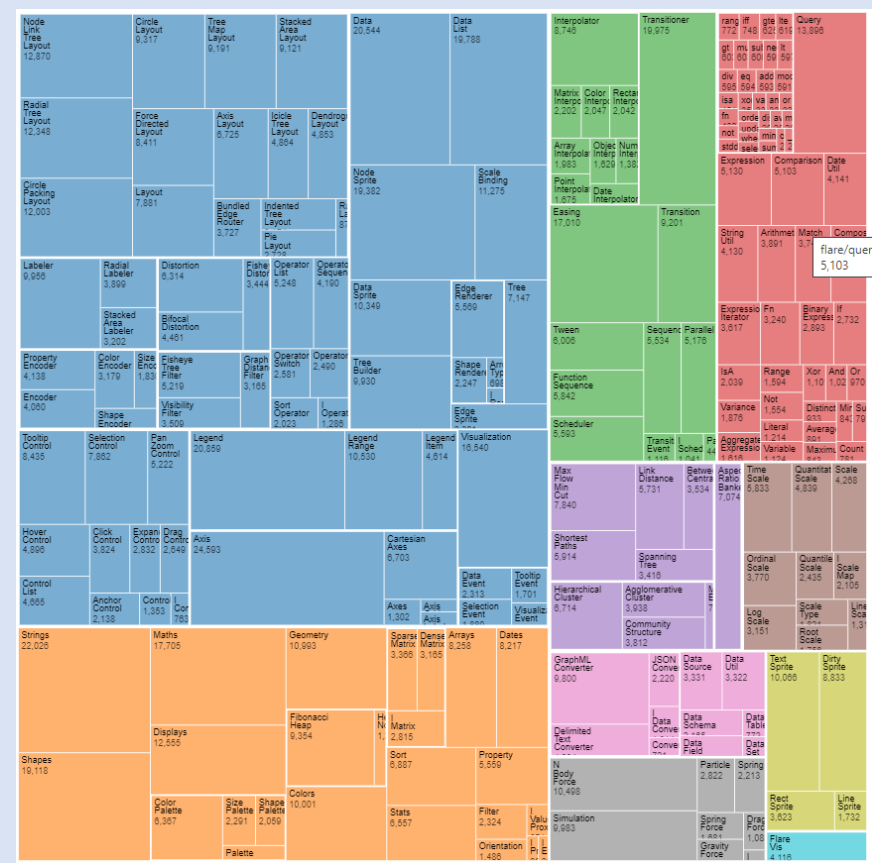
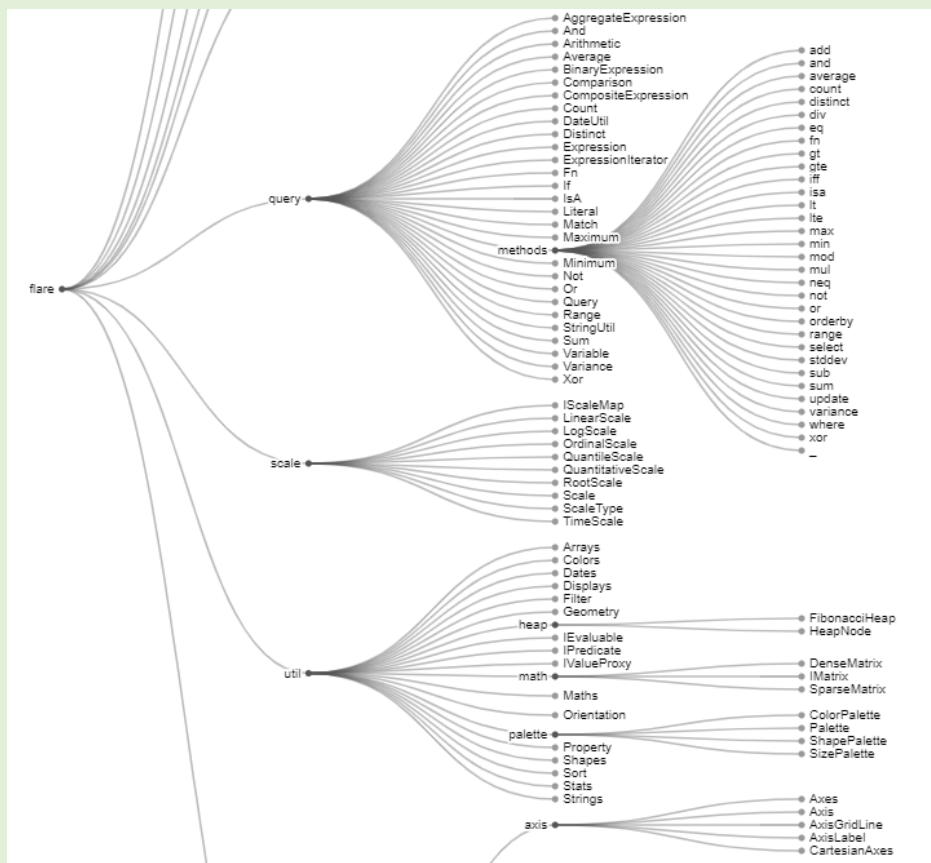


