NRS-528 – Geographical information systems in Python

Times: 13.30-16.30 on Friday ● **Location:** Woodward Hall 006 ● **Instructor:** Dr Andy Davies ● **Contact:** davies@uri.edu ● **Office Location:** CBLS 291 or Zoom ● **Office hours:** Wed 10am-12pm (Zoom only), Fri 10am-12pm in office, or other times during the week by appointment, I can do outside of normal working hours to accommodate your schedules as best I can.

Pre-requisites: Students should have a degree of proficiency using ArcGIS, usually obtained through successful completion of NRS 410. If you have not, then you should contact me to discuss whether you are prepared for this course and to obtain a permission number.

Course Description

One of the most powerful elements of ArcGIS, beyond its vast visualization and data management capabilities, is the ability to automate nearly all geoprocessing tasks using Python. Python is a popular programming language, at its core, it is easy for beginners to understand yet has vast capabilities, largely through the addition of numerous third-party packages. ArcGIS itself uses an additional proprietary package known as arcpy, which extends Python with powerful geospatial capabilities, spatial file management and also allows us to process data with other libraries such as numpy and scipy. In this class, I will introduce you to Python and how it functions within ArcGIS Pro, but much will be applicable to ArcGIS Desktop as well as other open-source programs such as QGIS. This is a skills-based course, so we will begin by learning basic-programming skills using the Python language. We will advance these within the arcpy environment and begin developing basic scripts to automate and extend some common geoprocessing tasks. We will develop an understanding of good coding practice, open-source programming and turn scripts into fully functioning toolboxes for wider dissemination to non-programming geospatial analysts. We will base our learning around Github and use this for developing and sharing our code.

Course Credit

NRS-528 is a three-credit course, that includes weekly self-study learning components, computer-based hands-on classes and assignments to work towards these credits (see below in the *Assessment* section and the *Schedule*).

Github repository

- A digital syllabus, all code, example data and assignments are available on the Github page: https://github.com/marecotec/Course_ArcGIS_Python

Brightspace

- Brightspace is used for all communications, i.e., as Announcements and for submission, feedback and grading of assignments. Panopto screen recordings can also be obtained from here as well.

Course goals

- Expose you to the Python programming language and provide opportunity to develop and practice your coding skills using basic functionality.
- Introduce arcpy and how this can be used to automate and extend geoprocessing tasks.
- Provide you with the skills needed to successfully develop Python tools that can be used, not only by yourself, by other users.
- Provide practice in the use of Github as a tool for code dissemination and storage.

Learning outcomes

- **LO-1** You will be able to produce basic Python code that is functional and extendable: 1) including operating system operations, including file creation, manipulation, searching and filtering. 2) Core Python functionality including for loops, if/else, lists and variable manipulation. 3) Extending the capabilities of Python by importing various libraries.
- **LO-2** You will be able to undertake basic geoprocessing tasks by using Python code and the arcpy library, that mirror routines that you would previously have used the graphical user interface or ModelBuilder to complete.
- **LO-3** You will be able to package code into usable Python Toolboxes that will be available to users via ArcToolbox, which will include adequate help and explanatory files for other users to execute your Toolbox.
- **LO-4** You will be able to participate in the open-source software movement including the practices of code sharing and dissemination through the Github website. You will be able to produce understandable code that is appropriately commented to allow other users to run your programs.

Self-learning component

Each week prior to meeting in the computer teaching laboratory, you will undertake self-study exercises to address simple coding problems in preparation for the class. These are called “Coding Challenges” and are a type of *flipped classroom* whereby you undertake self-learning prior to coming to class where we undertake more advanced topics. These challenges are designed to be achievable in approximately two to three hours and must be completed prior to the specific class for which they are assigned (see *Schedule*), as we will go discuss each challenge in the class (see *Assessment* below). Some challenges will require additional preparation including research on code-repositories such as textbooks, Github, or forums such as Stack Overflow. Your code from each challenge must be maintained in your Github account, as it forms part of your overall coding portfolio.

Classroom component

In class, we will explore basic programming concepts and advanced geospatial applications that are suited for automation using Python. During lab, we will work on training materials produced by the instructor, which are designed to ensure you meet the learning outcomes of the course. We will use the software development program *PyCharm*, which allows rapid code generation and can be linked into ArcGIS for code execution.

