Code-literacy for GIS librarians: a discussion of languages, use cases, and competencies

Abstract

Geospatial librarianship has always been inherently tied to technology, requiring that GIS librarians have specialized computer literacy. In recent years, the push for open science, transparency, and improved reproducibility in research has led to increased use of open source geospatial technology. These developments have led to a blending of GIScience with the emerging field of Data Science, and many students and researchers are increasingly using free and open source programming languages to complement or replace traditional GIS software. New data formats, modern modes of data exchange, and the data as a service trend require an intricate understanding of computer science. If GIS librarians are to meet the needs of current and future patrons, code-literacy will soon be a necessity, yet a discussion of these tools and skills are lacking in the literature of geospatial librarianship. This paper discusses the present revolution in geospatial technology, recommending an integration of coding skills into the most elemental levels of the Core Competencies for GIS Librarianship. We review two programming languages, R and Python, for their suitability for librarians wishing to advance their technical skills by either learning a new language or learning to program for the first time. Drawing from our own experiences, we discuss the potential of coding to transform how GIS librarians engage with their work.

Keywords: Librarianship; GIS; geospatial data; programming; data science

Introduction

Cartographic materials and geographic information of all types have long inhabited a unique corner of the library. As with many library materials, collections of cartographic information have evolved significantly over the past half-century, and the librarianship associated with these collections continues to evolve alongside the information itself. With the rise of digital information, geospatial technologies, and the World Wide Web, librarians dealing with spatial information have developed new skills in the areas of geographic information systems (GIS), digitization, and online data repository development. As with physical maps, developing a deep understanding of the evolving forms of digital cartographic information has been critical to successfully providing access to geospatial information and creating quality metadata for discovery of these information types.

The more recent emergence of the interdisciplinary field of Data Science in academia has brought about the need for librarians with new technology skills. The ability to work with and provide access to "big" and open data are important to data science librarianship, but these types of information often exist outside of the typical library ecosystem. Information technologists working in non-library settings frequently share data via web services or application programming interfaces (APIs) that cater to data consumers with basic computer programming abilities. Computer programming and script writing, colloquially known as coding, is the process of writing instructions in a programming language that computers perform automatically. When data science researchers and students turn to the library for assistance in acquiring, understanding, and working with large datasets in unfamiliar formats, it is necessary that librarians, in turn, attain the skills required to provide reference service and technical guidance desired by our patrons. In this digital age, that often equates to writing computer program code.

Data services in the academic library are not themselves new. GIS and geospatial data librarians, social science librarians, and government information librarians have provided services in this vein for at least the past three decades. Finding data on the World Wide Web is a point of expertise—and, indeed, a core competency—of GIS librarianship. What *is* new are the scale, formats, and methods by which data providers share data. Methods of access continue to evolve for both spatial and non-spatial data, and the ability to access new modes of data programmatically are increasingly important to GIS librarians.

Coding skills, however, may be useful to GIS librarians in more ways than just assisting with data access. As researchers use increasingly larger data sets, automation of GIS analysis will become more commonplace, and use of programmatic and machine learning techniques in GIS will likely increase. The open science movement also stands to make significant changes to the work of GIS librarians. Many researchers have begun shifting their methods toward free and open source technologies that allow others to repeat their methods with greater ease. Open source programming languages like R and Python and their supporting software applications are becoming more popular with researchers wishing to share their methods—and the code they wrote to apply those methods—with their research communities. For libraries, supporting the open science movement means both supporting researchers in their methodological advances as well as providing a means for the sharing of researchers' data in institutional repositories.

This paper discusses new technologies and changing modes of scholarship that are on the cusp of changing how GIS and spatial data librarians approach their work. Free and open source geospatial technologies—particularly those that are independent of the standard GIS user interface—have advanced to the point that there are now more GIS software options available than ever before. In terms of geospatial services in the library, it is up to librarians to move in concert with advances in technologies and the open future of spatial research methods. It is the authors' experiences that developing fundamental skills in writing code have improved our abilities in providing geospatial services. Basic coding skills have opened up new avenues toward answering research questions that may have once resulted in dead ends. The aim of this paper is to open a discussion of the advantages of writing code in the field of geospatial librarianship and share use cases for applying a coding approach toward solving practical problems in the library—a discussion that has yet to take place in the literature. We argue for an integration of new technical skills into the repertoire of GIS librarianship. In particular, R and Python, two programming languages used widely due to their accessibility and ease of learning, are discussed in the context of their geospatial capabilities and usefulness to GIS librarians, as well as other, related code-based applications. This paper fills an important gap in the literature of geospatial librarianship by discussing the latest advancements in open and code-based GIS methods and how librarians can apply them to their work.

Background

GIS and Spatial Technology in the Academic Library

Academic libraries have a long history of working with GIS. The Association of Research Libraries (ARL) GIS Literacy Project initiated in 1992—a collaboration between ARL and Esri, a leading GIS company—is often cited as the project that jump-started GIS services in academic libraries (Yu 1998; Association of Research Libraries 1999; March and Scarletto 2017). This project focused on educating librarians about GIS tools so that they could better help patrons access and use geospatial data. Over the past quarter century following the success of this initial collaboration, academic libraries have continued to develop and evolve GIS services (March and Scarletto 2017).

Initially, library GIS services in the US and Canada were closely associated with government information (French 2000; Davie, Fox, and Preece 1999). This was a logical result of the fact that much of the data that GIS users wanted to access was produced by government agencies. For example, the TIGER street data produced by the US Census is often cited in early articles about GIS librarianship (Yu 1998; Lamont 1997). The goal of connecting users with data was explicit in the ARL project: "This multi-phased Project seeks to introduce, educate, and equip librarians with the skills necessary to provide access to spatially referenced data in all formats, and to provide effective access to selected electronic information resources in library collections," (Davie, Fox, and Preece 1999, 16). Facilitating access to spatial data continues to be a key role that GIS librarians play (March & Scarletto 2017; Scholtz 2016).

Comfort with technology has long been a requirement for GIS librarians, with the observation that, "Computer literacy is critical in learning GIS," (Yu 1998, 99). As documented by the ARL surveys, one approach has been to hire GIS staff for the library who have backgrounds in GIS, rather than re-training librarians. This trend reflects the real complexities of dealing with GIS data and tools. As noted by Yu over twenty years ago, "An important issue facing libraries is how to make the GIS accessible and usable by users," (1998, 107). This is still a central issue for libraries. Support for using GIS tools remains a commonly cited role for library GIS services, both for data access support and for general support (Scholtz 2016). Indeed, the tools that GIS practitioners use to access, manipulate, and analyze data have always informed how academic libraries have built and provided GIS services.

