Homework 1
CSCE 557/MATH 587, Spring 2016
Noemi Glaeser

1 The ciphertext and plaintext

The ciphertext:

```
xkjurowmllpxwznpimbvbqjcnowxpcchhvvfvsllfvxhazityxohulxqojaxelxzxmyjaqfstsr
ulhhucdskbxknjqidallpqslluhiaqfpbpcidsvcihwhwewthbtxrljnrsncihuvffuxvoukjlj
swmaqfvjwjsdyljogjxdboxajultucpzmpliwmlubzxvoodybafdskxgqfadshxnxehsaruojaq
fpfkndhsaafvulluwtaqfrupwjrszxgpfutjqiynrxnyntwmhcukjfbirzsmehhsjshyonddzzn
tzmplilrwnmwmlvuryonthuhabwnvw
```

The plaintext:

```
wheninthecourseofhumaneventsitbecomesnecessaryforonepeopletodissolvethepoli
ticalbandswhichhaveconnectedthemwithanotherandtoassumeamongthepowersoftheea
rththeseparateandequalstationtowhichthelawsofnatureandofnaturesgodentitleth
emadecentrespecttotheopinionsofmankindrequiresthattheyshoulddeclarethecause
swhichimpelthemtotheseparation
```

Once formatted, we see that the plaintext is the first paragraph of the Declaration of Independence:

> *When in the Course of human events, it becomes necessary for one people to dissolve the political bands which have connected them with another, and to assume among the powers of the earth, the separate and equal station to which the Laws of Nature and of Nature's God entitle them, a decent respect to the opinions of mankind requires that they should declare the causes which impel them to the separation.*

To crack the ciphertext, I wrote a Java program that analyzes the ciphertext while allowing the user to maintain some control over what key length to use and what shifts to use in the multiple sequences of the Vignere cipher. The calculations performed by the program ensure the user is able to make informed decisions, and, to a certain extent, reverse incorrect ones without having to restart from scratch. The program is broken into two parts: finding the key length and deciphering the ciphertext through frequency analysis. These steps are described in the sections below.

2 Finding the key length

To find the key length, we use a method presented on the website Practical Cryptography.[1] The Index of Coincidence (I.C.) measures how similar a frequency distribution is to the standard English distribution using the following formula:

$$I.C. = \frac{\sum_{i=A}^{i=Z} f_i(f_i - 1)}{N(N-1)}$$

Here, f_i is the count of the ciphertext letter (A through Z) and N is the number of letters in the ciphertext.

The I.C. is unaffected by substitution ciphers, so it is ideal for our needs. The I.C. of English text is around 0.06, but that of a more random distribution is closer to zero. For the purposes of this program, we are looking for the highest I.C. to determine what the length of our key is.

To determine the most probably key length, the I.C. is calculated for each possible keyword length. So, starting at key length 2, we calculate the I.C. of the two sequences generated: the 1st, 3rd, 5th, ... letters of the ciphertext and the 2nd, 4th, 6th, ... letters. These two values are then averaged to find the I.C. of the ciphertext with an assumed key length of 2. This process is repeated with key length 3 (which entails averaging 3 I.C. values), key length 4, and so on, up to a user-specified maximum key length (I used 25).

The program then outputs a table of the I.C.s for each key length:

Period	Avg. I.C.
2:	0.04164818920916482
3:	0.04047817625799277
4:	0.03926274202501061
5:	0.07132867132867134
6:	0.04107744107744108
7:	0.04025153076956962
8:	0.04320557491289198
9:	0.038624338624338624
10:	0.07064393939393938
11:	0.03761755485893417
12:	0.0373931623931624
13:	0.03723865877712032
14:	0.03710239036325993
15:	0.07012987012987013
16:	0.040852130325814535
17:	0.041761265909872725
18:	0.038221916446890644
19:	0.0391296869625043
20:	0.06906862745098039
21:	0.03492063492063493
22:	0.03506493506493507
23:	0.03911450867972607
24:	0.04906898656898656
25:	0.06205128205128206

[1] http://practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher/

Next, the program prints the maximum I.C. and the key length at which it occurs — in this case, 0.07132867132867134 at key length 5. We can see that the other peak I.C.s occur at multiples of this number: 10, 15, 20, and 25 all have I.C.s between 0.06 and 0.07, which is the value we are looking for.

Because of the presence of these multiples and the peak I.C. at 5, we can conclude that the key is most likely 5 characters long (or a multiple thereof). *(Note: It is possible that the maximum I.C. might have been calculated at 10 instead, but still have a peak at 5. In this case, we would have looked at the table and noticed this trend and again concluded that the key length is a multiple of 5.)*

Now that we have the (probable) key length, the rest is trivial. Assuming a key length 5, we can now easily conduct frequency analysis to find the plaintext. (If using a length of 5 does not work, we might try multiples of 5, starting at 10, then 15, etc.)

3 Deciphering the Ciphertext

Now, the program prompts the user for a key length n to use for the remaining step (this should be the value at which the maximum I.C. occurs, calculated in the previous step). Next, it analyzes each of the n sequences one by one (i.e. Sequence 1 consists of the 1st, $(n+1)$th, $(2n+1)$th, ... letters of the ciphertext; Sequence 2 of the 2nd, $(n+2)$th, $(2n+2)$th, ...; and so on).

First, we take Sequence 1:

> XOPPBOCFFZOQEMFUCXIPUFIIETNIFUSFSOBUPIBOFGSEUFDFUFJGJRTUIES
> DTIMUTB

The program prints a frequency table of the letters in the sequence, and prompts the user for a ciphertext letter to translate to a plaintext e. In this sequence, F is the most common letter (with a count of 10), so we enter "f".

The program prints the entire ciphertext with the first sequence replaced by its plaintext counterpart:

> wKJURnWMLLoXWZNoIMBVaQJCNnWXPCbHHVVeVSLLeVXHAyITYXnHULXpOJ
> AXdLXZXlYJAQeSTSRtLHHUbDSKBwKNJQhDALLoQSLLtHIAQePBPChDSVChHW
> HWdWTHBsXRLJmRSNChHUVFeUXVOtKJLJrWMAQeVJWJrDYLJnGJXDaOXAJt
> LTUCoZMPLhWMLUaZXVOnDYBAeDSKXfQFADrHXNXdHSARtOJAQePFKNcHSA
> AeVULLtWTAQeRUPWiRSZXfPFUTiQIYNqXNYNsWMHCtKJFBhRZSMdHHSJrHY
> ONcDZZNsZMPLhLRWNlWMLVtRYONsHUHAaWNVW

We can now choose to accept the change or not. If we accept, the program moves on to the next sequence, Sequence 2, and we repeat the process. If not, we can choose a different decryption for Sequence 1.

We continue to proceed through the sequences in a similar way, choosing "H", "J", "L", and "N" as "e" for each successive sequence. *(Note: this does not yield a sensical keyword — it would be "bdfhj", which clearly is not a word in the English language.)*

Eventually, we arrive at the plaintext:

> whenin thecourseofhumaneventsitbecomesnecessaryforonepeopletodissolvethepoliticalbands
> whichhaveconnectedthemwithanotherandtoassumeamongthepowersoftheearththeseparatea
> ndequalstationtowhichthelawsofnatureandofnaturesgodentitlethemadecentrespecttotheopin
> ionsofmankindrequiresthattheyshoulddeclarethecauseswhichimpelthemtotheseparation