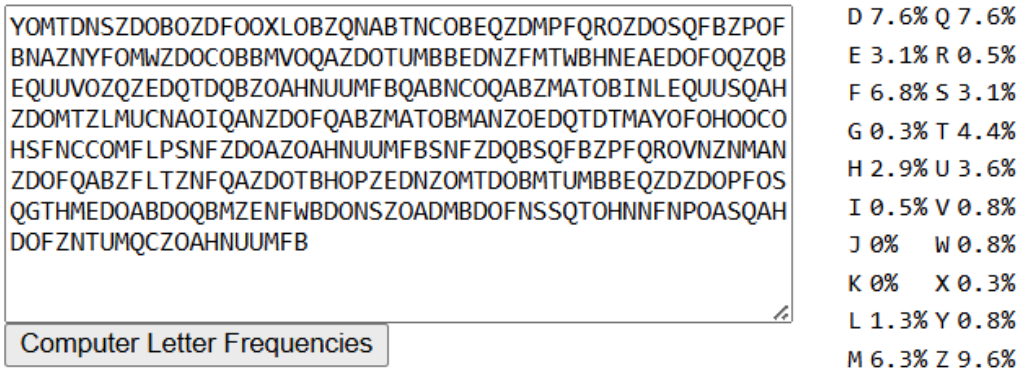
CIS 3362 Homework #2: Substitution Cipher, Vigenere

Part A: Code Break Questions

1) Decode the following message, which was encrypted using the substitution cipher. Make sure to discuss all the steps you took, the key you arrived at, and the decoded message.

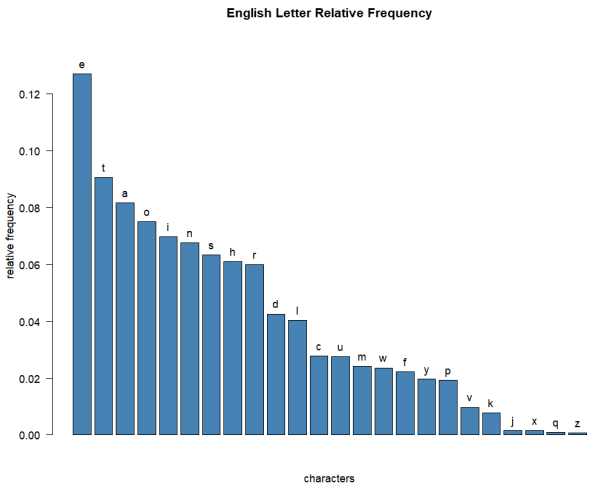
```
omtdnszdobozdfooxlobzqnabtncobeqzdmpfqrozdosqfbzpofbnaznyfom
wzdocobbmvoqazdotumbbednzfmtwbhneaedofoqzqbequuvozqzedqtdqbz
oahnuumfbqabncoqabzmatobinlequusqahzdomtzlmucnaoiqanzdofqabz
matobmanzoedqtdtmayofohoocohsfnccomflpsnfzdoazoahnuumfbsnfzd
qbsqfbzpfqrovnznmanzdofqabzfltznfqazdotbhopzednzomtdobmtumbb
eqzdzdopfosqgthmedoabdoqbmzenfwbdonszoadmbdofnssqtohnnfnpoas
qahdofzntumqczoahnuumfb
```

First thing I will do is a frequency analysis of the ciphertext and compare that with a frequency analysis of english characters. I used the tool [cryptotool.html] found on Arup's website to generate the frequency analysis.

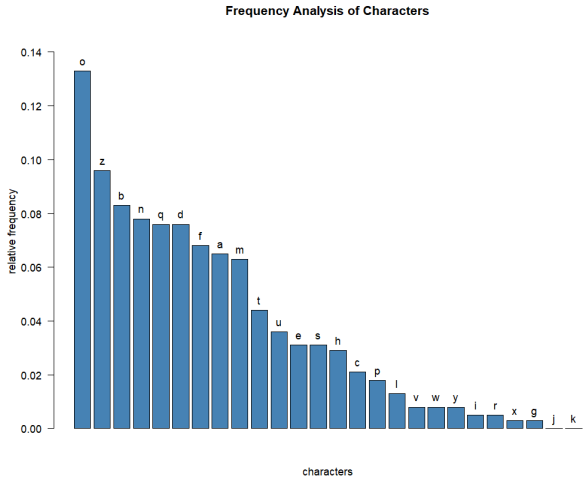


A	6.5%	N	7.8%
B	8.3%	O	13.3%
C	2.1%	P	1.8%
D	7.6%	Q	7.6%
E	3.1%	R	0.5%
F	6.8%	S	3.1%
G	0.3%	T	4.4%
H	2.9%	U	3.6%
I	0.5%	V	0.8%
J	0%	W	0.8%
K	0%	X	0.3%
L	1.3%	Y	0.8%
M	6.3%	Z	9.6%

I used RStudio to generate my graphs.



English character frequencies



Ciphertext character frequencies

Since we are working with a substitution cipher, we can also look for repeated characters. Based on what we know, these are the ones to keep an eye out for:

We are deciphering a paragraph so we will want to check the relative occurrences of repeat characters while ignoring spaces and using the data to help guess the character substitutions. The dataset is shown in the figure on the right, which I grabbed from this pdf.

