Orthogonal Matrices

1. Which transformation matricies are orthoganal?

Rotation matricies and reflection matrices

2. Explain the result that the row vectors and column vectors of A are *orthonormal*.

Because A is orthogonal, the product $A * A^T$ is the identity matrix. For any two different rows m and n of A, observe that the transpose operation converts those two rows into columns of A^T. As a result, position (m, n) of the product above must be the dot product of the two input rows (one of which has been converted to a column). Because $m \neq n$, the properties of the identity matrix guaruntee that this dot product is zero, meaning the two original row vectors were perpendicular. The same result can be had for column vectors by considering $A^T * A$.

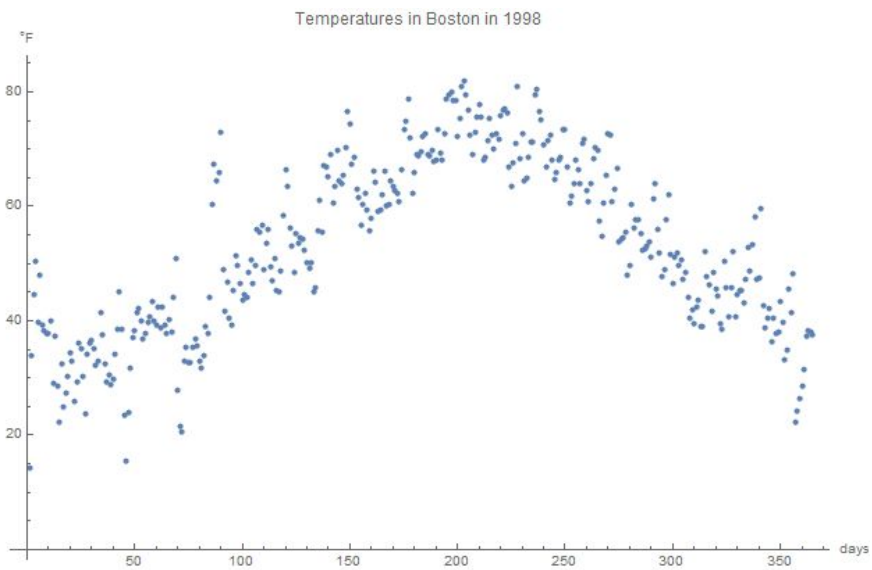
3. Determinants of Orthoganal matrices

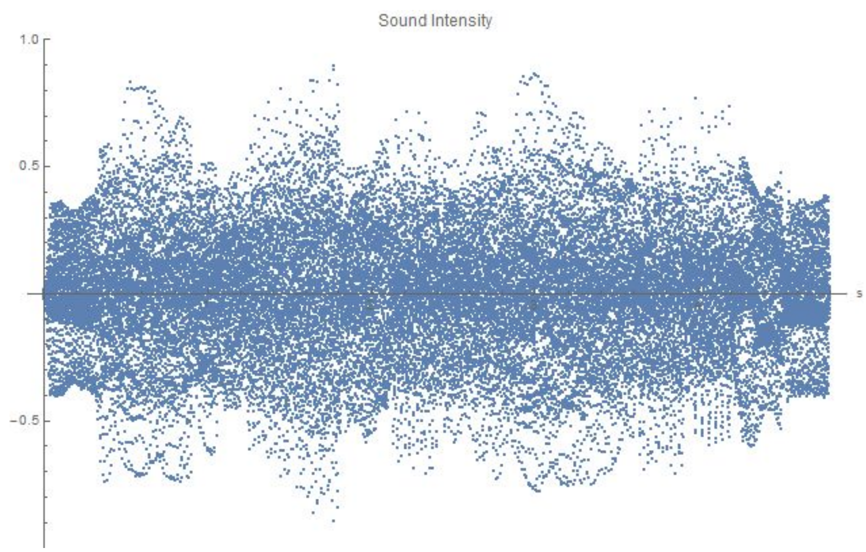
As previously established, $det(A) = det(A^T)$ and $det(A) = \frac{1}{det(A^{-1})}$, so for orthogonal matrices, $det(A) = \frac{1}{det(A)}$ so $det(A)^2 = 1 \Rightarrow det(A) = \pm 1$.

If $det(A) = 1$, then the matrix is a rotation matrix.
If $det(A) = -1$, the matrix represents a reflection.

