BOSS Technical Note 007

BREAKING A PORTA CIPHER WITH STATISTICS

Breaking a Porta cipher with statistics

We are going to show you how to break a ciphertext that was encrypted with a Porta cipher by using statistics alone. No guessing, no trial-and-error, just straightforward analysis. Our target audience consists of the students competing in the British National Cipher Challenge. But first, what is a Porta cipher?

The Porta cipher

The Porta cipher was an invention of Giovan Battista Bellaso, which clearly explains how it got its name. It is a periodic polyalphabetic substitution cipher. This means that it takes a collection of substitution ciphers and applies them one at a time to each letter of the plaintext. When it reaches the end of the collection, it cycles through them again as many times as necessary to encrypt the entire text. The simple substitution ciphers are called monoalphabetic, since they use one mixed alphabet and to distinguish them from the polyalphabetic Porta. There are thirteen monoalphabetic substitution ciphers to choose from, and our choices are specified by a keyword, according to this table below. There are two conventions for how keyword letters correspond to the thirteen substitutions. Version “2” is the one used in Britain; the other version is more logical, so it is used in America.

key (version)		plaintext alphabet
1	2	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
A/B	A/B	N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
C/D	Y/Z	O P Q R S T U V W X Y Z N M A B C D E F G H I J K L
E/F	W/X	P Q R S T U V W X Y Z N O L M A B C D E F G H I J K
G/H	U/V	Q R S T U V W X Y Z N O P K L M A B C D E F G H I J
I/J	S/T	R S T U V W X Y Z N O P Q J K L M A B C D E F G H I
K/L	Q/R	S T U V W X Y Z N O P Q R I J K L M A B C D E F G H
M/N	O/P	T U V W X Y Z N O P Q R S H I J K L M A B C D E F G
O/P	M/N	U V W X Y Z N O P Q R S T G H I J K L M A B C D E F
Q/R	K/L	V W X Y Z N O P Q R S T U F G H I J K L M A B C D E
S/T	I/J	W X Y Z N O P Q R S T U V E F G H I J K L M A B C D
U/V	G/H	X Y Z N O P Q R S T U V W D E F G H I J K L M A B C
W/X	E/F	Y Z N O P Q R S T U V W X C D E F G H I J K L M A B
Y/Z	C/D	Z N O P Q R S T U V W X Y B C D E F G H I J K L M A

Since each of the thirteen substitution ciphers that make up the Porta are their own inverses (check it and see), we can simplify the table like this:

key	A	B	C	D	E	F	G	H	I	J	K	L	M
A/B	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y/Z	O	P	Q	R	S	T	U	V	W	X	Y	Z	N
W/X	P	Q	R	S	T	U	V	W	X	Y	Z	N	O
U/V	Q	R	S	T	U	V	W	X	Y	Z	N	O	P
S/T	R	S	T	U	V	W	X	Y	Z	N	O	P	Q
Q/R	S	T	U	V	W	X	Y	Z	N	O	P	Q	R
O/P	T	U	V	W	X	Y	Z	N	O	P	Q	R	S
M/N	U	V	W	X	Y	Z	N	O	P	Q	R	S	T
K/L	V	W	X	Y	Z	N	O	P	Q	R	S	T	U
I/J	W	X	Y	Z	N	O	P	Q	R	S	T	U	V
G/H	X	Y	Z	N	O	P	Q	R	S	T	U	V	W
E/F	Y	Z	N	O	P	Q	R	S	T	U	V	W	X
C/D	Z	N	O	P	Q	R	S	T	U	V	W	X	Y

It this table, the action of a substitution is to swap letters from the top row with letters from one of the thirteen lower rows. An example will make it all much clearer.

