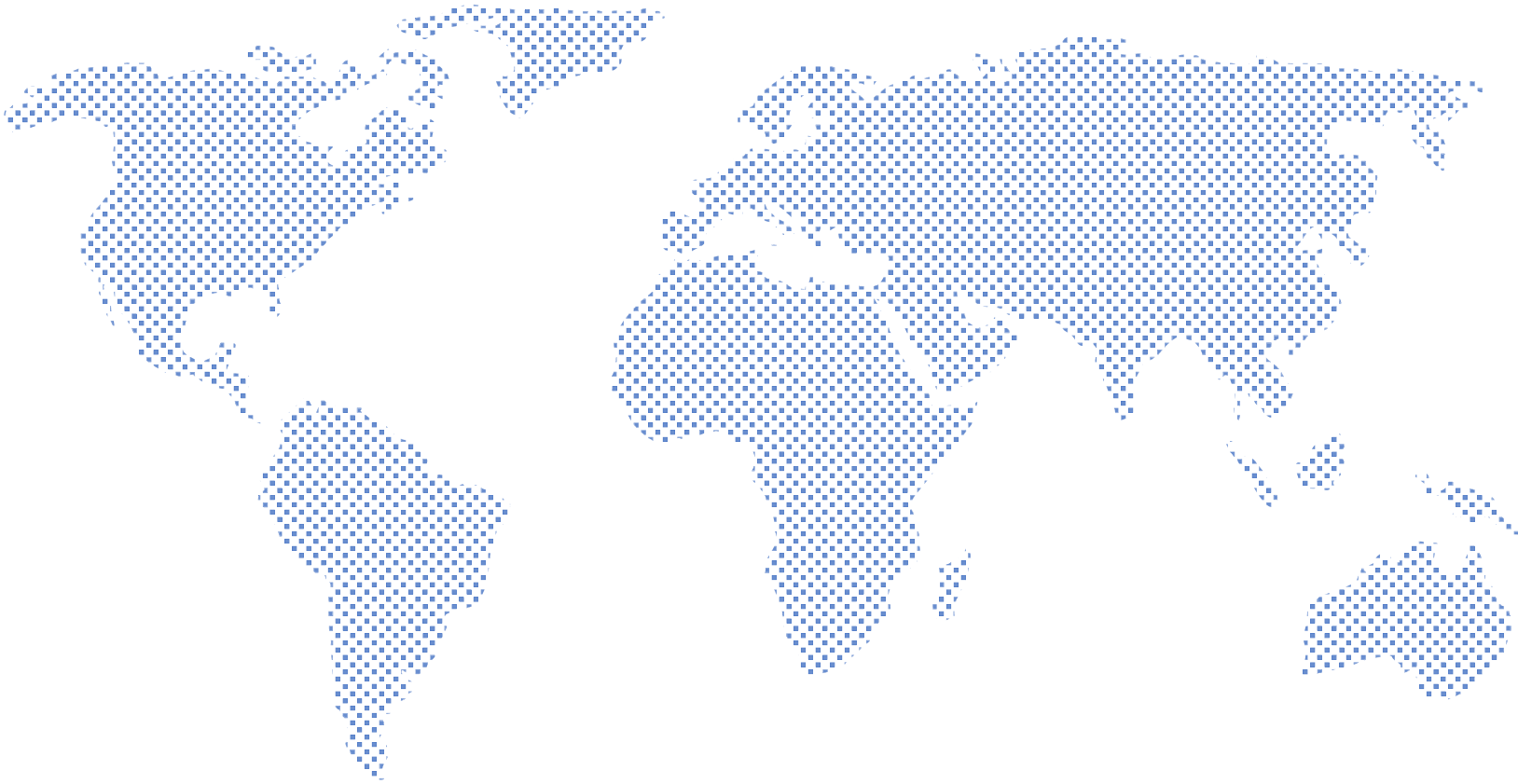
USING PYTHON TO CREATE FIGURES

Draft Tree Protection Zones

