Homework 03

⚠ Before you start ⚠

Duplicate this Jupyter Notebook in your `week-03` folder (right-click -> Duplicate) and then add your last name to the beginning of it (ie. `hw-03-blevins.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? (ie. `hw-03-blevins.ipynb`). If so, you can proceed ⚠

Student Name: *Kun Cheng/ Grey*
(Double-click this cell to type)

The Data

You're going to analyze several historical documents in this homework. In keeping with the theme of our first unit for the semester, **Slavery and Data**, I've chosen two 19th-century narratives written by formerly enslaved people: Sojourner Truth and Henry "Box" Brown.

You should have the following files:

- `hw-03-yourlastname.ipynb` (your working version Jupyter Notebook)
- `truth.txt` (Sojourner Truth's narrative)
- `brown.txt` (Henry Brown's narrative)

Load and Process the Data

Use the `open()` and `read()` functions to get the content of each of these files into Python, assigning them the corresponding variable names of `truth_fulltext` and `brown_fulltext`.

In [9]:
```
open('truth.txt', encoding='utf-8')
open('brown.txt', encoding='utf-8')
truth_fulltext=open('truth.txt', mode='r', encoding='utf-8').read()
brown_fulltext=open('brown.txt', mode='r', encoding='utf-8').read()
```

In the next two code cells, write print() statements that:

- Print the **first 500 characters** of Truth's narrative.
- Print characters **5000 to 6000** of Brown's narrative.

Hint: use the index and slice approaches for strings: https://melaniewalsh.github.io/Intro-Cultural-Analytics/02-Python/06-String-Methods.html.

In [11]:
```
truth_500 = truth_fulltext[0:500]
print(truth_500)
```

```
NARRATIVE OF SOJOURNER TRUTH



HER BIRTH AND PARENTAGE.


THE subject of this biography, SOJOURNER TRUTH, as she now calls
herself-but whose name, originally, was Isabella-was born, as near as
she can now calculate, between the years 1797 and 1800.  She was the
daughter of James and Betsey, slaves of one Colonel Ardinburgh, Hurley,
Ulster County, New York.

Colonel  Ardinburgh belonged to that class of people called Low Dutch.


Of her first master, she can give no account, as she must have be
```

In [12]:
```
brown_5000_6000 = brown_fulltext[5000:6000]
print(brown_5000_6000)
```

```
of pity, indignation and
horror.

I first drew the breath of life in Louisa County, Va., forty-five miles
from the city of Richmond, in the year 1816. I was born a slave. Not
because at the moment of my birth an angel stood by, and declared that
such was the will of God concerning me; although in a country whose most
honored writings declare that all men have a right to liberty, given
them by their Creator, it seems strange that I, or any of my brethren,
could have been born without this inalienable right, unless God had thus
signified his departure from his usual rule, as described by our
fathers. Not, I say, on account of God’s willing it to be so, was I born
a slave, but for the reason that nearly all the people of this country
are united in legislating against heaven, and have contrived to vote
down our heavenly father’s rules, and to substitute for them, that cruel
law which binds the chains of slavery upon one sixth part of the
inhabitants of this land. I was born a slave! and wh
```

In the next code cell complete the following:

- Look at the printed out "slice" of Brown's narrative. Make a new variable and assign it a value of **Brown's birth year.**
- Make a new variable and assign it a value of: **how old Henry Brown would have been in the year 1860.**
- Write a **print statement** using your new variable that says how old Henry Brown would have been in 1860.

In [14]:
```
birth_year = 1816
age_1860 = 1860 - birth_year
print(f"Henry Brown was {age_1860} years old in 1860.")
```

```
Henry Brown was 44 years old in 1860.
```

Suppose we want to compare how long each narrative is measured by the number of lines in each text. First, use the `split()` function for each narrative to break it apart by each new line. The new line character is `\n`. Make two new variables storing a list of the broken apart text: `truth_lines` and `brown_lines`.

In [16]:
```
truth_fulltext.split('\n')
truth_lines=truth_fulltext.split('\n')
```

In [17]:
```
brown_fulltext.split('\n')
brown_lines=brown_fulltext.split('\n')
```

Which narrative has more lines? You can calculate how many lines are in each narrative through the `len()` function which will calculate the **length** of each list of lines you made in the previous section.

- Write two print() statements to show **how many lines are in each narrative**.
- Add a third print() statement that calculates **the difference between these two narratives measured by their number of lines**.

In [19]:
```
len(truth_lines)
```

Out[19]: 3627

In [20]:
```
len(brown_lines)
```

Out[20]: 2223

In [21]:
```
print(f"Truth fulltext has {len(truth_lines)} lines in total")
```

```
Truth fulltext has 3627 lines in total
```

In [22]:
```
print(f"Brown fulltext has {len(brown_lines)} lines in total")
```

```
Brown fulltext has 2223 lines in total
```

In [23]:
```
line_difference=len(truth_lines) - len(brown_lines)
print(f"Truth fulltext and Brown fulltext differ by a total of {line_difference}
```

```
Truth fulltext and Brown fulltext differ by a total of 1404 lines.
```

Combine the `len()` and `comparison` functions with an `if statement` to print either `Sojourner Truth's narrative has more lines` or `Henry Brown's narrative has more lines` based on which has more lines.]

