

# Faulty LED seven-edge digit recognition

Josep Fortiana 2019-11-26

## Original problem

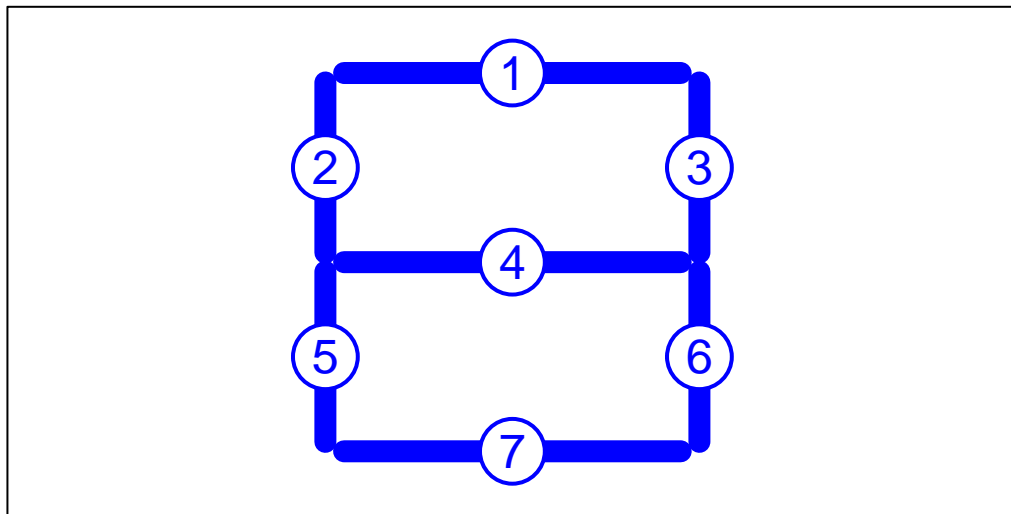
The problem involves representations of digits by seven LED lights which are selectively lit to form each digit. It is described in the book: Leo Breiman, Jerome Friedman, R.A. Olshen and Charles J. Stone (1984) *Classification and Regression Trees* (BFOS), p. 43-49.

`Digits.plot.r` contains auxiliary functions for visualizing them.

```
source("Digits.plot.r")
```

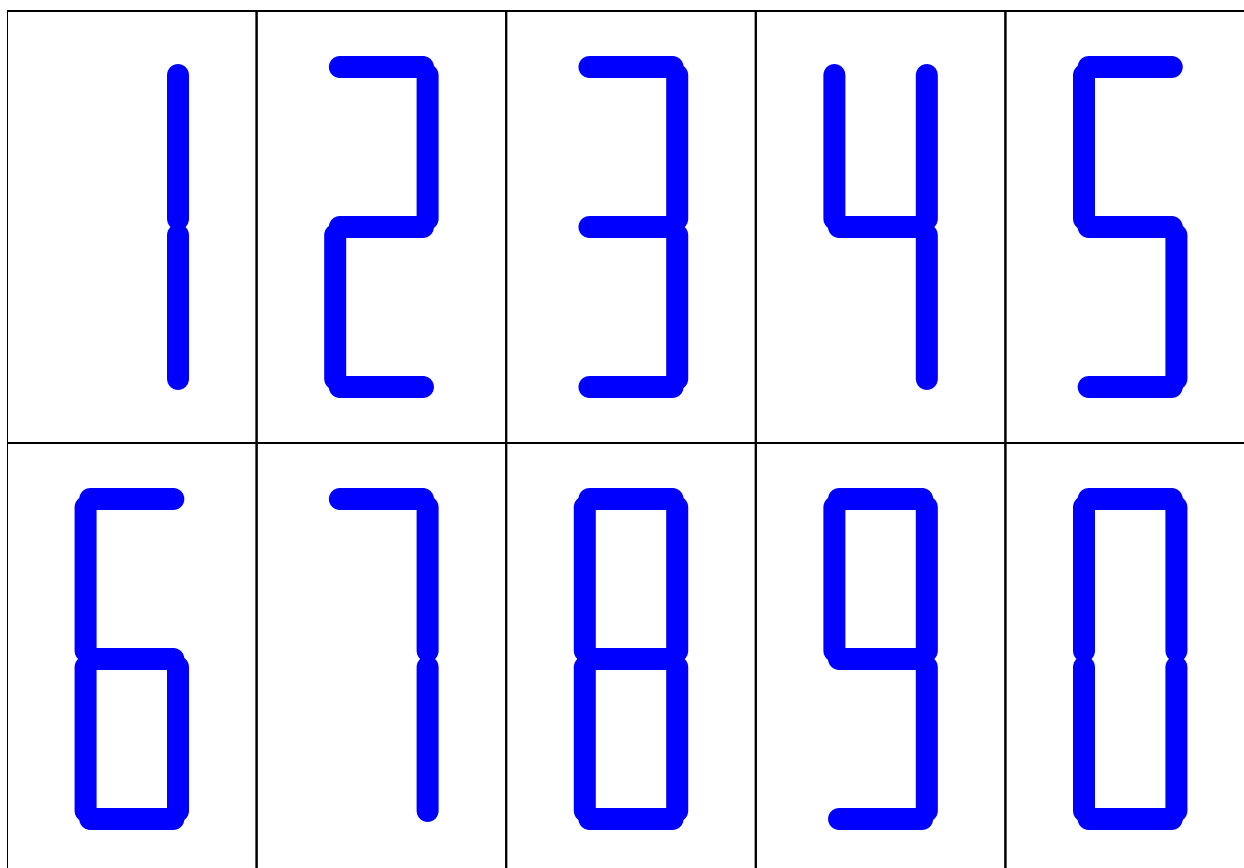
The figure below shows the numbering of the seven edges used to form the digits

```
options(repr.plot.width=4,repr.plot.height=4)
plot.digits.matrix.with.segment.indexes()
```



