Substitution cipher:

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

HELLO FRIEND

Gets ciphered to:

BUMMI JKFUNO

Ciphering: going from letters in the text to the letters in the mapping

If we see a ciphered message like “BUMMI JKFUNO”, can we figure out the original text?

Comparing possible cipher keys:

Potential key 1 (arbitrary shuffling of the letters)

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

Potential key 2 (another arbitrary shuffling of the letters)

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

Deciphering is using the letters in the mapping back to the original text.

With potential key 1 “BUMMI JKFUNO” the deciphered text is GUVVW YQPUCH

With potential key 2 “BUMMI JKFUNO” the deciphered text is RALLP CHOAMX

How can we compare keys?

$$Pr(\text{mapping} \mid \text{data: text}) = \frac{Pr(\text{text} \mid \text{mapping})\ Pr(\text{mapping})}{Pr(\text{text})}$$

text: intercepted / "deciphered" message

Likelihood = $Pr(\text{text} \mid \text{mapping})$ ← what is the prob that this is the original message

Prior: $Pr(\text{mapping})$ = constant · prior to looking at the message all potential letter mappings are equally likely.

Marginal: $Pr(\text{text})$ = constant = sum of the numerator across all possible mappings.

State space - all possible cipher mappings · 26! possible mappings. $\sim 4.03 \times 10^{26}$

Letter A can be mapped to 26 possible letters.
once A is mapped, B can be mapped to 25, then C can be mapped to one of the remaining 24 letters, etc.

posterior distribution $\Pr(\text{mapping} \mid \text{text}) = \frac{\Pr(\text{text} \mid \text{mapping}) \times \text{constant}}{\text{constant}}$

$\propto \Pr(\text{text} \mid \text{mapping})$

How do we evaluate $\Pr(\text{text} \mid \text{mapping})$?

Given a potential key, what is the probability that the "deciphered" text was the original message?

With potential key 1 "BUMMI JKFUNO" the deciphered text is GUVVW YQPUCH. ← what is the prob, this was the original message?

With potential key 2 "BUMMI JKFUNO" the deciphered text is RALLP CHOAMX ←

Our intuition tells us that "RALLP CHOAMX" is more likely the original message than "GUVVW YQPUCH", so potential key 2 is more probable than potential key 1.

One way to evaluate $Pr(\text{text} \mid \text{mapping})$ is to look at letter combinations in the deciphered text.

In English, the most common 2-letter combo is "TH"

26 letters + space = 27 characters (ignore everything else)

27 × 27 combinations:

AA
AB
AC
⋮
AY
AZ
A␣
BA
BB
BC
⋮
BZ
B␣
CA
⋮

⋮
ZX
ZY
ZZ
Z␣
␣A
␣B
⋮
␣X
␣Y
␣Z
␣␣

What is the probability of each letter combo?

We expect TH to have a very high prob.

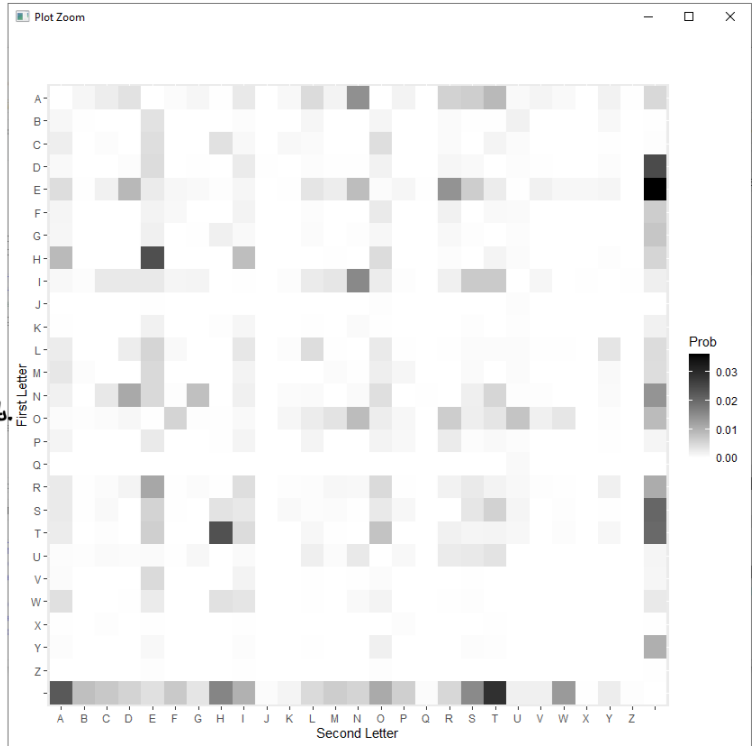
We expect other combos like "QJ" to be very improbable.

Code will read each line in a big chunk of text (War and peace) and count or tally each 2-letter combo.

"Sample text from War and peace"

_S +1
SA +1
AM +1
MP +1
PL +1
LE +1
E_ +1
_T +1
TE +1
. . . . .

Shows which letter combos are common. Power on-screen keyboards.



Neat trick: try typing the word “thing” into your phone, except, when you type the “h” and “g,” intentionally hit in the middle between h and g. It should register as an “h” when it follows the letter “t” and it should register as a “g” when it follows the letter “n.” The on-screen keyboard changes what area of the screen maps to each letter based on the letter that came before it. In the plot above, you can see t-h is much more common than t-g, and n-g is much more common than n-h.

$$Pr(\text{text} \mid \text{mapping}) = Pr(2 \text{ letter combos}) + \lambda\, Pr(\text{real English words})$$

If we see real words in the deciphered text, the will increase the probability of the potential mapping.

I have a "lexical database" - listing of words and their relative frequency in english.

"the" ~ .0528

"in" ~ .0196

"friend" ~ 7×10^{-5}

"friendship" ~ 1.7×10^{-5}

If the word does not exist, it returns null.