

writeup

i_lakash

September 5, 2019

0.1 knowledge

0.1.1 frequency

Frequency distribution English characters

a: 8.05%	b: 1.67%	c: 2.23%	d: 5.10%
e: 12.22%	f: 2.14%	g: 2.30%	h: 6.62%
i: 6.28%	j: 0.19%	k: 0.95%	l: 4.08%
m: 2.33%	n: 6.95%	o: 7.63%	p: 1.66%
q: 0.06%	r: 5.29%	s: 6.02%	t: 9.67%
u: 2.92%	v: 0.82%	w: 2.60%	x: 0.11%
y: 2.04%	z: 0.06%		

0.1.2 Frequent one letter word

'a','i'

0.1.3 Frequent two letter word

'of', 'to', 'in', 'it', 'is', 'be', 'as', 'at', 'so', 'we', 'he', 'by', 'or', 'on', 'do', 'if',
'me', 'my', 'up', 'an', 'go', 'no', 'us', 'am'

0.1.4 Frequent three letter word

'the', 'and', 'for', 'are', 'but', 'not', 'you', 'all', 'any', 'can', 'had', 'her', 'was',
'one', 'our', 'out', 'day', 'get', 'has', 'him', 'his', 'how', 'new', 'now', 'old',
'see', 'two', 'way', 'who', 'boy', 'did', 'let', 'put', 'say', 'she', 'too'

0.2 Encrypted paragraph

aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu thmjuv hs klu zlevu shv zcbkg guovz, upuv zcmdu lez vuwovroaeu jczyyuvomdu omj qmubyudkuj vukqvm. klu vcdluz lu loj avhqnlk aodr svhw lez kvopuez loj mht audhwu o ehdoe eunumj, omj ck toz yhyqeoveg auecupuj, tlokupuv klu hej sher wcnlk zog, klok klu lcee ok aon umj toz sqee hs kqmmuez zkqs-suj tckl kvuozquv. omj cs klok toz mhk umhqnl shv sowu, kluvu toz oezh lez yvhehmnij pcnhqv kh wovpue ok. kcwu thvu hm, aqk ck zuuwuj kh lopu eckkeu ussudk hm wv. aonncmz. ok mcmukg lu toz wqdl klu zowu oz ok scskg. ok mcmukg-mcmu klug aunom kh doee lew tuee-yvuzuvpuj; aqk qmdlomnij thqej lopu auum muovuv klu wovr. kluvu tuvz zhwu klok zlhhr klucv luojz omj klhqnlk klez toz khh wqdl hs o nhhj klcmm; ck zuuwuj qm-socv klok omghmu zlhqej yhzuz (oyyovumkeg) yuvyukqoe ghqkl oz tuee oz (vuyqkujeg) cmubloqzkaeu tuoekl. ck tcee lopu kh au yocj shv, klug zocj. ck czm'k mokqvoe, omj kvhqaue tcee dhwu hs ck! aqk zh sov kvhqaue loj mhk dhwu; omj oz wv. aonncmz toz numuvhqz tckl lez whmug, whzk yuhyeu tuvz tceecmn kh shvncpu lew lez hjjckuz omj lez nhhj shvkqmu. lu vuwocmuj hm pczckmn kuvwz tckl lez vueokepuz (ubduyk, hs dhqvzu, klu zodrpceeu-aonncmzuz), omj lu loj womg juphkuj ojwcvuvz owghmn klu lhaackz hs yhhv omj qmcwyhvkomm sowcecu. aqk lu loj mh dehzu svcumjz, qmkce zhwu hs lez ghqmmuv dhqzcmz aunom kh nvht qy. klu uejuzk hs kluzu, omj aceah'z sophqvcku, toz ghqmm svhjh aonncmz. tlum aceah toz mcmukg-mcmu lu ojhykuj svhjh oz lez lucv, omj avhqnlk lew kh ecpu ok aon umj; omj klu lhyuz hs klu zodrpceeu- aonncmzuz tuvz scmoeej jozluj. aceah omj svhjh loyyumuj kh lopu klu zowu acvkljog, zuykuwauv 22mj. ghq loj aukku dhwu omj ecpu luvu, svhjh wg eo, zocj aceah hmu jog; omj klum tu dom dueuavoku hqv acvkljog-yovkuz dhwshvkoaeg khnuklu. ok klok kcwu svhjh toz zkcee cm lez ktuumz, oz klu lhaackz doeeuj klu cvvuzymzcae kumkuz auktuum dlcejlhj omj dhwcmm hs onu ok klcvkg-klvuu

