```python
# import packages
import requests
import zipfile
import arcpy
import pprint
import json
import os
import csv
from io import BytesIO
import shutil
import arcpy.mp as mp

mn_gsc_api_url = "https://gisdata.mn.gov/api/3/action"

# MN geospatial commons api search datasets request for land use

search_datasets_url = mn_gsc_api_url + "/package_search"
query = "Land Use/Cover, Agricultural and Transition Areas, Minnesota, 1990"
request_url = search_datasets_url + f"?q={query}"
response = requests.get(request_url, verify=False) # verify=False disables SSL certification validation which makes it mad but i cant get it to work without disabling it
response_data = json.loads(response.text)
found_dataset_names = [dataset["title"] for dataset in response_data["result"]["results"]]
print(found_dataset_names)
```

```
C:\Users\18284\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\lib\
site-packages\urllib3\connectionpool.py:1045: InsecureRequestWarning:
Unverified HTTPS request is being made to host 'gisdata.mn.gov'.
Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-
warnings
  warnings.warn(
```

```
['Land Use/Cover, Agricultural and Transition Areas, Minnesota, 1990',
'Land Cover, Forested Area, Minnesota, 1990s', 'Minnesota Land Use and
Cover: 1990s Census of the Land (8 category statewide)', 'Land Use:
Camp Ripley and Beltrami Island State Forest, Minnesota', 'NLCD 2001
Land Cover', 'NLCD 2006 Land Cover', 'Generalized Land Use 2000',
'Generalized Land Use 2005', 'Generalized Land Use 2016', 'Generalized
Land Use 2010']
```

```python
# MN geospatial commons api get dataset request for land use

first_dataset_result_id = response_data["result"]["results"][0]["id"]
# grab dataset that is first in returned dataset results
get_dataset_url = mn_gsc_api_url + "/package_show"
id = first_dataset_result_id
```

```
request_url = get_dataset_url + f"?id={id}"
response = requests.get(request_url, verify=False) # again disabling
ssl cert validation
response_data = json.loads(response.text)
resources = response_data["result"]["resources"]
raw_zip_dataset_file_download_url = [resource for resource in
resources if resource["resource_type"] == "fgdb"][0]["url"]
print(raw_zip_dataset_file_download_url)
```

```
C:\Users\18284\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\lib\
site-packages\urllib3\connectionpool.py:1045: InsecureRequestWarning:
Unverified HTTPS request is being made to host 'gisdata.mn.gov'.
Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-
warnings
  warnings.warn(
```

```
https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/
biota_landcover_landsat_tm_1990/
fgdb_biota_landcover_landsat_tm_1990.zip
```

```
# download landuse
response = requests.get(raw_zip_dataset_file_download_url)
# check response.status_code to see if download was successful, should
return '200'
os.chdir(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3")
if response.status_code == 200:
    # Open zipfile
    with zipfile.ZipFile(BytesIO(response.content)) as z:
        # extract contents of zipfile
        z.extractall('landuse') # extracts contents into a folder
named 'landuse'

        # assuming there's only one landuse file in the zip, get its
name
        landuse_file = [name for name in z.namelist()]
        print(landuse_file)
```

```
['Land Use-Cover, Ag and Transition Areas, 1990.lyr', 'Land Use-Cover,
Ag and Transition Areas, 1990.lyrx', 'metadata/metadata.html',
'metadata/metadata.xml', 'metadata/preview.jpg', 'metadata/Thumbs.db',
'biota_landcover_landsat_tm_1990.gdb/a00000006.CatRelTypesByBackwardLa
bel.atx', 'biota_landcover_landsat_tm_1990.gdb/a00000004.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/a00000006.gdbindexes',
'biota_landcover_landsat_tm_1990.gdb/a00000007.gdbindexes',
'biota_landcover_landsat_tm_1990.gdb/a00000009.gdbtable.FDO_OBJECTID.g
dbtabidx',
'biota_landcover_landsat_tm_1990.gdb/a00000004.CatItemsByType.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000006.CatRelTypesByOriginItem
TypeID.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000005.FDO_UUID.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000007.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/a00000004.gdbindexes',
'biota_landcover_landsat_tm_1990.gdb/a00000004.horizon',
'biota_landcover_landsat_tm_1990.gdb/a00000005.CatRelsByDestinationID.