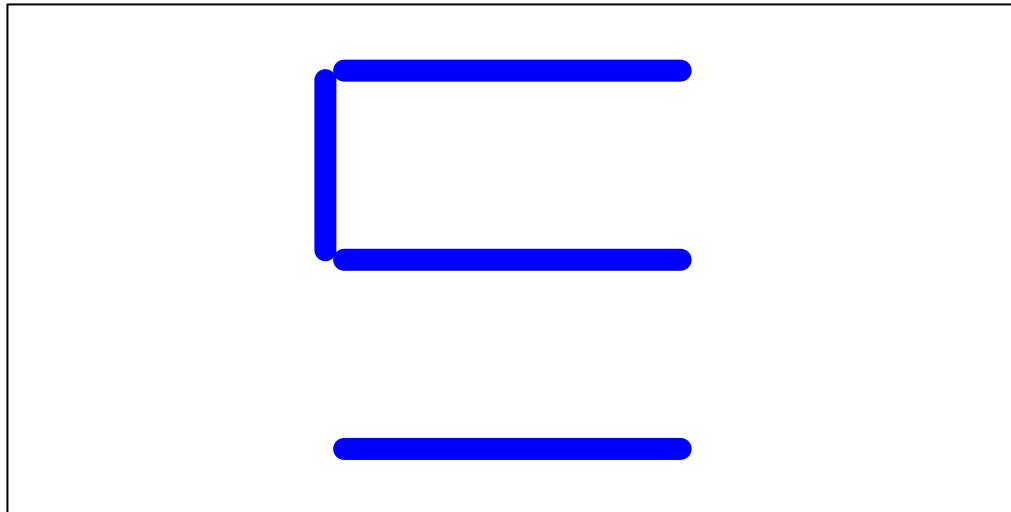
Here we see the ten digits, represented by the seven edges.

```
options(repr.plot.width=7.0,repr.plot.height=2.5)
show.all.digits()
```



The file `digits.csv` contains 500 sets (rows), each with seven ones and zeros, representing ON/OFF status for each of the seven edges, numbered as above, plus a label, the intended digit. Most rows, however, contain faulty edges, set ON when they should be OFF and vice-versa. For instance, the following picture shows a faulty '5' where the sixth edge failed to light on.

```
options(repr.plot.width=2.5,repr.plot.height=3.0)
v<-as.logical(c(1,1,0,1,0,0,1))
plot.faulty.digit(v)
```



Each row in `digits.csv` contains an additional variable, `DIGIT`, with the name of the intended digit.

```
digits.data<-read.csv("digits.csv")
str(digits.data)
```

```
## 'data.frame':    500 obs. of  8 variables:
## $ digit: Factor w/ 10 levels "eight","five",...: 6 5 3 9 1 5 2 7 9 1 ...
## $ x1   : int  1 0 0 1 0 0 1 1 1 1 ...
## $ x2   : int  0 0 1 1 1 0 1 0 0 1 ...
## $ x3   : int  1 1 1 1 1 1 0 0 1 1 ...
## $ x4   : int  0 0 1 1 1 0 1 1 1 1 ...
## $ x5   : int  0 0 0 1 1 0 0 1 1 0 ...
## $ x6   : int  1 1 1 0 1 1 1 1 0 1 ...
## $ x7   : int  0 0 0 0 1 0 1 1 1 1 ...
```

```
levels(digits.data$digit)
```

```
## [1] "eight" "five"  "four"  "nine"  "one"   "seven" "six"   "three" "two"
## [10] "zero"
```

Better with shorter (and equal length) names for levels

```
#install.packages("dplyr",dependencies=TRUE,repos="https://cloud.r-project.org")
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
digits.data$digit<-recode(digits.data$digit,"one"="1","two"="2",
  "three"="3","four"="4","five"="5","six"="6","seven"="7",
  "eight"="8","nine"="9","zero"="0")

str(digits.data)

## 'data.frame':   500 obs. of  8 variables:
## $ digit: Factor w/ 10 levels "8","5","4","9",...: 6 5 3 9 1 5 2 7 9 1 ...
## $ x1   : int  1 0 0 1 0 0 1 1 1 1 ...
## $ x2   : int  0 0 1 1 1 0 1 0 0 1 ...
## $ x3   : int  1 1 1 1 1 1 0 0 1 1 ...
## $ x4   : int  0 0 1 1 1 0 1 1 1 1 ...
## $ x5   : int  0 0 0 1 1 0 0 1 1 0 ...
## $ x6   : int  1 1 1 0 1 1 1 1 0 1 ...
## $ x7   : int  0 0 0 0 1 0 1 1 1 1 ...

table(digits.data$digit)

##
##  8  5  4  9  1  7  6  3  2  0
## 58 54 52 55 43 48 50 55 43 42
```

## Generating more data

Possibly the `digits.csv` dataset is extracted from the collection of datasets in the *Splus* program. In the BFOS book they mention an (unpublished, I believe) training dataset with 200 observations and a test sample of 5000. The generator program by David W. Aha, in C, can be found at the UCI Machine Learning repository.

A better alternative is the `rdigitsBFOS()` function from the `gencve` package, which generates digits with a given flipping probability `alpha` (that is, the probability of one of the seven LEDs changing its correct status to the opposite one). As a bonus the function computes the Bayes optimal missclassification rate, which can be used to have an objective lower bound for the quality of any classification method.

```
#install.packages("gencve",dependencies=TRUE,repos="https://cloud.r-project.org")
require(gencve)
```

```
## Loading required package: gencve
```