Chapter 1

Analysis

1.1 Frequency

1.1.1 one letter word

o : 2

1.1.2 two letter word

au : 1 ck : 6 cm : 1 cs : 1 hm : 3 hs : 10 kh : 8 lu : 6 mh : 1 ok : 8 oz : 6
qy : 1 tu : 1 wg : 1 vv : 2 zh : 1

1.1.3 three letter word

aon : 2 aqk : 4 dom : 1 eoj : 1 ghq : 1 hej : 1 hmu : 1 hqv : 1 khh : 1 klu :
15 lcw : 3 lez : 10 loj : 7 mhh : 2 mht : 1 omj : 18 onu : 1 shv : 3 sov : 1
toz : 11 umj : 1 zog : 1

1.1.4 only one letter word o=?

sentence 1 : klu vcdluz lu loj avhqnlk aodr svhw lez kvopuez loj mht
audhwu o ehdoe eunumj

sentence 2 : kluvu tuvuv zhuv klok zlhhr klucv luojz omj klhqnlk klez toz
khh wqdl hs o nhhj klcmm; ck zuuwuj qmsocv klok omghmu zlhqej yhzuzz

So 'o' can be a/i...but usually i doesn't sit in middle of simple sentence..even if i sits in the middle,then the sentence will be complex or compound..in this case there is no preceding comma(,) or conjunction(atleast 3 character word,but here is hs(2 characters in 2nd senntence)) in the above sentence

$$\text{mapper} : \mathbf{o}=\mathbf{a} \quad (1.1)$$

1.1.5 most frequent three letter word omj(18times)=?

We already know $\mathbf{o}=\mathbf{a}$...

So $\mathbf{omj} = \mathbf{amj}$

From knowledge section most frequent three letter word [0.1.4](#) we know that **and** start with **a**

we can guess(90%) that $\mathbf{omj} = \mathbf{and}$

then

$$\text{mapper} : \mathbf{m}=\mathbf{n} \quad (1.2)$$

$$\text{mapper} : \mathbf{j}=\mathbf{d} \quad (1.3)$$

1.1.6 second most frequent three letter word klu(15times)=?

We can also guess that $\mathbf{klu} = \mathbf{the}$.. If we map this time we can be sure about our guess..

$$\text{mapper} : \mathbf{k}=\mathbf{t} \quad (1.4)$$

$$\text{mapper} : \mathbf{l}=\mathbf{h} \quad (1.5)$$

$$\text{mapper} : \mathbf{u}=\mathbf{e} \quad (1.6)$$

In the last line : ok **klok** kcwu svjh toz zkcee cm lcz ktuumz
klok = that

1.1.7 2nd most frequent 2 letter word kh=?

we already know $\mathbf{k} = \mathbf{t}$

there is also three letter word **khh**

from knowledge two letter section 0.1.3 we can easily find out that

$$\text{mapper} : \mathbf{h}=\mathbf{o} \quad (1.7)$$

1.1.8 ck=?

we already know $\mathbf{k} = \mathbf{t}$

So $\mathbf{c} = \mathbf{i}$ because $\mathbf{c!} = \mathbf{a}$

$$\text{mapper} : \mathbf{c}=\mathbf{i} \quad (1.8)$$

1.1.9 first word aceah=? of the paragraph

starting sentence : aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu

Using our mapper value if we try to translate the underlined words ..

,omj loj auum klu=,and had (b/s)een the

as there is a comma(,) before **and** there is more possiblity for 'b'..

but we also use try both 'b' and 's'

however using 'a'='b' ..

$$\text{aceah} = \mathbf{bi?bo} \quad (1.9)$$

Then we if we google search **5 letter word end with bo** and click on this link

<https://www.thefreedictionary.com/words-that-end-in-bo>

we will see ..

