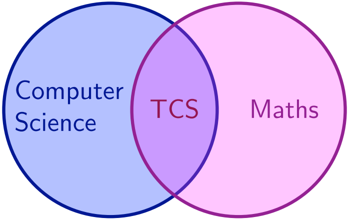
Breaking ciphers with computer science and statistics

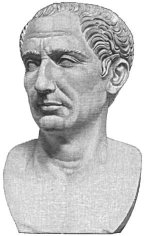
Dillon Mayhew

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:

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

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.

The key (the secret information that we need to decrypt the message) is the number of places we have shifted.

Question

How easy is it to break a shift cipher?

That is, how easy is it to decrypt a message even if we don't know that key?

Shift ciphers

Question

Can you break 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

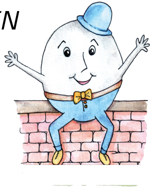
Shift ciphers

Question

Can you break 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

Substitution ciphers are a bit more complex that shift ciphers. We replace each letter by another, not necessarily by shifting.

$$\begin{array}{ccccccccccccc} A & B & C & D & E & F & G & H & I & J & K & L & M \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ D & S & L & C & G & V & Y & O & Z & P & J & F & N \end{array}$$

$$\begin{array}{ccccccccccccc} N & O & P & Q & R & S & T & U & V & W & X & Y & Z \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ R & X & U & T & A & B & E & K & Q & W & I & H & M \end{array}$$

$ABBEY\ ROAD$ is encrypted as $DSSGH\ AXDC$.

$FGE\ ZE\ SG$ is decrypted as $LET\ IT\ BE$.

Substitution ciphers

Question

Can you break this substitution cipher?
(The encrypted message is a nursery rhyme.)

CDPW X CUPW UB CDYNVPAV X NUAMVQ BTRR UB LZV
BUTL XPJ QSVPQZ GRXAMGDLJC GXMVJ DP X NDV
SEVP QEV NDV SXC UNVPVJ QEV GDLJC GVWXP QU CDPW
SXCPQ QEXQ X JXDPQZ JDCE QU CVQ GVBULV QEV MDPW

Substitution ciphers

Question

Can you break this substitution cipher?
(The encrypted message is a nursery rhyme.)

CDPW X CUPW UB CDYNVPAV X NUAMVQ BTRR UB LZV
BUTL XPJ QSVPQZ GRXAMGDLJC GXMVJ DP X NDV
SEVP QEV NDV SXC UNVPVJ QEV GDLJC GVWXP QU CDPW
SXCPQ QEXQ X JXDPQZ JDCE QU CVQ GVBULV QEV MDPW

SING A SONG OF SIXPENCE A POCKET FULL OF RYE
FOUR AND TWENTY BLACKBIRDS BAKED IN A PIE
WHEN THE PIE WAS OPENED THE BIRDS BEGAN TO SING
WASNT THAT A DAINTY DISH TO SET BEFORE THE KING

Substitution ciphers

Question

How could we make a substitution cipher more secure?

Substitution ciphers

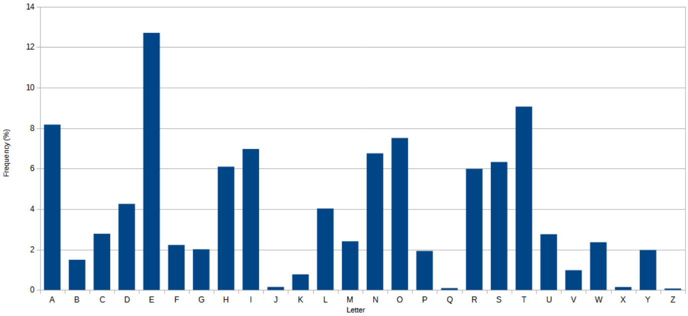
Question

How could we make a substitution cipher more secure?

- Remove spaces
- Insert extra letters to disguise words (salting the message)
- Replace common words with symbols

Substitution ciphers

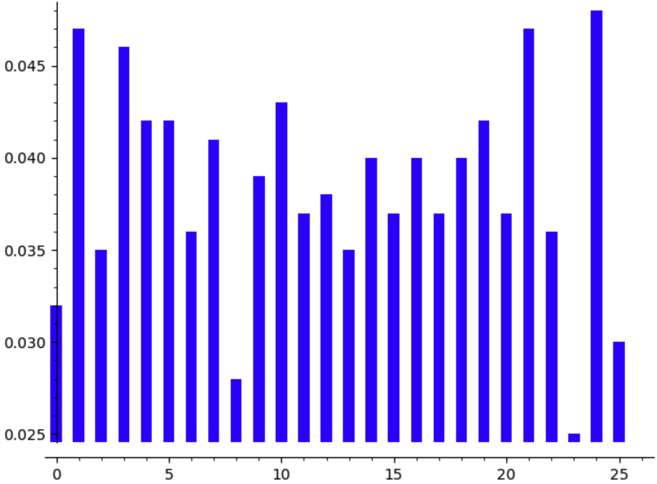
Even with improvements, substitution ciphers are easy to break because the letters in English follow a distinctive pattern of frequencies.