Letters	Frequency (per 10,000 letters)
aa	1
bb	1
cc	4
dd	13
ee	48
ff	11
gg	4
hh	6
ii	1
jj	0
kk	0
ll	56
mm	5
nn	8
oo	36
pp	10
qq	0
rr	14
ss	43
tt	56
uu	0
vv	0
ww	2
xx	0
yy	2
zz	0

To better illustrate the data, here are the common repeats we want to look out for:

Likely

L/T: 0.56%
E: 0.48%
S: 0.43%
O: 0.36%

Less likely

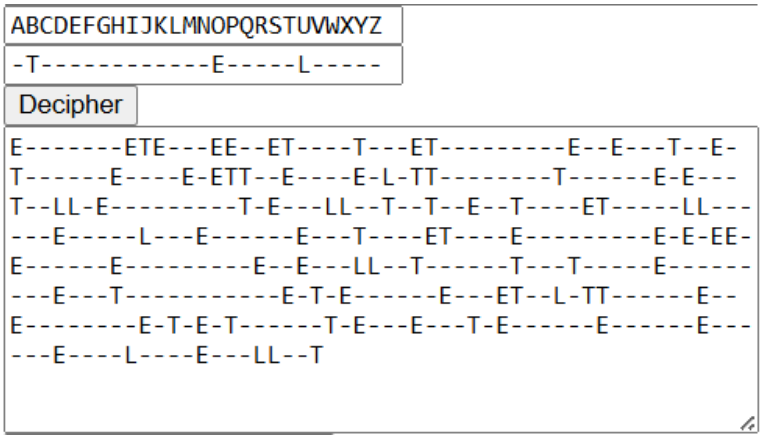
R: 0.14%
D: 0.13%
F: 0.11%
P: 0.10%

If we manually count the double letters in our ciphertext, this is what we find:
U: 5 B: 3 O: 2 C/N/S: 1

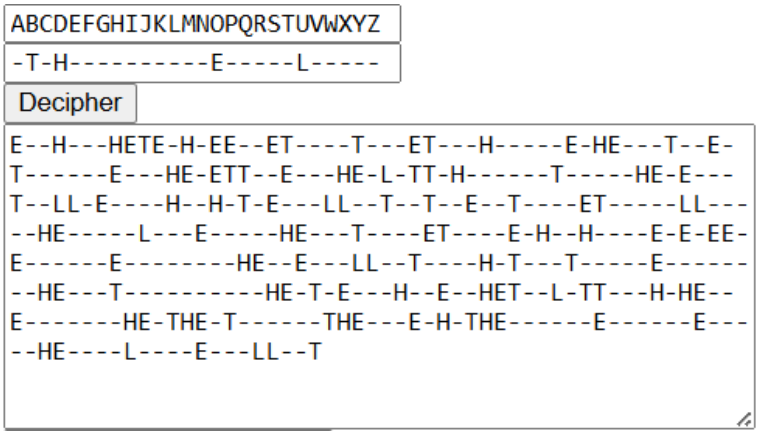
‘U’ has a high repeat count, but a relatively low frequency count in comparison, which makes me think it could be ‘L’.
U > L

‘B’ and ‘O’ also had high repeat counts, but also high frequency counts, which makes me think they could be either ‘E’ or ‘T’
B > E/T
O > E/T

Since O and E are both the most frequent letters in the frequency analysis, I settled on:
O > E
B > T



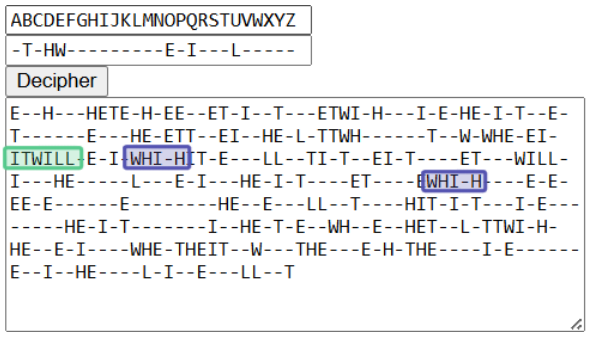
I then filled in H to satisfy the most common trigram ‘THE’

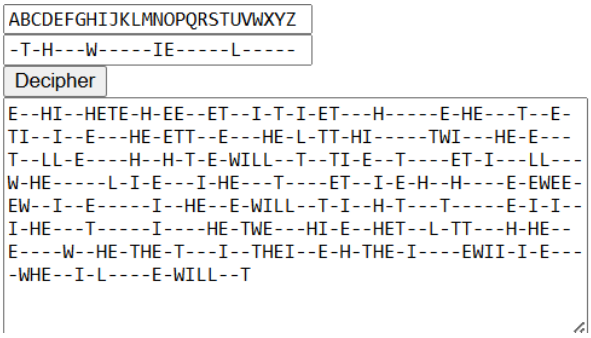


From this point onward, I simply tried making common words from the letters we already have. My first target was to see if the word ‘WILL’ is in here because we have 5 ‘LL’ pairs to work with and the letter ‘I’ is pretty common whereas ‘W’ is somewhat uncommon, making it fairly

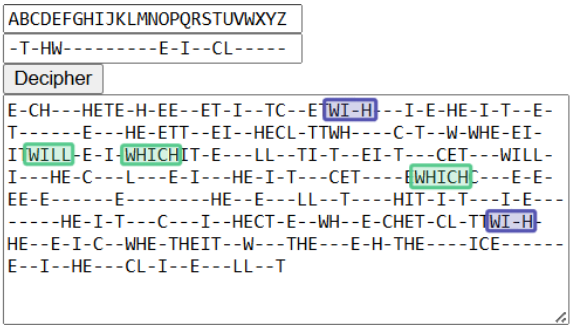
easy to identify a match. The letters that appear before ‘LL’ are N(3) and Q(2). We’ll also check what comes before as well to scope out any uncommon letters. Here are the results:
EQUU(2)
HNUU(3)

N/Q were both common and E/H were both uncommon frequencies for the ciphertext so we are left with a toss up. The only step now is to check and see which one looks more plausible.