As March & Scarletto (2017) describe in their survey of the evolution of GIS services in North American libraries from 1995–2016, the 1990s and early 2000s were dominated by a few major desktop software packages, such as the various Esri products (ArcInfo, ArcView, etc.) and some remote sensing tools. The past 10–15 years have seen an expansion of the number and type of programs being used for GIS applications, including developments such as web mapping and a move towards open source geospatial tools such as QGIS (March and Scarletto 2017; Scholtz, 2016). While the library literature has noted the shift toward open source tools, most references are to graphical user interface (GUI) tools and do not take into account programming languages that have mapping and spatial analysis capabilities. Indeed, the authors of the 2014 ARL survey phrased their inquiry into technology as “Geographic Information Software/Mapping Technologies Provided at ARL Member Academic Libraries,” which—while important in today’s data landscape—provides an overly narrowly accounting of the tools actually in use since it does not account for support that might be given to coding approaches (Holstein 2015). Yet, there has been little to no discussion in the GIS librarianship literature about the growing popularity of computer programming tools in geospatial work and how to support this trend.

The Data Science Tidal Wave

Library patrons drive the evolution of skill, technology, and competency trainings and expertise offered among the suite of GIS services in libraries. With the proliferation of geospatial technologies like Google Maps, global positioning systems, and numerous geo-enabled apps into everyday life, more researchers beyond the realm of geography and the natural sciences are integrating GIS into their work. Researchers familiar with a particular software package often look for extensions of their software of choice to perform GIS tasks rather than investing the time to learn a dedicated GIS application like ArcGIS. Open source technologies like R and Python have many GIS code libraries that now provide robust alternatives to desktop GIS in a variety of use-scenarios (Srivastava 2018). Coding, being a general skill, is useful in a multitude of scenarios; researchers now use their flexible coding skills to apply analyses of all types including GIS.

Coursework emphasizing computer theory and programming skills within the Earth Sciences have recently become more common—evidence of the proliferation of computer and data science across disciplines. In a 2014 article, Muller and Kidd discuss the necessity of computer programming training in geography education and share a model for computer science instruction in geography (2014). geography educators seem to have taken note. In a review of curriculum offered in university geography departments in the United States, Bowlick et al. discovered that 80% of the departments reviewed now offered coursework in GIS programming (Bowlick, Goldberg, and Bednarz 2017). At both of the authors' institutions (The University of Colorado Boulder and University of California Berkeley), coursework is offered in data science and programming applications in GIS. For example, the Earth Lab at the University of Colorado Boulder offers a popular professional certificate in Earth Analytics that focuses on open-source, code-based education in both R and Python as a means to facilitate collaboration and reproducible science.[1]

Separately from computer programming training within GIS-specific instruction, the converse is also increasing. As data science becomes more prevalent and routine in undergraduate education, students without a GIS background are exploring mapping in their data science work. At the University of California Berkeley, a tidal wave of data science has been rising over the past few years. The university began offering an entry-level massive class called "Foundations of Data Science" ("Data8" for short) in Fall 2015. Students have the opportunity to explore data science applications in different disciplines through "connector" courses, which

[1] https://www.colorado.edu/earthlab

have included spatial analysis focused courses from the departments of Geography and Environmental Science, Policy, and Management. Building on the success of the Data8 classes, in Fall 2018, the University of California Berkeley established a new Division of Data Sciences. One of the many possible domain emphases in the new accompanying data science major is Geospatial Information and Technology. The growing list of over twenty domain emphases encompasses such diverse topics as Evolution and Biodiversity, Inequalities in Society, and Data Arts and Humanities, with majors applying data science to a wide variety of disciplines. As data science gains more prevalence on campus outside of Computer Science, the library must change and adapt to support students and research.

Some authors advocate not just for coding education in geography, but also for the use of open source GIS tools in geography education. In a 2016 paper, Etherington discusses developing a GIS programming course based entirely on an open source, Python approach (2016). Etherington points to greater accessibility (i.e., freely available) and greater transparency (i.e., ease of replicability) as the chief benefits of open source geospatial methods. While noting the spatial analysis capabilities of R, Etherington cites Python as the prominent GIS programming language in part due to its integration with ArcGIS software but also for its stand-alone GIS computational abilities (2016). Muller and Kidd also advocate for open source software for GIS programming when possible, citing the ease of student access to software outside of the instructional laboratory (2014). In their case, R was chosen for one module of their programming course because of its free availability, all-purpose data analysis flexibility, and spatial analysis capabilities. Whether researchers, educators, and students choose R, Python, or another programming language to conduct their research, it is clear that both the established GIS community as well as newcomers to geospatial analysis increasingly turn to open source technology.

The progression of map/GIS librarianship and data librarianship follow parallel tracks. Data librarian positions—be they focused on data curation, research data management, data visualization, or general data services—are common in libraries, and have trended in recent years toward incorporating knowledge and skills in programming and open source code libraries. In an article about data librarian competencies, Semeler, Pinto, and Rozados frame data science skills as the foundation for data librarians (2017). They state that data science librarians "need not become a programmer, but should be interested in the languages and programming logic of computers" (Semeler, Pinto, and Rozados 2017). The authors suggest that technical skills like computer programming are an important extension of foundational data librarianship elements like data management, visualization, and curation. A review of data librarian job advertisements by Khan and Du corroborates this idea (2018). They found that 70% of the 50 job descriptions analyzed required skills in R, Python, or other data science software packages. 60% of the advertisements listed programming languages as preferred skills (Khan and Du 2018). Following the lead of data librarianship, GIS librarianship must also keep its finger on the pulse of geospatial researchers' needs on campus. The ongoing trend toward open science, open source software and formats, and emerging methods of data acquisition in research and higher education call for the adoption of new skills and expanded competencies in GIS librarianship.

Building on the Core Competencies

The factors described above point toward a new layer of technical knowledge and skills needed by librarians providing geospatial services. In 2008, the Map and Geography Roundtable of the American Library Association (now the Map and Geospatial Information Roundtable, or

MAGIRT) published its Map, GIS and Cataloging / Metadata Librarian Core Competencies document (Weimer, Andrew, and Hughes 2008). While the fundamental elements of the Core Competencies remain sound, GIS librarians should seek to expand their technical skills, particularly in the scripting languages that are popular around their campus.

