

RANKING HAZARDOUS WASTE GENERATORS IN THE BAY AREA OF CALIFORNIA BY VULNERABILITY TO EARTHQUAKE INDUCED HAZARDS

GEOGRAPHY 375: INTRO PYTHON FOR GIS

AMERICAN RIVER COLLEGE: SPRING 2019

KARRI PETERS

SUMMARY

According to the California Geological Survey website, “California is the most geologically diverse and the second most seismically active state in the USA”. The Bay Area is a densely populated and ecologically valuable region of the state that is vulnerable to natural disasters such as liquefaction and landslides triggered by earthquakes. Numerous hazardous waste generator sites are located in the Bay Area and present an elevated risk to human health and the environment should they be impacted by a natural disaster. By ranking each hazardous waste generator site by their potential to be affected by one or more natural disaster, emergency services can streamline their mitigation efforts.

One way to determine the potential hazard ranking for each site is to perform geospatial analysis using a standalone python script. I used datasets for the hazardous waste generators as well as for the potential risk of landslides and liquefaction to perform the analysis. The processes I used in the script include clipping feature classes, creating layers, selecting features by location, creating new feature classes, joining feature classes, adding new fields, performing update cursors, and populating fields using if statements with for loops. The script included modules, variable definitions, try and except blocks, and notes describing the analysis process. The output waste generator site feature class includes a field ranking the potential hazard level to each site during an earthquake induced natural disaster with values of Low, Medium, or High risk.

PURPOSE

The intention of this script is to create a hazard level ranking for hazardous waste generator sites within the Bay Area. The script is written so it can be updated with minimal effort as datasets are updated or new data becomes available.

METHODS

The first step in the process of writing the script was to obtain quality datasets that covered my study area. I downloaded the hazardous waste generator sites, landslide hazard, liquefaction hazard, and Bay Area outline feature classes from multiple sources. Once I had the feature classes I needed, I projected them into the same coordinate system in ArcMap and cleaned up unnecessary data from the Bay Area Boundary dataset, such as islands I didn't want included in the study area. I dissolved the boundaries on the landslide and liquefaction feature classes using the potential risk ranking fields. The landslide and liquefaction sites contained fields that ranked

the level of potential risk for each polygon. For these feature classes I added a new field and reclassified the risk rankings to a numerical scale beginning with 0 and ascending until a maximum rank was reached. This was done to normalize the risk fields and make classifying each waste generator a simpler process in future steps. Once the data was in the most manageable form, I began writing the geoprocessing script.

Using Python's IDLE version 2.7.14, I began by filling out descriptive notes and text outlining the purpose and general steps of the project. Next, I imported various modules including `arcpy`, `sys`, and `traceback` as well as set the workspace, outpath, and variables. I then added the try and except blocks including the traceback error dialog that was obtained through the exercises in the Python Primer Workbooks.

In the try block, I began the first steps of analysis by clipping the landslide and liquefaction feature classes to the bay area outline feature classes. Print statements were added throughout the whole script to indicate when a geoprocess was complete. Once the natural hazard feature classes were clipped, I created a feature class for hazardous waste generators that are only in the Bay Area.

I began by writing the code to create feature layers from the hazardous waste generator and Bay Area outline feature classes while checking to see if they exist, and if they do to delete them. Once the feature layers were created, I selected the sites from the waste generators layer that fell within the polygon of the Bay Area outline layer. Using the get count management code, I printed a statement reflecting the total number of waste generator sites within the Bay Area. Finally, I used the copy features script to export the selected features to a new feature class, once again checking to see if the feature class already existed and deleting it if it did.

Next, I wanted each hazardous waste generator site to contain information about the landslide and liquefaction risk within its attribute table, so that the potential risk values could be combined into an overall hazard risk level field. To do this I wrote a succession of spatial joins, first for the landslide and then for the liquefaction feature class, getting an output dataset that contained hazard risk fields from each natural disaster feature class. Using the add field data management tool, I added a long integer field to be populated with the sum of the original risk values, and a text field to translate the summed values to a final hazard level.

To populate the summed risk value field, I defined a python list containing the three fields in the waste generator feature class involved in the equation. To iterate through each row, I used a for loop within an update cursor where each value from the landslide risk field was added to the liquefaction risk field and the output was entered into the hazard rank sum field. The rows and cursors were deleted so the next code in the script would not refer back to this step. Following a similar approach, I populated the final text field indicating hazard risk level for each waste generator site. Using a python list, an update cursor, and a for loop, I assigned a text value based on the numerical values from the risk sum field. Fields with the value of 0 to 2 were given a Low risk rank, values of 3 to 4 were given a Medium risk rank, and values of 5 to 6 were given a High

risk rank. The rows and cursors were again deleted, and final print statements were written to indicate the end of the script processing.

RESULTS

The final output feature class consisted of points representing locations of hazardous waste generators in the Bay Area, with a field containing risk potential in the event of an earthquake induced natural disaster ranked low, medium and high. Difficulties writing this script were encountered at almost each step of the process. Finding quality data that covered the area of interest was difficult to come by in a downloadable format so analysis could be performed. The U.S. Geological Survey and the California Geological Survey both have geospatial data on earthquake hazards, however they are mainly shared as feature services or maps that were problematic to extract the data from. When searching for solutions of Geographic Information Systems (GIS) forums such as ESRI's GeoNet or stackexchange.com, the solutions offered included using programming skills beyond the scope of this course.

A challenge I encountered in writing this script was knowing at which point it was necessary to create and feature layer and when it was not. The work flow I would use in a GIS application did not necessarily align with what was required in the script. Multiple times I rearranged the processes in the python script, changing variable definitions when necessary, so that the final script made logical sense. I also learned that some data management tools, such as clip, do not require a feature layer to operate.

The largest issue I encountered involved populating the final text hazard level field based on the values of the risk sum field. I understood the process of using Boolean logic with if statements from using the field calculator in ArcMap, however translating this logic to a standalone script was more trying than I expected. Originally, I had tried to use a reclass script that I had discovered on a GIS forum, but found that the return function was not operable within an update cursor with a for loop. After spending some time looking into break and continue commands, I broke down the logic of the if statement code and was able to find a solution that didn't produce any errors and created the desired feature class field values.

CODE REFERENCES

Jennings, Nathan. *A Python Primer for ArcGIS Workbook II*. 2015. Digital.

Jennings, Nathan. *A Python Primer for ArcGIS Workbook III*. 2015. Digital.

DATA SOURCES

Association of Bay Area Governments Resilience Program. Accessed May 2019. Open data portal website: <http://resilience.abag.ca.gov/open-data/>

CalEPA_DTSC Feature Layer. Accessed May 2019. ESRI ArcGIS Online Feature Service: https://services3.arcgis.com/Oy2JTCD10wkoelxS/arcgis/rest/services/CalEPA_DTSC/FeatureServer

University of California, Berkeley Library. Accessed May 2019. Geodata portal website: <https://geodata.lib.berkeley.edu/catalog/ark28722-s7d02x>