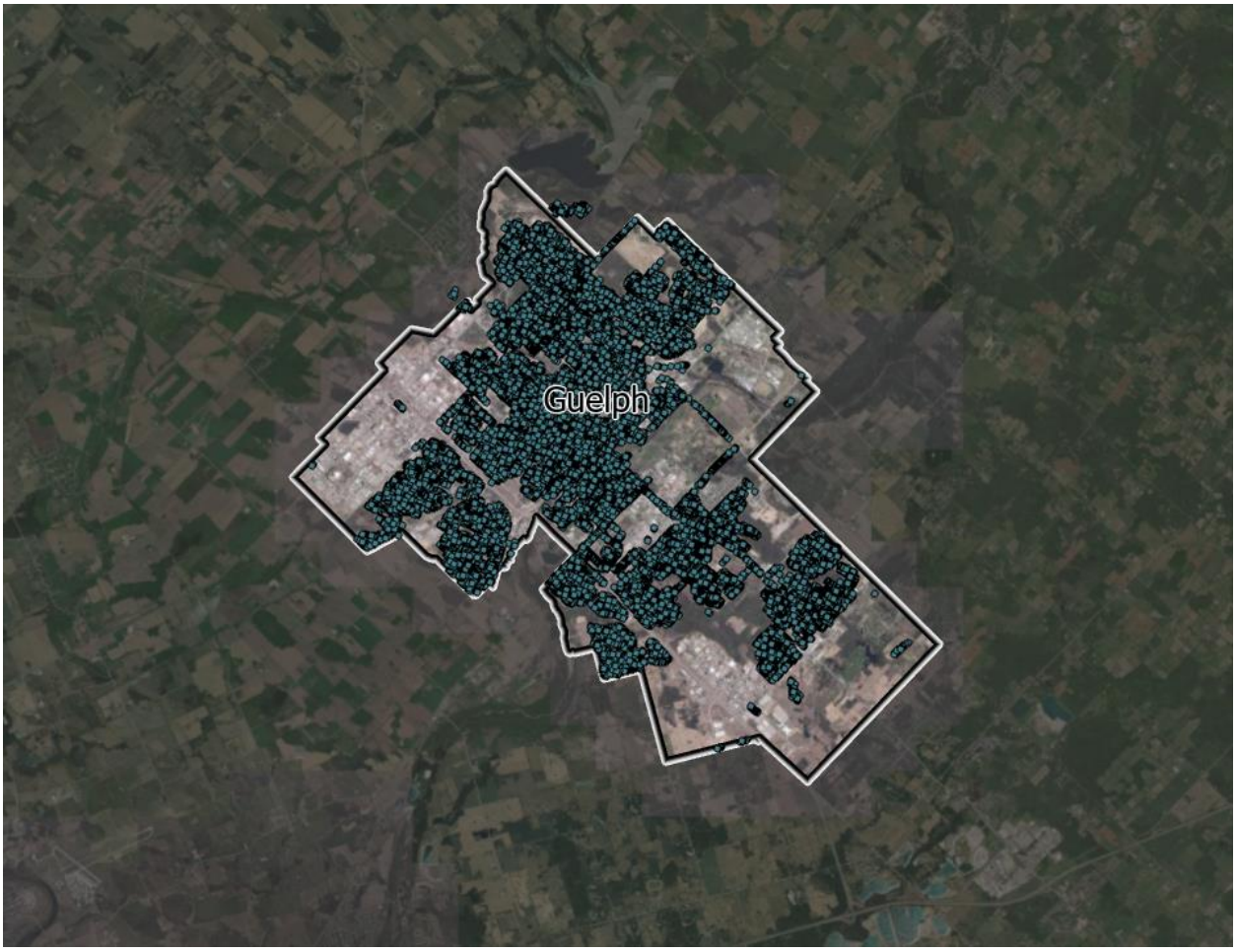


Andy Chand

Welcome to the wonderful world of Python.

