Nico Döttling
Lucjan Hanzlik

## Exercise 1 - Caesar Cipher

Find the plaintext of the following ciphertext which was encrypted using a Caesar cipher with an unknown shift.

WUYMULCMZCMBS

## Exercise 2 - Shift Cipher, Systematic Approach

Find the plaintext of the following ciphertext which was encrypted using a Caesar cipher with an unknown shift.

```
serdhrapln anylfvfvfn shaqnzragn ygrpuavdhr hfrqvapelc gnanylfvfg
bqrpvcurer apelcgrqzr ffntrfcneg vphyneylgu bfrrapelcg rqhfvatfvz
cyrfhofgvg hgvbapvcur efguronfvp cerzvfrbss erdhraplna nylfvfyvrf
vagursnpgg ungpregnva yrggrefbef lzobyfnccr nezberserd hragylvant
viraynathn trgunabgur efolnanylm vatgurserd hraplbsgur frpunenpgr
efjvguvana rapelcgrqz rffntrpelc gnanylfgfp naznxrrqhp ngrqthrffr
fnobhggurf hofgvghgvb afhfrqvagu rrapelcgvb acebprff
```

Rather than trying all 26 possible shifts by hand, you can write a small script to do the cryptanalysis in an automated way, exploiting *letter frequency distributions* in English. Define:

- $f_i$ as the relative frequency of the letter $\texttt{A} + i$ in typical English text (e.g., 'E' is the most common, 'Z' is the least).
- $c_j$ as the relative frequency of the letter $\texttt{A} + j$ in the *ciphertext*.

Given a candidate shift $k$, we can measure how well it aligns with English by computing the **correlation score**:

$$\text{Score}(k) = \sum_{i=0}^{25} c_{(i+k) \bmod 26} \times f_i.$$

You then:

1. Calculate this score for each $k \in \{0, 1, \ldots, 25\}$.
2. Identify the shift $\hat{k}$ that produces the *highest* score.
3. Decrypt the ciphertext by shifting letters back by $\hat{k}$.

You can use the Wikipedia page for the frequencies of the letters in the English language, or the following table.

```
freq_english = [
    0.08167, 0.01492, 0.02782, 0.04253, 0.12702, 0.02228,
    0.02015, 0.06094, 0.06966, 0.00153, 0.00772, 0.04025,
    0.02406, 0.06749, 0.07507, 0.01929, 0.00095, 0.05987,
    0.06327, 0.09056, 0.02758, 0.00978, 0.02360, 0.00150,
    0.01974, 0.00074
]
```

## Exercise 3 - Vigenère Cipher

Decrypt the following text, which is encrypted with a Vigenère cipher.

```
MLOKGIAYJG SITVKZGZWI QTBMTKTOZG POLAKFTOVI NYNMTEIYBG DOTBJQBGVM
MNJWHTABQP SNUBJUNMBQ POUVEQOXBY UCKAJQHGLR QEVMFUNZWV TEHWQWHKZU
USZMTIAYZG MDOVINUZQV TAJVQBIIBW DEYWTOOTDG DSGBKANYQP UTGVFIHGBK
ETNMWEEUNC NOUSVTOAOJ FARQEQWOBJ AUZXKOTAZG EOXKQZVKZU MTOWPESUAJ
QWGAEANYQF QROVIUNNMT AWTUKZDGAY QLRIUEHKKQ GLJNQDTNMJ ATJIAYAJMJ
QRLMGXVKZA ELKMRKATLU FUVQFIHKBJ QRZPGBLKIU GRKWHYAQQP SAJIKEYIPC
UNCWWXDHMY ARZPVTEZZQ GBRMQRGKBV UNMCRMNJXK OKOVIFHKLC USOMUIHKVU
GDJMPXYGEJ UTKZCNBOBY UTNXKZKKGG ERGVEXOYMD KHKZVTEXMY MSTWVTITOU
AVKZADESIT WAHTGUNZPC FNUZFUDGTK OEZPKZKOBU AVKZAYUIPQ GTUNVTECIA
FONMCDTNMT MBHQVEAEBQ UTYMNRONLG MRUPFQAXQU TARTDQLGBG IHKVUTEZPQ
GGNBKFOBMT MFZMTIAXLU UTUKEGRXMF FONMTFHGBU TEUCITTZWJ MVKEQZDKZG
PAZBJUSHCV MTZPGFISMK FARTUQESMF CUOBGZAZCT MLHCVIHKVV TEXIDNIZIE
FUGTNKTUWM MWGBETOABQ RIZAYMIYBE AAZXQOKKBC ZDRWQWEJIV UTGVFFHKVJ
GRXQGPOTIN UCKAVMRZMF FONMTREKBH AROBHXAYPG PAIZQESNMT YITLVTAZAJ
QHGLPQVKZD QFUZGEEKVC DAHJKFWOBJ QIZPGDACIK ETIWCFPUKM QTUZCIAZKJ
FOZIMQOABQ RIZIPPBAZP UNMEKFHICT UOYQVKSNMT MNGKTASYBJ QFOMNPALBG
DIZIPPFUZV GNGBGXYCIU VUYBKZTOUG FOYMGUTVWR POCVCXAXOG DAHJKFHUTG
GNJMTFHKPG PGK
```

Again, we can automate the cryptanalysis process with the help of a script doing some statistical analysis.

Recall that the first thing we have to do is to recover the length $m$ of the keyword $k$. One helpful tool in guessing the length $m$ is the *Index of Coincidence* (IC). For a string of length $N$, define:

$$\mathrm{IC} \;=\; \frac{1}{N(N-1)} \sum_{i=0}^{25} f_i\,(f_i - 1),$$

where $f_i$ is the count (not frequency) of letter $(\mathtt{A} + i)$ in the text. English text typically has $\mathrm{IC} \approx 0.0667$, while random text has $\mathrm{IC} \approx 0.0385$.

Using this tool, we can find the most probable length $m$ of the keyword by:

1. Guess a Key Length $m$.
2. Partition the ciphertext into $m$ columns, where column $j$ consists of every $m$-th character starting from position $j$.
3. Compute IC for each column and average them. If the average is close to the known English IC, that is a promising candidate for $m$.
4. Repeat for various $m$ (say from 1 up to some upper bound) and pick the one that yields an average IC near that of English text.

Once you *fix* a candidate key length $m$, each of the $m$ columns is effectively a *shift cipher* with some shift $k_j$. You can use the same correlation formula from Exercise 1 to determine that shift:

$$\text{Score}(k_j) \;=\; \sum_{i=0}^{25} c^{(j)}_{(i+k_j) \bmod 26}\, f_i,$$

where $c_r^{(j)}$ is the relative frequency of letter $\mathtt{A} + r$ in column $j$, and $f_i$ is again the typical English frequency for letter $\mathtt{A} + i$. Maximize this score over $k_j \in \{0, \ldots, 25\}$ to find the best shift for column $j$.