WTBF: Where's That Book From?

interWEBS: Gracey Wilson, Sam Eppinger, Rowan Sharman, Sarah Barden
Olin College of Engineering | Spring 2016 | Software Design

Learning Goals

Gracey: Integrating different files written in parallel to each other.
Sam: Improve text mining skills and writing well-structured code.
Rowan: Working with hardware and human interfacing.
Sarah: Getting familiar with map-plotting libraries and merge conflicts.

Team Goal:
Practicing creating quality documentation (i.e. well-commented code, website, etc.)

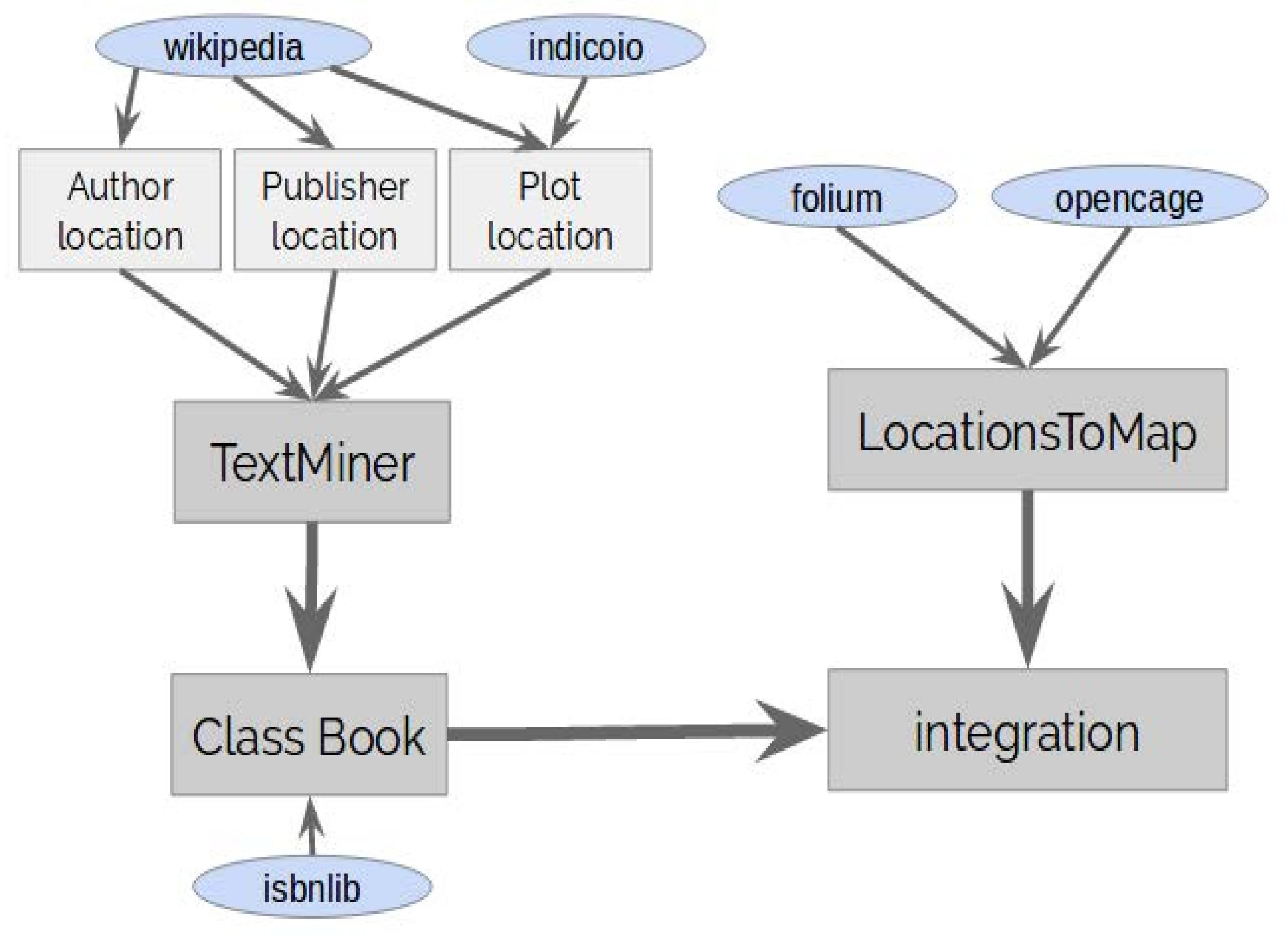
Code Structure

1) Use Python's isbnlib library to get the basic info attached to the inputted ISBN code.

