✓ Allow geoprocessing tools to overwrite existing datasets
	Tasks	Remove layers that reference data overwritten by geoprocessing tools
Options	Application	✓ Add output datasets to an open map
Python	General	Analyze script and model tools for ArcGIS Pro compatibility ()
	Map and Scene	Script Editor
Add-In Manager	Navigation	
	Selection	Logging
Help	Editing Geoprocessing	Write geoprocessing operations to XML log file
About	Share and Download	✓ Write geoprocessing operations to dataset metadata
About	Raster and Imagery	ModelBuilder Options
Exit	Display	Do not show warning when overwriting model from previous version
Exit	Layout	R-ArcGIS Support
	CAD	Detected R home directories
	Metadata	[R-3.5.1] C:\Users\jpetroski\Documents\R\R-3.5.1
	Indexing	⚠ Please install the ArcGIS R integration package 📴 🔻
	Location Referencing Proofing	
	Line Line Line Line Line Line Line Line	Earn more about geoprocessing options
		OK Cancel

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 <a href="https://cran.r-project.org/web/packages/ForestTools/vignettes/treetop\_analysis.html">https://cran.r-project.org/web/packages/ForestTools/vignettes/treetop\_analysis.html</a>) 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.



5. In the **Catalog Pane**, right-click on a custom toolbox and choose **New**  $\rightarrow$  **Script**.

Catalog					-
Project Portal Favorites History					
🕞 🏠 Search					
🖻 📷 Maps					
🔺 🗃 Toolboxes					
Default.tbx		New	+	5.0	Martin -
Databases	合	Make Default		F	
🖻 😿 Styles		IVIARE DETAUL		周	Script
Folders	3	Refresh		É.	Toolset
	1	Demonstra			

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

Tool Properties: Lidar	Tree Crown Delineation ×
General	Name
Parameters	LidarTreeCrownDelineation
Validation	Label
	Lidar Tree Crown Delineation
	Script File
	C:\Users\jpetroski\Documents\ArcGlS\treetoptool.R
	Options
	Import script
	Set password
	✓ Store tool with relative path
	Learn more about script tools
	OK Cancel

7. 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:

arameters Label Name Data Type Type D   alidation Input Tree Canopy Height Raster Input_Tree_Canopy_Height_Raster Raster Layer Required Input   1 Input Minimum Tree Height Input_Minimum_Tree_Height Double Required Inp   2 Output Tree Point Shapefile Output_Tree_Point_Shapefile Shapefile Required Output   3 Output Tree Crown Polygon Shapefile Output_Tree_Crown_Polygon_Shapefile Shapefile Required Output
1   Input Tree Canopy Height Raster   Input_Tree_Canopy_Height_Raster   Raster Layer   Required   Input     1   Input Minimum Tree Height   Input_Minimum_Tree_Height   Double   Required   Input     2   Output Tree Point Shapefile   Output_Tree_Point_Shapefile   Shapefile   Required   Output
2 Output Tree Point Shapefile Output_Tree_Point_Shapefile Shapefile Required Output_Tree_Point_Shapefile Required Output_Tree_Point_Shapefile Required Output_Tree_Point_Shapefile Shapefile Required Output_Tree_Point_Shapefile Shapefile Required Output_Tree_Point_Shapefile Shapefile Required Output_Tree_Point_Shapefile Shapefile Shapef
2 Output Tree Crown Polygon Shanefile, Output Tree Crown Polygon Shanefile, Shanefile, Required, Ou
output nee clown polygon shapeme output_mee_clown_polygon_shapeme shapeme hequired output
* String Required Ing

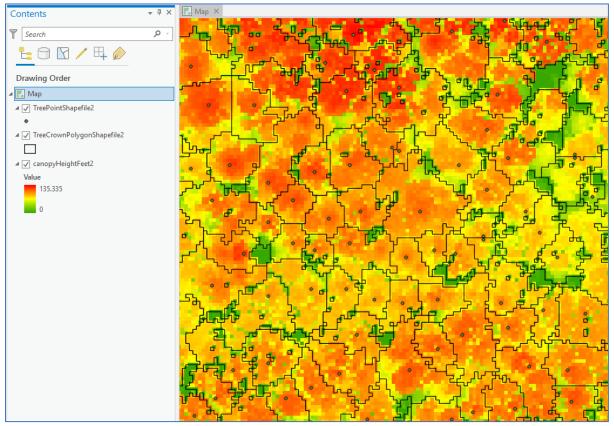
## 8. Your new geoprocessing tool should look like this:

Catalog	₩ 4 ×
Project Portal Favorites History	≡
	<u>-</u> م
D 🗑 Maps	
🔺 🗃 Toolboxes	
🔺 🙀 Default.tbx	
E Lidar Tree Crown Delineation	
▷ 🥫 Databases	

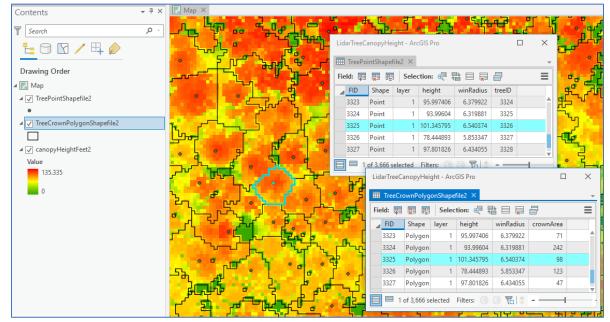
9. Open the new tool, specify values for the parameters, and run it.

Geoprocessing		* + ×
	Lidar Tree Crown Delineation	$\oplus$
Parameters Environm	ents	?
Input Tree Canopy Heig	ght Raster	
canopyHeightFeet2		
Input Minimum Tree H	eight	30
Output Tree Point Shap	efile	
TreePointShapefile2.sh	p	🕋
Output Tree Crown Pol	ygon Shapefile	
TreeCrownPolygonSha	pefile2.shp	<b>~</b>

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/treetop\_analysis.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)
}