Connolly Lab 3_Part1

November 29, 2023

[70]:
```
import requests
import urllib3
import os
import zipfile
urllib3.disable_warnings()
```

0.0.1 Note: This code is repurposed from Lab 2, as such you will find code denoting folders for Lab 2 instead of Lab 3

[ ]:
```
#First we set up our work environment
workspace = r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3"

#Now we have to set our start point.
x = -92.148796
y = 44.127985


#Then we define the spatial reference, in this case :WGS 1984 - EPSG:4326
spatial_reference = arcpy.SpatialReference(4326)

#Then we create the point geometry
point = arcpy.Point(x, y)
pointGeometry = arcpy.PointGeometry(point, spatial_reference)

#Next we create a feature class
featureClass = "Start.shp"
arcpy.CreateFeatureclass_management(workspace, featureClass, "POINT",␣
↪spatial_reference=spatial_reference)

#Then we open an insert cursor and insert the point
with arcpy.da.InsertCursor(featureClass, ["SHAPE@"]) as cursor:
    cursor.insertRow([pointGeometry])
```

[ ]:
```
#Now we have to do the same for the end point
# Define the x and y coordinates
x = -92.04482631030432
y = 44.054409951002576
```

```
spatial_reference = arcpy.SpatialReference(4326)
point = arcpy.Point(x, y)
pointGeometry = arcpy.PointGeometry(point, spatial_reference)


featureClass = "End.shp"
arcpy.CreateFeatureclass_management(workspace, featureClass, "POINT",␣
↪spatial_reference=spatial_reference)

with arcpy.da.InsertCursor(featureClass, ["SHAPE@"]) as cursor:
    cursor.insertRow([pointGeometry])
```

[58]:
```
#Now we have to fetch the land class data we'll need for this from the␣
↪Minnesota Geospatial Commons
#We'll use an API call via the requests tool
api_url="https://gisdata.mn.gov/api/3/action/package_show?
↪id=biota-landcover-nlcd-mn-2019"
response = requests.get(api_url, verify=False)
json=response.json()
```

[ ]:
```
#Next we'll acquire the link to the shapefile folder we need
resources = json['result']['resources']
for resource in resources:
    if resource['format'] == 'tif':
        #Once we find the shapefile, we can save the associated URL.
        zip_url = resource['url']
        #Next, we can save the zip_url to our directory as a zip file.
        zip_filename = os.path.basename(zip_url)

        # We can finally download the zip file and produce some code that can␣
↪will generate a text output indicating the name of the file we have␣
↪downloaded.
        response = requests.get(zip_url)
        if response.status_code == 200:
            with open(zip_filename, 'wb') as file:
                file.write(response.content)
            print(f"Downloaded {zip_filename}")
        else:
            print(f"Failed to download {zip_filename}. Status code: {response.
↪status_code}")
        break
```

[ ]:
```
#We define the path to the zipfile
```

```
zip_file_path = r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\tif_biota_landcover_nlcd_mn_2019.zip"

#Then we define the folder where we want to extract the contents
extracted_folder = r'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3'

#Then we can open the ZIP file
with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
    zip_ref.extractall(extracted_folder)

print(f'All files have been extracted to {extracted_folder}')
```

```
[ ]: import subprocess
#Next we need to import several LAZ files from the MN DNR website for our␣
↪search area
#This ist defines all the LiDAR tiles we need data for
laz_files = ["4342-28-59","4342-28-60","4342-28-61","4342-28-62","4342-28-63",
             "4342-29-59","4342-29-60","4342-29-61","4342-29-62","4342-29-63",
             "4342-30-59","4342-30-60","4342-30-61","4342-30-62","4342-30-63",
             "4342-31-59","4342-31-60","4342-31-61","4342-31-62","4342-31-63",
             "4342-32-59","4342-32-60","4342-32-61","4342-32-62","4342-32-63"]

#We create a base url
base_url = "https://resources.gisdata.mn.gov/pub/data/elevation/lidar/q250k/
↪q4342/laz/"

#Then we define our destination file
destination_dir = r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3"

#Now, we can create a for loop that will loop through the list and download our␣
↪LAZ files using a curl command
for laz_file in laz_files:
    url = base_url + laz_file + ".laz"
    output_file = destination_dir + "\\" + laz_file + ".laz"
    subprocess.run(['curl', '-o', output_file, url])
    print(f"Downloaded {laz_file}.laz")

    #This for loop will also convert any incoming LAZ files into LAS files
    arcpy.conversion.ConvertLas(
        in_las=output_file,
        target_folder=destination_dir,
        file_version="SAME_AS_INPUT",
        point_format="",
        compression="NO_COMPRESSION",
```

```
        las_options="REARRANGE_POINTS",
        out_las_dataset=None,
        define_coordinate_system="FILES_MISSING_PROJECTION",
       ␣
 ↪in_coordinate_system='GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.
 ↪0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.
 ↪0174532925199433]],VERTCS["NAVD88 - Geoid09␣
 ↪(Meters)",VDATUM["unknown"],PARAMETER["Vertical_Shift",0.
 ↪0],PARAMETER["Direction",1.0],UNIT["Meter",1.0]]'
    )

print("All Laz files downloaded and converted.")
```

