Enter the Matrix with Coding

Tim Chartier

Department of Math & CS

DAVIDSON

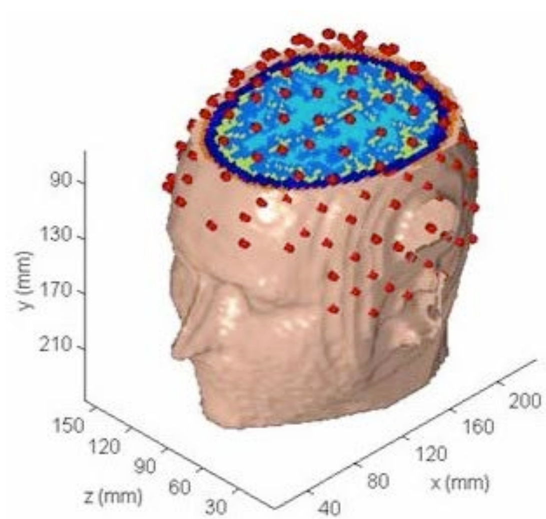
who am I?

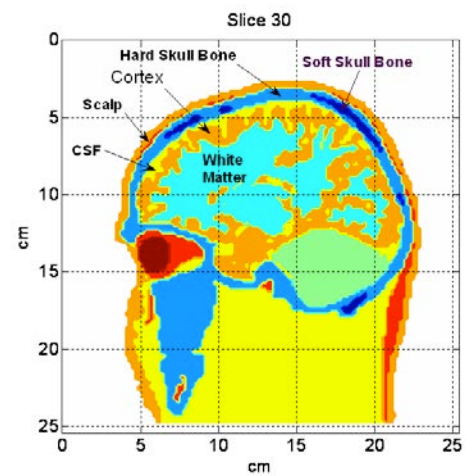
step 1 – linear

$$A\mathbf{x} = \mathbf{b}$$

getting ahead

EEG/EMG data of the human head created linear system with over 1.5 million variables.





- Accurately solved in under 30 sec (on laptop)
- 20 to 66 times faster than previous methods.

