Histograms for Luka-chucK system - Michael Lukaszuk

These four histograms were designed to process data in a way that would be useful in music performance/creation scenarios. For example, understanding how similar array contents are gives us a sense of pitch centricity (or rhythmic centricity once rhythm tracking additions are included in the system). The fourth histogram could allow the system to understand how much high vs. low frequency materials are being inputted or outputted. Adjustments could be made to improve audio output.

Histogram 1 - Set similarity

```
60 spork ~ pitchSets(["hi","bye", "good", "bad"],["hi", "bye","good","great"],1);
61
```

Two sets, each containing 4 elements - the int after the sets controls whether we are returned a degree of similarity or difference between the two sets

Results tell us: there are 3 common elements out of 4 elements per set - the sets are 80% similar.

For this Luka-chucK system, can be used according to a timer, check how often certain chords are being detected over a period of time. System currently defaults to 1 minute intervals.

If set similarity is less than 15% then the ensemble operates with more randomization and dissonance in pitch content.

Histogram 2 - Modal Value

Array used for testing

```
7 ["hi","hi","hi","bye","hi"] @=> string set[];
```

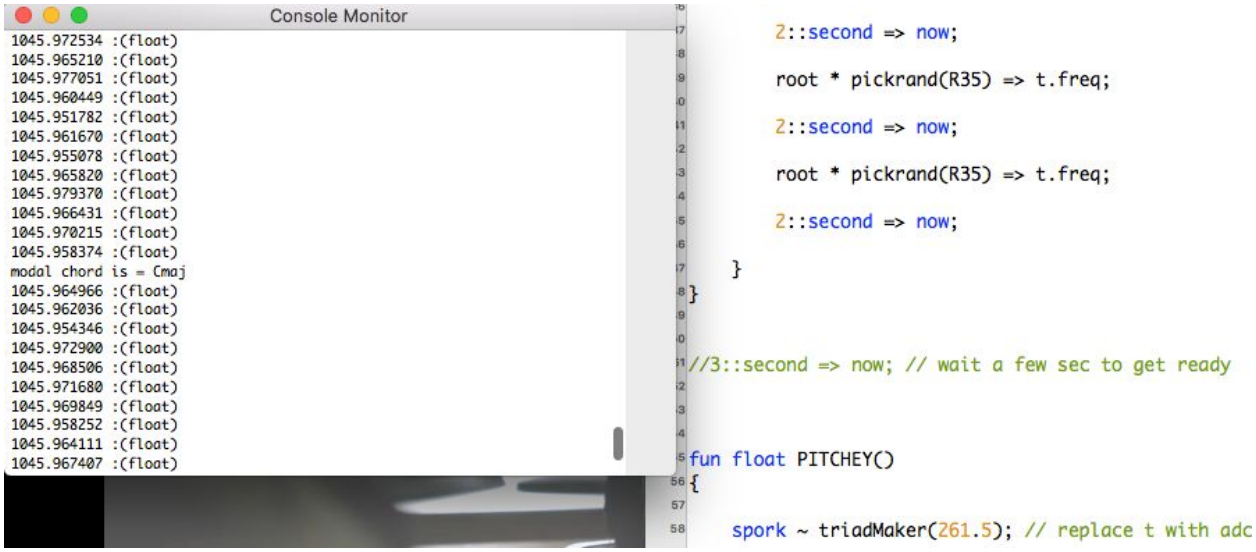
Calling the function

```
99 modal(set) => int mode;
100
101 finalTest(set,mode) => string final;
102
```

Result

```
[chuck](VM): sporking incoming shred: 2 (Histo2 Modal Val)...
modal val = hi
```

The base frequency of a simple synthesizer is set to 261.5Hz middle C. The synth plays single notes of C major triads. Occasionally if there is too much focus on the chordal 5th, the G, the system will interpret a Gmaj triad. Over a 30'' period the system detected more Cmaj triads than any other, therefore Cmaj is recognized as the modal value.



Histogram 3 - How Many From an Array

Array used for testing

```
["hi","bye","bye","bye"] @=> string set[];
```

Testing the string "bye"

```
spork ~ pitchSet(set, "bye");
```

Result tells us that there are three instances of the string "bye"

```
[chuck](VM): sporking incoming shred: 1 (Histo3 Arrayamount)...
3 :(int)
```

Histogram 4 - Frequency Threshold

Array used for testing

```
[220.,440.,660.,770.,1100.,2200.] @=> float set[];
```

Threshold set to 500Hz

```
33 thresh(set, 500.) => string result;
34
```

Code from inside the function: if 60% of array members > threshold, a string called “scaleDown” is returned and frequencies are scaled down.

```
if (set.cap() / exceeds > 0.6)
    return dropFreqs;
else
    return normalVals;
```

Result

```
[chuck](VM): sporking incoming shred: 1 (Histo4 Freq Threshold )...
"scaleDown" : (string)
```