TREEMAP E GERARCHIE

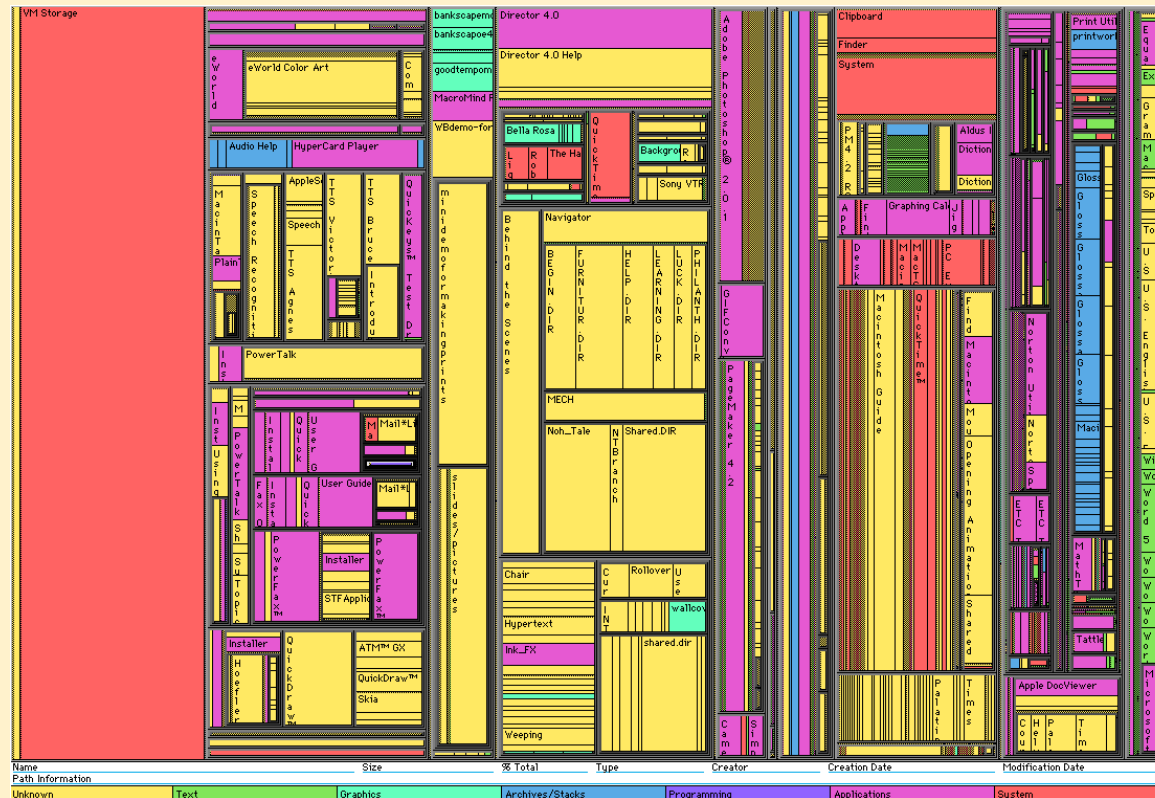
Un metodo basato su aree e colori per mostrare dati gerarchici

Non sono alberi, ma una loro rappresentazione



Storia

"Since the 80 Megabyte hard disk in the HCIL was shared by 14 users it was difficult to determine how and where space was used."



