MSSP 607: Practical Programming for Data Science

Homework 2, Due November 18, 2019

Part 1: Yelp

In class, we made use of a subset of 10% of Pennsylvania takeout restaurants and their associated reviews, about 20,000 in total. Uploaded to Canvas in the Homework 2 folder are two new files, one of which contains business descriptions for each takeout restaurant in Pennsylvania (**`PA_businesses.json`**), and one of which contains all reviews for those restaurants, over 210,000 in total (**`PA_reviews_full.json`**). Import the two dataset files using Pandas and answer the following three questions about the data:

1. For each star count, 1-5, what percentage of restaurants receive that score on average? What is the average word count of reviews that give each star count?
2. Philadelphia ZIP codes begin with 19xxx, while Pittsburgh ZIP codes begin 15xxx. The rest of Pennsylvania ZIP codes begin with 16, 17, or 18. For each of the three regions, how many reviews are in our dataset? What is the mean score of reviews for each region?
3. What are some common features of restaurants that receive higher-scoring reviews? This can be extracted either from attributes of the restaurant itself or from the review texts.

Part 2: Wikipedia

This homework will require you to use HTTP requests to receive JSON objects, and clean real HTML data from the open web. In particular, we will be scraping content from Wikipedia. The English-language Wikipedia has over 5 million articles, and about 0.1% of those have been reviewed by the community as **Featured Articles**. The full list of all featured articles is at the following URL:

https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

Wikipedia makes all public edits in the history of the wiki available through a public API, documented at https://www.mediawiki.org/wiki/API:Main_page.

IMPORTANT: In the homework assignment, I've already included the function **`page_text()`** in the homework file **`wiki_api.py`**, which will get the plain text of any Wikipedia page and cache the result locally. To use this function you will need to use Conda to install the **`lxml`** library locally on your computer.

The contents of individual articles can also be retrieved and manipulated. These pages often have an introductory paragraph overviewing the topic before the table of contents begins. For instance, author Ursula K. Le Guin's biographical page is a Featured Article. Here's her first paragraph:

> *Ursula Kroeber Le Guin (/ˈkroʊbər lə ˈɡwɪn/;[1] October 21, 1929 – January 22, 2018) was an American author. She is best known for her works of speculative fiction, including science fiction works set in her Hainish universe, and the Earthsea fantasy series. She was first published in 1959, and her literary career spanned nearly sixty years, yielding more than*

> *twenty novels and over a hundred short stories, in addition to poetry, literary criticism, translations, and children's books. Frequently described as an author of science fiction, Le Guin has also been called a "major voice in American Letters",[2] and herself said she would prefer to be known as an "American novelist".[3]*

1. Individual Page Scraping
Write a function **`get_featured_biographies()`** to scrape the contents of the list of featured articles and returns a list of names for all featured articles that are also biographies. Then, answer the following questions:

- How did you determine which featured articles were biographies?
- What percentage of featured articles are biographies?

2. Scraping a dataset
Next, write code that scrapes all of the individual pages for featured article biography titles in the list you created in part 1. Write a function **`get_first_paragraph(page)`** that extracts the first paragraph of each biography.

These functions will probably not be able to cover 100% of pages in you dataset; because the data is messy and formatted differently from page to page, they will fail on some of them. With the code you wrote, what percentage of infoboxes and first paragraphs were you able to scrape? What are the characteristics of the pages that your code fails to scrape?

3. Extracting information from messy content
Using regular expressions, write a new method **`get_pronouns(text)`** that determines the most common gender of pronouns in a given string of any length. Typically but not always, the three ways gender are marked in pronouns are:

- Male: he/his/him
- Female: she/her/hers
- Plural, or singular non-binary: they/them/their

Answer the following questions abut your calculations:

- What are the drawbacks of your approach, and what types of content are excluded or missed because of the choices you made?
- What percentage of biographies use he/his pronouns, she/her, or they/them pronouns?
- What percentage of pages did your code fail to parse, or have unclear gender? Why?

4. Additional analysis
Define and write a function that will extract one additional quantifiable feature of Wikipedia biographies based on the raw data you scraped. What question did you ask, and why is it interesting? Did you draw any new conclusions based on the feature you found and its distribution in your data? Share any statistics that support your analysis, and include those statistics in your final report.

5. Preparing a dataset for sharing
Then, either using Pandas or built-in data structures like dictionaries and lists, write a function **`export_dataset(df)`** that will export the values that your pronouns function calculated to a CSV or JSON file with at least three columns for each biography: the page title, the most common pronoun used in the introduction of that page, and the additional variable that you defined in part 4.

