2 Statistical Models and Cryptanalysis

2.1 Frequency Analysis of Monoalphabetic Ciphers

Fortunately, language has a structure to it: not all strings of letters are equally common. (If I tell you my message is either "brown" or "yoltk", you can make a pretty good guess as to which it is).

The first and simplest tool we have is the relative frequency of letters in English text. This approach is usually credited to the Arab philosopher Abu Yusuf Ya'qub ibn Ishaq Al-Kindi in the ninth century CE. The basic idea is that not all letters occur equally often; if your ciphertext has one letter appearing ten times in fifty total letters, it's probably not a "q" or "z".

There are also more sophisticated approaches we can take to frequency analysis, because English does not have its letters distributed in a random order. (That is, our model of the plaintext has more information than just the frequencies of individual letters). That is, if we see the same pair of letters appearing in the same order many times, we might guess that this pair is "th" or "he" or "an". If we see the same trio of letters appearing many times, we might guess that it is "the" or "and".

Below are tables showing the frequency with which each letter appears in English texts (Figure 2.1), and the frequencies of the most common English bigrams (Figure 2.2), drawn from Hoffstein, Pipher, and Silverman. (Note that different sources will have slightly different numbers due to using different corpuses).

E	13.11%	M	2.54%	A	8.15%	N	7.10%
T	10.47%	U	2.46%	B	1.44%	O	8.00%
A	8.15%	G	1.99%	C	2.76%	P	1.98%
O	8.00%	Y	1.98%	D	3.79%	Q	0.12%
N	7.10%	P	1.98%	E	13.11%	R	6.83%
R	6.83%	W	1.54%	F	2.92%	S	6.10%
I	6.35%	B	1.44%	G	1.99%	T	10.47%
S	6.10%	V	0.92%	H	5.26%	U	2.46%
H	5.26%	K	0.42%	I	6.35%	V	0.92%
D	3.79%	X	0.17%	J	0.13%	W	1.54%
L	3.39%	J	0.13%	K	0.42%	X	0.17%
F	2.92%	Q	0.12%	L	3.39%	Y	1.98%
C	2.76%	Z	0.08%	M	2.54%	Z	0.08%

Figure 2.1: English Letter Frequencies

th	he	an	re	er	in	on	at	nd	st	es	en	of	te	ed
168	132	92	91	88	86	71	68	62	53	52	51	49	46	46

Figure 2.2: Most common English bigrams (frequency per 1000 words)

Example 2.1. Let's try to decrypt the ciphertext:

```
JNRZR BNIGI BJRGZ IZLQR OTDNJ GRIHT USDKR ZZWLG OIBTM NRGJN IJTZJ LZISJ NRSBL
QVRSI ORIQT QDEKJ JNRQW GLOFN IJTZX QLFQL WBIMJ ITQXT HHTBL KUHQL JZKMM LZRNT
OBIMI EURLW BLQZJ GKBJT QDIQS LWJNR OLGRI EZJGK ZRBGS MJLDG IMNZT OIHRK MOSOT
QHIJL QBRJN IJJNT ZFIZL WIZTO MURZM RBTRZ ZKBNN LFRVR GIZFL KUHIM MRIGJ LJNRB
GKHRT QJRUU RBJLW JNRZI TULGI EZLUK JRUST QZLUK EURFT JNLKJ JNRXR S
```

We begin by counting the frequency of each letter, and put our results in figure 2.3

The most common letter is "R" so we'll guess that "R" is encrypting "e". We notice that the most common bigrams in the ciphertext are "JN" and "NR", and the most common bigrams in English are "th" and "he"; this leads us to guess that "JNR" is "the". (This is also reassuring since the second most common English letter is "t" and the second-most-common letter in the ciphertext is "J", to which we've just assigned the letter "t").

```
theZe BhIGI BteGZ IZLQe OTDht GeIHT USDKe ZZWLG OIBTM heGth ItTZt LZISt heSBL
QVeSI OeIQT QDEKt theQW GLOFh ItTZX QLFQL WBIMt ITQXT HHTBL KUHQL tZKMM LZehT
OBIMI EUeLW BLQZt GKBtT QDIQS LWthe OLGeI EZtGK ZeBGS MtLDG IMhZT OIHeK MOSOT
```

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency	0	15	0	5	5	6	15	8	27	30	13	25	12	19	10	0	16	33	9	20	12	2	7	3	0	24

Letter	R	J	I	L	Z	T	N	Q	B	G	K	U	M	O	S	H	W	F	E	D	X	V
Frequency	33	30	27	35	24	20	19	16	15	15	13	12	12	10	9	8	7	6	5	5	3	2