The core competencies describe skills needed by GIS librarians in three levels, with level 1 being the most basic competencies and level 3 being the most advanced. The document refers to coding skills only once, on page 13, describing creation of scripts as a level 3 skill in the context of batch processing, data ingestion, web services, and geoprocessing (Weimer, Andrew, and Hughes 2008). With the recent proliferation of data science and open geospatial code libraries into the academic sphere, there is a case to be made that coding skills in GIS librarianship should shift downward from a level 3 competency toward levels 2 or even 1. Whereas previously scripting abilities may have seemed most relevant to data and collection management, today GIS librarians seem likely to be called upon to use coding skills in everyday patron interactions. This “filtering down” of coding skills into the lower level core competencies elevates their importance. Here, we describe how programming knowledge and coding skills might be integral to multiple parts of MAGIRT’s Core Competencies:

- **II.B.1. Data Acquisition:** Knowledge of and the ability to interact programmatically with emerging data resources like APIs allows for automated extraction of data sets of varying scales.
- **II.B.2. Data Discovery:** Geoprocessing tasks can be automated over multiple items or collections to aid in metadata manipulation and crosswalking, aiding in the production of discovery and administrative metadata for geospatial data collections.

- **II.B.3. Information Technology:** Librarians should strive to include the free and open source geospatial programming tools that researchers now frequently employ on library computers and in GIS labs.
- **II.B.4 Technical Infrastructure:** Basic knowledge of geospatial code libraries installed in computer labs will be important to ensuring their functionality, including basic troubleshooting.
- **II.C.1. Reference:** Librarians should expand their understanding of emerging data formats such as JSON and GeoJSON and how to parse, visualize, manipulate, and obtain them from web services.
- **II.C.2. Instruction:** Introductory and intermediate-level workshops on automation of GIS tasks, open source GIS alternatives, and GIS code libraries stand to benefit both traditional desktop GIS users as well as statisticians and computer and data scientists who wish to integrate geospatial methods into their current workflows.
- **II.C.3. Research Consultation:** An expanded knowledge of the emerging methods of conducting geospatial analysis and obtaining geospatial data to include coding/programmatic methods will aid in serving researchers and students that utilize the increasingly broad array of GIS software, resources, and discovery tools.

In their article about computer programming instruction in geography, Muller and Kidd describe an objective of their course as aiming to produce not programming experts, but rather “advanced beginners” (2014). Similarly, in a paper discussing professional development in the realm of coding for librarians, Popowich discusses the differences between a “hacker” and a “programmer” approach to coding—hacking being more akin to writing scripts to accomplish specific, sometimes single-use tasks as opposed to large-scale development of software (programming)(2017). Both Muller and Kidd's and Popowich's arguments support the idea that simple coding skills, whether framed as "advanced beginner" or "hacker", are useful to the work of librarians working outside of IT departments. In this vein, the present paper also does not advocate that librarians become computer-programming experts, but, rather, we encourage librarians working in geospatial roles to develop fundamental—that is, advanced beginner or hacker—coding abilities that will complement their current knowledge and abilities. Rather than proposing new core competencies in GIS Librarianship, we instead advocate for new skill development that builds on the core competencies of the profession.

In the following sections, we discuss the advantages of learning either Python or R, and share code libraries from both languages that we have found to be of particular use in providing geospatial library services. We will also discuss related code-based GIS technologies of relevance and share example use cases where basic coding knowledge aided reference interactions and saved time in everyday work tasks.

Emerging Languages & Technology Relevant to GIS Librarians

Choosing a Language

There are many programming languages out there, but R and Python may be the best options for GIS librarians due to their ease of learning, accessibility, large and diverse user communities, and well-developed GIS code libraries. Anyone can use either of the languages and their associated software since they are free and open source. Both are also compatible with Windows, Mac, or Linux operating systems. Either would be a good choice, but each one has different strengths. Both languages share much of the same all-purpose data and GIS-specific functionality, so which language should GIS librarians choose to learn?

R is most popular among statisticians and academics in the sciences but also has a wide user community in social sciences, business, engineering, and even medicine (Gardener 2012). R was developed by statisticians, and, hence, is strong in statistical analysis, modeling, and visualization (The R Foundation 2019). One advantage to using R is that it is developed in particular for working with data and statistics, whereas Python was developed as an all-purpose programming language. Many tools for statistical analysis and data manipulation are native to R. Python users, on the other hand, often must extend the language by installing additional Python-compatible code libraries. Ozgur et. al write that R users—beginners in particular—have fewer choices to make up front regarding data formats and manipulation tools (2017). R can also be easily extended by "packages," which are extensions to the standard functions of R (Table 1). Because of its roots in statistics and data, R has powerful, straightforward tools for conducting exploratory data analyses. R has long been used in spatial statistical research and modeling. Because it is open source, it is also a good platform for creating reproducible research. Essentially, R is a good choice for GIS librarians who are doing statistical analysis and/or work with departments whose members use R heavily.

<Insert Table 1 in this area>

Although spatial analysis in R is not new (Bivand et al. 2008), it has been gaining in popularity and acceptance within the geospatial community over the past few years. At the 2015 Esri User Conference Esri announced the R-ArcGIS Bridge, a set of tools hosted in a Github repository that provide a way for users to transfer data between R and ArcGIS software.[2] Though not nearly as integrated with the Esri ecosystem as Python, the development of the R-ArcGIS Bridge is evidence of R's growth.

[2] https://r-arcgis.github.io/

Python is used by a wide audience in many disciplines. Hetland, in the book, *Beginning Python: from Novice to Professional*, states, “...one might as well ask, ‘Who isn't using Python?’" (2017). It is often used by programmers working in data science, web development, game development, and in many other areas (Hetland 2017). Python has advantages of its own. One big advantage to learning Python is that it is easy. In the book *Python for Data Analytics*, Nelli writes that it is the choice of most newcomers to programming due to its simple syntax and ease of human readability (2015). Even though it is great for beginners, Python is also a good choice for expert programmers because it can work with modules and programs written in other languages (Hetland 2017). One of the benefits of learning Python is that it can be used to do almost anything; there is a saying among Python users: “there’s a Python library for that.” In fact, there are over 160,000 packages available in the Python Package Index.[3] If a librarian has the goal of learning a language that can do more than scientific computing and data manipulation (such as web development, for example), then Python would be a good choice.

Python is also popular among GIS users. Python has been integrated into ArcGIS software since version 9 was introduced in 2004 (Esri 2019a). Both ArcMap and ArcGIS Pro provide Python interpreters within the user interface (i.e., a dedicated window for typing and executing Python code) designed for automating tasks. The ArcGIS Python module (known as ArcPy is also available as a library that can extend Python, meaning you can run ArcGIS operations from a Python interface without ever opening ArcMap or ArcGIS Pro (Esri 2019b). For these reasons, Python has become very popular among GIS professionals.

For librarians who wish to conduct operations in wholly open source environment, there are many options as well. Similar to ArcPy, QGIS has supported Python since version 0.9 in the

[3] https://pypi.org/

form of the PyQGIS library and also includes a Python interpreter in the user interface (QGIS 2019). Python developers have contributed many other free and open source code libraries to the community. Like ArcPy, users can use PyQGIS in a Python terminal without ever launching the QGIS application interface. Some other Python libraries that may be of particular use to librarians are GeoPy (for geocoding), Pandas and GeoPandas (vector data manipulation), Numpy (scientific computing), rasterio (raster manipulation) and Matplotlib (visualization)—though there are many other code libraries available for Python (Table 1).