atx', 'biota_landcover_landsat_tm_1990.gdb/a00000001.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/a00000007.CatItemTypesByUUID.atx'
, 'biota_landcover_landsat_tm_1990.gdb/a00000003.gdbindexes',
'biota_landcover_landsat_tm_1990.gdb/a00000007.gdbtablx',
'biota_landcover_landsat_tm_1990.gdb/a00000001.gdbtablx',
'biota_landcover_landsat_tm_1990.gdb/a00000001.TablesByName.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000006.CatRelTypesByUUID.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000005.CatRelsByType.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000004.gdbtablx',
'biota_landcover_landsat_tm_1990.gdb/a00000007.CatItemTypesByName.atx'
, 'biota_landcover_landsat_tm_1990.gdb/a00000004.spx',
'biota_landcover_landsat_tm_1990.gdb/a00000005.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/a00000002.gdbtablx',
'biota_landcover_landsat_tm_1990.gdb/a00000009.gdbtable.cdf',
'biota_landcover_landsat_tm_1990.gdb/a00000002.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/a00000004.freelist',
'biota_landcover_landsat_tm_1990.gdb/a00000006.CatRelTypesByForwardLab
el.atx', 'biota_landcover_landsat_tm_1990.gdb/a00000005.gdbtablx',
'biota_landcover_landsat_tm_1990.gdb/a00000009.gdbtable.XLUSE_CODE_IDX
.gdbtabidx',
'biota_landcover_landsat_tm_1990.gdb/a00000001.gdbindexes',
'biota_landcover_landsat_tm_1990.gdb/a00000003.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/a00000006.CatRelTypesByDestItemTy
peID.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000007.CatItemTypesByParentTyp
eID.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000005.CatRelsByOriginID.atx',
'biota_landcover_landsat_tm_1990.gdb/gdb',
'biota_landcover_landsat_tm_1990.gdb/a00000005.gdbindexes',
'biota_landcover_landsat_tm_1990.gdb/a00000003.gdbtablx',
'biota_landcover_landsat_tm_1990.gdb/a00000009.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/timestamps',
'biota_landcover_landsat_tm_1990.gdb/a00000006.CatRelTypesByName.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000009.gdbtable.LUSE_CODE_IDX.
gdbtabidx', 'biota_landcover_landsat_tm_1990.gdb/a00000006.gdbtable',
'biota_landcover_landsat_tm_1990.gdb/a00000006.gdbtablx',
'biota_landcover_landsat_tm_1990.gdb/a00000004.FDO_UUID.atx',
'biota_landcover_landsat_tm_1990.gdb/a00000004.CatItemsByPhysicalName.
atx']

mn_gsc_api_url = "https://gisdata.mn.gov/api/3/action"

# MN geospatial commons api search datasets request for counties

search_datasets_url = mn_gsc_api_url + "/package_search"
```

```python
query = "County Boundaries, Minnesota"
request_url = search_datasets_url + f"?q={query}"
response = requests.get(request_url, verify=False) # verify=False disables SSL certification validation which makes it mad but i cant get it to work without disabling it
response_data = json.loads(response.text)
found_dataset_names = [dataset["title"] for dataset in response_data["result"]["results"]]
print(found_dataset_names)
```

```
C:\Users\18284\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\lib\
site-packages\urllib3\connectionpool.py:1045: InsecureRequestWarning:
Unverified HTTPS request is being made to host 'gisdata.mn.gov'.
Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-
warnings
  warnings.warn(
```

```
['County Boundaries in Minnesota', 'County Boundaries, Minnesota',
'Municipal Boundaries, Dakota County, Minnesota', 'Administrative
Boundaries, Ramsey County, Minnesota', 'Political and Administrative
Boundaries, Dakota County, Minnesota', 'State Boundary, Minnesota',
'Boundaries of Minnesota', 'Senate District Boundaries, Minnesota,
2012', 'House District Boundaries, Minnesota, 1994', 'House District
Boundaries, Minnesota, 2002']
```

```python
# MN geospatial commons api get dataset request for counties

first_dataset_result_id = response_data["result"]["results"][1]["id"]
# grab dataset that is second in returned dataset results
get_dataset_url = mn_gsc_api_url + "/package_show"
id = first_dataset_result_id
request_url = get_dataset_url + f"?id={id}"
response = requests.get(request_url, verify=False) # again disabling ssl cert validation
response_data = json.loads(response.text)
resources = response_data["result"]["resources"]
raw_zip_dataset_file_download_url = [resource for resource in resources if resource["resource_type"] == "fgdb"][0]["url"]
print(raw_zip_dataset_file_download_url)
```

```
C:\Users\18284\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\lib\
site-packages\urllib3\connectionpool.py:1045: InsecureRequestWarning:
Unverified HTTPS request is being made to host 'gisdata.mn.gov'.
Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-
warnings
  warnings.warn(
https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/
bdry_counties_in_minnesota/fgdb_bdry_counties_in_minnesota.zip
```

```python
# download counties
response = requests.get(raw_zip_dataset_file_download_url)
# check response.status_code to see if download was successful, should
return '200'
os.chdir(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3")
if response.status_code == 200:
    # Open zipfile
    with zipfile.ZipFile(BytesIO(response.content)) as z:
        # extract contents of zipfile
        z.extractall('counties') # extracts contents into a folder
named 'counties'

        # assuming there's only one landuse file in the zip, get its
name
        counties_file = [name for name in z.namelist()]
        print(counties_file)
```

```
['Counties in Minnesota.qlr', 'Counties in Minnesota.lyr',
'metadata/metadata.html', 'metadata/metadata.xml',
'bdry_counties_in_minnesota.gdb/a0000000e.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000d.gdbtablx',
'bdry_counties_in_minnesota.gdb/a0000000e.gdbindexes',
'bdry_counties_in_minnesota.gdb/a00000006.CatRelTypesByBackwardLabel.a
tx', 'bdry_counties_in_minnesota.gdb/a00000004.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000c.horizon',
'bdry_counties_in_minnesota.gdb/a00000006.gdbindexes',
'bdry_counties_in_minnesota.gdb/a00000007.gdbindexes',
'bdry_counties_in_minnesota.gdb/a0000000c.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000004.CatItemsByType.atx',
'bdry_counties_in_minnesota.gdb/a00000006.CatRelTypesByOriginItemTypeI
D.atx', 'bdry_counties_in_minnesota.gdb/a00000005.FDO_UUID.atx',
'bdry_counties_in_minnesota.gdb/a00000007.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000d.spx',
'bdry_counties_in_minnesota.gdb/a00000009.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000004.gdbindexes',
'bdry_counties_in_minnesota.gdb/a0000000f.gdbtablx',
'bdry_counties_in_minnesota.gdb/a0000000a.I8Status.atx',
'bdry_counties_in_minnesota.gdb/a0000000a.G8FeatureID.atx',
'bdry_counties_in_minnesota.gdb/a00000004.horizon',
'bdry_counties_in_minnesota.gdb/a00000005.CatRelsByDestinationID.atx',
'bdry_counties_in_minnesota.gdb/a00000001.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000e.I12AnnotationCla.atx',
'bdry_counties_in_minnesota.gdb/a0000000a.spx',
'bdry_counties_in_minnesota.gdb/a0000000e.I12Status.atx',
'bdry_counties_in_minnesota.gdb/a0000000b.gdbtablx',
'bdry_counties_in_minnesota.gdb/a0000000a.horizon',
'bdry_counties_in_minnesota.gdb/a00000007.CatItemTypesByUUID.atx',
'bdry_counties_in_minnesota.gdb/a00000003.gdbindexes',
'bdry_counties_in_minnesota.gdb/a0000000a.gdbindexes',
'bdry_counties_in_minnesota.gdb/a0000000a.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000007.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000009.horizon',
'bdry_counties_in_minnesota.gdb/a0000000b.gdbindexes',