Bigram	JN	NR	TQ	LW	RB	RZ	JL
Frequency	11	8	6	5	5	5	5

Figure 2.3: Frequency count for example 2.1

```
QHItL QBeth ItthT ZFIZL WIZTO MUeZM eBTeZ ZKBhh LFeVe GIZFL KUHIM MeIGt LtheB
GKHeT QteUU eBtLW theZI TULGI EZLUK teUST QZLUK EUeFT thLKt theXe S
```

There are a few things we could do now. We can look at our other list of common bigrams. We see that "JL" is common in the ciphertext, which means we need a common English bigram whose first letter is "t"; "te" is the most common, but is ruled out since we know that "e" is "R". No others appear on our list.

We also have "RB" and "RZ" as common bigrams, and we know that "R" is "e". Looking at our list, it looks like these bigrams are probably "er" and "es". It's not entirely clear which should be which; one would make the first word of our text "there" and the second would give "these", which are both perfectly reasonable. It's not clear what to do here.

We can also just go ahead and guess that our next-most-common ciphertext letters "I" and "L" are our next-most-common plaintext letters "a" and "o". (This also makes "JL" into "to", which seems quite plausible!) That would give us

```
theZe BhaGa BteGZ aZoQe OTDht GeaHT USDKe ZZWoG OaBTM heGth atTZt oZaSt heSBo
QVeSa OeaQT QDEKt theQW GoOFh atTZX QoFQo WBaMt aTQXT HHTBo KUHQo tZKMM oZehT
OBaMa EUeoW BoQZt GKBtT QDaQS oWthe OoGea EZtGK ZeBGS MtoDG aMhZT OaHeK MOSOT
QHato QBeth atthT ZFaZo WaZTO MUeZM eBTeZ ZKBhh oFeVe GaZFo KUHaM MeaGt otheB
GKHeT QteUU eBtoW theZa TUoGa EZoUK teUST QZoUK EUeFT thoKt theXe S
```

At this point we might want to go looking through the text for guessable words. We see "at" several times, and might start noticing some patterns. What sticks out to me is the string "eth atth", which looks like it ends in "something-eth at th-". Almost certainly, the next letter should be a vowel; since we have "e,a,o" already spoken for, it should be "i" or "u". The ciphertext letter "T" is quite common; since "i" is a common English letter and "u" is not, we'll guess that "T" becomes "i".

```
theZe BhaGa BteGZ aZoQe OiDht GeaHi USDKe ZZWoG OaBiM heGth atiZt oZaSt heSBo
QVeSa OeaQi QDEKt theQW GoOFh atiZX QoFQo WBaMt aiQXi HHiBo KUHQo tZKMM oZehi
```

```
OBaMa EUeoW BoQZt GKBti QDaQS oWthe OoGea EZtGK ZeBGS MtoDG aMhZi OaHeK MOSOi
QHato QBeth atthi ZFaZo WaZiO MUeZM eBieZ ZKBhh oFeVe GaZFo KUHaM MeaGt otheB
GKHei QteUU eBtoW theZa iUoGa EZoUK teUSi QZoUK EUeFi thoKt theXe S
```

Looking back at our list of bigrams, we see that "TQ" is common. An English bigram whose first letter is "i" is probably "in", so we might guess that "Z" is "n"; but this gives some unlikely strings in our message like "thene BhaGa" or "that in tonaSt" or "-eth at thinFano" or "toW thenaiUoGa". While any one of these is possible, they don't seem likely. Unfortunately we don't have any other common i-initial bigrams to look at.

So let's go back to our idea that maybe "Z" is "s" or "r". "r" is the more common English letter, so we might try that first. "there BhaGa" seems plausible; but "that irtora" is improbable, as is "-eth at thirFaro" or "toW theraiUoGa".

So we try "s", and we get "these BhaGa"; "that is to sa", "-eth at this Faso" and "toW thesaiUoGa". The first three seem extremely likely, and the fouth possible, so we guess that "Z" is "s".

