

## INPUT ARGUMENTS

case: → call a function from here

- $n = \star$  -- now =  $\star \leftarrow \text{now} - n$  → implicitly default value  
"true" in this case
- $b = \star$  -- buckets =  $\star \leftarrow \text{buckets} - b$  → implicitly default values  
"intelligent" value finding (?)
- $match = \text{strict/loose } \{ ? \}$ .

Consider some generalization and separate function for assigning default parameters...  
Have the ~~elephant~~s at the beginning? The order should be as follows → if non-" $=$ " found  
then assign the default?

# PYTHON VISUALIZER

→ for the percentages found  
P1 E GRAPH

Support for having the arguments anywhere?

We cannot even export the array from Bash, so gives we are stuck with pickling or some clever way of encoding the in the commandline args, like  
it needs to be the same as \*

py X -f Vol 1, Vol 2, Vol 3, Vol 4, ... Is that feasible?

Q:

1. How to pass data from Bash to the Python script?

2. What kind of args are accepted by Python (`input/CLI args`)

3. gnm plot (?)

4. Bind C++ version to Python Vmulator and the Bash version...

# HOW TO PASS VARIABLES TO PYTHON SCRIPT

stmt  
name  
"errorCode" →  
py n × val1 val2 ... val(n) name1 name2 name(n)  
↑ (map)  
number -----  
of -----  
stmt -----  
(like error code: 1  
errorCode: 2 3 stmt  
errorCode: 4 2

Buckets and logarithmic -- find-gaps =  $y$  (?)

We have a number of occurrences of the stat name ...

error code \*

number of buckets

5 error code: 0

↑  
↑  
↑  
↑  
↑  
number of name  
occurrences  $\xrightarrow{x=3}$  stat  
value

$\lceil \min, \max \rceil \cdot (\max - \min) / x$

$$5 \quad 15 \quad (15 - 5) / 3 = 10 / 3 = 3, (3) \approx 3, 33$$
$$-10 \quad 20 \quad (20 - (-10)) / 3 = 30 / 3 = 10$$

Bucketed Occurrence Value Array  $[x] = \text{occur} (?)$

~~buckets Intervals  $[x]$~~  ~~array of arrays!~~ Bash does not support multi-dim arrays!  
~~array of arrays!~~: Yeah, we need to find some other way to store first information.

buckets vs non-numerical (?)

buckets vs outliers

-- ignore-outliers =  $y$

some percentile

# STAT AMBIGUITY PROBLEM . . .

error code → will be found by grep

error code derived → because of the "\*" in grep that would also be found but won't be counted to specific stat percentage share

## C++ BASED ENGINE

Just open the file, getline, we std::regex to search for the parameter/stat...