The example that we will use throughout this document is from the National Cipher Challenge itself. Spoiler alert: We already know the solution and are going to tell you the answer now. Here is the ciphertext from the 2011 challenge number 6B:

```
CXTCVJMNXXGGBJMBASCNAEPAXRYNOJZEKSSBJZHAZHMRPUPTVHWRYHLHXZYFCKON
CUKVCRLUXIBGCIUPOOGTKOMRQJYFCWNRJALNJZWBQSAAZYUGZIGFYWLOVXGFCNIC
VWACZEMUVDUXVFZBDLLBDKORAHUPCRBVCOYFZKPFRWBVBWVYVAHEVZAPVTSYVYHZ
QMMBRVPAZSAZREXGKUKVJPPATBKEVEMYHTWGZMYCVLLBJEYYCNYFCWMRKYIYRCUC
LXUEBKHOVTLSKUSBFMMJKCHHJZIRKGSRJTTRUYONARPRREXZRLRURMYFCBTOPNXN
TLHFBKAEZHNFLWIRAMPAFQPPYNYBDKSVJXXGYNOVBAHEHFZGVSIRBKMUVFONENVR
VHMERYRRUXSRTKKBJOWNPUENJWCRYWBRUXWVLQYEVWMUVRKRQTPYBWGQKAORAYHZ
QBGVTWMVKHLGYNENLJYNAKHURCYPAWWXVWMHARGTBXGPACIGZIGNJZONEXUTKFXT
ATLCKOMUVNPFCFKVTTSEKUYBWIAEKINNJOLNCRHARHXNPWKZZHNYHQUIVSUQVWSV
JQCVCQHHASHEVIYPVHMSZEUATOUYRYMVEOMVVJMUVFXBJFMNLJYNAKHURCYHJZYE
BAHBUQHJTNUAXNLVJOGGVIGNCOHARUZVJTGPVQUIVTSGVIYQKBKNLGKBRVOGYFAT
YAORHQUIVJYEBRLGVWPAFFKXZHNGYIHHXNMHARGTBHHGVJUAUAORANPFVCPQVEWR
CNUGCQYLYTBRSNYABAAQHRGTKAORAJHHAVYFCFMEHAHGAWWXKBKNTKPIZAPRBKOR
QMYYENLGYXEFVNTGKNUIVZYGVVMRUFAEZHMEDJPBJOGGKKORZLHJJEYGFIKXREXN
AXUJRIYGYTMJVVULSXMERYRVJZMUVVCRYTBRANUFKHMBSNSVVCYGYWMPYTKYZNON
BTZNQRSLTIGAVYMVKHCVCQMUVMYPDIPGHMYEERWRBUAGJFZVASYIZZYATXMURKHH
AVAEANGGKJYERKPBJMONENVRVHWBQGKBQOLRUEYIVLMUVUYFBOMVBNLFVHMVRUMU
RACRZEWEVTLRKLKVJAYEJWSNJWYKCNKARRLGRKYBWLYNURGRBMPATWLRZAPFJNWR
BMUEHKHEVTWGXRBRJAORZVIBAAUATNHSUXMRAVPAZHNGYNZHPRXRXIYRCICUZYOB
DLHCVIUGZIGFRIYPDLKRJKSLTITCAFTVBXXBSJYEETMVKELFYIAYUXYZROGGRRGR
UUAGCQYFVMOBDUXBJREVJMHYEXHHAZYRLXLGUNYCTIBRAWNRJALUKFRRFOSYSNMN
OOGTTFGGAISBWJHHCNYEJIYTZIGBLNKNCOHABWGQTXGGAWSODLLNACCVPRIEKMPQ
VTSYJNWRBMUEHIYFKBKPVJGBRVMVKELUKBSQSNMNOXGHJKPYCNYSDUSYVCYYKONB
EXKAQNGGZHMRANLGZMRAKAG
```

The key is SONATINA, and here is the plaintext:

```
TEMPESTAGENTSSTORMWARNINGLEAKSFROMLONDONINTELLIGENCEHQSUGGESTTHA
TBRITISHGOVTTRACKINGOFTEMPESTAGENTSANDCOMMUNICATIONSHASBEENSTEPP
EDUPINTHEWAKEOFOURSOUTHERNACTIVITIESITISADVISABLETOREDUCEALLECOM
MSTOAMINIMUMANDTOBRINGINCURRENTLYACTIVEPERSONNELTHESTATEOFPLAYAP
PEARSTOBEASFOLLOWSTWOYOUNGPEOPLENAMEDCHARLIEANDMARKHAVESTUMBLEDA
CROSSTURINGSPAPERSINWHICHHEOUTLINEDTHEHISTORYOFTEMPESTTHEYHAVEBE
ENTRACKEDELECTRONICALLYANDWEHAVEDECIPHEREDTHEIREMAILSANDOTHERCOM
MUNICATIONSTHEYAPPEARTOHAVECRACKEDTURINGSENCRYPTIONANDHAVEAGOODG
RASPOFTHEHISTORICALROLEOFOURORGANISATIONANDALARMINGLYHAVEMADEALI
NKWITHOURMORERECENTFINANCIALACTIVITIESTHEYDONOTAPPEARTOHAVEUNDER
STOODHOWCHANGESININTERNATIONALFINANCEHAVEALTEREDOURAPPROACHTHOUG
HTHEYHAVEPERSISTEDINWORKINGTHROUGHTURINGSNOTESANDTHEREISEVIDENCE
THATTHEYHAVEBEENSTUDYINGOTHERSOURCESTOTRYTOTRACKOURACTIVITIESTHE
MSELVESTHEYSEEMTOHAVEDETECTEDOURINTRUSIONINTOTHEIROWNNETWORKANDA
REAWARETHATWEMAYBETRACKINGTHEMWEHAVEREASONTOBELIEVETHATCHARLIEHA
SAFAMILYCONNECTIONWITHTHESECURITYSERVICESBUTNOFIRMEVIDENCETHATOU
RCURRENTOPERATIONSHAVEBEENCOMPROMISEDNEVERTHELESSITISESSENTIALTH
ATWEINCREASEOURINTERNALANDEXTERNALSTATEOFREADINESSINCASEITISNECE
SSARYTOREACTGIVENTHEIMPORTANCEOFDETERMININGTHEFULLDEGREETOWHICHO
UROPERATIONSARECURRENTLYCOMPROMISEDOBSERVATIONSSHOULDBEMAINTAINE
DBUTTHESESHOULDONLYINVOLVEOURDEEPESTDEEPCOVERAGENTSHOOKEWILLBETA
KINGCONTROLOFSOUTHERNREGIONOPERATIONSANDCENTRALBURSARYWILLPROVID
EALLNECESSARYRESOURCESNOACTIONSHOULDBETAKENUNTILTHEFULLLEVELOFGO
VERNMENTINTERESTISKNOWN
```

