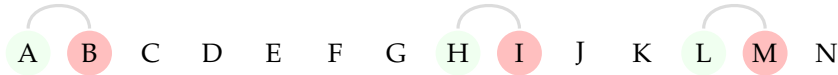


① 予備知識

2001年宇宙の旅に登場するコンピュータの名前「HAL」を1字ずつずらすと「IBM」になるという話はあまりに有名ですが、一般に、文字を置き換えることで元の情報（文章）を隠す手法を、換字式暗号と呼びます。



アルファベットの置き換えは（大文字と小文字の違いを無視すると）全部で $26! \approx 4 \times 10^{26}$ 通りもあり、一定数ずつずらすような安易な暗号（シフト暗号）ではなく、手の込んだ置き換えを考えれば、十分な安全性が確保できそうな気がしますが、換字式暗号は危険な（解読される可能性が高い）暗号と言われています。その理由の一つは、文字の出現頻度に偏りがあることが知られているからです。

② 文字数を数える

Maxima を使って、文章に含まれる文字数を、アルファベット毎に集計する関数を作ってみます。なお、下記の関数では、アルファベットを全て小文字に変換して数えることにしています。

```

1  count(file) := block([s,                                     - count.mac -
2      L: makelist(0, i, 1, 26)],
3      for i in read_list(file) do (
4          s: sdowncase(sprintf(false, "~a", i)),
5          for j in charlist(s) do
6              if alphacharp(j) then
7                  L[cint(j) - 96]: L[cint(j) - 96] + 1
8      ),
9      return(L)
10 );
```

3行目は、関数 `read_list`（要 `numericalio.mac`）によってファイルを読み込み、各単語（変数 `i`）毎に `do` 文を実行する処理です。4行目は、関数 `sprintf` を用いて各単語（変数 `i`）を「文字列（string）」に変換し、かつ、全て小文字に変換する処理です。

5行目の関数 `charlist` は、文字列を引数にとり、文字毎に分解したリストを返す関数です。従って、3行目の変数 `i` が例えば「Alice」ならば、`charlist(s)` は `[a, l, i, c, e]` となり、変数 `j` には、`a, l, i, c, e` がこの順に代入されます。変数 `j` にはアルファベット以外

の記号も代入されることがあるため、アルファベットか否かを判別し、アルファベットの場合のみ数えるようにしています (6行目)。その際、アルファベットの ASCII コード

a	b	c	d	e	f	g	h	i	j	k	l	m
97	98	99	100	101	102	103	104	105	106	107	108	109
n	o	p	q	r	s	t	u	v	w	x	y	z
110	111	112	113	114	115	116	117	118	119	120	121	122

を利用し、a、b、c、... の個数を順に L[1]、L[2]、L[3]、... に代入するために、それぞれのアルファベットの ASCII 番号から 96 を引いています (7行目)。

③ 不思議の国のアリスとタイムマシン

アルファベットの出現回数を調べるため、適当は文書を用意します。英語のテキスト文書なら何でも良いのですが、ここでは、Project Gutenberg から入手できる Lewis Carroll の「Alice's Adventures in Wonderland」と Herbert George Wells の「The Time Machine」を調査対象とします。

- Alice's Adventures in Wonderland ... <http://www.gutenberg.org/etext/11>
- The Time Machine ... <http://www.gutenberg.org/etext/35>

Project Gutenberg で配布されている文書には、ファイルの最初と最後に License 関連の文言等が含まれていますので、あらかじめ当該部分を削除しておきます。ファイルが用意できたら、早速前節の関数を使って、文字数を数えてみましょう。

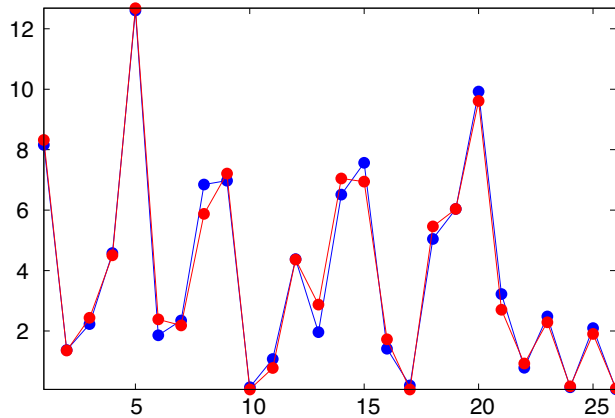
```
(%i1) load("numericalio")$
(%i2) load("count.mac");
(%o2)                                     count.mac
(%i3) A: count("11.txt");
(%o3) [8791, 1475, 2399, 4931, 13575, 2001, 2531, 7374, 7514, 146, 1158,
4716, 2107, 7016, 8146, 1524, 209, 5438, 6500, 10688, 3468, 846, 2675,
148, 2262, 78]
(%i4) T: count("35.txt");
(%o4) [11704, 1897, 3424, 6337, 17837, 3354, 3075, 8257, 10137, 97, 1087,
6146, 4043, 9917, 9758, 2427, 95, 7673, 8486, 13513, 3804, 1295, 3225,
236, 2679, 144]
```

関数 count 内で関数 real_list を利用するため、最初に numericalio.mac を読み込んでいます。

出力 (%o3) が「Alice's Adventures in Wonderland」です。a が 8761 個、b が 1475 個、... という結果です。一方「The Time Machine」(%o4) は a が 11704 個、b が 1897 個、... という結果です。

両者の類似性を調べるため、グラフを書いてみます。

```
(%i5) load("draw")$
(%i6) draw2d(points_joined = true, point_type = 7, point_size = 2,
             color = blue, points(A/apply("+", A)*100),
             color = red, points(T/apply("+", T)*100));
(%o6) [gr2d(points, points)]
```



青の折れ線が「Alice's Adventures in Wonderland」、赤の折れ線が「The Time Machine」です。明らかに相関が認められますが、念のため、相関係数も計算してみたいと思います。

```
(%i7) a: apply("+", A)/26;
(%o7) 53858
-----
13
(%i8) t: apply("+", T)/26;
(%o8) 10819
-----
2
(%i9) sum((A[i] - a)*(T[i] - t), i, 1, 26)
      /sqrt(sum((A[i] - a)^2, i, 1, 26))
      /sqrt(sum((T[i] - t)^2, i, 1, 26)), numer;
(%o9) 0.993320189909119
```

あらかじめ平均値を計算しておき、定義通りに相関係数を計算した結果、0.99 という非常に高い値であることが分かりました。