R and Python may be the most approachable tools for GIS librarians new to programming, but there are many other open-source GIS technologies available. The Open Source Geospatial Foundation (OSGeo) is a leader in advocating for open source geospatial technology (OSGeo 2019a). OSGeo promotes and sponsors many of the most popular open source geospatial projects, including desktop applications (i.e., QGIS, GRASS GIS), web mapping technology (i.e., OpenLayers, GeoServer), spatial databases (PostGIS), and code libraries (i.e., the geospatial data abstraction library or GDAL). GDAL is of particular note as it is a widely used C++ code library for working with both vector and raster data formats (OSGeo 2019b). ArcGIS, QGIS, and GRASS GIS all rely on GDAL for many data manipulation, processing, and translation tasks, and most librarians providing geospatial support have used GDAL even if unknowingly. More information on the open source technologies noted above, as well as many more, are available on the OSGEO website.[4]

While so many free and open source geospatial technologies are available, R and Python are still likely the best choices for librarians due to their ease of learning, prolific use, broad applicability, and their well-developed geospatial libraries. In addition, there are bindings

[4] https://osgeo.org/projects

available that allow use of free code libraries in both R and Python. For example, R and Python users can use GDAL by installing GDAL-based libraries specific to those languages (OSGeo 2019c). For these reasons, it is important that librarians providing GIS support are aware of the depth and breadth of open source geospatial tools but, in terms of programming languages, should be practiced in R, or Python, or both. Librarians would do well to choose either language; the two are nearly equal in performance and capabilities. The choice of which to learn boils down to one's own computing needs as well as those of one's library patrons.

Related Technologies

Familiarity with these new environments for conducting geospatial research also requires an understanding of modern methods of data exchange and data formats. APIs are becoming increasingly important to librarians providing data reference services. Libraries have made use of APIs like the many provided by OCLC for some time to automate metadata creation and to integrate external services into their websites (Michel 2013). In a 2018 article, however, Adams advocates for the embrace of APIs by librarians providing research support (2018). Adams cites the shifting information landscape as a prime reason, and theorizes that working with APIs may soon be a skill as essential as navigating library indexes (2018). Fortunately, the technology behind APIs has shifted toward methods that are more accessible to people with beginner programming skills. Many web services now follow the Representational State Transfer (REST) model, which can be interacted with very simply in a web browser. Because many data providers, including Esri, now grant access to data via REST APIs, having a working knowledge and an ability to interact with them is important for librarians providing both GIS and data services. The librarian that adds some basic programming abilities to their toolkit will be able to write a basic for-loop that can automate data collection from APIs and advise researchers that wish to do so. It seems likely that librarians and library patrons will obtain data via APIs with greater frequency in the near future.

Use of APIs also requires an understanding of the most often used format for API data exchange: JavaScript Object Notation (JSON). JSON is an open-standard data format that has grown in popularity since its introduction in the mid-2000s due to its utility as a data exchange format on the web (Marrs 2017). Many web services that follow the Representational State Transfer (REST) API model exchange information in JSON format (some also offer eXtensible Markup Language (XML) as an alternative format). Like XML, JSON follows a nested data structure, and obtaining data from an API often necessitates parsing JSON-formatted data into a structure like a data table or spreadsheet that researchers will likely desire. Both Python and R offer packages that facilitate API interaction and JSON and XML transformations (Table 1).

Also of note is GeoJSON, an extension of JSON with the ability to store vector data structures. Librarians providing geospatial services will likely have heard of GeoJSON, as it is a widely used format for displaying vector data on the web. Because of the rise in use of both JSON and GeoJSON as new methods of data exchange and storage, knowledge and skills in manipulating them may soon be as critical to GIS librarians as using a spreadsheet.

Basic skills related to web development are also increasingly important to librarians of all types and especially to GIS librarians. Library patrons are often interested in presenting the results of their GIS analysis or visualization in an interactive manner on the Web. Esri tools like ArcGIS Online offer easy methods for creating web maps, but those that wish to create web maps in an open-source environment often must rely on programming and markup languages. If GIS librarians are to support data visualization on the web, then knowledge of several languages will be useful. In terms of basic web-development, Hyper Text Markup Language (HTML) and

Cascading Style sheets (CSS) are fundamental, but adding a working knowledge of JavaScript can assist in a variety of tasks related to web mapping. JavaScript is programming language for web pages, and it is frequently used for creating web-maps. Openlayers, Leaflet, and Mapbox GL JS are all free and open source JavaScript libraries that can facilitate creation of web maps and their integration into web pages.[5] While JavaScript may not be as easy to learn as R or Python, librarians practiced in either language will find it easier to adapt their coding skills to a new language—a benefit of becoming code-literate.

Use Cases

There are numerous opportunities to employ coding skills in GIS Librarianship in both "front of the house" and "back of the house" settings. While tasks like metadata creation might be the first to come to mind, basic programming knowledge and abilities can also help in reference, instruction, and consultation scenarios. Here, we provide example use cases that have occurred in our work.

Technology Literacy

Patrons who do not have GIS skills often approach the reference desk or GIS librarians for help. This has long presented a tension for GIS reference providers who must balance the desire to help the patron with the amount of time required to teach the tools to accomplish the desired analysis or visualization. As more patrons develop skill sets that include programming, this opens up opportunities to introduce spatial tools that exist within realms of their existing knowledge. In other words, a patron who knows a programming language but does not have GIS

[5] See https://developers.google.com/maps/documentation/javascript, https://leafletjs.com/, https://docs.mapbox.com/help/glossary/mapbox-gl-js/ for more information.

tool experience will be able to use the spatial analysis and visualization tools contained within the libraries of their programming language. In this scenario, patrons like this turn out to be easier to help because they do not have to start with learning an entirely new software in order to conduct spatial analysis and visualization. GIS librarians can simply discuss which code libraries or packages can help them achieve their task without teaching them how to use GIS software from scratch. GIS librarians who are at a minimum familiar with what programming languages like R and Python are capable of in terms of GIS operations will be able to guide these patrons.

A sample reference interaction that occurred recently at University of California Berkeley illustrates this point. A graduate student made an appointment with the GIS librarian to discuss a number of GIS-related topics in relation to her thesis research project, including finding datasets and performing simple overlay analysis in order to get results for performing statistical analyses. The student did not have any prior GIS knowledge and, already very consumed with her primary research project, was hesitant to learn new software. During the reference interview, the GIS librarian learned that the student was already comfortable using R. Based on this context the GIS librarian recommended some R packages and tutorials that would enable the student to perform the simple overlay analysis in R. The patron was satisfied without the reference interaction turning into a lengthy consultation about learning a desktop GIS software program. This interaction exemplifies how GIS librarians' knowledge of programming languages can help patrons find an appropriate tool for their purpose, existing skills, and future goals.