Now, let’s see how encryption works (decryption is exactly the same). Put the plaintext on one line, and repeat the keyword under it as many times as necessary:

```
TEMPESTAGENTSSTORMWARNINGLEAKSFROMLONDONINTELLIGENCEHQ...
SONATINASONATINASONATINASONATINASONATINASONATINASONATI...
```

For each letter in the plaintext, find the row in the table for the key letter under it, and compare to the top row. The ciphertext letter is the letter in the same column as the plaintext letter. For the first letter (T), the key letter is S, so the ciphertext letter is C:

```
       A B C D E F G H I J K L M
**S**/T  R S T U V W X Y Z N O P Q
```

For the second letter (E), the key letter is O, and so the ciphertext letter is X:

```
       A B C D E F G H I J K L M
**O**/P  T U V W X Y Z N O P Q R S
```

The process continues until the text is completely encrypted.

```
TEMPESTAGENTSSTORMWARNINGLEAKSFROMLONDONINTELLIGENCEHQ...
SONATINASONATINASONATINASONATINASONATINASONATINASONATI...
CXTCVJMNXXGGBJMBASCNAEPAXRYNOJZEKSSBJZHAZHMRPUPTVHWRYH...
```

Now that we know how the cipher works, let’s pretend that we do not know the solution, and look at the methods for finding it. The first step is to find the period, by which we mean the length of the keyword. To do so, we are going to use a statistic called the index of coincidence.

Index of coincidence

Human languages are notoriously repetitive. Very repetitive. Looking at the repetitiveness of a text, compared to random nonsense, can give us a handle on determining whether we are dealing with a natural language. Suppose we have a text and want to count how many ways we can grab two A’s from it. Well, if there are n_{A} of them in the text, then there are n_{A} ways to pick one. That leaves $n_{\mathrm{A}} - 1$ ways to select another one. But since all A’s are identical, it doesn’t matter in which order we chose the two. So, to avoid double-counting, we have to divide by two. The result is that the number of ways to choose two A’s from the text is

$$\frac{n_{\mathrm{A}}(n_{\mathrm{A}}-1)}{2}$$

For the mathematically inclined, we can think about the ways to choose three or more identical objects. For three, the result is

$$\frac{n(n-1)(n-2)}{3\cdot 2}$$

See a pattern yet? For selecting *k* out of *n*, the result is

$$\frac{n(n-1)(n-2)\cdots(n-k+1)}{k\cdot(k-1)\cdots 3\cdot 2\cdot 1}$$

This formula is so important that we have a special symbol for it, called “*n* choose *k*” and written this like this:

$$\binom{n}{k}=\frac{n!}{(n-k)!\,k!}$$

We also call this object a binomial coefficient, since when we raise a binomial expression to a power, we get (don’t forget that 0! = 1)

$$(x+y)^n = \binom{n}{0} x^n y^0 + \binom{n}{1} x^{n-1} y^1 + \binom{n}{2} x^{n-2} y^2 + \cdots + \binom{n}{n} x^0 y^n$$

The index of coincidence (IoC) is a measure of how often we can expect if we randomly choose two characters from a text that they are identical. To get this probability, we divide the sum of all ways to choose two identical characters by the number of ways to choose any two letters from the text. If the numbers of each letter present are n_A, ..., n_Z, and there are N total letters in the text,

$$\text{IoC} \;=\; \frac{\sum_{i=A}^{Z} \binom{n_i}{2}}{\binom{N}{2}}$$

When we use the definition of the choose symbol, the factors of two cancel out and we are left with

$$\text{IoC} \;=\; \sum_{i=A}^{Z} \frac{n_i(n_i-1)}{N(N-1)}$$

For a random string of letters, the IoC is about 1/26 = 0. 385. For typical English text, it is about 0.0673 (75% larger than for random).

