Sentiment Analysis of W.E.B. Du Bois' The Souls of Black Folk

by Erika Harrell

September 2021

Executive summary

The following project was conducted to determine the sentiment of text from 15 essays in W.E.B. Du Bois' book, The Souls of Black Folk. Text from each essay in the book was scraped from the Project Gutenberg webpage and each sentence was placed into a row in a data frame. The cleaned dataset contained 2,640 sentences. Compound scores of sentiment for each sentence were calculated. Based on these scores, a label was created classifying each sentence as either negative, positive, or neutral. The mean compound score for all sentences was about 0.02 showing that, on average, the sentences in the book were neutral (-0.05 < compound score < 0.05). Tokenization and word clouds revealed similar prominent words in the negative, postive, and neutral sentences (e.g. black, negro, world, south). A Naive Bayes classification model was trained on the data to predict sentiment in the text and generated an accuracy score of 54%. In an attempt to generate a higher accuracy score, the neutral sentences were removed from the data, leaving 1,892 cases. Another Naive Bayes classification model was trained on the revised data and generated an accuracy score of 70% meaning that the model performed better at predicting sentiment when neutral sentences were removed from the data.

Import libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import string
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from lxml import html
import requests
import nltk
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
import re
```

Web scraping to get essay titles and text

Data wrangling to put text in data frames

In [2]:

```
#Essay titles-There were 16 essays in the book.
```

```
page = requests.get('https://www.gutenberg.org/files/408/408-h/408-h.htm')
mytree = html.fromstring(page.content)
titles = mytree.xpath('body/table/tr/td/a/text()')
titles
```

Out[2]:

```
['The Forethought',
 'Of Our Spiritual Strivings',
 'Of the Dawn of Freedom',
 'Of Mr. Booker T. Washington and Others',
 'Of the Meaning of Progress',
 'Of the Wings of Atalanta',
 'Of the Training of Black Men',
 'Of the Black Belt',
 'Of the Quest of the Golden Fleece',
 'Of the Sons of Master and Man',
 'Of the Faith of the Fathers',
 'Of the Passing of the First-Born',
 'Of Alexander Crummell',
 'Of the Coming of John',
 'Of the Sorrow Songs',
 'The Afterthought']
```

In [3]:

```
#Thirteen rows were generated from "Forethought".
fore = mytree.xpath('body/div[@class="chapter"]/p/text()')[0:4]
fore=[a.replace("\r\n"," ") for a in fore]
fore=[b.replace("Mr.","Mr") for b in fore]
fore=[c.replace("Mrs.","Mrs") for c in fore]
fore=[d.replace("MRS.","MRS") for d in fore]
fore=[e.replace("MR.","MR") for e in fore]
fore=[g.strip() for g in fore]
fore1=[re.split(r'\. |\? |\! |\" ',h) for h in fore]
fore = []
for inner_list in fore1:
    for ele in inner_list:
        fore.append(ele)
title=[]
i=1
while i <= len(fore):
    title.append(titles[0])
    i+=1
dict={'text':fore,'title':title}
fore_df=pd.DataFrame(dict)
fore_df
```

Out[3]:

	text	title
0	Herein lie buried many things which if read wi...	The Forethought
1	This meaning is not without interest to you, G...	The Forethought
2	I pray you, then, receive my little book in al...	The Forethought
3	I have sought here to sketch, in vague, uncert...	The Forethought
4	First, in two chapters I have tried to show wh...	The Forethought
5	In a third chapter I have pointed out the slow...	The Forethought
6	Then, in two other chapters I have sketched in...	The Forethought
7	Venturing now into deeper detail, I have in tw...	The Forethought
8	Leaving, then, the white world, I have stepped...	The Forethought
9	All this I have ended with a tale twice told b...	The Forethought
10	Some of these thoughts of mine have seen the l...	The Forethought

	text	title
11	For kindly consenting to their republication h...	The Forethought
12	Before each chapter, as now printed, stands a ...	The Forethought
13	And, finally, need I add that I who speak here...	The Forethought

In [4]:

```
#"Of Our Spiritual Strivings" generated 109 observations.
e1 = mytree.xpath('body/div[@class="chapter"]/p/text()')[19:40]
e1=[a.replace("\r\n"," ") for a in e1]
e1=[b.replace("Mr.","Mr") for b in e1]
e1=[c.replace("Mrs.","Mrs") for c in e1]
e1=[d.replace("MRS.","MRS") for d in e1]
e1=[e.replace("MR.","MR") for e in e1]
e1=[g.strip() for g in e1]
e1[17]=e1[17]+' Sturm und Drang:'+ e1[18]
del e1[18]
e11=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e1]
e1 = []
for inner_list in e11:
    for ele in inner_list:
        e1.append(ele)
title=[]
i=1
while i <= len(e1):
    title.append(titles[1])
    i+=1
dict={'text':e1,'title':title}
e1_df=pd.DataFrame(dict)
e1_df
```

Out[4]:

	text	title
0	Between me and the other world there is ever a...	Of Our Spiritual Strivings
1	All, nevertheless, flutter round it	Of Our Spiritual Strivings
2	They approach me in a half-hesitant sort of wa...	Of Our Spiritual Strivings
3	they say, I know an excellent colored man in m...	Of Our Spiritual Strivings
4	At these I smile, or am interested, or reduce ...	Of Our Spiritual Strivings
...	...	...
104	Will America be poorer if she replace her brut...	Of Our Spiritual Strivings
105	or her coarse and cruel wit with loving jovial...	Of Our Spiritual Strivings
106	or her vulgar music with the soul of the Sorro...	Of Our Spiritual Strivings
107	Merely a concrete test of the underlying princ...	Of Our Spiritual Strivings
108	And now what I have briefly sketched in large ...	Of Our Spiritual Strivings

109 rows × 2 columns

In [5]:

```
#"Of the Dawn of Freedom" generated 243 observations.
e2 = mytree.xpath('body/div[@class="chapter"]/p/text()')[52:95]
e2=[a.replace("\r\n"," ") for a in e2]
e2=[b.replace("Mr.","Mr") for b in e2]
e2=[c.replace("Mrs.","Mrs") for c in e2]
e2=[d.replace("MRS.","MRS") for d in e2]
e2=[e.replace("MR.","MR") for e in e2]
```

```
e2=[g.strip() for g in e2]
e2[30]=e2[30]+' personnel ' + e2[31]
del e2[31]
e2[38]=e2[38] + ' proteges '+ e2[39]
del e2[39]
e21=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e2]
e2 = []
for inner_list in e21:
    for ele in inner_list:
        e2.append(ele)
title=[]
i=1
while i <= len(e2):
    title.append(titles[2])
    i+=1
dict={'text':e2,'title':title}
e2_df=pd.DataFrame(dict)
e2_df
```

Out[5]:

	text	title
0	The problem of the twentieth century is the pr...	Of the Dawn of Freedom
1	It was a phase of this problem that caused the...	Of the Dawn of Freedom
2	Curious it was, too, how this deeper question ...	Of the Dawn of Freedom
3	No sooner had Northern armies touched Southern...	Of the Dawn of Freedom
4	Peremptory military commands this way and that...	Of the Dawn of Freedom
...	...	...
238	I have seen a land right merry with the sun, w...	Of the Dawn of Freedom
239	And there in the King’s Highways sat and sits ...	Of the Dawn of Freedom
240	On the tainted air broods fear	Of the Dawn of Freedom
241	Three centuries’ thought has been the raising ...	Of the Dawn of Freedom
242	The problem of the Twentieth Century is the pr...	Of the Dawn of Freedom

243 rows × 2 columns

In [6]:

```
#"Of Mr. Booker T. Washington and Others" generated 157 observations.
e3 = mytree.xpath('body/div[@class="chapter"]/p/text()')[101:142]
e3=[a.replace("\r\n"," ") for a in e3]
e3=[b.replace("Mr.","Mr") for b in e3]
e3=[c.replace("Mrs.","Mrs") for c in e3]
e3=[d.replace("MRS.","MRS") for d in e3]
e3=[e.replace("MR.","MR") for e in e3]
e3=[g.strip() for g in e3]
e3[13]=e3[13] + ' through ' + e3[14]
del e3[14]
e31=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e3]
e3 = []
for inner_list in e31:
    for ele in inner_list:
        e3.append(ele)
title=[]
i=1
while i <= len(e3):
    title.append(titles[3])
    i+=1
dict={'text':e3,'title':title}
```

```
e3_df=pd.DataFrame(dict)
e3_df
```

Out[6]:

	text	title
0	Easily the most striking thing in the history ...	Of Mr. Booker T. Washington and Others
1	Washington	Of Mr. Booker T. Washington and Others
2	It began at the time when war memories and ide...	Of Mr. Booker T. Washington and Others
3	Mr Washington came, with a simple definite pro...	Of Mr. Booker T. Washington and Others
4	His programme of industrial education, concili...	Of Mr. Booker T. Washington and Others
...	...	...
152	If worse come to worst, can the moral fibre of...	Of Mr. Booker T. Washington and Others
153	The black men of America have a duty to perfor...	Of Mr. Booker T. Washington and Others
154	So far as Mr Washington preaches Thrift, Patie...	Of Mr. Booker T. Washington and Others
155	But so far as Mr Washington apologizes for inj...	Of Mr. Booker T. Washington and Others
156	By every civilized and peaceful method we must...	Of Mr. Booker T. Washington and Others

157 rows × 2 columns

In [7]:

```
#"Of the Meaning of Progress" generated 178 observations.
e4 = mytree.xpath('body/div[@class="chapter"]/p/text()')[152:177]
e4=[a.replace("\r\n"," ") for a in e4]
e4=[b.replace("Mr.","Mr") for b in e4]
e4=[c.replace("Mrs.","Mrs") for c in e4]
e4=[d.replace("MRS.","MRS") for d in e4]
e4=[e.replace("MR.","MR") for e in e4]
e4=[g.strip() for g in e4]
e41=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e4]
e4 = []
for inner_list in e41:
    for ele in inner_list:
        e4.append(ele)
title=[]
i=1
while i <= len(e4):
    title.append(titles[4])
    i+=1
dict={'text':e4,'title':title}
e4_df=pd.DataFrame(dict)
e4_df
```

Out[7]:

	text	title
0	Once upon a time I taught school in the hills ...	Of the Meaning of Progress
1	I was a Fisk student then, and all Fisk men th...	Of the Meaning of Progress
2	Young and happy, I too went, and I shall not s...	Of the Meaning of Progress
3	First, there was a Teachers’ Institute at the ...	Of the Meaning of Progress
4	A picnic now and then, and a supper, and the r...	Of the Meaning of Progress
...	...	...
173	How shall man measure Progress there where the...	Of the Meaning of Progress

	text	title
174	How many heartfuls of sorrow shall balance a b...	Of the Meaning of Progress
175	How hard a thing is life to the lowly, and yet...	Of the Meaning of Progress
176	And all this life and love and strife and fail...	Of the Meaning of Progress
177	Thus sadly musing, I rode to Nashville in the ...	Of the Meaning of Progress

178 rows × 2 columns

In [8]:

```python
#"Of the Wings of Atalanta" generated 96 observations.
e5 = mytree.xpath('body/div[@class="chapter"]/p/text()')[187:211]
e5=[a.replace("\r\n"," ") for a in e5]
e5=[b.replace("Mr.","Mr") for b in e5]
e5=[c.replace("Mrs.","Mrs") for c in e5]
e5=[d.replace("MRS.","MRS") for d in e5]
e5=[e.replace("MR.","MR") for e in e5]
e5=[g.strip() for g in e5]
e5[1]=e5[1] + ' reclame ' + e5[2]
del e5[2]
e5[11]=e5[11] + ' trivium ' + e5[12] + ' quadrivium ' + e5[13]
del e5[12]
del e5[12]
e51=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e5]
e5 = []
for inner_list in e51:
    for ele in inner_list:
        e5.append(ele)
title=[]
i=1
while i <= len(e5):
    title.append(titles[5])
    i+=1
dict={'text':e5,'title':title}
e5_df=pd.DataFrame(dict)
e5_df
```

Out[8]:

	text	title
0	South of the North, yet north of the South, li...	Of the Wings of Atalanta
1	I have seen her in the morning, when the first...	Of the Wings of Atalanta
2	Once, they say, even Atlanta slept dull and dr...	Of the Wings of Atalanta
3	And the sea cried to the hills and the hills a...	Of the Wings of Atalanta
4	It is a hard thing to live haunted by the ghos...	Of the Wings of Atalanta
...	...	...
91	And all this is gained only by human strife an...	Of the Wings of Atalanta
92	When night falls on the City of a Hundred Hill...	Of the Wings of Atalanta
93	And at its bidding, the smoke of the drowsy fa...	Of the Wings of Atalanta
94	And they say that yon gray mist is the tunic o...	Of the Wings of Atalanta
95	Fly, my maiden, fly, for yonder comes Hippomenes!	Of the Wings of Atalanta

96 rows × 2 columns

In [9]:

```
#"Of the Training of Black Men" generated 181 observations.
e6 = mytree.xpath('body/div[@class="chapter"]/p/text()')[217:251]
e6=[a.replace("\r\n"," ") for a in e6]
e6=[b.replace("Mr.","Mr") for b in e6]
e6=[c.replace("Mrs.","Mrs") for c in e6]
e6=[d.replace("MRS.","MRS") for d in e6]
e6=[e.replace("MR.","MR") for e in e6]
e6=[g.strip() for g in e6]
e6[1]=e6[1] + ' tertium quid' + e6[2]
del e6[2]
e6[4]=e6[4] + ' dilettante' + e6[5]
del e6[5]
e6[28]=e6[28] + ' dilettante ' + e6[29]
del e6[29]
e61=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e6]
e6 = []
for inner_list in e61:
    for ele in inner_list:
        e6.append(ele)
title=[]
i=1
while i <= len(e6):
    title.append(titles[6])
    i+=1
dict={'text':e6,'title':title}
e6_df=pd.DataFrame(dict)
e6_df
```

Out[9]:

```
                                                   text                           title
  0        From the shimmering swirl of waters where many...   Of the Training of Black Men
  1           Hence arises a new human unity, pulling the en...   Of the Training of Black Men
  2               The larger humanity strives to feel in this co...   Of the Training of Black Men
  3             To be sure, behind this thought lurks the afte...   Of the Training of Black Men
  4    The second thought streaming from the death-sh...   Of the Training of Black Men
...                                                         ...                             ...
176     From out the caves of evening that swing betwe...   Of the Training of Black Men
177                  So, wed with Truth, I dwell above the Veil   Of the Training of Black Men
178         Is this the life you grudge us, O knightly Ame...   Of the Training of Black Men
179           Is this the life you long to change into the d...   Of the Training of Black Men
180           Are you so afraid lest peering from this high ...   Of the Training of Black Men
```

181 rows × 2 columns

In [10]:

```
#"Of the Black Belt" generated 327 observations.
e7 = mytree.xpath('body/div[@class="chapter"]/p/text()')[260:295]
e7=[a.replace("\r\n"," ") for a in e7]
e4=[b.replace("Mr.","Mr") for b in e7]
e7=[c.replace("Mrs.","Mrs") for c in e7]
e7=[d.replace("MRS.","MRS") for d in e7]
e7=[e.replace("MR.","MR") for e in e7]
e7=[g.strip() for g in e7]
e7[18]=e7[18]+ ' nouveau riche'+ e7[19]
del e7[19]
e71=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e7]
e7 = []
```

```
for inner_list in e71:
    for ele in inner_list:
        e7.append(ele)
title=[]
i=1
while i <= len(e7):
    title.append(titles[7])
    i+=1
dict={'text':e7,'title':title}
e7_df=pd.DataFrame(dict)
e7_df
```

Out[10]:

	text	title
0	Out of the North the train thundered, and we w...	Of the Black Belt
1	Here and there lay straggling, unlovely villag...	Of the Black Belt
2	Yet we did not nod, nor weary of the scene; fo...	Of the Black Belt
3	Right across our track, three hundred and sixt...	Of the Black Belt
4	Here sits Atlanta, the city of a hundred hills...	Of the Black Belt
...	...	...
322	I worked for him thirty-seven days this spring...	Of the Black Belt
323	But he never cashed them,—kept putting me off	Of the Black Belt
324	Then the sheriff came and took my mule and cor...	Of the Black Belt
325	But furniture is exempt from seizure by law	Of the Black Belt
326	“Well, he took it just the same,” said the har...	Of the Black Belt

327 rows × 2 columns

In [11]:

```
#"Of the Quest of the Golden Fleece" generated 313 observations.
e8 = mytree.xpath('body/div[@class="chapter"]/p/text()')[306:354]
e8=[a.replace("\r\n"," ") for a in e8]
e8=[b.replace("Mr.","Mr") for b in e8]
e8=[c.replace("Mrs.","Mrs") for c in e8]
e8=[d.replace("MRS.","MRS") for d in e8]
e8=[e.replace("MR.","MR") for e in e8]
e8=[g.strip() for g in e8]
e8[4]=e8[4]+ ' regime ' + e8[5]
del e8[5]
e8[10]=e8[10]+ ' all '+ e8[11]
del e8[11]
e8[14]=e8[14]+ ' Humph! '+ e8[15]
del e8[15]
e8[15]=e8[15]+ ' i.e.'+ e8[16]
del e8[16]
e8[26]=e8[26]+ ' It’s wrong.' + e8[27]
del e8[27]
e81=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e8]
e8 = []
for inner_list in e81:
    for ele in inner_list:
        e8.append(ele)
title=[]
i=1
while i <= len(e8):
    title.append(titles[8])
    i+=1
```

```
dict={'text':e8,'title':title}
e8_df=pd.DataFrame(dict)
e8_df
```

Out[11]:

	text	title
0	Have you ever seen a cotton-field white with h...	Of the Quest of the Golden Fleece
1	I have sometimes half suspected that here the ...	Of the Quest of the Golden Fleece
2	And now the golden fleece is found; not only f...	Of the Quest of the Golden Fleece
3	For the hum of the cotton-mills is the newest ...	Of the Quest of the Golden Fleece
4	All through the Carolinas and Georgia, away do...	Of the Quest of the Golden Fleece
...	...	...
308	Now in 1890 there were forty-four holdings, bu...	Of the Quest of the Golden Fleece
309	The great increase of holdings, then, has come...	Of the Quest of the Golden Fleece
310	And for every land-owner who has thus hurried ...	Of the Quest of the Golden Fleece
311	Is it not strange compensation	Of the Quest of the Golden Fleece
312	The sin of the country districts is visited on...	Of the Quest of the Golden Fleece

313 rows × 2 columns

In [12]:

```
#"Of the Sons of Master and Man" generated 197 observations.
e9 = mytree.xpath('body/div[@class="chapter"]/p/text()')[359:396]
e9=[a.replace("\r\n"," ") for a in e9]
e9=[b.replace("Mr.","Mr") for b in e9]
e9=[c.replace("Mrs.","Mrs") for c in e9]
e9=[d.replace("MRS.","MRS") for d in e9]
e9=[e.replace("MR.","MR") for e in e9]
e9=[g.strip() for g in e9]
e9[2]=e9[2]+ ' tertium quid '+e9[3]
del e9[3]
e9[8]=e9[8]+ ' personnel '+e9[9]
del e9[9]
e9[16]=e9[16]+ ' ipso facto '+e9[17]
del e9[17]
e9[24]=e9[24]+ ' all '+ e9[25]
del e9[25]
e91=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e9]
e9 = []
for inner_list in e91:
    for ele in inner_list:
        e9.append(ele)
title=[]
i=1
while i <= len(e9):
    title.append(titles[9])
    i+=1
dict={'text':e9,'title':title}
e9_df=pd.DataFrame(dict)
e9_df
```

Out[12]:

	text	title
0	The world-old phenomenon of the contact of div...	Of the Sons of Master and Man
1	Indeed, the characteristic of our age is the c...	Of the Sons of Master and Man

	text	title
2	Whatever we may say of the results of such con...	Of the Sons of Master and Man
3	War, murder, slavery, extermination, and debau...	Of the Sons of Master and Man
4	Nor does it altogether satisfy the conscience ...	Of the Sons of Master and Man
...	...	...
192	And the condition of the Negro is ever the exc...	Of the Sons of Master and Man
193	Only by a union of intelligence and sympathy a...	Of the Sons of Master and Man
194	“That mind and soul according well,	Of the Sons of Master and Man
195	May make one music as before,	Of the Sons of Master and Man
196	But vaster.”	Of the Sons of Master and Man

197 rows × 2 columns

In [13]:

```
#"Of the Faith of the Fathers" generated 168 observations.
e10 = mytree.xpath('body/div[@class="chapter"]/p/text()')[407:437]
e10=[a.replace("\r\n"," ") for a in e10]
e10=[b.replace("Mr.","Mr") for b in e10]
e10=[c.replace("Mrs.","Mrs") for c in e10]
e10=[d.replace("MRS.","MRS") for d in e10]
e10=[e.replace("MR.","MR") for e in e10]
e10=[g.strip() for g in e10]
e10
e10[27]=e10[27]+ ' Dum vivimus, vivamus' +e10[28]
del e10[28]
e101=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e10]
e10 = []
for inner_list in e101:
    for ele in inner_list:
        e10.append(ele)
title=[]
i=1
while i <= len(e10):
    title.append(titles[10])
    i+=1
dict={'text':e10,'title':title}
e10_df=pd.DataFrame(dict)
e10_df
```

Out[13]:

	text	title
0	It was out in the country, far from home, far ...	Of the Faith of the Fathers
1	The road wandered from our rambling log-house ...	Of the Faith of the Fathers
2	I was a country schoolteacher then, fresh from...	Of the Faith of the Fathers
3	To be sure, we in Berkshire were not perhaps a...	Of the Faith of the Fathers
4	And so most striking to me, as I approached th...	Of the Faith of the Fathers
...	...	...
163	Feeling deeply and keenly the tendencies and o...	Of the Faith of the Fathers
164	Between the two extreme types of ethical attit...	Of the Faith of the Fathers
165	Their churches are differentiating,—now into g...	Of the Faith of the Fathers

	text	title
166	But back of this still broods silently the dee...	Of the Faith of the Fathers
167	Some day the Awakening will come, when the pen...	Of the Faith of the Fathers

168 rows × 2 columns

In [14]:

```
#"Of the Passing of the First-Born" generated 87 observations.
e11 = mytree.xpath('body/div[@class="chapter"]/p/text()')[445:460]
e11=[a.replace("\r\n"," ") for a in e11]
e11=[b.replace("Mr.","Mr") for b in e11]
e11=[c.replace("Mrs.","Mrs") for c in e11]
e11=[d.replace("MRS.","MRS") for d in e11]
e11=[e.replace("MR.","MR") for e in e11]
e11=[g.strip() for g in e11]
e11[12]=e11[12]+ ' Thou shalt forego! '+e11[13]
del e11[13]
e111=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e11]
e11 = []
for inner_list in e111:
    for ele in inner_list:
        e11.append(ele)
title=[]
i=1
while i <= len(e11):
    title.append(titles[11])
    i+=1
dict={'text':e11,'title':title}
e11_df=pd.DataFrame(dict)
e11_df
```

Out[14]:

	text	title
0	“Unto you a child is born,” sang the bit of ye...	Of the Passing of the First-Born
1	Then the fear of fatherhood mingled wildly wit...	Of the Passing of the First-Born
2	And I thought in awe of her,—she who had slept...	Of the Passing of the First-Born
3	I fled to my wife and child, repeating the whi...	Of the Passing of the First-Born
4	Wife and child?”—fled fast and faster than boa...	Of the Passing of the First-Born
...	...	...
82	Was not the world’s alembic, Time, in his youn...	Of the Passing of the First-Born
83	Are there so many workers in the vineyard that...	Of the Passing of the First-Born
84	The wretched of my race that line the alleys o...	Of the Passing of the First-Born
85	Perhaps now he knows the All-love, and needs n...	Of the Passing of the First-Born
86	Sleep, then, child,—sleep till I sleep and wak...	Of the Passing of the First-Born

87 rows × 2 columns

In [15]:

```
#"Of Alexander Crummell" generated 152 observations.
e12 = mytree.xpath('body/div[@class="chapter"]/p/text()')[467:499]
e12=[a.replace("\r\n"," ") for a in e12]
e12=[b.replace("Mr.","Mr") for b in e12]
e12=[c.replace("Mrs.","Mrs") for c in e12]
e12=[d.replace("MRS.","MRS") for d in e12]
```

```
e12=[e.replace("MR.","MR") for e in e12]
e12=[g.strip() for g in e12]
e12[8]=e12[8]+ ' you '+e12[9]
del e12[9]
e12[9]=e12[9]+ ' No '+e12[10]
del e12[10]
e12[11]=e12[11]+ ' expect?'+e12[12]
del e12[12]
e121=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e12]
e12 = []
for inner_list in e121:
    for ele in inner_list:
        e12.append(ele)
title=[]
i=1
while i <= len(e12):
    title.append(titles[12])
    i+=1
dict={'text':e12,'title':title}
e12_df=pd.DataFrame(dict)
e12_df
```

Out[15]:

	text	title
0	This is the story of a human heart,—the tale o...	Of Alexander Crummell
1	Three temptations he met on those dark dunes t...	Of Alexander Crummell
2	Above all, you must hear of the vales he cross...	Of Alexander Crummell
3	I saw Alexander Crummell first at a Wilberforc...	Of Alexander Crummell
4	Tall, frail, and black he stood, with simple d...	Of Alexander Crummell
...	...	...
147	He sat one morning gazing toward the sea	Of Alexander Crummell
148	He smiled and said, “The gate is rusty on the ...	Of Alexander Crummell
149	That night at star-rise a wind came moaning ou...	Of Alexander Crummell
150	I wonder where he is to-day	Of Alexander Crummell
151	I wonder if in that dim world beyond, as he ca...	Of Alexander Crummell

152 rows × 2 columns

In [16]:

```
#"Of the Coming of John" generated 273 observations.
e13 = mytree.xpath('body/div[@class="chapter"]/p/text()')[509:569]
e13=[a.replace("\r\n"," ") for a in e13]
e13=[b.replace("Mr.","Mr") for b in e13]
e13=[c.replace("Mrs.","Mrs") for c in e13]
e13=[d.replace("MRS.","MRS") for d in e13]
e13=[e.replace("MR.","MR") for e in e13]
e13=[g.strip() for g in e13]
e13[12]=e13[12]+ ' will '+ e13[13]+ ' well'+ e13[14]
del e13[13]
del e13[13]
e13[28]=e13[28]+ ' I '+e13[29]
del e13[29]
e131=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e13]
e13 = []
for inner_list in e131:
    for ele in inner_list:
        e13.append(ele)
```

```
title=[]
i=1
while i <= len(e13):
    title.append(titles[13])
    i+=1
dict={'text':e13,'title':title}
e13_df=pd.DataFrame(dict)
e13_df
```

```
Out[16]:
                                               text                title
  0         Carlisle Street runs westward from the centre ...  Of the Coming of John
  1            It is a broad, restful place, with two large b...  Of the Coming of John
  2     When at evening the winds come swelling from t...  Of the Coming of John
  3         Tall and black, they move slowly by, and seem ...  Of the Coming of John
  4            Perhaps they are; for this is Wells Institute,...  Of the Coming of John
...                                                   ...                  ...
268                          “Freudig geführt, ziehet dahin.”  Of the Coming of John
269        Amid the trees in the dim morning twilight he ...  Of the Coming of John
270  Oh, how he pitied him,—pitied him,—and wondere...  Of the Coming of John
271        Then, as the storm burst round him, he rose sl...  Of the Coming of John
272                      And the world whistled in his ears.  Of the Coming of John

273 rows × 2 columns
```

```
In [17]:
#"Of the Sorrow Songs" generated 159 observations.
e14 = mytree.xpath('body/div[@class="chapter"]/p/text()')[579:643]
e14=[a.replace("\r\n"," ") for a in e14]
e14=[b.replace("Mr.","Mr") for b in e14]
e14=[c.replace("Mrs.","Mrs") for c in e14]
e14=[d.replace("MRS.","MRS") for d in e14]
e14=[e.replace("MR.","MR") for e in e14]
e14=[g.strip() for g in e14]
e141=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in e14]
e14 = []
for inner_list in e141:
    for ele in inner_list:
        e14.append(ele)
title=[]
i=1
while i <= len(e14):
    title.append(titles[14])
    i+=1
dict={'text':e14,'title':title}
e14_df=pd.DataFrame(dict)
e14_df
```

```
Out[17]:
                                              text               title
0     They that walked in darkness sang songs in the...  Of the Sorrow Songs
1      And so before each thought that I have written...  Of the Sorrow Songs
2          Ever since I was a child these songs have stir...  Of the Sorrow Songs
3   They came out of the South unknown to me, one ...  Of the Sorrow Songs
```

	text	title
4	Then in after years when I came to Nashville I...	Of the Sorrow Songs
...	...	...
154	Even so is the hope that sang in the songs of ...	Of the Sorrow Songs
155	If somewhere in this whirl and chaos of things...	Of the Sorrow Songs
156	Free, free as the sunshine trickling down the ...	Of the Sorrow Songs
157	My children, my little children, are singing t...	Of the Sorrow Songs
158	And the traveller girds himself, and sets his ...	Of the Sorrow Songs

159 rows × 2 columns

In [18]:

```
#"The Afterthought" generated 4 observations.
after = mytree.xpath('body/div[@class="chapter"]/p/text()')[643]
after=after.split('.')
after=[a.replace("\r\n"," ") for a in after]
after=[b.replace("Mr.","Mr") for b in after]
after=[c.replace("Mrs.","Mrs") for c in after]
after=[d.replace("MRS.","MRS") for d in after]
after=[e.replace("MR.","MR") for e in after]
after=[g.strip() for g in after]
after1=[re.split(r'\. |\? |\! |\" |\.\” |\?\” ',h) for h in after]
after = []
for inner_list in after1:
    for ele in inner_list:
        after.append(ele)
title=[]
i=1
while i <= len(after):
    title.append(titles[15])
    i+=1
dict={'text':after,'title':title}
after_df=pd.DataFrame(dict)
after_df
```

Out[18]:

	text	title
0	Hear my cry, O God the Reader; vouchsafe that ...	The Afterthought
1	Let there spring, Gentle One, from out its lea...	The Afterthought
2	Let the ears of a guilty people tingle with tr...	The Afterthought
3	Thus in Thy good time may infinite reason turn...	The Afterthought

In [19]:

```
#Merging all dataframes created from the essay text created a single data frame with 2,65
data=fore_df.append([e1_df,e2_df,e3_df,e4_df,e5_df,e6_df,e7_df,e8_df,e9_df,e10_df,e11_df,e
data
```

Out[19]:

	text	title
0	Herein lie buried many things which if read wi...	The Forethought
1	This meaning is not without interest to you, G...	The Forethought
2	I pray you, then, receive my little book in al...	The Forethought
3	I have sought here to sketch, in vague, uncert...	The Forethought

	text	title
4	First, in two chapters I have tried to show wh...	The Forethought
...	...	...
2653	And the traveller girds himself, and sets his ...	Of the Sorrow Songs
2654	Hear my cry, O God the Reader; vouchsafe that ...	The Afterthought
2655	Let there spring, Gentle One, from out its lea...	The Afterthought
2656	Let the ears of a guilty people tingle with tr...	The Afterthought
2657	Thus in Thy good time may infinite reason turn...	The Afterthought

2658 rows × 2 columns

In [20]:

```
#The length or number of characters for each observation was obtained and stored in the le
data['length']=data['text'].apply(len)
data
```

Out[20]:

	text	title	length
0	Herein lie buried many things which if read wi...	The Forethought	146
1	This meaning is not without interest to you, G...	The Forethought	134
2	I pray you, then, receive my little book in al...	The Forethought	206
3	I have sought here to sketch, in vague, uncert...	The Forethought	135
4	First, in two chapters I have tried to show wh...	The Forethought	103
...	...	...	...
2653	And the traveller girds himself, and sets his ...	Of the Sorrow Songs	88
2654	Hear my cry, O God the Reader; vouchsafe that ...	The Afterthought	104
2655	Let there spring, Gentle One, from out its lea...	The Afterthought	116
2656	Let the ears of a guilty people tingle with tr...	The Afterthought	186
2657	Thus in Thy good time may infinite reason turn...	The Afterthought	123

2658 rows × 3 columns

Data cleaning to get rid of rows with low lengths

In [21]:

```
#Looking for rows that had no text (length of zero).
data[data['length']==0].value_counts()
```

Out[21]:

```
text  title                                 length
      Of Alexander Crummell                 0         2
      Of Mr. Booker T. Washington and Others  0         1
dtype: int64
```

In [22]:

```
#Rows with a length of zero were dropped from the data frame leaving 2,655 observations.
data=data[data['length']>0].reset_index()
data
```

Out[22]:

	index	text	title	length

	index	text	title	length
0	0	Herein lie buried many things which if read wi...	The Forethought	146
1	1	This meaning is not without interest to you, G...	The Forethought	134
2	2	I pray you, then, receive my little book in al...	The Forethought	206
3	3	I have sought here to sketch, in vague, uncert...	The Forethought	135
4	4	First, in two chapters I have tried to show wh...	The Forethought	103
...	...	...	...	...
2650	2653	And the traveller girds himself, and sets his ...	Of the Sorrow Songs	88
2651	2654	Hear my cry, O God the Reader; vouchsafe that ...	The Afterthought	104
2652	2655	Let there spring, Gentle One, from out its lea...	The Afterthought	116
2653	2656	Let the ears of a guilty people tingle with tr...	The Afterthought	186
2654	2657	Thus in Thy good time may infinite reason turn...	The Afterthought	123

2655 rows × 4 columns

In [23]:

```
#Looking at rows with low lengths of lengths of three or fewer.
data[data['length']<=3]['text']
```

Out[23]:

```
457       1
459       2
461       3
467       1
469       2
471       3
479       W
480       E
483       1
485       2
487       3
531     Yes
943      No
2136    Yes
2171      “
Name: text, dtype: object
```

In [24]:

```
#Rows with a length of three or lower were dropped, leaving 2,640 observations.
data=data[data['length']>3].reset_index()
data=data.drop(['level_0','index'],axis=1)
data
```

Out[24]:

	text	title	length
0	Herein lie buried many things which if read wi...	The Forethought	146
1	This meaning is not without interest to you, G...	The Forethought	134
2	I pray you, then, receive my little book in al...	The Forethought	206
3	I have sought here to sketch, in vague, uncert...	The Forethought	135
4	First, in two chapters I have tried to show wh...	The Forethought	103
...	...	...	...
2635	And the traveller girds himself, and sets his ...	Of the Sorrow Songs	88

	text	title	length
2636	Hear my cry, O God the Reader; vouchsafe that ...	The Afterthought	104
2637	Let there spring, Gentle One, from out its lea...	The Afterthought	116
2638	Let the ears of a guilty people tingle with tr...	The Afterthought	186
2639	Thus in Thy good time may infinite reason turn...	The Afterthought	123

2640 rows × 3 columns

Apply labels to text

In [25]:

```
#Using the Sentiment Intensity Analyzer from the Vader package, a compound score of the se
nltk.download('vader_lexicon')
vds=SentimentIntensityAnalyzer()
for lab,row in data.iterrows():
    data.loc[lab,'compoundscore']=vds.polarity_scores(row['text'])['compound']

conditions=[(data['compoundscore'] >= 0.05),
    (data['compoundscore'] > -0.05) & (data['compoundscore'] < 0.05),
    (data['compoundscore'] <= -0.05)]
choices=[0,-1,1]
data['label']=np.select(conditions,choices)
data.head()
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\E_A_H\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Out[25]:

	text	title	length	compoundscore	label
0	Herein lie buried many things which if read wi...	The Forethought	146	-0.2023	1
1	This meaning is not without interest to you, G...	The Forethought	134	-0.1040	1
2	I pray you, then, receive my little book in al...	The Forethought	206	0.9136	0
3	I have sought here to sketch, in vague, uncert...	The Forethought	135	-0.3818	1
4	First, in two chapters I have tried to show wh...	The Forethought	103	0.0000	-1

Exploratory Data Analysis

In [26]:

```
#Exploratory analysis took place once the label variable was generated. Descriptive statis
data.describe()
```

Out[26]:

	length	compoundscore	label
count	2640.000000	2640.000000	2640.000000
mean	146.829167	0.021158	0.058333
std	97.398577	0.477644	0.788564
min	4.000000	-0.991200	-1.000000
25%	79.000000	-0.296000	-1.000000
50%	128.000000	0.000000	0.000000

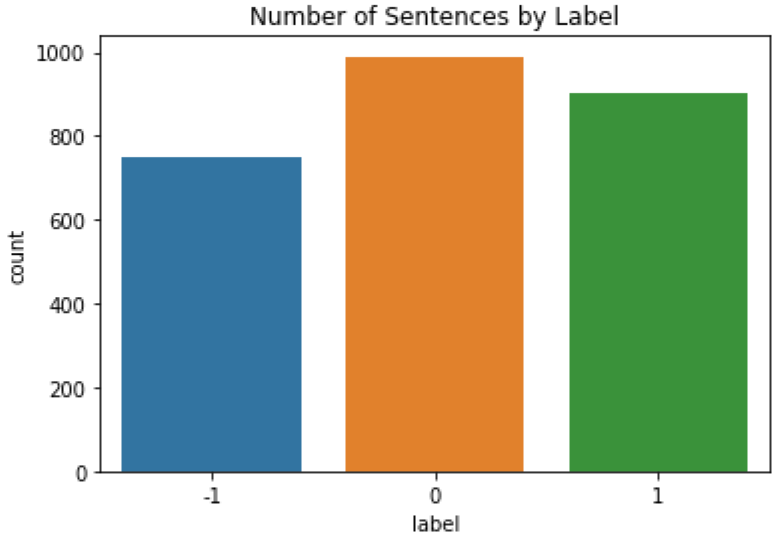
	length	compoundscore	label
75%	194.000000	0.381800	1.000000
max	882.000000	0.982900	1.000000

In [27]:

```
#Bar chart of label variable (-1 is neutral, 0 is positive and 1 is negative)
plt.title('Number of Sentences by Label')
sns.countplot(x=data['label'],label='Count')
```

Out[27]:

```
<AxesSubplot:title={'center':'Number of Sentences by Label'}, xlabel='label', ylabel='coun
t'>
```



In [28]:

```
#The label variable shows that there were more positive sentences (990) than either negat
data['label'].value_counts()
```

Out[28]:

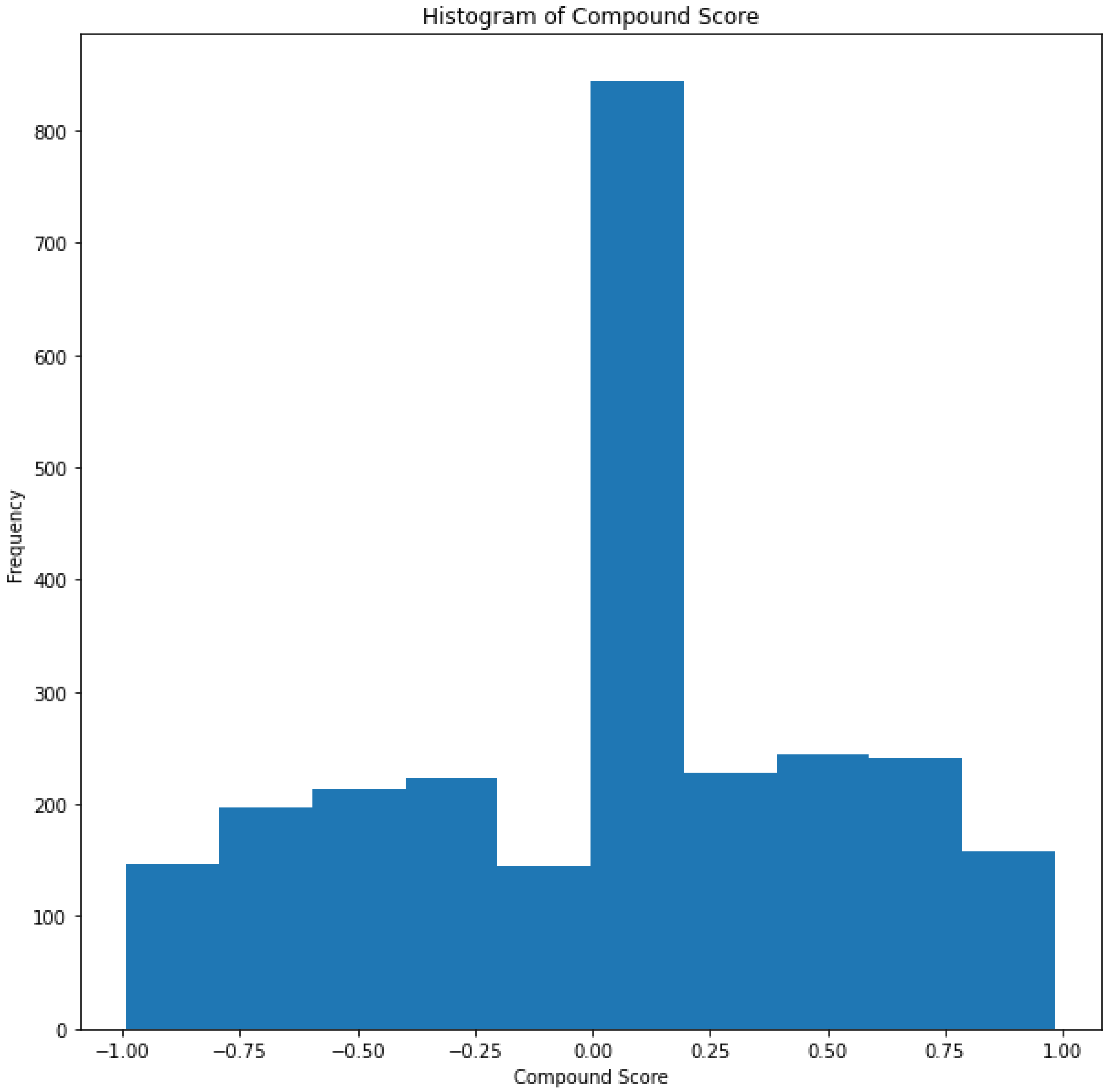
```
 0    990
 1    902
-1    748
Name: label, dtype: int64
```

In [29]:

```
#A histogram of compound score variable revealed a somewhat normally distributed variable.
plt.figure(figsize=(10,10))
plt.xlabel('Compound Score')
data['compoundscore'].plot(kind='hist',title='Histogram of Compound Score')
```

