Problem 1

Consider the follow ciphertext in $\mathbf{Z}_{26}$, length n=362:

PPKVFAZUNGSSHUNKZLQYMUNMHFOWKIZYNAWHGSALUBHWKVVBOYTVJOHAPLJR
LGIGYTLUGBYAUAPLCRZYCQULOULMAOCSANUVLWATSWHHCLFOVAQZMIKYAFLAL
LVLXVKJBVLYVPETSQCYRWIONHBZNATZOTKJYCDNYTJLMUCGTUTKJLPXILLDTVOIU
WOIMKEMKNHLUTTPKZQIABTJYMHNIUUNGUKVONAYRVOIAQAZWOOWMACTPPET
HBOYRABAPJWTJESFIPNEVHTOYBCABSYOMNHGZBYCKLLSYPBOFICYRRVWSMFLLNCU
LVNOYLEUAWTUKLXEEPAPPEJINVYQIOTPINUHVKMEAOPEOMSMEHMWKU

You may assume our standard probability model for the original plaintext from which this ciphertext is produced, with the characters in $\mathbf{Z}_{26}$ having probability distribution given by:

A	B	C	D	E	F	G	H	I	j	K	L	M	N
0.082	0.015	0.028	0.043	0.127	0.022	0.020	0.061	0.070	0.002	0.008	0.040	0.024	0.067

O	P	Q	R	S	T	U	V	W	X	Y	Z
0.075	0.019	0.001	0.060	0.063	0.091	0.028	0.010	0.023	0.001	0.020	0.001

Moreover, you may assume that this ciphertext was encrypted from plaintext by use of a periodic "Vigenere" cipher. You should not assume that periodic function takes values in $\mathbf{Z}_{26}$ that create an English word, as is typical in "classic" Vigenere ciphers.

a) Compute the frequency of occurrence of each character in the ciphertext.

Counting every letter in the n=362 length ciphertext is a bit tedious so this will be done in MATLAB.

Refer to hw5p1sol.m and computeic.m for the complete code solution.

```
8 alphabet = {'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'};
9 counts = [alphabet cellfun(@(x) nnz(ismember(ciphertext,x)),alphabet,'un',0)]
```

```
2×26 cell array

Columns 1 through 14

  {'A' }    {'B' }    {'C' }    {'D'}    {'E' }    {'F'}    {'G'}    {'H' }    {'I' }    {'J' }    {'K' }    {'L' }    {'M' }    {'N' }
  {[24]}    {[12]}    {[12]}    {[2]}    {[12]}    {[7]}    {[8]}    {[15]}    {[15]}    {[10]}    {[16]}    {[27]}    {[15]}    {[18]}

Columns 15 through 26

  {'O' }    {'P' }    {'Q'}    {'R'}    {'S' }    {'T' }    {'U' }    {'V' }    {'W' }    {'X'}    {'Y' }    {'Z' }
  {[22]}    {[17]}    {[7]}    {[7]}    {[11]}    {[19]}    {[21]}    {[18]}    {[13]}    {[3]}    {[21]}    {[10]}
```

b) Compute the observed Index of Coincidence (IC_hat) for the ciphertext.

From Bach's notes: $\hat{IC} = \frac{\sum_i f_i(f_i-1)}{n(n-1)}$, given: f_i: occurrences of symbol i in length n text

$$\hat{IC} = \frac{f_A(f_A-1)+f_B(f_B-1)+\cdots+f_Z(f_Z-1)}{362(362-1)} = \frac{24(24-1)+12(12-1)+\cdots+10(10-1)}{362(362-1)}$$

$$\hat{IC} = 0.0435$$

c) Using IC_hat, compute the estimate of period length, of the form denoted m_E in lecture.

From lecture, this estimate was given by: $m_E = \frac{n(\kappa_s-\kappa_e)}{(n-1)\hat{IC}-n\kappa_e+\kappa_s}$

For English text, $\kappa_s = 0.066$.