TREEMAPS

Storia

Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures

(Brian Johnson, Ben Shneiderman) – Ottobre 1991

Sviluppo di un software per Macintosh a colori, TreeViz

Visualization of Hierarchical Information Structures

Page 11

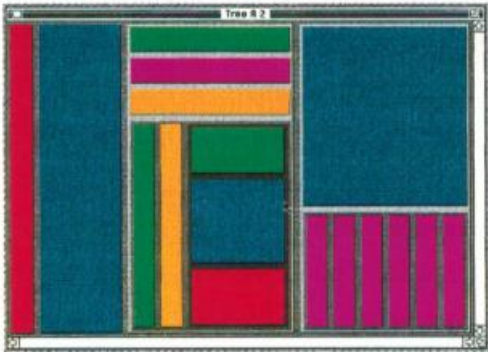


Figure 6. Nested Tree-Map
Screen Snapshot

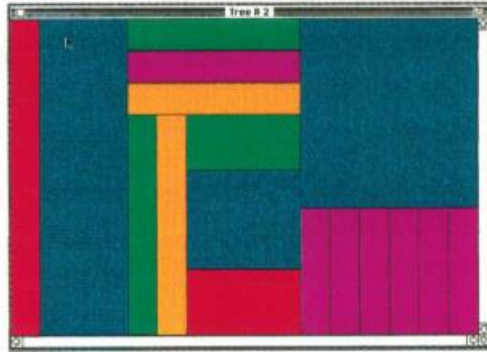


Figure 7. Non-nested Tree-Map
Screen Snapshot

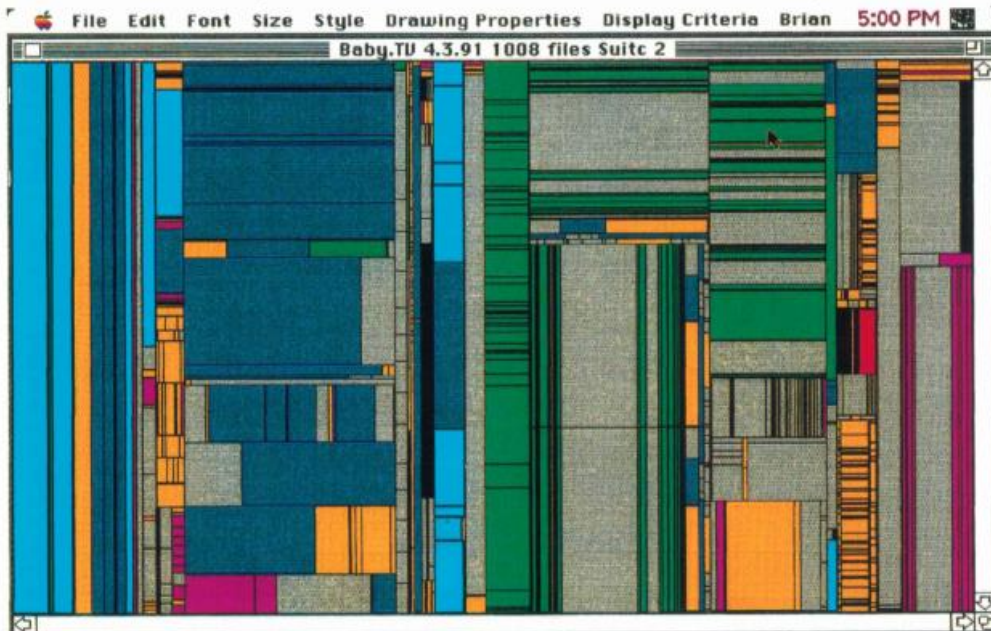