To go along with the file, write technical documentation for how the file is constructed; your intended audience is classmates or peers that could work with you. The documentation should include:

- Explanation of the meaning of each column in your CSV file.
- Explanation of what data was not successfully scraped by your dataset-building process, and what the limitations are on any future analyses.
- Instructions on how other data scientists could use your parsing code in their own work (you may include Python code samples if necessary).
- Sample code that allows future users to load your file using Python and instructions on how to run a basic analysis to confirm that they successfully downloaded the dataset.

Submission Instructions:
Your final source code should be committed to a Github repository that contains source code for all functions defined in this homework, organized and documented, along with the dataset file you generated for the final Wikipedia question 5. Make sure that either the repository is public, or if it is private, that my username (emayfield) is added as a contributor.

Your submission to Canvas should be a report containing all of your answers, analyses, and discussion, along with a link to the Github repository where I can find your source code and data.

IMPORTANT GRADING UPDATES, NOVEMBER 5, 2019:

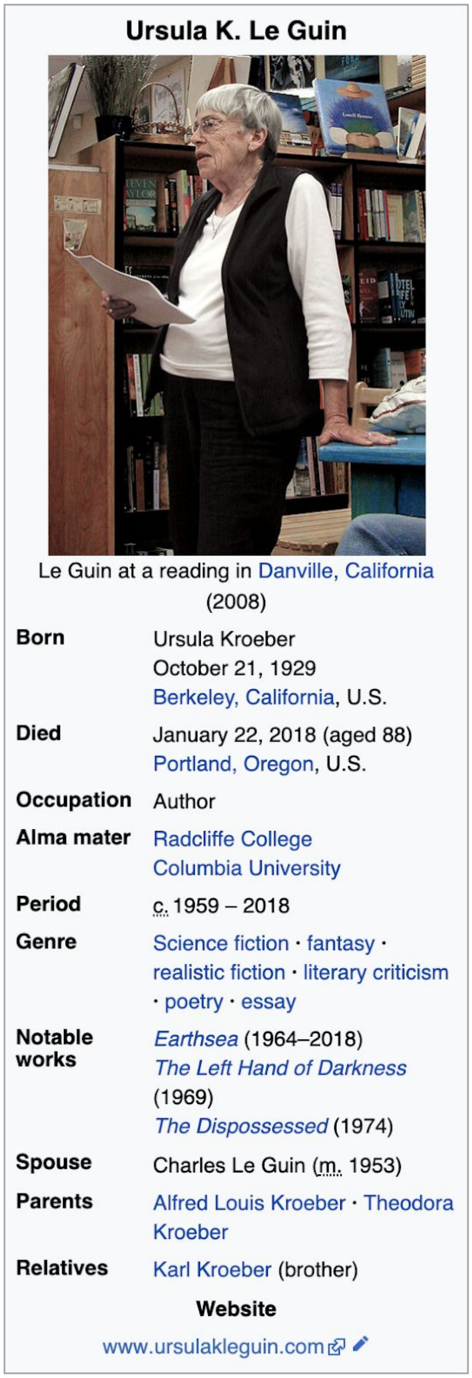
- The original version of this assignment required that files be submitted this way as a GIthub repository, but we did not spend as much time covering Github's workflow in class as I anticipated. For this reason, no points will be deducted if you submit your source code as a direct file upload in Canvas instead.

- In the first homework some students received 1 point deductions for submitting source code in alternate formats like .ipynb or exports from those formats, instead of clean .py files. However, to resolve Conda environment issues especially on Windowos, several students have found it easier to work in an alternate environment like Spyder or Jupyter Notebook instead of Visual Studio Code.

  To prevent environment configuration issues from impacting your grade, for this assignment there will be <u>no penalty</u> for submitting .ipynb or other formats of source code so long as that code or notebook is fully portable to a new computer. However, if you use a development environment other than VS Code, I will be unable to provide any technical support over email on package installation or Python interpreters.

Extra Credit: Full Infobox Extraction

Wikipedia pages also frequently contain *infoboxes*, which are typically on the right-hand side at the top of each page. The format of these pages is fairly standardized. In biographies, the infobox typically includes a profile picture with caption, some common fields like date of birth and death, and then an extremely wide list of topic-specific fields that vary from page to page.