$\kappa_e = \frac{1}{N} = \frac{1}{26}$

$$m_E = \frac{362(0.066-\frac{1}{26})}{(362-1)0.0435-362\cdot\frac{1}{26}+0.066} = 5.4044$$

d) We recognize that m_E is a biased estimator, and by itself may not produce a reliable estimate of the period. But suppose we instead use it to estimate a plausible range, over which we search for a more accurate estimate of period; themore accurate estimate is the quantity we denoted as m_hat in lecture. Searching over values $2 \le m \le 2xm_E$, compute the m_hat estimate of the period.

The procedure for finding this $\hat{m}$ estimate is from 13-1 of Bach's notes, and requires many IC calculations for the range $2 \le m \le \lceil 2 \cdot m_E \rceil = 12$

Friedman derived the following sharper test for period length. Let the ciphertext be

$$y_1 y_2 \cdots y_n.$$

To test the hypothesis that the period is m, compute the coincidence index separately on each group of letters that are distance m apart:

$$\begin{aligned} \mathrm{IC}_1 &= \mathrm{IC}(y_1, y_{m+1}, y_{2m+1}, \ldots) \\ \mathrm{IC}_2 &= \mathrm{IC}(y_2, y_{m+2}, y_{2m+2}, \ldots) \\ &\cdots \\ \mathrm{IC}_m &= \mathrm{IC}(y_m, y_{2m}, y_{3m}, \ldots) \end{aligned}$$

and then average these values. If we choose the m that maximizes this average, this is an estimate for the period.

UVA ECE/CS Spring 2021 Eric Bach © 2019

This will be done in MATLAB. See hw5plsol.m

```
%% 1.d: Finding a better estimate over the range 2<m<2*m_E
% The idea is for each period length m, we compute the individual IC for
% each particular character position in the period, then find the mean
% value. This gives us the IC_m for that period. Then, the most probable
% key length will have the largest magnitude IC_m, and this will be our
% m_hat.

ic(1) = IC_hat;
for m = 2:2*ceil(m_E)
    tmp = 0;
    for i=1:m
        tmp = tmp + computeic(ciphertext(i:m:end));
    end
    tmp = tmp/m;
    ic(m) = tmp;
end

[~,m_hat] = max(ic);
ic
m_hat
```

```
ic =

    0.0435    0.0429    0.0540    0.0425    0.0421    0.0537    0.0454    0.0418    0.0657    0.0404    0.0462    0.0552

m_hat =

     9
```

Problem 2

Consider the sample space and probability distribution associated with a coin toss, using a possibly unfair coin. The sample space has dimension 2: (i) "the outcome of the coinflip is heads" with probability p, $0 \le p \le 1$; and (ii) "the outcome of the coinflip is tails" with probability (1–p). Recall that the coin is said to be fair when p=0.5

Prove that the entropy of this probability distribution is maximized when p=0.5; i.e. when the coin is fair.

Hint: Recall that to establish a maximum, you must examine both the first and second derivatives of the function in question – your proof will involve establishing appropriate conditions on these derivative values.

Formula of entropy: $H(\{p_i\}) = -\sum_i p_i \cdot \log_2(p_i)$

For a Bernoulli process (i.e. a coin flip): $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$

First derivative: $\frac{\partial}{\partial p} H(p) = -\log_2\left(\frac{p}{1-p}\right)$

Second derivative: $\frac{\partial^2}{\partial p^2} H(p) = \frac{1}{\ln(2)\, p(1-p)}$ } H is concave!

Find critical point by setting $\frac{\partial}{\partial p} H = 0$:

$$0 = -\log_2\left(\frac{p}{1-p}\right)$$

$$p = 0.5$$

Is this a maximum or minimum over our support $0 \le p \le 1$?

→ plug $p = 0.5$ into $\frac{\partial^2}{\partial p^2} H$

$$\frac{\partial^2}{\partial p^2} H(p=0.5) = -\frac{1}{\ln(2)\, 0.5(1-0.5)} = -5.77$$

$\frac{\partial^2}{\partial p^2} H(p)$ is negative for $p = 0.5$, so this is the maximum of $H(p)$.