[44]:
```
#Next, we will create a LAS Dataset that will contain all of our LAS tiles
arcpy.management.CreateLasDataset(
```

```
    input=r"'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-28-59.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-28-60.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-28-61.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-28-62.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-28-63.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-29-59.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-29-60.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-30-62.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-30-61.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-30-60.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-30-59.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-29-63.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-29-62.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-29-61.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-32-59.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-31-63.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-31-62.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-31-61.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-31-60.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-31-59.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-30-63.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-32-63.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-32-62.las';'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\4342-32-61.
↪las';'C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\4342-32-60.las'",
    out_las_dataset=r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS␣
↪5571 Lab2_3\LasDataset.lasd",
    folder_recursion="NO_RECURSION",
    in_surface_constraints=None,
```

```
↪spatial_reference='GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.
↪0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.
↪0174532925199433]],VERTCS["NAVD88 - Geoid09␣
↪(Meters)",VDATUM["unknown"],PARAMETER["Vertical_Shift",0.
↪0],PARAMETER["Direction",1.0],UNIT["Meter",1.0]]',
    compute_stats="COMPUTE_STATS",
    relative_paths="ABSOLUTE_PATHS",
    create_las_prj="NO_FILES",
    extent="DEFAULT",
    boundary=None,
    add_only_contained_files="INTERSECTED_FILES"
)
```

[44]: <Result 'C:\\Users\\conno\\OneDrive\\Documents\\ArcGIS\\Projects\\GIS 5571 Lab2_3\\GIS 5571 Lab2_3.gdb\\c43423163_lasd2'>

[21]:
```
#Next we will convert this LAS dataset into a DEM model
arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="LasDataset.lasd",
    out_raster=r"c:\users\conno\onedrive\documents\arcgis\projects\gis 5571␣
↪lab2_3\gis 5571 lab2_3.gdb\lasdat_lasda",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)
```

[21]: <Result 'c:\\Users\\conno\\OneDrive\\documents\\ArcGIS\\Projects\\gis 5571 lab2_3\\gis 5571 lab2_3.gdb\\lasdat_lasda'>

[22]:
```
#To prepare the DEM model, we will use the Fill command
Fill = arcpy.sa.Fill(
    in_surface_raster="lasdat_lasda",
    z_limit=None
)
Fill.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\Fill")
```

[23]:
```
#Then we will then run the output of the fill tool through the Slope command
Slope = arcpy.sa.Slope(
    in_raster=Fill,
    output_measurement="DEGREE",
    z_factor=1,
    method="PLANAR",
```

```
    z_unit="METER",
    analysis_target_device="GPU_THEN_CPU"
)
Slope.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\Slope")
```