Finding the key length with the index of coincidence

This technique was probably invented by Sinkov. It relies on the idea that encryption with a monoalphabetic substitution cipher does not change the index of coincidence. So if we take every n^{th} letter of a ciphertext that was encrypted with a Porta cipher, and if the length of the keyword is the same n, then the letters we took were all encrypted with the same substitution, and therefore the IoC of this set of letters should look like English. The technique works like this: We take a ciphertext and write it into n columns. Then we find the IoC of each column and average the results. If the average is close to the IoC of English, then we believe that we have found the correct key length.

We return to our example. We are going to try key lengths starting from one and going until we are confident that we have found the correct one. Writing the ciphertext in one column looks like this:

```
C
X
T
C
V
J
M
N
X
X
⋮
```

Writing the whole thing out would be a waste of paper, and this document is already enough of a waste. If we calculate the index of coincidence for this one column, we get 0.0430, and if we average that we get 0.0430. Let's build a table of our results:

columns	indices of coincidence	average
1	0.0430	0.0430

Now we try two columns:

```
C    X
T    C
V    J
M    N
X    X
G    G
B    J
M    B
A    S
C    N
⋮    ⋮
```

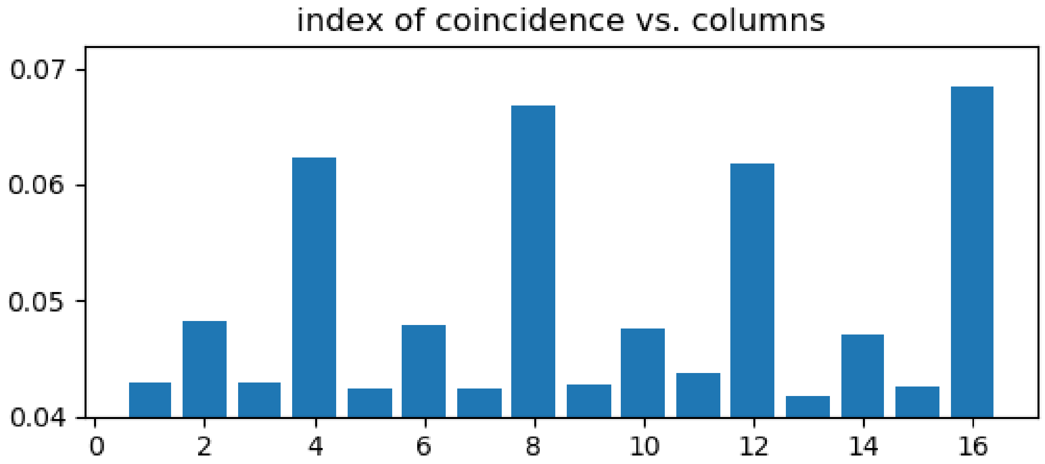
The IoC of the first is 0.0481, and of the second 0.0483. Their average is 0.482.

columns	indices of coincidence	average
1	0.0430	0.0430
2	0.0481 0.0483	0.0482

When we get to nine columns, our table looks like this:

columns	indices of coincidence	average
1	0.0430	0.0430
2	0.0481 0.0483	0.0482
3	0.0423 0.0426 0.0438	0.0429
4	0.0680 0.0470 0.0665 0.0681	0.0624
5	0.0441 0.0429 0.0417 0.0395 0.0437	0.0424
6	0.0461 0.0447 0.0503 0.0524 0.0466 0.0473	0.0479
7	0.0420 0.0417 0.0435 0.0416 0.0445 0.0416 0.0423	0.0425
8	0.0712 0.0625 0.0659 0.0714 0.0649 0.0700 0.0647 0.0639	0.0668
9	0.0435 0.0418 0.0456 0.0429 0.0432 0.0425 0.0389 0.0410 0.0456	0.0427

Four looks tempting, but eight is better, and very close to the expected value for English. Here is a graph for you. We went to sixteen columns, so that you can see the repeating pattern.



While in this case it is a close call between key lengths four and eight, we will pick eight. If it turns out that the key length is really four, then in the end we will have a keyword that repeats the same four letters. But if the key length is truly eight, but we choose four, then in the end we will get garbled nonsense for a plaintext. With a long ciphertext like the one in NCC 2011 6B, where we have a lot of letters with which to do statistics, it is safer to choose the longer key length. (You might be wondering why we don't take sixteen; it would work, and the keyword would be two copies of the same eight-letter word.)