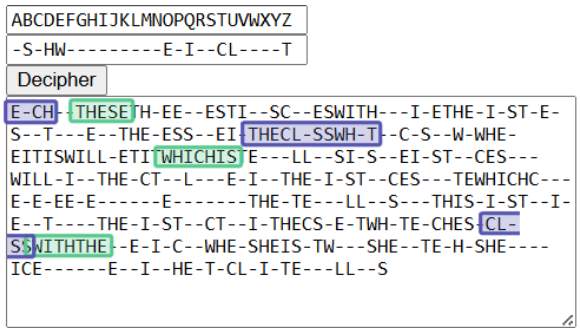




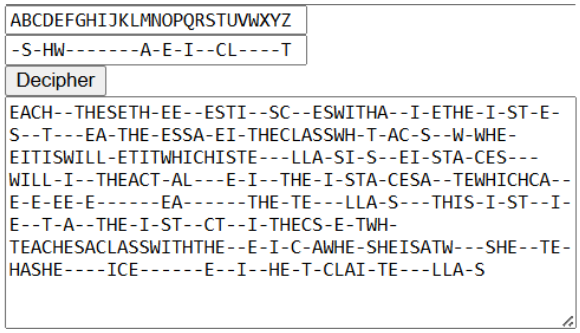
Substituting E and Q started to form words that looked like actual English. The next possible word I noticed was ‘WHICH’ so I went ahead and tried that.



I noticed that not many letters can make ‘WI_H’ make sense aside from ‘T’ so I tried placing the ‘T’ in that spot instead and replacing the old ‘T’ with ‘S’ since ‘IS’ and ‘IT’ are similar in frequency.



I was more confident with this rearrangement and I could also piece together ‘Each’, ‘Class’ and ‘The Class Wh_t…’.



The rest was just matching what made sense

ABCDEFGHIJKLMNOPQRSTUVWXYZ

-S-HW-------AOE-I-FCL----T

Decipher

```
EACHOFTHESETH-EE--ESTIO-SCO-ESWITHA--I-ETHEFI-ST-E-
SO-TO--EA-THE-ESSA-EI-THECLASSWHOT-AC-S-OW-WHE-
EITISWILL-ETITWHICHISTE--OLLA-SI-SO-EI-STA-CES-O-
WILLFI--THEACT-AL-O-E-I-OTHE-I-STA-CESA-OTEWHICHCA--
E-E-EE-E-F-O--EA---FO-THE-TE--OLLA-SFO-THISFI-ST--I-
E-OTOA-OTHE-I-ST--CTO-I-THECS-E-
TWHOTEACHESACLASSWITHTHE--EFI-C-AWHE-SHEISATWO--
SHEOFTE-HASHE-OFFICE-OO-O-E-FI--HE-TOCLAI-TE--OLLA-S
```

Each of these…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NS-HW--D---UAOE-I-FCL----T

Decipher

```
EACHOFTHESETH-EE-UESTIONSCO-ESWITHA--I-ETHEFI-ST-E-
SONTO--EA-THE-ESSA-EINTHECLASSWHOT-AC-SDOWNWHE-
EITISWILL-ETITWHICHISTENDOLLA-SINSO-EINSTANCES-
OUWILLFINDTHEACTUAL-ONE-INOTHE-
INSTANCESANOTEWHICHCAN-E-EDEE-EDF-O--EA-U-FO-
THENTENDOLLA-SFO-THISFI-ST--I-E-OTOANOTHE-INST-UCTO-
INTHECSDE-TWHOTEACHESACLASSWITHTHE--EFI-
CDAWHENSHEISATWO--SHEOFTENHASHE-OFFICEDOO-O-
ENFINDHE-TOCLAI-TENDOLLA-S
```

Will find the actual…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NSMHWR-D---UAOE-I-FCL--Q-T

Decipher

```
EACHOFTHESETHREEQUESTIONSCOMESWITHA-RI-ETHEFIRST-
ERSONTO-REA-THEMESSA-EINTHECLASSWHOTRAC-
SDOWNWHEREITISWILL-
ETITWHICHISTENDOLLARSINSOMEINSTANCES-
OUWILLFINDTHEACTUALMONE-
INOTHERINSTANCESANOTEWHICHCAN-EREDEEMEDFROMMEARU-
FORTHENTENDOLLARSFORTHISFIRST-RI-E-
OTOANOTHERINSTRUCTORINTHECSDE-
TWHOTEACHESACLASSWITHTHE-REFI-CDAWHENSHEISATWOR-
SHEOFTENHASHEROFFICEDOORO-
```

Each of these three questions comes with a…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NSMHWR-DY--UAOEPI-FCLGKQ-T

Decipher

```
EACHOFTHESETHREEQUESTIONSCOMESWITHAPRI-
ETHEFIRSTPERSONTO-
REAKTHEMESSAGEINTHECLASSWHOTRACKSDOWNWHEREITISWILL
GETITWHICHISTENDOLLARSINSOMEINSTANCESYOUWILLFINDTH
EACTUALMONEYINOTHERINSTANCESANOTEWHICHCAN-
EREDEEMEDFROMMEARUPFORTHENTENDOLLARSFORTHISFIRSTPR
I-
EGOTOANOTHERINSTRUCTORINTHECSDEPTWHOTEACHESACLASSW
ITHTHEPREFI-
CDAWHENSHEISATWORKSHEOFTENHASHEROFFICEDOOROPENFIND
```

The first person to…
The message in the class who tracks down…
In some instances you will find the actual money…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NSMHWRXDY--UAOEPIZFCLGKQBT

