Python Code to get the Frequency

```
from nltk.stem import PorterStemmer

from nltk.tokenize import word_tokenize

porter = PorterStemmer()


keyword =
['risk','threat','vulnerability','cybersecurity','security','privacy','officer','access','sensitive','breach','attack','encryption','technolo
gy','management','critical','protection','framework','network','secure','private']


##Stem the keyword to obtain the root of the word

keyword_Stem = [porter.stem(w) for w in keyword]


##open and read the file into the variables d1,d2,d3,d4,d5

d1 = open("File1.txt","r",encoding="utf-8").read()

d2 = open("File2.txt","r",encoding="utf-8").read()

d3 = open("File3.txt","r",encoding="utf-8").read()

d4 = open("File4.txt","r",encoding="utf-8").read()


##Replace the punctuation mark with "" and turn the words into lower case

replacePuncMark = [',','\n','.','â€™s','â€™','?','(',')','-','=','`','!','&','%','$','@','~','*','#','/','\','"',':','·']

for mark in replacePuncMark:

    d1 = d1.replace(mark," ")

    d2 = d2.replace(mark," ")

    d3 = d3.replace(mark," ")

    d4 = d4.replace(mark," ")

    d5 = d5.replace(mark," ")

d1 = d1.lower()

d2 = d2.lower()

d3 = d3.lower()

d4 = d4.lower()

d5 = d5.lower()
```

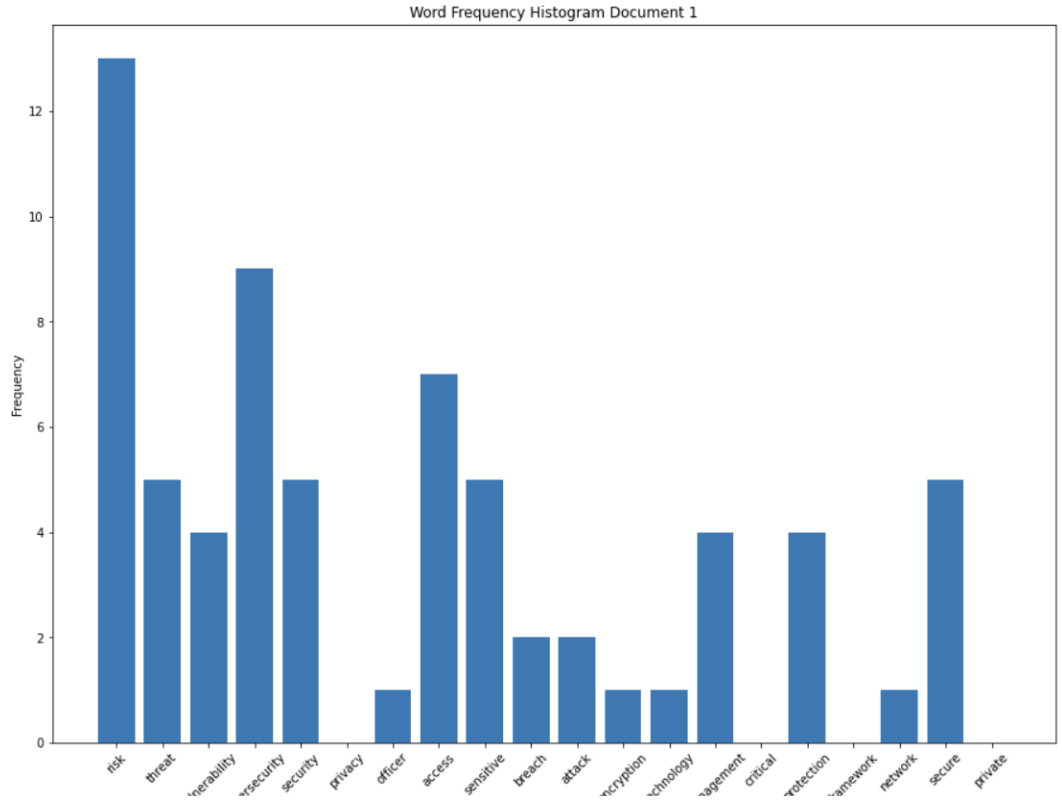
```
d1_tokenized = word_tokenize(d1)
d2_tokenized = word_tokenize(d2)
d3_tokenized = word_tokenize(d3)
d4_tokenized = word_tokenize(d4)
d5_tokenized = word_tokenize(d5)
d1Stemmed = [porter.stem(w) for w in d1_tokenized]
d2Stemmed = [porter.stem(w) for w in d2_tokenized]
d3Stemmed = [porter.stem(w) for w in d3_tokenized]
d4Stemmed = [porter.stem(w) for w in d4_tokenized]
d5Stemmed = [porter.stem(w) for w in d5_tokenized]
d1WordFreq = {}
d2WordFreq = {}
d3WordFreq = {}
d4WordFreq = {}
d5WordFreq = {}
for num in range(len(keyword_Stem)):
    numOfWords1 = d1Stemmed.count(keyword_Stem[num])
    d1WordFreq.update({keyword[num]:numOfWords1})
    numOfWords2 = d2Stemmed.count(keyword_Stem[num])
    d2WordFreq.update({keyword[num]:numOfWords2})
    numOfWords3 = d3Stemmed.count(keyword_Stem[num])
    d3WordFreq.update({keyword[num]:numOfWords3})
    numOfWords4 = d4Stemmed.count(keyword_Stem[num])
    d4WordFreq.update({keyword[num]:numOfWords4})
    numOfWords5 = d5Stemmed.count(keyword_Stem[num])
    d5WordFreq.update({keyword[num]:numOfWords5})
```

