Feedback Form

School of Geography FACULTY OF ENVIRONMENT UNIVERSITY OF LEEDS *Students to complete all of the following (or work will not be accepted)*	Name (optional): Richard Thomas
Student ID: **200818341**	Assessment Name (e.g. Essay 2, Group Project etc): Assignment #1: Uncertainty in Sea Level Rise
Degree Programme & Level (e.g. BA1 Geog): MSc GIS	Assessment Marker: Steve Carver
Module Title & Code: GEOG5060M: GIS & Environment	Word Count: (750 words max) **735**

Content, research and reading

Structure and argument

Writing, presentation and referencing

Areas for improvement to prioritise

Marker's signature:	Date:	Provisional mark:

Any mark given here is provisional and subject to moderation by the School's Board of Examiners. This is to ensure comparable marking standards for all students. In a minority of individual cases moderation can lead to either the raising or lowering of the provisional marks.

Modelling uncertainty in climate-induced sea level rise

INTRODUCTION

With IPCC (2007) predictions of a rise in sea levels of up to 89cm by the end of the 21st century, there is a strong incentive to model the vulnerability of coastal areas (particularly populated ones) to flooding and sea-borne damage. Traditional flooding models have focussed on short duration bursts and a dependency on slope, whereas flooding due to sea level rise is predominantly dependent on elevation (Poulter and Halpin, 2008). This paper will address this by estimating (using Monte Carlo (MC) simulation methods) uncertainty in flooding location due to sea-level rise in an area of the Norfolk coast, based on 3 different Digital Elevation Models (DEMs):

Table 1: Comparison of Digital Elevation Models (DEMs) used

Digital Elevation Model (DEM)	Cell Size (m)	Height Resolution (m)	Height RMS Error (m)
OS Landform Panorama	50	1.0	5.0
OS Landform Profile	10	0.1	2.5
Environment Agency LiDAR	2	0.001	0.15

The bounding values of these 2007 IPCC sea-level rise predictions (11cm – 43cm) together with the 89cm possibility are shown (fig 1) extracted from these 3 DEMs and superposed over a base-map. Note that as the Panorama DEM has only 1.0m vertical resolution, only a +1m band is shown (close to 0.89cm). It is immediately apparent that as well as vertical resolution limitations, the large cell size of the Panorama DEM is poorly suited to mapping the relatively narrow river channel, whereas LiDAR DEM even highlights low-lying marshes.