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

Substitution ciphers

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



Index of coincidence

We can measure how close a text is to uniformly random 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).

Index of coincidence

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

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

The incidence of coincidence is much higher for text in English.

```
IndexOfCoincidence(RandomText(1000))
```

```
0.03826
```

```
IndexOfCoincidence(TestText)
```

```
0.06466
```

Vigenère cipher

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

Vigenère cipher

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

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?

WIFBQ GYVAV WEFGU IEZIV XCFBW SYBNV WECIU IFBKW HIAOR GIZIT XLLWP IHRPW BAAXC HTNAC CAPIF TMVKF XSPQR AIAMK IAAIN NSRAC CDVVV TRCZG ISRDK SEAKG IOPWP HTECE INNZT PTVDG HAOWW IWUIV WACXG CEQIP SEKXN PIAEJ NAALJ DWVBJ PPCMP TDFWO TTUMQ GIFBU RAGMI DRVHG WIFBQ GYNAC HOPQC ASPQG CCREJ XLRWV WEEAU TEVBC HPNZV DFGPG WUZIP XTVMU DRPWP HIQMT XTNPA QRVLF XSPQR AIAMU XMVTC GDRJC IEFAW GRBCP STUMR JRCWU TOSPK HTBZA UOEMZ PMCTG LHRBJ TRVBU BAVVC XMVAV WEBZG IIPIN IOHVE DVRZV WEGZW IHBZR GAPBK RAYBQ AENZP AEFAQ CSSZQ BTUMR PSGQP PMBZG VEAMT PLFMP HEGPG IEEUJ XSGWT NRRNG GSAWV IONVC RAQMO XCSQG ADOCV IOGPG EAFBK ISRTH IIZMU XNGPG EAFBQ GTBQP SIIQF JAYBG MTFID DUGBJ TPNAV

Breaking the Vigenère Cipher

Question

The idea is that the Vigenère cipher is much more secure because every letter is encrypted with a different shift cipher. But is this really true?

Breaking the Vigenère Cipher

Question

The idea is that the Vigenère cipher is much more secure because every letter is encrypted with a different shift cipher. But is this really true?

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

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

We can detect this by calculating the index of coincidence.

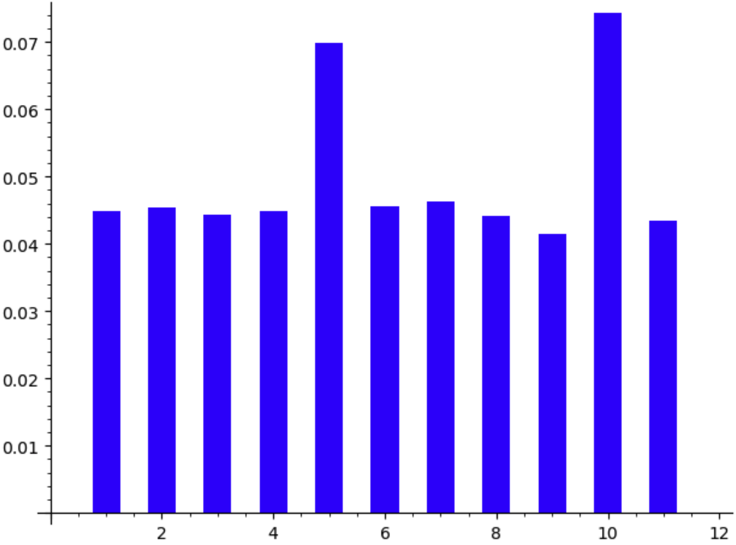
We will test every possible period between 1 and 12, and look for the largest index of coincidence.

Breaking the Vigenère Cipher

```
def PeriodicTexts(Text, Period):
    PeriodicTexts = [""] * Period
    NewText = RemoveSpaces(CleanText(Text))
    for i in range(len(NewText)):
        PeriodicTexts[i % Period] = PeriodicTexts[i % Period] + NewText[i]
    return(PeriodicTexts)
```

```
MeanIndices = [0]
for Period in range(1,12):
    PTexts = PeriodicTexts(CryptText, Period)
    Indices = []
    for i in range(Period):
        Indices.append(IndexOfCoincidence(PTexts[i]))
    MeanIndices.append(mean(Indices))
bar_chart(MeanIndices)
```

Breaking the Vigenère Cipher



Breaking the Vigenère Cipher

Question

We get much larger indices of coincidence when we sample every 5th letter or every 10th letter. Why do we get two 'spikes'?

