Similarity of Days

Describes the methods used to find similar days by walking through an example, and proposes 2 possible ways of minimizing differences.

Date [Total MW, Total CO2]

```
01-01-2012 [176310.0, 76255.74200000017]
01-02-2012 [202264.0, 88365.64000000019]
01-03-2012 [231864.0, 102525.63099999998]
01-04-2012 [231103.0, 102609.72299999998]
01-05-2012 [ 214403.0, 93889.53299999986]
01-06-2012 [ 211127.0, 93325.40600000002]
01-07-2012 [186409.0, 82403.01500000019]
01-08-2012 [186644.0, 83536.34099999994]
01-09-2012 [229211.0, 106546.84899999987]
01-10-2012 [246229.0, 111672.96000000012]
01-11-2012 [259398.0, 116822.04499999995]
01-12-2012 [265139.0, 119428.55900000002]
01-13-2012 [266393.0, 120153.58399999997]
01-14-2012 [228481.0, 103219.03800000006]
```

What are the two most similar days to 1-07-2012 in terms of Total MW and Total CO2?

The values for 01-07-2012 are 186409.0 MW and 82403.01500000019 tons of CO2

First, we find the two most similar days in terms of MW and CO2 separately:

For MW: Find the two values whose differences between 186409 are the smallest.
If we have an ordered list of MW, this is as easy as reading off the two next on the list, which here are 186644.0 and 176310, corresponding to 01-08-2012 and 01-01-2012.

We found this out by comparing the absolute differences, which were 235 and 10099.

For CO2: Find the two values whose differences between 82403 are the smallest,
which here are 83536.34 and 88365.64,
corresponding to 01-08-2012 and 01-02-2012.

We found this out by finding the absolute differences, which were 1133.34 and 5962.64.

We know the differences, but how surprising is it that the differences happened to be those values? Are these values abnormally close to eachother? How can we compare what a difference of 235 in MW means for a day compared to what a difference of 1133.34 means for a different parameter?

Mean difference is:
E[X-Y], where X and Y are two variables.

Using this, we can define an expected difference for our data sets:

For MW: E[| 186409.0 - Y |]
where x was substituted for our day's value of MW, and Y is a random variable chosen from the values of all other days.

This simplifies to 186409.0 - E[Y]

The expectation of our Y random variable is the average of all points in the original dataset, excluding 186409.0, which in this example is 226812.77

The expression becomes | 186409.0 - 226812.77 |
and our expected difference is 40403.77

For CO2: E[| 882403 - Y |]
where x was substituted for our value of CO2, and Y is a random variable chosen from the values of all other days

This simplifies to 82403 - E[Y]

The expectation of our Y random variable is the average of all points in the original dataset, excluding 82493, which in this example is 101411.6

The expression becomes | 82493 - 101411.6 |
and our expected difference is 18918.6

Now we know our expected differences, which we can combine as a coordinate pair: (40403.77, 18918.6), after which which can compare the distance from this pair to the pairs of actual differences we find among our values and take the minimum of that.
This can be done in n dimensions with a generalized distance formula.

_____

Another way (much easier for n dimensions) that seems intuitive is to
find the proportion of our actual difference and our expected difference.
We do so by finding: difference/expected difference for each variable dimension.

- A value greater than 1 means our difference is larger than expected.
- A value of exactly 1 means our difference is exactly as we expect (it's average).
- As our value approaches 0, it means our difference is increasingly better than we expect.

The only time the value is 0 is when the actual difference is 0, meaning the parameter values are identical.

Now it is easy to calculate this proportion for every variable in question(like CO2, cost, MW...), adding them up for each variable and find the day with the lowest sum.
This is our most similar day in n dimensions.

Efficient Implementation

How can this be implemented such that we minimize the amount of times we have to search through the entire data set to perform operations on each element?
Possible improvements:

-Use rolling sum to calculate average.
-Use tree data structure for efficient
-If we only want to compare x most similar days, only need to perform the entire method on x days for each variable.