Figure 8. Tree-Map Screen Snapshot with 1000 Files

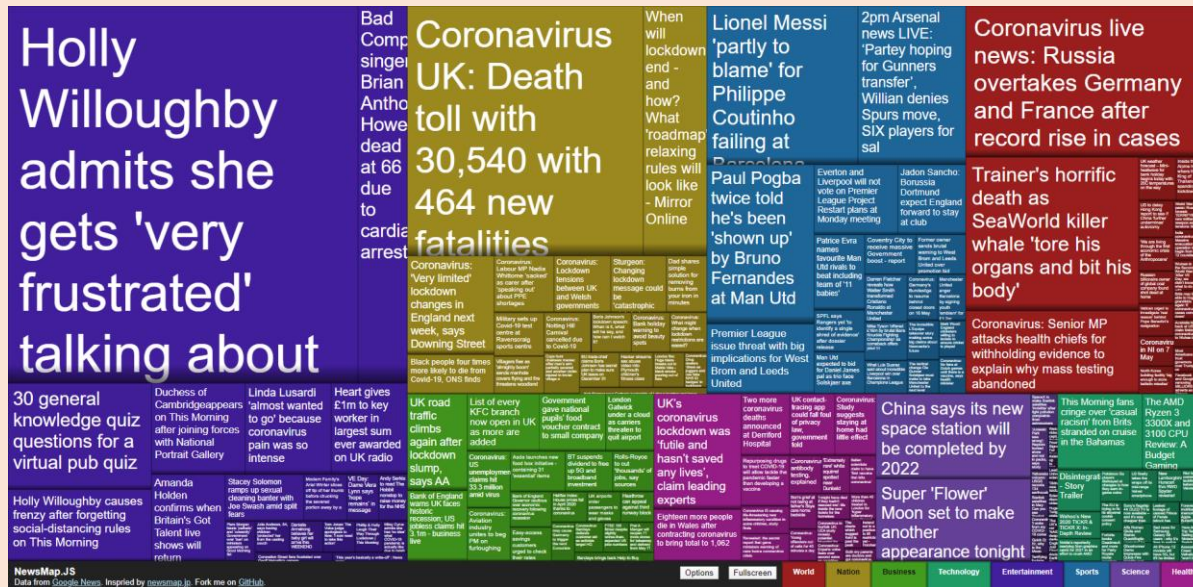
TREEMAPS

Struttura

Un riquadro rappresenta un nodo dell'albero

L'area del riquadro rappresenta il valore di un suo attributo numerico

Opzionalmente, il colore rappresenta un secondo attributo, anche categorico

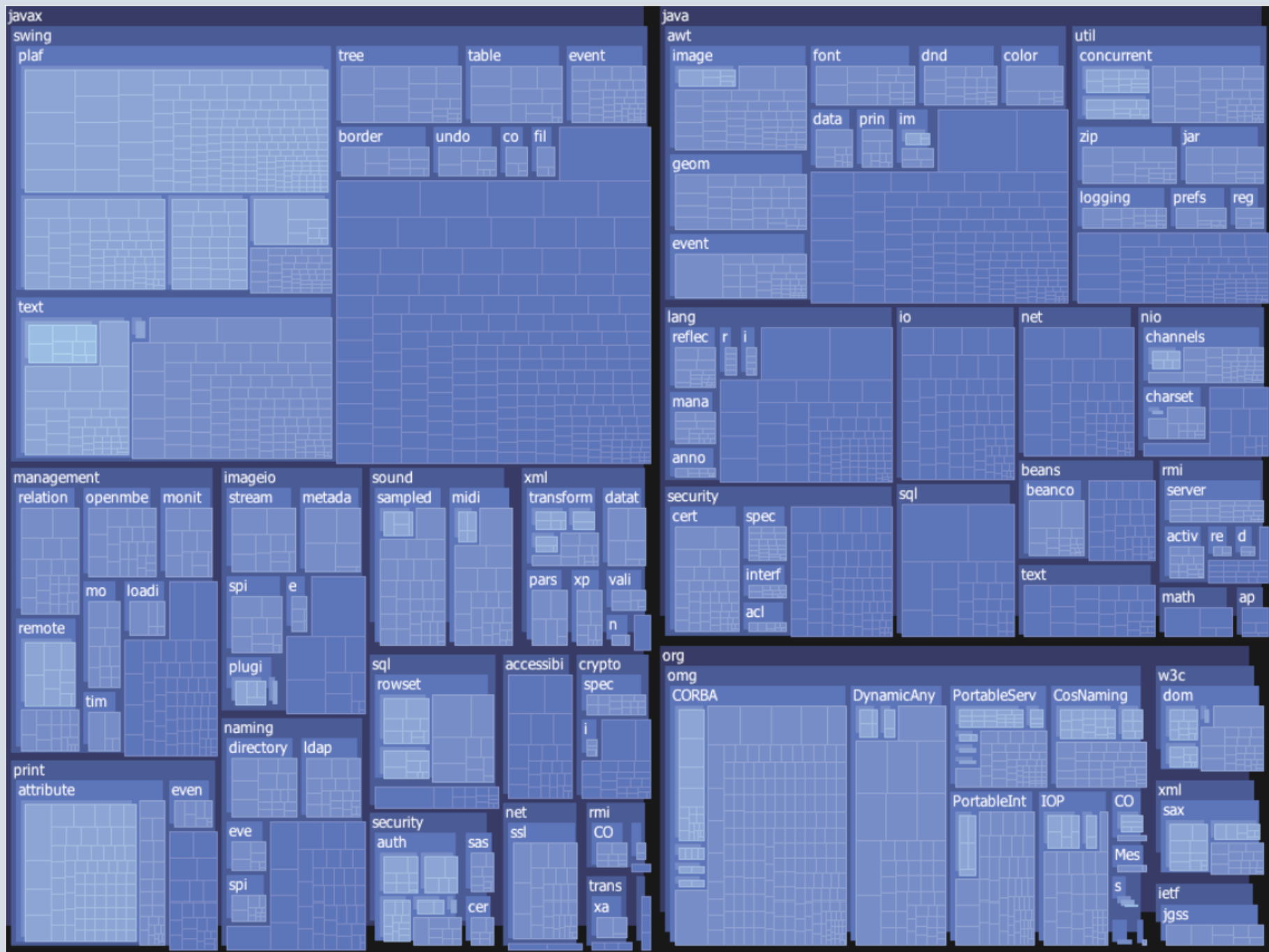




Slice and Dice (Nested)

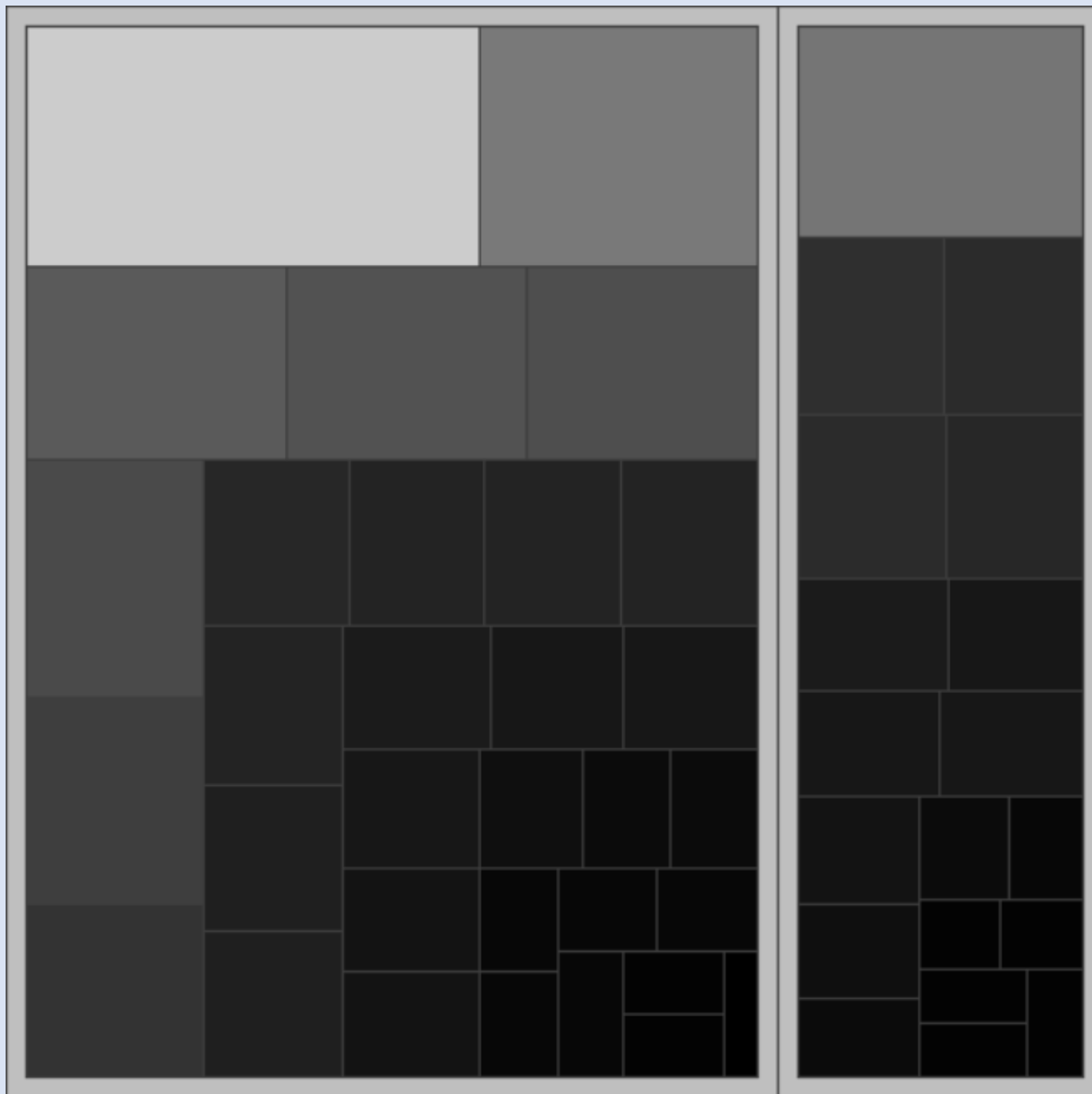
Partizione dello spazio alternata per ogni livello dell'albero: verticale e orizzontale