Canvas questions: You were asked to evaluate $H(p)$ for $p=0.5$, $p=\frac{1}{3}$, and $p=1$.

$H(p=0.5) = 1$

$H(p=\frac{1}{3}) = 0.918$

$H(p=1) = 0$ → no uncertainty for $p=1$; $1-p=0$!

Problem 3

Your previous assignment examined a message constructed from the restricted character space of only the 14 uppercase English letters:

{A, B, C, D, E, F, G, H, I, M, N, R, S, T}

Suppose that for plaintext messages composed from this character set, the probability distribution for occurrence of these characters is given by:

	A	B	C	D	E	F	G	H	I	M	N	R	S	T
$\{p_i\}$	0.1061	0.0194	0.0362	0.0556	0.1643	0.0285	0.0259	0.0789	0.0906	0.031	0.0867	0.0776	0.0815	0.1177

a) Compute the entropy of this probability distribution.

Recall the definition of entropy: $H(\{p_i\}) = -\sum_i p_i \log_2(p_i)$

$H(\{p_i\}) = -(0.1061 \cdot \log_2 0.1061 + 0.0194 \cdot \log_2 0.0194 + \cdots + 0.1177 \cdot \log_2 0.1177)$

$= 3.5782$

Canvas question also asked to find the entropy of the distribution in Problem 1

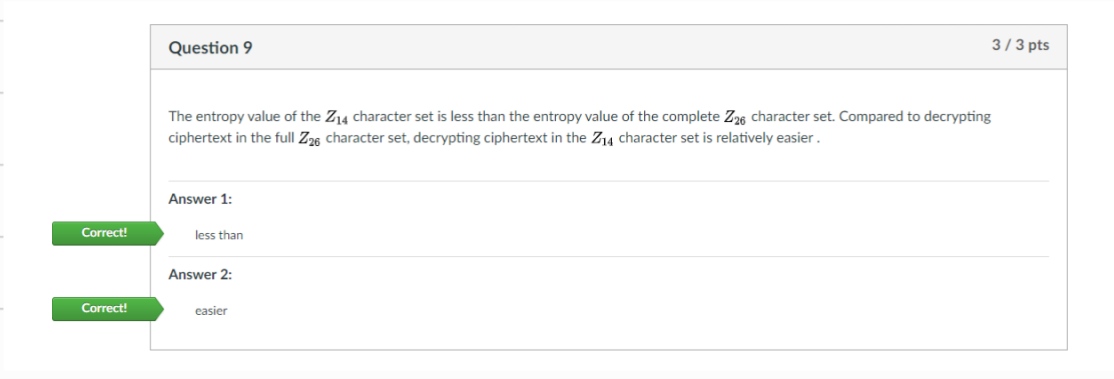
A	B	C	D	E	F	G	H	I	j	K	L	M	N
0.082	0.015	0.028	0.043	0.127	0.022	0.020	0.061	0.070	0.002	0.008	0.040	0.024	0.067

O	P	Q	R	S	T	U	V	W	X	Y	Z
0.075	0.019	0.001	0.060	0.063	0.091	0.028	0.010	0.023	0.001	0.020	0.001

In this case: $H(\{p_i\}) = -(0.082 \cdot \log_2 0.082 + \cdots + 0.001 \cdot \log_2 0.001) = 4.1807$

b) How does the entropy value of this Z_{14} character set compare to the entropy value of the complete Z_{26} character set? Interpret this result in regard to the relative difficulty of decrypting ciphertext in the Z_{14} character set, versus decrypting ciphertext in the Z_{26} character set.

This was asked as a fill-in-the-blank question on Canvas.



We can take the entropy value as a measure of uncertainty of a probability distribution.

The smaller character set has less entropy than the larger one, since there is less potential for randomness in a smaller domain when generating the ciphertext. This means it is relatively easier to decrypt a ciphertext with this smaller character set.