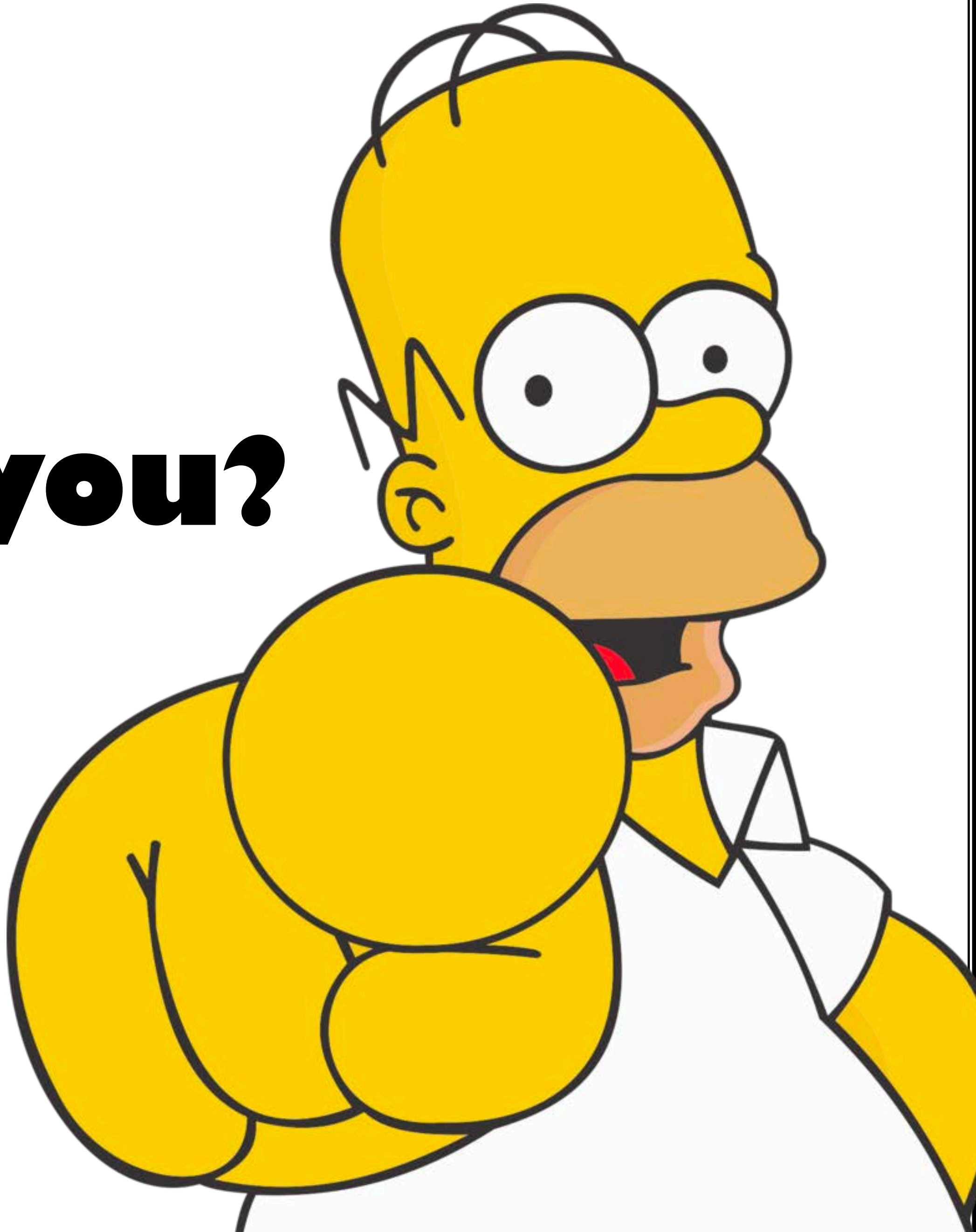
step 2 – ranking

step 3 – data science

'CATS STATS
DAVIDSON COLLEGE

tresata

Who are you?



Ranking

- What's your favorite? Who's number 1? Each of these questions involves ranking.
- In this course, we will learn these methods, how to code them, and how to adapt datasets so items can be ranked.
- For the final project, students will pick an application to rank, perform the ranking, analyze the results and make conclusions about the strengths and weaknesses of the results.


What lies ahead?

Winning percentage

- A common way to rank is with winning percentage.
- Many professional sports in the United States use this to determine who makes the playoffs.
- The best teams in the NBA, for instance, make the playoffs.

Easy but dangerous

- Winning percentage is easy to compute
  (number of wins)/(total number of games)
- But, it assumes parity among the teams.
- If you switch from the NBA to NCAA college basketball, for instance, this isn't true at all.
- Winning percentage isn't sufficient for March Madness predictions, for example.

Rank with Matrices

- Due to this, we will be using linear algebra, which you've seen in the form of system of linear equations like:

$$\begin{array}{rcrcrcr} 3x & + & 2y & - & z & = & 8 \\ x & + & y & + & z & = & 2 \\ 2x & + & y & - & z & = & 5 \end{array}$$

- For us, the unknowns are the ratings of the teams.

Entering the Matrix

- Rather than write a system like:

$$\begin{cases} 3x + 2y - z = 8 \\ x + y + z = 2 \\ 2x + y - z = 5 \end{cases}$$

- We will write it in matrix form:

$$\left[\begin{array}{ccc|c} 3 & 2 & -1 & 8 \\ 1 & 1 & 1 & 2 \\ 2 & 1 & -1 & 5 \end{array}\right]$$

Entering the Matrix

- While it’s important to know how many parts of linear algebra work, we will need programming languages to do the computations.
- We’ll be working in python.

2 do

- For the next session's assignment, you'll get ready to program.
- So, you'll need to get your programming environment ready.

python

- You may want to use Anaconda which bundles everything you need into one tidy download.
- Visit: https://www.anaconda.com/products/individual
- You could also use online version of python such as: https://cocalc.com/

Other options

- Other programming languages, like R, are certainly possible.
- I have less familiarity with python than Octave/Matlab.
- The slides and codes I offer will be in Octave and/or python.

?

