Problem 1 Find the entropy of the alphabet with letter probabilities shown below.

<table>
<thead>
<tr>
<th>k</th>
<th>p_k</th>
<th>-log_2 p_k</th>
<th>-p_k log_2 p_k</th>
<th>k</th>
<th>p_k</th>
<th>-log_2 p_k</th>
<th>-p_k log_2 p_k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1310</td>
<td>2.932</td>
<td>0.3841</td>
<td>14</td>
<td>0.0250</td>
<td>5.322</td>
<td>0.1330</td>
</tr>
<tr>
<td>2</td>
<td>0.1050</td>
<td>3.252</td>
<td>0.3414</td>
<td>15</td>
<td>0.0250</td>
<td>5.322</td>
<td>0.1330</td>
</tr>
<tr>
<td>3</td>
<td>0.0860</td>
<td>3.540</td>
<td>0.3044</td>
<td>16</td>
<td>0.0200</td>
<td>5.644</td>
<td>0.1129</td>
</tr>
<tr>
<td>4</td>
<td>0.0800</td>
<td>3.644</td>
<td>0.2915</td>
<td>17</td>
<td>0.0200</td>
<td>5.644</td>
<td>0.1129</td>
</tr>
<tr>
<td>5</td>
<td>0.0710</td>
<td>3.816</td>
<td>0.2709</td>
<td>18</td>
<td>0.0200</td>
<td>5.644</td>
<td>0.1129</td>
</tr>
<tr>
<td>6</td>
<td>0.0680</td>
<td>3.878</td>
<td>0.2637</td>
<td>19</td>
<td>0.0130</td>
<td>6.265</td>
<td>0.0814</td>
</tr>
<tr>
<td>7</td>
<td>0.0630</td>
<td>3.989</td>
<td>0.2513</td>
<td>20</td>
<td>0.0120</td>
<td>6.381</td>
<td>0.0766</td>
</tr>
<tr>
<td>8</td>
<td>0.0610</td>
<td>4.035</td>
<td>0.2461</td>
<td>21</td>
<td>0.0092</td>
<td>6.764</td>
<td>0.0622</td>
</tr>
<tr>
<td>9</td>
<td>0.0530</td>
<td>4.238</td>
<td>0.2246</td>
<td>22</td>
<td>0.0042</td>
<td>7.895</td>
<td>0.0332</td>
</tr>
<tr>
<td>10</td>
<td>0.0480</td>
<td>4.718</td>
<td>0.1793</td>
<td>23</td>
<td>0.0017</td>
<td>9.200</td>
<td>0.0156</td>
</tr>
<tr>
<td>11</td>
<td>0.0330</td>
<td>4.921</td>
<td>0.1624</td>
<td>24</td>
<td>0.0014</td>
<td>9.480</td>
<td>0.0133</td>
</tr>
<tr>
<td>12</td>
<td>0.0290</td>
<td>5.108</td>
<td>0.1481</td>
<td>25</td>
<td>0.0013</td>
<td>9.587</td>
<td>0.0125</td>
</tr>
<tr>
<td>13</td>
<td>0.0280</td>
<td>5.158</td>
<td>0.1444</td>
<td>26</td>
<td>0.0012</td>
<td>9.703</td>
<td>0.0116</td>
</tr>
</tbody>
</table>

Solution: The entropy is found by summing the terms in columns 4 and 8. The result is \( H = 4.12355 \) bits/letter.

Problem 2 Construct an instantaneous binary code for this alphabet. Compute the average length and efficiency for your code.

Solution: Use the Huffman coding procedure. The program huffman.pro is listed on the web site and was used to compute the following results.

Average Codeword Length\(= 4.15000 \)
Entropy\(= 4.12355 \)
Code Efficiency\(= 99.3626\% \)
Problem 3 Suppose that we want to construct a language that consists only of words that contain at most one occurrence of any letter. How many different words can constructed from an alphabet of size $r$? Note that this language has a limit in size.

Solution: Each letter can be used 0 or 1 time. The minimum word length is 1 and the maximum word length is $r$. There are $r$ words of length 1, $r(r-1)$ words of length 2, $r(r-1)(r-2)$ words of length 3,... $r!$ words of length $r$. Hence,

$$n(r) = r + r (r-1) + r (r-1) (r-2) + \cdots + r!$$

This expression can be written in terms of an iteration

$$n(r) = r [1 + n(r-1)]$$

For the first few values of $r$ the results are $n(1) = 1$, $n(2) = 4$, $n(3) = 15$, $n(4) = 64$. These cases are small enough to write out to verify the result.

Problem 4 Consider another language where the words are constructed of symbols drawn from a binary alphabet $A_2 = \{s_1, s_2\}$. In every word, $s_1$ occurs exactly $n_1$ times and $s_2$ occurs exactly $n_2$ times, with $n_1$ and $n_2$ fixed. How many words of length $n = n_1 + n_2$ can be formed in this manner?