Reading resources

- "*ArcPy and ArcGIS*", 2017, 2nd Edition by Silas Toms and Dara O'Beirne, freely available through the URI Library and you can download it as a full PDF for use offline. I do not expect you to buy this unless you wish to have reference material.
- Additional material for certain lectures will be posted on Github or listed in the description of each lecture.

Assessment

%	Topic
10	Attendance and participation
20	Coding challenges
25	Midterm tool challenge
25	Python Toolbox that includes several tools that have been coded and documented by yourself and released as a Github repository.
20	Github account portfolio

1. Attendance and participation: Attendance is critical to your success in this class. To earn full credit, you must attend and participate in each and every class. If you have an emergency, you are ill, or you have an absence covered under the University policies (i.e., Holy Days, University Sanctioned Events) you are expected to contact the instructor in advance of class. For illness, this particularly pertains to flu-like symptoms, so if you do exhibit signs of flu, contact the instructor and stay home to minimize transference of the virus. Unexcused absences will result in a lower course grade.

2. Weekly coding challenges: Learning to code is not something that can be done entirely within a classroom, as it requires much practice and reinforcement, which can only be achieved individually. Between each class you will undertake a self-study "Coding challenge", which will take approximately two to three hours of your time, with feedback provided after submission. The first 10 minutes of each class is reserved for a coding challenge review. *Addresses learning outcomes 1 and 2.*

3. Midterm tool challenge: In this assignment, you are instructed to produce a small tool that takes advantage of arcpy and Python. You will need to provide example data, and the code should run on all PCs. The tool needs to manipulate a dataset across three different processes, for example, extracting, modifying and exporting data. The exact workflow is entirely up to yourself. You are expected to take 3-4 hours on this coding assignment, and you should deposit your code and example files within a

Github repository for feedback and grading. Criteria are, 1) cleanliness of code, 2) functionality, appropriate use of documentation and depth of processing operation. Assignment due week 9. *Addresses learning outcomes 1, 2 and 3.*

4. Python Toolbox: In your final assignment for this course, you should create a Python Toolbox that contains a minimum of three simple tools for undertaking geoprocessing and file management operations. These tools can be discrete or part of a larger workflow. However, the caveats are that you should create a "single file" toolbox (no includes, or external file tools) and not exceed 2000 lines of code in its entirety. You should document the toolbox using Github README.md and provide example data for running each of your tools. Grading and feedback will focus on: 1) Does the toolbox install, and the tools run successfully? 2) cleanliness of code, 3) functionality and depth of processing operation, and 4) appropriate use of documentation. Assignment due week 14. *Addresses learning outcomes 1, 2, 3 and 4.*

5. Github account portfolio: The final assessment component is your Github coding portfolio. As a coder, your Github account can also act as an unofficial resume, providing potential employers, graduate supervisors or collaborators with ready access to your prior work. You must maintain a high quality Github account for the other assignments set, this includes providing adequate and well-organized readme files, well commented code and a coherent file structure. Grading and feedback will focus on 1) High level of organization and 2) Descriptive readme files and well commented code. Assignment due week 14. *Addresses learning outcome 4.*

Weekly schedule

Date	Topic	Preparation
Week 1	Introduction, overview of different assignments for this class and modes of teaching. Set up Github accounts. The different Python environments and ways to interact with Python through ArcGIS (and not) – Python.exe, bat files, just clicking on a *.py, geoprocessing command line, PyCharm, toolboxes (*.pyt and traditional).	
Week 2	Introduction to Python basics Commenting, import statements for packages, variables and data types (str, int, float, lists, tuples, dictionaries). Iteration using for loops, if/elif/else statements, while statements. Code cleanliness (indentation using tabs/spaces, spotting indentation errors), Using Functions to block code. Zero-based indexing.	Coding challenge 1

Week 3	Introduction to Python modules os, sys. Basic file and system manipulation. arcpy. Present some arcpy functionality, where can you find scripts, basic resources.	Coding challenge 2
Week 4	Building basic scripts Cover basic coding tasks, and introduce error handling, print statements and various messaging functionality.	Coding challenge 3
Week 5	Building your first script by cheating Using ModelBuilder and ArcToolbox to export python scripts. Extending exported python scripts.	Coding challenge 4
Week 6	Environments, functions and file handling Setting environments within arcpy. How to interact with and code input for tools that are available through arcpy. Avoid repeating code using functions.	Coding challenge 5
Week 7	Introduction to Cursors Selecting, searching, updating data using arcpy functions, and non-arcpy alternatives.	Coding challenge 6
Week 8	Geometry objects and raster manipulation Creating geometry objects, points, lines and polygons. Creating and working with raster datasets	Coding challenge 7
Week 9	Spatial analyst and other extensions in Python Practice using various ArcGIS extensions through arcpy. Check out/in licenses, advanced functionality.	Coding challenge 8
Week 10	Interacting with ArcGIS Desktop from code Techniques to manipulate the desktop environment.	Coding challenge 9