Breaking the Vigenère Cipher

Question

We get much larger indices of coincidence when we sample every 5th letter or every 10th letter. Why do we get two 'spikes'?

If every 5th letter is encrypted using the same shift, then so is every 10th letter.

So we can feel safe in assuming that the keyword has length 5.

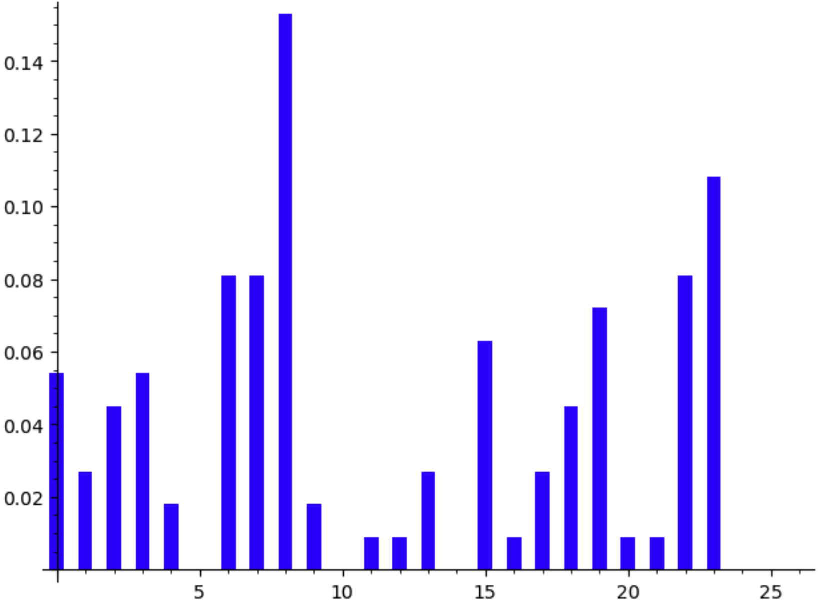
Breaking the Vigenère Cipher

We think the keyword has length 5. Now we need to find the keyword.

The 0th, 4th, 9th, 14th letters are all shifted by the same amount, corresponding to the first letter of the key word.

Let's look at the frequency distributions of these letters.

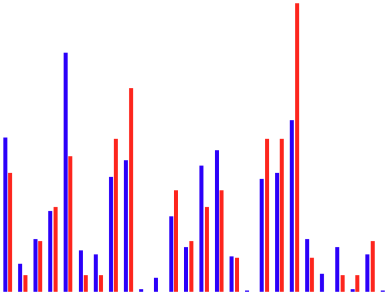
Breaking the Vigenère Cipher



Breaking the Vigenère Cipher

The big spike might be the letter T. This suggests the 0th, 4th, 9th, 14th letters have been shifted 15 spaces.

```
CompareFreqs(EnglishFreqs, Frequencies(ShiftEncryption(PTexts[0],-15)))
```



The blue chart shows expected English frequencies, and the red chart shows the 0th, 4th, 9th, 14th letters with a shift of 15 reversed.

Breaking the Vigenère Cipher

Now we think the keyword starts with the letter in position 15, which is P.

We could use a more statistical method: 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.

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

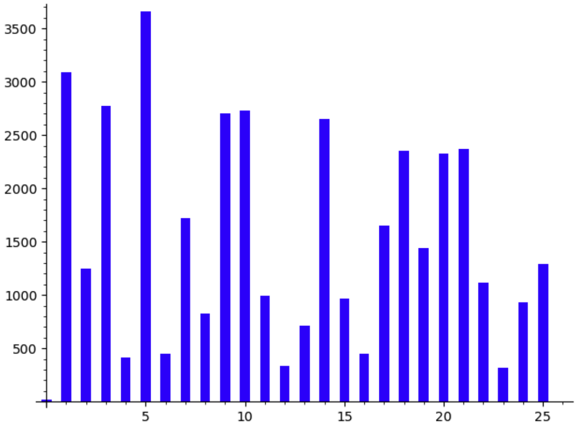
If we have correctly guessed the shift, this value will be low.

