In [1]:

```
import arcpy
import requests
import io
import os
import zipfile
arcpy.env.workspace = r'G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb'
working_dir = r'G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb'
```

Data Aquirement

In [ ]:

```
#Land Use

#landuse_file = r'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_sta
te_dnr/biota_landcover_nlcd_mn_2019/tif_biota_landcover_nlcd_mn_2019.zip'
#landuse_actual.content = requests.post(landuse_file)
#landusezipfile = zipfile.ZipFile(io.BytesIO(landuse_post_request.content))
#landusezipfile.extractall(working_dir)
```

In [ ]:

```
#Zone Numbers

#zone_file = r'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_
dnr/bdry_deer_permit_areas/shp_bdry_deer_permit_areas.zip'
#zone_actual.content = requests.post(data_file)
#zonezipfile = zipfile.ZipFile(io.BytesIO(data_post_request.content))
#zonezipfile.extractall(working_dir)
```

In [ ]:

```
#CWD Data

#There is no easy way to download the data as it is just a JPEG. The JPEG come
s from a report which is a PDF so no help there either. I made a excel sheet a
nd made a column for zone number and for CWD 1=no 0=yes
```

In [ ]:

```
#Harvest Data

#harvest_file = r'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_sta
te_dnr/env_mn_deer_harvest/shp_env_mn_deer_harvest.zip'
#harvest_actual.content = requests.post(data_file)
#harvestzipfile = zipfile.ZipFile(io.BytesIO(data_post_request.content))
#harvestzipfile.extractall(working_dir)
```

In [ ]:

```
#DEM

#DEM_file = r'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_d
nr/elev_30m_digital_elevation_model/fgdb_elev_30m_digital_elevation_model.zip'
#DEM_actual = requests.post(DEM_file)
#DEMzipfile = zipfile.ZipFile(io.BytesIO(DEM_post_request.content))
#DEMzipfile.extractall(working_dir)
```

Clip Landuse only

In [ ]:

```
#arcpy.management.Clip("NLCD_2019_Land_Cover_Change_Index.tif", "189783.56 4816309.33 761653.52 5472346.4998", r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\Land_Cover_Clip", "Zones", "255", "NONE", "NO_MAINTAIN_EXTENT")
```

Transform/Raster/Join (CWD and Harvest)

In [ ]:

```
#Transform Code
#arcpy.management.ConvertCoordinateNotation("CWD", r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\CWD_ConvertCoordinateNota", "UNIQUE_ID", "BKWYNAME", "DD_2", "DD_2", None, 'GEOGCS["NAD_1983_UTM_Zone_15N",DATUM["NAD_1983_UTM_Zone_15N",SPHEROID["GRS_1980",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]];-400 -400 1000000000;-100000 10000;-100000 10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision', '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],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-93.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]]', "INCLUDE_INVALID")
```

In [ ]:

```
#Create Raster CWD
#arcpy.management.CreateRasterDataset(r"G:\My Drive\GIS 5571\Final\FinalP", "CWD_raster", None, "8_BIT_UNSIGNED", None, 1, '', "PYRAMIDS -1 NEAREST DEFAULT 75 NO_SKIP NO_SIPS", "128 128", "LZ77", None)
```

In [ ]:

```
#Join Field
#inFeatures = "CWD_CSV"
#joinField = "Disease"
#joinTable = "CWD_final"
#arcpy.management.JoinField(inFeatures, joinField, joinTable, joinField)
```

In [ ]:

```
#Create Raster Harvest
#arcpy.management.CreateRasterDataset(r"G:\My Drive\GIS 5571\Final\FinalP", "harvest_raster", None, "8_BIT_UNSIGNED", None, 1, '', "PYRAMIDS -1 NEAREST DEFAULT 75 NO_SKIP NO_SIPS", "128 128", "LZ77", None)
```

In [ ]:

```
#Join Field

#inFeatures = "Harvest"
#joinField = "Success"
#joinTable = "Harvest_final"
#arcpy.management.JoinField(inFeatures, joinField, joinTable, joinField)

#no need to standardize because Success is a percentage 0-.99
```