We can now look to sort out words, or just go back to our frequency charts. The next most common bigram is "TQ" which is "iQ", and the most common English bigram beginning with "i" is "in. Also, the next most common letter is "Q", and the next most common English letter is "n", so we might guess from both of these things that "Q" is "n".

```
these BhaGa BteGs asone OiDht GeaHi USDKe ssWoG OaBiM heGth atist osaSt heSBo
nVeSa Oeani nDEKt thenW GoOFh atisX noFno WBaMt ainXi HHiBo KUHno tsKMM osehi
OBaMa EUeoW Bonst GKBti nDanS oWthe OoGea EstGK seBGS MtoDG aMhsi OaHeK MOSOi
nHato nBeth atthi sFaso WasiO MUesM eBies sKBhh oFeVe GasFo KUHaM MeaGt otheB
GKHei nteUU eBtoW thesa iUoGa EsoUK teUSi nsoUK EUeFi thoKt theXe S
```

The next most common letter after this is "B". Our first thought is that we can take the next most common English letter of "r", but then we get "rhaGarte" which really doesn't look likely. On the other hand, if we replace our "G" with "r" we get "BharaBters" which suggests "B" for "c".

```
these chara cters asone OiDht reaHi USDKe ssWor OaciM herth atist osaSt heSco
nVeSa Oeani nDEKt thenW roOFh atisX noFno WcaMt ainXi HHico KUHno tsKMM osehi
OcaMa EUeoW const rKcti nDanS oWthe Oorea EstrK secrS MtoDr aMhsi OaHeK MOSOi
nHato nceth atthi sFaso WasiO MUesM ecies sKchh oFeVe rasFo KUHaM Meart othec
rKHei nteUU ectoW thesa iUora EsoUK teUSi nsoUK EUeFi thoKt theXe S
```

At this point we really should be looking to recognize words in teh text. The phrase "that is to saS theS" strongly suggests that "S" is "y". We see the string "constrKct" which

tells us "K" has to be a vowel; the only one left is "u", and "construct" is a reasonable word.

```
these chara cters asone OiDht reaHi UyDue ssWor OaciM herth atist osayt heyco
nVeya Oeani nDEut thenW roOFh atisX noFno WcaMt ainXi HHico uUHno tsuMM osehi
OcaMa EUeoW const ructi nDany oWthe Oorea Estru secry MtoDr aMhsi OaHeu MOyOi
nHato nceth atthi sFaso WasiO MUesM ecies suchh oFeVe rasFo uUHaM Meart othec
ruHei nteUU ectoW thesa iUora EsoUu teUyi nsoUu EUeFi thout theXe y
```

"sMecies" tells us that "M" is probably "p". "Fithout" suggests that "F" is "w", and then "without the Xey" seriously limits what "X" can stand for; we guess "X" is "k". "the saiUor" is probably "the sailor", so "U" is probably "l".

```
these chara cters asone OiDht reaHi lyDue ssWor Oacip herth atist osayt heyco
nVeya Oeani nDEut thenW roOwh atisk nowno Wcapt ainki HHico ulHno tsupp osehi
Ocapa EleoW const ructi nDany oWthe Oorea Estru secry ptoDr aphsi OaHeu pOyOi
nHato nceth atthi swaso WasiO plesp ecies suchh oweVe raswo ulHap peart othec
ruHei ntell ectoW thesa ilora Esolu telyi nsolu Elewi thout theke y
```

"aEsolutely" implies that "E" is "b". The most common leftover letter is "O"; we see it in "these characters, as one OiDht reaHily..." and in "not suppose hiO capable" and "the Oore abstruse cryptoDraph". From these, we guess "O" is "m" (especially to fill out "him") and then guess that "D" is "g", giving us "might" and "cryptograph".

```
these chara cters asone might reaHi lygue ssWor macip herth atist osayt heyco
nVeya meani ngbut thenW romwh atisk nowno Wcapt ainki HHico ulHno tsupp osehi
mcapa bleoW const ructi ngany oWthe morea bstru secry ptogr aphsi maHeu pmymi
nHato nceth atthi swaso Wasim plesp ecies suchh oweVe raswo ulHap peart othec
ruHei ntell ectoW thesa ilora bsolu telyi nsolu blewi thout theke y
```

Finally, "reaHily guess" implies that "H" is "d". We're running out of letters now; we look at the "V" and see it twice, in "that is to say they conVey a meaning" and "a simple species such howeVer as would", and it looks like "V" is actually "v"! We just need to translate the "W", and "but then Wrom" tells us that "W" is "f".

```
these chara cters asone might readi lygue ssfor macip herth atist osayt heyco
nveya meani ngbut thenf romwh atisk nowno fcapt ainki ddico uldno tsupp osehi
mcapa bleof const ructi ngany ofthe morea bstru secry ptogr aphsi madeu pmymi
ndato nceth atthi swaso fasim plesp ecies suchh oweve raswo uldap peart othec
rudei ntell ectof thesa ilora bsolu telyi nsolu blewi thout theke y
```

Going through and respacing, we get:

These characters, as any one might readily guess, form a cipher—that is to say, they

convey a meaning; but then from what is known of Kidd, I could not suppose him capable of constructing any of the more abstruse cryptographs. I made up my mind, at once, that this was of a simple species—such, however, as would appear to the crude intellect of the sailor, absolutely insoluble without the key.

