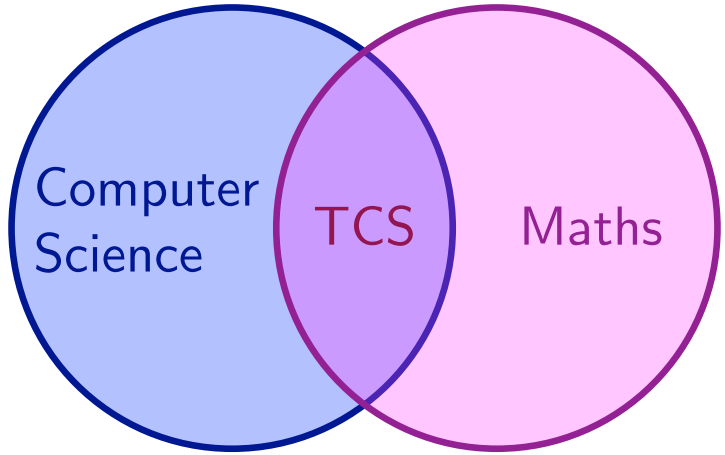
Breaking the Vigenère cipher

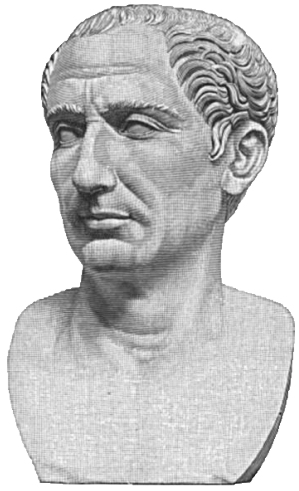
Dillon Mayhew

Outreach Coordinator and Professor of Theoretical Computer Science at the School of Computer Science, University of Leeds.



UNIVERSITY OF LEEDS

Caesar cipher



The Caesar cipher shifts every letter three places:

$$\begin{array}{cccccccccc} A & B & C & D & \cdots & W & X & Y & Z \\ \downarrow & \downarrow & \downarrow & \downarrow & \cdots & \downarrow & \downarrow & \downarrow & \downarrow \\ D & E & F & G & \cdots & Z & A & B & C \end{array}$$

YELLOW SUBMARINE is encrypted as BHOORZ VXEPDULQH.

UXEEHU VRXO is decrypted as RUBBER SOUL.

Shift ciphers

In general, a shift cipher shifts each letter the same number of places.

Shift ciphers

In general, a shift cipher shifts each letter the same number of places.

Try breaking this shift cipher:

NASVZE JASVZE YGZ UT G CGRR
NASVZE JASVZE NGJ G MXKGZ LGRR
GRR ZNK QOTMY NUXYKY GTJ GRR ZNK QOTMY SKT
IUARJTZ VAZ NASVZE ZUMKZNKX GMGOT

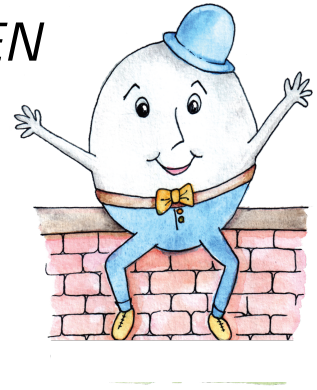
Shift ciphers

In general, a shift cipher shifts each letter the same number of places.

Try breaking this shift cipher:

NASVZE JASVZE YGZ UT G CGRR
NASVZE JASVZE NGJ G MXKGZ LGRR
GRR ZNK QOTMY NUXYKY GTJ GRR ZNK QOTMY SKT
IUARJTZ VAZ NASVZE ZUMKZNKX GMGOT

HUMPTY DUMPTY SAT ON A WALL
HUMPTY DUMPTY HAD A GREAT FALL
ALL THE KINGS HORSES AND ALL THE KINGS MEN
COULDNT PUT HUMPTY TOGETHER AGAIN



