

MATLAB include:

1. Based on Python, a full-featured modern object-oriented programming language suitable for large-scale software development
2. Free and open source
3. Native SVG support

## 5.3 Pseudo code

The pseudo code for various algorithms are mentioned in this section.

### 5.3.1 Determine optimal bands procedure

This module groups the events into bands. The band formation is based on the Jump clustering algorithm.

**Function** `determineOptimalBands(S.W[], K)`

**Input:** Array  $A = \{a_0..a_N\}$ , window size for stream  $N$ , number of bands  $K$

**Output:** 2-dimensional array `band`, containing indices of per group elements in  $A$

1.  $J[N] \leftarrow \{0\}$  //store jumps between consecutive values of  $A$
2.  $J_s[K - 1] \leftarrow \{0\}$  // stores the top  $(K - 1)$  jumps
3. `upper, lower, nelements`  $\leftarrow 0$
4.  $I[K - 1] \leftarrow 0$ ,  $I_s[K - 1] \leftarrow 0$   
// stores indices in  $A$  of top  $((K - 1))$  jumps.
5. Sort  $A$  in descending order and copy into array  $A_s$   
// Determining jumps between consecutive values of  $A_s$
6. for  $i = 0$  to  $N$  do
7.    $J[i] \leftarrow A_s[i] - A_s[i + 1]$
8. end
9. Determine the top  $K - 1$  jumps from array  $J$
10. Store them in  $J_s[0]$ ,  $J_s[1]$  . . .  $J_s[K - 2]$
11. Store indices of  $J_s[0]$  to  $J_s[K - 2]$  in  $J$  into array  $I$
12. Sort array  $I$  in ascending order and store in  $I_s$
13. for  $i = 0$  to  $K - 1$  do
14.   `upper`  $\leftarrow I_s[i]$