'bdry_counties_in_minnesota.gdb/a00000001.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000001.TablesByName.atx',
'bdry_counties_in_minnesota.gdb/a00000006.CatRelTypesByUUID.atx',
'bdry_counties_in_minnesota.gdb/a0000000b.spx',
'bdry_counties_in_minnesota.gdb/a00000005.CatRelsByType.atx',
'bdry_counties_in_minnesota.gdb/a00000004.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000007.CatItemTypesByName.atx',
'bdry_counties_in_minnesota.gdb/a00000004.spx',
'bdry_counties_in_minnesota.gdb/a00000005.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000a.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000f.gdbindexes',
'bdry_counties_in_minnesota.gdb/a00000002.gdbtablx',
'bdry_counties_in_minnesota.gdb/a0000000c.gdbtable',
'bdry_counties_in_minnesota.gdb/a00000002.gdbtable',
'bdry_counties_in_minnesota.gdb/a00000006.CatRelTypesByForwardLabel.at
x', 'bdry_counties_in_minnesota.gdb/a00000005.gdbtablx',
'bdry_counties_in_minnesota.gdb/a0000000d.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000d.horizon',
'bdry_counties_in_minnesota.gdb/a00000001.gdbindexes',
'bdry_counties_in_minnesota.gdb/a0000000e.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000003.gdbtable',
'bdry_counties_in_minnesota.gdb/a00000006.CatRelTypesByDestItemTypeID.
atx',
'bdry_counties_in_minnesota.gdb/a00000007.CatItemTypesByParentTypeID.a
tx', 'bdry_counties_in_minnesota.gdb/a00000005.CatRelsByOriginID.atx',
'bdry_counties_in_minnesota.gdb/a0000000a.I8AnnotationClas.atx',
'bdry_counties_in_minnesota.gdb/a0000000e.horizon',
'bdry_counties_in_minnesota.gdb/a0000000d.gdbindexes',
'bdry_counties_in_minnesota.gdb/gdb',
'bdry_counties_in_minnesota.gdb/a00000005.gdbindexes',
'bdry_counties_in_minnesota.gdb/a00000003.gdbtablx',
'bdry_counties_in_minnesota.gdb/a0000000f.horizon',
'bdry_counties_in_minnesota.gdb/a00000009.gdbtable',
'bdry_counties_in_minnesota.gdb/timestamps',
'bdry_counties_in_minnesota.gdb/a0000000c.gdbindexes',
'bdry_counties_in_minnesota.gdb/a00000009.spx',
'bdry_counties_in_minnesota.gdb/a00000006.CatRelTypesByName.atx',
'bdry_counties_in_minnesota.gdb/a00000009.gdbindexes',
'bdry_counties_in_minnesota.gdb/a0000000b.horizon',
'bdry_counties_in_minnesota.gdb/a0000000e.spx',
'bdry_counties_in_minnesota.gdb/a00000006.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000f.gdbtable',
'bdry_counties_in_minnesota.gdb/a0000000c.spx',
'bdry_counties_in_minnesota.gdb/a00000006.gdbtablx',
'bdry_counties_in_minnesota.gdb/a00000004.FDO_UUID.atx',
'bdry_counties_in_minnesota.gdb/a0000000b.gdbtable',
'bdry_counties_in_minnesota.gdb/a00000004.CatItemsByPhysicalName.atx',
'bdry_counties_in_minnesota.gdb/a0000000f.spx']
```

```
# export out wabasha & winona & olmstead county
arcpy.conversion.ExportFeatures("mn_county_boundaries", r"C:\Users\
18284\Documents\ArcGIS\Projects\lab3\arc1lab3\counties", "CTY_ABBR =
'WABA' OR CTY_ABBR = 'WINO' OR CTY_ABBR = 'OLMS'")
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
arc1lab3\\counties.shp'>
```

```
# clip landuse data with county data
arcpy.analysis.Clip(
    in_features="landcover_landsat_tm_1990",
    clip_features="counties",
    out_feature_class=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\landcover_landsat_tm_19_Clip",
    cluster_tolerance=None
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
lab3.gdb\\landcover_landsat_tm_19_Clip'>
```

```
# make a dictionary for the .laz api urls
# pulled from lab 2
folder = r'C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz'
laz_dict = {
    "wino1":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-30-62.laz',
    "wino2":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-30-63.laz',
    "wino3":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-30-64.laz',
    "wino4":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-31-64.laz',
    "wino5":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-31-63.laz',
    "wino6":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-31-62.laz',
    "wino7":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-29-63.laz',
    "wino8":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
winona/laz/4342-29-64.laz',
    "waba1":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-28-59.laz',
    "waba2":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-28-60.laz',