Decipher

```
EACHOFTHESETHREEQUESTIONSCOMESWITHAPRIZETHEFIRSTPERS
ONTOBREAKTHEMESSAGEINTHECLASSWHOTRACKSDOWNWHEREITISW
ILLGETITWHICHISTENDOLLARSINSOMEINSTANCESYOUWILLFINDT
HEACTUALMONEYINOTHERINSTANCESANOTEWHICHCANBEREDEEMED
FROMMEARUPFORTHENTENDOLLARSFORTHISFIRSTPRIZEGOTOANOT
HERINSTRUCTORINTHECSDEPTWHOTEACHESACLASSWITHTHEPREFI
XCDAWHENSHEISATWORKSHEOFTENHASHEROFFICEDOOROPENFINDH
ERTOCLAIMTENDOLLARS
```

Comes with a prize the first person to break the message…
Who teaches a class with the prefix CDA…

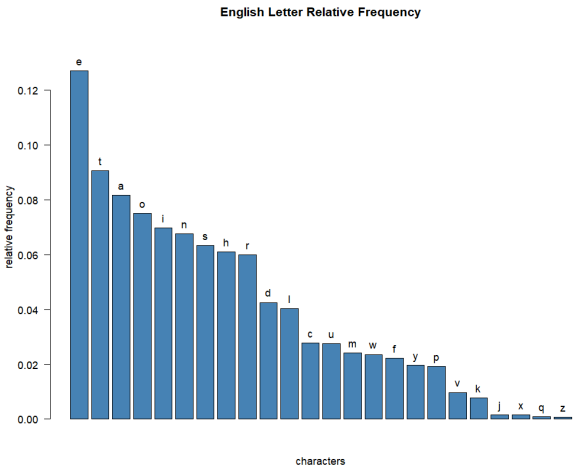
Each of these three questions comes with a prize, the first person to break the message in the class who tracks down where it is will get it which is ten dollars. In some instances you will find the actual money, in other instances a note which can be redeemed from me are? up for the? ten dollars. For this first prize, go to another instructor in the CS dept. who teaches a class with the prefix CDA. When she is at work she often has her office door open, find her to claim ten dollars.

2) Decode the following message, which was encrypted using the substitution cipher. Make sure to discuss all the steps you took, the key you arrived at, and the decoded message.

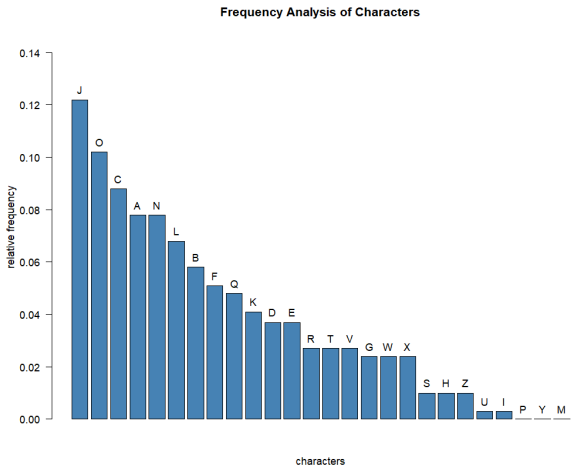
Here is the ciphertext:

```
ogawoaxjaranngadjlvajkjcfvlvjqlkcevcbfcqojbdcnnlqwoglotceklb
qjdjjjxfqcxxjbloeqlnntaoawkncwjocxtcffakjkcxjocxtcffakjlbdfl
kjogjdccqoeqbnjfobabjotdjhqjjwrlnzorjbotvlkjwfcqrlqdoeqbnjfo
lhlabrlnzfceqvlkjwoctceqnjforannsjlnaoonjscuseanoabocogjrlnn
cvjbaonafoogjdjiakjabaoevlbdtcerannfabdogjbcojhccdnekz
```

I'll be following a nearly identical method as I did to solve Q1 so I'll try not to repeat myself when it comes to explaining my steps.



English character frequencies



Ciphertext character frequencies

If we manually count the double letters in our ciphertext, this is what we find:
N: 6 C/F/J/O: 2 X: 1

‘J’ is special because there exists a triple repeat, which means it is a double repeat that very likely appears at the end of a word. This ‘J’ and our ‘N’ which has a high repeat count will be what we rely on.

E and J have the highest frequency counts for their graphs, and E has a high double letter count and I feel safe saying that there are plenty of words that end in ‘ee’.
J > E

We are left with the remaining letters with high double letter counts, L and T. However since N and T have relatively high frequency counts for their respective graphs, we’ll go with this substitution for now.
N > T

Let’s find the ‘THE’ trigram.

Probably “That” and “There”

Might be a bit of a stretch, but I could see the paragraph starting with “This time…” since it is building off of the paragraph we decrypted in Q1

“It is close to my…”?

ABCDEFGHIJKLMNOPQRSTUVWXYZ

I-OR--H--ECA-LT----Y--SM--

Decipher

```
THISTIMEI-ILLHIREA-IECEO--A-E-ACO--O--O-TE-ROLLA-
STHATYO-CA--EREEEM--OMME-AT--ALLYITISCLOSETOMYO--
ICECOMETOMYO--ICEA-R-ACETHEROO-T---LE-T-I-ETYRE--
EES-AL-T-E-TY-ACES-O--A-RT---LE-TA-AI--AL--O---
ACESTOYO--LE-T-ILL-EALITTLE-O---ILTI-TOTHE-ALLO-E-
ITLI-TTHERE-ICEI-IT--A-RYO--ILL-I-RTHE-OTE-OORL-C-
```

This time I will hire…
It is close to my office…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