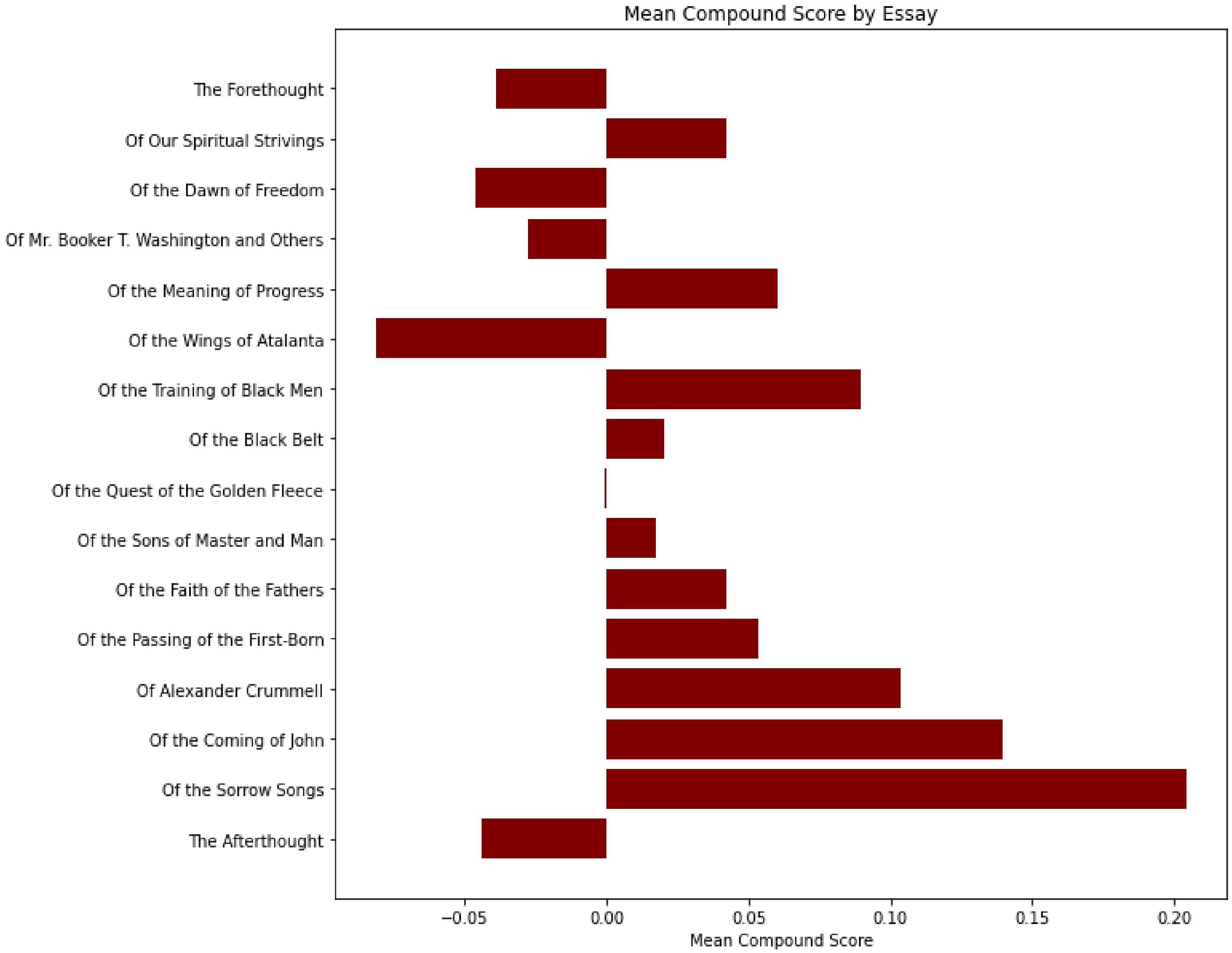
Out[29]:

```
<AxesSubplot:title={'center':'Histogram of Compound Score'}, xlabel='Compound Score', ylab
el='Frequency'>
```



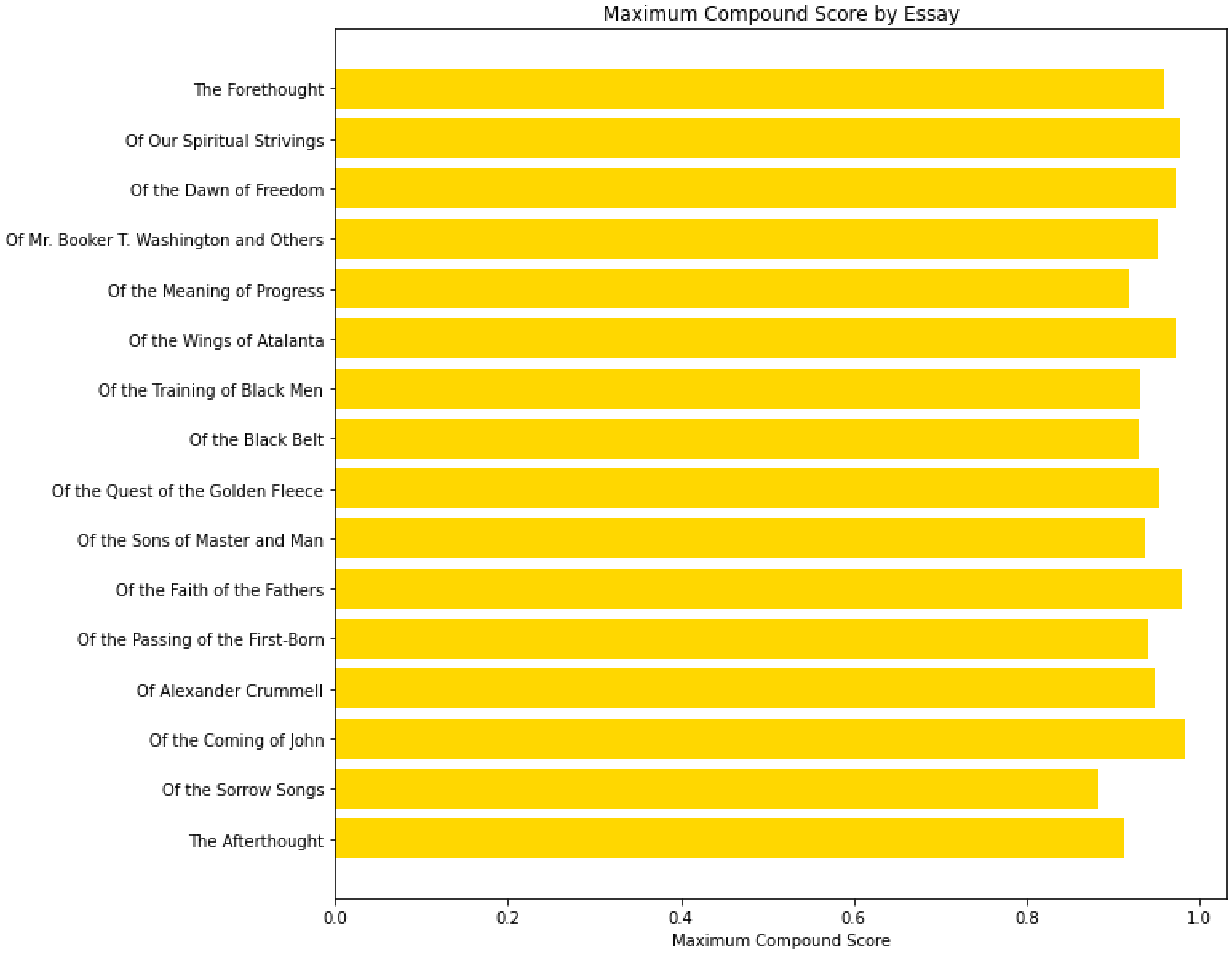
In [30]:

```
#A bar chart of the mean compound score by essay shows variety in the scores across the bo
mean_cps_by_essay=data.groupby('title')['compoundscore'].mean()
means=[mean_cps_by_essay[0],mean_cps_by_essay[1],mean_cps_by_essay[2],mean_cps_by_essay[3]
plt.figure(figsize=(10,10))
plt.barh(y=titles,width=means,height=0.8,left=0, align='center', color='maroon')
plt.gca().invert_yaxis()
plt.xlabel('Mean Compound Score')
plt.title('Mean Compound Score by Essay')
plt.show()
```



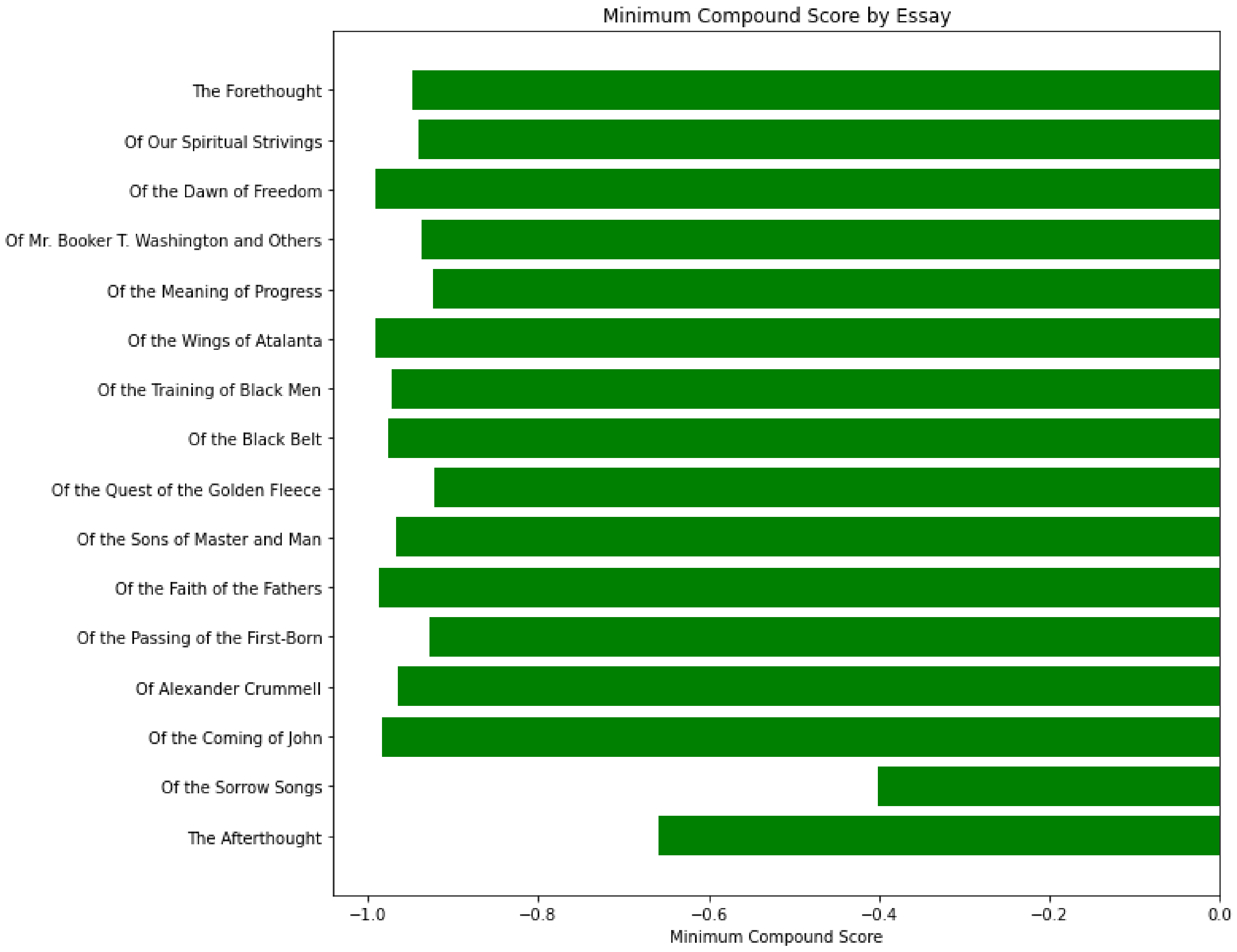
In [31]:

```
#A bar chart of maximum compound score by essay shows that for all of the essays, the high
max_cps_by_essay=data.groupby('title')['compoundscore'].max()
maximum=[max_cps_by_essay[0],max_cps_by_essay[1],max_cps_by_essay[2],max_cps_by_essay[3],m
plt.figure(figsize=(10,10))
plt.barh(y=titles,width=maximum,height=0.8,left=0, align='center', color='gold')
plt.gca().invert_yaxis()
plt.xlabel('Maximum Compound Score')
plt.title('Maximum Compound Score by Essay')
plt.show()
```



In [32]:

```
#A bar chart of minimum compound score by essay showed that for most essays in the book, 
min_cps_by_essay=data.groupby('title')['compoundscore'].min()
minimum=[min_cps_by_essay[0],min_cps_by_essay[1],min_cps_by_essay[2],min_cps_by_essay[3],m
plt.figure(figsize=(10,10))
plt.barh(y=titles,width=minimum,height=0.8,left=0, align='center', color='green')
plt.gca().invert_yaxis()
plt.xlabel('Minimum Compound Score')
plt.title('Minimum Compound Score by Essay')
plt.show()
```

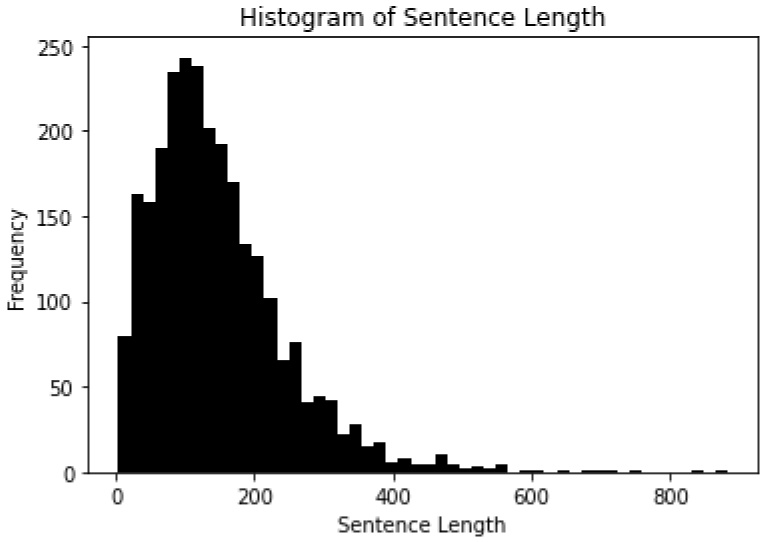


In [33]:

```
#A histogram reveled that the distribution of length variable was positively skewed.
plt.xlabel('Sentence Length')
data['length'].plot(bins=50, kind='hist', title='Histogram of Sentence Length', color='bla
```