[24]:
```
#Now, we can group the slope layer into classes
Slope_Class = arcpy.sa.Reclassify(
    in_raster=Slope,
    reclass_field="VALUE",
    remap="0 1.720000 1;1.720000 3.430000 2;3.430000 5.710000 3;5.710000 8.
↪530000 4;8.530000 11.300000 5;11.300000 14.040000 6;14.040000 16.700000 7;16.
↪700000 21.800000 8;21.800000 30.960000 9;30.960000 45 10;45 90 11",
    missing_values="DATA"
)
Slope_Class.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\Slope_Class")
```

[ ]:
```
#Now we can create a weighted overlay
#For scenario 1, we'll be using 50/50 weighting between land class and slope,␣
↪fields included
weighted_overlay = arcpy.sa.WeightedOverlay(
    in_weighted_overlay_table=r"('C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\GIS 5571␣
↪Lab2_3.gdb\Slope_Class' 50 'Value' (1 10; 2 9; 3 8; 4 7; 5 6; 6 5; 7 4; 8 3;␣
↪9 2; 10 1; 11 1; NODATA NODATA); 'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\NLCD_2019_Land_Cover.tif' 50 'NLCD_Land' ('Unclassified' NODATA;␣
↪'Open Water' 2; 'Developed, Open Space' 10; 'Developed, Low Intensity' 10;␣
↪'Developed, Medium Intensity' 10; 'Developed, High Intensity' 10; 'Barren␣
↪Land' 10; 'Deciduous Forest' 10; 'Evergreen Forest' 10; 'Mixed Forest' 10;␣
↪'Shrub/Scrub' 10; 'Herbaceous' 10; 'Hay/Pasture' 1; 'Cultivated Crops' 1;␣
↪'Woody Wetlands' 1; 'Emergent Herbaceous Wetlands' 1; NODATA NODATA));1 10 1"
)
out_raster.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\weighted_overlay")
```

[12]:
```
#Now, we create a cost distance using the start point and the weighted overlay
cost_distance = arcpy.sa.CostDistance(
    in_source_data="Start",
    in_cost_raster="weighted_overlay",
    maximum_distance=None,
    out_backlink_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
```

```
    source_direction=""
)
cost_distance.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\cost_distance")
```

[13]:
```
#Then we create a cost backlink using the start point and the weighted overlay␣
↪again
cost_backlink = arcpy.sa.CostBackLink(
    in_source_data="Start",
    in_cost_raster="weighted_overlay",
    maximum_distance=None,
    out_distance_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
cost_backlink.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\cost_backlink")
```

[14]:
```
#Finally, we use our end point to create a cost path
cost_path = arcpy.sa.CostPath(
    in_destination_data="End",
    in_cost_distance_raster="cost_distance",
    in_cost_backlink_raster="cost_backlink",
    path_type="EACH_CELL",
    destination_field="Id",
    force_flow_direction_convention="INPUT_RANGE"
)
cost_path.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\cost_path")
```

[ ]:
```
#We will use the run the same code for scenarios 2 and 3
#Scenario 2 consists of 50/50 weighting between land class and slope, no fields
weighted_overlay_2 = arcpy.sa.WeightedOverlay(
```

```
    in_weighted_overlay_table=r"('C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\GIS 5571␣
↪Lab2_3.gdb\Slope_Class' 50 'Value' (1 10; 2 9; 3 8; 4 7; 5 6; 6 5; 7 4; 8 3;␣
↪9 2; 10 1; 11 1; NODATA NODATA); 'C:
↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\NLCD_2019_Land_Cover.tif' 50 'NLCD_Land' ('Unclassified' NODATA;␣
↪'Open Water' 2; 'Developed, Open Space' 10; 'Developed, Low Intensity' 10;␣
↪'Developed, Medium Intensity' 10; 'Developed, High Intensity' 10; 'Barren␣
↪Land' 10; 'Deciduous Forest' 10; 'Evergreen Forest' 10; 'Mixed Forest' 10;␣
↪'Shrub/Scrub' 10; 'Herbaceous' 10; 'Hay/Pasture' NODATA; 'Cultivated Crops'␣
↪NODATA; 'Woody Wetlands' 1; 'Emergent Herbaceous Wetlands' 1; NODATA␣
↪NODATA));1 10 1"
)
weighted_overlay_2.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS␣
↪5571 Lab2_3\GIS 5571 Lab2_3.gdb\weighted_overlay")

cost_distance_2 = arcpy.sa.CostDistance(
    in_source_data="Start",
    in_cost_raster="weighted_overlay_2",
    maximum_distance=None,
    out_backlink_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
cost_distance_2.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS␣
↪5571 Lab2_3\GIS 5571 Lab2_3.gdb\cost_distance")

cost_backlink_2 = arcpy.sa.CostBackLink(
    in_source_data="Start",
    in_cost_raster="weighted_overlay_2",
    maximum_distance=None,
    out_distance_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
cost_backlink_2.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS␣
↪5571 Lab2_3\GIS 5571 Lab2_3.gdb\cost_backlink")

cost_path_2 = arcpy.sa.CostPath(
    in_destination_data="End",
    in_cost_distance_raster="cost_distance_2",
```

```
    in_cost_backlink_raster="cost_backlin_2",
    path_type="EACH_CELL",
    destination_field="Id",
    force_flow_direction_convention="INPUT_RANGE"
)
cost_path_2.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
 ↪Lab2_3\GIS 5571 Lab2_3.gdb\cost_path")
```

