D.d tick trace cruncher in Elisp

```
;;; -*- lexical-binding: t -*-

;; for a buffer/file of tick-time integers
;; one at each line
;; this Elisp number cruncher
;; is used to process the tick times
;; in order to find out how much
;; the ticks deviated
;; from the desired, fixed-interrupt rate

;; for the particular trace
;; stats are presented
;; as well as every drift from the specified ideal
;; as an indicator of an inexact clock
;; or if this needs to be further analyzed

(require 'cl-macs)

(defun get-variance (mean)
  (save-excursion
    (goto-char 1)
    (let ((sum 0) (offsets 0))
      (cl-loop
       (let ((offset (thing-at-point 'number)))
         (if offset
             (progn
               (cl-incf offsets)
               (setq sum (+ sum (expt (- offset mean) 2)))
               (forward-line 1) )
           (cl-return) )))
      (/ sum offsets) )))

(defun tick-stats (desired-tick)
  (interactive "n desired tick: ")
  (save-excursion
    (goto-char 1)
    (let ((sum 0) (offsets 0) (max nil) (min nil) )
      (cl-loop
       (let((t0 (thing-at-point 'number)))
```