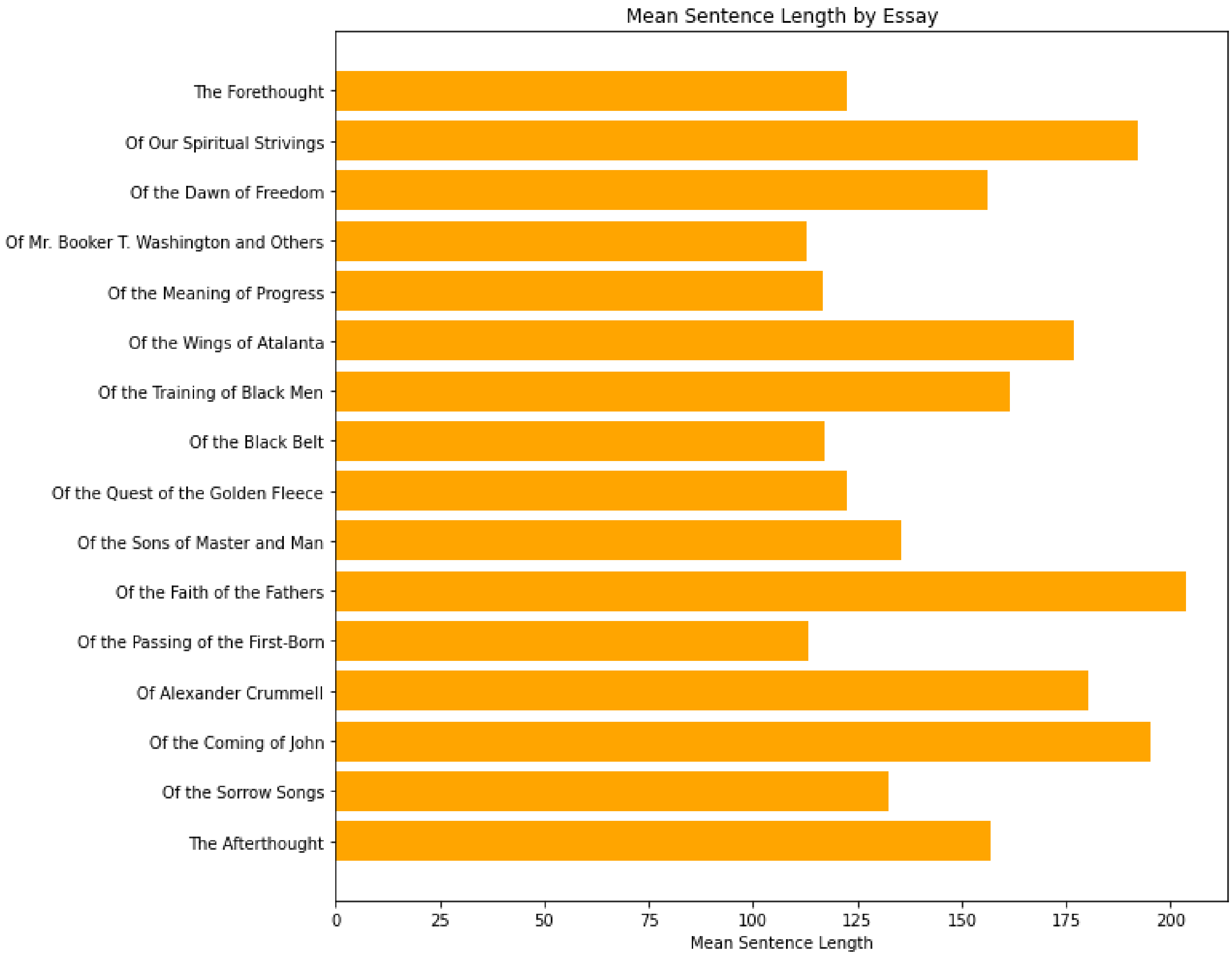
Out[33]:

```
<AxesSubplot:title={'center':'Histogram of Sentence Length'}, xlabel='Sentence Length', yl
abel='Frequency'>
```



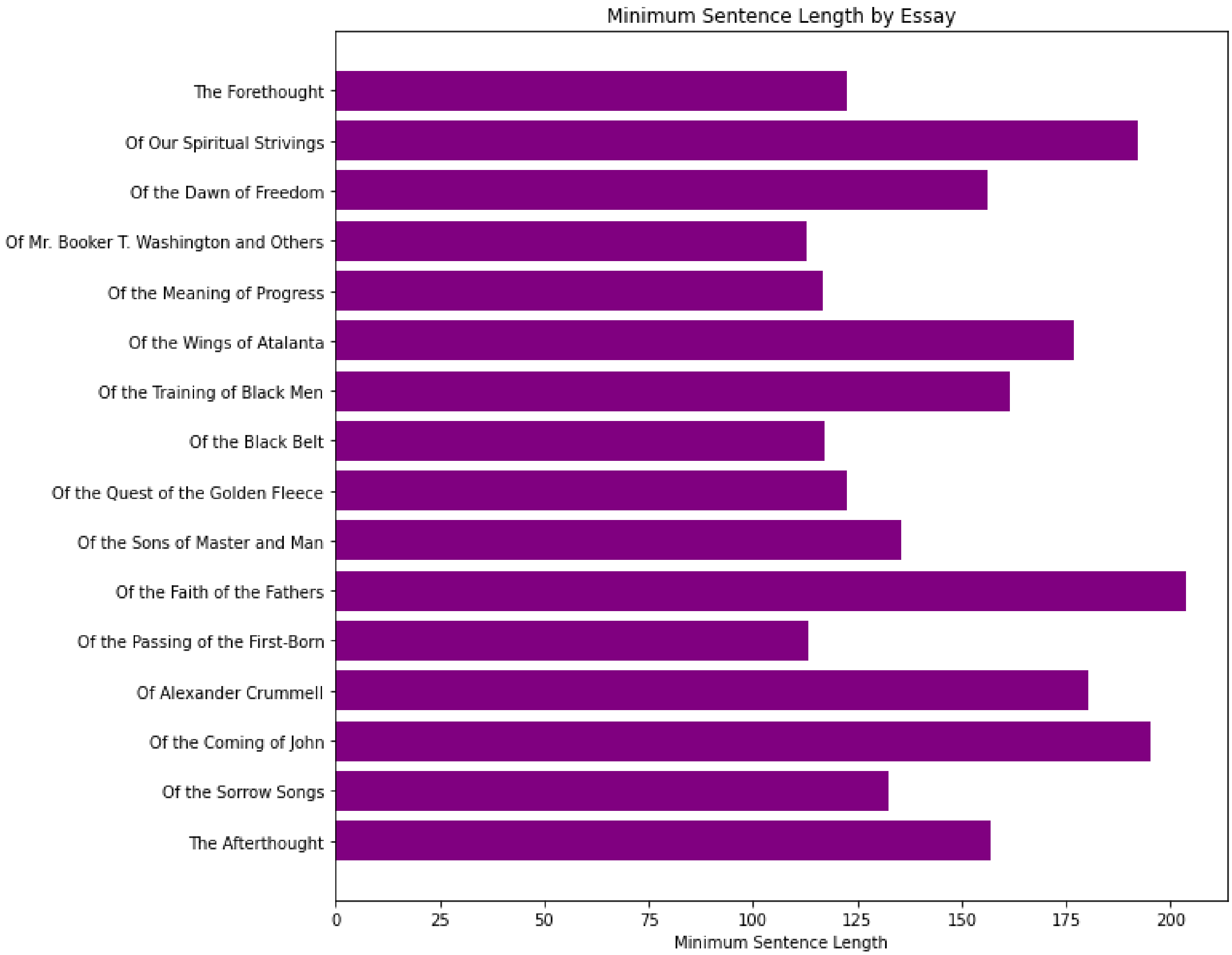
In [34]:

```
#Looking at the mean of the length variable by essay revealed that, on average, "Of the Fa
mean_length_by_essay=data.groupby("title")['length'].mean()
mean_length=[mean_length_by_essay[0],mean_length_by_essay[1],mean_length_by_essay[2],mean_
plt.figure(figsize=(10,10))
plt.barh(y=titles,width=mean_length,height=0.8,left=0, align='center', color='orange')
plt.gca().invert_yaxis()
plt.xlabel('Mean Sentence Length')
plt.title('Mean Sentence Length by Essay')
plt.show()
```



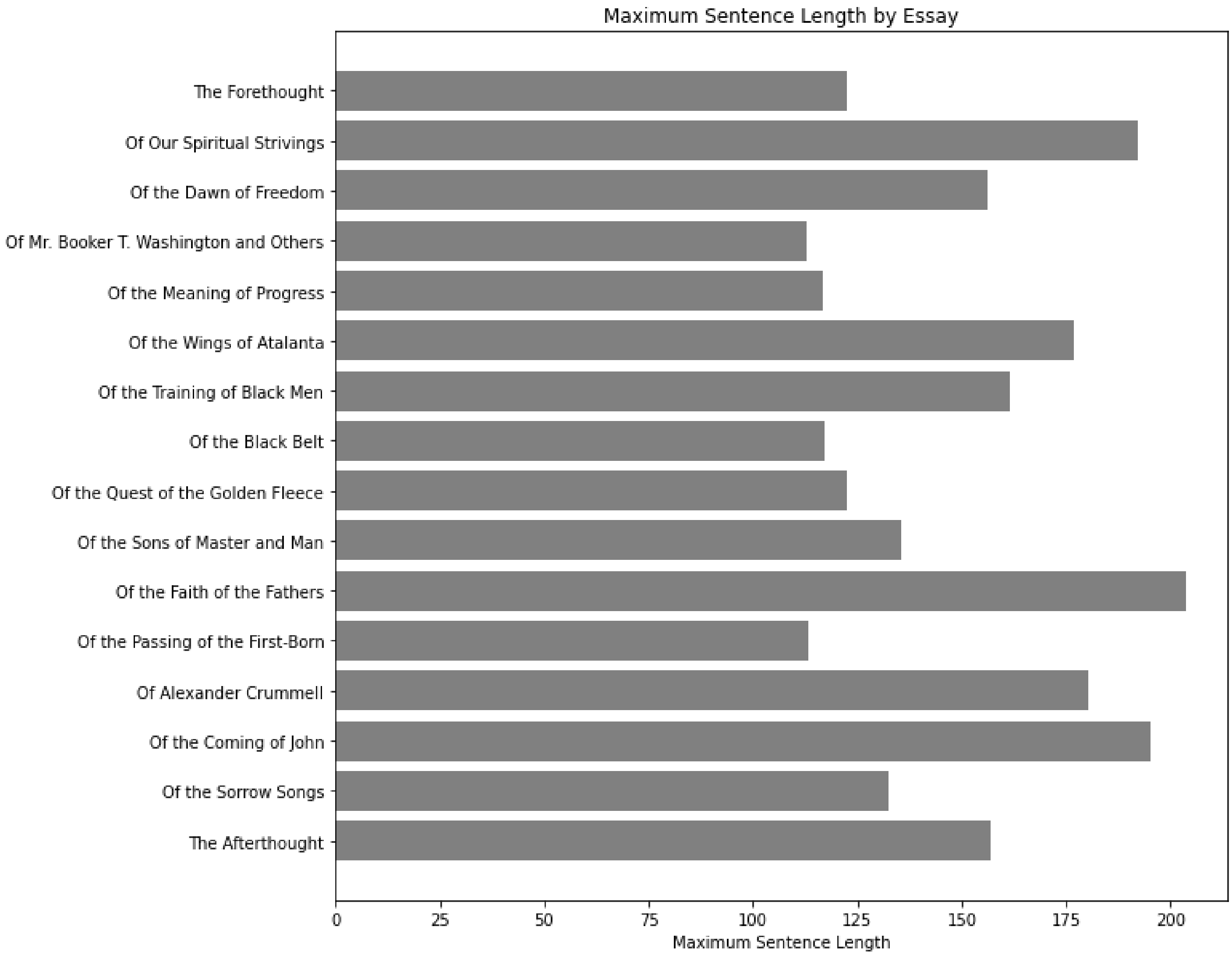
In [35]:

```
#"Of Mr. Booker T. Washington and Others" and "Of the Passing of the First-Born" seemed t
min_length_by_essay=data.groupby("title")['length'].mean()
min_length=[min_length_by_essay[0],min_length_by_essay[1],min_length_by_essay[2],min_lengt
plt.figure(figsize=(10,10))
plt.barh(y=titles,width=min_length,height=0.8,left=0, align='center', color='purple')
plt.gca().invert_yaxis()
plt.xlabel('Minimum Sentence Length')
plt.title('Minimum Sentence Length by Essay')
plt.show()
```



In [36]:

```
#On average, "On the Faith of Our Fathers" and "Of Our Spiritual Strivings" had the longes
max_length_by_essay=data.groupby("title")['length'].mean()
max_length=[max_length_by_essay[0],max_length_by_essay[1],max_length_by_essay[2],max_lengt
plt.figure(figsize=(10,10))
plt.barh(y=titles,width=max_length,height=0.8,left=0, align='center', color='gray')
plt.gca().invert_yaxis()
plt.xlabel('Maximum Sentence Length')
plt.title('Maximum Sentence Length by Essay')
plt.show()
```



Data cleaning-removing punctuation & stopwords & prep for tokenization

In [37]:

```
# A pipeline that removed punctuation and stopwords was created to prepare the textfor tok
def message_cleaning(message):
    sentences=message.tolist()
    sentences_as_one_string = " ".join(sentences)
    Text_punc_removed1 = [char for char in sentences_as_one_string if char not in string.p
    Text_punc_removed2=''.join(Text_punc_removed1)
    Text_punc_removed3=list(Text_punc_removed2.split(" "))
    Text_punc_removed4 = [w.lower() for w in Text_punc_removed3]
    filtered_text = []
    stop_words = set(stopwords.words('english'))
    for w in Text_punc_removed4:
       if w not in stop_words:
          filtered_text.append(w)
    Text_punc_stop_removed=' '.join(filtered_text)
    return Text_punc_stop_removed
```

Tokenization-entire book

In [38]:

```
#The data was cleaned with the pipeline and tokenization was conducted over the cleaned da
cleaned_text=message_cleaning(data['text'])
tokenized_word = word_tokenize(cleaned_text)
len(tokenized_word)
```

```
Out[38]: 36656
```