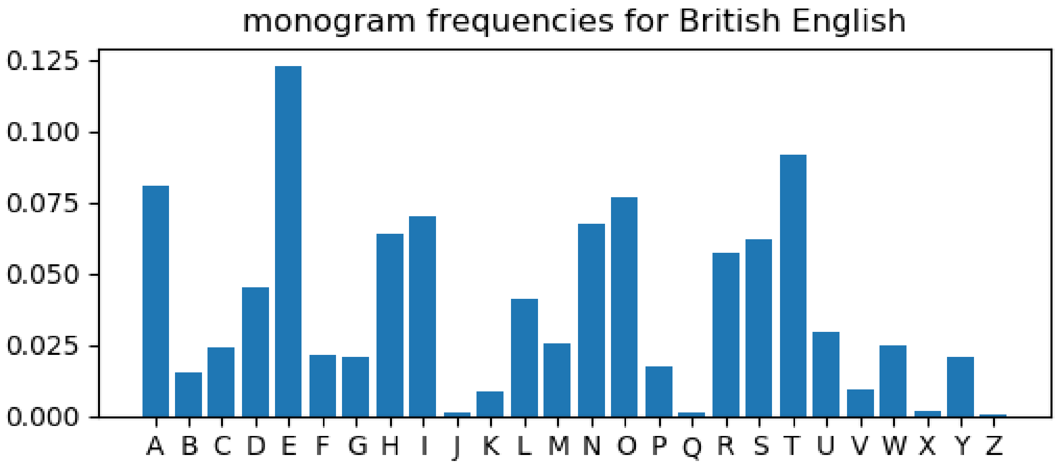
Finding the keyword by matching monogram frequencies

“Monogram” is just a fancy word for “single-letter”. We took a dozen or so British novels by Charles Dickens, Douglas Adams, and some other British writers and compiled a list of monogram frequencies for British English (also known as English English). Here they are; feel free to use them if you don't want to read any Dickens yourself.

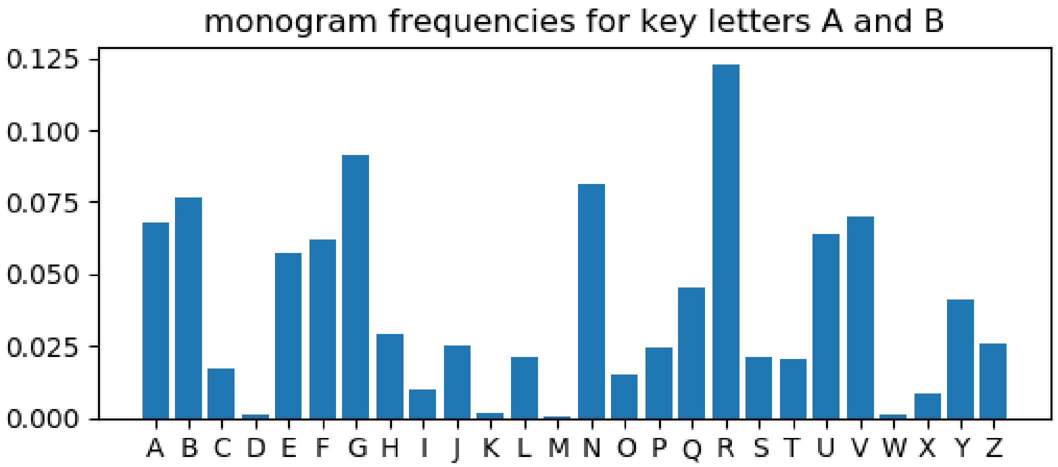
A	0.081	N	0.068
B	0.015	O	0.077
C	0.024	P	0.017
D	0.045	Q	0.00094
E	0.12	R	0.057
F	0.021	S	0.062
G	0.021	T	0.091
H	0.064	U	0.029
I	0.070	V	0.0094
J	0.0013	W	0.025
K	0.0086	X	0.0016