Uso dell'annidamento per mostrare la gerarchia



Cascaded (Nested)

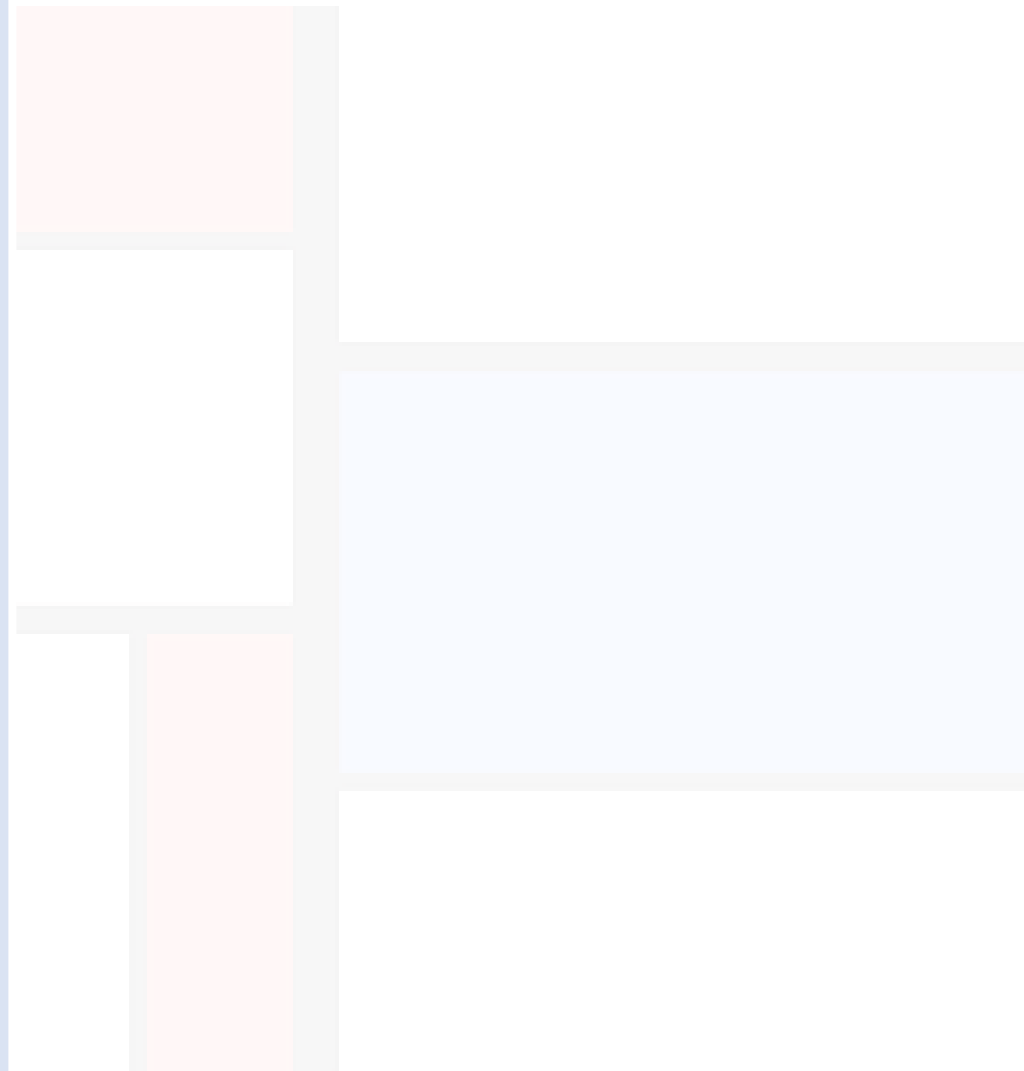
Uso della sovrapposizione (cascading) per mostrare la gerarchia



Unordered (Squarified)

I riquadri sono ordinati per dimensione, pertanto la mappa è non-ordinata se i dati hanno un proprio ordinamento naturale (indipendente dai valori)

Tipico esempio: gerarchie temporali



APPLICAZIONE: INQUINAMENTO

La gerarchia è il tempo (date delle rilevazioni)

Dataset originale lineare (formato CSV)