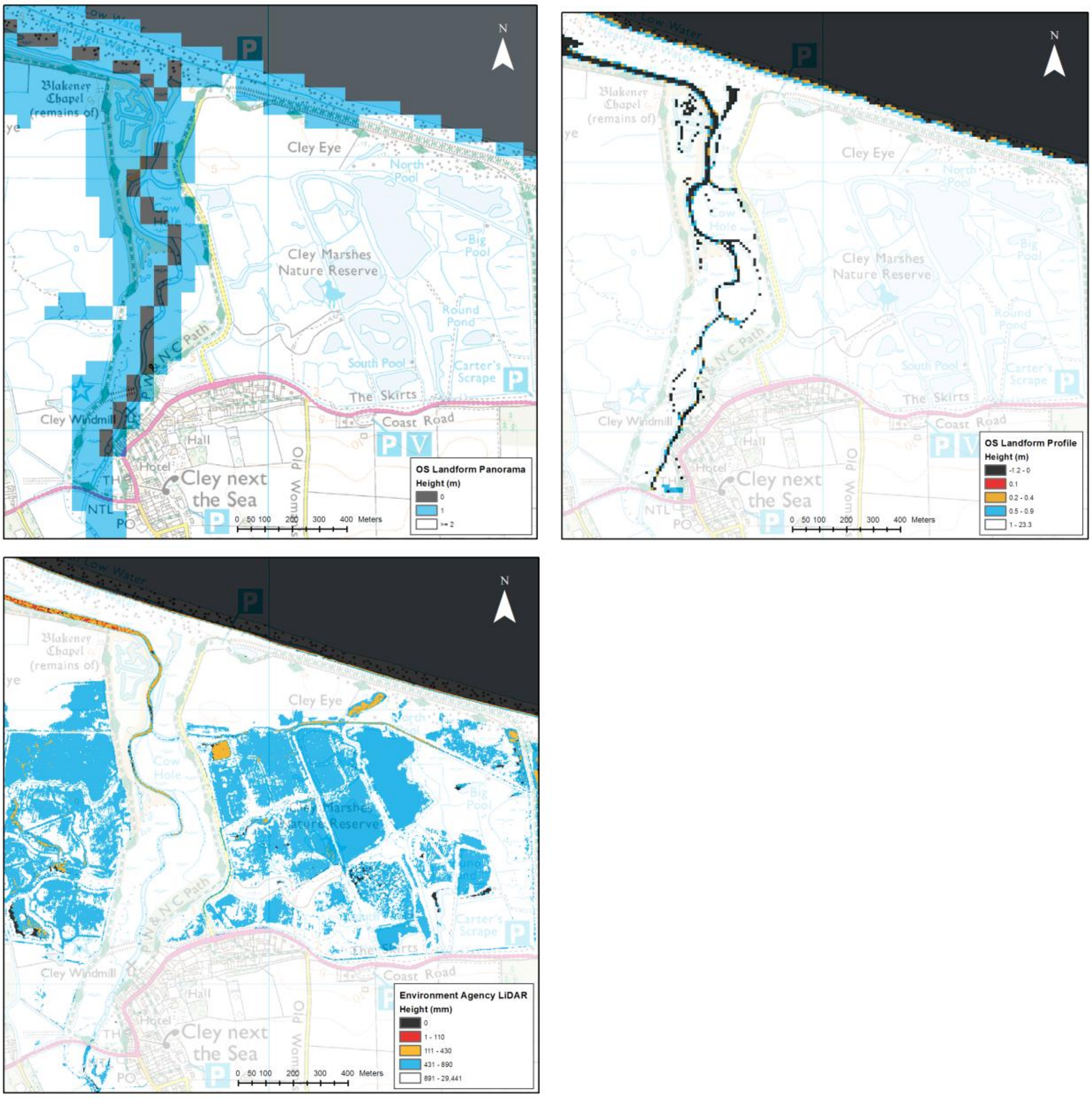

Figure 1: Heights of predicted sea levels on 3 types of DEM (base-maps Crown Copyright)

MONTE CARLO SIMULATION METHODOLOGY

Because of the complexity of the shape of the coastal landscape and the gentle gradients, it is difficult to directly estimate how sensitive the range of areas flooded would be to DEM error ranges. Using the elevation RMS error data, MC simulations for each sea level rise, for each DEM were modelled in ArcGIS using a Python script (see Appendix). Looking at the distribution of DEM height estimates over many (100) iterations gives a visual indication of the likelihood of areas to be inundated.

For this example, MC simulation is effective within reasonable computation limits. However, if multiple variables were varied, this quickly becomes prohibitive (2 variables each iterated randomly 100 times would require 10,000 simulations). Such raster calculations in ArcGIS are painfully inefficient (rasters get written to disk at almost every step!) Dramatic improvements might be achieved by performing more calculations within Python standard memory-based arrays (using the ArcGIS RasterToNumPyArray() function to translate). In this MC simulation, a uniform error distribution was used. However, choosing correct error distributions can be critical (Zhou et al, 2003) and as the resolution (precision) of our height measurements are markedly smaller than their accuracy, something closer to a 'normal' distribution might be more appropriate.

Burrough and McDonnell (1998, p243) show that for increasing number of variables MC simulations are better replaced with analytical methods using standard statistical theory to model error propagation, allowing for additional adaptation for correlation between variables.

RESULTS

Considering each DEM in turn, maps are shown for each of the three sea level rises, indicating probability distribution of land remaining above sea-level. Thus to assess a 90% confidence level ignore red and orange pixels.