Ursula K. Le Guin

Le Guin at a reading in Danville, California (2008)

Born	Ursula Kroeber October 21, 1929 Berkeley, California, U.S.
Died	January 22, 2018 (aged 88) Portland, Oregon, U.S.
Occupation	Author
Alma mater	Radcliffe College Columbia University
Period	c. 1959 – 2018
Genre	Science fiction · fantasy · realistic fiction · literary criticism · poetry · essay
Notable works	*Earthsea* (1964–2018) *The Left Hand of Darkness* (1969) *The Dispossessed* (1974)
Spouse	Charles Le Guin (m. 1953)
Parents	Alfred Louis Kroeber · Theodora Kroeber
Relatives	Karl Kroeber (brother)

Website

www.ursulakleguin.com

To receive up to 4 points of extra credit, in addition to extracting pronoun data from the first paragraph of each article, also extract the date of birth and date of death for each biography that contains an infobox. The provided function `page_text()` in `wiki_api.py` includes an optional parameter, **`include_tables`**, that is set to False by default. If you pass in the value True instead when you call the function you will get all the tables on the page, as well, including infoboxes.

Write a function **`get_birth_and_death(infobox)`** that extracts two integers from infoboxes, the year of birth and year of death, if either is present. The method should have a consistent and graceful way of failing with exceptions if data is missing. Include these two dates as additional columns in your shared dataset from part 4.

Answer the following question: What percentage of featured biographies describe people who are currently alive? What percentage of pages did your code fail to find data on, and why? What is the distribution of dates of birth and death on featured biographies, and what does that tell us about the site?

For an additional 4 points of extra credit, construct a new file that will contain the *full* contents of *all* the infoboxes in the Wikipedia featured biographies you extracted, including columns that may only appear rarely, like Alma Mater and Genre, not common ones like date of birth and date of death. Export the resulting dataset in either JSON or CSV format.

After you have completed generation of a full infobox dataset, it will be large and unwieldy, and other data scientists will not know what to do with it. In your report, document the assumptions you made and additional instructions for how data scientists can use the full dataset that you've generated.

MSSP 607: Homework 2 Scoring Rubric, Page 1/3

Part 1: Yelp

Individual Page Scraping (15 Points Total)			
Your Score	**Maximum Score**	**Dimension**	**Description**
	9	Technical Approach and Correctness (3 points each)	Credit will be given based on well-organized code that is easily understandable and has been cleaned up. Points may be deducted for work-in-progress lines, lack of comments especially for confusing sections, or logical flow that makes the code especially hard to follow. For the first two questions, you may lose points for answers that are clearly incorrect, even if your technical approach is reasonable. The final question is more open-ended and has no single correct answer.
	6	Written Analysis (2 points each)	Credit will be based on a clear explanation of the answers to particular questions from this section of the homework, along with an explanation of any gaps or errors that are known in the behavior of your code, especially when you made choices that don't have an obvious right answer.

Part 2: Wikipedia

Individual Page Scraping (5 Points Total)			
Your Score	**Maximum Score**	**Dimension**	**Description**
	3	Technical Approach (biography names)	
	2	Written Analysis	

Building a dataset (5 Points Total)			
Your Score	**Maximum Score**	**Dimension**	**Description**
	3	Technical Approach (get_first_paragraph)	
	2	Written Analysis	

Extracting information from messy content (5 Points Total)			
Your Score	**Maximum Score**	**Dimension**	**Description**
	3	Technical Approach (get_pronouns)	
	2	Written Analysis	

Part 5: Additional analysis (5 Points Total)			
Your Score	**Maximum Score**	**Dimension**	**Description**
	3	Technical Approach	
	2	Reasoning and Written Analysis	Up to 2 points will be given based on your choice of a reasonable question that is interesting and can be answered with the data you have. Points will be lost for analyses that are too simple (those that can be answered with a single line of code), or for analyses that calculate statistics with no motivation for why the question is interesting.

Part 4: Preparing a dataset for sharing (5 Points Total)			
Your Score	**Maximum Score**	**Dimension**	**Description**
	2	Formatting	Credit will be given for a well-formatted CSV or JSON file that contains the results of your extraction early in this part of the homework.
	2	Documentation	You’ll receive up to 2 points based on readable and intuitive documentation of your dataset.
	1	Sample Code	You’ll receive up to 1 point for including sample code with instructions on how an external user can use the data you created.

MSSP 607: Homework 2 Scoring Rubric, Page 3/3

Extra Credit - Infobox Extraction (8 Points Total)

Your Score	**Maximum Score**	**Dimension**	**Description**
	2	Technical Approach (dates of birth and death)	
	2	Written Analysis	
	2	Technical Approach (full extraction)	You’ll receive up to 2 points by coming up with a reasonable structure for your dataset to manage complex data that will, by necessity, have many fields and a lot of missing data.
	2	Documentation (full extraction)	You’ll receive up to 2 points for clear additional detail about how to use the features that you’ve extracted from the infoboxes.

Your Total Score: / 40