

APEC 8601: Natural Resource Economics

Problem Set 4

Matt Braaksma

Table of contents

InVEST Model: Scenic Quality	1
1 Description	1
2 Calculations	1
3 Data Requirements	2
4 Output	2
5 Sensitivity Analysis	3
Code Appendix	3

InVEST Model: Scenic Quality

1 Description

Scenic quality plays an important role in many communities by attracting tourism, increasing property values, and generally providing utility. The InVEST Scenic Quality model focuses on coastal areas to generate viewsheds, which show how offshore development may impact visibility. The beauty of the natural landscape is important to both residents and visitors to certain areas. This model helps to predict how development might impact this visual amenity.

2 Calculations

The model calculates the visibility of each structure point based on the Wang et al reference-plane viewshed algorithm and creates a 0/1 raster. Next, the visibility rasters are weighted and summed to generate a weighted sum of the number of all visible points. It also provides a raster composed of quartiles

representing visual quality. If the optional valuation function and parameters are provided, the model can determine the visibility utility or disutility and create a weighted aggregate raster.

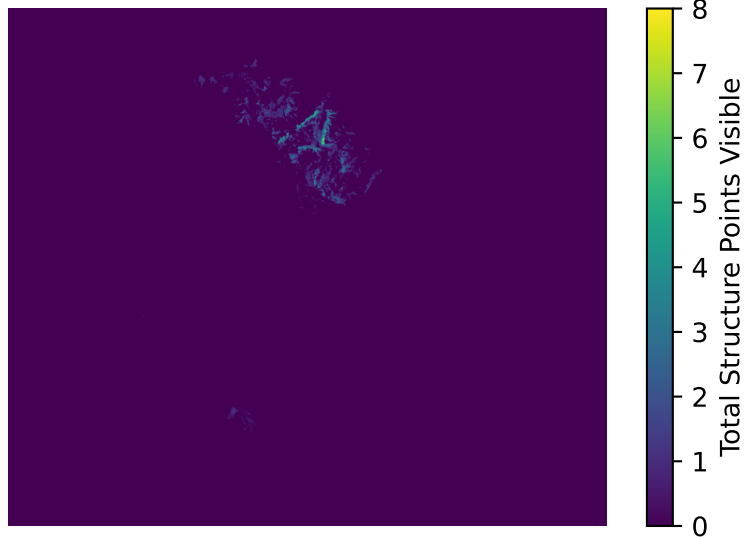
3 Data Requirements

- Area of Interest (vector)
 - Polygon of area over which to aggregate
- Features Impacting Scenic Quality (vector)
 - Locations of objects that negatively impact scenic quality
 - Optional fields include:
 - * radius (meters): maximum line of sight from a structure
 - * weight (default=1): viewshed importance coefficient
 - * height (meters): viewpoint elevation
- Digital Elevation Map (raster)
- Refractivity Coefficient (ratio)
 - Correct for curvature of the earth and refraction of visible light in the air
 - Default value is 0.13

4 Output

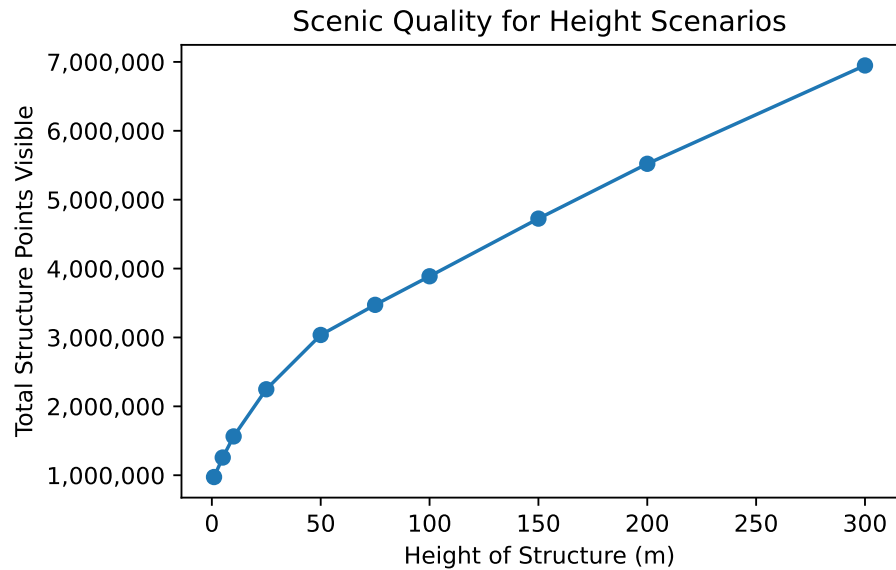
Results using sample InVEST data.