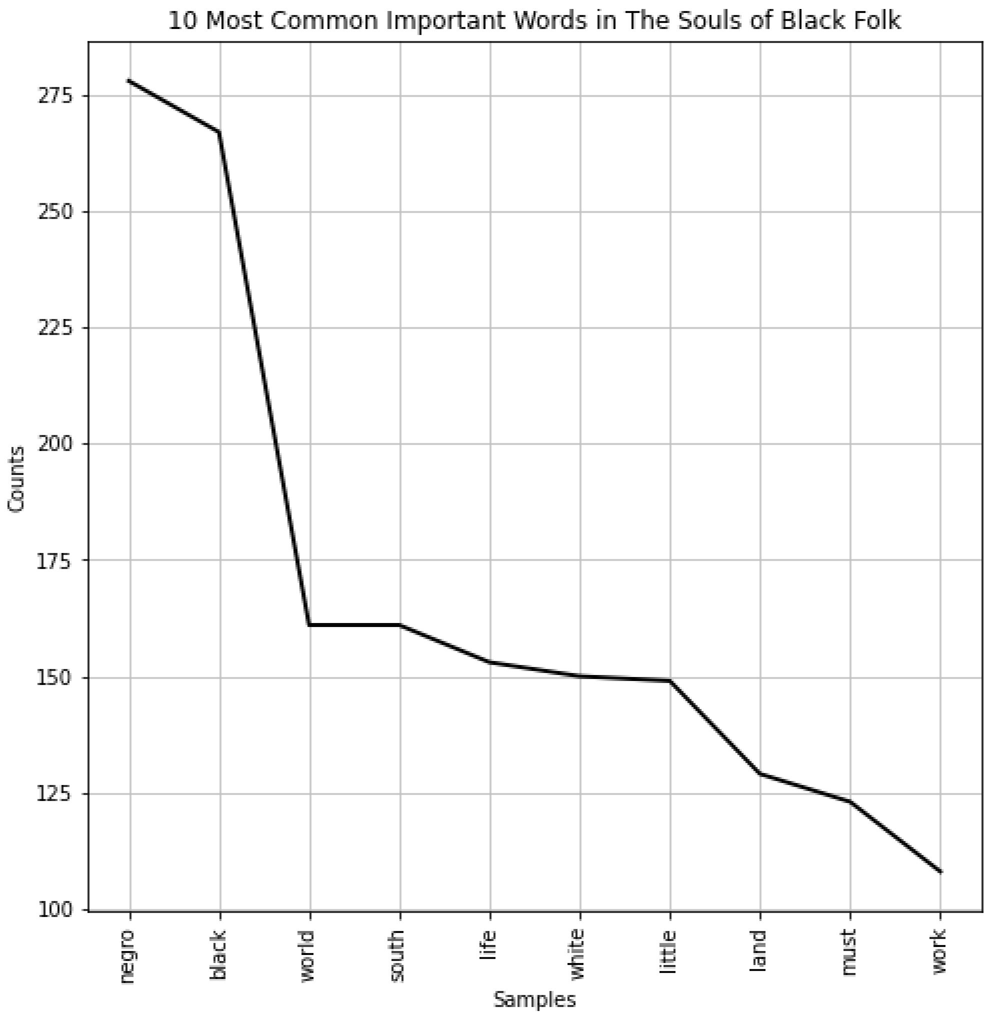
```
In [39]:
#9,041 unique tokens were found in the text.
len(set(tokenized_word))
```

```
Out[39]: 9041
```

```
In [40]:
#About one in four or 25% of the tokens were unique.
len(set(tokenized_word))/len(tokenized_word)*100
```

```
Out[40]: 24.664447839371455
```

```
In [41]:
#A frequency distribution of the tokens revealed that the 10 most common tokens in the boo
fdict={'text':tokenized_word}
filter_df=pd.DataFrame(fdict)
filter_df['length']=filter_df['text'].apply(len)
filter_df=filter_df[~filter_df.text.isin(['“','’','”'])]
filter_df=filter_df.loc[filter_df['length']>3]
f1=filter_df['text'].tolist()
fdist = FreqDist(f1)
plt.figure(figsize=(8,8))
fdist.plot(10,title="10 Most Common Important Words in The Souls of Black Folk", color='bl
```



```
Out[41]: <AxesSubplot:title={'center':'10 Most Common Important Words in The Souls of Black Folk'},
xlabel='Samples', ylabel='Counts'>
```

Tokenization of postive sentences in book

In [42]:

```
#Tokenization of the sentences labeled as positive in the book generated 16,032 tokens.
positive=data[data['label']==0]
cleaned_text_p=message_cleaning(positive['text'])
tokenized_word_p = word_tokenize(cleaned_text_p)
len(tokenized_word_p)
```

Out[42]:

```
16032
```

In [43]:

```
#Of the tokens found, 5,450 were unique.
len(set(tokenized_word_p))
```

Out[43]:

```
5450
```

In [44]:

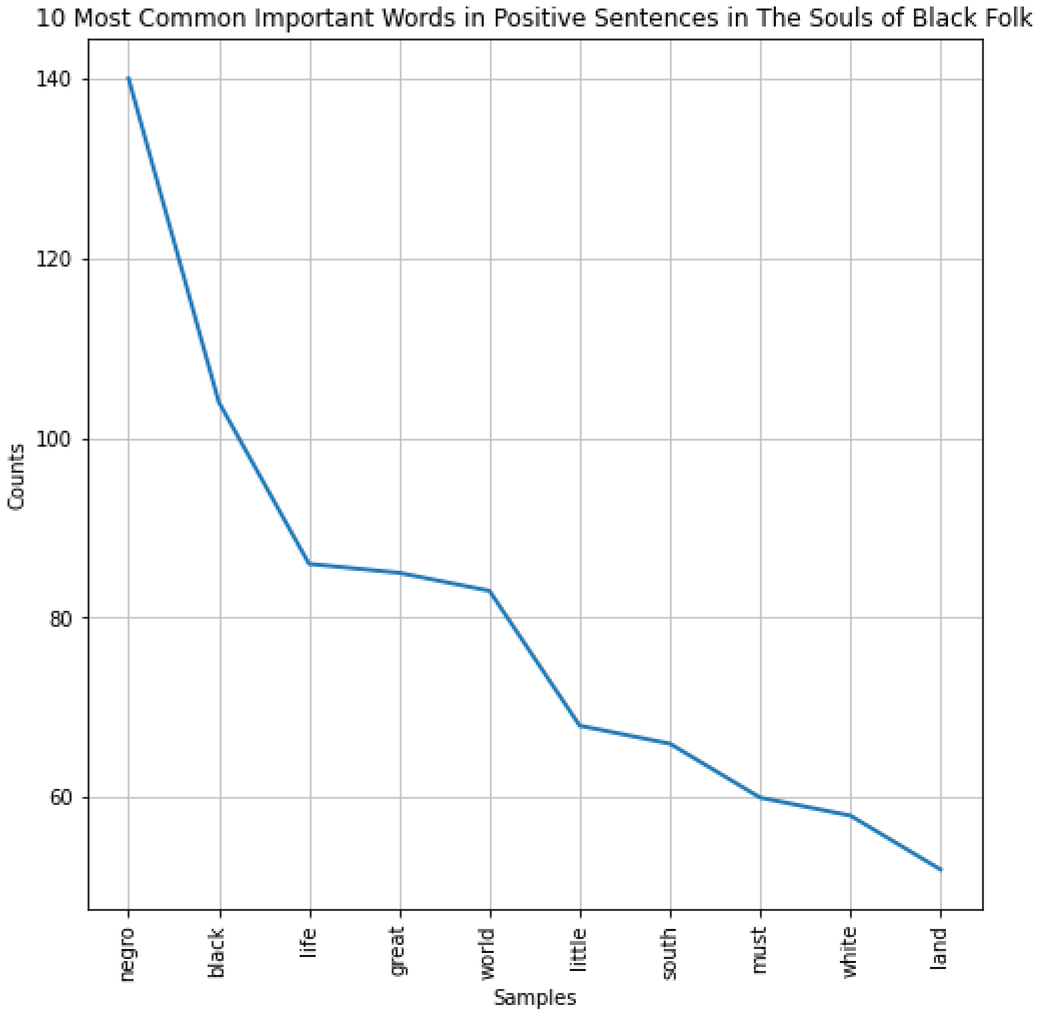
```
#There was higher lexical richness (34%) in the positive sentences than in the entire bool
len(set(tokenized_word_p))/len(tokenized_word_p)*100
```

Out[44]:

```
33.99451097804391
```

In [45]:

```
#A frequency distribution of the tokens revealed that the 10 most common tokens in the pos
fdict_p={'text':tokenized_word_p}
filter_df_p=pd.DataFrame(fdict_p)
filter_df_p['length']=filter_df_p['text'].apply(len)
filter_df_p=filter_df_p[~filter_df_p.text.isin(['“','’','”'])]
filter_df_p=filter_df_p.loc[filter_df_p['length']>3]
f1_p=filter_df_p['text'].tolist()
fdist_p = FreqDist(f1_p)
plt.figure(figsize=(8,8))
fdist_p.plot(10,title="10 Most Common Important Words in Positive Sentences in The Souls c
```



Out[45]:
```
<AxesSubplot:title={'center':'10 Most Common Important Words in Positive Sentences in The
Souls of Black Folk'}, xlabel='Samples', ylabel='Counts'>
```

Tokenization of negative sentences

In [46]:
```
#Tokenization of the negative sentences revealed 13,703 tokens.
negative=data[data['label']==1]
cleaned_text_n=message_cleaning(negative['text'])
tokenized_word_n = word_tokenize(cleaned_text_n)
len(tokenized_word_n)
```

Out[46]:
```
13703
```

In [47]:
```
#Of the tokens generated, 5,137 were unique.
len(set(tokenized_word_n))
```

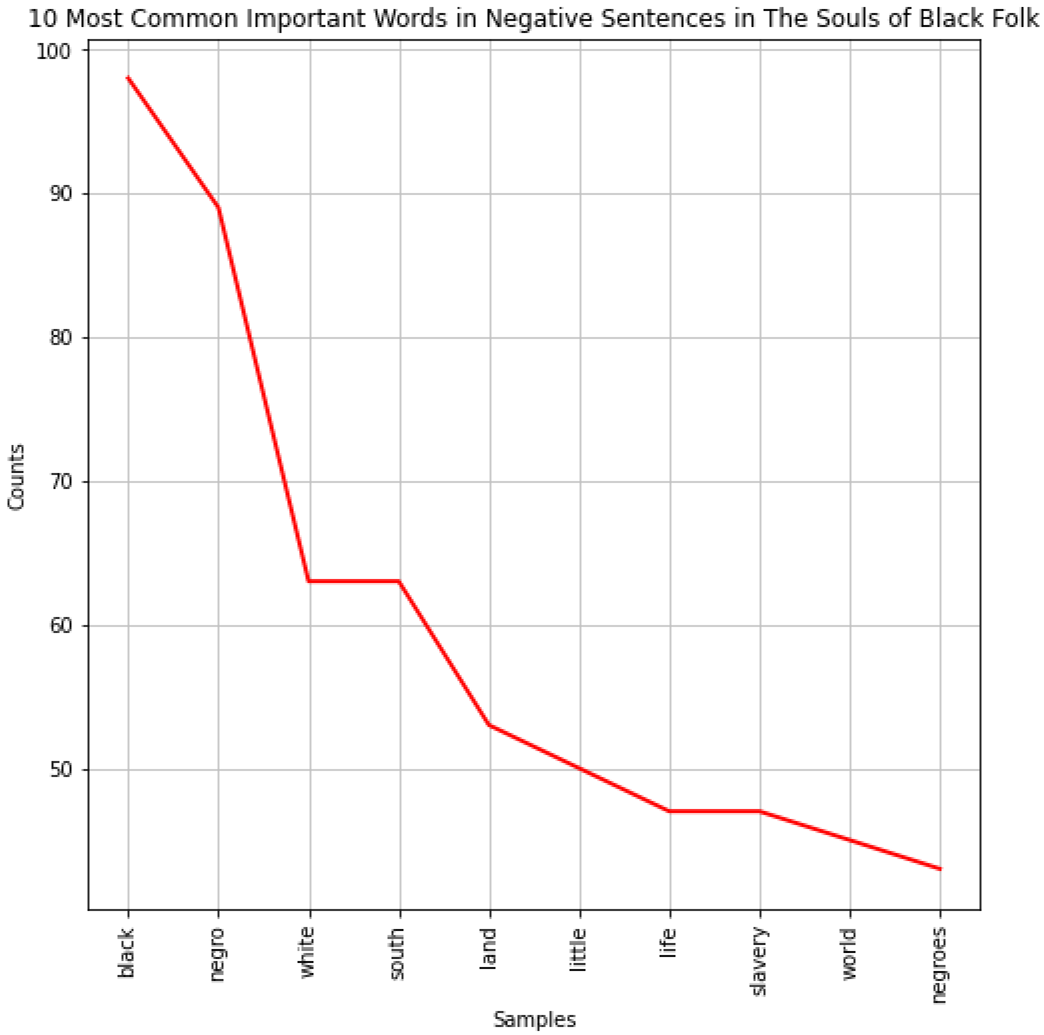
Out[47]:
```
5137
```

In [48]:
```
#The lexical richness of the negative text was about 37%.
len(set(tokenized_word_n))/len(tokenized_word_n)*100
```

Out[48]:
```
37.48814128293074
```

In [49]:
```
#A frequency distribution of the tokens revealed that the 10 most common tokens in the ne
fdict_n={'text':tokenized_word_n}
filter_df_n=pd.DataFrame(fdict_n)
filter_df_n['length']=filter_df_n['text'].apply(len)
```

```
filter_df_n=filter_df_n[~filter_df_n.text.isin(['“','’','”'])]
filter_df_n=filter_df_n.loc[filter_df_n['length']>3]
f1_n=filter_df_n['text'].tolist()
fdist_n = FreqDist(f1_n)
plt.figure(figsize=(8,8))
fdist_n.plot(10,title="10 Most Common Important Words in Negative Sentences in The Souls o
```



Out[49]:
```
<AxesSubplot:title={'center':'10 Most Common Important Words in Negative Sentences in The
Souls of Black Folk'}, xlabel='Samples', ylabel='Counts'>
```

Tokenization of neutral sentences

In [50]:
```
#Tokenization of the neutral sentences generated 6,921 tokens.
neutral=data[data['label']==-1]
cleaned_text_0=message_cleaning(neutral['text'])
tokenized_word_0 = word_tokenize(cleaned_text_0)
len(tokenized_word_0)
```

Out[50]:
```
6921
```

In [51]:
```
#Of the tokens generated, 2,959 were unique.
len(set(tokenized_word_0))
```