I-OR-FH--ECA-LT--W-Y--SM--

Decipher

```
THISTIMEIWILLHIREA-IECEOF-A-E-ACO--O-FO-TE-ROLLA-
STHATYO-CA--EREEEMF-OMME-AT--
ALLYITISCLOSETOMYOFFICECOMETOMYOFFICEA-RFACETHEROO-
T---LEFT-I-ETYRE--EESWAL-TWE-TY-ACESFO-WA-RT---
LEFTA-AI-WAL-FO---ACESTOYO--LEFTWILL-EALITTLE-O---
ILTI-TOTHEWALLO-E-ITLIFTTHERE-ICEI-IT--A-RYO-WILLFI-
RTHE-OTE-OORL-C-
```

I realized I made a mistake, since it makes more sense for the D to replace the R, and to use the R elsewhere. This would form words like “dollars” and “from” which are words we are expecting.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

I-OD-FH--ECA-LT-RW-Y--SM--

Decipher

```
THISTIMEIWILLHIDEA-IECEOF-A-ERACO--O-FORTE-
DOLLARSTHATYO-CA-REDEEEMFROMME-AT-
RALLYITISCLOSETOMYOFFICECOMETOMYOFFICEA-
DFACETHEDOORT-R-LEFT-I-ETYDE-REESWAL-TWE-TY-
ACESFORWARDT-R-LEFTA-AI-WAL-FO-R-ACESTOYO-RLEFTWILL-
EALITTLE-O---ILTI-TOTHEWALLO-E-ITLIFTTHEDE-ICEI-IT--
A-DYO-WILLFI-DTHE-OTE-OODL-C-
```

I will hide a piece of…
That you can redeem…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

INODUFH--ECA-LT-RW-Y-PSM--

Decipher

```
THISTIMEIWILLHIDEAPIECEOFPAPERACOUPONFORTENDOLLARSTH
ATYOUCANREDEEEMFROMMENATURALLYITISCLOSETOMYOFFICECOM
ETOMYOFFICEANDFACETHEDOORTURNLEFTNINETYDE-REESWAL-
TWENTYPACESFORWARDTURNLEFTA-AINWAL-
FOURPACESTOYOURLEFTWILL-EALITTLE-O--
UILTINTOTHEWALLOPENITLIFTTHEDE-
ICEINITUPANDYOUWILLFINDTHENOTE-OODLUC-
```

Turn left ninety degrees walk…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

INODUFHG-ECA-LT-RW-Y-PSM-K

Decipher

```
THISTIMEIWILLHIDEAPIECEOFPAPERACOUPONFORTENDOLLARSTH
ATYOUCANREDEEEMFROMMENATURALLYITISCLOSETOMYOFFICECOM
ETOMYOFFICEANDFACETHEDOORTURNLEFTNINETYDEGREESWALKTW
ENTYPACESFORWARDTURNLEFTAGAINWALKFOURPACESTOYOURLEFT
WILL-EALITTLE-O--UILTINTOTHEWALLOPENITLIFTTHEDE-
ICEINITUPANDYOUWILLFINDTHENOTEGOODLUCK
```

Built into the wall…
Lift the device in it up and…

ABCDEFGHIJKLMNOPQRSTUVWXYZ

INODUFHGVECA-LT-RWBY-PSM-K

Decipher

```
THISTIMEIWILLHIDEAPIECEOFPAPERACOUPONFORTENDOLLARSTH
ATYOUCANREDEEEMFROMMENATURALLYITISCLOSETOMYOFFICECOM
ETOMYOFFICEANDFACETHEDOORTURNLEFTNINETYDEGREESWALKTW
ENTYPACESFORWARDTURNLEFTAGAINWALKFOURPACESTOYOURLEFT
WILLBEALITTLEBO-
BUILTINTOTHEWALLOPENITLIFTTHEDEVICEINITUPANDYOUWILLF
INDTHENOTEGOODLUCK
```

Will be a little box…

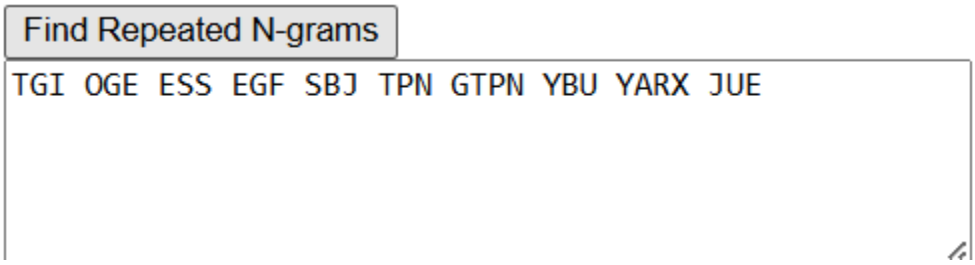
This time I will hide a piece of paper. A coupon for ten dollars that you can redeem from me. Naturally it is close to my office. Come to my office and face the door, turn left ninety degrees, walk twenty paces forward, turn left again, walk four paces to your left. Will be a little box built into the wall. Open it. Lift the device in it up and you will find the note. Good luck.

3) Decode the following message, which was encrypted using the Vigenere cipher. Make sure to discuss all the steps you took, the key you arrived at, and the decoded message.

tgicbsbwogessifdmnmsoqqfoegfiylzpjpbmrnuebarmkcytuelezosbjiew
zczotpnlbbacfgtpnbhikhzognaozfksfsovnaylslvxagcoghdxfqftxoefn
yixhmwybusebfywdryarxjuexhohwhpctemjlnrfigjltjrjislrtqcjrrcqs
jtsmltvomglsvmuestzvfvnlchvsomcmazrtfasxrdhmmtbtgifvtardroexy
fbcrlqimknjtpjbyabojfcbroaaaslusqgfresswgtpngeotrujtqbhrrdxyb
tqqaiesegfdbqeyarxtpuxxnvfxslbrmaendhrxuhqbybuviifajuegogeeel
mbievhkvoeznsuamhzxoztsbjnfnfltmoae

Naive Approach

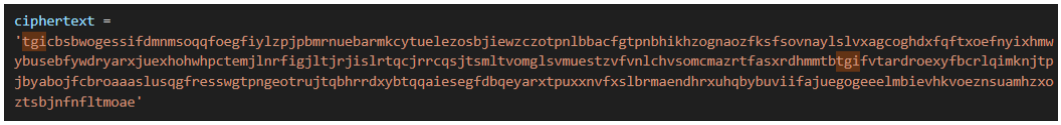
First thing I did was use the `cryptotool.html` tool provided to find repeated N-grams. I did modify the code slightly to allow for diagrams as well.