Viewshed for Sample Data (Claybark)



5 Sensitivity Analysis

The sample data uses the default height of zero. The following plot shows how the total number of structure points visible across the AOI differs by the height of the structures (uniformly across the points).



Code Appendix

```
# Model: Scenic Quality

import logging
import sys
import geopandas as gpd
import os
import pygeoprocessing as pygeo
import numpy as np

import natcap.invest.scenic_quality.scenic_quality
import natcap.invest.utils

LOGGER = logging.getLogger(__name__)
root_logger = logging.getLogger()

handler = logging.StreamHandler(sys.stdout)
formatter = logging.Formatter(
```

```

        fmt=natcap.invest.utils.LOG_FMT,
        datefmt='%m/%d/%Y %H:%M:%S ')
handler.setFormatter(formatter)
logging.basicConfig(level=logging.INFO, handlers=[handler])

args = {
    'a_coef': '',
    'aoi_path': '/Users/mbraaksma/Files/base_data/invest_sample_data/ScenicQuality/Input/AOI',
    'b_coef': '',
    'dem_path': '/Users/mbraaksma/Files/base_data/invest_sample_data/ScenicQuality/Input/cla',
    'do_valuation': False,
    'max_valuation_radius': '',
    'n_workers': '-1',
    'refraction': '0.13',
    'results_suffix': '',
    'structure_path': '/Users/mbraaksma/Files/base_data/invest_sample_data/ScenicQuality/Inp',
    'valuation_function': 'linear',
    'workspace_dir': '/Users/mbraaksma/Files/base_data/invest_sample_data/ScenicQuality/Outp
}

# Create 5 different versions of the AquaWEM_points shapefile
# Add varying height variable
scenario_list = [1, 5, 10, 25, 50, 75, 100, 150, 200, 300]
os.chdir(os.path.dirname(__file__))
for scenario in scenario_list:
    # Read in shapefile
    gdf = gpd.read_file('Input/AquaWEM_points.shp')
    # Generate height variable
    gdf['height'] = scenario
    # Export the modified GeoDataFrame to a GPKG shapefile
    gdf.to_file(f'Input/AquaWEM_points_height{scenario}.gpkg', driver='GPKG')

if __name__ == '__main__':

    # Run scenarios
    for scenario in scenario_list:
        args['structure_path'] = f'Input/AquaWEM_points_height{scenario}.gpkg'

        # Create output directory if it doesn't exist
        output_dir = f'output_scenario_height{scenario}'
        os.makedirs(output_dir, exist_ok=True)

        args['workspace_dir']= output_dir

```

```

print(args)
natcap.invest.scenic_quality.scenic_quality.execute(args)

# Calculate and print total visibility
for scenario in scenario_list:
    output_dir = f'output_scenario_height{scenario}/output'
    raster_path = output_dir + '/vshed.tif'
    # Import as array
    visibility_array = pygeo.raster_to_numpy_array(raster_path)
    # Remove missing values (-1) and sum
    total_structure_visibility = np.where(visibility_array==-1, 0, visibility_array).sum()
    print('Total Structure Points Visible for HEIGHT=', scenario, ':', total_structure_v

```