And the key to the cipher is

Ciphertext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Plaintext	-	c	-	g	b	w	r	d	a	t	u	o	p	h	m	-	n	e	y	i	l	v	f	k	-	s
Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	I	E	B	H	R	W	D	N	T	-	X	U	O	Q	L	M	-	G	Z	J	K	V	F	-	S	-
Ciphertext	R	J	I	L	Z	T	N	Q	B	G	K	U	M	O	S	H	W	F	E	D	X	V				
Plaintext	e	t	a	o	s	i	h	n	c	r	u	l	p	m	y	d	f	w	b	g	k	v				

A few things to notice: by chance, "V" becomes "v", and "F" and "W" are interchanged. Nothing requires that sort of thing to happen, but nothing prohibits it either.

Also notice that there are some ciphertext and plaintext letters that we don't have correspondences for. The plaintext simply never used an "x" or a "z", so we don't know what rule it would have used for them, had it needed one. But if we got a future message in the same cipher, it would be quite easy to determine the meanings of the "A" and "C" in the message.

Note that this process requires experimentation and can take a number of wrong turns; I personally spent quite a while convinced that "L" was "i" in the preceeding cipher. If something isn't work, revisit one of your earlier guesses and try something different.

This sort of approach can fail in a few ways:

1. The message could be too short. If the message is ten letters long we can't possibly do any useful statistical analysis on it. (In fact a ten-letter message is generally impossible to decipher even in principle; an English message typically must be at least 27.6 letters to be amenable to frequency analysis. We'll discuss this general topic later, in **??**).

2. The text could be atypical. The pangram "A quick brown fox jumped over the lazy dog" is often used as a test sentence in many applications. But this message has four "o"s and only two "e"s, so statistical approaches will be somewhat misleading.

   The likelihood of this happening by accident, and the difficulty of it happening on purpose, decrease as the messages get longer. But it's completely possible that even a long message is highly atypical in this way; Ernest Vincent Wright wrote a full novel, called "Gadsby", without using the letter "e".

3. The text might not even be English. For instance, in Portuguese the most common letter is "a", not "e", and the letter "t" barely cracks the top ten. If you do statistical analysis assuming the encrypted message is English, but it's actually Portuguese, you may never even make enough progress to realize your error.

4. The message might not be enciphered with a monoalphabetic substitution cipher at all.

The first problem is a problem of not enough data; the third and fourth problems are problems of having a flawed model. (The second problem is a mix of the two). Both of these are important problems that come up any time we do statistical analysis. If our modelling assumptions are wrong, all the statistics in the world won't help us.

However, this sort of statistical analysis is powerful enough and well-enough understood that monoalphabetic ciphers are considered pretty thoroughly insecure; despite the number of possible keys, anyone who knows what they're doing can break these ciphers easily, and this has been true for over a millennium.

2.2 The Index of Coincidence

So how can we tell if our modelling assumptions are good? One answer comes from the index of coincidence.

Definition 2.2. Let $\mathbf{s} = c_1 c_2 \dots c_n$ be a string of n letters. The *index of coincidence* of $mathbfs$ is denoted $\text{IndCo}(\mathbf{x})$ and is defined to be the probability that two randomly chosen characters in the string $\mathbf{s}$ are identical.

Proposition 2.3. *Let $\mathbf{s} = c_1 c_2 \dots c_n$ be a string of n, and let F_i be the frequency with which the letter i appears in the string $\mathbf{s}$. Then*

$$\text{IndCo}(\mathbf{s}) = \frac{1}{n(n-1)} \sum_{i=0}^{25} F_i(F_i - 1). \tag{1}$$

Proof. There are $\binom{n}{2} = \frac{n(n-1)}{2}$ different ways to select two letters at random from $\sim$.

Each letter i appears F_i times, so there are $\binom{F_i}{2} = \frac{F_i(F_i)-1}{2}$ ways to choose the letter i twice. Adding these ways for each letter, there are

$$\sum_{i=0}^{25} \frac{F_i(F_i-1)}{2} \tag{2}$$

ways to choose two of the same letter from the string.

The chance of two randomly chosen letters being identical is equal to the number of ways to choose two identical letters, divided by the total number of ways to choose letters. Thus we divide equation (2) by $\frac{n(n-1)}{2}$ and get the formula in equation (1). □

For any given string we can calculate the index of coincidence from the frequency table.