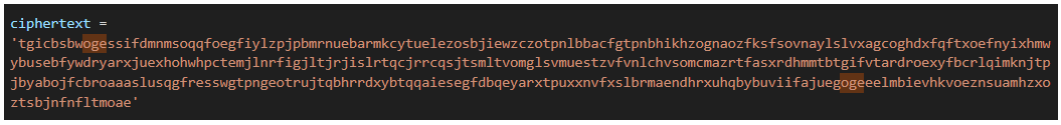
Then I wrote up a quick program in Python to find the distance(s) between a provided N-gram as input. Here is what I found:

```
(.venv) C:\Users\IBins\Desktop\Everything Folder\Coding\Cryptology>"C:/Users/IBins/Desktop/Everything Folder/Coding/Cryptology/.venv/Scripts/
python.exe" "c:/Users/IBins/Desktop/Everything Folder/Coding/Cryptology/HW2Solution.py"
[230, 352, 270, 287, 333, 10, 210, 210, 220, 180, 213]
{1: 11, 2: 8, 3: 6, 4: 3, 5: 7, 6: 4, 7: 3, 8: 1, 9: 3, 10: 7, 11: 2, 12: 1, 13: 0, 14: 2, 15: 4, 16: 1, 17: 0, 18: 2, 19: 0, 20: 2}
```

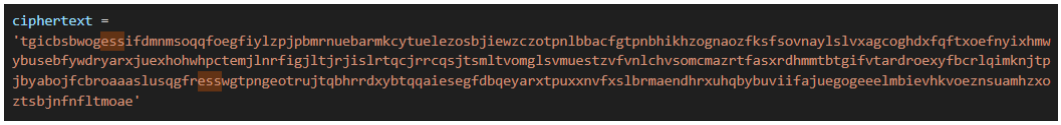
Here is how to make sense of the data it spat out. The first output details the distance between repeat N-grams. The second output prints a dictionary for distances that are valid under mod. For example, if we saw a spike for a mod 5, that would mean that there are more N-grams than normal that have a distance separation value which is a multiple of 5. Spikes in our data help us figure out what the keyword length potentially is.