There are $n!$ ways to arrange a sequence of $n$ objects. For example, there are $4! = 24$ ways to arrange the objects $\{a, b, c, d\}$. If some of them are indistinguishable, then some of the arrangements look alike. For example, the set $\{a, a, c, d\}$ still has 24 arrangements, but now switching the two a’s does not change the appearance. Hence, there are $4!/2 = 12$ unique arrangements. Similarly, $\{a, a, a, d\}$ has $4!/3! = 4$ unique arrangements. If a second symbol
can have repetitions, then we have to divide by the number of rearrangements of its subsequences. For example \(\{a, a, a, b, b\}\) will have \(5!/(3!2!) = 10\) unique arrangements. By this line of reasoning, the number of words of length \(n\) is

\[
N(n_1, n_2) = \frac{(n_1 + n_2)!}{n_1!n_2!}
\]

**Problem 5** In this problem we will investigate a formula that provides a good approximation to the number of words of length \(n = n_1 + n_2\) that can be constructed with the alphabet \(A_2 = \{s_1, s_2\}\) in which the fraction of each letter is kept constant as the length is increased. Let \(p\) be a parameter in the interval \([0, 1]\). Let \(n_1 = np\) and \(n_2 = n(1 - p)\). Clearly, \(n_1 + n_2 = n\). Let \(f(n, p)\) be the exact function for the number of such words (constructed by substituting \(n_1\) and \(n_2\) into the result of the previous problem). Let \(g(n, p) = -n[p \log_e p + (1 - p) \log_e (1 - p)]\). Let

\[
r(n, p) = \frac{\log_e f(n, p)}{g(n, p)}
\]

Plot \(r(n, p)\) vs \(\log_{10} n\) for \(p = 0.5\). Use \(n = 10, 100, 1000, \ldots, 10^6\). What do you conclude about the quality of the approximation?

**Solution:** From the previous problem,

\[
f(n, p) = \frac{n!}{(np)! (n - np)!}
\]

This is a difficult calculation to perform for large \(n\). However, we can do a sequence of calculations beginning with a smaller number \(k\) such that \(kp\) is an integer. Then \(k(1 - p) = k - kp\) is also an integer. Having computed the function for \(k\), one can calculate it for \(k, 2k, \ldots, n\) by iteration. The ratio \(f(mk, p)/f((m - 1)k, p)\) has only \(k\) terms in the numerator and \(2k\) terms in the denominator after cancellations. The program below will do the computation reasonably well up to \(n = 10^6\). The function returns \(\ln f(n, p)/g(n, p)\).
Function comby(n,k,p)
    ;+  ;f=comby(n,k,p) computes ln(n!)-ln((np)!)-ln((n(1-p))!) by starting with
    ;n=k and increasing by k at each step using the exact expansion of
    ;the factorials. This algorithm works only for kp=integer. It works ok up
    ;to n=10^6.
    ;
    ; SOLUTION SCRIPT
    ; p=0.25
    ; n=[10,100,1000,10000,100000L,1000000L,10000000L]
    ; r=fltarr(7)
    ; For t=0,6 Do r[t]=comby(n[t],t,p)
    ; Plot,Alog10(n),r,yrange=[0,1.2],xrange=[0,7],xticklen=1,yticklen=1,$
    ; xgridstyle=1,ygridstyle=1,xtitle='log n',ytitle='r(n)',psym=-4
    ;
    ; np=n*p
    ; kp=Fix(k*p)
    ; kq=k*(1-p)
    ; IF kp-k*p LT 0 THEN MESSAGE,'k*p must be an integer in comby(n,k,p)'
    ; g=-p*Alog(p)-(1-p)*Alog(1-p)
Problem 6 Derive the formula for $g(n, p)$ by starting with the formula for $f(n, p)$ and making use of the Stirling approximation

\[
\log_e(N!) \approx (N + \frac{1}{2}) \log_e N - N + \frac{1}{2} \log_e 2\pi
\]

Keep only the dominant terms as you let $n$ increase.

Solution: Replace each of the factorials with the above expression:

\[
\log_e f = n \log_e n + \frac{1}{2} \log_e n - n + \frac{1}{2} \log_e 2\pi
- np \log_e np - \frac{1}{2} \log_e np + np - \frac{1}{2} \log_e 2\pi
- n(1-p) \log_e n(1-p) - \frac{1}{2} \log_e n(1-p) + (n-1)p - \frac{1}{2} \log_e 2\pi
\]

Regroup and keep only terms that are linear in $n$.

\[
\log_e f \approx n \log_e n - [n - np - (n-1)p] - np \log_e np - n(1-p) \log_e n(1-p)
\]

The term in square brackets is zero. Expand the log terms and regroup again

\[
\log_e f \approx n \log_e n - np \log_e np + np \log_e n(1-p) - n \log_e n(1-p)
\]

A bit of simplification leads to the final answer

\[
\log_e f \approx -np \log_e p - n(1-p) \log_e (1-p)
\]
Noting that the entropy of a binary source in which one symbol has probability $p$ has entropy $H(p) = -p \log_2 p - (1 - p) \log_2 (1 - p)$, we find that the number of sequences of length $n$ grows as $e^{nH(p) \ln 2} = 2^{nH(p)}$.

If one would want to encode all sequences of length $n$ with binary words of length $m$, we would have to have $2^m \geq 2^{nH(p)}$, or $m \geq nH(p)$ digits per message. The average rate is $m/n$ binary digits per message symbol.