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: *Jessie Aldridge*

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You're going to analyze several historical documents in this homework.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')
truth_fulltext = open('truth.txt', mode='r', encoding='utf-8').read()
```

```
open('brown.txt', encoding='utf-8')
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_fulltext[0:499]
```

```
Out[11]: 'NARRATIVE OF SOJOURNER TRUTH\n\n\n\n\nHER BIRTH AND PARENTAGE.\n\n\nTHE su
         bject of this biography, SOJOURNER TRUTH, as she now calls\nherself-but who
         se name, originally, was Isabella—was born, as near as\nshe can now calcula
         te, between the years 1797 and 1800.  She was the\ndaughter of James and Be
         tsey, slaves of one Colonel Ardinburgh, Hurley,\nUlster County, New York.\n
         \nColonel  Ardinburgh belonged to that class of people called Low Dutch.\n\
         n\nOf her first master, she can give no account, as she must have b'
```

```
In [12]: brown_fulltext[4999:5999]
```

```
Out[12]: ' of pity, indignation and\nhorror.\n\nI first drew the breath of life in L
         ouisa County, Va., forty-five miles\nfrom the city of Richmond, in the year
         1816. I was born a slave. Not\nbecause at the moment of my birth an angel s
         tood by, and declared that\nsuch was the will of God concerning me; althoug
         h in a country whose most\nhonored writings declare that all men have a rig
         ht to liberty, given\nthem by their Creator, it seems strange that I, or an
         y of my brethren,\ncould have been born without this inalienable right, unl
         ess God had thus\nsignified his departure from his usual rule, as described
         by our\nfathers. Not, I say, on account of God's willing it to be so, was I
         born\na slave, but for the reason that nearly all the people of this countr
         y\nare united in legislating against heaven, and have contrived to vote\ndo
         wn our heavenly father's rules, and to substitute for them, that cruel\nlaw
         which binds the chains of slavery upon one sixth part of the\ninhabitants o
         f this land. I was born a slave! and w'
```

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]:

```
brown_birthyr = 1816
brown_age1860 = 1860 - brown_birthyr
print(f"Henry Brown would have been around the age of {brown_age1860} in the
```

```
Henry Brown would have been around the age of 44 in the year 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_lines = truth_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 [18]:

```
truth_lines_len = len(truth_lines)
brown_lines_len = len(brown_lines)
print(f"The difference in lines between these two works is {truth_lines_len
```

```
The difference in lines between these two works is 1404.
```

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 [20]:

```
if truth_lines_len > brown_lines_len:
    print("Sojourner Truth's work has more lines.")
elif brown_lines_len > truth_lines_len:
    print("Henry Brown's work has more lines.")
else:
    print("These two works have the same amount of lines! Incredible.")
```

```
Sojourner Truth's work 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 [23]:

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

In [24]:

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

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

brown_all_the_words = split_into_words(brown_full_text)
brown_meaningful_words = [word for word in brown_all_the_words if word not i
brown_meaningful_words_tally = Counter(brown_meaningful_words)
brown_most_frequent_meaningful_words = brown_meaningful_words_tally.most_com

brown_most_frequent_meaningful_words
```

Out[24]:

```
[('man', 86),
 ('slave', 85),
 ('slavery', 83),
 ('master', 81),
 ('upon', 80),
 ('slaves', 75),
 ('us', 62),
 ('god', 52),
 ('could', 52),
 ('people', 49),
 ('time', 48),
 ('south', 43),
 ('may', 43),
 ('wife', 38),
 ('men', 37),
 ('government', 33),
 ('yet', 31),
 ('made', 31),
 ('must', 31),
 ('never', 30)]
```

