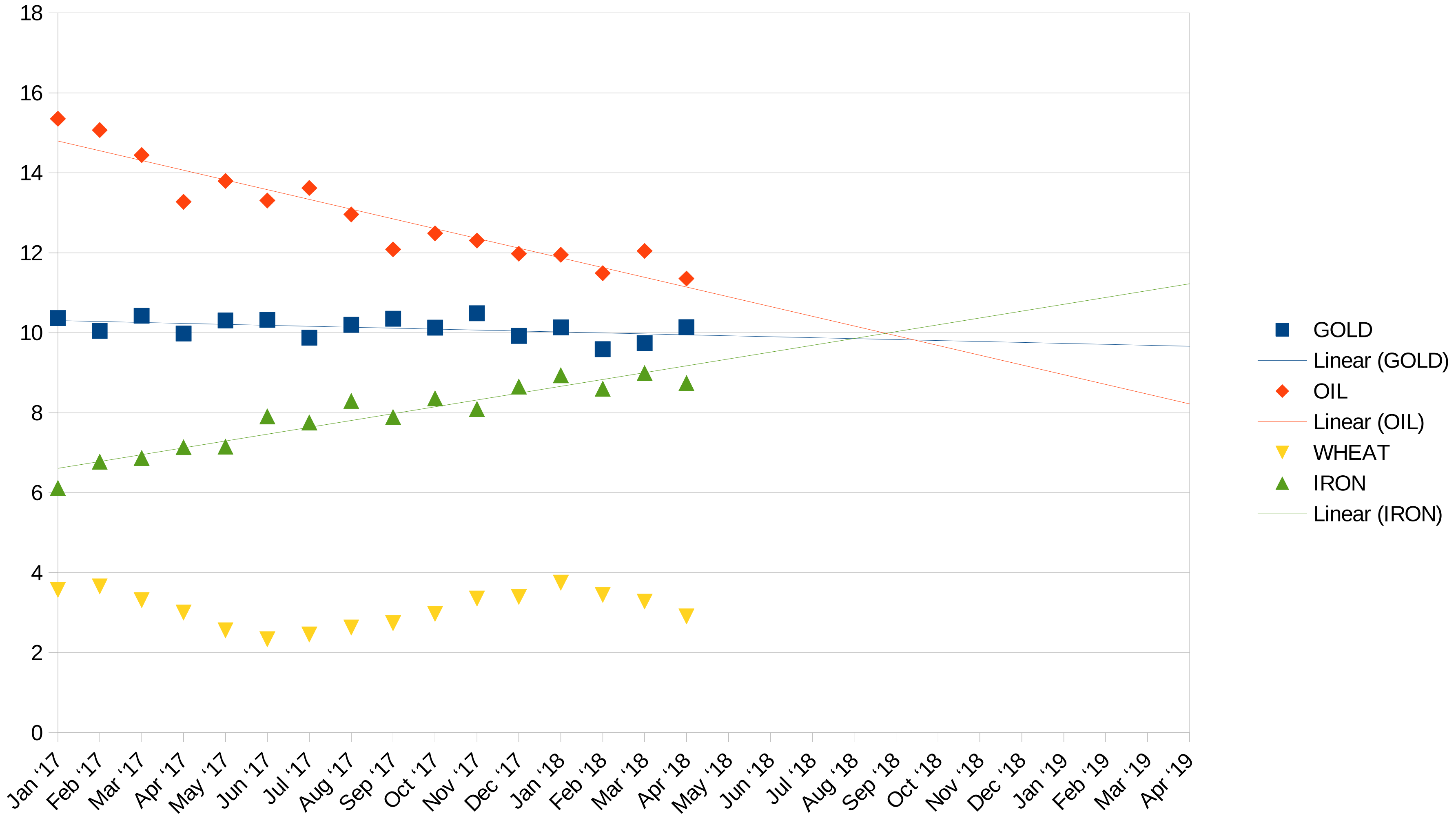
Village Software Data Quest

Team: **Kadane Combinatorics**

Jiri Dohnalek & Edward Barker

Selecting the 3 optimal plots of resources to maximise prosperity!

Analysis of price trends



Normalising range of values

X is based upon future sale price prediction.

Iron has maximum sales price, wheat has minimum.

- GOLD = (10 – 2.5)/(10.7-2.5) = 0.91
- OIL = (10.5 – 2.5)/(10.7-2.5) = 0.98
- WHEAT = (2.5-2.5)/(10.7-2.5) = 0.12
- IRON = (10.7 – 2.5)/(10.7-2.5) = 1.00

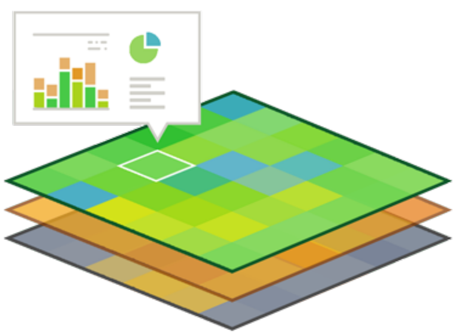
Use normalised values to transform arrays

to reflect relative values

$$\frac{x - min}{max - min}$$

Problem

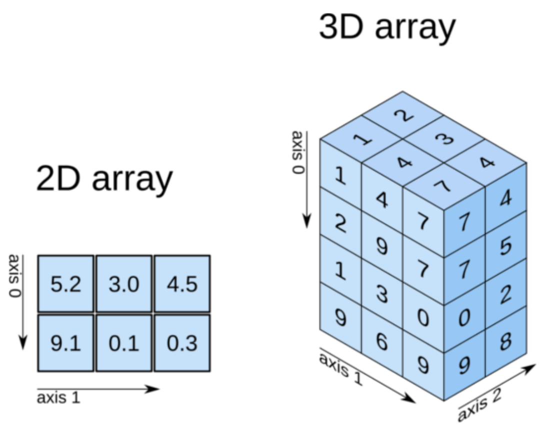
- Given 1000 x 1000 square matrix, find 3 highest sub-squares of 10 x 10



- Find highest combination of squares such that sub-squares do not overlap

Algorithm approach

- Apply weighting to each array based upon relative values
- Stack each 2D 1000 x 1000 array into 1 3D array.
- “Flatten” 3D array by summing “z-axis”.



Kadane's algorithm

- **Brute force approach would be time complexity of at least n^3**
- **Kadane's algorithm allows n^2 time**
  - Preprocess matrix
    - Calculate sum of all vertical strips of size k x 1 (k=10)
    - Calculate first sub-square in row as sum of first k strips
    - For remaining sub-squares, calculate sum in O(1) time by removing leftmost strip of previous square and adding rightmost strip of new square.

K-combinatorics

- **Highest squares may be overlapping (as may share same high resource tiles)**
  - Must ensure that top squares do not overlap, but also that the top 3 squares are the optimum combination. For instance, 2nd and 3rd highest squares taking the place of the 1st highest square will have a higher total.
  - Using combinatorics, generate all combinations, for each combination
  - Must choose 3 squares from 35, as 35 overlapping squares is the minimum number of squares to ensure that the three optimum squares are non-overlapping.
    - Check for overlap
    - Store highest sum of 3 squares.

$$C(n,k)=\frac{n!}{(n-k)!\,k!}$$