L	0.041	Y	0.021
M	0.026	Z	0.00064

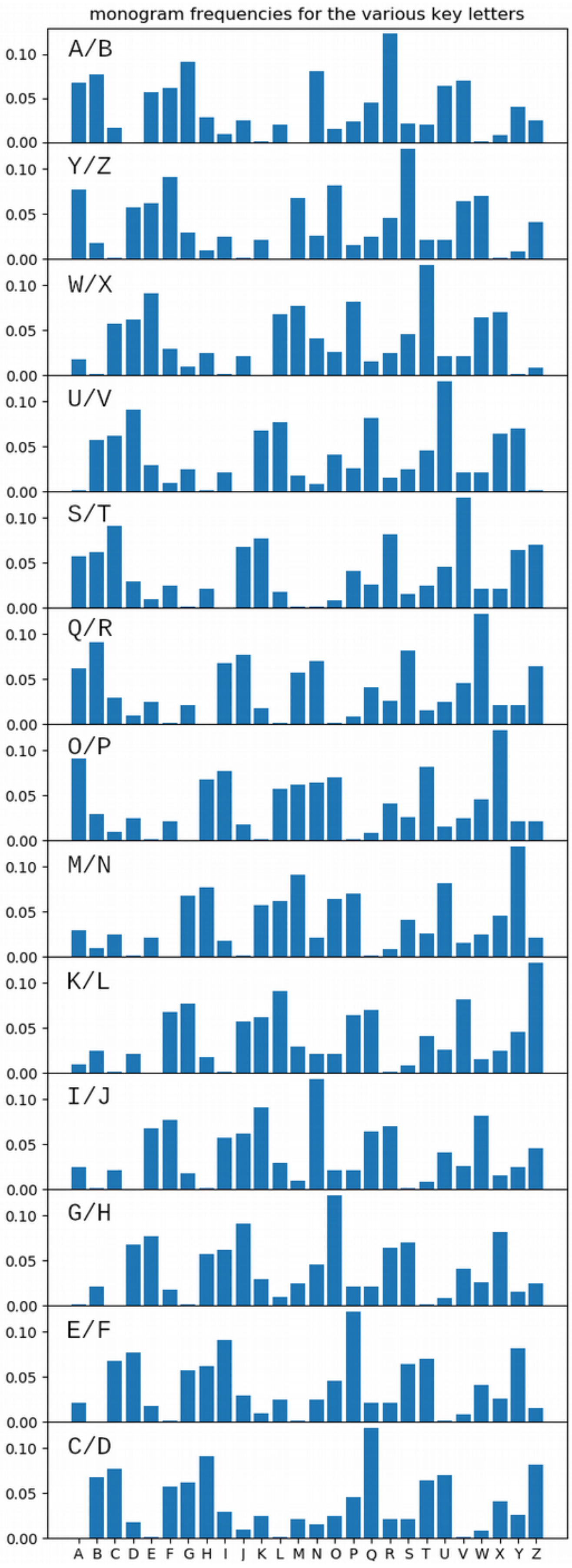
Notice that Z is pronounced “zee”. For your added enjoyment, here they are in graphical form:



For key letters A and B, letters A and N are swapped, as are B and O, C and P, etc. (Do you recognize ROT13?). The resulting frequencies are then



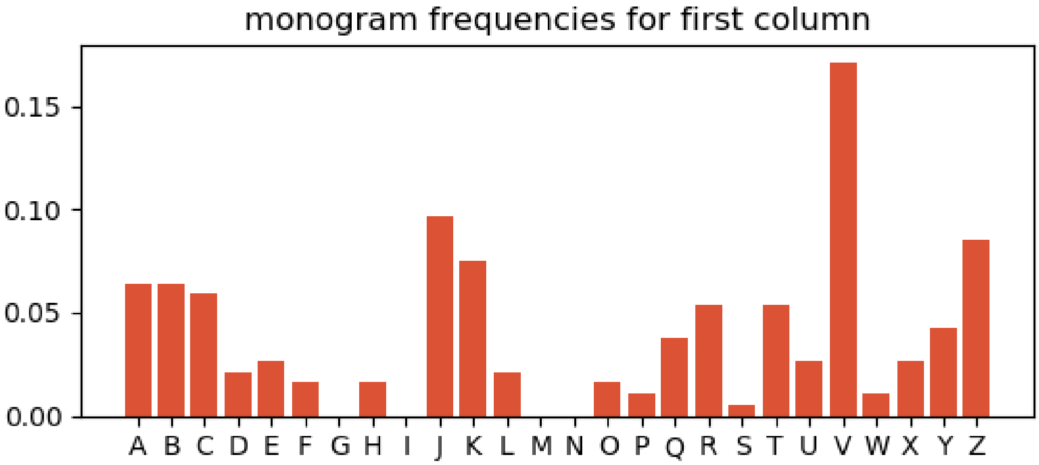
We need to do this for each possible substitution used in the Porta cipher. There are thirteen graphs, which are combined below:


monogram frequencies for the various key letters
A/B
Y/Z
W/X
U/V
S/T
Q/R
O/P
M/N
K/L
I/J
G/H
E/F
C/D
0.10
0.05
0.00
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

