Minnesota DNR LiDAR Data ETL (Lab 2 - Part 1.1)

GIS 5571: ArcGIS I
University of Minnesota

Luke Zaruba
October 9, 2022

In [1]:
```
# Import Libraries
import arcpy
import requests
import os
from IPython import display
```

Downloading LiDAR Data (Part 1.1.A)

In [2]:
```
# Function to Prep for Downloading LAS Files
def downloadPrep(wd, county=None):
    # Create Base Data Storage Folder
    data_path = os.path.join(wd, "data")

    if os.path.exists(data_path):
        pass
    else:
        os.mkdir(data_path)

    # Check if County or Example, Create Proper Dirs
    if county is not None:
        county = county.lower()
        cty_path = os.path.join(data_path, county)

        if os.path.exists(cty_path):
            pass
        else:
            os.mkdir(cty_path)

        base_url = "https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/CTY/"
        base_url = base_url.replace("CTY", county)

        # Request Tile Map PDF
        tile_map_url = base_url + "tile_index_map.pdf"

        resp = requests.get(tile_map_url)

        pdf_path = os.path.join(cty_path, "tile_map.pdf")

        # Write Tile Map to a PDF file
        if os.path.exists(pdf_path):
            pass
        else:
            with open(pdf_path, "wb") as pdf:
                pdf.write(resp.content)

        # Display PDF
        pdf_name = "./data/CTY/tile_map.pdf".replace("CTY", county)

        # THIS IS NOT SUPPORTED WITHIIN ARCGIS PRO
        #return display.IFrame(pdf_name, width=600, height=900)

    else:
        examples_path = os.path.join(data_path, "examples")

        if os.path.exists(examples_path):
            pass
        else:
            os.mkdir(examples_path)
```

In [3]:
```
# Function to Download an LAS File
def downloadLAS(wd, tile, county=None):
    # Check if file is from a County or from Examples
    if county is not None:
        county = county.lower()
        base_url = "https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/CTY/laz/"
        base_url = base_url.replace("CTY", county)

        # Check if Necessary Dir is Created
        cty_path = os.path.join(wd, "data", county)

        if os.path.exists(cty_path):
            pass
        else:
            raise Exception("Run the downloadPrep function, before running the downloadLAS function.")

        # Download Tile
        tile_url = base_url + tile + ".laz"

        resp = requests.get(tile_url, stream = True)

        laz_path = os.path.join(cty_path, f"{tile}.laz")

        if os.path.exists(laz_path):
            print(f"Tile {tile} already exists.")
            pass
        else:
            with open(laz_path, "wb") as laz:
                laz.write(resp.content)
                print(f"Download complete for tile {tile}")
    else:
        # Download Tile
        tile_url = "https://resources.gisdata.mn.gov/pub/data/elevation/lidar/examples/lidar_sample/las/" + tile + ".laz"

        resp = requests.get(tile_url, stream = True)

        ex_path = os.path.join(wd, "data/examples")

        laz_path = os.path.join(ex_path, f"{tile}.las")

        if os.path.exists(laz_path):
            print(f"Tile {tile} already exists.")
            pass
        else:
            with open(laz_path, "wb") as laz:
                laz.write(resp.content)
                print(f"Download complete for tile {tile}.")
```

In [4]:
```
# Calling the Prep Function
directory = r"C:\gitFiles\GIS5571\Lab2"

downloadPrep(directory, "chisago")
```

In [5]:
```
# Calling the Download Function
directory = r"C:\gitFiles\GIS5571\Lab2"

downloadLAS(directory, "3542-20-3", "chisago")
```

```
Tile 3542-20-3 already exists.
```

Converting and Storing LiDAR Data (Part 1.1.B-C)

In [6]:
```
# Inputs
laz = r"C:\gitFiles\GIS5571\Lab2\data\aitkin\1942-32-08.laz"
out_dir = r"C:\gitFiles\GIS5571\Lab2\data\aitkin"
lasd = r"C:\gitFiles\GIS5571\Lab2\data\aitkin\1942-32-08.lasd"
sr = 'PROJCS["NAD_1983_UTM_Zone_15N",GEOGCS["GCS_North_American_1983",DATUM["D_North_American_1983",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Greenwich",0.0

# Convert LAZ to LAS
las = arcpy.conversion.ConvertLas(laz, out_dir, "1.4", "6", "NO_COMPRESSION", "REARRANGE_POINTS", lasd, "ALL_FILES", sr)
```

In [7]:
```
# Convert LASD to TIN
tin = arcpy.ddd.LasDatasetToTin(lasd, r"C:\gitFiles\GIS5571\Lab2\data\aitkin\1942-32-08_TIN", "WINDOW_SIZE", "MIN", 15, 5000000, 1, "CLIP")
```

In [8]:
```
# Convert LASD to DEM
dem = arcpy.conversion.LasDatasetToRaster(lasd, r"c:\gitFiles\GIS5571\Lab2\data\aitkin\dem.tif")
```

Mapping LiDAR Data (Part 1.1.D)

In [9]:
```
# Set APRX and Layout
aprx = arcpy.mp.ArcGISProject("CURRENT")
tin_lyt = aprx.listLayouts("TIN")[0]

# Export Layout to PDF
tin_lyt.exportToPDF(r"C:\gitFiles\GIS5571\Lab2\data\aitkin\1942-32-08_tin.pdf", resolution = 300)
```

Out[9]:
```
'C:\\gitFiles\\GIS5571\\Lab2\\data\\aitkin\\1942-32-08_tin.pdf'
```

In [10]:
```
# Set APRX and Layout
aprx = arcpy.mp.ArcGISProject("CURRENT")
dem_lyt = aprx.listLayouts("DEM")[0]

# Export Layout to PDF
dem_lyt.exportToPDF(r"C:\gitFiles\GIS5571\Lab2\data\aitkin\1942-32-08_dem.pdf", resolution = 300)
```

Out[10]:
```
'C:\\gitFiles\\GIS5571\\Lab2\\data\\aitkin\\1942-32-08_dem.pdf'
```