The Mathematics of Enigma

The Enigma machine is the most famous cipher machine to date. Nazi Germany used it during World War II to encrypt messages so that enemies could not understand them. The story of the British cryptanalysts who successfully deciphered Enigma has become the subject of multiple movies *Enigma* (2001); *The Imitation Game* (2014). In this exercise, we will focus our attention on a simplified version of the Enigma machine, which we name "Little Enigma." Like the real Enigma machine shown in the picture above, this machine consists of two key components. First, the Little Enigma machine has 5 different *rotors*, each of which comes with 10 pins with numbers ranging from 0 to 9. Second, a component called the *plugboard* contains 26 holes, corresponding to the 26 letters of the alphabet. In addition, 13 cables connect all possible pairs of letters. Since a cable has two ends, one can connect, for example, the letter A with any other of the other 25 letters present in the plugboard.

To either encode a message or decode an encrypted message, one must provide the Little Enigma machine with a correct five-digit passcode to align the rotors and a correct configuration of the plugboard. The rotors are set up just like many combination locks. For example, the passcode 9–4–2–4–9 means that five rotors display the numbers 9, 4, 2, 4, and 9 in that order. In addition, the 13 cables connecting the letters in the plugboard must be appropriately configured. The purpose of the plugboard is thus to scramble the letters. For example, if B is connected to W, the Little Enigma machine will switch B with W and W with B to encode a message or decode an encoded message. Thus, a sender types a message on the keyboard, the plugboard scrambles the letters, and the message is sent in its encrypted form. A receiver decodes the encrypted message by re-typing it on a paired Little Enigma machine that has the same passcode and plugboard configuration.

Question 1

How many different five-digit passcodes can be set out of the 5 rotors?

Question 2

How many possible configurations does the plugboard provide? In other words, how many ways can 26 letters be divided into 13 pairs?

Question 3

Based on the previous two questions, what is the total number of possible settings for the Little Enigma machine?

Question 4

Five cryptanalytic machines have been developed to decode 1,500 messages encrypted by the Little Enigma machine. The table below presents information on the number of messages assigned to each machine and the machine's failure rate (i.e., the percentage of messages the machine was unable to decode). Aside from this information, we do not know anything about the assignment of each message to a machine or whether the machine was able to correctly decode the message.

Machine	Number of messages	Failure Rate
`Banburismus`	300	10%
`Bombe`	400	5%
`Herivel`	250	15%

Machine	Number of messages	Failure Rate
`Crib`	340	17%
`Hut 6`	210	20%

Suppose that we select one message at random from the pool of all 1,500 messages but found out this message was not properly decoded. Which machine is most likely responsible for this mistake?

Question 5

Write an R function that randomly configures the plugboard. This function will take no input but randomly selects a set of 13 pairs of letters. The output object should be a 2×13 matrix for which each column represents a pair of letters. You may use the built-in R object `letters`, which contains the 26 letters of the alphabet as a character vector. Name the function `plugboard`.

Question 6

Write an R function that encodes and decodes a message given a plugboard configuration set by the `plugboard` function from the previous question. This function should take the output of the `plugboard` function as well as a message to be encoded (decoded) as inputs, and return an encoded (decoded) message. You may wish to use the `gsub` function, which replaces a pattern in a character string with another specified pattern. The `tolower` function, which makes characters in a character vector lowercase, and `toupper` function, which capitalizes characters in a character vector, can also help.

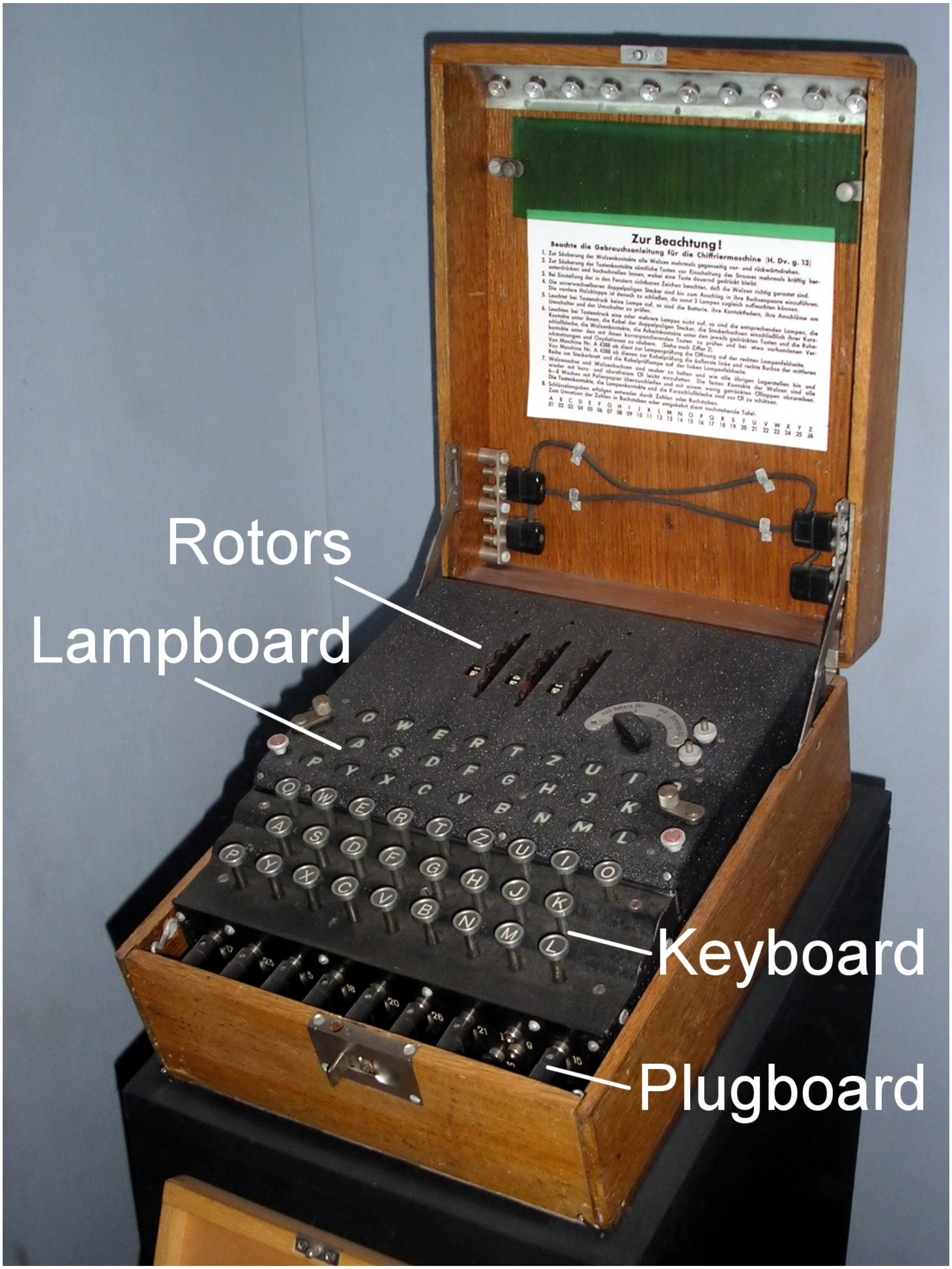

Figure 1: The Enigma Machine