Overview

⚠️ Before you start ⚠️

Duplicate this Jupyter Notebook in your `week-06` folder (right-click -> Duplicate) and then add your last name to the beginning of it (ie. `blevins-hw-05.ipynb` - otherwise you risk having all your work overwritten when you try to sync your GitHub repository with your instructor's repository.

⚠️ *No, seriously: check the name of this file. Is it the copy you made and not the origina file? If so, you can proceed* ⚠️

Student Name: *Jessie Aldridge*

This will help you better learn how to use lists and loops, dictionaries, and functions in Python in order to work with textual data.

Getting to Know the Data

In this homework you're going to work with the diary of Martha Ballard, a midwife from Maine in the 1700s and early 1800s made famous by historian Laurel Ulrich's *A Midwife's Tale*. A project at George Mason University digitized her diary and put it online. I've done some research using the entries, and am supplying you with two years' worth of Ballard's diary entries (1804 and 1805). Each entry is contained in a separate text file that I've already processed and cleaned.

You can find all of her diary entries as several hundred text files in the `data` subfolder. Navigate to the `data` folder in Jupyter Lab and open up a few of the .txt files to get a feel for what sort of historical documents you will be working with and how they are structured.

1. What are some observations you have about these as **historical sources**? What jumps out at you?

For me, I feel like these diary entries are not very useful when looking at them by themselves, unless we wanted to put together a weather report of Maine weather in the

early 1800s. Much of these documents are short entries detailing the weather and events of her day.

2. Look at the filenames of Ballard's diary entries (ex. `18040323.txt`). Try to figure out: what information is stored in the file's name and how is that information structured? What does the file name tell you about the diary entry that is NOT contained in the text file itself?

These file names are going to correspond to dates, with the first 4 numbers being the year "1804", the following two being the month "03" for March, and the final two are the days. For this example, the file would describe the events on March 23, 1804.

3. Find and open the diary entry for February 5, 1804. What major event happened to Ballard's family that day?

From what I can tell, this was a very eventful day, as Martha Ballard's son got married to a Mary Farewel, as well as her being surrounded by her family.

Wrangling the Data

The goal of this section is to take your hundreds of text files worth of diary entries and add them into a dictionary. Each entry in the dictionary is going to consist of a **key** that corresponds to the name of the file (diary entry) and a **value** that contains the contents of the file (the written text of the entry).

We will be implementing the following steps across several questions:

- Look inside data folder and have Python generate a list of filenames of all the files inside that folder
- Loop through our list of filenames, open each diary entry, and read its contents
- Decide whether each diary entry was written in 1804 or 1805 and put the entry into a corresponding list

First we're going to import the `pathlib` library, which helps us more easily work with folder and files. Run this code:

In [18]:
```
from pathlib import Path
```

I've provided some code below that will allow you to create two new lists: `file_names`

and `file_paths`. The list `file_names` contains a list of all the names of the files ending in `.txt` in our `data` folder (ie. `18040101.txt`). The list `file_paths` is a string with the "path" to that file within the `data` folder (ie. `data/18040101.txt`). Run the following code cell:

In [20]:
```
txt_files = list(Path('data').glob('*.txt'))

file_paths = []

# Display the files
for file in txt_files:
    file_paths.append(str(file))
```

4. Add code to loop through the first **10 items** in your list of **file paths** and print out each of those ten file paths in order to make sure you've done this correctly.

In [22]:
```
# Your code here

for i in range(10):
    print(file_paths[i])
```

```
data/18040608.txt
data/18050427.txt
data/18040620.txt
data/18040813.txt
data/18040807.txt
data/18050619.txt
data/18050625.txt
data/18040422.txt
data/18050802.txt
data/18050816.txt
```

We're eventually going to open all of the files in your directory, but with the principal "start small" let's start by just opening and reading just **one** of the diary entry files from January 1, 1804. Run the code cell below:

In [24]:
```
diary_text=open('data/18040101.txt', encoding='utf-8').read()
print(diary_text)
```

```
Cloudy, Snowd at night. mr Ballard and Ephraim to meeting. I have been unwel
l. Son Jonathan, his wife and 6 children Sup#t$ here. we had a puding and ro
ast Spare rib. I was very unwell all nigh#t$ but, as is usual, did with out
much Care taken of me. Rachel to bed at 8 oClock. at home, very unwell.
```

5. Open, read, and print out the contents of the **February 5, 1804** diary entry.

In [26]:
```
diary_text=open('data/18040205.txt', encoding='utf-8').read()
print(diary_text)
```

```
Clear. Son*s Pollard and Lambard, their wives and par#t$ of their children C
ame here. Rhoda, Hannah, Samuel & Dolly tarried here, their parents went to
meeting. mr Black and Oldes#t$ Daughter Came with them after meeting and par
took with me of a Turkey my husband Sent to me Since he went from home. Son
Ephraim and Mary Farewel were Joind in wedlock this evening. at home. my chi
ldren here,mr Black allso. Son Ephraim was Married to Mary Farewel, Oldes#t$
Daug#t$ to y#e$ Widdow.
```

6. Let's try to isolate JUST the filename rather than the full path - ie. we want to go from `data/18040101.txt` to `18040101.txt`. Write a new function called `isolate_filename` that does the following:

- Use the `split()` function to separate the string of the full path into a list with two strings: `data` and `18040101.txt`. Hint: you can specify a specific letter or character to "split" it on.
- Returns the second item in that two-item list (ie. `18040101.txt`)

In [28]:
```
# Your code here

def isolate_filename(filename):
    name_parts = filename.split("/")
    correct_name = name_parts[1]
    return correct_name

test_file = "data/18040101.txt"
isolate_filename(test_file)
```

Out[28]:
```
'18040101.txt'
```

7. Let's stitch together all of our the above steps and apply them to every diary entry in the folder.

- Create an **empty dictionary** named `diary_dictionary`
- Set up a `for loop` to go through your `file_paths` list of file names (ex. `data/18040101.txt`, `data/18040102.txt`, etc.) that you generated above.
- **Inside** your `for loop` you are going to do the following:
  - Assign a new variable called `filename` that gets filled with the returned value from sending the full file path to your function `isolate_filename`
  - Assign a new variable called `diary_text` and assign it the contents of the file

using your new variable.
    - Add a new item to your dictionary, with the `filename` as the **key** (ex. `18040101.txt`) and the contents of the file (`diary_text`) as the **value**.
- Print out **the number of entries** that are now in your dictionary

In [30]:
```
#Your Code Here

diary_dictionary = {}

for file in file_paths:
    filename = isolate_filename(file)
    diary_text = open(file, "r").read()
    diary_dictionary[filename] = diary_text

print(len(diary_dictionary))
```

731

8. Complete the following with `diary_dictionary` of entries:

- Use the **key** to access and print the contents for Ballard's entry for **February 5, 1804**.
- Create a new `list` of **words** in the above entry (hint: String Methods)
- Print the number of **words** in the above entry.

In [32]:
```
#Part 1

test_file = "18040205.txt"
feb_fifth = diary_dictionary[test_file]
feb_fifth
```

Out[32]:
```
'Clear. Son*s Pollard and Lambard, their wives and par#t$ of their children
Came here. Rhoda, Hannah, Samuel & Dolly tarried here, their parents went t
o meeting. mr Black and Oldes#t$ Daughter Came with them after meeting and
partook with me of a Turkey my husband Sent to me Since he went from home.
Son Ephraim and Mary Farewel were Joind in wedlock this evening. at home. m
y children here,mr Black allso. Son Ephraim was Married to Mary Farewel, Ol
des#t$Daug#t$ to y#e$ Widdow.'
```

In [33]:
```
# Part 2

feb_fifth_words = feb_fifth.split(" ")
feb_fifth_words
```

Out[33]:
```
['Clear.',
 'Son*s',
 'Pollard',
```

```
'and',
'Lambard,',
'their',
'wives',
'and',
'par#t$',
'of',
'their',
'children',
'Came',
'here.',
'Rhoda,',
'Hannah,',
'Samuel',
'&',
'Dolly',
'tarried',
'here,',
'their',
'parents',
'went',
'to',
'meeting.',
'mr',
'Black',
'and',
'Oldes#t$',
'Daughter',
'Came',
'with',
'them',
'after',
'meeting',
'and',
'partook',
'with',
'me',
'of',
'a',
'Turkey',
'my',
'husband',
'Sent',
'to',
'me',
'Since',
'he',
'went',
'from',
'home.',
```

```
 'Son',
 'Ephraim',
 'and',
 'Mary',
 'Farewel',
 'were',
 'Joind',
 'in',
 'wedlock',
 'this',
 'evening.',
 'at',
 'home.',
 'my',
 'children',
 'here,mr',
 'Black',
 'allso.',
 'Son',
 'Ephraim',
 'was',
 'Married',
 'to',
 'Mary',
 'Farewel,',
 'Oldes#t$Daug#t$',
 'to',
 'y#e$',
 'Widdow.']
```

In [34]:
```
#Part 3

print(len(feb_fifth_words))
```

```
82
```

Bonus Question 1:

Let's say we want to do the same thing as Question 8 (finding the length of an entry) but we don't want to write the same code over and over. Review Walsh's Functions chapter. Define a new function that calculates and prints the length of any given diary entry measured by **number of words.** After you've defined the function, "call" it for the entry written on September 22, 1805.

In [36]:
```
#Your Code Here

def word_counter(filename):
```

```
    full_text = diary_dictionary[filename]
    broken_text = full_text.split(" ")
    num_of_words = len(broken_text)
    return num_of_words

word_counter("18050922.txt")
```

Out[36]: 106

Bonus Question 2:

- How long is the longest entry Ballard wrote in these years measured by the number of words?
- Which entry was it?
- Print the contents of that entry

Functions you might use:

- len()
- max()
- dictionary.values()

In [38]:
```
#Your Code Here

longest_file = None
max_words = 0

for filename, diary_text in diary_dictionary.items():
    if word_counter(filename) > max_words:
        max_words = word_counter(filename)
        longest_file = filename

split_file = longest_file.split(".")
date = split_file[0]

print(f"Out of all of the diary entries from Martha Ballard, the longest was
```

```
Out of all of the diary entries from Martha Ballard, the longest was on 1804
0317 and it was 229 words long!
```

Submission

Follow the instructions to submit the assignment on Canvas in two files (one `.ipynb` and one `.pdf`).

1. Save your notebook

2. Go to Kernel -> Restart Kernel and Run All Cells
3. Export as PDF or HTML