# https://github.com/Drozd363/ArcGIS\_arcpy\_S9\_15...git

# P1: Моделювання зон опадів у штаті Небраска

Припустимо, що ви працюєте над проектом для департаменту сільського господарства Небраски і вам доручили зробити деякі карти щодо опадів в штаті. Члени департаменту бажають побачити, які частини штату були відносно сухими і вологими торік, класифіковано по зонах. Все, що ви маєте - це серії спостережень метеостанції із сукупної кількості опадів на 2008 рік, що ви отримали для Небраски і прилеглих районів. Цей shapefile із точок називається **Precip2008Readings.shp.** Знаходиться у вашій папці даних Lesson 1.

Precip2008Readings.shp це вигаданий набір даних створений для цього проекту. Місцях не відповідають фактичному місцезнаходженню метеостанцій. Однак вимірювання у 2008 приводилися реальні, створені [*PRISM Climate Group*](http://www.prismclimate.org) at Oregon State University, 2009.

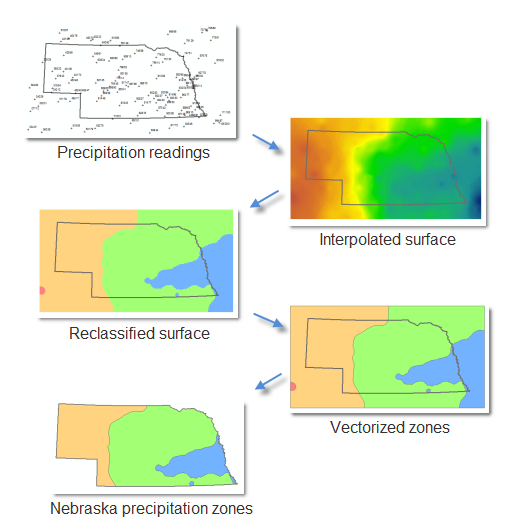
Ви повинні зробити кілька завдань для того, щоб отримати дані готові для відображення:

• Проінтерполюйте для ваших точок поверхню опадів. В результаті буде створено растрові дані з розрахунковими значеннями опадів для всієї вашої області. Необхідно використовувати для цього метод зворотних зважених відстаней (IDW) або Kriging. Ви також вибрали точки, що включають деякі райони навколо Небраски, щоб уникнути крайових ефектів в інтерполяції.

• Рекласифікуйте інтерпольовані поверхні в порядку класифікації "зон" опадів, які окреслюють відносно сухі, середньої вологості і вологі регіони.

• Створіть векторні полігони із зон.

• Обріжте зони полігонів по кордону штату Небраска.



Можливо, що ви хочете повторити описаний вище процес для того, щоб перевірити різні параметри інтерполяції IDW або інші аналогічні карти з іншими наборами даних (таких як опади у наступному році). Таким чином, розглянутий ряд завдань добре підходить для ModelBuilder.

**Але Ваше завдання, створити сценарій Python та підключити його через інструмент в ArcMap, який може зробити наведений вище ряд кроків, без необхідності вручну відкривати чотири різних інструменти.**

**Параметри інструменту**

Ваша **інструмент** повинен мати ці (хоча можна додати деякі, за погодженням) параметри:

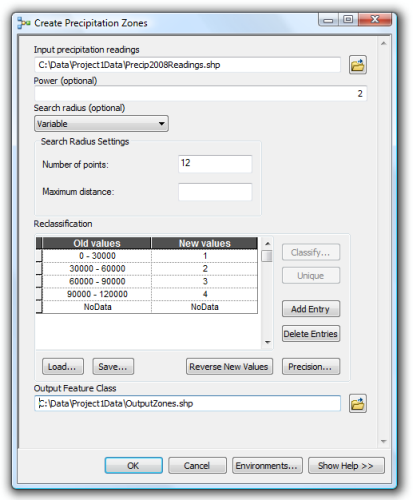
1. **Input precipitation readings** - Це розташування спостережених опадів з наведеними показаннями в точках. Цей параметр такий, що **інструмент** може бути легко повторно запущено з іншими наборами даних.
2. **Power** - IDW налаштування значення того, як будуть впливати оточуючі точки – вплив буде зменшується в міру віддалення від точки при інтерполяції.
3. **Zone boundaries** - Це таблиця, що дозволяє користувачу моделі вказати межі зон. Наприклад, можна налаштувати значення опадів 0 - 30000, що призведе до рекласифікації 1 (щоб відповідати зоні 1), 30 000 - 60 000 може призвести до класифікації 2 (щоб відповідати зоні 2), і так далі. Спосіб отримати цю таблицю, це зробити змінну з **Reclassification** параметра з **Reclassify tool** і встановити її в якості параметра моделі. На початковому етапі можна опустити і доробити пізніше.
4. **Output precipitation zones** - Це розташування, де ви хочете зберігати вихідний набору даних (dataset) обрізаних векторною зоною.

Коли ви будуєте свій сценарій, вам необхідно налаштувати деякі параметри, які *не* будуть подаватися в якості параметрів. Вони включають в себе Сlip feature (властивостей), які в штаті Небраска є контуром **Nebraska.shp** у вашій папці даних Lesson 1. Є багато інших параметрів, таких як "Z Value field" і "Input barrier polyline features" (для IDW) або "Reclass field" (для Reclassify), які не повинні подаватися в якості параметрів.