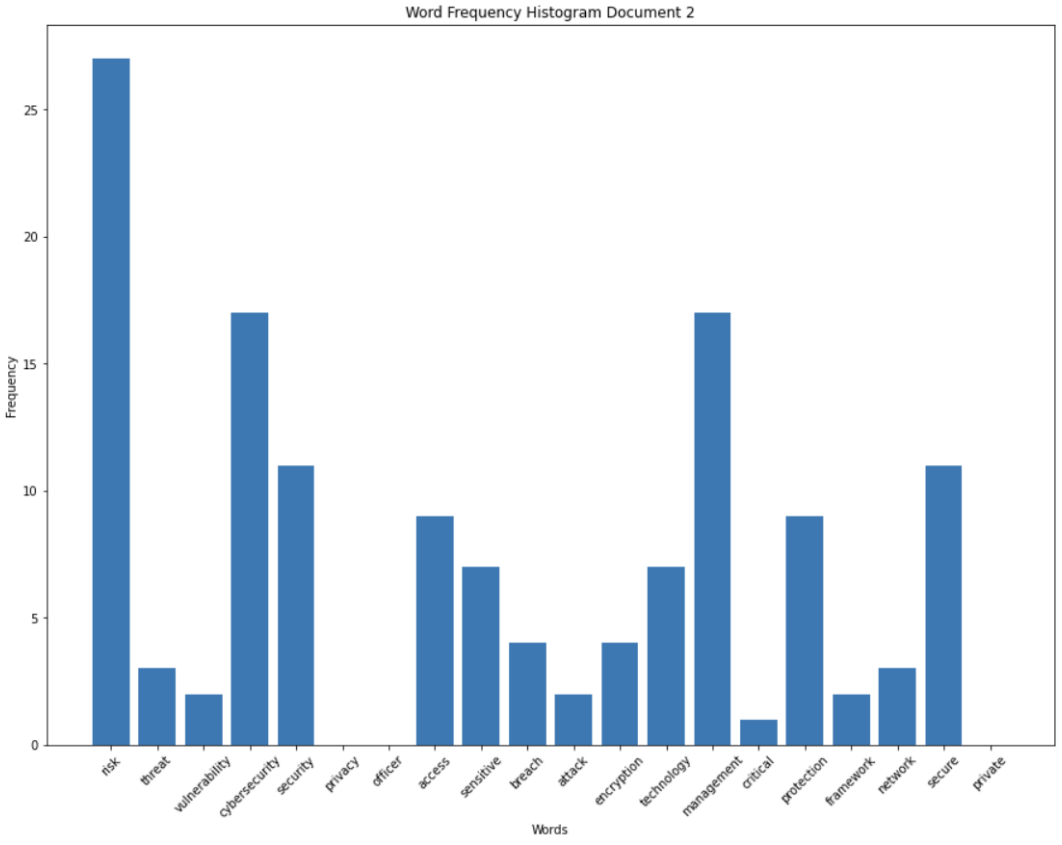
```
#Print out the frequency table
maxx = 0
for w in keyword:
    if len(w) > maxx:
        maxx = len(w)
for num in range(maxx):
    print(" ",end=""),
print("  ",end=""),
print("d1  d2  d3  d4  d5")
print("")
for w in keyword:
    space = maxx - len(w)
    word_space = ""
    for n in range(space):
        word_space += " "
    row = w+word_space
    if len(str(d1WordFreq[w]))==2:
        row = row + "  " +str(d1WordFreq[w])
    else:
        row = row + "   " +str(d1WordFreq[w])
    if len(str(d2WordFreq[w]))==2:
        row = row + "  " +str(d2WordFreq[w])
    else:
        row = row + "   " +str(d2WordFreq[w])
    if len(str(d3WordFreq[w]))==2:
        row = row + "  " +str(d3WordFreq[w])
    else:
        row = row + "   " +str(d3WordFreq[w])
    if len(str(d4WordFreq[w]))==2:
        row = row + "  " +str(d4WordFreq[w])
    else:
        row = row + "   " +str(d4WordFreq[w])
    if len(str(d5WordFreq[w]))==2:
        row = row + "  " +str(d5WordFreq[w])
    else:
        row = row + "   " +str(d5WordFreq[w])


    print(row)
```