[ ]:
```
#Scenario 3 consists of a weighted overlay for 70% land classes and 30% slope,␣
 ↪no fields
weighted_overlay_3 = arcpy.sa.WeightedOverlay(
    in_weighted_overlay_table=r"('C:
 ↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571 Lab2_3\GIS 5571␣
 ↪Lab2_3.gdb\Slope_Class' 30 'Value' (1 10; 2 9; 3 8; 4 7; 5 6; 6 5; 7 4; 8 3;␣
 ↪9 2; 10 1; 11 1; NODATA NODATA); 'C:
 ↪\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
 ↪Lab2_3\NLCD_2019_Land_Cover.tif' 70 'NLCD_Land' ('Unclassified' NODATA;␣
 ↪'Open Water' 2; 'Developed, Open Space' 10; 'Developed, Low Intensity' 10;␣
 ↪'Developed, Medium Intensity' NODATA; 'Developed, High Intensity' 10;␣
 ↪'Barren Land' 10; 'Deciduous Forest' 10; 'Evergreen Forest' 10; 'Mixed␣
 ↪Forest' 10; 'Shrub/Scrub' 10; 'Herbaceous' 10; 'Hay/Pasture' NODATA;␣
 ↪'Cultivated Crops' NODATA; 'Woody Wetlands' 1; 'Emergent Herbaceous␣
 ↪Wetlands' 1; NODATA NODATA));1 10 1"
)
weighted_overlay_3.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS␣
 ↪5571 Lab2_3\GIS 5571 Lab2_3.gdb\weighted_overlay")

cost_distance_3 = arcpy.sa.CostDistance(
    in_source_data="Start",
    in_cost_raster="weighted_overlay_3",
    maximum_distance=None,
    out_backlink_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
cost_distance_3.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS␣
 ↪5571 Lab2_3\GIS 5571 Lab2_3.gdb\cost_distance")

cost_backlink_3 = arcpy.sa.CostBackLink(
    in_source_data="Start",
    in_cost_raster="weighted_overlay_3",
    maximum_distance=None,
    out_distance_raster=None,
    source_cost_multiplier=None,
```

```
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
cost_backlink_3.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS␣
↪5571 Lab2_3\GIS 5571 Lab2_3.gdb\cost_backlink")

cost_path_3 = arcpy.sa.CostPath(
    in_destination_data="End",
    in_cost_distance_raster="cost_distance_3",
    in_cost_backlink_raster="cost_backlin_3",
    path_type="EACH_CELL",
    destination_field="Id",
    force_flow_direction_convention="INPUT_RANGE"
)
cost_path_3.save(r"C:\Users\conno\OneDrive\Documents\ArcGIS\Projects\GIS 5571␣
↪Lab2_3\GIS 5571 Lab2_3.gdb\cost_path")
```