Vectors and Matricies as Data

1. Imports

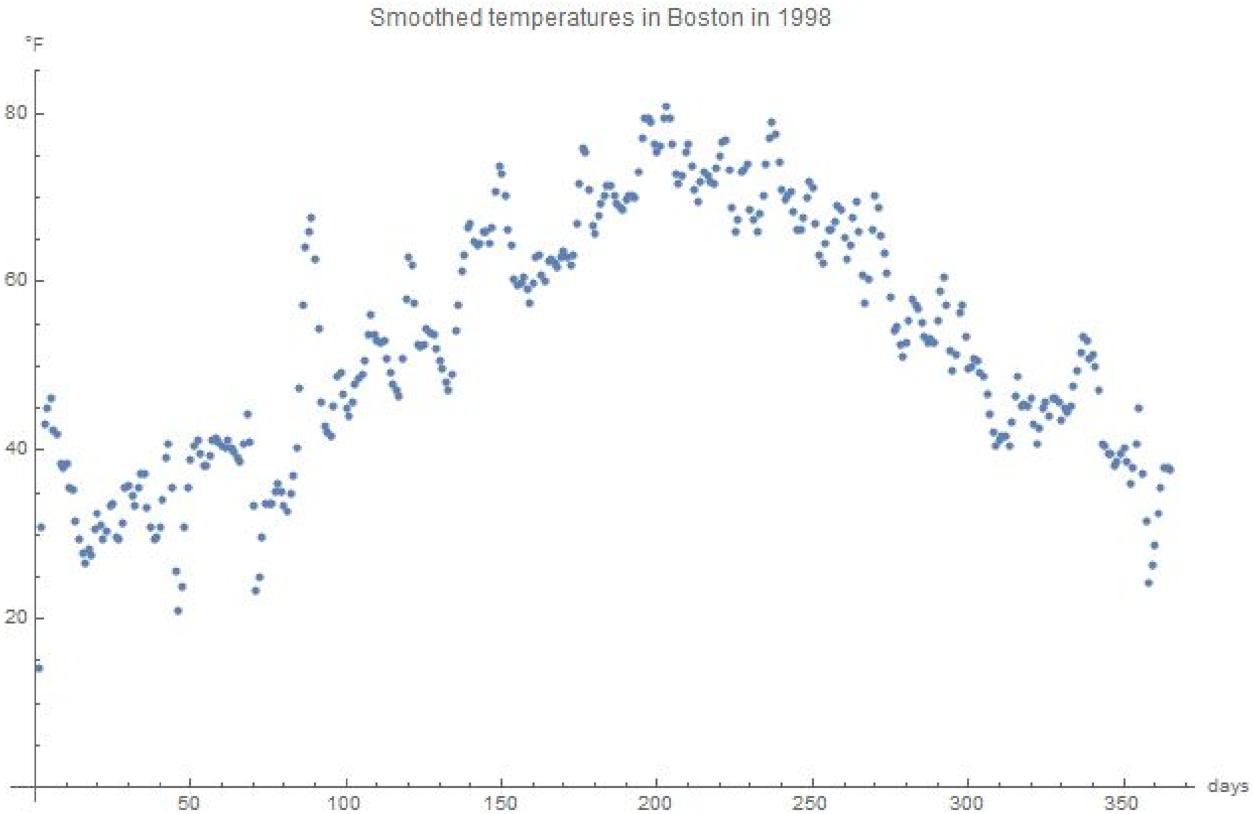




3. Manual kernel

Apart from the first and last elements, which stay the same, each element in the output array becomes the average of the three elements surrounding it in the input. This resembles a "blur" operation. A naive calculation requires n^3 calculations, but sparse matrix implementations are better.

4. Implementation



5. Kernels 1

The kernel [⅓, ⅓, ⅓] accomplishes this, and requires the same O(3n) mults as a sparse matrix implementation, but less than naive.