OS Landform Panorama DEM

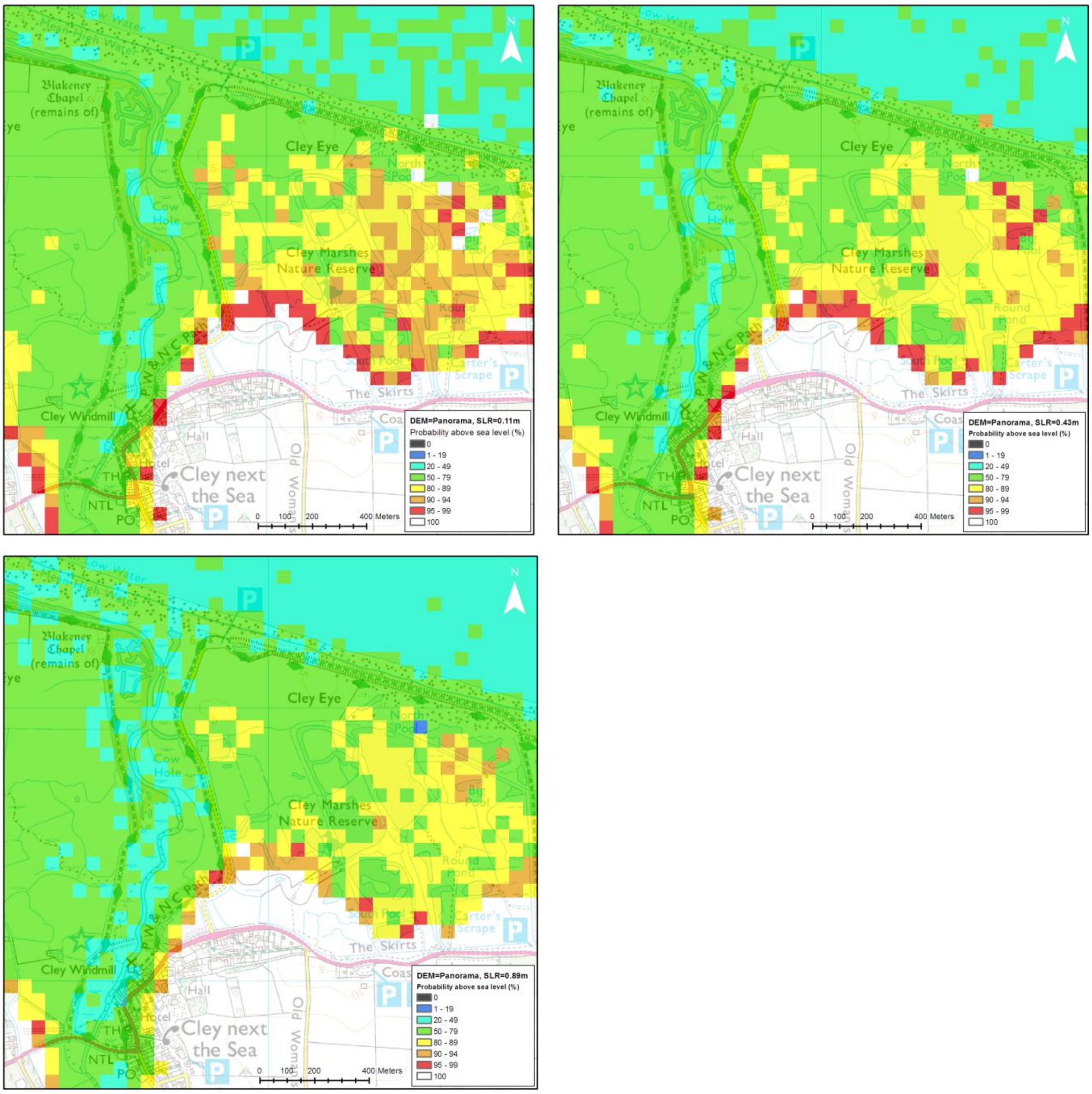

Figure 2: OS Landform Panorama sea level Monte Carlo simulation results

The course resolution Panorama images suggest that gross inundation is likely and not very sensitive to varying sea level rises.

OS Landform Profile DEM

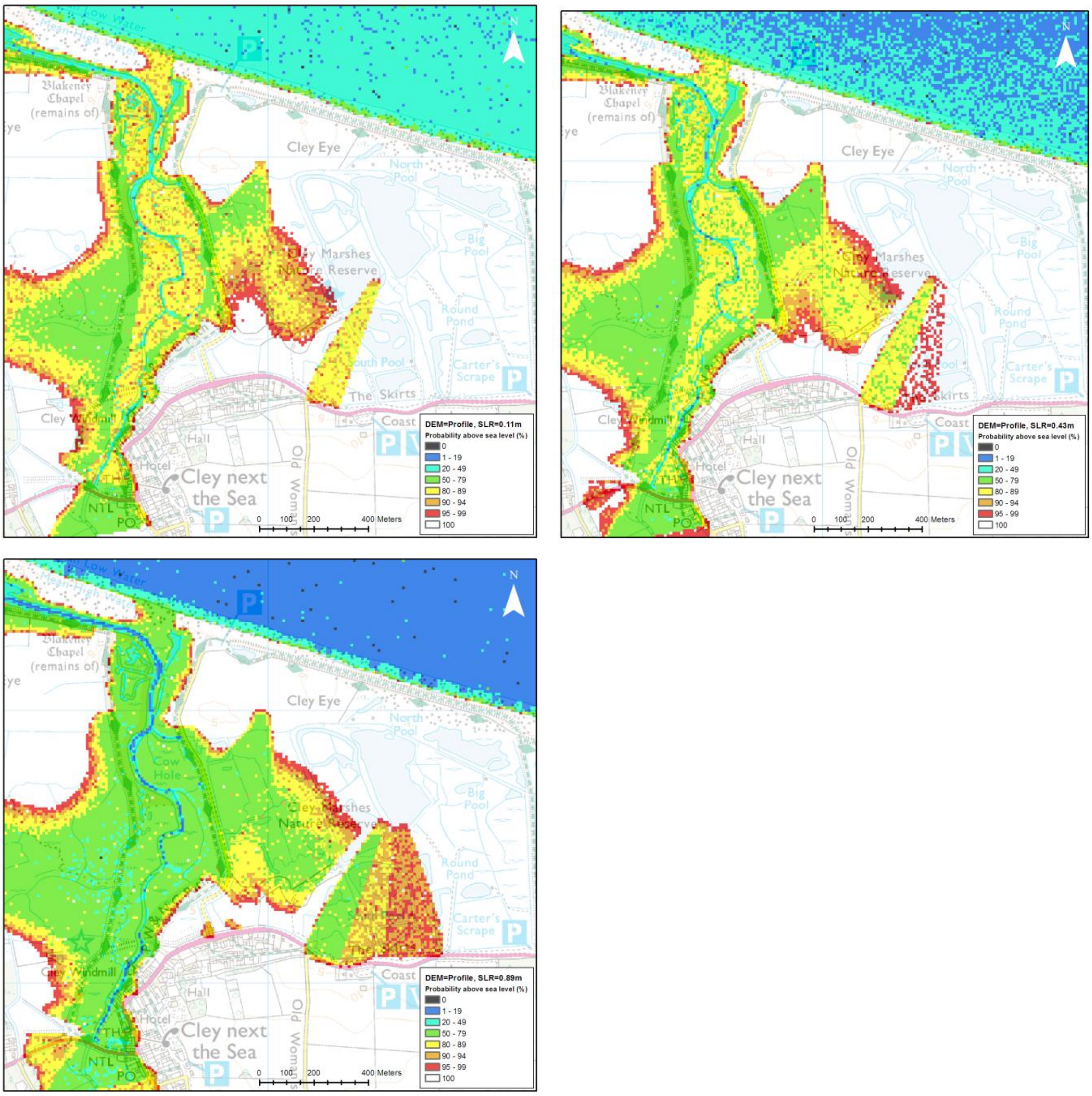

Figure 3: OS Landform Profile sea level Monte Carlo simulation results

More accurate 'Profile' data indicates more progressive change with sea level, swelling round the river channel. However, geometric artefacts are visible on the right, possibly due to errors from these maps being generated by digitizing paper maps followed by vector-to-raster conversions.