2) Use the Wikipedia library to generate HTML code of the book's Wikipedia page and the book's publisher's Wikipedia page.

3) Use text mining techniques and the Wikipedia API to parse the HTML code of the book's page for the author and publisher locations. Use the indicoio library to parse the book page's HTML code for plot location.

4) Use Python's Folium library to plot the discovered locations on a world map.

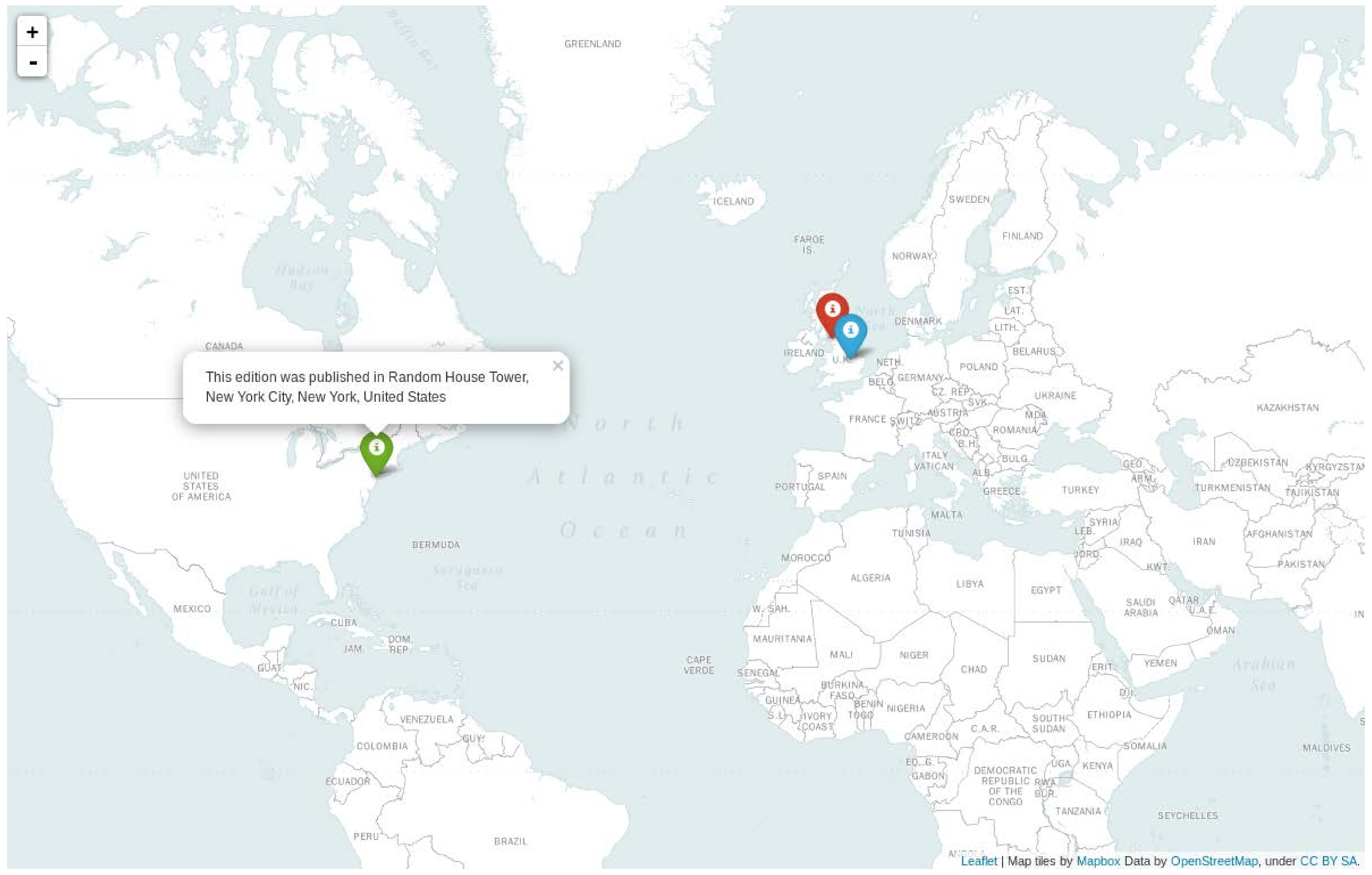


A visual representation of the python files we wrote and how they interact with each other.

