GIS Tips & Tricks

Matthew Bauld and Al Karlin, Ph.D. CMS-L, GISP

Order Does Not Matter for LAStools

For most of us with any programming experience, especially us old FORTRAN programmers, we know that programming languages are carefully structured and the order of instructions in a program really matters. Getting things out of order may produce unexpected results. Rules, like using parenthesis around operators, as in converting angular coordinates into their decimal equivalents:

Latitude = ((Degrees)+(Minutes/60)+(Seconds/3600))

to ensure that operations are performed and grouped in the expected order are typically observed. Arguments for instructions are expected to follow a certain order otherwise the instruction may error (or go into that infinite loop). The same is generally true for Python scripting.

Here at Dewberry, we make use of a several commercial, off the shelf software tools when processing lidar data, but we usually customize them to our specific needs. The customization usually entails calling them from custom scripts, most of which are written in the Python programming language. Because LAStools (rapidlasso GmbH:: https://rapidlasso.com/lastools/) contains numerous independent executable files, LAStools presents Dewberry the ideal opportunity to construct custom lidar processing Python scripts.

We have noticed that LAStools uses a syntax structure for Python scripting that is less demanding than most that we have previously encountered. Unlike Python interpreters, such as Arcpy (packaged with the Esri ArcGIS/ArcGISPro suite) which expects arguments to be given in a set order and separated by commas, LAStools accepts arguments in essentially any order and separated by spaces. While this feature of LAStools is well documented on the University of North Carolina LAStools page (https://www.cs.unc.edu/~isenburg/lastools/) it is a bit difficult to find. So, below is an example of a LAStools command from a recent project that reclasses points with return number 0 from any ASPRS point class to ASPRS Class 42.

lastoolspath\las2las -i *.las -keep_return 0 -filtered_transform -set_extended_classification 42 -odix _reclass -olas -cores 8

Without previous knowledge, the code above doesn't make a ton of sense. This statement specifies the following:

Las2las -i *.las: call the las2las executable for each input files with the .las extension and do the following:

- keep_return 0: Filters point records with a return number of 0
- filtered_transform: Sets the filtered point records as the points to be manipulated
- set_extended_classification 42: Reclassifies the filtered points to ASPRS Class 42
- odix_reclass: Sets the output file name as the input name plus "_reclass"
- olas: Output file as LAS
- cores 8: use 8 computer cores for processing

The above code could also be input and successfully run as follows:

lastoolspath\las2las -i *.las -filtered_transform -olas -set_extended_classification 42 -cores 8 -keep_return 0 -odix _reclass

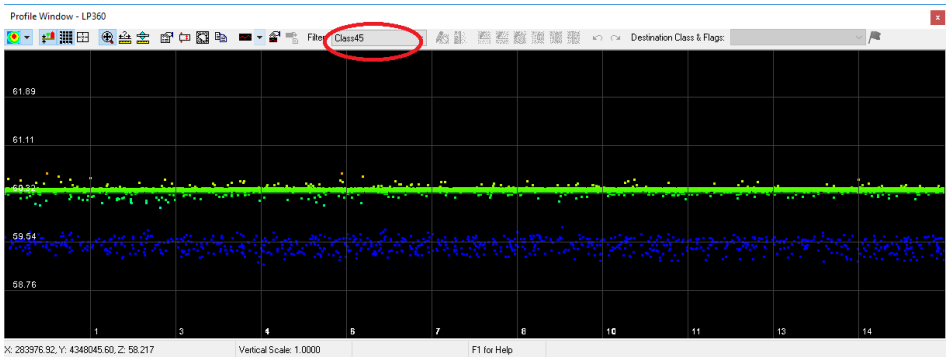

Figure 1. Synthetic (Return number zero) points have been erroneously assigned to ASPRS Class 45- (Green points) in this profile showing ONLY Class 45 points)

Photogrammetric Engineering & Remote Sensing
Vol. 86, No. 10, October 2020, pp. 595–596.
0099-1112/20/595–596

doi: 10.14358/PERS.86.10.595

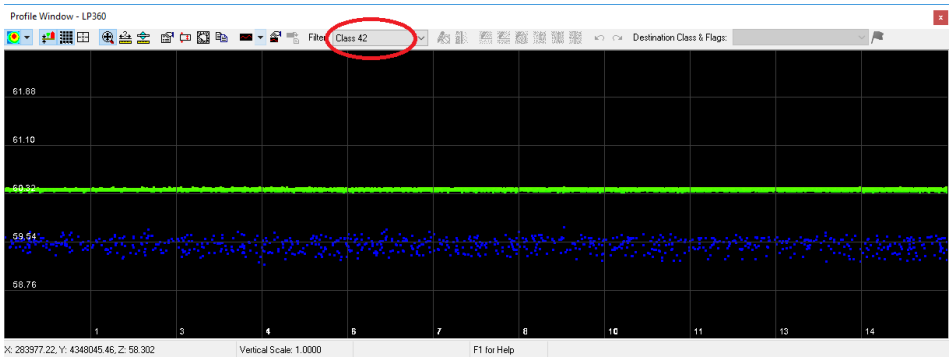

Figure 2. After running the aforementioned LAStools command, Return number zero points have been properly reclassed to ASPRS Class 42- Synthetic Water Surface (Green points) in this profile showing ONLY Class 42. The Terrascan suite does not currently support reclassification by return number, but this operation is supported in LAStools.

And there you have it, two LAStools Python scripts that produce the same results, but with the arguments in a different order.

Matt Bauld and **Al Karlin, Ph.D., CMS-L, GISP** *are with Dewberry's Geospatial and Technology Services group in Tampa, FL. Matt is a Geospatial Analyst who works primarily with processing topobathymetric lidar and constructing lidar-derived DEMs and breaklines. As a Senior Geospatial Scientist, Al works with all aspects of Lidar, remote sensing, photogrammetry, and GIS-related projects.*