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))
```

Breaking the Vigenère Cipher

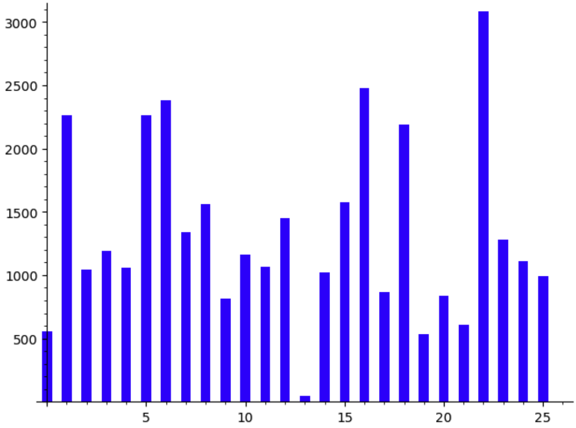
```
ChiSquared(CryptText, 5, 1)
```



A very low χ^2 with a shift of 0 suggests the second letter of the keyword is A.

Breaking the Vigenère Cipher

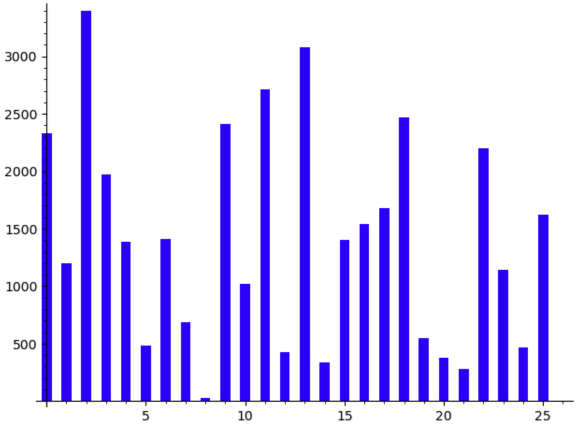
```
ChiSquared(CryptText, 5, 2)
```



A very low χ^2 with a shift of 13 suggests the third letter of the keyword is N.

Breaking the Vigenère Cipher

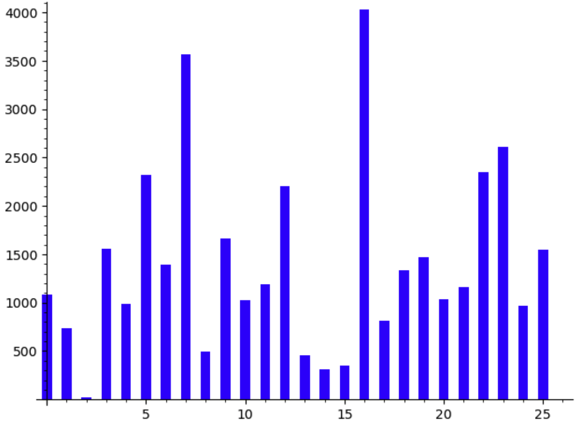
```
ChiSquared(CryptText, 5, 3)
```



A very low χ^2 with a shift of 8 suggests the fourth letter of the keyword is *I*.

Breaking the Vigenère Cipher

```
ChiSquared(CryptText, 5, 4)
```



A very low χ^2 with a shift of 2 suggests the fifth letter of the keyword is C.

Breaking the Vigenère Cipher

Now we think the keyword might be *PANIC*.

Let's try it!

```
RemoveSpaces(VigenereDecryption(CryptText,'PANIC'))
```

Breaking the Vigenère Cipher

HISTORYISTHESYSTEMATICSTUDYOFTHEPASTFOCUSINGPRIMARILYONTHEHUMANPAST

ASANACADEMICDISCIPLINEITANALYSESANDINTERPRETSEVIDENCETOCONSTRUCT

NARRATIVESABOUTWHATHAPPENEDANDEXPLAINWHYANDHOWITHAPPENEDSOME

THEORISTSCATEGORIZEHISTORYASASOCIALSCIENCEWHILEOTHERSSEEITASPART

OFTHEHUMANITIESORCONSIDERITAHYBRIDDISCIPLINESIMILARDEBATESSURROUND

THEPURPOSEOFHISTORYFOREXAMPLEWHETHERITSMAINAIMISTHEORETICALTO

UNCOVERTHETRUTHORPRACTICALTOLEARNLESSONSFROMTHEPASTINAMORE

GENERALSENSETHETERMHISTORYREFERSNOTTOANACADEMICFIELDBUTTOTHE

PASTITSELFTIMESINTHEPASTORTOINDIVIDUALTEXTSABOUTTHEPAST

Thanks for listening!

Contact me at `d.mayhew@leeds.ac.uk`

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/`