- Provided data from the Arc Collector App, in this example it is tree data with the following headings:
    - Tree Number,
    - Tree Species,
    - Diameter at Breast Height (DBH),
    - Estimated Crown Width,
    - Tree Health,
    - Tree Structure,
    - Recommendation, and
    - Additional Notes/Photos.
- TPZ is set by the municipality, which is based on the DBH. However, if the canopy width is greater than the TPZ then the canopy width applies. The script automates this calculation using the DBH and estimated crown width column.
- The result is a draft figure that can be provided to the end user (arborist). The following showcases the steps in the figure generation.

An example dataset is the Guelph Tree Inventory (http://data.open.guelph.ca/dataset/tree-inventory/resource/ac473a28-21b4-4400-a801-1e810cca3deb).



OBJECTID *	Shape *	OBJECTID_1	SPECIES	DBH	HEALTH	OWNER	ADDRESS	COMMENTS	HTCLASS	OH_UTIL
1	Point	1	Sugar Maple	5	Good	City	135 CLAIRFIELDS DR W			
2	Point	2	Red Oak	5	Good	City	135 CLAIRFIELDS DR W			
3	Point	3	Sugar Maple	5	Good	City	135 CLAIRFIELDS DR W			
4	Point	4	Red Maple	5	Good	City	135 CLAIRFIELDS DR W			
5	Point	5	Red Maple	5	Good	City	135 CLAIRFIELDS DR W			
6	Point	6	Red Maple	5	Good	City	135 CLAIRFIELDS DR W			
7	Point	7	Sugar Maple	31	Excellent	Boundary	111 PTARMIGAN DR		<5m	NO
8	Point	8	Sugar Maple	20	Good	Boundary	109 PTARMIGAN DR		<5m	NO
9	Point	9	Sugar Maple	28	Good	Private	107 PTARMIGAN DR		<5m	NO
10	Point	10	Sugar Maple	26	Good	Boundary	103 PTARMIGAN DR			
11	Point	11	Sugar Maple	28	Excellent	Private	101 PTARMIGAN DR		<5m	NO
12	Point	12	Sugar Maple	24	Excellent	Private	97 PTARMIGAN DR		<5m	NO
13	Point	13	Sugar Maple	26	Excellent	Private	95 PTARMIGAN DR		<5m	NO
14	Point	14	Sugar Maple	17	Good	Boundary	84 PTARMIGAN DR	*REMOVED*		NO
15	Point	15	Sugar Maple	26	Good	Private	94 PTARMIGAN DR		<5m	NO
16	Point	16	Sugar Maple	28	Good	Private	28 PTARMIGAN DR		<5m	NO
17	Point	17	English Oak	8	Fair	City	48 TANGER DR			
18	Point	18	Red Oak	24	Excellent	Boundary	31 TANAGER DR			
19	Point	19	Silver Maple	5	Excellent	Boundary	17 TANAGER DR		<5m	NO
20	Point	20	White Pine	10	Good	City	8 TRENDELL LANE		<5m	NO
21	Point	21	Ash	24	Fair	Boundary	11 TRENDELL LANE	OWNERSHIP VERIFIED...	5m - <10m	NO
22	Point	22	White Birch	8	Fair	Boundary	7 TANAGER DR			

0 of 56,683 selected

Filters:

This dataset contains 56,683 individual trees with DBH. Since the tree protection zones are calculated using the DBH, it would take a long time going through each tree individually, or even through selection by groups. Using Python coding and the Notebook feature of ArcGIS Pro you can run a script that will calculate the buffer distance and put them on a map within minutes.