Example 2.4. In your homework, we encrypted the string $\mathbf{s}$ ="Rats live on no evil star". This message has twenty letters total; it has ten distinct letters, and each appears twice. Thus we have

$$\text{IndCo}(\mathbf{s}) = \frac{1}{20 \cdot 19} \cdot 10(2 \cdot 1) = \frac{1}{19} \approx 0.53.$$

We also looked at the string $\mathbf{t}$ = "A man a plan a canal panama". This message has 21 total letters, with 10 "a"s, 2 "m"s, 4 "n"s, 2 "p"s, 2 "l"s, and 1 "c". Thus we compute

$$\text{IndCo}(\mathbf{t}) = \frac{1}{21 \cdot 20}(10 \cdot 9 + 2 \cdot 1 + 4 \cdot 3 + 2 \cdot 1 + 2 \cdot 1 + 1 \cdot 0) = \frac{108}{420} \approx .257.$$

As we'll see, this index of coincidence is really unusually high. But this isn't really surprising; you probably noticed already that this text has a ridiculous number of "a"s in it.

(Notice also that the term from the "c" is $1 \cdot 0 = 0$. This makes sense because the odds of the "c" matching another letter in the text are in fact zero, since there's only one of them').

We can also calculate two very important and common values for the index of coincidence:

Proposition 2.5. *1. If* $\mathbf{s}$ *is a string of letters generated uniformly at random, then* $\text{IndCo}(\mathbf{s}) \approx .038$.

2. If $\mathbf{s}$ *is a string of letters with the frequencies common in written English, then* $\text{IndCo}(\mathbf{s}) \approx .068$.

Proof. 1. The letters are generated uniformly at random, so $F_i \approx \frac{n}{26}$ for each i. Thus we have

$$\begin{aligned}\text{IndCo}(\mathbf{s}) &\approx \frac{1}{n(n-1)} \sum_{i=0}^{25} \frac{n}{26}\left(\frac{n}{26} - 1\right) = \frac{1}{n(n-1)} \sum_{i=1}^{25} \frac{n^2}{26^2} - \frac{n}{26} \\ &= \frac{1}{n(n-1)}\left(\frac{n^2}{26} - n\right) = \frac{n/26 - 1}{n-1} \approx \frac{1}{26} \approx .038.\end{aligned}$$

We probably could have guessed this without working through the computation—if the letters are chosen randomly, the chances of two of them matching should indeed be about 1/26.

2. This is a straightforward if tedious calculation from the letter frequencies given in figure 2.1. I might write it up for here later.

□

Corollary 2.6. *If* $\mathbf{s}$ *is a string of letters corresponding to an English text encrypted by a simple (monalphabetic) substitution cipher, then we should expect* $\text{IndCo}(\mathbf{s}) \approx .068$.

Proof. A monoalphabetic substitution cipher is just a permutation of the alphabet; so while the frequency of individual letters is altered by applying a substitution cipher, the index of coincidence is not □

This gives us a way to test whether a string of letters was likely enciphered by a simple substitution cipher or not. We compute the index of coincidence of the string. If the index is close to .068 then we likely have a string encrypted with a monoalphabetic cipher. If the index is close to .038 then it is likely that our string is not encrypted monoalphabetically.

This test is especially useful when we proceed to break the Vigenère cipher in the next section.

2.3 Breaking the Vigenère cipher

Monoalphabetic ciphers are fairly simple to implement, but also quite easy to break. The Vigenère cipher is much harder to break; for three centuries it was known as "The Undecipherable Cipher". In 1854 Charles Babbage successfully broke it, and in 1863 Friedrich Kasiski published an attack.

2.3.1 Finding the Key Length

Cryptanalysis of the Vigenère cipher proceeds in two steps. The first (and more difficult) step is to determine the length of the key. There are two basic approaches to this, but both use the same basic idea.

The low-tech way is to look for repeated strings in the ciphertext. If the same string appears in more than one place, it's likely (but not definite!) that it's the same plaintext string encrypted by the same part of the keyword, so the distances between these reoccurrences gives us information about possible keyword lengths.

A simple version of this is to displace the ciphertext by $2, 3, 4, \ldots$ places, and see which displacement generates the most coincidences—points where the displaced ciphertext is identical to the original.

Another variant is to look for places where the same trigram is repeated more than once in the ciphertext, and measure the offset or distance between them. Then find a number that is a factor of most (but not necessarily all) of thse offset numbers; that's probably the length of the keyword.