    "waba3":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-28-61.laz',
    "waba4":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-28-62.laz',
    "waba5":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-29-59.laz',
    "waba6":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-29-60.laz',
    "waba7":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-29-61.laz',
    "waba8":
'https://resources.gisdata.mn.gov/pub/data/elevation/lidar/county/
wabasha/laz/4342-29-62.laz'

}

# for-loop goes thru and opens all dictionary items
for key, url in laz_dict.items():
    response = requests.get(url, stream = True)

    if response.status_code == 200:
        # Save the data with the name of the dictionary key
        with open(os.path.join(folder, f"{key}.laz"), "wb") as file:
         file.write(response.content)
        print(os.path.join(folder, f"{key}.laz"))
    else:
        print(f"Failed to download data from {key}. Status code:
{response.status_code}")
```

```
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino1.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino2.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino3.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino4.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino5.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino6.laz
```

```
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino7.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\wino8.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba1.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba2.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba3.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba4.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba5.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba6.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba7.laz
C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\laz\waba8.laz
```

```
# convert laz to las

arcpy.conversion.ConvertLas(
    in_las=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\
laz",
    target_folder=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\las",
    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='PROJCS["datum_D_North_American_1983_HARN_UTM_Zon
e_15N",GEOGCS["GCS_datum_D_North_American_1983_HARN",DATUM["D_unknown"
,SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Greenwich",0.0]
,UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"],
PARAMETER["false_easting",500000.0],PARAMETER["false_northing",0.0],PA
RAMETER["central_meridian",-
93.0],PARAMETER["scale_factor",0.9996],PARAMETER["latitude_of_origin",
0.0],UNIT["Meter",1.0]],VERTCS["NAVD88 - Geoid03
(Meters)",VDATUM["unknown"],PARAMETER["Vertical_Shift",0.0],PARAMETER[
"Direction",1.0],UNIT["Meter",1.0]]'
)
```

```
<Result ''>
```

```
# convert all las files to dem
# pulled from lab2
arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba1.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\waba1tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba2.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\waba2tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba3.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\waba3tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba4.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\waba4tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba5.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\waba5tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)
```

```
arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba6.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\waba6tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba7.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\waba7tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="waba8.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\waba8tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino1.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\arc1lab3\wino1tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino2.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\wino2tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino3.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\wino3tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino4.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\wino4tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino5.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\wino5tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino6.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\wino6tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino7.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\wino7tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)