What Does it Do?

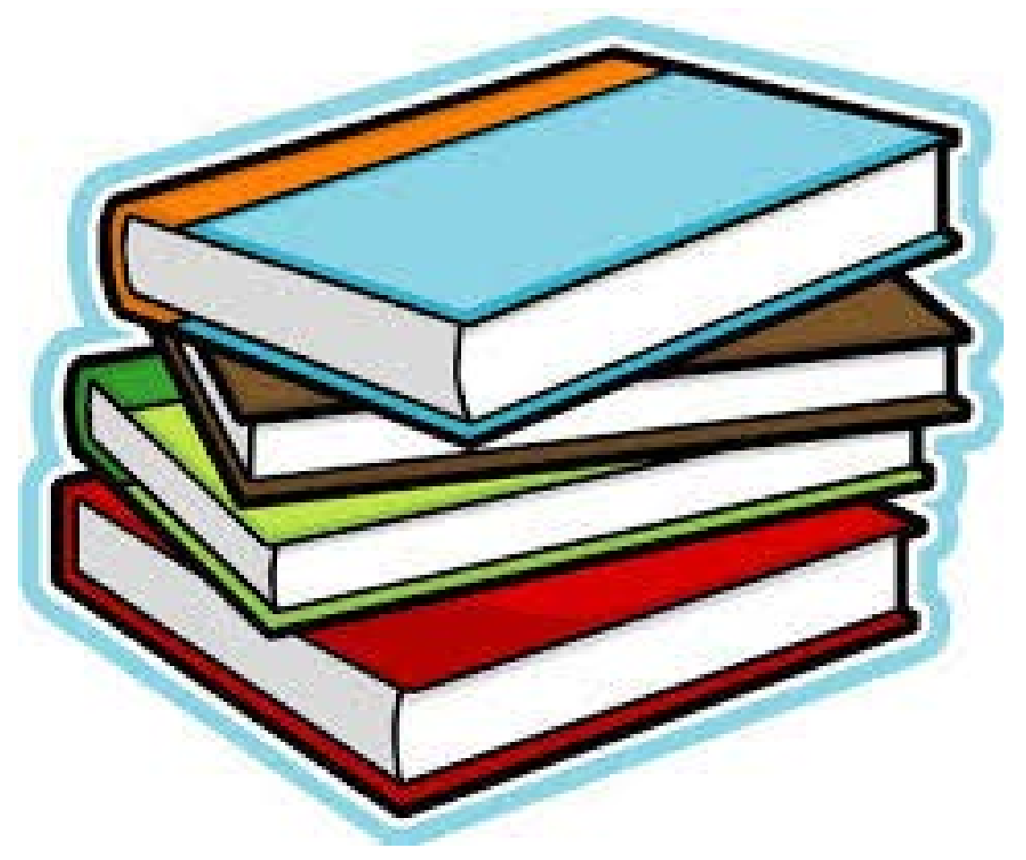
Input:
- A book title, or
- A book's ISBN code

Output: A map highlighting the place where the author is from, where the book was published, and where the the story takes place.


This edition was published in Random House Tower, New York City, New York, United States

An example: the map produced for the book Jane Eyre.

Why Does It Matter?



