**Project Report: Elevation Extraction and Integration with Google Earth Engine**

**1. Project Overview**

This project focuses on extracting elevation data for specified geographic points from the USGS 3DEP 10m National Map Seamless dataset using Google Earth Engine (GEE). The elevation values are integrated into a shapefile, providing an efficient and automated method for spatial analysis. Additionally, an ArcGIS toolbox was developed to create a user-friendly interface for implementing this workflow.

**2. Objectives**

* Extract elevation values from the USGS 3DEP dataset for coordinates provided in a CSV file.
* Write the extracted elevation values to a shapefile.
* Develop an ArcGIS Toolbox with a user interface for easier implementation.
* Visualize data in ArcGIS Pro with a map layout, symbology, and export functionality.

**3. Tools and Technologies Used**

* **Google Earth Engine API**: For retrieving elevation data from the USGS 3DEP dataset.
* **Python Libraries**: arcpy, pandas, os, and ee for data manipulation and spatial operations.
* **ArcGIS Pro**: To create the final map, visualize data, and generate a custom toolbox.
* **ArcGIS Toolbox Script**: Implemented as a parameterized script for ease of use.

**4. Data Description**

* **Input Data**:
  + boundary.csv: A CSV file containing four columns (row, column, X, and Y) with geographic coordinates for water boundaries.
* **Output Data**:
  + A shapefile (e.g., pnt\_elev4.shp) containing the original points along with their respective elevation values.

**5. Development Process**

**Step 1: Initial Development in Jupyter Notebook**

The workflow was initially developed in a Python notebook to ensure proper data processing. Key functionalities included:

* Reading CSV input data using pandas.
* Sampling elevation data from the USGS 3DEP dataset using GEE’s sampleRegions function.
* Writing the extracted elevation values into a shapefile using ArcPy.

**Step 2: Conversion to Python Script**

The notebook code was converted into a standalone Python script (project2.py). The script includes:

* **Modular functions**: For easier reusability (e.g., getGeeElevation function).
* **Command-line arguments**: To allow dynamic input of workspace, CSV file, shapefile name, and EPSG codes.

**Step 3: Integration into an ArcGIS Toolbox**

Using ArcGIS Pro, a custom geoprocessing tool was created. The tool interface allows users to:

* Select a workspace folder.
* Specify input CSV and output shapefile names.
* Set the EPSG code for the spatial reference

**7. Instructions for Use**

1. **Set Up Environment**:
   * Install required Python libraries (arcpy, pandas, ee).
   * Authenticate with GEE.
2. **Run the Tool**:
   * Open ArcGIS Pro and add the custom toolbox.
   * Provide inputs (workspace, CSV file, shapefile name, EPSG code) via the tool interface.
   * Click "Run" to execute the process.
3. **Access the Code**:
   * The Python script and other project files are available in the linked GitHub repository.

**8. Conclusion**

This project successfully integrates Python scripting, GEE, and ArcGIS to automate elevation data extraction and visualization. The developed ArcGIS toolbox enhances usability, making the process accessible to non-programmers.