Out[51]:
```
2959
```

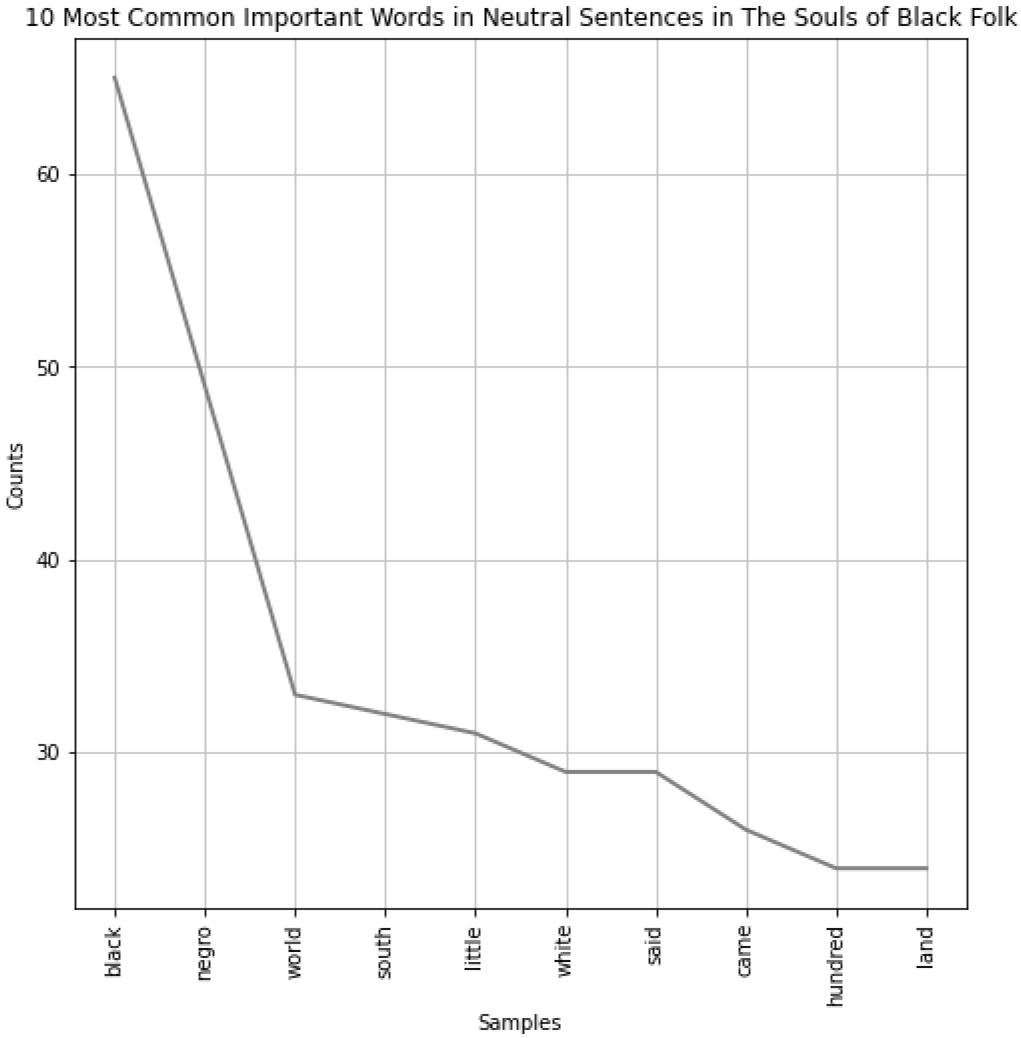
In [52]:
```
#The lexical richness of the neutral sentences (43%) was higher than that for the positive
len(set(tokenized_word_0))/len(tokenized_word_0)*100
```

Out[52]:

```
42.7539372922988
```

In [53]:

```
#A frequency distribution of the tokens revealed that the 10 most common tokens in the neu
fdict_0={'text':tokenized_word_0}
filter_df_0=pd.DataFrame(fdict_0)
filter_df_0['length']=filter_df_0['text'].apply(len)
filter_df_0=filter_df_0[~filter_df_0.text.isin(['“','’','”'])]
filter_df_0=filter_df_0.loc[filter_df_0['length']>3]
f1_0=filter_df_0['text'].tolist()
fdist_0 = FreqDist(f1_0)
plt.figure(figsize=(8,8))
fdist_0.plot(10,title="10 Most Common Important Words in Neutral Sentences in The Souls of
```



Out[53]:

```
<AxesSubplot:title={'center':'10 Most Common Important Words in Neutral Sentences in The S
ouls of Black Folk'}, xlabel='Samples', ylabel='Counts'>
```

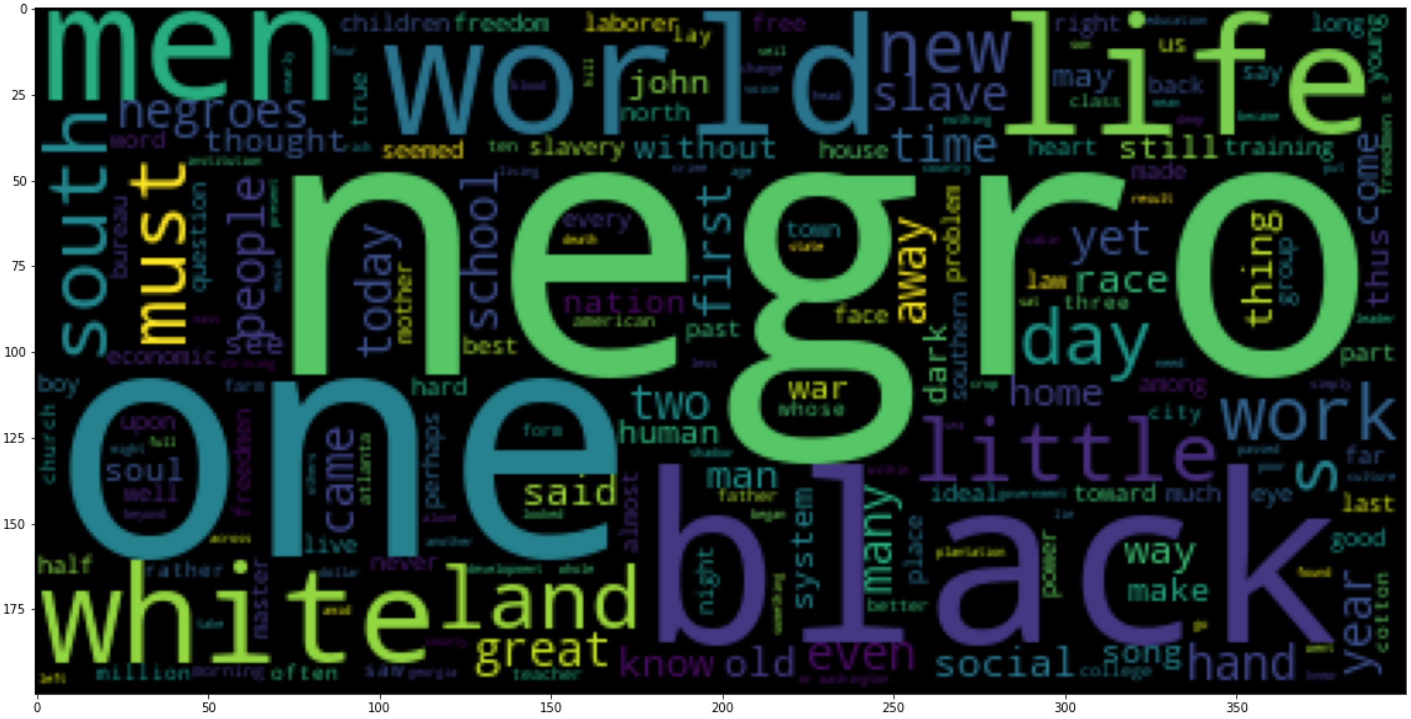
Plot word cloud for entire book

In [54]:

```
#Four word clouds for the tokens for the entire book, positive, negative, and neutral sent
tokenized_word_single_string = " ".join(tokenized_word)
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(tokenized_word_single_string))
```

Out[54]:

```
<matplotlib.image.AxesImage at 0x1d32da8c880>
```



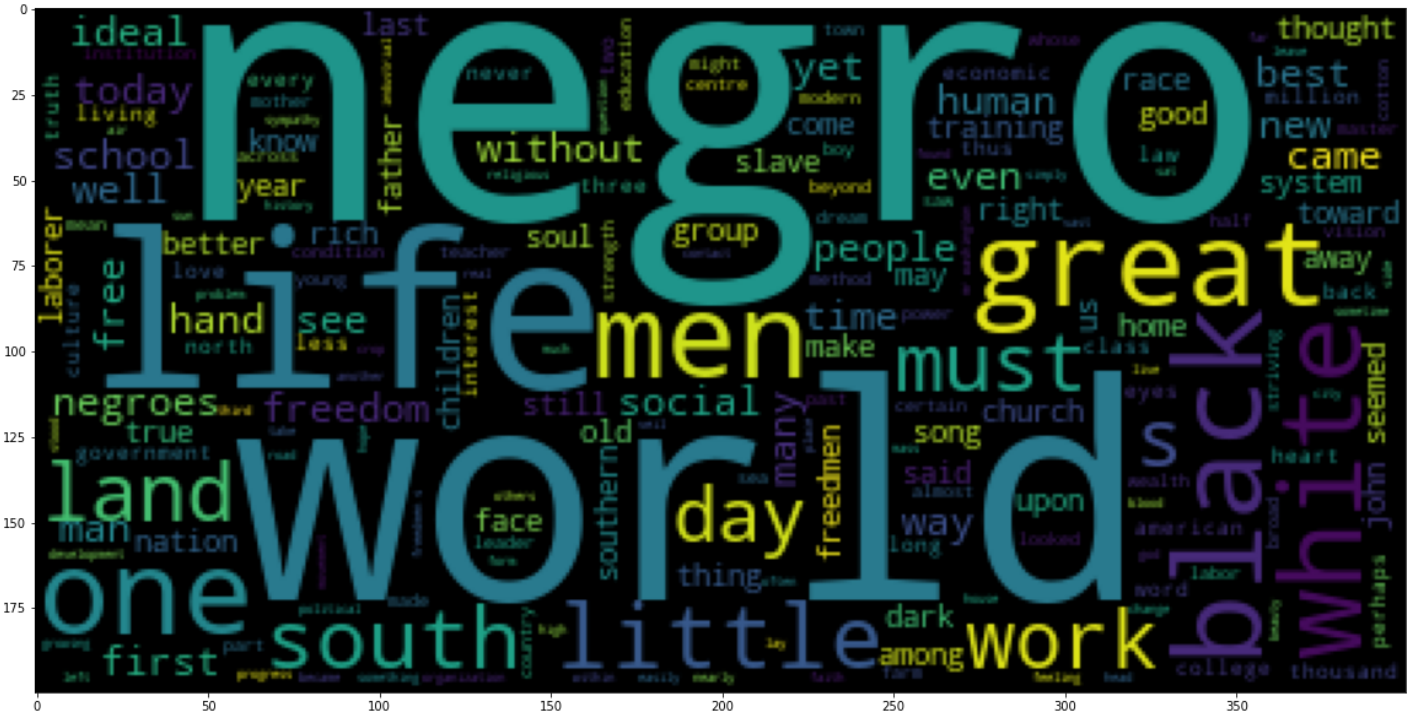
Plot word cloud for positive sentences

In [55]:

```
tokenized_word_single_string_p = " ".join(tokenized_word_p)
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(tokenized_word_single_string_p))
```

Out[55]:

```
<matplotlib.image.AxesImage at 0x1d32fb68100>
```

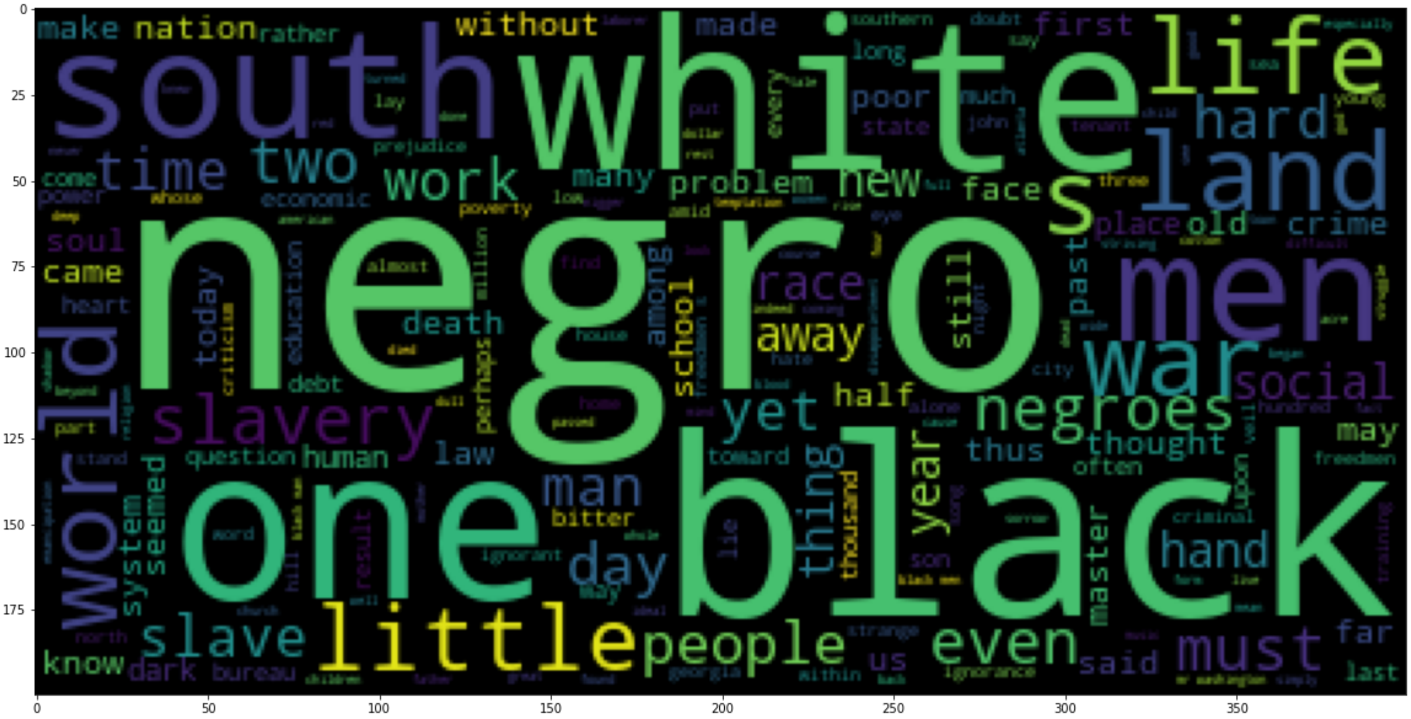


Plot word cloud for negative sentences

In [56]:

```
tokenized_word_single_string_n = " ".join(tokenized_word_n)
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(tokenized_word_single_string_n))
```