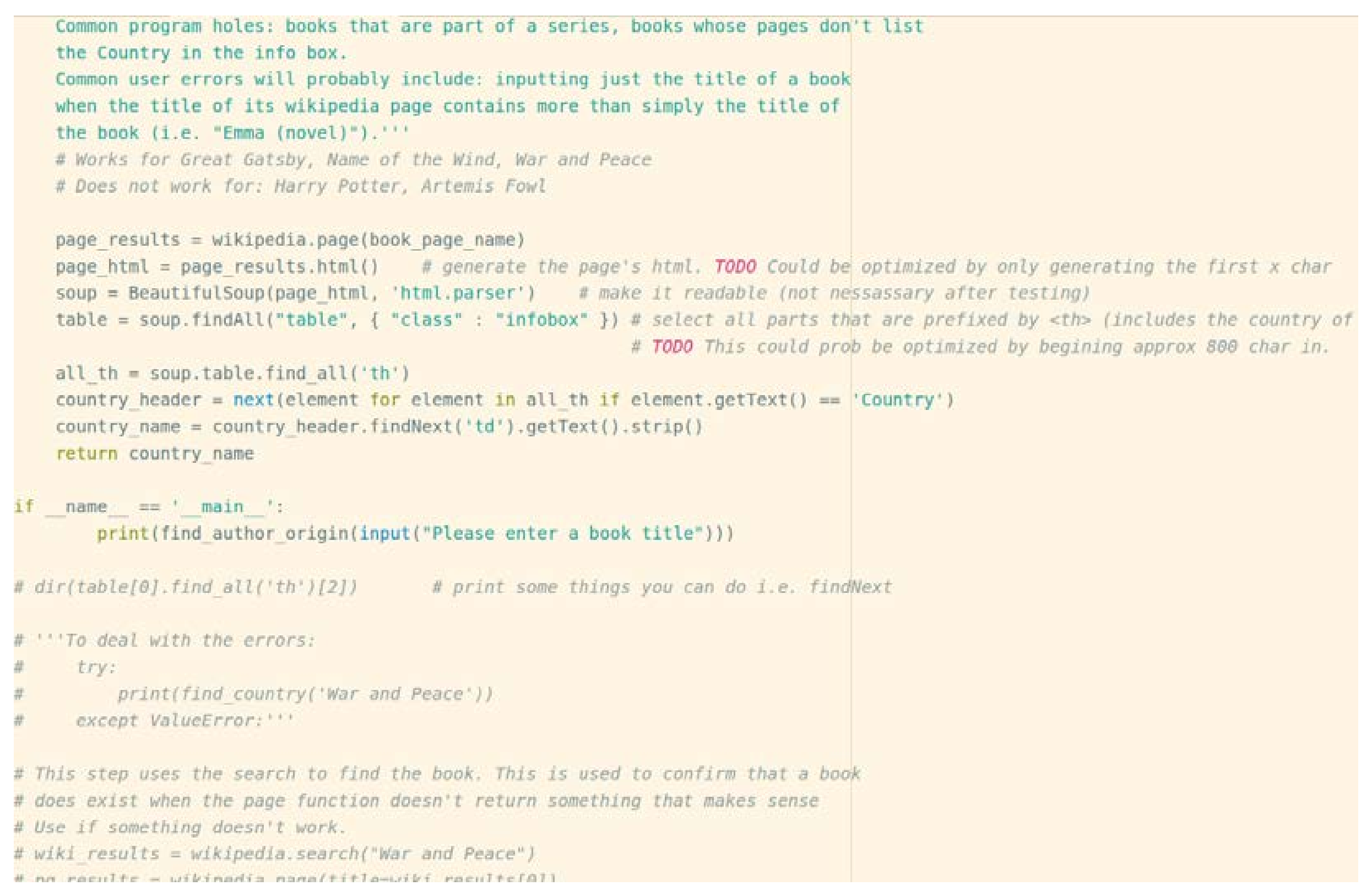
In a world where a global mindset and education are highly relevant conversations, our project educates users about where their books come from using a visual, interactive interface. It also provides a visual representation of the way in which goods, knowledge, and stories make their way around the world.

Sources:
http://nur.hmu.edu.krd/
https://pages.shanti.virginia.edu/mcintireabroad/2012/08/08/new-global-commerce-immersion-courses-announced-for-2013/