In [25]:
```
if len(truth_lines) > len(brown_lines):
    print("Sojourner Truth's narrative has more lines.")
else:print("Henry Brown's narrative has more lines.")
```

```
Sojourner Truth's narrative has more lines.
```

Counting Word Frequency

Look at the code below from Melanie Walsh's "Anatomy of a Python Script" that she used to calculate the most frequently occurring words in a novel "The Yellow Wallpapper." You are going to use this code as a starting point but change it to apply this same approach to the two texts we've been working with. Your goal: **compare the most frequently occuring words in both Truth's narrative and Brown's narrative.**

Note: don't edit Walsh's code cell directly. Instead, copy and paste the code into **the two empty code cells below it** that you can then edit. If you accidentally overwrite it and need to find the original, you can copy it from the original tutorial.

Adjustments you'll need to make to Walsh's code:

- Open the right .txt file.
- Find the most frequent **20 words** instead of 40 words.

In [28]:
```
#Walsh's Code - copy this into a new code cell
import re
from collections import Counter

def split_into_words(any_chunk_of_text):
    lowercase_text = any_chunk_of_text.lower()
    split_words = re.split(r"\W+", lowercase_text)
    return split_words

filepath_of_text = "The-Yellow-Wallpaper_Charlotte-Perkins-Gilman.txt"
number_of_desired_words = 40

stopwords = ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers',
 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselve
 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', '
 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'do
 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'u
 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into'
 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', '
 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once',
 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'm
 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so
 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now',

full_text = open(filepath_of_text, encoding="utf-8").read()

all_the_words = split_into_words(full_text)
meaningful_words = [word for word in all_the_words if word not in stopwords]
meaningful_words_tally = Counter(meaningful_words)
most_frequent_meaningful_words = meaningful_words_tally.most_common(number_of_de

most_frequent_meaningful_words
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[28], line 26
     11 number_of_desired_words = 40
     13 stopwords = ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves',
'you', 'your', 'yours',
     14  'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her',
'hers',
     15  'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'th
emselves',
   (...)
     23  'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'sam
e', 'so',
     24  'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should',
'now', 've', 'll', 'amp', 'would', 'one']
---> 26 full_text = open(filepath_of_text, encoding="utf-8").read()
     28 all_the_words = split_into_words(full_text)
     29 meaningful_words = [word for word in all_the_words if word not in stopwor
ds]

File E:\python\Lib\site-packages\IPython\core\interactiveshell.py:324, in _modifi
ed_open(file, *args, **kwargs)
    317 if file in {0, 1, 2}:
    318     raise ValueError(
    319         f"IPython won't let you open fd={file} by default "
    320         "as it is likely to crash IPython. If you know what you are doin
g, "
    321         "you can use builtins' open."
    322     )
--> 324 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'The-Yellow-Wallpaper_Cha
rlotte-Perkins-Gilman.txt'
```

In [ ]:

```
import re
from collections import Counter

def split_into_words(any_chunk_of_text):
    lowercase_text = any_chunk_of_text.lower()
    split_words = re.split(r"\W+", lowercase_text)
    return split_words

filepath_of_text = "truth.txt"
number_of_desired_words = 20

stopwords = ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers',
 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselve
 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', '
 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'do
 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'u
 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into'
 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', '
 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once',
 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'm
 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so
 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now',

full_text = open(filepath_of_text, encoding="utf-8").read()
```

```
all_the_words = split_into_words(full_text)
meaningful_words = [word for word in all_the_words if word not in stopwords]
meaningful_words_tally = Counter(meaningful_words)
most_frequent_meaningful_words = meaningful_words_tally.most_common(number_of_de

most_frequent_meaningful_words
```

In [ ]:

```
import re
from collections import Counter

def split_into_words(any_chunk_of_text):
    lowercase_text = any_chunk_of_text.lower()
    split_words = re.split(r"\W+", lowercase_text)
    return split_words

filepath_of_text = "brown.txt"
number_of_desired_words = 20

stopwords = ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers',
 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselve
 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', '
 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'do
 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'u
 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into'
 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', '
 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once',
 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'm
 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so
 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now',

full_text = open(filepath_of_text, encoding="utf-8").read()

all_the_words = split_into_words(full_text)
meaningful_words = [word for word in all_the_words if word not in stopwords]
meaningful_words_tally = Counter(meaningful_words)
most_frequent_meaningful_words = meaningful_words_tally.most_common(number_of_de

most_frequent_meaningful_words
```

Look at the 20 most frequent words for each narrative. In the Markdown cell below, write down **three observations you have about this data.** These might be similarities between the two narratives, differences between the two, or any other patterns or questions you notice based on their word frequency.

Observation 1:

Observation 2:

Observation 3:

Bonus Questions

The text files you've used in this homework were not the original text files of these narratives. Instead, they've been cleaned by your instructor to make them shorter

and easier to analyze. Your goal is to use Python to download the original `.txt` files from the website Project Gutenberg. Adapt the code from these examples and use Python's `urllib` package to download the narratives and save them as local files named `truth-original.txt` and `brown-original.txt`.

Here are the URL's for the two original text files on Project Gutenberg:

- Truth's narrative: https://www.gutenberg.org/cache/epub/1674/pg1674.txt
- Brown's narrative: https://www.gutenberg.org/cache/epub/64992/pg64992.txt

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

Write code for the following:

- Open and read each of the new text files you just downloaded.
- Print out the **number of lines** in each of the original (newly downloaded) text files.

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

Compare the length of the original text files you just downloaded to the cleaned text files you used for the rest of the homework, measured by the number of lines.

Write two print() statements that calculate **how many lines were removed by the instructor for each narrative.**

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

What sort of text did the instructor remove? Write Python code that allows you to compare the two versions Sojourner Truth's narrative. Then write a few sentences in the empty Markdown cell below explaining what you found.

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

Write your explanation here