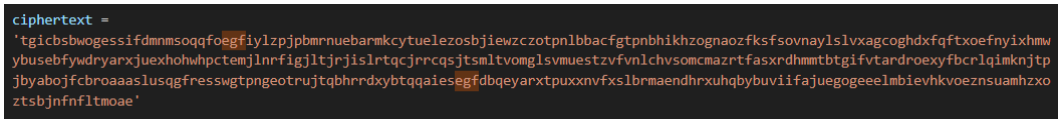
TGI: 230 digits apart



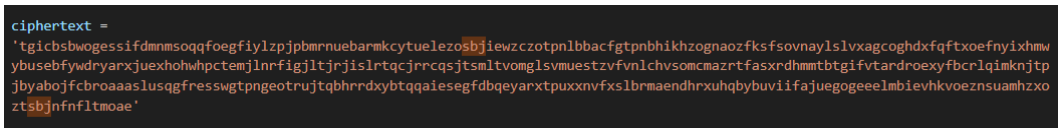
OGE: 352



ESS: 270

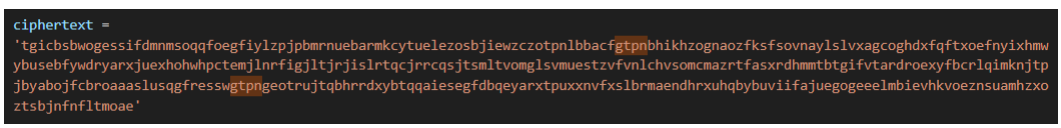


EGF: 287

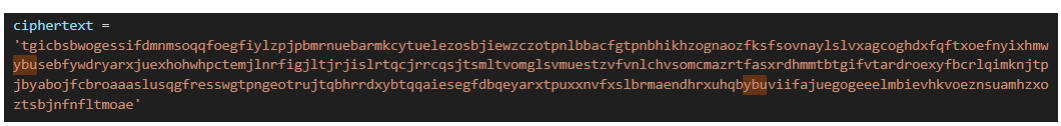


SBJ: 333

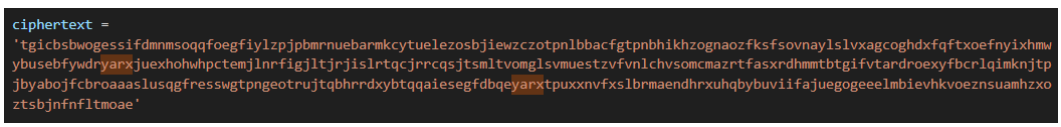
TPN: 10,210



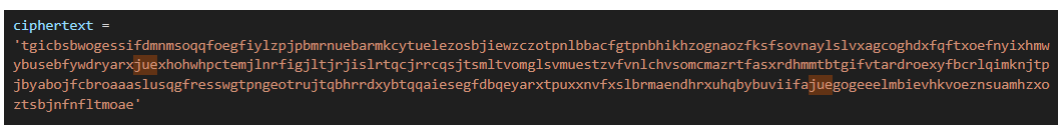
GTPN: 210



YBU: 220

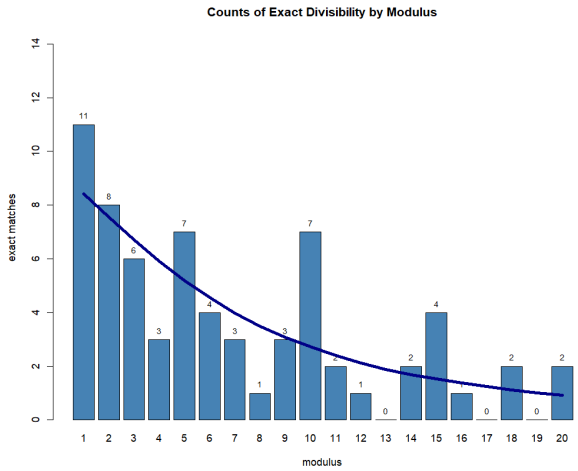


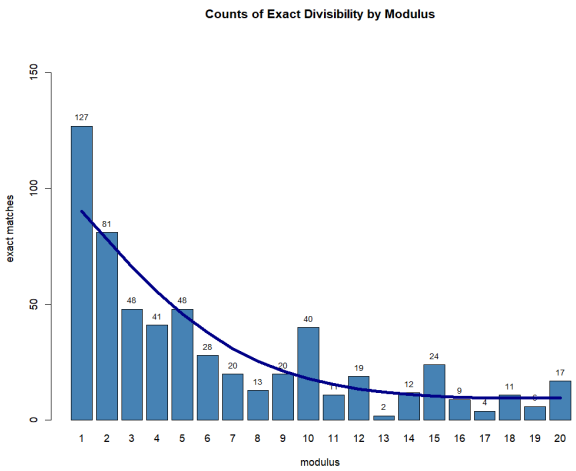
YARX: 180



JUE: 213

I also tinkered with the code on cryptotool.html so that it takes in n-grams of length 2 to get more data and it produced similar results. Here is how it looks visualized. We see a spike near 5, 10, and 15, with the biggest local spike being at 10. The keyword could have a length of either 5 or 10 so we'll check both.





We can then see where the letters end up when applying a mod 10 to our ciphertext. If they don't overlap nicely, then we will omit those specific N-grams. Once we pretty up our data, we will want to run a brute force algorithm to guess what word was used to encrypt these repeat trigrams.

```
tgi_______
ege_____og
ess_______
__egfegf__
j____sbjsb
_____tpn__
____gtpn__
u_______yb
arx______y
___juejue_
```

Sorting out the useful N-grams

```
tgi_______
ess_______
_____tpn__
____gtpn__
u_______yb
arx______y
```

After filtering the useful N-grams, we'll check for gaps.

Note: since there are 3 'tpn' repeats and only 2 'gtpn' repeats, I will be using only 'tpn' as it is more reliable and we will mainly be using trigram frequency data so we don't have to worry about gathering 4-gram data.

```
tgiXXtpnyb
```

We will ignore spaces where there is no useful information when brute forcing

The 77 most common trigrams *after blanks have been removed*:

the 1,49	*ing* 0.77	*tha* 0.52	*and* 0.50	*hat* 0.47	*ion* 0.45	*ent* 0.43
you 0.41	*thi* 0.38	*for* 0.38	*ati* 0.38	*tio* 0.38	*her* 0.35	*ere* 0.35
eth 0.34	*int* 0.32	*our* 0.28	*tth* 0.27	*all* 0.27	*rea* 0.26	*ter* 0.26
nth 0.26	*ome* 0.25	*hin* 0.25	*ver* 0.25	*not* 0.24	*res* 0.23	*est* 0.22
oul 0.22	*ont* 0.22	*ate* 0.21	*uld* 0.21	*ers* 0.21	*tin* 0.21	*oth* 0.20
pro 0.20	*sth* 0.20	*ons* 0.20	*his* 0.19	*ith* 0.19	*ave* 0.19	*eri* 0.19
sin 0.19	*ess* 0.18	*are* 0.18	*hav* 0.18	*ist* 0.18	*ill* 0.18	*out* 0.18
com 0.18	*rth* 0.18	*ese* 0.17	*ore* 0.17	*ple* 0.17	*con* 0.17	*one* 0.16
att 0.16	*iti* 0.16	*ert* 0.16	*ica* 0.16	*ein* 0.16	*eto* 0.16	*som* 0.16
han 0.15	*oft* 0.15	*nte* 0.15	*ine* 0.15	*sto* 0.15	*ted* 0.15	*ive* 0.15
ear 0.15	*fth* 0.15	*nce* 0.15	*ret* 0.14	*ngt* 0.14	*ble* 0.14	*lin* 0.14