Principale problematica: gerarchizzazione del dataset

- Uso di script in Java per effettuare parsing di date e osservazioni e generare la gerarchia in JSON

Necessità di generare gerarchie diverse a seconda della visualizzazione utilizzata

- Esempio: le treemap non supportano i dati qualitativi

```
date,avgPM10,avgNO2,avgCO_8h,avgO3,avgC6H6,avgSO2,avgPM25,valutazione
01/01/2008,134,191.6666667,3.875,14,1.5,8,101,B
02/01/2008,141,144,3.5,31,1.5,2.5,133,B
03/01/2008,62.33333333,78.4,2.433333333,18.33333333,,2.5,48,B
04/01/2008,38.66666667,85.57142857,1.875,25.66666667,1.5,2.5,33,G
05/01/2008,42,128.1428571,2.68,11.33333333,4,2.5,41,G
06/01/2008,42,95.14285714,2.3,11,3,2.5,36,G
07/01/2008,30,74,1.98,17.66666667,1.5,2.5,25,G
08/01/2008,58.33333333,99.42857143,2.32,12.66666667,4,2.5,38,B
09/01/2008,55.33333333,102.625,2.14,13,3,2.5,41,B
10/01/2008,52,87.14285714,1.82,11.66666667,4,6,42,B
11/01/2008,125.5,179.7142857,2.8,18,11.1,15,94,B
12/01/2008,74,132.6666667,1.7,17,6.5,2.5,69,B
13/01/2008,25.33333333,77.71428571,1.425,14,3.25,2.5,22,G
14/01/2008,27.33333333,75.42857143,1.225,11.33333333,2.75,2.5,23,G
15/01/2008,55.33333333,110.1428571,1.725,11,5,2.5,52,B
16/01/2008,24.66666667,102.1428571,1.375,28.66666667,2.75,2.5,21,G
17/01/2008,25.66666667,93.28571429,1.35,22.66666667,2.25,2.5,23,G
18/01/2008,43.33333333,121.625,2.12,23.33333333,4.5,2.5,35,G
19/01/2008,68,123.25,2.08,24.66666667,7,7,55,B
20/01/2008,97,151,2.48,17.33333333,10,6,78,B
21/01/2008,86,116,2.12,11.66666667,7,2.5,52,B
22/01/2008,78,133.25,2.28,12,7.5,7,62,B
23/01/2008,50.66666667,91.75,1.6,69.66666667,4.5,6,40,B
24/01/2008,32.33333333,156.625,1.34,66.66666667,2.75,2.5,15,G
25/01/2008,56.66666667,145.5,1.44,17,5.5,2.5,35,B
```

```
boolean signalMore = true; // Indica se ci sono ancora righe da scorrere
boolean signalMoreM = true; // Indica se ci sono ancora righe da scorrere (mesi)
boolean signalMoreD = true; // Indica se ci sono ancora righe da scorrere (giorni)

System.out.print ("{\n\"name\": \"Years\", \"children\": {\n\"");

for (int yCount = 2008; signalMore; yCount++)
{
    signalMoreM = true;
    System.out.print ("{\n\"name\": \"\" + Integer.toString(yCount) + "\", \"children\": {\n\"");

    while (signalMoreM)
    {
        signalMoreD = true;
        System.out.print ("{\n\"name\": \"\" + monthParser(Integer.parseInt(rows[currentRow].substring(3,5))) + "\", \"children\": {\n\"");

        for (; signalMoreD; currentRow++) // NB: FORMATO DATA: DD/MM/AAAA // GIORNO: substring(8,2) // MESE: substring(3,5) //
            ANNO : substring(6,10)
            {
                tokenArr = rows[currentRow].split(",");

                System.out.print ("{\n\"name\": \"\" + tokenArr[0].substring(0,2) + "\", \"children\": {\n\"");

                // Controllo valori mancanti

                // Scrittura valori
                System.out.print("{ \"name\": \"PM10\": \"\" + NaIfEmpty(tokenArr[1]) + \"\",");
                System.out.print("{ \"name\": \"NO2\": \"\" + NaIfEmpty(tokenArr[2]) + \"\",");
                System.out.print("{ \"name\": \"CO\": \"\" + NaIfEmpty(tokenArr[3]) + \"\",");
                System.out.print("{ \"name\": \"O3\": \"\" + NaIfEmpty(tokenArr[4]) + \"\",");
                System.out.print("{ \"name\": \"C6H6\": \"\" + NaIfEmpty(tokenArr[5]) + \"\",");
                System.out.print("{ \"name\": \"SO2\": \"\" + NaIfEmpty(tokenArr[6]) + \"\",");
                System.out.print("{ \"name\": \"PM25\": \"\" + NaIfEmpty(tokenArr[7]) + \"\",");
                System.out.print("{ \"name\": \"Valutazione\": \"\" + NaIfEmpty(tokenArr[8].strip()) + \"\","); // L'ultima voce contiene newline che
                // vanno rimosse

                //DUMMY

                // lookahead: compara le date della stringa attuale con quella successiva e controlla se essa è lo stopcode
                // si appoggia al fatto che tutte le righe iniziano sempre con una data in un formato dato
                if (!stringEquals(rows[currentRow].substring(3,5), rows[currentRow+1].substring(3,5))) signalMoreD = false; // Fine del mese
                if (!stringEquals(rows[currentRow].substring(6,10), rows[currentRow+1].substring(6,10))) signalMoreM = false; // Fine dell'anno
                if (rows[currentRow+1].equals("TERMINATIONCODE")) signalMore = false; // Fine del file
            }
        }
    }
}
```