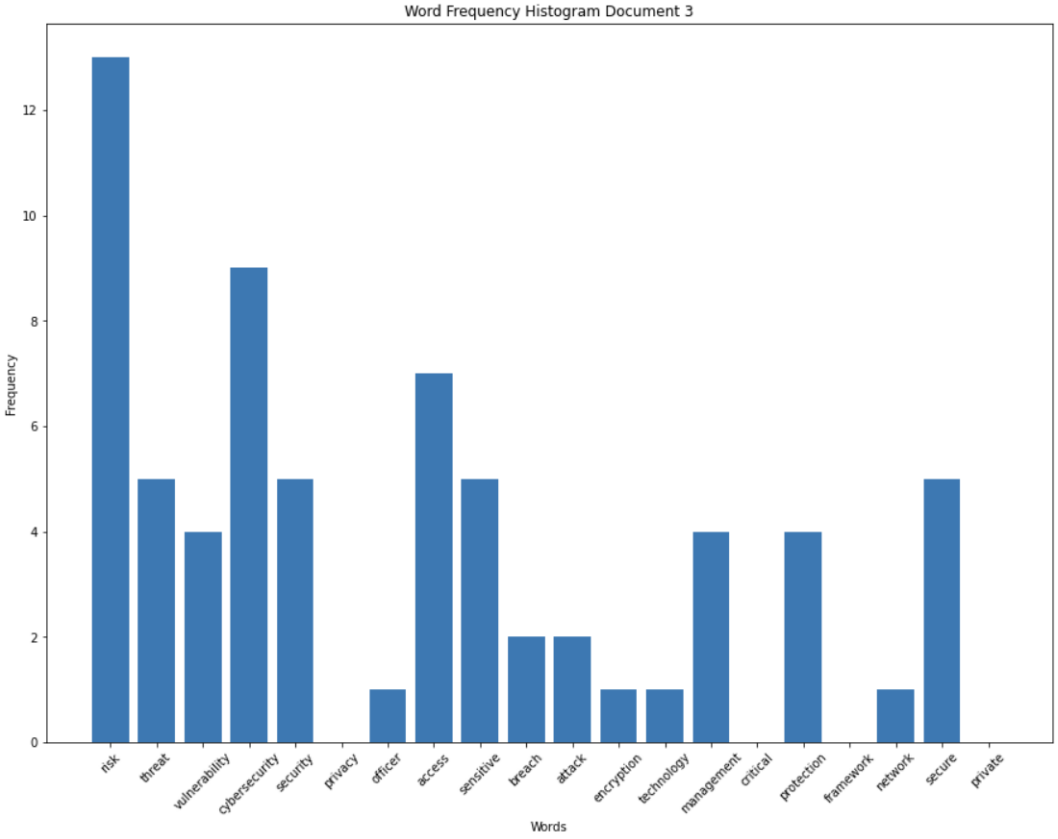
Output:

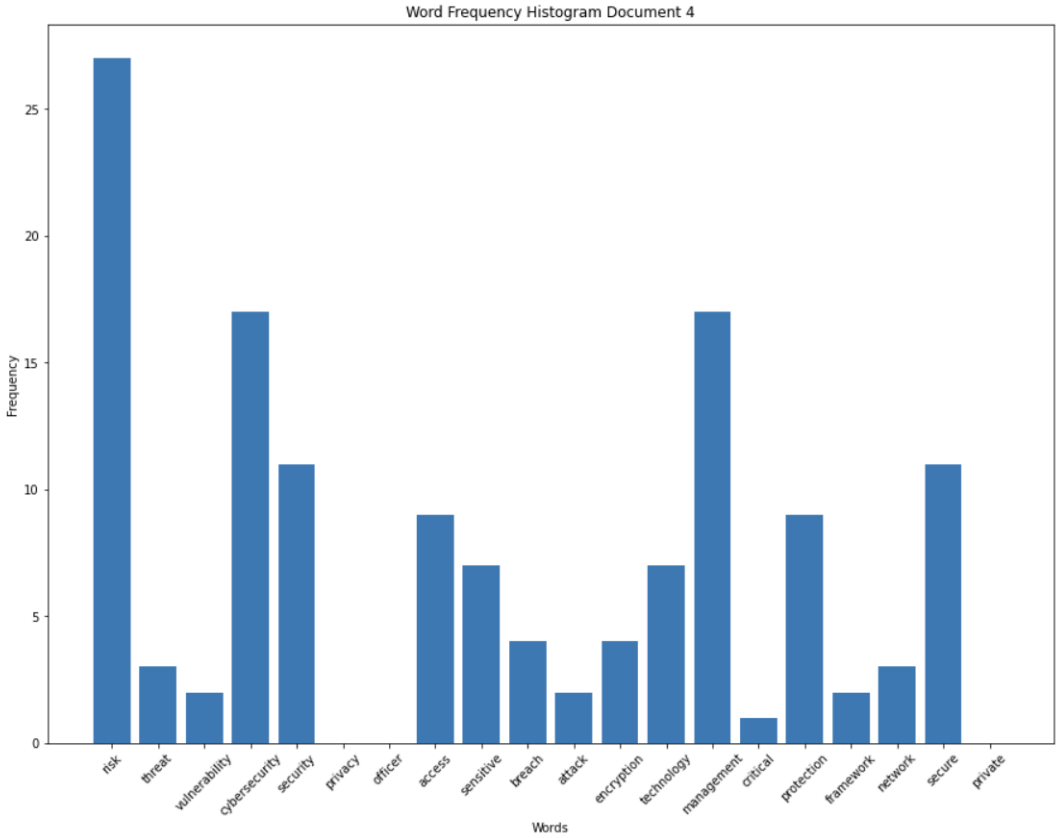
	d1	d2	d3	d4	d5
risk	13	27	13	27	0
threat	5	3	5	3	0
vulnerability	4	2	4	2	0
cybersecurity	9	17	9	17	0
security	5	11	5	11	0
privacy	0	0	0	0	0
officer	1	0	1	0	0
access	7	9	7	9	0
sensitive	5	7	5	7	0
breach	2	4	2	4	0
attack	2	2	2	2	0
encryption	1	4	1	4	0
technology	1	7	1	7	5
management	4	17	4	17	0
critical	0	1	0	1	0
protection	4	9	4	9	1
framework	0	2	0	2	0
network	1	3	1	3	0
secure	5	11	5	11	0
private	0	0	0	0	0

Histogram of Each File against The Keywords






Word Frequency Histogram Document 3
Frequency
Words
0
2
4
6
8
10
12
risk
threat
vulnerability
cybersecurity
security
privacy
officer
access
sensitive
breach
attack
encryption
technology
management
critical
protection
framework
network
secure
private


Word Frequency Histogram Document 4
Frequency
Words
0
5
10
15
20
25
risk
threat
vulnerability
cybersecurity
security
privacy
officer
access
sensitive
breach
attack
encryption
technology
management
critical
protection
framework
network
secure
private