This method is called the *Kasiski Method* or the *Kasiski Test*, since it was first developed by Charles Babbage.

Example 2.7. Consider the ciphertext :

```
zpgdl rjlaj kpylx zpyyg lrjgd lrzhz qyjzq repvm swrzy rigzh
zvreg kwivs saolt nliuw oldie aqewf iiykh bjowr hdogc qhkwa
jyagg emisr zqoqh oavlk bjofr ylvps rtgiu avmsw lzgms evwpc
dmjsv jqbrn klpcf iowhv kxjbj pmfkr qthtk ozrgq ihbmq sbivd
ardym qmpbu nivxm tzwqv gefjh ucbor vwpcd xuwft qmoow jipds
fluqm oeavl jgqea lrkti wvext vkrrg xani
```

First, we can compute that the index of coincidences for this ciphertext is about .039. This suggests it wasn't encrypted with a monoalphabetic substitution cipher.

We think that it was encrypted with a Vigenère cipher, and we want to find the key length. The first method is to look for coincidences. To do that we can lay out the lines of ciphertext shifted. So the first line would be:

```
0:zpgdl rjlaj kpylx zpyYg lrjgd lrzhz qyjzq repvm swrzy rigzh
1: zpgd lrjla jkpyl xzpYy glrjg dlrzh zqyjz qrepv mswrz yrigz h
```

1 coincidence

```
0:zpgdl rjlaj kpylx zpyyg lrjgd lrzhZ qyjzq repvm swrzy rigzh
2:  zpg dlrjl ajkpy lxzpy yglrj gdlrZ hzqyj zqrep vmswr zyrig zh
```

1 coincidence

```
0:zpgdl rjLaJ kpylx zpyyg lrjgd lrzhz qyjzq repvm swrzy Rigzh
3:   zp gdLrJ lajkp ylxzp yyglr jgdlr zhzqy jzqre pvmsw Rzyri gzh
```

3 coincidences

```
0:zpgdl rjlaj kpylx zpyyg lrjGd lrzhz qyjZQ repvm swrzy rigzh
4:    z pgdlr jlajk pylxz pyyGl rjgdl rzhZQ yjzqr epvms wrzyr igzh
```

3 coincidences

```
0:zpgdl rjlaj kpylx zPYyg lrjgd LRzhz qyjzq repvm swrzy rigZh
5:       zpgdl rjlaj kPYlx zpyyg LRjgd lrzhz qyjzq repvm swrZy rigzh
```

5 coincidences

```
0:zpgdl rjlaj kpyLx zpyYg lrjgd lrzhz qyjZq repvm swrzy rigzh
2:        zpgd lrjLa jkpYl xzpyy glrjg dlrZh zqyjz qrepv mswrz yrigz h
```

3 coincidences

```
0:zpgdl rjlaj kpylx zpyyg lrjgd lrzhz qyjzq repvm swrzy rigzh
2:         zpg dlrjl ajkpy lxzpy yglrj gdlrz hzqyj zqrep vmswr zyrig zh
```

2 coincidences

From this we might guess that the keyword has length five; but this is a small sample size. If we do this count for the entire ciphertext (preferably but not necessarily by computer), we get:

Shift	1	2	3	4	5	6	7	8	9
Coincidences	6	6	9	5	8	13	15	11	11

This suggests but does not prove that the keyword has length 7.

If we look at repeated trigrams instead, we get the following results:

Trigram	Places	Offset	Trigram	Places	Offset
avl	117 and 258	$141 = 3 \cdot 47$	bjo	86 and 121	$35 = 5 \cdot 7$
dlr	4 and 25	$21 = 3 \cdot 7$	gdl	3 and 24	$16 = 2^4$
lrj	5 and 21	$98 = 2 \cdot 7^2$	msw	40 and 138	$84 = 2^2 \cdot 3 \cdot 7$
pcd	149 and 233	$13 = 13$	qmo	241 and 254	$98 = 2 \cdot 7^2$
vms	39 and 137	$84 = 2^2 \cdot 3 \cdot 7$	vwp	147 and 231	$84 = 2^23 \cdot 7$
wpc	148 and 232	$21 = 3 \cdot 7$	zhz	28 and 49	$21 = 3 \cdot 7$

We see that the factor 7 appears often in the offset column; thus the keyword is probably length 7. This matches our tentative conclusion from earlier.

The higher-tech version of this is to use the index of coincidence again. The index of coincidence only pays attention to *letter* distributions, and doesn't care about their order. So if we take all the letters that are encrypted by the same letter of the keyword, and calculate the index of coincidence, we should get a number close to .068; if not, we should get an index closer to .038.