- This initial feature allows the user to code different input variables – hence it is reusable for different maps and locations with minor input changes.

```
#---------------------------------------------------------
# Program: Creating Tree Protection Zones with Diameter at Breast Height
# Author: Andy Chand
# Version: 1.0
# Date: Jan. 19, 2021
# Purpose: Takes in points and measured DBH's (Diameter at breast height) of trees to find
# and then apply a tree protection zone (TPZ) by dissolving overlapping zones
# Dependencies: arcpy
#---------------------------------------------------------

import arcpy

# Listing the feature class and attribute table - used instead of a new Database
inFC = "TreeInventory"
inField = "DBH"
```

- Using the Guelph provided tree technical manual we were able to grab the buffer values based on the different DBH ranges; the manual also included a calculation for trees larger than 100 DBH, which was calculated and returned.

```
# Calculates the buffer area using the DBH and the tree technical manual provided by Guelph
def BArea(DBH):
    if DBH < 10:
        BuffArea = "1.2 METERS"
    elif DBH >= 10 and DBH <= 29:
        BuffArea = "1.8 METERS"
    elif DBH >= 30 and DBH <= 40:
        BuffArea = "2.4 METERS"
    elif DBH >= 41 and DBH <= 50:
        BuffArea = "3.0 METERS"
    elif DBH >= 51 and DBH <= 60:
        BuffArea = "3.6 METERS"
    elif DBH >= 61 and DBH <= 70:
        BuffArea = "4.2 METERS"
    elif DBH >= 71 and DBH <= 80:
        BuffArea = "4.8 METERS"
    elif DBH >= 81 and DBH <= 90:
        BuffArea = "5.4 METERS"
    elif DBH >= 91 and DBH <= 100:
        BuffArea = "1.8 METERS"
    elif DBH > 100 and DBH < 999:
        BuffCalc = (DBH * 6)/100
        BuffArea = str(BuffCalc) + " METERS"
    else:
        BuffArea = "0 METERS"

    if DBH > 0:
        return BuffArea
    else:
        return 0
```

- The TreeInventory Feature layer and attribute table was then updated to include the buffer distance for each tree.

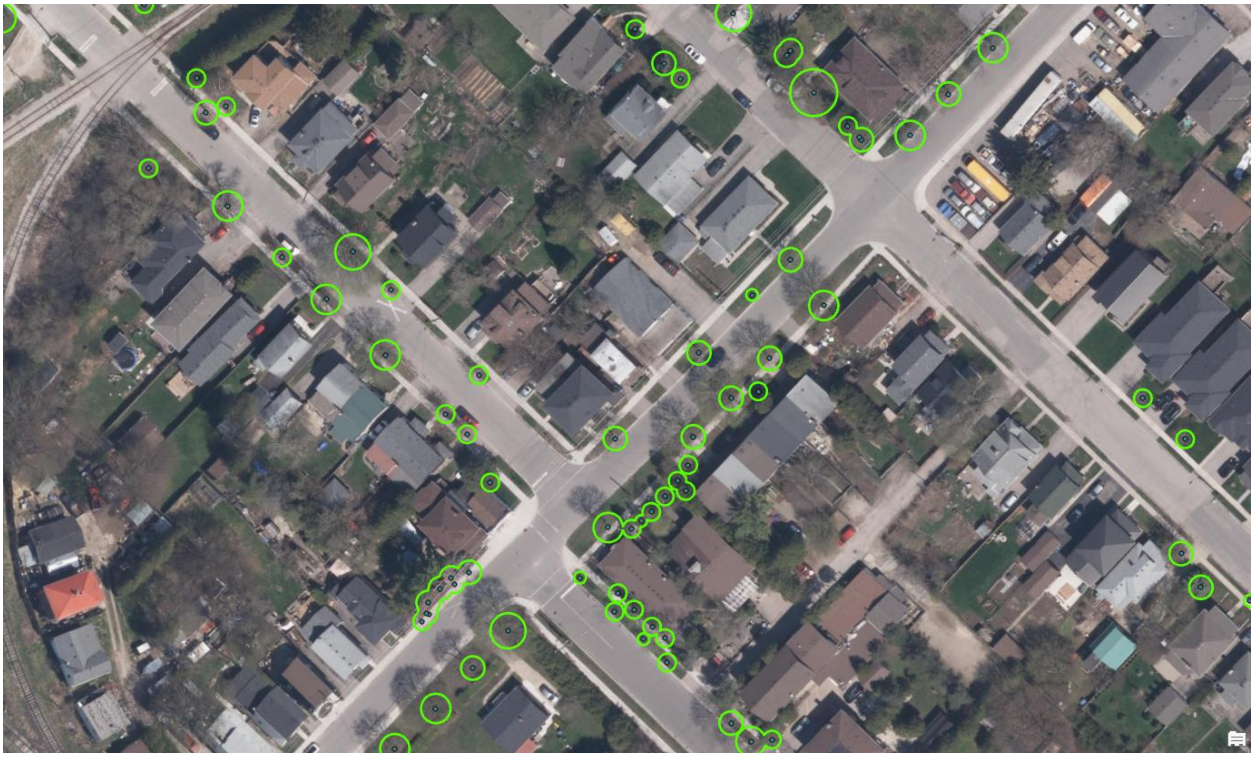
```
#Adds buffer field that will hold the buffer area
arcpy.management.AddField(inFC, "Buffer_Area", "TEXT")
myCursor = arcpy.da.UpdateCursor(inFC, [inField, "Buffer_Area"])

# Runs a loop to go through the list and calculate the buffer area using the function above
for row in myCursor:
    row[1] = BArea (row[0])
    myCursor.updateRow(row)
```