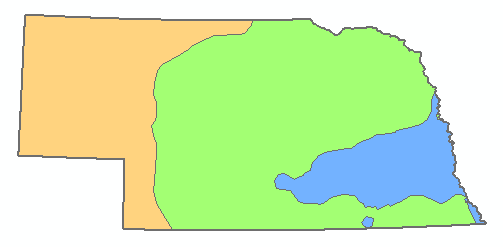
Ви повинні просто встановити ці значення один раз, коли ви будуєте свій сценарій. Хоча Ви можете додати можливість їх встановлення динамічно.

Для цієї конкретної моделі, ви повинні вважати, що будь-який вхідний набір даних буде відповідати тій же схемі даних, як і **Precip2008Readings.shp**. Наприклад, аналітик повинен бути в змозі пов’язати аналогічний набір даних **Precip2009Readings** з тими ж полями, іменами полів і типами даних. Тим не менш, він або вона не повинні чекати, щоб надати будь-яку функцію класу з різним набором полів та їх імена, і т.д.

Якщо Ви користувалися на попередньому етапі ModelBuilder та створили модель, то коли Ви двічі клацніть на моделі, щоб запустити її, інтерфейс повинен виглядати наступним чином:



Запуск інструменті (скрипта, моделі) з точними параметрами, перерахованими вище, повинен привести до нижче наведених результатів (я символізував зони у ArcMap з різними кольорами, щоб це допомогло відрізнити їх). Це один із способів, яким ви можете перевірити свою роботу:



**Очікується**

Результати для цього проекту:

* .tbx файл інструментарію, що містить інструмент (модель).
* Сценарій, який реалізує наведені кроки. Перш ніж запустити сценарій (результат повинен виглядати дуже схоже на наведене зображення, хоча ви можете встановити свої власні значення рекласифікації, power і т.д.) в ArcMap, перевірте його в IDE.
* Інструмент в ArcMap, який реалізує сценарій і у якого налаштований інтрерфейс. Ви не повинні використовувати Layout для цього проекту.

import arcpy  
arcpy.env.overwriteOutput = True  
# Script arguments  
InputPoints = arcpy.GetParameterAsText(0)  
  
Power = arcpy.GetParameterAsText(1)  
if Power == '#' or not Power:  
 Power = "2" # provide a default value if unspecified  
  
Reclassification = arcpy.GetParameterAsText(2)  
if Reclassification == '#' or not Reclassification:  
 Reclassification = "0 30000 1;30000 60000 2;60000 90000 3;90000 120000 4" # provide a default value if unspecified  
  
Clip\_Features = arcpy.GetParameterAsText(3)  
  
Output\_Feature\_Class = arcpy.GetParameterAsText(4)  
  
  
# Process: IDW  
arcpy.gp.Idw\_sa(InputPoints, "RASTERVALU", "Idw\_shp2", "", Power, "VARIABLE 12", "")  
  
# Process: Reclassify  
arcpy.gp.Reclassify\_sa("Idw\_shp2", "VALUE", Reclassification, "Output\_raster\_", "DATA")  
  
# Process: Raster to Polygon  
arcpy.RasterToPolygon\_conversion("Output\_raster\_", "PolygonRe", "SIMPLIFY", "VALUE")  
  
# Process: Clip  
arcpy.Clip\_analysis("PolygonRe", Clip\_Features, Output\_Feature\_Class, "")

* **P2: Створення контурів для Fox Lake DEM**

У другій частині проекту отримаємо деяку практику в Python. Ваше завдання - написати свій власний скрипт. Цей скрипт створить векторний контур ліній з набору висот растру. Не забувайте, що довідка [ArcGIS Desktop Help](http://desktop.arcgis.com/ru/documentation/) дійсно може бути корисною, якщо ви повинні з'ясувати, синтаксис для конкретної команди.

Fox Lake DEM знаходиться у вашій папці даних Lesson 1. Він являє собою висоти в Fox Lake Quadrangle, Utah. Напишіть скрипт, який використовує **Contour** **tool** з панелі інструментів **Spatial Analyst** для створення контурних ліній для чотирикутника. Переріз рельєфу повинен бути 25 метрів і база контуру має бути 0. Пам'ятайте, що рідна одиниця DEM - метри, тому ніякої таблиці переведення не потрібно.

Запуск скрипту повинен негайно створити shapefile контурних ліній на диску.

Дотримуйтесь цих рекомендацій при написанні скрипту:

* Мета цієї вправи просто щоб ви отримали деяку практику написання коду Python та налаштування інструменту. Таким чином, ви зобов'язані використовувати arcpy.GetParameterAsText () для отримання вхідних параметрів для різного набору даних.
* Також, Вам потрібно створити інструмент, підключити скріпт і налаштувати інструмент для цієї вправи. Параметри інструменту визначте самостійно.
* Ваш код повинен працювати правильно з IDE. Також, скріпт повинен містити коментарі, перехоплення помилок, вивід повідомлень, а також використовувати вірні та інтуїтивні імена змінних.

**Очікується**

Результати для цього проекту:

* .py файл, що містить скріпт.
* .tbx файл інструментарію, що містить інструмент.
* Налаштований інструмент.

Всі результати повинні бути портованими та представлені на Git. Крім того не забувайте, що ця робота є командною, тож **використовуйте** Git.

import arcpy  
arcpy.env.overwriteOutput = True  
inrast = arcpy.GetParameterAsText(0)  
outcont = arcpy.GetParameterAsText(1)  
interval = arcpy.GetParameterAsText(2)  
if interval == '#' or not interval:  
 interval = "25"  
basecont = arcpy.GetParameterAsText(3)  
if basecont == '#' or not basecont:  
 basecont = "0"  
  
try:  
 arcpy.gp.Contour\_sa(inrast, outcont, interval, "0", "1")  
 arcpy.AddMessage("Created contour!")  
except:  
 arcpy.GetMessage(0)