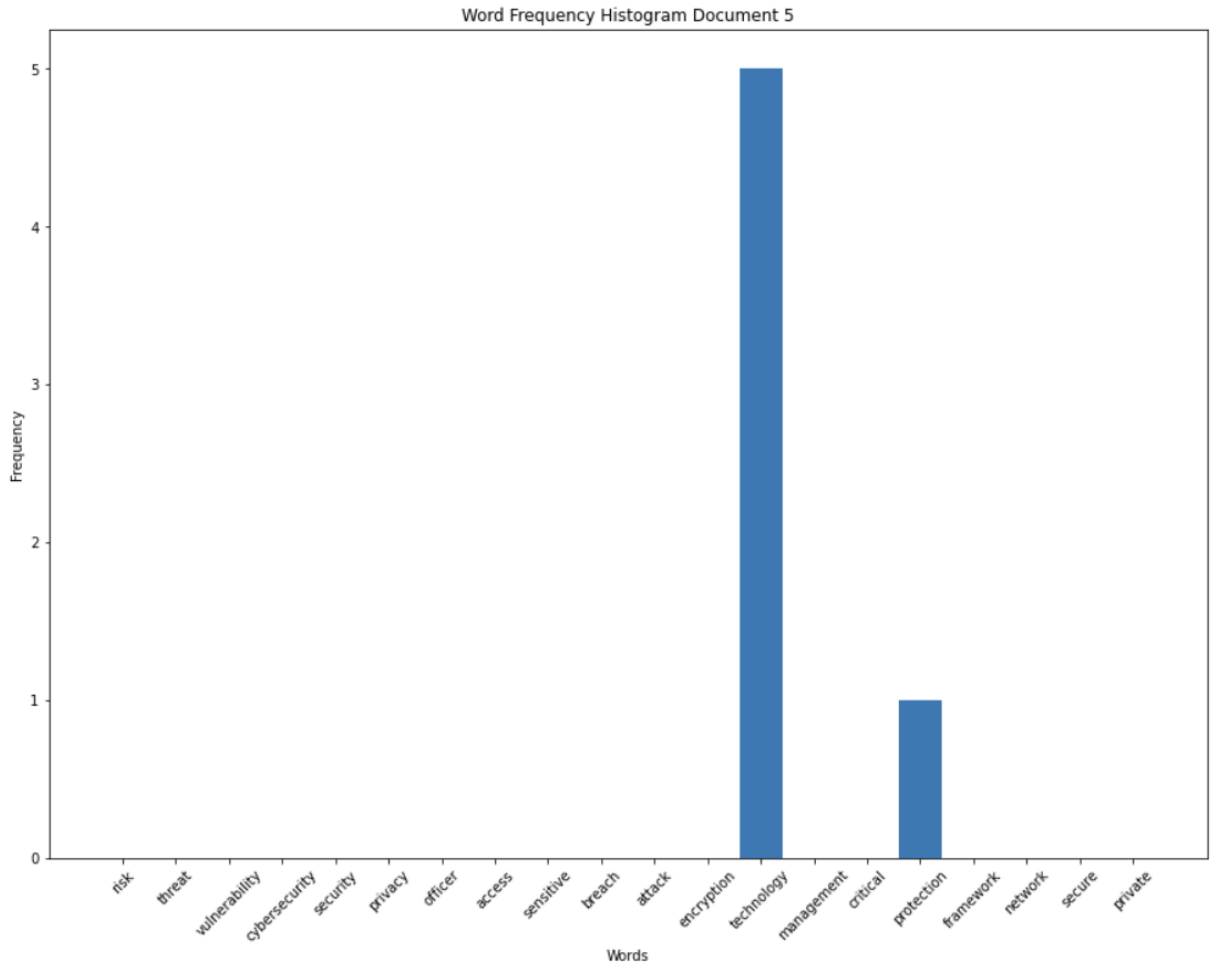

Word Frequency Histogram Document 5
Frequency
Words
0
1
2
3
4
5
risk
threat
vulnerability
cybersecurity
security
privacy
officer
access
sensitive
breach
attack
encryption
technology
management
critical
protection
framework
network
secure
private

Table of Cosine Similarity Values of Documents

	File1	File2	File3	File4	File5
File1	0	0.9332	1.0000	0.9332	0.0824
File2	0.9332	0	0.9332	1.0000	0.1994
File3	1.0000	0.9332	0	0.9332	0.0824
File4	0.9332	1.0000	0.9332	0	0.1994
File5	0.0824	0.1994	0.0824	0.1994	0

Conclusion

For File1, the highest cosine similarity is with File3, 1.0000, while the lowest cosine similarity is with File5, 0.0824.

For File2, the highest cosine similarity is with File4, 1.0000, while the lowest cosine similarity is with File5, 0.1994.

For File3, the highest cosine similarity is with File1, 1.0000, while the lowest cosine similarity is with File5, 0.0824.

For File4, the highest cosine similarity is with File2, 1.0000, while the lowest cosine similarity is with File5, 0.1994.

For File5, the highest cosine similarity is File2 and File4, 0.1994, while the lowest cosine similarity is with File1 and File3, 0.0824.

Overall, the highest cosine similarity is File1 with File3 and File2 with File4. The lowest cosine similarity is File1 with File5 and File3 with File5.