Environment Agency LiDAR DEM

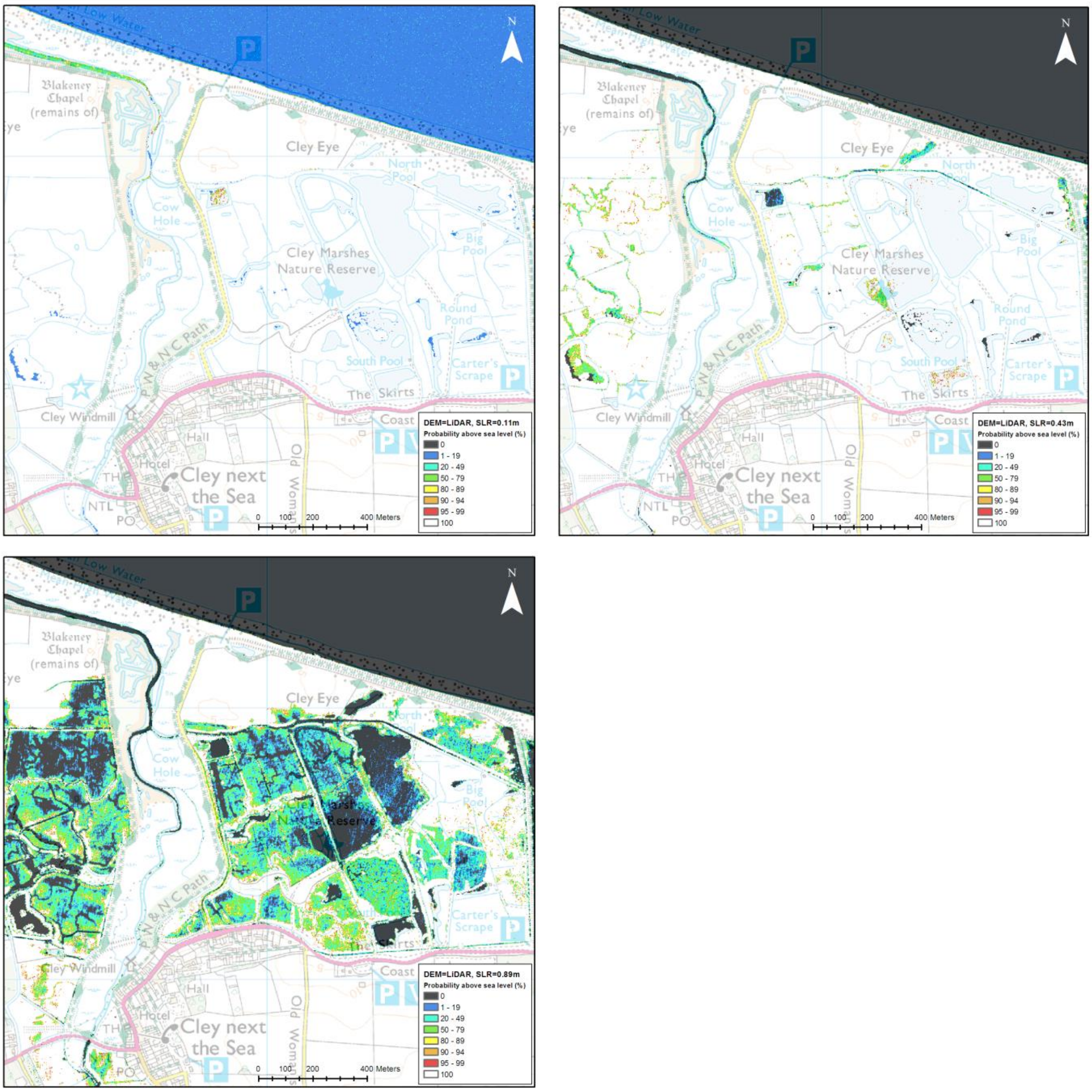

Figure 4: Environment Agency LiDAR sea level Monte Carlo simulation results

Conversely, the more accurate LiDAR data suggests the 2 lesser sea-level rises have limited effect. The 0.89cm rise highlights low lying marsh areas, suggesting potentially not a change of coastline as such, but perhaps a switch to predominantly saltwater rather than freshwater habitats.

ANALYSIS AND CONCLUSIONS

The LiDAR 0.89m SLR map shows the problem of using the “bath tub” method (Poulter and Halpin, 2008) in which a grid cell is assumed flooded if below sea level: additional analysis of hydrological connectivity is required. However with rasters this can be problematic: 4-way pixel connections may under-estimate, whereas 8-way (diagonal) may over-estimate connectivity.

MC simulation can be helpful in assessing both probability of inundation and suitability of available DEMs – in the test region, the course resolution of the Landform DEMs were shown to be misleading. Poulter and Halpin (2008) emphasize the gains to be had from improved horizontal (as well as vertical) accuracy. For better assessment of the effects of sea level rise, other factors must also be considered such as erosion/accretion of material by the sea (Chu-Agor et al, 2011) and tidal and wave dynamics (Gesch, 2009).

REFERENCES

Brunsdon, C., Carver, S., Charlton, M. & Openshaw, S. 1990. A review of methods for handling error propagation in GIS. in *Proceedings of 1st European Conference on Geographical Information Systems.* Amsterdam, Netherlands, April 1990, 106-116.

Burrough, P & McDonnell, R. 1998. *Principles of geographial information.* Oxford University Press, Oxford.