DEM to Slope

In [ ]:

```
#Convert DEM to Slope
#out_raster = arcpy.sa.Slope("DEM", "DEGREE", 1, "PLANAR", "METER"); out_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP\Slope")
```

In [ ]:

```
#Standardize
#output_raster = arcpy.ia.RasterCalculator(' "Slope" /90'); output_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\Slope")
```

Raster Calculator

In [ ]:

```
#Raster Calculator for landcover
#output_raster = arcpy.ia.RasterCalculator(' ("landcover"==0)  | ( "landcover"==11)  | ("landcover"==24)  | ( "landcover"==21)  |( "landcover"==22)  | ( "landcover"==23) |( "landcover"==90)'); output_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\landcover_final")
# 0 is undefined, 11 is open water, 21-24 are developed and 90 is emerging wet lands, none of those types would be good for hunting
```

In [ ]:

```
#Raster Calculator for CWD
#output_raster = arcpy.ia.RasterCalculator('("CWD_final"==1)'); output_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\CWD")
#This is only selecting zones with no CWD presence
```

Creating Cost Surface

In [ ]:

```
#Day
#output_raster = arcpy.ia.RasterCalculator('("slope_standard") + 3*(success) + 1.5*( "landcover_final") + 2*(CWD)'); output_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\daycostsurface")
```

In [ ]:

```
#Night
#output_raster = arcpy.ia.RasterCalculator('(-"slope_standard") + 3*(success) + 1.5*( "landcover") + 2*(CWD)'); output_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\nightcostsurface")
```

In [ ]:

```
#Standardize
#output_raster = arcpy.ia.RasterCalculator(' "daycostsurface" /12.2'); output_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\daycost_standard")
#output_raster = arcpy.ia.RasterCalculator(' "nightcostsurface" /7.9'); output_raster.save(r"G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\nightcost_standard")
```

Zonal Statistics (bond the index to the zones)

In [ ]:

```
#Zonal Statistics
#inZoneData = "zones.shp"
#zoneField = "Zone_ID"
#inValueRaster = "daycost_surface.tif"

#outZonalStatistics = ZonalStatistics(inZoneData, zoneField, inValueRaster)

#outZonalStatistics.save("G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\DayFinal.tif")
```

In [ ]:

```
#inZoneData = "zones.shp"
#zoneField = "Zone_ID"
#inValueRaster = "nightcost_surface.tif"

#outZonalStatistics = ZonalStatistics(inZoneData, zoneField, inValueRaster)

#outZonalStatistics.save("G:\My Drive\GIS 5571\Final\FinalP\FinalP.gdb\NightFinal.tif")
```

Accuracy Assesment

In [ ]:

```
#Create Accuracy Assessment Points
#arcpy.gp.CreateAccuracyAssessmentPoints("Day_Final.tif", "AccuracyPoints.shp", "65", "RANDOM")
```

In [ ]:

```
#Confusion Matrix
#arcpy.gp.ComputeConfusionMatrix("AccuracyPoints.shp", "CMatrix.dbf")
```

Exporting

In [ ]:

```
#Export Layout as PDF
#aprx= arcpy.mpArcGISProject(r"G:\My Drive\GIS 5571\Final\FinalP.aprx")
#lyt = aprx.listLayouts("Index_Map*")[0]
#lyt.exportToPDF(r"G:\My Drive\GIS 5571\Final\FinalP\Output\EveningIndex_Map.pdf" , resolution = 300)
```

In [ ]:

```
#aprx= arcpy.mpArcGISProject(r"G:\My Drive\GIS 5571\Final\FinalP.aprx")
#lyt = aprx.listLayouts("Index_Map*")[0]
#lyt.exportToPDF(r"G:\My Drive\GIS 5571\Final\FinalP\Output\MorningIndex_Map.pdf" , resolution = 300)
```

In [ ]:

```
#Export to Webmap
#arcpy.server.ExportWebMap(FinalP_as_JSON, "G:\My Drive\GIS 5571\Final\FinalP_webmap.json")
```