arcpy.conversion.LasDatasetToRaster(
    in_las_dataset="wino8.las",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3\wino8tif",
    value_field="ELEVATION",
    interpolation_type="BINNING AVERAGE LINEAR",
    data_type="FLOAT",
    sampling_type="CELLSIZE",
    sampling_value=10,
    z_factor=1
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
arc1lab3\\wino8tif'>
```

```
# merge tifs into one raster

arcpy.management.MosaicToNewRaster(

input_rasters="wino8tif;wino7tif;wino6tif;wino5tif;wino4tif;wino3tif;w
ino2tif;wino1tif;waba8tif;waba7tif;waba6tif;waba5tif;waba4tif;waba3tif
;waba2tif;waba1tif",
    output_location=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
arc1lab3",
    raster_dataset_name_with_extension="interestarea1",

coordinate_system_for_the_raster='PROJCS["datum_D_North_American_1983_
HARN_UTM_Zone_15N",GEOGCS["GCS_datum_D_North_American_1983_HARN",DATUM
["D_unknown",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Gre
enwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse
_Mercator"],PARAMETER["false_easting",500000.0],PARAMETER["false_north
ing",0.0],PARAMETER["central_meridian",-
93.0],PARAMETER["scale_factor",0.9996],PARAMETER["latitude_of_origin",
0.0],UNIT["Meter",1.0]],VERTCS["NAVD88 - Geoid03
(Meters)",VDATUM["unknown"],PARAMETER["Vertical_Shift",0.0],PARAMETER[
"Direction",1.0],UNIT["Meter",1.0]]',
    pixel_type="8_BIT_UNSIGNED",
    cellsize=None,
    number_of_bands=1,
    mosaic_method="LAST",
    mosaic_colormap_mode="FIRST"
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
arc1lab3\\interestarea1'>
```

```
# create dory's starting point

# create feature class

feature_class_path = arcpy.management.CreateFeatureclass(
    out_path=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb",
    out_name="dory_start",
    geometry_type="POINT",
    template=None,
    has_m="DISABLED",
    has_z="DISABLED",

spatial_reference='GEOGCS["GCS_North_American_1983",DATUM["D_North_Ame
rican_1983",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Gree
nwich",0.0],UNIT["Degree",0.0174532925199433]],VERTCS["NAVD88 -
Geoid03
(Meters)",VDATUM["unknown"],PARAMETER["Vertical_Shift",0.0],PARAMETER[
"Direction",1.0],UNIT["Meter",1.0]];-400 -400 1000000000;-100000
10000;-100000 10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision',
    config_keyword="",
    spatial_grid_1=0,
    spatial_grid_2=0,
    spatial_grid_3=0,
    out_alias=""
)

# create point

latitude = 44.127985
longitude = -92.148796

point = arcpy.Point(longitude, latitude)
```

```
# insert the point into the feature class

with arcpy.da.InsertCursor(feature_class_path, ["SHAPE@"]) as cursor:
    cursor.insertRow([point])

# create dory's destination

# create feature class

feature_class_path = arcpy.management.CreateFeatureclass(
    out_path=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb",
    out_name="dory_end",
    geometry_type="POINT",
    template=None,
    has_m="DISABLED",
    has_z="DISABLED",

spatial_reference='GEOGCS["GCS_North_American_1983",DATUM["D_North_Ame
rican_1983",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Gree
nwich",0.0],UNIT["Degree",0.0174532925199433]],VERTCS["NAVD88 -
Geoid03
(Meters)",VDATUM["unknown"],PARAMETER["Vertical_Shift",0.0],PARAMETER[
"Direction",1.0],UNIT["Meter",1.0]];-400 -400 1000000000;-100000
10000;-100000 10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision',
    config_keyword="",
    spatial_grid_1=0,
    spatial_grid_2=0,
    spatial_grid_3=0,
    out_alias=""
)

# create point

latitude = 44.0626553
longitude = -92.044557

point = arcpy.Point(longitude, latitude)

# insert the point into the feature class

with arcpy.da.InsertCursor(feature_class_path, ["SHAPE@"]) as cursor:
    cursor.insertRow([point])

# merge dory start and end

arcpy.management.Merge(
    inputs="dory_end;dory_start",
    output=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\lab3.gdb\
dory_points",
    field_mappings=None,
    add_source="NO_SOURCE_INFO"
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
lab3.gdb\\dory_points'>
```

```
# fill dem

out_surface_raster = arcpy.sa.Fill(
    in_surface_raster="interestarea1",
    z_limit=None
)
out_surface_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\arc1lab3\filled_dem")

# calculate slope

arcpy.ddd.Slope(
    in_raster="out_surface_raster",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\Slope_filled1",
    output_measurement="DEGREE",
    z_factor=1,
    method="PLANAR",
    z_unit="METER",
    analysis_target_device="GPU_THEN_CPU"
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
lab3.gdb\\Slope_filled1'>
```

```
# reclassify slope raster to integer classes

arcpy.ddd.Reclassify(
    in_raster="Slope_filled1",
    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",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\slope_class",
    missing_values="DATA"
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
lab3.gdb\\slope_class'>
```

```
# create cost-distance
# note: I tried doing this with the connect optimal regions tool and
```

```python
for whatever reason,
# running that tool instantly hard-crashes arcpro for me, like submit-
email-incident-report
# crashes immediately. im not sure why that is but im behind enough on
this and so i'm just
# doing it the way I did it in lab2. sorry :<

out_distance_raster = arcpy.sa.CostDistance(
    in_source_data="dory_start",
    in_cost_raster="slope_class",
    maximum_distance=None,
    out_backlink_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_distance_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostDis_dory1")

# create backlink raster

out_backlink_raster = arcpy.sa.CostBackLink(
    in_source_data="dory_start",
    in_cost_raster="slope_class",
    maximum_distance=None,
    out_distance_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_backlink_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostBac_dory1")

# conduct cost-path

out_raster = arcpy.sa.CostPath(
    in_destination_data="dory_end",
    in_cost_distance_raster="out_distance_raster",
    in_cost_backlink_raster="out_backlink_raster",
    path_type="EACH_CELL",
    destination_field="OBJECTID",
    force_flow_direction_convention="INPUT_RANGE"
)
out_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\dorypath1")
```

```
# turn land use polygon to raster

arcpy.conversion.PolygonToRaster(
    in_features="landcover_landsat_tm_19_Clip",
    value_field="XLUSE_CODE",
    out_rasterdataset=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\landcover_landsat_tm_19_Clip_PolygonToRaster",
    cell_assignment="CELL_CENTER",
    priority_field="NONE",
    cellsize=280,
    build_rat="BUILD"
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
lab3.gdb\\landcover_landsat_tm_19_Clip_PolygonToRaster'>
```

```
# reclassify landuse bc there are too many classes
# combine cultivated land class with farmsteads and rural residences

arcpy.ddd.Reclassify(
    in_raster="landcoverInterest.tif",
    reclass_field="XLUSE_CODE",
    remap="'Deciduous Forest' 1;Grassland 2;'Cultivated Land'
3;'Farmsteads and Rural Residences' 3;'Urban and Industrial' 4;'Other
Rural Developments' 5;'Grassland-Shrub-Tree (deciduous)' 6;Water
7;'Gravel Pits and Open Mines' 8;Wetlands 10;' ' 11",
    out_raster=r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\Reclass_landcover1",
    missing_values="DATA"
)
```

```
<Result 'C:\\Users\\18284\\Documents\\ArcGIS\\Projects\\lab3\\
lab3.gdb\\Reclass_landcover1'>
```

```
# weight the relevant variables appropriately
# this version weighs slope and landuse factors evenly, 50-50

out_raster = arcpy.sa.WeightedOverlay(
    in_weighted_overlay_table=r"('C:\Users\18284\Documents\ArcGIS\
Projects\lab3\lab3.gdb\Reclass_landcover1' 50 'Value' (1 9; 2 9; 3 1;
4 9; 5 2; 6 9; 7 3; 8 9; 10 9; 11 9; NODATA NODATA); 'C:\Users\18284\
Documents\ArcGIS\Projects\arc1lab2\arc1lab2.gdb\slope_class' 50
'Value' (1 9; 2 8; 3 7; 4 6; 5 5; 6 4; 7 3; 8 2; 9 1; 10 1; 11 1;
NODATA NODATA));1 9 1"
)
out_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\Weighte_Recl1")

# cost distance with weights v1

out_distance_raster = arcpy.sa.CostDistance(
    in_source_data="dory_start",
    in_cost_raster="out_raster",