Data Literacy

Data literacy questions often arise in the work of GIS librarians. Deploying programmatic techniques to transform, clean, or manipulate data streamline preparation of data for GIS can open up teaching moments between librarians and their patrons. Recently, a student at the

University of Colorado Boulder sought assistance preparing data to create a time-lapse visualization in ArcGIS. The student started with tabular data representing 42 years of annual carbon dioxide emissions data by country organized as ‘wide data,’ meaning the table consisted of one row per country with yearly emissions observations for each country organized in columns. Enabling time display in ArcGIS, however, requires organization of the time series data in ‘long’ format—that is, each country in the data set should be repeated in rows for each time observation. Reorganizing a data table of this size from wide to long format presented a time-intensive task; the final data in long form would transform from 265 rows to over 11,000. Microsoft Excel cannot perform this type of operation easily without aggregating the data, but fortunately, both R and Python provide packages that can perform pivot functions. The librarian knew how to perform such a task in Python, while the student was familiar with R. In an hour consultation, the student and librarian effectively communicated about how the pivot procedure worked conceptually and were able to translate a Python script written by the librarian into a similar operation written by the student in R (Figure 1). The resulting R script consisted of three commands and reorganized the student’s data from wide to long format in a few seconds, leaving the data prepared for import into ArcGIS. Although the librarian was less familiar with R in this case, having an understanding of simple programmatic concepts in one language translated easily into another.

<Insert Figure 1 in this area>

As data access continues to shift toward APIs and cloud-based, “data as a service” (DaaS) environments, programming skills may soon be a necessity for data access. There are already scenarios in which a GIS librarian might call upon their coding abilities to overcome obstacles that may have otherwise resulted in a data dead-end. One such example occurred recently during a reference interaction with an undergraduate student at the University of Colorado Boulder. The student requested assistance obtaining coordinate data for rock-climbing routes in Colorado. The data was available from the Mountain Project, a website that collects and shares volunteered geographic information (VGI) contributed by recreational climbers.[6] The data, however, was only available in JSON format via the Mountain Project's REST API. The librarian and student successfully implemented a Python script to interact with the API, download the data needed by the student, and transform the JSON output into a table that could be imported into a GIS. Another example of GIS DaaS is Google Earth Engine, a cloud-based web GIS for analyzing planetary scale spatial data.[7] Interacting with raster data in Earth Engine takes place primarily in a code environment, and Google provides analysis tools in Python and JavaScript for Earth Engine. Because cloud-based analysis offers benefits in scale and efficiency, it is likely that researchers will turn to code-based tools like Earth Engine with increasing frequency and GIS librarians should expect to understand how to use tools like this in the near future.

Instruction

Bringing coding knowledge into instruction settings can lead to a deeper understanding of a topic and lengthen the reach of instruction efforts to broader audiences. A workshop on the topic of geocoding, for example, could be improved by introducing multiple methods that appeal to attendees familiar with the traditional graphical GIS interface as well as those unfamiliar with GIS software but comfortable with coding. This workshop might start with attendees using ArcGIS software and its World Geocoding Service, and progress to using code to query

[6] https://www.mountainproject.com/
[7] https://earthengine.google.com/

OpenStreetMap’s Nominatim Geocoding API. Presenting geocoding in this way helps learners understand that geocoding does not necessarily require proprietary software. Instead, it demonstrates that the geocoding process, on a conceptual level, is essentially a query of an address database. Presenting GIS workshops using a mixed method approach has potential to facilitate conceptual learning rather than simple point-and-click procedures and tool-reliance.

Spatial analysis and visualization done with coding rather than GUI applications also allows for more flexibility in how GIS librarians might create online, asynchronous learning materials. This addresses the concern raised by March and Scarletto (2017) about increasing this type of instruction as part of GIS librarians’ portfolios. Workshop materials for desktop GIS applications require either that the learner install these software applications, find a workstation with them installed, or that the instructor provide a virtual environment, which can be difficult to maintain. In contrast, workshop materials for coding-based GIS operations can be set up to run simply within a web browser using Github and markdown files or an environment like JupyterHub so people can participate without installing any special software or code libraries. At University of California Berkeley, an undergraduate fellow mentored by the GIS librarian created a tutorial to document how to download, subset, and process a large climate dataset using R as part of her fellowship. The workflow documented by the undergraduate fellow teaches several core skills for working with spatial data in R, such as accessing remote data and subsetting and performing other pre-processing steps, while also providing a clear set of reproducible instructions for accessing and working with a common public environmental dataset.

Collection Management and Metadata

Librarians managing spatial data collections stand to benefit significantly from using their code skills to automate repetitive collection development practices. GIS librarians can write scripts to programmatically move files, clean and standardize data, rename files, and transform data from one format to another. For example, GIS and map set metadata creation often requires working with many similar spreadsheets. At the University of California Berkeley, the GIS Librarian is working on an ongoing project in which a student assistant creates a metadata spreadsheet for each map set. Rather than manually opening each spreadsheet to trim whitespace and do other data cleaning, the librarian wrote an R script to trim whitespace from the untidy data and rename and move the cleaned files to another folder. This continues to save time and provide more consistent results.

Metadata tasks are also made easier by making use of GIS code libraries to auto-populate geographic metadata fields. At the University of Colorado Boulder, the Earth Sciences & Map Library has been working with the library's Digital Asset Management team to digitize historical aerial photographs of Colorado. The Aerial Photographs of Colorado Collection is available in the CU Digital Library and makes use of the United States Geological Survey's (USGS) topographic quadrangle names and Geographic Names Information System (GNIS) place names to aid in discovery. Populating these item-level discovery metadata fields has now been automated with a Python script utilizing spatial joins. Now populating these metadata fields for hundreds of photographs at a time takes only a few seconds. An advantage of ArcPy is that ArcGIS desktop applications provide the Python code-snippets for individual tasks conducted in the geoprocessing history. This is advantageous because the GIS librarian can simply develop a workflow in the desktop application, then copy and paste the Python code-snippets into an interpreter to build a complete script. With a little bit of code-literacy, GIS librarians can easily transform GUI workflows into automated, coding workflows.