Week 11	Effective Python Toolboxes Pythonizing your toolboxes to provide usable interactive scripts all within a single python file.	Coding challenge 10
Week 12	Designing scripts for others Building scripts that can be used by others, open source licenses to protect you and your code and dissemination through Github.	
Week 13	Code review and end of class discussion	

Additional information:

Grade Scale: A = 94.0–100%; A- = 90.0–93.9%; B+ = 87.5–89.9%; B = 84.0–87.4%; B- = 80.0–83.9%; C+ = 77.5–79.9%; C = 74.0–77.4%; C- = 70.0–73.9%; D+ = 67.5–69.9%; D = 60.0–67.4%; F = below 59.9%

Viral Illness Precautions: The University is committed to delivering its educational mission while protecting the health and safety of our community. Students who are experiencing symptoms of viral illness should NOT go to class/work. Those who test positive for COVID-19 should follow the isolation guidelines from the CDC. If you are unable to attend class, please notify me prior to the start of class via email.

Academic Honesty: Students are expected to be honest in all academic work. A student's name on any written work, quiz or exam shall be regarded as assurance that the work is the result of the student's own independent thought and study. Work should be stated in the student's own words, properly attributed to its source. Students have an obligation to know how to quote, paraphrase, summarize, cite and reference the work of others with integrity. If you use sources such as *Stack Overflow* or other web-based materials in assembling your code, please cite them as comments. If you are using generative AI tools to help you produce your code, also please cite the tool used and the prompt that you utilized.

Excused Absences: Absences due to serious illness or traumatic loss, religious observances, military service, or participation in a university sanctioned event are considered excused absences. Students are responsible for work missed during an excused absence but will not be penalized by grading or assignment/exam make-up policies. Students should notify faculty in advance of absences due to religious observance or university-sanction events, and as soon as possible for other absences See University Manual sections 8.51.11-8.51.16 for details.

Mental Health and Wellness: We understand that college comes with challenges and stress associated with your courses, job/family responsibilities and personal life. URI offers students a range of services to support your mental health and wellbeing, including the URI Counseling Center, TELUS Health Student Support App, the Wellness Resource Center, and Well-being Coaching.

Land Acknowledgement: The University of Rhode Island land acknowledgment is a statement written by members of the University community in close partnership with members of the Narragansett Tribe. The statement recognizes and pays tribute to the people who lived on and stewarded the land on which the University now resides. The statement seeks to show gratitude and respect to Indigenous people and cultures and build community with the Narragansett Nation and other Native American tribes.

The University of Rhode Island occupies the traditional stomping ground of the Narragansett Nation and the Niantic People. We honor and respect the enduring and continuing relationship between the Indigenous people and this land by teaching and learning more about their history and present-day communities, and by becoming stewards of the land we, too, inhabit.

Anti-Bias Syllabus Statement: We respect the rights and dignity of each individual and group. We reject prejudice and intolerance, and we work to understand differences. We believe that equity and inclusion are critical components for campus community members to thrive. If you are a target or a witness of a bias incident, you are encouraged to submit a report to the URI Bias Resource Team at www.uri.edu/brt. There you will also find people and resources to help.

Disability, Access, and Inclusion Services for Students: Your access in this course is important. Please send me your Disability, Access, and Inclusion (DAI) accommodation letter early in the semester so that we have adequate time to discuss and arrange your approved academic accommodations. If you have not yet established services through DAI, please contact them to engage in a confidential conversation about the process for requesting reasonable accommodations in the classroom. DAI can be reached by calling: 401-874-2098, visiting: web.uri.edu/disability, or emailing: dai@etal.uri.edu.

Late Work: Due dates (in Brightspace) have been designed to provide you with a clear workplan for this course so that you cannot leave everything to the last minute; you must take a regular approach to handing in work and building your portfolios. Work submitted after this time for each deadline, but no more than 5 days late will have 50% grade reduction and feedback will be provided. Work submitted more than 5 but less than 10 days late will receive 0%, but feedback will be provided. Work that is 10 days late will not be accepted. If you think you have a valid excuse for handing in work late, please contact the instructor to discuss any issues.