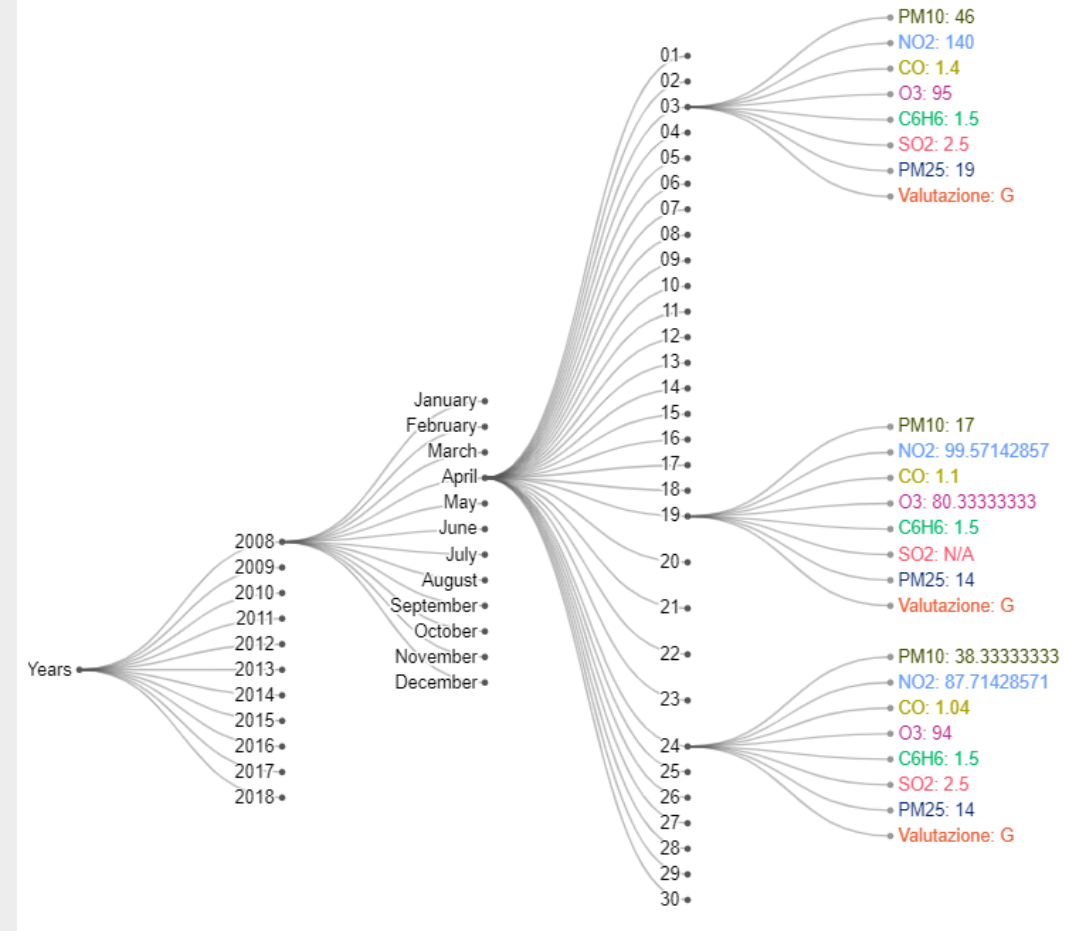

ALBERO ESPLORABILE

Contiene l'intero dataset

Foglie colorate per facilitare l'individuazione degli inquinanti

Interazione: permette di esaminare qualunque punto del dataset

Scopo: esplorazione del dataset



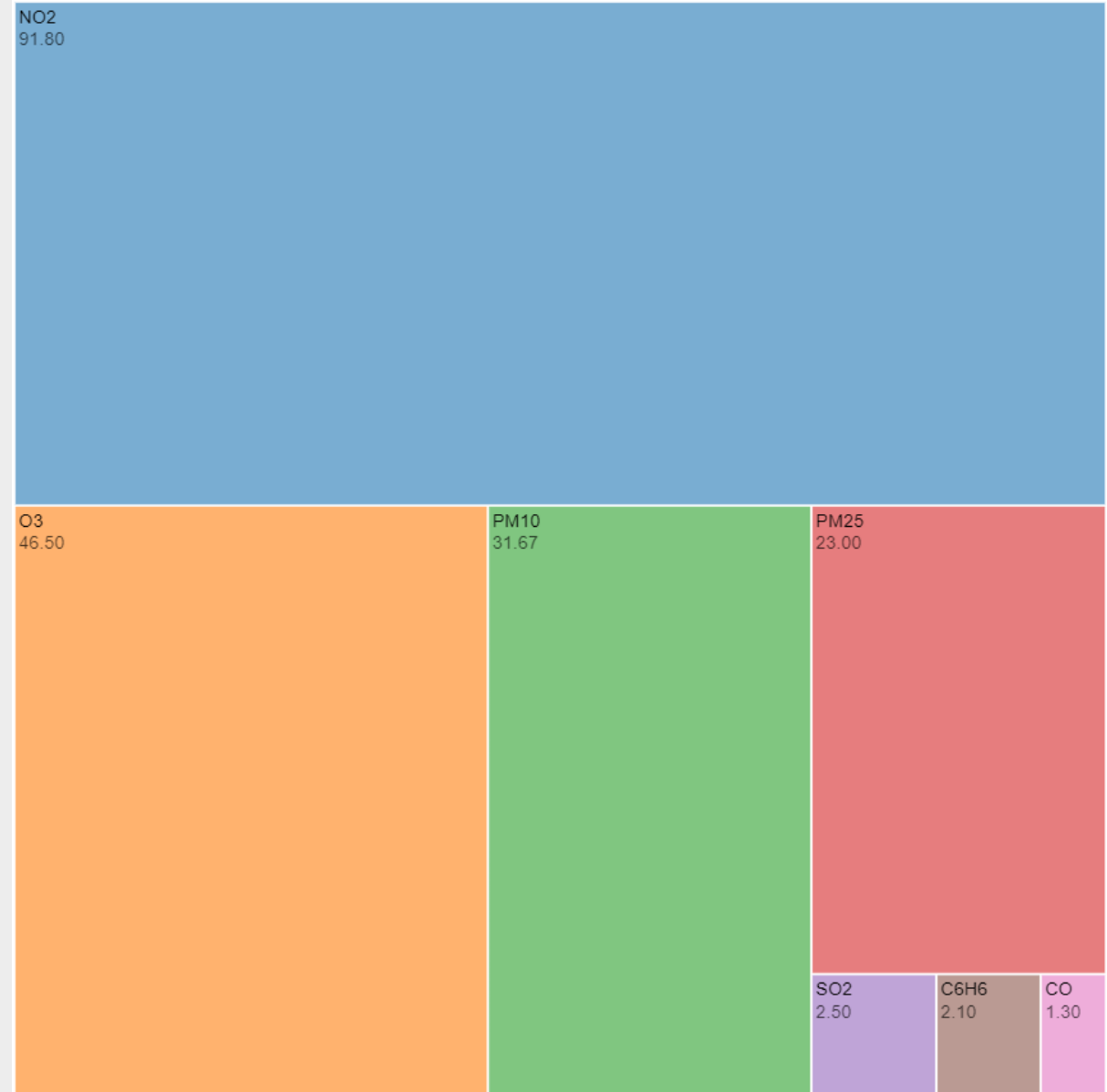
TREEMAP

Contiene un punto del dataset comprensivo di tutti gli inquinanti, ma non della valutazione, poiché essa non è quantitativa

Scarsa percezione dei trend

Scopo: Confrontare le proporzioni di inquinanti in un punto della gerarchia

Problema: eccessiva preponderanza del CO



ICICLE GRAPH

Contiene una porzione del dataset: la gerarchia e un inquinante

Parte della gerarchia è innestata a destra

Interazione: drill-down in una porzione della gerarchia

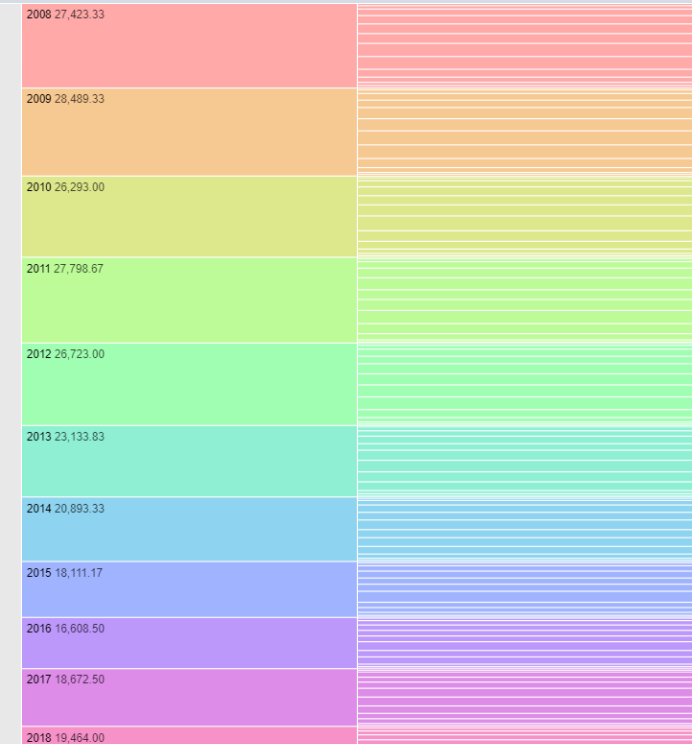
Scopo: visualizzazione dei trend relativamente alla gerarchia

PM10

Years 136,719.92



Years 253,610.67



Ozono



Riferimenti (in ordine di comparsa)

- ❖ <https://observablehq.com/@d3/tidy-tree>
- ❖ <https://observablehq.com/@d3/treemap>
- ❖ <https://www.cs.umd.edu/hcil/treemap-history>
- ❖ <https://newsmap-js.herokuapp.com>
- ❖ <https://finviz.com/map.ashx>
- ❖ <https://homes.cs.washington.edu/~jfogarty/publications/gi2008.pdf>
- ❖ <https://pdfs.semanticscholar.org/727a/91e5ec6af61c9eb9007b59cdfdd5e575038a.pdf>
- ❖ <https://qz.com/278645/these-beautiful-works-of-art-were-made-using-algorithms/>
- ❖ <http://www.cpnas.org/exhibitions/archive/treemaps.pdf>