    maximum_distance=None,
    out_backlink_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_distance_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostDis_dory2")

# cost backlink with weights v1

out_backlink_raster = arcpy.sa.CostBackLink(
    in_source_data="dory_start",
    in_cost_raster="out_raster",
    maximum_distance=None,
    out_distance_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_backlink_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostBac_dory2")

# cost_path weighted v1

out_raster = arcpy.sa.CostPath(
    in_destination_data="dory_end",
    in_cost_distance_raster="out_distance_raster",
    in_cost_backlink_raster="out_backlink_raster",
    path_type="EACH_CELL",
    destination_field="OBJECTID",
    force_flow_direction_convention="INPUT_RANGE"
)
out_raster.save(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\CostPat_dory1")

# weight the relevant variables appropriately
# this version weighs slope more heavily than land-use, 80-20

out_raster2 = arcpy.sa.WeightedOverlay(
    in_weighted_overlay_table=r"('C:\Users\18284\Documents\ArcGIS\
Projects\lab3\lab3.gdb\Reclass_landcover1' 80 'Value' (1 9; 2 9; 3 1;
4 9; 5 2; 6 9; 7 3; 8 9; 10 9; 11 9; NODATA NODATA); 'C:\Users\18284\
Documents\ArcGIS\Projects\arc1lab2\arc1lab2.gdb\slope_class' 20
'Value' (1 9; 2 8; 3 7; 4 6; 5 5; 6 4; 7 3; 8 2; 9 1; 10 1; 11 1;
NODATA NODATA));1 9 1"
)
out_raster2.save(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\Weighte_Recl2")

# cost distance with weights v2

out_distance_raster2 = arcpy.sa.CostDistance(
    in_source_data="dory_start",
    in_cost_raster="out_raster2",
    maximum_distance=None,
    out_backlink_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_distance_raster2.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostDis_dory3")

# cost backlink with weights v2

out_backlink_raster2 = arcpy.sa.CostBackLink(
    in_source_data="dory_start",
    in_cost_raster="out_raster2",
    maximum_distance=None,
    out_distance_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_backlink_raster2.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostBac_dory3")

# cost_path weighted v2

out_raster2 = arcpy.sa.CostPath(
    in_destination_data="dory_end",
    in_cost_distance_raster="out_distance_raster2",
    in_cost_backlink_raster="out_backlink_raster2",
    path_type="EACH_CELL",
    destination_field="OBJECTID",
    force_flow_direction_convention="INPUT_RANGE"
)
out_raster2.save(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\CostPat_dory2")
```

```
# weight the relevant variables appropriately
# this version weighs land-use more heavily than slope, 80-20

out_raster3 = arcpy.sa.WeightedOverlay(
    in_weighted_overlay_table=r"('C:\Users\18284\Documents\ArcGIS\
Projects\lab3\lab3.gdb\Reclass_landcover1' 20 'Value' (1 9; 2 9; 3 1;
4 9; 5 2; 6 9; 7 3; 8 9; 10 9; 11 9; NODATA NODATA); 'C:\Users\18284\
Documents\ArcGIS\Projects\arc1lab2\arc1lab2.gdb\slope_class' 80
'Value' (1 9; 2 8; 3 7; 4 6; 5 5; 6 4; 7 3; 8 2; 9 1; 10 1; 11 1;
NODATA NODATA));1 9 1"
)
out_raster3.save(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\Weighte_Recl3")

# cost distance with weights v3

out_distance_raster3 = arcpy.sa.CostDistance(
    in_source_data="dory_start",
    in_cost_raster="out_raster3",
    maximum_distance=None,
    out_backlink_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_distance_raster3.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostDis_dory4")

# cost backlink with weights v3

out_backlink_raster3 = arcpy.sa.CostBackLink(
    in_source_data="dory_start",
    in_cost_raster="out_raster3",
    maximum_distance=None,
    out_distance_raster=None,
    source_cost_multiplier=None,
    source_start_cost=None,
    source_resistance_rate=None,
    source_capacity=None,
    source_direction=""
)
out_backlink_raster3.save(r"C:\Users\18284\Documents\ArcGIS\Projects\
lab3\lab3.gdb\CostBac_dory4")

# cost_path weighted v3

out_raster3 = arcpy.sa.CostPath(
    in_destination_data="dory_end",
    in_cost_distance_raster="out_distance_raster3",
    in_cost_backlink_raster="out_backlink_raster3",
    path_type="EACH_CELL",
    destination_field="OBJECTID",
    force_flow_direction_convention="INPUT_RANGE"
)
out_raster3.save(r"C:\Users\18284\Documents\ArcGIS\Projects\lab3\
lab3.gdb\CostPat_dory3")
```