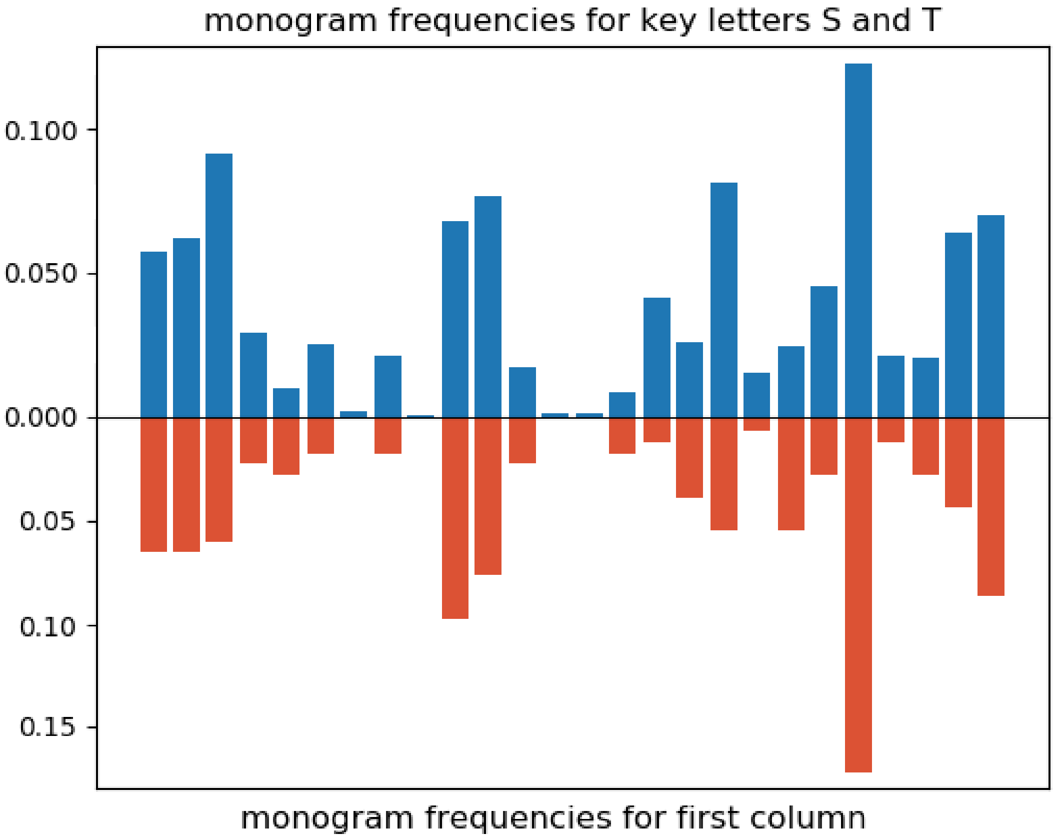
Remember that we decided that the key length for our example was eight. Write our ciphertext into eight columns:

```
C X T C V J M N
X X G G B J M B
A S C N A E P A
X R Y N O J Z E
K S S B J Z H A
Z H M R P U P T
V H W R Y H L H
X Z Y F C K O N
C U K V C R L U
X I B G C I U P
⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮
```

For each column, we are going to look at the frequencies of the letters and compare the graph to the graphs that we made for the thirteen possible substitution ciphers. Here are the frequencies for the first column:



If we compare this graph to the thirteen earlier ones, we get the best match for key letters S and T:



We have to do this for each of the eight columns. When we have done so, we find that the key letters are `S/T`, `O/P`, `M/N`, `A/B`, `S/T`, `I/J`, `M/N`, and `A/B`. And now you can see why a key length of four was so seductive: the key is *almost* a repeated sequence of four.

But hold on a minute! Didn't we say that we were going to use *statistics* to solve the cipher? In that case, we should find a way to measure the quality of a match between two frequency distributions. Some of you who know a little about statistics may want to use the χ^2 statistic. But a little knowledge is a dangerous thing. The χ^2 statistic divides each of its terms by the expected value, and for rare letters like `Q` and `Z` this can cause some terms to get overly large and possibly deceive us. So we are going to suggest a different method, based on measuring the closeness of two vectors in 26-dimensional space. It works like this: A vector is an ordered list of numbers $\mathbf{V} = (V_1, V_2, V_3, ..., V_{26})$. The numbers V_1, V_2, ... are called its components. From two vectors we can make a scalar (just a number) by adding up the products of components. Because the result is just a number, it is called the scalar product, the inner product, or the dot product (because we write it with a big black dot). So for two vectors **U** and **V**,

$$\mathbf{U} \cdot \mathbf{V} \ = \ \sum_i U_i V_i = U_1 V_1 + U_2 V_2 + ... + U_{26} V_{26}$$

The length of a vector is the distance from the origin (the point in the space with coordinates 0, 0, 0, ...) to the point whose coordinates are the components of the vector. Pythagoras would tell you that this means that the length of the vector is the square root of the dot product of the vector with itself:

$$\|\mathbf{V}\| = \sqrt{\mathbf{V}\cdot\mathbf{V}}$$

Without proving it, we are telling you that the cosine of the angle between two vectors is the normalized inner product between them.

$$\cos\theta = \frac{\mathbf{U}\cdot\mathbf{V}}{\sqrt{(\mathbf{U}\cdot\mathbf{U})(\mathbf{V}\cdot\mathbf{V})}}$$

Its value varies from −1 (antiparallel vectors) to +1 (parallel vectors). A large cosine means that two vectors point in directions that are close together, and this is the measure of closeness that we prefer to use in breaking ciphers. In breaking a Porta cipher, we can drop the denominator and just find the dot product. The best match between two frequency distributions will have the largest dot product for us, since our thirteen distributions are merely shuffled versions of the same distribution, so the denominator always has the same value. Furthermore, we can just use the letter counts for each of the columns of the text, and don't have to find actual frequencies.