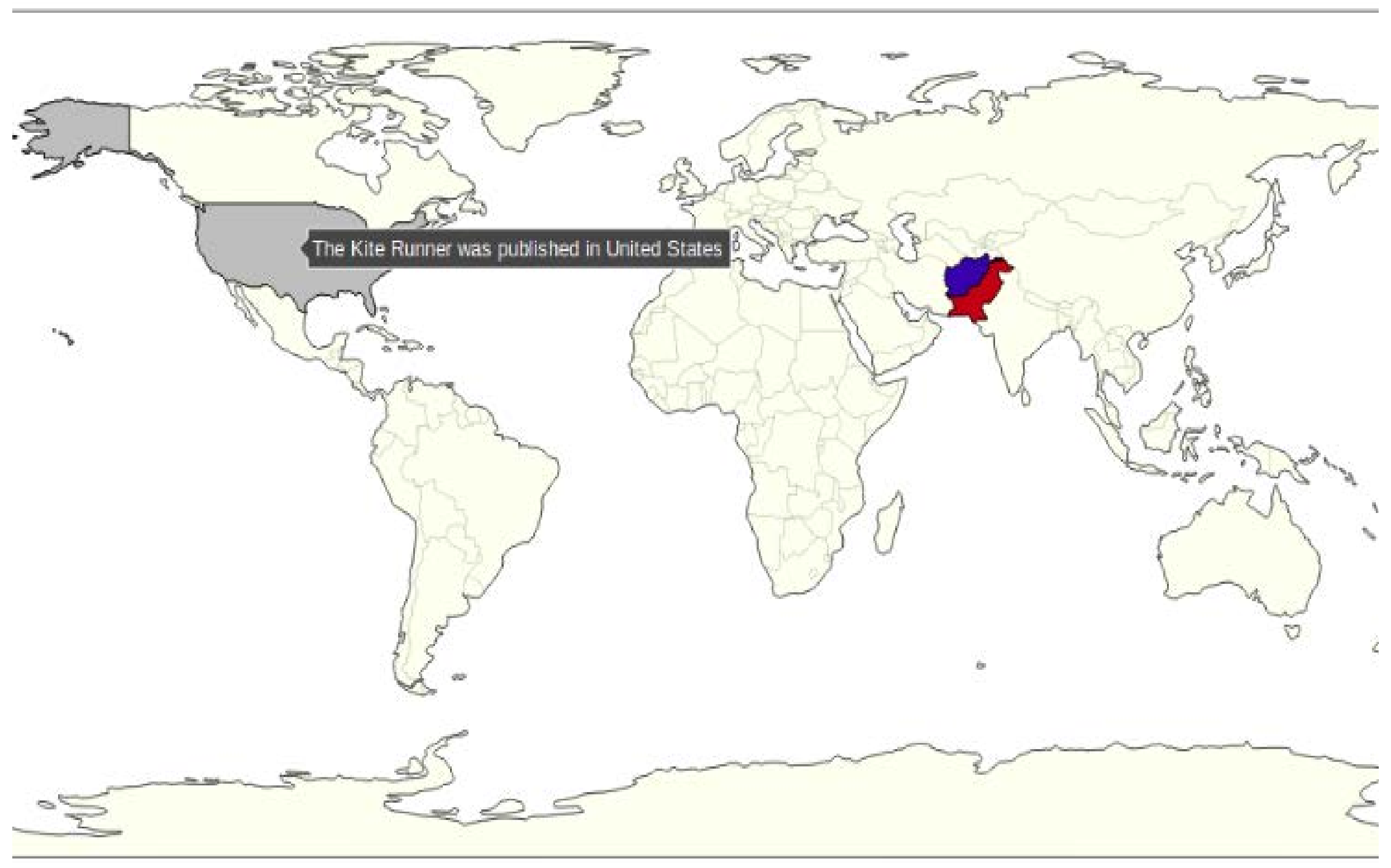
Process

We started out with just 3 things: a desire to work with maps and visualization, a curiosity about where things come from, and a barcode scanner.

This eventually evolved into the "Where's That Book From" program we present to you today, but not without some battles.


```
    Common program holes: books that are part of a series, books whose pages don't list
    the Country in the info box.
    Common user errors will probably include: inputting just the title of a book
    when the title of its wikipedia page contains more than simply the title of
    the book (i.e. "Emma (novel)").'''
    # Works for Great Gatsby, Name of the Wind, War and Peace
    # Does not work for: Harry Potter, Artemis Fowl

    page_results = wikipedia.page(book_page_name)
    page_html = page_results.html()    # generate the page's html. TODO Could be optimized by only generating the first x char
    soup = BeautifulSoup(page_html, 'html.parser')    # make it readable (not nessassary after testing)
    table = soup.findAll("table", { "class" : "infobox" }) # select all parts that are prefixed by <th> (includes the country of
                                                           # TODO This could prob be optimized by begining approx 800 char in.
    all_th = soup.table.find_all('th')
    country_header = next(element for element in all_th if element.getText() == 'Country')
    country_name = country_header.findNext('td').getText().strip()
    return country_name

if __name__ == '__main__':
        print(find_author_origin(input("Please enter a book title")))

# dir(table[0].find_all('th')[2])       # print some things you can do i.e. findNext

# '''To deal with the errors:
#     try:
#         print(find_country('War and Peace'))
#     except ValueError:'''

# This step uses the search to find the book. This is used to confirm that a book
# does exist when the page function doesn't return something that makes sense
# Use if something doesn't work.
# wiki_results = wikipedia.search("War and Peace")
# pg_results = wikipedia.page(title=wiki_results[0])
```

Some notable struggles included working with the large but occasionally unreliable Wikipedia API, shown above, and working with the Plotly library, shown below. When we could not find a simple way to plot cities instead of just countries using Plotly, we we chose to work with the Folium plotting library instead.


The Kite Runner was published in United States