In [25]:

```
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

truth_filepath_of_text = "truth.txt"
number_of_desired_words = 20

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

truth_full_text = open(truth_filepath_of_text, encoding="utf-8").read()

truth_all_the_words = split_into_words(truth_full_text)
truth_meaningful_words = [word for word in truth_all_the_words if word not i
truth_meaningful_words_tally = Counter(truth_meaningful_words)
```

```
truth_most_frequent_meaningful_words = truth_meaningful_words_tally.most_com

truth_most_frequent_meaningful_words
```

```
Out[25]: [('god', 128),
          ('isabella', 114),
          ('time', 114),
          ('could', 104),
          ('master', 78),
          ('go', 69),
          ('good', 67),
          ('said', 67),
          ('mr', 67),
          ('mother', 65),
          ('see', 64),
          ('much', 57),
          ('found', 53),
          ('like', 51),
          ('never', 50),
          ('well', 50),
          ('place', 50),
          ('son', 49),
          ('little', 49),
          ('new', 48)]
```

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:

Although they both have a focus on religiosity with the mentions of God throughout, Truth's document has "God" as the most referenced word in the whole work.

Observation 2:

Brown's work is heavily laden with references to the institution of slavery, with many of the top words being directly or indirectly related to that.

Observation 3:

Sojourner Truth's breakdown of the document shows family played a large role in her life.

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 [30]:

```
import urllib.request
urllib.request.urlretrieve("https://www.gutenberg.org/cache/epub/1674/pg1674

import urllib.request
urllib.request.urlretrieve("https://www.gutenberg.org/cache/epub/64992/pg649
```

Out[30]:

```
('brown-original.txt', <http.client.HTTPMessage at 0x107b1a4e0>)
```

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 [32]:

```
open('truth-original.txt', encoding='utf-8')
truth_original_fulltext = open('truth-original.txt', mode='r', encoding='utf

open('brown-original.txt', encoding='utf-8')
brown_original_fulltext = open('brown-original.txt', mode='r', encoding='utf

truth_original_lines = truth_original_fulltext.split("\n")
brown_original_lines = brown_original_fulltext.split("\n")

truth_original_lines_len = len(truth_original_lines)
brown_original_lines_len = len(brown_original_lines)
print(f"The difference in lines between these two original works is {truth_c
```

```
The difference in lines between these two original works is 1273.
```

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 [34]:

```
print(f"Professor Blevins removed {truth_original_lines_len - truth_lines_le
```

```
Professor Blevins removed 460 lines from Truth's document, as well as 591 fr
om Brown's document! How could he...
```

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 [36]:

```
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

truth_original_filepath_of_text = "truth-original.txt"
number_of_desired_words = 20

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

truth_original_full_text = open(truth_original_filepath_of_text, encoding="u

truth_original_all_the_words = split_into_words(truth_original_full_text)
truth_original_meaningful_words = [word for word in truth_original_all_the_w
truth_original_meaningful_words_tally = Counter(truth_original_meaningful_wo
truth_original_most_frequent_meaningful_words = truth_original_meaningful_wo

print("These are the most frequent words in the original document by Sojourn

print(f"{truth_original_most_frequent_meaningful_words}")

print("These are the most frequent words in the edited down version by Profe

print(f"{truth_most_frequent_meaningful_words}")
```

```
These are the most frequent words in the original document by Sojourner Trut
h:
[('god', 128), ('isabella', 118), ('time', 114), ('could', 105), ('gutenber
g', 97), ('project', 88), ('master', 80), ('work', 78), ('go', 69), ('good',
67), ('see', 67), ('said', 67), ('mr', 67), ('mother', 66), ('much', 58), ('
found', 55), ('new', 53), ('like', 51), ('place', 51), ('son', 50)]
These are the most frequent words in the edited down version by Professor Bl
evins:
[('god', 128), ('isabella', 114), ('time', 114), ('could', 104), ('master',
78), ('go', 69), ('good', 67), ('said', 67), ('mr', 67), ('mother', 65), ('s
ee', 64), ('much', 57), ('found', 53), ('like', 51), ('never', 50), ('well',
50), ('place', 50), ('son', 49), ('little', 49), ('new', 48)]
```

Honestly it seems like the professor cut out some dialogue and other small random stuff that maybe did not add to the story. The amount of "God" appearing not changing makes me think that the document could be in different parts for her different parts of her life. Also the professor has cut out all of the text relating to the host of the text, Project Gutenberg, so the text is solely what was written by Truth so we could analyze it better.

```
In [ ]:
```