Learning Opportunities

There are many options for learning programming languages, including ones conducive to different types of learning styles. Programming languages are a popular focus for short in-person workshops, often offered for no cost either through the library or other on-campus learning centers. These can sometimes be one-shot sessions, but often are strung together into workshop series that provide a basic overview to a programming language. In-person workshops are also offered through such programs as Software Carpentry, Data Carpentry, and Library Carpentry, which follow a peer-to-peer instruction model. In addition to or in combination with in-person learning options, there are many online tutorials, MOOCs (Massive Open Online Courses) and published books that can be used for self-guided learning. Some of the more popular websites with free learning options include CodeAcademy, DataCamp, and Learn Python the Hard Way.

Programmers write code in a variety of applications, with the simplest perhaps being a basic text editor (like Microsoft Notepad or TextEdit for Mac). Code written in a text editor can then be executed in a command prompt, but authoring code in an integrated development environment (IDE) is more practical and conducive to learning. IDEs are software applications specifically designed for authoring code, and they allow programmers to write and run code all in the same interface. An IDE will often autocomplete snippets of code while the user is typing and also provide tools for troubleshooting and debugging code. A popular and free IDE for R is

RStudio; free IDEs for Python include Spyder, PyCharm, Atom, and many others.[8, 9, 10, 11] For beginner coders, Jupyter Notebooks are a great IDE for learning. Jupyter Notebooks are an interactive IDE that allows the user write and execute code a few lines at a time independent of the rest of the code in the script.[12] New programmers often struggle to determine in which part of a script an error is occurring, but executing pieces of code independently in a Jupyter Notebook can help beginner (as well as advanced) programmers quickly find mistakes. Jupyter Notebooks are also great for data exploration and creating quick visualizations (Figure 2). While Jupyter Notebooks mostly used by Python coders, they are compatible with R as well.

<insert Figure 2 in this area>

No matter how a librarian goes about learning, there are a few core tasks that will be useful to know. Reading/writing files and navigating file directory structures are invaluable building blocks for being able to work with data and files in a programming environment. Especially for coding tasks that involve automation, these skills along with creating for-loops and defining variables are essential. In addition, creating quick visualizations can be an important tool for checking data quality and the output of various analytical operations. GIS librarians developing their code skills should learn these basic tasks first.

Conclusion

Learning a programming language can be challenging, but the benefits—to both librarians and library users—are well worth the effort. Like the process of learning a new spoken

[8] https://www.rstudio.com/
[9] https://www.spyder-ide.org/
[10] https://www.jetbrains.com/pycharm/
[11] https://atom.io/
[12] https://jupyter.org/

language, becoming literate in a programming language requires practice, patience, and self-motivation. We argue that this effort is well worth the rewards of being better able to provide GIS services and staying on top of current trends in spatial analysis and visualization.

Libraries and librarians constantly strive to evolve and meet the information needs of patrons. Geospatial techniques continue to proliferate into new fields of study, and modes of information transfer are changing. GIS Librarians are already called upon to provide both information and technology literacy to patrons from across disciplines. But more patrons are approaching geospatial methods from outside the traditional structure of GIS training and education. Because geospatial librarianship is so intertwined with technology, it is crucial that librarians and library workers providing GIS services continue to evolve along with the tools of their trade. Failing to do so means neglecting our patrons and researchers.

Python and R are two options that can help GIS Librarians meet the expanding needs of our users and meet these challenges head-on. Learning either language would be a great decision: both are approachable and well equipped for working with geospatial data, with abundant opportunities to learn. With basic scripting knowledge in hand, the code-literate librarian will likely soon have the ability to translate processes in one language to the other. It is difficult to understate the importance of those core concepts because they can both improve our practice of librarianship as well as broaden our understanding of geospatial methods.

The code-literate GIS Librarian also has the opportunity to facilitate the uptake of new technology and skills across campus. Adapting GIS workshops to include multiple methods of performing the same task (i.e., GUI-based as well as code-based) can lead to a more nuanced understanding of spatial methodologies. It may also serve to encourage faculty members to refresh their skills and revise their curricula in favor free and open technologies.

Basic coding skills extend and build upon some of the most fundamental competencies in geospatial librarianship. When the Core Competencies for GIS Librarianship was authored over a decade ago, programming may have been thought of as an advanced level of specialization in geospatial librarianship, but it is quickly becoming a necessary skill for the profession. From advanced technical assistance all the way down to the most simple spatial data reference question, fundamental code-literacy can make us better at our jobs and help us connect with a broader audience.

Both individual librarians and the community can take action to bring coding into the forefront of GIS librarianship. Individuals should seek projects and opportunities in their everyday work to try code-based workflows. The benefits—speed, efficiency, repeatability, automating workflows—will introduce new ways of thinking about GIS and broaden approaches toward helping library patrons. Library managers and leaders in the community of geospatial librarians should seek to advance the competencies of its members by promoting skill development and training specifically in the area of coding. The successful past efforts of the ARL Geographic Information Systems Literacy Project provide an exemplar for how GIS librarians might organize once again to promote new technology-literacy among their ranks. Future iterations of the Core Competencies should elevate the importance of coding and programming languages and integrate these concepts into the more fundamental competencies from which they are presently absent. Building basic code-literacy will prepare GIS librarians for ongoing and future technological advancements in geospatial methods.

Literature Cited

Adams Jr, Richard Manly. 2018. “Overcoming Disintermediation: A Call for Librarians to Learn to Use Web Service APIs.” Library Hi Tech 36 (1): 180–90. https://doi.org/10.1108/LHT-03-2017-0056.

Association of Research Libraries. 1999. “The ARL Geographic Information Systems Literacy Project.” SPEC Kit 238, 43.

Bowlick, Forrest J., Daniel W. Goldberg, and Sarah Witham Bednarz. 2017. “Computer Science and Programming Courses in Geography Departments in the United States.” The Professional Geographer 69 (1): 138–50. https://doi.org/10.1080/00330124.2016.1184984.

Davie, D. Kevin, James Fox, and Barbara Preece. 1999. The ARL Geographic Information Systems Literacy Project. SPEC Kit 238 and SPEC Flyer 238. ARL Distribution Center, P. https://eric.ed.gov/?id=ED429609.

Dorman, Michael. 2014. Learning R for Geospatial Analysis. Olton Birmingham, UNITED KINGDOM: Packt Publishing Ltd. http://ebookcentral.proquest.com/lib/ucb/detail.action?docID=1911526.

Esri. 2019a. “Python in ArcGIS Pro—ArcPy Get Started | ArcGIS Desktop.” 2019. http://pro.arcgis.com/en/pro-app/arcpy/get-started/installing-python-for-arcgis-pro.htm.

———. 2019b. “What Is ArcPy?—ArcPy Get Started | ArcGIS Desktop.” 2019. https://pro.arcgis.com/en/pro-app/arcpy/get-started/what-is-arcpy-.htm.

Etherington, Thomas R. 2016. “Teaching Introductory GIS Programming to Geographers Using an Open Source Python Approach.” Journal of Geography in Higher Education 40 (1): 117–30. https://doi.org/10.1080/03098265.2015.1086981.

