GIS 6103 Programming for GIS
Class 6 Exercise

Problem Overview

As a GIS Programmer-Consultant for ArcpyManiacs Inc, you have been tasked with helping to solve a sampling problem for another company - Reader Global Enterprises Inc (hereafter RGE). This company has 14 possible locations for monitoring sites but the geographic range of any site varies with cost. The company envisages using somewhere between 4 and 8 locations, the fewer the better in terms of cost. Meanwhile, the company contracts with another company (Antonov-Vaziri Airborne Ltd) to supply sample points, but this company can only guarantee that the sample points will fall within the *combined rectangular extent* of any monitoring sites in use. The number of sample points provided can also vary but, again, with increasing cost as the number increases. Finally, due to operational constraints, sample points falling too close to roads or streams are not useable, although there is some doubt as to the distances at which this occurs.

RGE wishes to maximize the number of useable sample points from all those generated while minimizing cost. Therefore, they want to know something about how the number of useable points varies as the number of monitoring stations changes, the geographic range of the stations changes, and the number of generated sample points change. Also, since it is unclear at what distance away from roads/streams points become unusable, they need to be able to test different values for these distances for the two themes respectively.

The number of possible scenarios of different numbers of monitoring stations (chosen from a large set), different geographic ranges, different numbers of sample points, and different distances to roads/streams is GIGANTICALLY HUGE, even if we categorize the variation. However, the company has a highly trained division of Bayesian statisticians who can make estimates of the most cost-effective choices as long as they can be provided with sufficient random sample data. This is where we come in …….for the BIG $$;) …..or, in your case, a fun size snickers bar from your professor (i.e. RGE)!

As a secondary consideration, any sample point data produced to create the random sample will itself be used by RGE for other analyses, so some characteristics of the points will also be needed – which (drainage) basins they fall in, and what are the soil drainage types and habitat suitability scores associated with the points.

The First Problem

The company's previous consulting company on this project (Joseph, Christy, Campbell & Partners) quit the project on bad terms (they were only offered 'skittles') and the data they left behind (vendor.gdb) was a mess and with little guidance as to what was the correct data to be used (are we surprised given this company?). We know that vendor.gdb basically contains data on monitoring sites, (drainage) basins,

streams, roads, soil drainage types, and habitat suitability scores but there seems to be myriad feature classes for each of these!

Fortunately, a group of hackers known only by their name Appl(e)Py (but widely thought to consist of Tafani, Perry, Pleasant, and Leon) have managed to uncover certain characteristics of the correct data that they then sold to ArcpyManiacs at an exorbitant fee. These characteristics are:

- For polygon and polyline feature classes, the map projection of the data should be "NAD_1983_HARN_Albers", and the data should contain a field named "Verified" that has a length of 3.
- There is only one correct feature class for each of the polygon/polyline 'themes' – basins, roads etc…
- For point feature classes, the shape type should be point and there should only be one feature in the feature class (i.e. each monitoring station has its own feature class, so there are 14 in total)

Your first task therefore in your solution script, is to basically find the correct feature classes and then to copy them to a feature data set (indata) within the analysis_inputs.gdb. Tip: One thing to consider as you develop/test your script is that you may (or will!?) - thru mistakes - copy the wrong feature classes and so it would make sense to ensure that the indata feature data set is clear of all feature classes before we begin any copying – this capability should be in your script.

< Coding Break >

The Basic Analysis (Model)

The basic analysis for this problem was created in a ModelBuilder model. Fortunately, ArcpyManiacs has managed to recruit, from the previous consulting company, the small team who worked on this model (Coonts, McGraw, and Vandayer-Wise) and they have recreated a version of it for ArcpyManiacs (ok, so that part is made up!). You can access the model thru the map document and toolbox provided. The model for the basic analysis produces a point feature class for just one 'scenario' (useable sample points), along with a count of how many points are in the feature class. The latter is what RGE is interested in since they want to maximize the number of useable sample points while minimizing cost. Again, that will be left to the statisticians but you need to produce enough scenarios to provide the data.

Once you have an understanding of the model, export it so that you can copy-paste from it, especially the tools, where needed, in your solution script.

(Extra note – when performing the union of basins, soil drainage, and habitat suitability feature classes, priority ranks were used and RGE wants you to incorporate this capability in your script should they decide to change them at some point)

The Analysis

Based on the overview above, the various parameters can take the following values:

Number of Generated Points	50, 75, 100
Geographic Range of the Stations	2.5, 3.0, 3.5 (Kilometers)
Distance Bands from Roads	1.0, 1.5, 2.0 (Kilometers)
Distance Bands from Streams	0.5, 1.0, 1.5 (Kilometers)
Number of Stations	4, 5, 6, 7, 8

For any "run", you will produce 9 feature classes representing all the different numbers of generated points for all the different geographic ranges of the stations (i.e. 3x3). For each run, distance values for the roads and streams will each take one value selected at random from the possible values for each. Finally, your run will randomly choose the number of stations from the values possible, and then will randomly choose that many stations from the 14 available.

Your output point feature classes should be placed in results.gdb and take names of the example form:

Stns_6_25_50pts_Rds_15_Strm_05

(i.e. Stations 6, Range 2.5km, 50 points, Roads Band 1.5km, Streams Band .5 km).

In naming your feature classes, how could you handle the possibility of more than one feature class with the same characteristics? – do so.

Your summary print output should be across four lines and follow the example form below:

The number of sample points in the sampling space for:
7 Stations with Buffer 35, 100 points,
Road Buffer of 10, Stream Buffer of 05 is:
28

Finally, you need to merge all the generated feature classes into one for use in the secondary analyses that RGE wants to do. Create this feature class in results.gdb with the name *allpoints*.