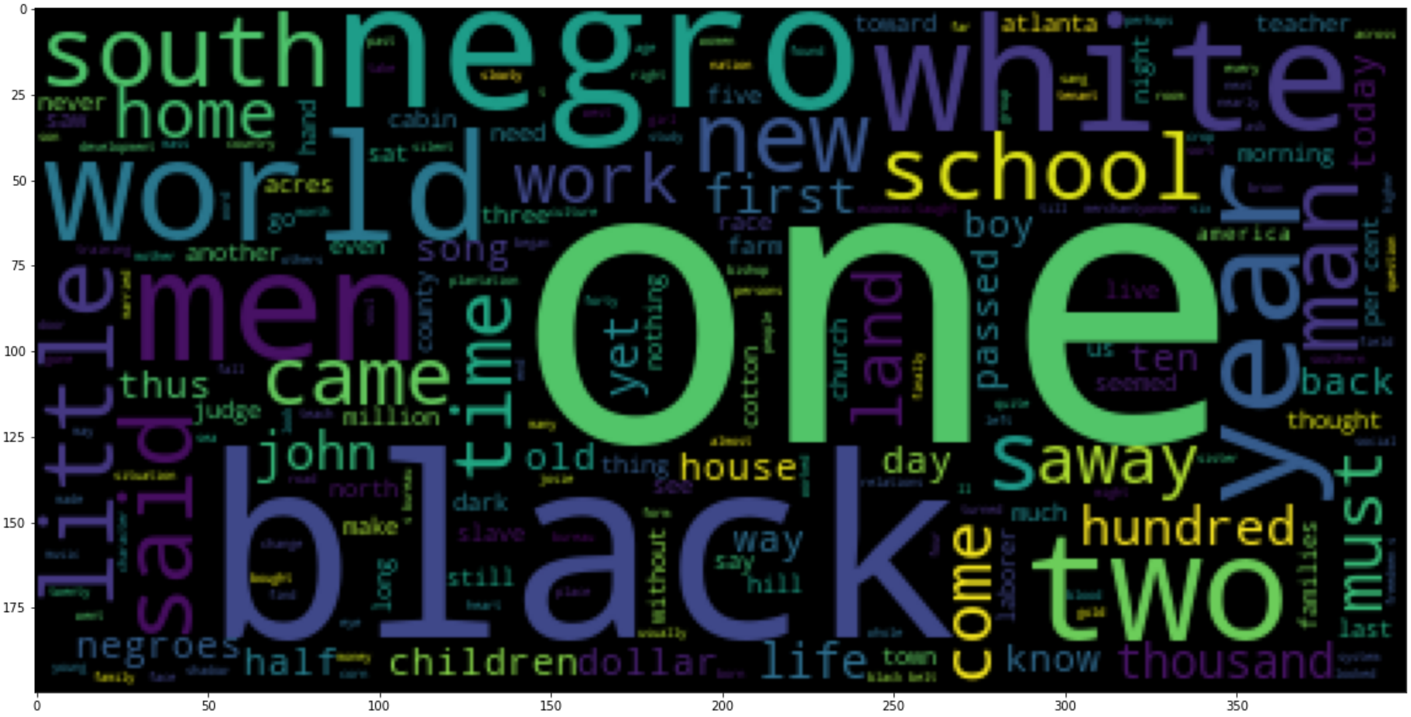
```
Out[56]: <matplotlib.image.AxesImage at 0x1d330193d60>
```



Plot word cloud for neutral sentences

```
In [57]: tokenized_word_single_string_0 = " ".join(tokenized_word_0)
         plt.figure(figsize=(20,20))
         plt.imshow(WordCloud().generate(tokenized_word_single_string_0))
```

```
Out[57]: <matplotlib.image.AxesImage at 0x1d32f5a6b80>
```



Machine Learning-Training Naive Bayes Classification Models to Predict Sentiment

```
In [58]: #The text column from the data frame was cleaned and vectorized in preparation for machine
         def message_cleaning1(message):
```

```
    Text_punc_removed = [char for char in message if char not in string.punctuation]
    Text_punc_removed_join = ''.join(Text_punc_removed)
    Text_punc_removed_join_clean = [word for word in Text_punc_removed_join.split() if wor
    return Text_punc_removed_join_clean
text_countvectorizer = CountVectorizer(analyzer = message_cleaning1, dtype = 'uint8').fit_
```

In [59]:

```
#The modified text column was put into a data frame.
x=text_countvectorizer
x.shape
```

Out[59]:

```
(2640, 9866)
```

In [60]:

```
#A data frame with just the label variable, the outcome variable, was created.
y = data['label']
y.shape
```

Out[60]:

```
(2640,)
```

In [61]:

```
#Test and train datasets were created with test dataset containing 20% of cases and the re
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,random_state=0)
NB_classifier = MultinomialNB()
NB_classifier.fit(x_train, y_train)
```

Out[61]:

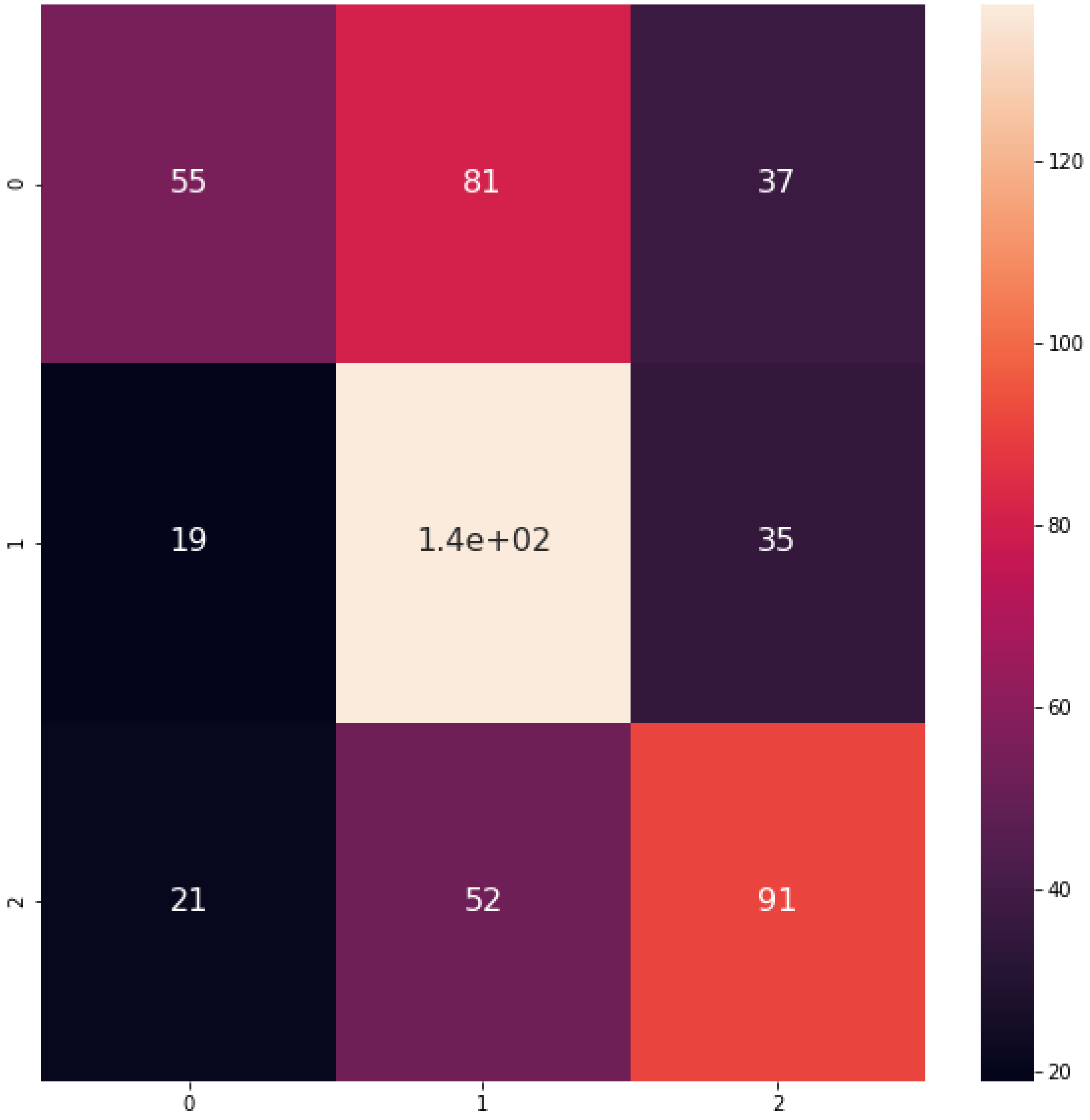
```
MultinomialNB()
```

In [62]:

```
#The model was then applied to the test data. A heat map with the confusion matrix showing
y_predict_test = NB_classifier.predict(x_test)
cm = confusion_matrix(y_test, y_predict_test)
plt.figure(figsize=(10,10))
sns.heatmap(cm, annot=True,annot_kws={"fontsize":16})
```

Out[62]:

```
<AxesSubplot:>
```



In [63]:

```
#A classification report reveals an accuracy score of 54%, showing that the model did not
print(classification_report(y_test, y_predict_test))
```

```
              precision    recall  f1-score   support

          -1       0.58      0.32      0.41       173
           0       0.51      0.72      0.59       191
           1       0.56      0.55      0.56       164

    accuracy                           0.54       528
   macro avg       0.55      0.53      0.52       528
weighted avg       0.55      0.54      0.52       528
```

In [64]:

```
#To deal with the low accuracy of the previous model, the neutral cases were removed from
data_np=data[data['label'] !=-1]
text_countvectorizer = CountVectorizer(analyzer = message_cleaning1, dtype = 'uint8').fit_
```

In [65]:

```
#The cleaned text column from the revised data frame was put into a data frame.
x=text_countvectorizer
x.shape
```

Out[65]:

```
(1892, 8682)
```

In [66]:

```
#A data frame with just the label variable from the revised data frame was created.
y = data_np['label']
y.shape
```

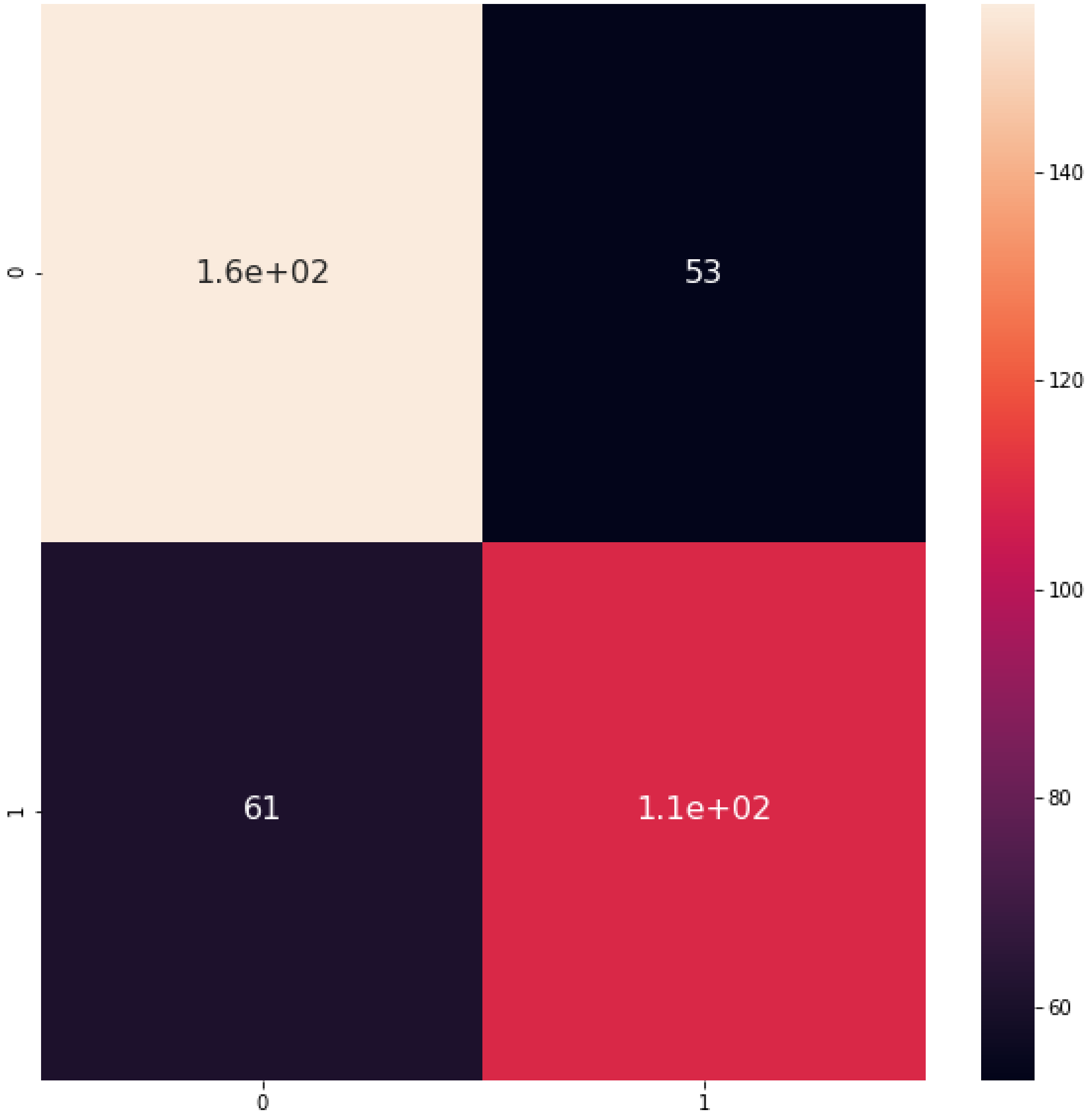
```
Out[66]: (1892,)
```

```
In [67]: #Test and training datasets were generated from the revised data frame with test dataset c
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
NB_classifier = MultinomialNB()
NB_classifier.fit(x_train, y_train)
```

```
Out[67]: MultinomialNB()
```

```
In [68]: #The model was then applied to the test data. A heat map showing the confusion matrix show
y_predict_test = NB_classifier.predict(x_test)
cm = confusion_matrix(y_test, y_predict_test)
plt.figure(figsize=(10,10))
sns.heatmap(cm, annot=True,annot_kws={"fontsize":16})
```

```
Out[68]: <AxesSubplot:>
```



```
In [69]: #Looking at the classification report shows that dropping the neutral text caused an incre
print(classification_report(y_test, y_predict_test))
```

```
              precision    recall  f1-score   support
```

```
           0       0.72      0.75      0.73       209
           1       0.67      0.64      0.66       170

    accuracy                           0.70       379
   macro avg       0.70      0.69      0.69       379
weighted avg       0.70      0.70      0.70       379
```

In [ ]: