```
import logging
import arcpy
logging.basicConfig(format='%(asctime)s\t\t%(message)s',
level=logging.DEBUG)
#
# Here you should define the path of the geodatabase containing all
the layers
#
path=r"C:
\Users\Dimitris\Desktop\OnSSET\AFG_GIS_10km\Afghanistan.gdb"
path1=r"C:
\Users\Dimitris\Desktop\OnSSET\AFG_GIS_10km\Assistingfolder"
outpath = r"C:\Users\Dimitris\Desktop\OnSSET"
#
arcpy.env.workspace = path
arcpy.env.overwriteOutput = True
arcpy.env.addOutputsToMap = False
arcpy.CheckOutExtension("Spatial")
#
# The variables needed
#
SET_COUNTRY = 'Country'  # This cannot be changed, lots of code will
break
SET_X = 'X'  # Coordinate in kilometres
SET_Y = 'Y'  # Coordinate in kilometres
SET_X_DEG = 'X_deg'  # Coordinates in degrees
SET_Y_DEG = 'Y_deg'
SET_POP = 'Pop'  # Population in people per point (equally, people
per 100km2)
SET_POP_CALIB = 'PopStartCalibrated'  # Calibrated population to
reference year, same units
SET_POP_FUTURE = 'PopFuture'  # Project future population, same
units
SET_GRID_DIST_CURRENT = 'GridDistCurrent'  # Distance in km from
current grid
SET_GRID_DIST_PLANNED = 'GridDistPlan'  # Distance in km from
current and future grid
SET_ROAD_DIST = 'RoadDist'  # Distance in km from road network
SET_NIGHT_LIGHTS = 'NightLights'  # Intensity of night time lights
(from NASA), range 0 - 63
SET_TRAVEL_HOURS = 'TravelHours'  # Travel time to large city in
hours
SET_GHI = 'GHI'  # Global horizontal irradiance in kWh/m2/day
SET_WINDVEL = 'WindVel'  # Wind velocity in m/s
SET_WINDCF = 'WindCF'  # Wind capacity factor as percentage (range 0
- 1)
SET_HYDRO = 'power'  # Hydropower potential in kW
SET_HYDRO_DIST = 'HydropowerDist'  # Distance to hydropower site in
km
SET_HYDRO_FID = 'HydropowerFID'  # the unique tag for eah
hydropower, to not over-utilise
SET_SUBSTATION_DIST = 'SubstationDist'
SET_ELEVATION = 'Elevation'  # in metres
SET_SLOPE = 'Slope'  # in degrees
```

```
SET_LAND_COVER = 'LandCover'
SET_SOLAR_RESTRICTION = 'SolarRestriction'
#
# Here are the layers in the geodatabase. Make sure that the naming
convection is the same as it appears on ArcGIS
#
pop = 'pop2015'  # Type: raster, Unit: people per 100km2, must be in
resolution 10km x 10km
ghi = 'ghi'  # Type: raster, Unit: kWh/m2/day
windvel = 'windvel'  # Type: raster, Unit: capacity factor as a
percentage (range 0 - 1)
travel = 'traveltime'  # Type: raster, Unit: hours
grid_existing = 'existing_grid'  # Type: shapefile (line)
grid_planned = 'planned_grid'  # Type: shapefile (line)
hydro_points = 'hydro_points'  # Type: shapefile (points), Unit: kW
(field must be named Hydropower)
admin_raster = 'admin_0'  # Type: raster, country names must conform
to specs.xlsx file
admin1_raster = 'admin_1'  # Type: raster, country names must
conform to specs.xlsx file
roads = 'completedroads'  # Type: shapefile (lines)
nightlights = 'nightlights'  # Type: raster, Unit: (range 0 - 63)
substations = 'allsubstations'
elevation = 'elevation'
slope = 'slope'
land_cover = 'landcover'
solar_restriction = 'solar_restrictions'
settlements_fc = 'Afghanistan10km'  # Here you can select the name
of the feature class that will aggregate all the results
##
## All the commands that are together (no gaps inbetween) can be
executed together.
## Depending on computational cababilities more commands can be
executed together.

arcpy.RasterToPoint_conversion(pop, settlements_fc)
arcpy.AlterField_management(settlements_fc, 'grid_code', SET_POP)

arcpy.AddXY_management(settlements_fc)
arcpy.AddField_management(settlements_fc, SET_X, 'FLOAT')
arcpy.CalculateField_management(settlements_fc, SET_X, '!POINT_X! /
1000', 'PYTHON_9.3')
arcpy.AddField_management(settlements_fc, SET_Y, 'FLOAT')
arcpy.CalculateField_management(settlements_fc, SET_Y, '!POINT_Y! /
1000', 'PYTHON_9.3')
arcpy.DeleteField_management(settlements_fc, 'POINT_X; POINT_Y')

arcpy.sa.ExtractMultiValuesToPoints(settlements_fc,
[[solar_restriction, SET_SOLAR_RESTRICTION]])

arcpy.sa.ExtractMultiValuesToPoints(settlements_fc, [[travel,
SET_TRAVEL_HOURS]])

arcpy.sa.ExtractMultiValuesToPoints(settlements_fc, [[nightlights,
SET_NIGHT_LIGHTS]])

arcpy.sa.ExtractMultiValuesToPoints(settlements_fc, [[elevation,
SET_ELEVATION]])

arcpy.sa.ExtractMultiValuesToPoints(settlements_fc, [[slope,
SET_SLOPE]])

arcpy.sa.ExtractMultiValuesToPoints(settlements_fc, [[land_cover,
SET_LAND_COVER]])

arcpy.Near_analysis(settlements_fc, grid_existing)
arcpy.AddField_management(settlements_fc, SET_GRID_DIST_CURRENT,
'FLOAT')
arcpy.CalculateField_management(settlements_fc,
SET_GRID_DIST_CURRENT, '!NEAR_DIST! / 1000', 'PYTHON_9.3')
arcpy.DeleteField_management(settlements_fc, 'NEAR_DIST; NEAR_FID')

arcpy.Near_analysis(settlements_fc, [grid_existing, grid_planned])
arcpy.AddField_management(settlements_fc, SET_GRID_DIST_PLANNED,
'FLOAT')
arcpy.CalculateField_management(settlements_fc,
SET_GRID_DIST_PLANNED, '!NEAR_DIST! / 1000', 'PYTHON_9.3')
arcpy.DeleteField_management(settlements_fc, 'NEAR_DIST; NEAR_FID;
NEAR_FC')

arcpy.Near_analysis(settlements_fc, substations)
arcpy.AddField_management(settlements_fc, SET_SUBSTATION_DIST,
'FLOAT')
arcpy.CalculateField_management(settlements_fc, SET_SUBSTATION_DIST,
'!NEAR_DIST! / 1000', 'PYTHON_9.3')
arcpy.DeleteField_management(settlements_fc, 'NEAR_DIST; NEAR_FID;
NEAR_FC')

arcpy.Near_analysis(settlements_fc, roads)
arcpy.AddField_management(settlements_fc, SET_ROAD_DIST, 'FLOAT')
arcpy.CalculateField_management(settlements_fc, SET_ROAD_DIST, '!
NEAR_DIST! / 1000', 'PYTHON_9.3')
arcpy.DeleteField_management(settlements_fc, 'NEAR_DIST; NEAR_FID')

arcpy.Near_analysis(settlements_fc, hydro_points)
arcpy.AddField_management(settlements_fc, SET_HYDRO_DIST, 'FLOAT')
arcpy.CalculateField_management(settlements_fc, SET_HYDRO_DIST, '!
NEAR_DIST! / 1000', 'PYTHON_9.3')
arcpy.JoinField_management(settlements_fc, 'NEAR_FID', hydro_points,
arcpy.Describe(hydro_points).OIDFieldName, [SET_HYDRO])
arcpy.AlterField_management(settlements_fc, 'NEAR_FID',
SET_HYDRO_FID, SET_HYDRO_FID)
arcpy.DeleteField_management(settlements_fc, 'NEAR_DIST')

# Here the process changes due to some peculiarities of the
following datasets
path2=path1+"\Afghanistan"
path3=path1+"\Afghanistan_Provinces"
```

```
path4=path1+"\GlobalHI"
path5=path1+"\WIND"

arcpy.sa.ExtractValuesToPoints(settlements_fc,admin_raster,path2,"NONE", "ALL")
arcpy.sa.ExtractValuesToPoints(settlements_fc,admin1_raster,path3,"NONE", "ALL")
arcpy.sa.ExtractValuesToPoints(settlements_fc,ghi,path4,"INTERPOLATE","VALUE_ONLY")
arcpy.sa.ExtractValuesToPoints(settlements_fc,windvel,path5,"INTERPOLATE","VALUE_ONLY")

arcpy.env.workspace = path1
in_features = ['WIND.shp', 'GlobalHI.shp', 'Afghanistan.shp', 'Afghanistan_Provinces.shp']
out_location = path
arcpy.FeatureClassToGeodatabase_conversion(in_features, out_location)

arcpy.env.workspace = path

arcpy.JoinField_management(settlements_fc,"pointid","WIND","pointid","RASTERVALU")

arcpy.JoinField_management(settlements_fc,"pointid","GlobalHI","pointid","RASTERVALU")

arcpy.JoinField_management(settlements_fc,"pointid","Afghanistan","pointid","CNTRY_NAME")

arcpy.JoinField_management(settlements_fc,"pointid","Afghanistan_Provinces","pointid","Prov_Name")

# Final command extracting the settlements file into your specified outpath (given in the beggining) under the given name (here AfghanistanSett10k).
arcpy.TableToTable_conversion(settlements_fc,outpath,"AfghanistanSett10k")
```