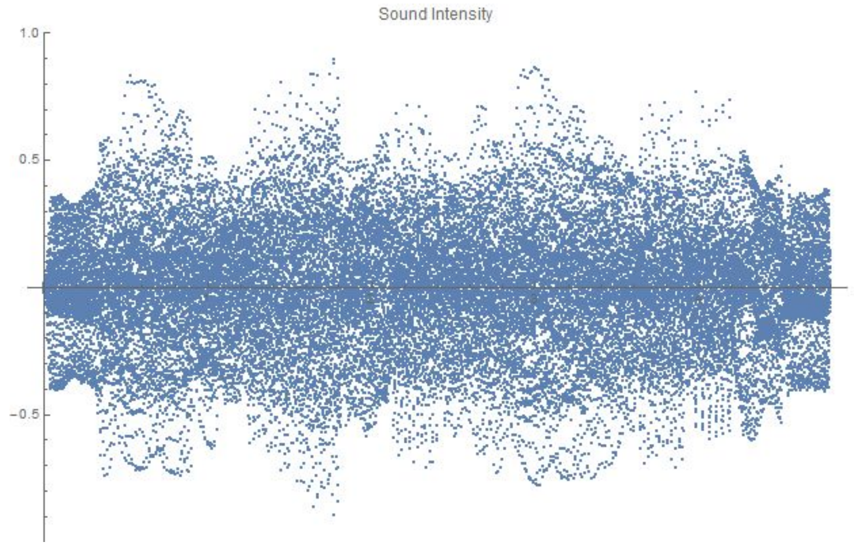
6. Sample kernels

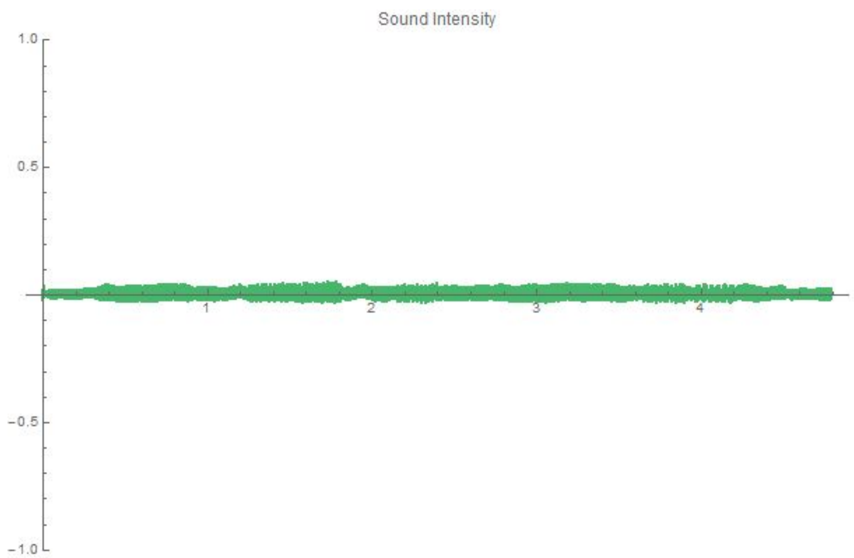
Kernel d approximates the derivative of the data in the input signal by highlighting rapid changes.
Kernel e approximates the second derivitive of the input data by finding cases where a piece of data is higher or lower than the average of its neighbors.

Convolution

8. Smoothed audio

Applying a kernel like the above decreases the overall amplitude of the signal, and also filters for low frequencies, especially when a longer kernel is used.





9. Edge filter

I was suprised to find that the sound file still retained the same recognizable melody after the convolution / kernel operation was applied. Higher notes had a slightly more pronounced nature, and there was a “tininess” added on top. I hypothesize that comes from accentuating very high frequency noise in the input signal.

11. Low pass filter

The moving average here is expected to behave as a lowpass filter, and amplify the lower notes.

12. Echo

The kernel for this looks something like

[1,0,0,0,.....,½]

The length determines the echo time delay.

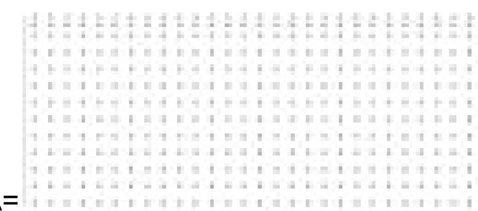
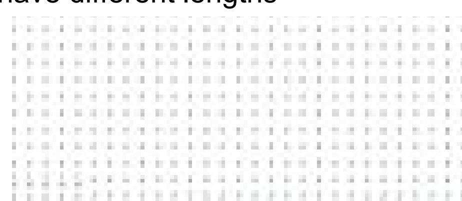
To get a true “multiple echo” behaviour, the kernel should take the form

[1,0,0,0,.....,½,0,0,....,¼ ...]