French, Mary. 2000. “The ARL GIS Literacy Project: Support for Government Data Services in the Digital Library.” IASSIST Quarterly 24 (1): 19. https://doi.org/10.29173/iq579.

Gardener, Mark. 2012. Beginning R: The Statistical Programming Language. Somerset, UNITED STATES: John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/ucb/detail.action?docID=836575.

Grolemund, Garrett, and Hadley Wickham. n.d. R for Data Science. Accessed January 25, 2019. https://r4ds.had.co.nz/.

Hetland, Magnus Lie. 2017. Beginning Python: From Novice to Professional. New York (NY): Apress.

Holstein, Ann L. 2015. “Geographic Information and Technologies in Academic Libraries: An ARL Survey of Services and Support.” Information Technology & Libraries 34 (1): 38–51.

Lamont, Melissa. 1997. “Managing Geospatial Data and Services.” Journal of Academic Librarianship 23 (6): 469. https://doi.org/10.1016/S0099-1333(97)90171-3.

March, Gregory, and Edith Scarletto. 2017. “The Evolution of GIS Services within North American Academic Libraries: Documenting Change through the Decades (1995–2016).” Journal of Map & Geography Libraries 13 (2): 222–45. https://doi.org/10.1080/15420353.2017.1313803.

Marrs, Tom. 2017. JSON at Work: Practical Data Integration for the Web. Sebastopol, CA: O’Reilly Media.

Michel, Jason Paul. 2013. Web Service APIs and Libraries. Chicago, Ill: ALA.

Muller, Catherine L., and Chris Kidd. 2014. “Debugging Geographers: Teaching Programming to Non-Computer Scientists.” Journal of Geography in Higher Education 38 (2): 175–92. https://doi.org/10.1080/03098265.2014.908275.

Nelli, Fabio. 2015. Python Data Analytics: Data Analysis and Science Using PANDAs, Matplotlib and the Python Programming Language.

OSGeo. 2019a. “About OSGeo.” OSGeo. 2019. https://www.osgeo.org/about/.

———. 2019b. “GDAL Wiki.” 2019. https://trac.osgeo.org/gdal/#GDALOGRInOtherLanguages.

———. 2019c. “GDAL/OGR.” OSGeo. 2019. https://www.osgeo.org/projects/gdal/.

Popowich, Sam1, Sam.Popowich@ualberta.ca. 2017. “Coding and Professional Development—Part 1: A Study in Contradictions.” Partnership: The Canadian Journal of Library & Information Practice & Research 12 (1): 1–9. https://doi.org/10.21083/partnership.v12i1.3961.

Scholtz, Nicole. 2016. “Supporting Geospatial Data.” In Databrarianship: The Academic Data Librarian in Theory and Practice, edited by Lynda Kellam and Kristi Thompson, 127–35. Chicago, IL, UNITED STATES: Association of College and Research Libraries. http://ebookcentral.proquest.com/lib/berkeley-ebooks/detail.action?docID=4625403.

Srivastava, Naveenchandra. 2018. Emerging Trends in Open Source Geographic Information Systems. Hershey, PA: IGI Global.

The R Foundation. 2019. “R: What Is R?” 2019. https://www.r-project.org/about.html.

Weimer, Katherine, Paige Andrew, and Tracey Hughes, eds. 2008. “Map, GIS and Cataloging / Metadata Librarian Core Competencies.” American Library Association Map and

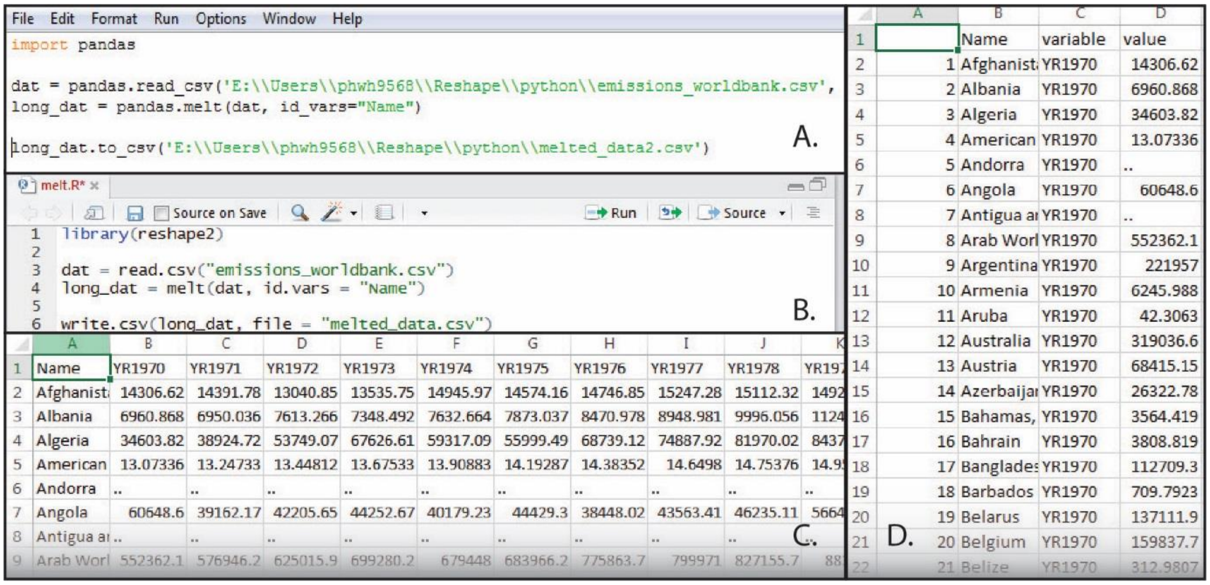
Geography Round Table. http://www.ala.org/rt/sites/ala.org.rt/files/content/publicationsab/MAGERTCoreComp2008.pdf.

Yu, Lixin. 1998. “Geographic Information Systems in Library Reference Services.” The Reference Librarian 28 (60): 87–110. https://doi.org/10.1300/J120v28n60_06.

Table 1. R and Python packages useful to GIS Librarians. [a, b, c.]

Function:	**R packages**	**Python packages**
Reading, writing, manipulating vector data	sp raster rgdal	Shapely Fiona GeoPandas PyQGIS ArcPy
Reading, writing, manipulating raster data	sp raster rgdal	Rasterio PyQGIS ArcPy
Transforming projections/coordinate systems	rgdal proj4 sp	PyProj PyQGIS ArcPy GeoPandas GDAL/OGR
Geocoding	ggmap tmap (connects to OSM Nominatim)	GeoPy Requests/json Geopandas ArcPy
Distance calculations	gdistance	GeoPy PyProj
Visualization	sp raster rasterVis leaflet tmap ggmap (only handles data stored in non-spatial data frame) rColorBrewer	Plotly Matplotlib Seaborn folium (Leaflet) Rasterio ggplot Geoplotlib Bokeh
General purpose data management, manipulation, analysis, computing, parsing	ggplot2 rjson jsonio XML	Pandas Numpy json ElementTree

[a.] A comprehensive package index for R is available at https://cran.r-project.org/web/packages;
[b.] The Python Package Index (PyPI) is available at https://pypi.org/.
[c.] See also the Awesome Geospatial list at https://github.com/sacridini/Awesome-Geospatial



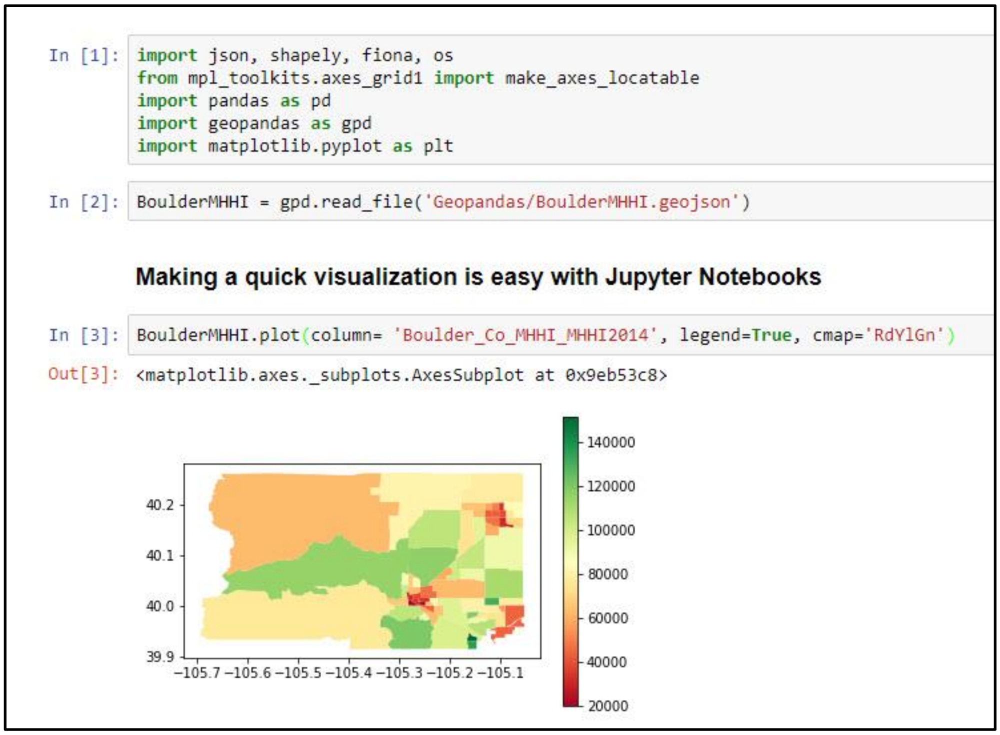
```
import pandas

dat = pandas.read_csv('E:\\Users\\phwh9568\\Reshape\\python\\emissions_worldbank.csv',
long_dat = pandas.melt(dat, id_vars="Name")

long_dat.to_csv('E:\\Users\\phwh9568\\Reshape\\python\\melted_data2.csv')
```

```
library(reshape2)

dat = read.csv("emissions_worldbank.csv")
long_dat = melt(dat, id.vars = "Name")

write.csv(long_dat, file = "melted_data.csv")
```

	A	B	C	D	E	F	G	H	I	J	K
1	Name	YR1970	YR1971	YR1972	YR1973	YR1974	YR1975	YR1976	YR1977	YR1978	YR197
2	Afghanist	14306.62	14391.78	13040.85	13535.75	14945.97	14574.16	14746.85	15247.28	15112.32	1492
3	Albania	6960.868	6950.036	7613.266	7348.492	7632.664	7873.037	8470.978	8948.981	9996.056	1124
4	Algeria	34603.82	38924.72	53749.07	67626.61	59317.09	55999.49	68739.12	74887.92	81970.02	8437
5	American	13.07336	13.24733	13.44812	13.67533	13.90883	14.19287	14.38352	14.6498	14.75376	14.9
6	Andorra	..	..	..	..	..	..	..	..	..	..
7	Angola	60648.6	39162.17	42205.65	44252.67	40179.23	44429.3	38448.02	43563.41	46235.11	5664
8	Antigua ar	..	..	..	..	..	..	..	..	..	
9	Arab Worl	552362.1	576946.2	625015.9	699280.2	679448	683966.2	775863.7	799971	827155.7	88

	A	B	C	D
1		Name	variable	value
2	1	Afghanist	YR1970	14306.62
3	2	Albania	YR1970	6960.868
4	3	Algeria	YR1970	34603.82
5	4	American	YR1970	13.07336
6	5	Andorra	YR1970	..
7	6	Angola	YR1970	60648.6
8	7	Antigua ar	YR1970	..
9	8	Arab Worl	YR1970	552362.1
10	9	Argentina	YR1970	221957
11	10	Armenia	YR1970	6245.988
12	11	Aruba	YR1970	42.3063
13	12	Australia	YR1970	319036.6
14	13	Austria	YR1970	68415.15
15	14	Azerbaijar	YR1970	26322.78
16	15	Bahamas,	YR1970	3564.419
17	16	Bahrain	YR1970	3808.819
18	17	Banglades	YR1970	112709.3
19	18	Barbados	YR1970	709.7923
20	19	Belarus	YR1970	137111.9
21	20	Belgium	YR1970	159837.7
22	21	Belize	YR1970	312.9807

Figure 1. Transforming "wide" data into "long" data with just three lines of code in either Python or R. A) A Python script using the Pandas package to complete the transformation; B) An equivalent R script using the Reshape2 package; C) The original, wide-formatted data table; D) The resulting long-formatted data table.



```
In [1]: import json, shapely, fiona, os
        from mpl_toolkits.axes_grid1 import make_axes_locatable
        import pandas as pd
        import geopandas as gpd
        import matplotlib.pyplot as plt

In [2]: BoulderMHHI = gpd.read_file('Geopandas/BoulderMHHI.geojson')
```

Making a quick visualization is easy with Jupyter Notebooks

```
In [3]: BoulderMHHI.plot(column= 'Boulder_Co_MHHI_MHHI2014', legend=True, cmap='RdYlGn')
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x9eb53c8>
```

Figure 2. Using a Jupyter Notebook to visualize data with the Python packages GeoPandas and Matplotlib. The visualization depicts median household Income in Boulder County, Colorado, USA, based on 2014 data from the United States Census Bureau.