5-letter words that end in bo

turbo	combo	jumbo
limbo	bimbo	mambo
dumbo	gumbo	carbo
bilbo	adobo	lesbo
yobbo	sambo	gombo

And we match with our **bi?bo** then we can easily match with **bi?bo = bilbo**

$$\mathbf{bi?bo=bilbo} \quad (1.10)$$

So now if search about bilbo we can know that **bilbo(Bilbo Baggins) is lead character of hobbit**

this link show some similarity with our encrypted paragraph..

http://www.henneth-annun.net/bios_iew.cfm?scid=11

1.2 Decrypted paragraph

bilbo was very rich and very peculiar, and had been the wonder of the shire for sixty years, ever since his remarkable disappearance and unexpected return. the riches he had brought back from his travels had now become a local legend, and it was popularly believed, whatever the old folk might say, that the hill at bag end was full of tunnels stuffed with treasure. and if that was not enough for fame, there was also his prolonged vigour to marvel at. time wore on, but it seemed to have little effect on mr. baggins. at ninety he was much the same as at fifty. at ninety-nine they began to call him well-preserved; but unchanged would have been nearer the mark. there were some that shook their heads and thought this was too much of a good thing; it seemed unfair that anyone should possess (apparently) perpetual youth as well as (reputedly) inexhaustible wealth. it will have to be paid for, they said. it isn't natural, and trouble will come of it! but so far trouble had not come; and as mr. baggins was generous with his money, most people were willing to forgive him his oddities and his good fortune. he remained on visiting terms with his relatives (except, of course, the sackvillebagginses), and he had many devoted admirers among the hobbits of poor and unimportant families. but he had no close friends, until some of his younger cousins began to grow up. the eldest of these, and bilbo's favourite, was young frodo baggins. when bilbo was ninety-nine he adopted frodo as his heir, and brought him to live at bag end; and the hopes of the sackville- bagginses were finally dashed. bilbo and frodo happened to have the same birthday, september 22nd. you had better come and live here, frodo my lad, said bilbo one day; and then we can celebrate our birthday-parties comfortably together. at that time

frodo was still in his tweens, as the hobbits called the irresponsible twenties between childhood and coming of age at thirty-three

1.3 Code to decrypt

keep the data.txt file in same folder

```
from string import ascii_lowercase
import re

one=['a','i']
two=['of','to','in','it','is','be','as','at','so','we','he','by','or',
three=['the','and','for','are','but','not','you','all','any','can','had',

def decrypt():
    key="bxicl yo dthngavukfworm ps"

    words=[]
    ciphertext=""

    with open('data.txt','r') as fd:
        lines=fd.readlines()
        ciphertext=lines[0]

    for j in range(len(ciphertext)):
        c=ciphertext[j]
        if c in ascii_lowercase:
            index=ord(c)-97
            ciphertext=ciphertext[:j]+str(key[index])+ciphertext[j+1:]

    print("="*30,"decrypted paragraph","="*30)
    print()
    print(ciphertext)
    print()
```



```

def getUnique(list=[]):
    ret=[]
    for c in list:
        if c not in ret:
            ret.append(c)

    return ret

def getWordFrequency(freq={},word=""):
    if word in freq:
        freq[word]=freq[word]+1
    else:
        freq[word]=1
    return freq

def printFreq(freq={},header="a"):
    print('\n',"="*30,header,"="*30)
    print()
    for key in sorted(freq.keys()):
        print(key," :",freq[key],end=" ")
    print('\n\n',"-"*80)

def analysis():
    words=[]
    oneLetterFreq={}
    twoLetterFreq={}
    threeLetterFreq={}

    with open('data.txt','r') as fd:

```

```

lines=fd.readlines()

line=lines[0]
line=re.sub(","," ",line)
line=re.sub("\.+", " ",line)
line=re.sub("\!+", " ",line)
line=re.sub(" +", " ",line)
words=line.split(" ")

for word in words:
    if len(word)==1:
        oneLetterFreq=getWordFrequency(oneLetterFreq,word)

    if len(word)==2:
        twoLetterFreq=getWordFrequency(twoLetterFreq,word)
    if len(word)==3:
        threeLetterFreq=getWordFrequency(threeLetterFreq,word)

printFreq(oneLetterFreq,'one leter freq')
printFreq(twoLetterFreq,'two leter freq')
printFreq(threeLetterFreq,'three leter freq')


freq={}
for c in line:
    if c not in freq:
        freq[c]=1
    else :
        freq[c]=freq[c]+1

print("="*30,"Character frequency","="*30)
print()
for c in sorted(freq.keys()):
    print(c," : ",freq[c],end=" ")
print("\n","-"*80)

```

```
analysis()  
decrypt()
```