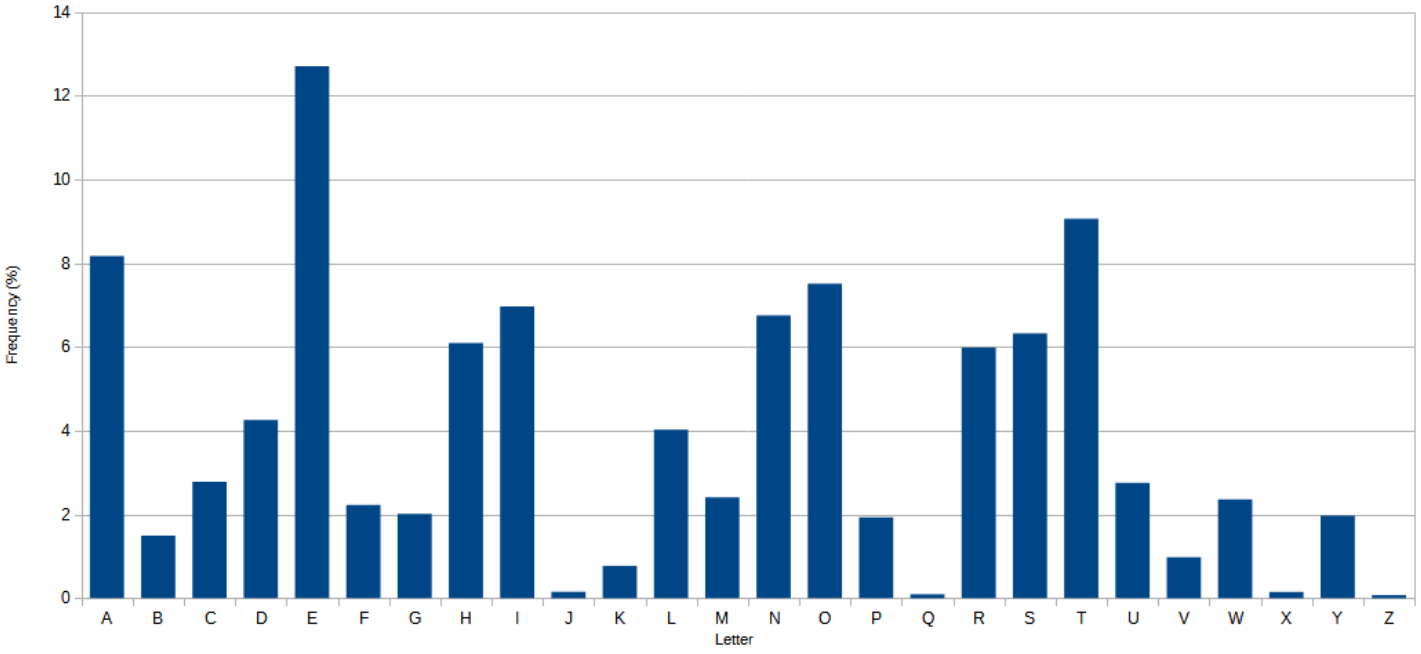
Substitution ciphers

A substitution cipher replaces each letter by another, not necessarily by shifting. (We might say that the cipher permutes the alphabet.)

$$\begin{array}{cccccccccc} A & B & C & D & \cdots & W & X & Y & Z \\ \downarrow & \downarrow & \downarrow & \downarrow & \cdots & \downarrow & \downarrow & \downarrow & \downarrow \\ K & A & E & M & \cdots & Q & T & E & L \end{array}$$

Substitution ciphers

Substitution ciphers are easy to break because the letters in English follow a distinctive pattern of frequencies.



This is very different from the distribution of letters selected uniformly at random.

Index of coincidence

We can measure how close a distribution is to uniform by calculating the index of coincidence.

This measures the probability that when we randomly select a pair of letters from a text of N letters, the pair will be identical.

$$\text{Index of coincidence} = \frac{\#A(\#A-1)}{N(N-1)} + \cdots + \frac{\#Z(\#Z-1)}{N(N-1)}$$

(assuming that each letter appears at least once).

If the text is chosen uniformly at random, the index of coincidence will be close to

$$\frac{1}{26} \approx 0.0385.$$

Index of coincidence

```
def IndexOfCoincidence(Text):
    CharacterCount = [0] * 26
    for Char in Text:
        if Char.upper() in Alphabet:
            i = Alphabet.index(Char.upper())
            CharacterCount[i] = CharacterCount[i] + 1
    Sum = 0
    N = len(RemoveSpaces(CleanText(Text)))
    for i in CharacterCount:
        if i > 1:
            Sum = Sum + ( i * (i-1) ) / ( N * (N-1) )
    return(Sum.numerical_approx(digits = 4))
```

Vigenère cipher

The Vigenère cipher was regarded as being unbreakable between 1550 and 1850 (approximately).