To run this test, we break our ciphertext into k pieces: the first has letters $1, k+1, 2k+1, \ldots$, the second has letters $2, k+2, 2k+2, \ldots$, and so on. We compute the index of coincidence for each of these strings. If most of them are close to .068 then the keyword is probably of length k; if most of them are close to .038 then the keyword is probably not of length k.

Example 2.8. Continuing the look at our previous ciphertext, we can compute the indices of coincidence for each substring, for each possible shift.

Shift	indices							
2	.038	0.40						
3	0.39	0.42	0.38					
4	0.34	0.42	0.39	0.35				
5	0.38	0.39	0.43	0.28	0.36			
6	0.38	0.40	0.39	0.38	0.32	0.33		
7	0.62	0.57	0.65	0.60	0.60	0.64	0.64	
8	0.37	0.29	0.38	0.33	0.34	0.57	0.40	0.39

One of these rows looks very unlike the others; this again gives evidence that the key length is 7.

2.3.2 Finding the Key

Once we have found the key length, there are two basic approaches we can use to finding the key.

First, we can simply do a frequency analysis on subsets of the ciphertext. If we believe that the key has length 5, then the first, sixth, eleventh, etc. letters are all shifted by the same amount. So we can do a frequency analysis on this set to decipher the shift.

Note that the frequency analysis is made much easier by the fact that we know we're working with a Caesar cipher, so there are only 26 possibilities. Thus we're trying to select a shift that makes *all* the high-frequency ciphertext letters into high-frequency plaintext letters.

Example 2.9. In the previous example, we've now seen that the key length is seven. So we can look at the substring $\mathbf{s}_1$ consisting of the 1st, 8th, 15th, ... letters, which is `zlxrh rrhwl oehdw eokli lwvlh phqby nwhwf julrx x` which has frequency counts

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency	0	1	0	1	2	1	0	6	1	1	1	6	0	1	2	1	1	4	0	0	1	1	5	3	1	1

We see the frequent letters here are H, L, and less so W, R, and X in that order. We might guess that either H or L corresponds to a plaintext e. If H corresponds to e, that sends L to i, R to o, W to t, and X to u, which isn't unreasonable; if L corresponds to e, then that sends H to a, R to k, W to p, and X to q, which seems somewhat less likely, since we don't expect k and q to be among the most common letters in an English string.

This isn't dispositive, but it is highly suggestive that the first letter of the keyword corresponds to a shift three to the right; thus the first letter might be D. We can repeat this process for each substring.

This works, but involves a lot of guesswork and intuition and puzzle-solving. There is a second method that requires a bit more calculation, but is also rather more robust and automatic.

Definition 2.10. Let $\mathbf{s} = c_1c_2\dots c_n, \mathbf{t} = d_1d_2\dots d_m$ be two strings of letters. Then we define the *mutual index of coincidence* to be $\text{MutIndCo}(\mathbf{s},\mathbf{t})$, the chance that a randomly selected letter of $\mathbf{s}$ is the same as a randomly selected letter of $\mathbf{t}$.

Proposition 2.11. *Let $\mathbf{s} = c_1c_2\dots c_n, \mathbf{t} = d_1d_2\dots d_m$ be two strings of letters, and let $F_i(\mathbf{s})$ be the number of times the ith letter appears in the string $\mathbf{s}$. Then:*

1. $$\text{MutIndCo}(\mathbf{s},\mathbf{t}) = \frac{1}{nm}\sum_{i=0}^{25} F_i(\mathbf{s})F_i(\mathbf{t}).$$
2. *If the letters of $\mathbf{s}$ and $\mathbf{t}$ are drawn from the same distribution, given by taking English frequencies and permuting the letters, then* $\text{MutIndCo}(\mathbf{s},\mathbf{t}) \approx .068$.
3. *If the letters of $\mathbf{s}$ and $\mathbf{t}$ are drawn from different such distributions, then* $\text{MutIndCo}(\mathbf{s},\mathbf{t}) \approx .038$.

We can use the mutual index of coincidence to test if two strings are drawn from the same substitution. To decrypt a Vigenere cipher, we need to figure out the shift of each substring. We can use the mutual index of coincidence to compute the *relative* shifts.

Let $\mathbf{s_i} = c_i, c_{i+k}, c_{i+2k}, \dots$, and define $\mathbf{s}_i + \sigma$ to be the string $\mathbf{s}_i$ with each letter shifted by σ. Then we can compute $\text{MutIndCo}(\mathbf{s}_i, \mathbf{s}_j + \sigma)$ for $0 \le \sigma \le 25$. We expect 25 of these computations to be low like .038, and the last to be high like .068; this last one gives us the relative shift between the ith and jth letter of the keyword.