13. Monthly temperatures

This operation is not possible with a kernel or convolution because...

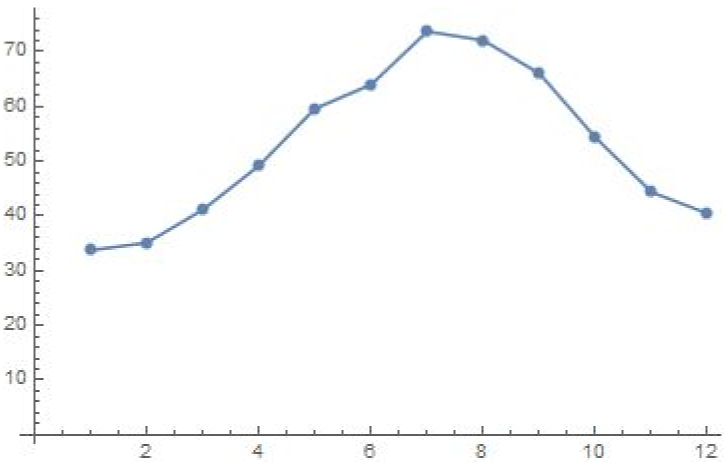
1. The length of the output is not equal(ish) to the length of the input
2. Different months have different lengths

A=  ... 

Code:

```
dates =
 QuantityMagnitude@
  Differences[DateRange[DateObject[{1998, 1, 1}],
    DateObject[{1999, 1, 1}], "Month"]]
acc = Prepend[Accumulate[dates], 0]
bands = Table[Band[{acc[[i]] + 1, i}, {acc[[i + 1]], i}, {1, 0}], {i, 1, Length@acc - 1}]
Transpose@SparseArray[bands → 1 / dates]
```

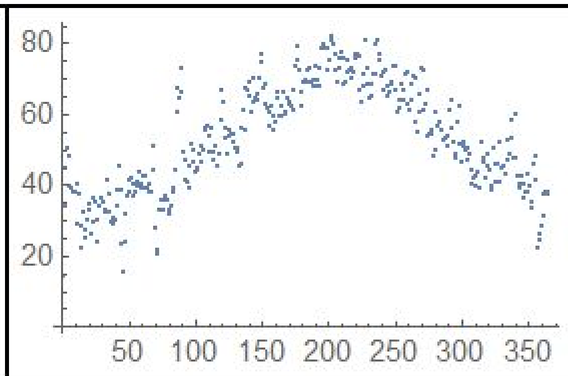
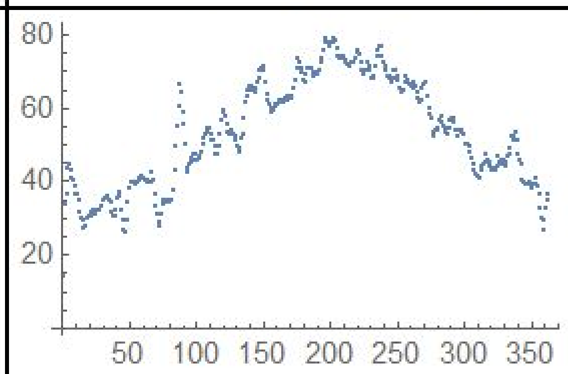
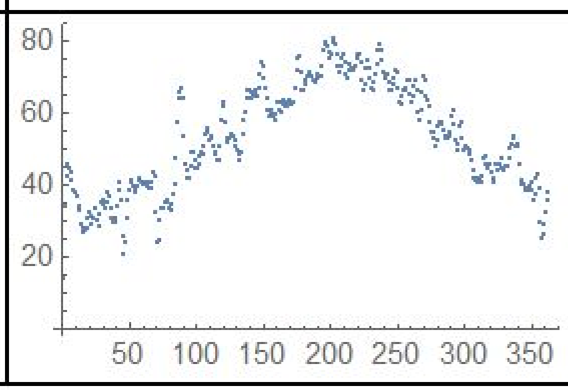
Result:



14. Moving Average

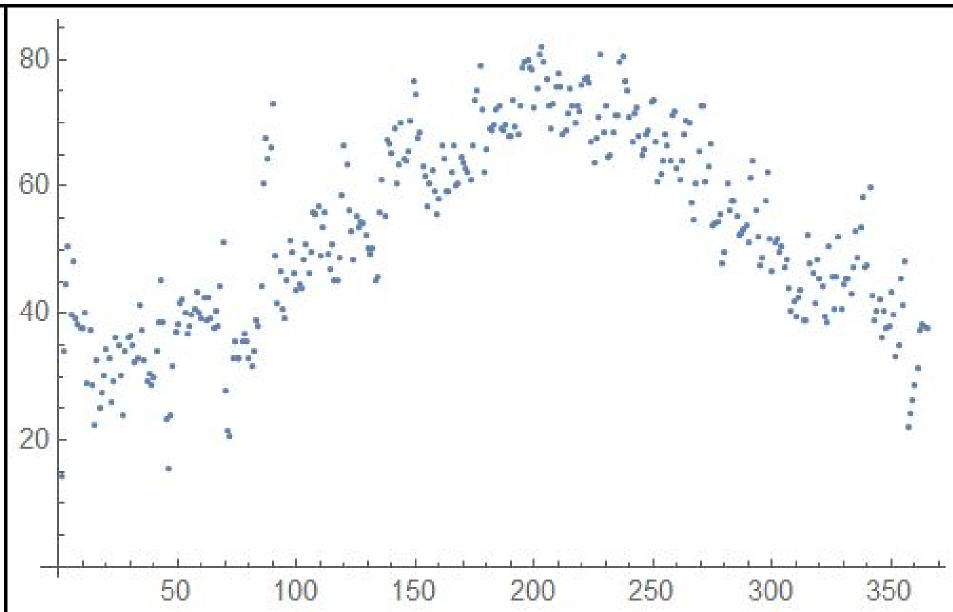
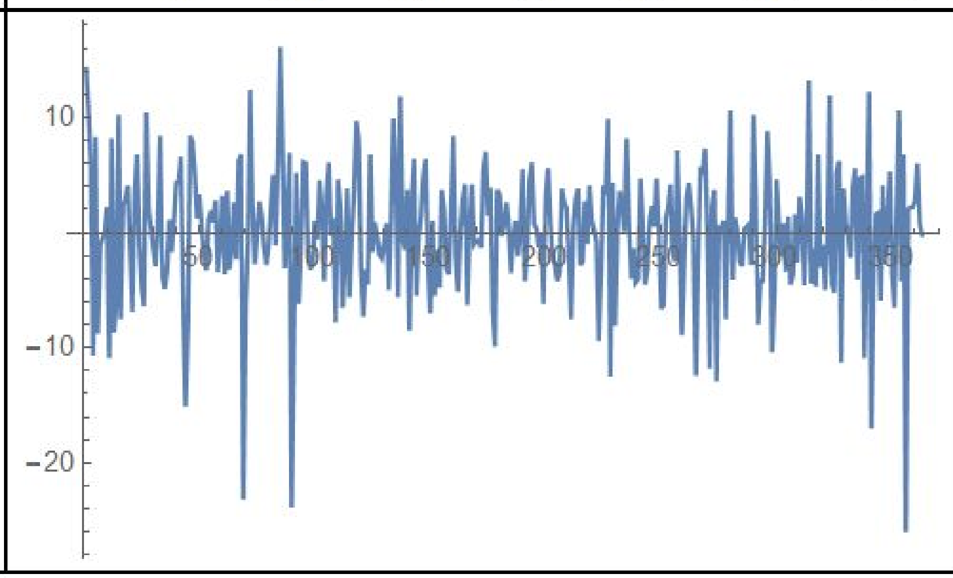
$$k = \begin{pmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{pmatrix}$$

15. Alternative kernel

Original	
Moving Average	
Distribution Kernel	

The moving average removes much of the noise in the input signal, making it follow closer to the seasonal trendlines. The weighted moving average of problem 15 does much the same thing, but less so. Also, the downward spike of 1 point around day 50 is transformed to a smaller spike of ~6 points by the moving average, but shortened without dragging in other points when using the weighted convolution kernel.

16. Temperature edge detect

Original	
Edge Detect	

Large upward spikes indicate days after which the temperature will rise significantly, and the opposite for downward spikes.