- Choose a key word. E.g. ALERT
- Repeat the key below the input text.
- Shift each letter by the number of places corresponding to that letter of the key.

Input:	*A*	*T*	*T*	*A*	*C*	*K*		*A*	*T*		*D*	*A*	*W*	*N*
Key:	*A*	*L*	*E*	*R*	*T*	*A*		*L*	*E*		*R*	*T*	*A*	*L*
Shift:	0	11	4	17	19	0		11	4		17	19	0	11
Output:	*A*	*E*	*X*	*R*	*V*	*K*		*L*	*X*		*U*	*T*	*W*	*Y*

Breaking the Vigenère cipher

This text has been encoded using the Vigenère cipher. Can we break it?

UIGIP FOFVI VVDLO SHPSR NSTBN EWIRH YNOWO SKLKN SCPTR EDFNE BAEHO RIGXT YRWEH

HREPI KSOXD ACHCP OCXSC FKLAW XTYTW SPAWI AMKPD EGMQI NATOX DQECI CAZTT FIXKP

EFYXS MPHOL ITWSC IJEDT KDYMT GDIMC WWMKO NZWGI KRTHR ECMPZ VECTG ODIDE HKRIM

HPZCO SNRZE DOOHH BIMHT GOMES GVOAC CPGCO GAGWO SKNSH REOXI WXCDE ASZHK NIPSR

NSPBN MYASB KSOQJ SXCOC DAZAB IHCXS RAKSI IOLSS NTRER CXCVU HWYND HPHUI GIPFO

MECWA YROCA CCEVY GSVAD ESHYT RETLD IXCIA KLKGP GIEVE EVKND BXFNS DHPBD ODHTA

YAGII VGHSC WHREI SWOBE NNTKJ EKLPB NTREG SKROF XJORO CDUXI CESGZ EMITG POERD

TGHSC WOBEM UGFON DLNZS SDESO CVELC SBALL TOXDY NTCPW RIRVS SXEPF DHBEP HONOD

PZVSZ ERWOS RAKSL EONCS QADIK SVYKF USMTO DQMRI CTDFS CNEUC BECTP HSOXB JHDHO

IGFOM KICWX GRAQW DADIH KOLVP GCDEM TTRSN VAGUO FYRTG DROST FFECA CRXAD IDBKL

ZAGYC ADPGS CEXTI VOGBE PHOSD TWFOA DTDHR ESRHI BVSVP ZSSZR TRKTS OCPII XVPGS

VOMPA WAVIP BZROD PHYRC

Breaking the Vigenère Cipher

If the key word has length P, then every Pth letter is encrypted using the same shift cipher.

Extracting every Pth letter should produce a distribution that is close to the English alphabet.

We can detect this by calculating the index of coincidence.

Breaking the Vigenère Cipher

Once we have found the key length P, we need to find the key word.

This means we need to find the correct shift for the 0th, Pth, $2P$th letters, then the 1st, $(P+1)$th, $(2P+1)$th letters, and so on.

To decide if we have the correct shift, we can compare the distribution of the letters with the expected distribution of the English alphabet, using the chi-squared statistic.

$$\chi^2 = \frac{(\#A - \text{Expected } \#\ A)^2}{\text{Expected } \#\ A} + \cdots + \frac{(\#Z - \text{Expected } \#\ Z)^2}{\text{Expected } \#\ Z}$$

If we have correctly guessed the shift, this value will be low.

Breaking the Vigenère Cipher

```
def ChiSquared(Text, Period, StartingPosition):
    ExtractedText = PeriodicTexts(Text, Period)[StartingPosition]
    N = len(ExtractedText)
    Counts = CharacterCount(ExtractedText)
    ChiValues = []
    for Shift in range(26):
        Chi = 0
        for i in range(26):
            ShiftedFreq = EnglishFreqs[(i - Shift) % 26]
            Chi = Chi + (Counts[i] - ShiftedFreq * N)^2 / (ShiftedFreq * N)
        ChiValues.append(Chi)
    return(bar_chart(ChiValues))
```

Thanks for listening!

Code used in this demonstration can be found here:

`https://github.com/dillon128/Vigenere`

You can implement SageMath code online at:

`https://sagecell.sagemath.org/`