Thus if β_i is the ith letter of the keyword, this does not and cannot tell us what β_i is; but it can give us relationships among the β_i.

After computing $k-1$ of these, with luck we should know the relative shifts of all the letters of the keyword; this means there are only 26 possible keys, and we can try all of them. Of course, sometimes the mutual index of coincidence happens, randomly to not give enough information; this is a problem we can solve by simply computing more possible mutual indices, possibly up to all $\frac{k(k-1)}{2}$ possibilities.

Example 2.12. If we compute all the possible mutual indices of coincidence in our ciphertext, for a key length of seven, then we get the following "large" indices that indicate a true collision:

i	j	σ	MutIndCo$(i, j+\sigma)$	Relative shift equation
1	3	1	.067	$\beta_1 - \beta_3 = 1$
3	7	10	.069	$\beta_3 - \beta_7 = 10$
1	4	19	.071	$\beta_1 - \beta_4 = 19$
1	6	16	.071	$\beta_1 - \beta_6 = 16$
3	4	18	.073	$\beta_3 - \beta_4 = 18$
3	5	24	.067	$\beta_3 - \beta_5 = 24$
3	6	15	.074	$\beta_3 - \beta_6 = 15$
4	6	23	.066	$\beta_4 - \beta_6 = 23$
4	7	18	.071	$\beta_4 - \beta_7 = 18$
6	7	21	.069	$\beta_6 - \beta_7 = 21$

Our goal is to solve the equations in the right-hand column—or at least as many as possible! It's entirely possible that one of the high indices will be an accident; when this happens, you can try dropping one constraint or another and see what you get.

But in this case, the system is fairly straightforward to solve. We get:

$$\beta_3 = \beta_1 + 25 \qquad \beta_4 = \beta_1 + 7$$
$$\beta_6 = \beta_1 + 10 \qquad \beta_7 = \beta_3 + 16 = \beta_1 + 15$$
$$\beta_5 = \beta_3 + 2 = \beta_1 + 1$$

and we can check that this doesn't generate any inconsistencies. Notice that we don't have a solution for β_2 in here, because we didn't get any high indices of coincidence involving $\mathbf{s}_2$. One option is to take the best ones we have; the highest is MutIndCo$(2, 4+24) = .061$, which suggests $\beta_2 = \beta_4 + 24 = \beta_1 + 5$. The other is to look at the results not involving β_2 and basically guess.

So what do our results give us? Well, they tell us that if we know β_1, we know the entire keyword. For instance, if $\beta_1 = 0 = A$, then the keyword is AFZHBKP. If we try decrypting our ciphertext with this word (by *subtracting* it from the ciphertext), we get `zkhwkhulvkdoowxuq...` which isn't very promising.

But recall that there are now only 26 possible keys, so we can simply try each of them. If we do that, we get a table something like this:

β_1	Keyword	Potential plaintext
0	AFZHBKP	`zkhwkhulvkdoowxuq`
1	BGAICLQ	`yjgvjgtkujcnnvwtp`
2	CHBJDMR	`xifuifsjtibmmuvso`
3	DICKENS	`whetherishallturn`
4	EJDLFOT	`vgdsgdqhrgzkkstqm`
5	FKEMGPU	`ufcrfcpgqfyjjrspl`
6	GLFNHQV	`tebqebofpexiiqrok`

We see that when $\beta_1 = 3$, the keyword is "DICKENS", and we get recognizable English out of the ciphertext: we get

`wheth erish alltu rnout tobet heher oofmy ownli feorw hethe rthat stati onwil lbehe ldbya nybod yelse these pages musts howto begin mylif ewith thebe ginni ngofm ylife ireco rdtha tiwas borna sihav ebeen infor medan dbeli eveon afrid ayatt welve ocloc katni ghtit wasre marke dthat thecl ockbe ganto strik eandi began tocry simul taneo usly`

and after replacing spacing and punctuation, we get:

"Whether I shall turn out to be the hero of my own life, or whether that station will be held by anybody else, these pages must show. To begin my life with the beginning of my life, I record that I was born (as I have been informed and believe) on a Friday, at twelve oclock at night. It was remarked that the clock began to strike, and I began to cry, simultaneously."

Remark 2.13. These sorts of attacks can totally work on a Vigenère cipher applied to binary messages. But you have to rely entirely on things more like trigram coincidences and less like single-letter coincidences, since there are only two "letters".