Chu-Agor, M. L., Muñoz-Carpena, R., Kiker, G., Emanuelsson, A., & Linkov, I. 2011. Exploring vulnerability of coastal habitats to sea level rise through global sensitivity and uncertainty analyses. *Environmental Modelling & Software*, 26(5), 593-604.

Gesch, D. B. 2009. Analysis of lidar elevation data for improved identification and delineation of lands vulnerable to sea-level rise. *Journal of Coastal Research*, 49-58.

IPCC. 2007. Fourth Assessment Report: Climate Change 2007. [Online]. [Accessed 5 March 2014]. Available from:
https://www.ipcc.ch/

Poulter, B., & Halpin, P. N. (2008). Raster modelling of coastal flooding from sea-level rise. *International Journal of Geographical Information Science*, 22(2), 167-182.

Zhou, G., Esaki, T., Mitani, Y., Xie, M., & Mori, J. 2003. Spatial probabilistic modeling of slope failure using an integrated GIS Monte Carlo simulation approach. *Engineering Geology*, 68(3), 373-386.

APPENDIX: ArcGIS Python Script for Monte Carlo Simulations

```
#-------------------------------------------------------------------------------
# Monte Carlo Simulations to Model uncertainty in sea level rise
#
# Copyright (c) Richard Thomas 2014. All rights reserved.
#
# (Based on a script supplied by Steve Carver 2014)
#-------------------------------------------------------------------------------

import os      # Standard package for operating system dependent functionality
import sys     # Error handling
import random  # Standard package for random number generation
import arcpy   # ArcGIS's Python site package
import time    # Standard package to measure time taken

# ---- Define simulation parameters ----

# Predicted sea level rises (in metres)
seaLevel1 = 0.11
seaLevel2 = 0.43
seaLevel3 = 0.89

# Number of simulations
nsims = 100

# Uncomment line to select DEM
selectDEM = "Panorama"
#selectDEM = "Profile"
#selectDEM = "LiDAR"

if selectDEM == "Panorama":

    # DEM raster file name (relative to folder containing this script)
    demFile = "cley_panorama.asc"

    # RMS error of DEM height values (in same units as DEM)
    heightErrorRMS = 5.0

    # Filenames for "above sea level" counts for each sea level rise
    # (13 chars max)
    countAboveSL1File = "Pano_abo_SL1"
    countAboveSL2File = "Pano_abo_SL2"
    countAboveSL3File = "Pano_abo_SL3"


elif selectDEM == "Profile":

    # DEM raster file name (relative to folder containing this script)
    demFile = "cley_profile.asc"

    # RMS error of DEM height values (in same units as DEM)
    heightErrorRMS = 2.5

    # Filenames for "above sea level" counts for each sea level rise
    # (13 chars max)
    countAboveSL1File = "Prof_abo_SL1"
    countAboveSL2File = "Prof_abo_SL2"
    countAboveSL3File = "Prof_abo_SL3"

elif selectDEM == "LiDAR":

    # DEM raster file name (relative to folder containing this script)
    demFile = "cley_lidar.asc"

    # RMS error of DEM height values (in same units as DEM)
    heightErrorRMS = 150.0

    # Filenames for "above sea level" counts for each sea level rise
    # (13 chars max)
    countAboveSL1File = "LiDAR_abo_SL1"
    countAboveSL2File = "LiDAR_abo_SL2"
    countAboveSL3File = "LiDAR_abo_SL3"

    # Convert sea levels from m to mm (to match units of DEM)
    seaLevel1 *= 1000
    seaLevel2 *= 1000
    seaLevel3 *= 1000

else:
    print "Unrecognized DEM specified"
    sys.exit(1)


# ---- Start execution ----
```

```
class LicenseError(Exception):
    pass

try:
    # Check out Spatial Analyst Licence (if available)
    print "Checking out Spatial Analyst licence.."
    if arcpy.CheckExtension("Spatial") == "Available":
        arcpy.CheckOutExtension("Spatial")
    else:
        raise LicenseError

    # Throw an exception if an ArcGIS tool produces a warning
    arcpy.SetSeverityLevel(1)

    # Sets folder with script and map file(s) as workspace
    arcpy.env.workspace = os.getcwd()

    # Allow over-writing of files
    arcpy.env.overwriteOutput = True

    # Get DEM raster from file
    demRas = arcpy.sa.Raster(demFile)