Let's see how this works for the first column of our ciphertext. If we count the letters we get one vector:

$$\mathbf{U} = (12, 12, 11, 4, 5, 3, 0, 3, 0, 18, 14, 4, 0, 0, 3, 2, 7, 10, 1, 10, 5, 32, 2, 5, 8, 16)$$

Remember our English monogram frequencies? They can be a vector, too:

$$\begin{aligned}\mathbf{E} = (&0.081, 0.015, 0.024, 0.045, 0.12, 0.021, 0.021, 0.064, 0.07,\\ &0.0013, 0.0086, 0.041, 0.026, 0.068, 0.077, 0.017, 0.00094,\\ &0.057, 0.062, 0.091, 0.029, 0.0094, 0.025, 0.0016, 0.021, 0.00064)\end{aligned}$$

For key letter A and B, we had to swap the frequencies for A and N, B and O, etc. This is the vector we get:

$$\begin{aligned}\mathbf{V}_{A/B} = (&0.068, 0.077, 0.017, 0.00094, 0.057, 0.062, 0.091, 0.029,\\ &0.0094, 0.025, 0.0016, 0.021, 0.00064, 0.081, 0.015, 0.024,\\ &0.045, 0.12, 0.021, 0.021, 0.064, 0.07, 0.0013, 0.0086, 0.041, 0.026)\end{aligned}$$

The dot product of this with **U** is

$$\mathbf{U}\cdot\mathbf{V}_{A/B} = 12 \cdot 0.068 + 12 \cdot 0.077 + 11 \cdot 0.017 + \ldots = 8.23$$

If we repeat the calculation for each of the thirteen shuffled frequency distributions, we have the following, where we included the cosine for comparison.

key letter	dot product	cos θ
A/B	8.23	0.629
Y/Z	6.55	0.500
W/X	5.60	0.428
U/V	6.77	0.517
S/T	12.3	0.939
Q/R	7.85	0.599
O/P	6.02	0.460
M/N	5.33	0.407
K/L	9.30	0.710
I/J	7.16	0.547
G/H	6.83	0.521
E/F	5.22	0.399
C/D	6.70	0.512

The best match is to key letter S or T. Repeating the process for the remaining columns tells us that the key letters are again S/T, O/P, M/N, A/B, S/T, I/J, M/N, and A/B.

The last step is to untangle the keyword from the possible letters. I suppose the keyword could be the Spanish girl's name TOMASINA, but the theme for the 2011 NCC was a musical one, so we take SONATINA:

S	**O**	M	**A**	S	**I**	M	**A**
T	P	**N**	B	**T**	J	**N**	B

The last section

Now what? Now that you understand the attack, what should you do? It's a long and tedious procedure, so you should get a computer to do it for you. You will be amazed at how fast it can be. We implemented the attack in Python, which is the slowest computer language known to humans, and breaking the NCC 2011 6B ciphertext took 0.42 seconds. In C language, which is one of the fastest, it took 0.02 seconds. That's 1/50 of a second. Wowsers.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; pages 119-121

American Cryptogram Association, “The ACA and You,” www.cryptogram.org/cdb/aca.info/aca.and.you/aca.and.you.pdf; 2005 version: web.archive.org/web/*/http://www.cryptogram.org/cdb/aca.info/aca.and.you/aca.and.you.pdf; 2016 version: web.archive.org/web/*/http://cryptogram.org/docs/acayou16.pdf; the page describing the Porta cipher (American version) is www.cryptogram.org/downloads/aca.info/ciphers/Porta.pdf

Giambattista della Porta [Giovanni Battista della Porta] [Ioan. Baptista Porta], *De Furtivis Literarum Notis*, Naples [Neapoli]: Ioa. Maria Scotus, 1563, HDL: 2027/gri.ark:/13960/t37142x6g, book 2 chapter XVI

William F. Friedman, The Index of Coincidence and Its Applications in Cryptography, Riverbank Laboratories Department of Ciphers Publication 22, Geneva, Illinois, 1920, www.marshallfoundation.org/library/methods-solution-ciphers

William F. Friedman and Lambros D. Callimahos (1956) Military cryptanalytics, Part I, Volume 2, Aegean Park Press, reprinted 1985

Marjorie Mountjoy, The bar statistics, NSA Technical Journal VII (2, 4), 1963

James Lyons, Practical Cryptography website,
practicalcryptography.com/ciphers/porta-cipher
practicalcryptography.com/cryptanalysis/text-characterisation/index-coincidence
practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 376-380

Abraham Sinkov, *Elementary Cryptanalysis: A Mathematical Approach*, 2nd edition, revised by Todd Feil, published by Mathematical Association of America, 2009; www.jstor.org/stable/10.4169/j.ctt19b9krf; section 3.3