- Updating the Python code, ArcGIS was able to buffer individual trees based on the city by-laws.

OBJECTID *	Shape *	OBJECTID_1	SPECIES	DBH	HEALTH	OWNER	ADDRESS	COMMENTS	HTCLASS	OH_UTIL	Buffer_Area
1	Point	1	Sugar Maple	5	Good	City	135 CLAIRFIELDS DR W				1.2 METERS
2	Point	2	Red Oak	5	Good	City	135 CLAIRFIELDS DR W				1.2 METERS
3	Point	3	Sugar Maple	5	Good	City	135 CLAIRFIELDS DR W				1.2 METERS
4	Point	4	Red Maple	5	Good	City	135 CLAIRFIELDS DR W				1.2 METERS
5	Point	5	Red Maple	5	Good	City	135 CLAIRFIELDS DR W				1.2 METERS
6	Point	6	Red Maple	5	Good	City	135 CLAIRFIELDS DR W				1.2 METERS
7	Point	7	Sugar Maple	31	Excellent	Boundary	111 PTARMIGAN DR		<5m	NO	2.4 METERS
8	Point	8	Sugar Maple	20	Good	Boundary	109 PTARMIGAN DR		<5m	NO	1.8 METERS
9	Point	9	Sugar Maple	28	Good	Private	107 PTARMIGAN DR		<5m	NO	1.8 METERS
10	Point	10	Sugar Maple	26	Good	Boundary	103 PTARMIGAN DR				1.8 METERS

```
# Buffers the points based on attribute tables
arcpy.analysis.Buffer(inFC, "TreeBuffer", "Buffer_Area", "FULL", "ROUND", "LIST")
```

<u>Final Results at Various Points Across Guelph:</u>



As shown in the few examples above, thousands of points can be run through the script with different variables and calculations, turning it automatically into a draft figure. In our example, each tree has an individually calculated TPZ, which can be provided on top of an area of disturbance as a preliminary figure to the arborist. Removal recommendations can also be included or edited at this stage. Furthermore, overlapping buffer zones have been combined in the above examples to provide initial TPZ fencing. Python scripts can be saved, applied to other projects, and modified as needed e.g., different DBH calculations/ranges.