    # Sets study area to have the same extent as that of 'dem'
    extent = demRas.extent

    # Sets cell size of output raster layers to be the same as that of 'dem'
    cellSize = (demRas.meanCellHeight + demRas.meanCellWidth)/2

    # ---- Start simulation ----

    print "Starting simulation for DEM: %s" % demFile

    # Initializes random number generator
    random.seed()

    # Note start time of loop for timing number crunching
    startTime = time.time()

    # Initializes for loop from 1 through 100 iterations
    for i in range(nsims):
        print "Iteration {0}".format(i+1)
        randRasName = "tmp_rand"
        arcpy.CreateRandomRaster_management(
              out_path = arcpy.env.workspace,   # Location of the output the raster dataset
              out_name = randRasName,           # Name of output random layer
              distribution = "UNIFORM 0 1",     # Random probability model (Uniform in the
range 0 to 1)
              raster_extent = extent,           # Extent of random layer
              cellsize = cellSize)              # Cell size of random layer

        # Creates random height values based on RMS error signature of the
        # terrain model raster layer 'dem'
        scaledRandRas = heightErrorRMS \
                - (arcpy.sa.Raster(randRasName) * 2.0 * heightErrorRMS)

        # Creates a new terrain model raster layer with random values added to
        # the terrain model raster layer 'dem'
        dtmRas = demRas + scaledRandRas

        # Reclassifies the randomised terrain model raster layer into
        # above (1) and below (0) sea level
        aboveSeaLevel1Ras = arcpy.sa.Con(dtmRas > seaLevel1, 1, 0)
        aboveSeaLevel2Ras = arcpy.sa.Con(dtmRas > seaLevel2, 1, 0)
        aboveSeaLevel3Ras = arcpy.sa.Con(dtmRas > seaLevel3, 1, 0)

        # If it is the first simulation:
        if i == 0:
            # Set counts of land above sea level to first sim results
            arcpy.CopyRaster_management(aboveSeaLevel1Ras, countAboveSL1File)
            arcpy.CopyRaster_management(aboveSeaLevel2Ras, countAboveSL2File)
            arcpy.CopyRaster_management(aboveSeaLevel3Ras, countAboveSL3File)

        else:
            # Accumulate 0 or 1 resultsto above sea level counts for each level
            accum1Ras = arcpy.sa.Raster(countAboveSL1File) + aboveSeaLevel1Ras
            accum2Ras = arcpy.sa.Raster(countAboveSL2File) + aboveSeaLevel2Ras
            accum3Ras = arcpy.sa.Raster(countAboveSL3File) + aboveSeaLevel3Ras
            accum1Ras.save("accum1")
            accum2Ras.save("accum2")
            accum3Ras.save("accum3")
            arcpy.CopyRaster_management("accum1", countAboveSL1File)
            arcpy.CopyRaster_management("accum2", countAboveSL2File)
            arcpy.CopyRaster_management("accum3", countAboveSL3File)

    # Calculate total time for number crunching
    elapsedTime = time.time() - startTime
```

```
    # Delete intermediate calculation rasters
    # (Note: temporary raster files would be deleted when Python window or
    #  ArcGIS quits anyway)
    arcpy.Delete_management(randRasName)
    arcpy.Delete_management(scaledRandRas)
    arcpy.Delete_management(dtmRas)
    arcpy.Delete_management(aboveSeaLevel1Ras)
    arcpy.Delete_management(aboveSeaLevel2Ras)
    arcpy.Delete_management(aboveSeaLevel3Ras)
    arcpy.Delete_management(accum1Ras)
    arcpy.Delete_management(accum2Ras)
    arcpy.Delete_management(accum3Ras)

    print "Completed", nsims, "iterations in", elapsedTime, "seconds."
    print "Check workspace folder for output raster layers:"
    print os.getcwd()

# Catch licence server problems
except LicenseError:
    print "Spatial Analyst license is unavailable"

# Catch warnings from geoprocessing tools (if severity level = 1)
except arcpy.ExecuteWarning:
    print arcpy.GetMessages()

# Catch errors from geoprocessing tools
except arcpy.ExecuteError:
    print arcpy.GetMessages()

# Return Spatial Analyst Licence (this would happen anyway when python quits)
finally:
    arcpy.CheckInExtension("Spatial")
```