We will be using this data from this pdf to grade the weighted score of the trigrams we find when searching for multiple trigram matches during our brute force search. More trigram matches and matching common trigrams will result in a higher score.

```
Keyword Matches Weight Trigrams
--------------------------------------------------
aze 2 1.65 ['tgi->the', 'ess->eto']
lop 2 0.33 ['tgi->ist', 'ess->ted']
llo 1 1.49 ['ess->the']
```

```
Keyword Matches Weight Trigrams
--------------------------------------------------
aij 1 1.49 ['tpn->the']
ain 1 0.52 ['tpn->tha']
tck 1 0.50 ['tpn->and']
mpu 1 0.47 ['tpn->hat']
lba 1 0.45 ['tpn->ion']
pcu 1 0.43 ['tpn->ent']
vbt 1 0.41 ['tpn->you']
obw 1 0.38 ['tpn->for']
aif 1 0.38 ['tpn->thi']
twf 1 0.38 ['tpn->ati']
ahz 1 0.38 ['tpn->tio']
lcu 1 0.32 ['tpn->int']
tec 1 0.27 ['tpn->all']
awg 1 0.27 ['tpn->tth']
alw 1 0.26 ['tpn->ter']
mha 1 0.25 ['tpn->hin']
gbu 1 0.24 ['tpn->not']
fcu 1 0.22 ['tpn->ont']
pxu 1 0.22 ['tpn->est']
aha 1 0.21 ['tpn->tin']
twj 1 0.21 ['tpn->ate']
zek 1 0.21 ['tpn->uld']
eyz 1 0.20 ['tpn->pro']
tuj 1 0.19 ['tpn->ave']
bha 1 0.19 ['tpn->sin']
lxu 1 0.18 ['tpn->ist']
fvu 1 0.18 ['tpn->out']
lec 1 0.18 ['tpn->ill']
tyj 1 0.18 ['tpn->are']
rba 1 0.17 ['tpn->con']
eej 1 0.17 ['tpn->ple']
twu 1 0.16 ['tpn->att']
pha 1 0.16 ['tpn->ein']
pyu 1 0.16 ['tpn->ert']
alk 1 0.15 ['tpn->ted']
owg 1 0.15 ['tpn->fth']
luj 1 0.15 ['tpn->ive']
mpa 1 0.15 ['tpn->han']
fku 1 0.15 ['tpn->oft']
gju 1 0.14 ['tpn->ngt']
clu 1 0.14 ['tpn->ret']
sej 1 0.14 ['tpn->ble']
iha 1 0.14 ['tpn->lin']
abs 0 0.00 []
```

```
Keyword Matches Weight Trigrams
--------------------------------------------------
fuq 1 1.49 ['ybu->the']
qoo 1 0.77 ['ybu->ing']
fuu 1 0.52 ['ybu->tha']
yor 1 0.50 ['ybu->and']
uob 1 0.43 ['ybu->ent']
ana 1 0.41 ['ybu->you']
fum 1 0.38 ['ybu->thi']
ftg 1 0.38 ['ybu->tio']
yim 1 0.38 ['ybu->ati']
tnd 1 0.38 ['ybu->for']
ukq 1 0.35 ['ybu->ere']
uin 1 0.34 ['ybu->eth']
qob 1 0.32 ['ybu->int']
fin 1 0.27 ['ybu->tth']
lin 1 0.26 ['ybu->nth']
hxu 1 0.26 ['ybu->rea']
rth 1 0.25 ['ybu->hin']
kob 1 0.22 ['ybu->ont']
ujb 1 0.22 ['ybu->est']
eqr 1 0.21 ['ybu->uld']
fth 1 0.21 ['ybu->tin']
ukc 1 0.21 ['ybu->ers']
yiq 1 0.21 ['ybu->ate']
koc 1 0.20 ['ybu->ons']
gin 1 0.20 ['ybu->sth']
kin 1 0.20 ['ybu->oth']
qin 1 0.19 ['ybu->ith']
rtc 1 0.19 ['ybu->his']
gth 1 0.19 ['ybu->sin']
ukm 1 0.19 ['ybu->eri']
wni 1 0.18 ['ybu->com']
hin 1 0.18 ['ybu->rth']
ujc 1 0.18 ['ybu->ess']
ujq 1 0.17 ['ybu->ese']
uig 1 0.16 ['ybu->eto']
gni 1 0.16 ['ybu->som']
qzu 1 0.16 ['ybu->ica']
qim 1 0.16 ['ybu->iti']
koq 1 0.16 ['ybu->one']
yib 1 0.16 ['ybu->att']
ukb 1 0.16 ['ybu->ert']
uth 1 0.16 ['ybu->ein']
qoq 1 0.15 ['ybu->ine']
ubd 1 0.15 ['ybu->ear']
tin 1 0.15 ['ybu->fth']
gig 1 0.15 ['ybu->sto']
liq 1 0.15 ['ybu->nte']
nth 1 0.14 ['ybu->lin']
abx 0 0.00 []
```

We don't have good data due to minimal overlap, but we can take refined data we now have and write another script to see which combinations between these could make sense.

Merge the 3 datasets using a different script and brute force again…

```
 1. aze??aijan
 2. aze??ainan
 3. aze??tckan
 4. aze??mpuan
 5. aze??lbaan
 6. aze??pcuan
 7. aze??vbtan
 8. aze??obwan
 9. aze??aifan
10. aze??twfan
11. aze??ahzan
12. aze??lcuan
13. aze??tecan
14. aze??awgan
15. aze??alwan
16. aze??mhaan
17. aze??gbuan
18. aze??fcuan
19. aze??pxuan
20. aze??twjan
21. aze??zekan
22. aze??eyzan
23. aze??tujan
24. aze??bhaan
25. aze??lxuan
26. aze??fvuan
27. aze??lecan
28. aze??tyjan
29. aze??rbaan
30. aze??eejan
31. aze??twuan
32. aze??phaan
33. aze??pyuan
34. aze??alkan
35. aze??owgan
36. aze??lujan
37. aze??mpaan
38. aze??fkuan
39. aze??gjuan
40. aze??cluan
41. aze??sejan
42. aze??ihaan
```

Now I will try this on a site that checks for valid words, such as https://www.onelook.com/ to find potential keywords.

`Azerbaijan`

`The last note to redeem for money will also be in hec. Walk up the main stairway in the lobby of the building from floor one all the way to the top floor four. When you take your last step you will see a large pillar with a square cross section curve. Left around this pillar its attached to the outside of the building with a black section about six feet off the ground. It is here that I have taped the last coupon. If you are reading this you were able to handle sir vigenere's handiwork so job well done`