# 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: *Kun Cheng*
(Double-click this cell to type)

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 Navigate to the `data` folder in Jupyter Lab and open up a few of the .txt files to get a feel for what sort of historical documents you will be working with and how they are structured.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?

Martha Ballard's diary offers us a picture of the daily life of early American women, covering various aspects from housework to childbirth. She truthfully recorded the situations of birth, death and illness, demonstrating the universality of these events. In the diary, I think she paid more attention to survival and community, and seldom mentioned politics or personal reflection.

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?

The file names follow the format of "Year Month Day.txt", clearly indicating the exact date of each diary entry. I think this feature is very useful. The file names also help to organize the diaries chronologically and show whether any dates are missing.

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

On February 5, 1804, Martha Ballard's son Ephraim married Mary Farewel. The diary records that at that time, family members were all gathered at her home for a party, and then they had a meal together after attending the meeting, including the turkey sent by her husband.

# 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 file_path in file_paths[:10]:
              print(file_path)
```

```
data\18040101.txt
data\18040102.txt
data\18040103.txt
data\18040104.txt
data\18040105.txt
data\18040106.txt
data\18040107.txt
data\18040108.txt
data\18040109.txt
data\18040110.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 unwell. So
n Jonathan, his wife and 6 children Sup#t$ here. we had a puding and roast 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 Came h
ere. Rhoda, Hannah, Samuel & Dolly tarried here, their parents went to meeting. m
r 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 Fare
wel 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.
```

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:

2025/3/1 01:13

greyll-hw-05

- 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]:  def isolate_filename(file_path):
              split_path = file_path.split('/')
              return (split_path[-1])

          test_path = "data/18040101.txt"
          filename = isolate_filename(test_path)
          print(filename)
```

```
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
          #Your Code Here
              #Your Code Here
              #Your Code Here
              #Your Code Here

          #Your Code Here

          diary_dictionary = {}
          for path in file_paths:
              filename=isolate_filename(path)
              diary_text=open(path, encoding='utf-8').read()
              diary_dictionary[filename]=diary_text
          len(diary_dictionary)
```

```
Out[30]:  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)

localhost:8889/lab/tree/working-with-data/cheng-sp25-data-materials/week-06/greyll-hw-05.ipynb?                                                                    4/6

- Print the number of **words** in the above entry.

```
In [32]:  key = "18040205.txt"
          diary_entry = diary_dictionary.get(key, "Entry not found.")

          print("Diary Entry for February 5, 1804:")
          print(diary_entry)
          word_list = diary_entry.split()

          word_count = len(word_list)
          print(f"\nNumber of words in the entry: {word_count}")
```

```
Diary Entry for February 5, 1804:
Entry not found.

Number of words in the entry: 3
```

---

**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 [ ]:
```

**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 [35]:  def find_longest_entry(diary_dictionary):
              max_length = 0
              longest_entry_key = None
              longest_entry_text = ""

              for key, text in diary_dictionary.items():
                  word_count = len(text.split())

                  if word_count > max_length:
                      max_length = word_count
                      longest_entry_key = key
                      longest_entry_text = text

              if longest_entry_key:
```

```
        print(f"Longest entry is: {longest_entry_key} with {max_length} words.\n
        print("Content of the longest entry:\n")
        print(longest_entry_text)

find_longest_entry(diary_dictionary)
```

Longest entry is: data\18040317.txt with 229 words.

Content of the longest entry:

Clear Part of the day. Son & Daughter Pollard and part of thier children here. Sh
ee went on to See her Father. I went afternoon Conducted by Lemuel Witham, a lad
who is Com to work with Son Ephraim. I returnd at evening very unwell. Lemuel wen
t to take Son Lambard*s hors and Sleigh to Jones*s for Jonathan and him to Come u
p in but not finding them ready to Come returnd with it. they Came up on foot and
Jonathan Came here with ou#t$ his hat. took him from his Supper, pusht him out ad
ors, Drove him home to his house Damning and pushing him down and Struck him. Sha
w and Burr went on after to prevent his being deprived of life. I followed on fal
ling as I went till meeting Daughter Lambard was assisted by her. I reacht his ho
use, find him Cursing and Swearing he would go and giv him a Hard whiping. my Dau
ghter Lambard desired Haman to go and Conducted him to Ephraim. he went and tarri
ed all night. Son Lambard brot me home in his Sleigh. O that the God of all Mercy
would forgiv him this and all other misconduc#t$. at Shubael Pitts*s, feel very u
nwel and what a Sceanhad I to go at evening. may a Good God Support me. funeral m
r James Hinkley of Hollowell, his Death was very Suden.

# 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

```
In [ ]:
```