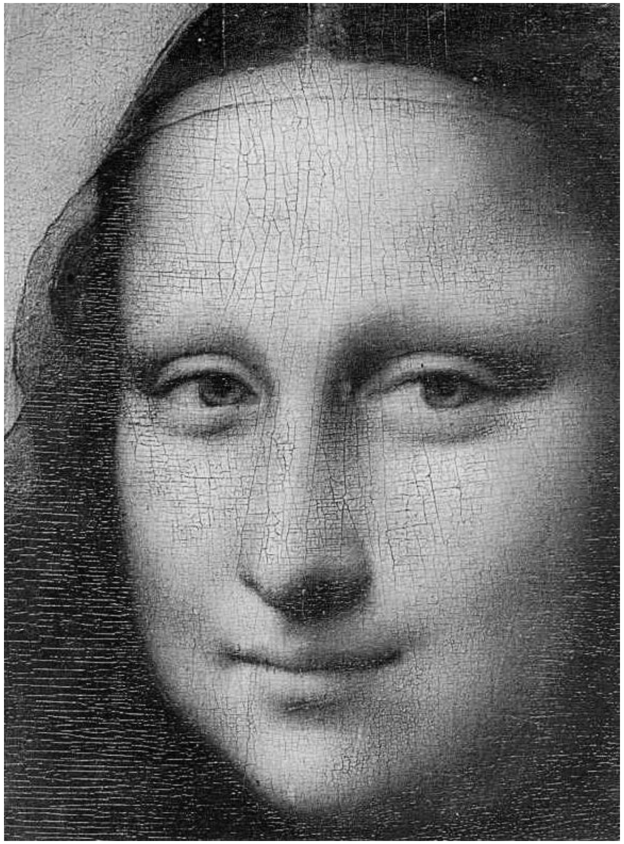
on programming

Linear intro

- Let’s introduce some fundamental ideas of linear algebra today.
- These ideas can play a fundamental role in your research to come so having them here can be handy.
- We’ll use images, since they can be read in or converted into matrices.

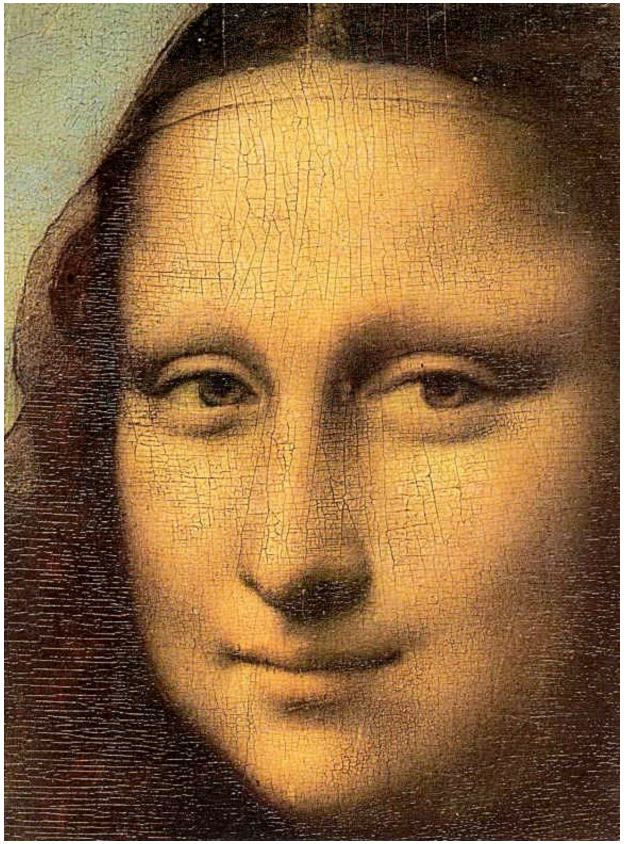
shades of gray

- Grayscale images are stored in matrices containing integers from 0 (black) to 255 (white).
- So, a 600 (high) by 400 (wide) image would be stored in a 600x400 matrix.



in living color

- Color images store color in 3 matrices for the red, green and blue channels.
- So, a 600 (high) by 400 (wide) image would be stored in a 600x400x3 matrix.



enter the matrix

The power of matrices comes in the math that can be done with them.

matrix addition

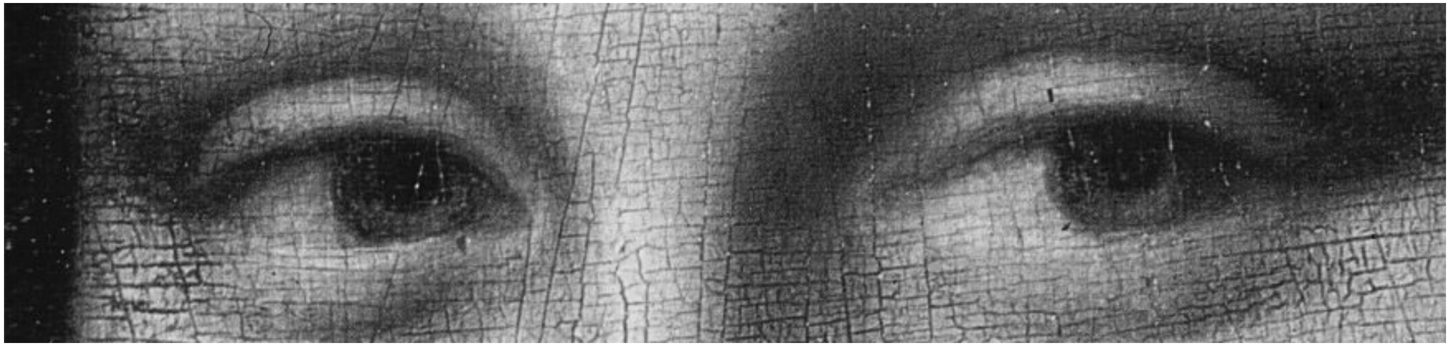
- Matrix addition is so easy it could almost become an afterthought.
- What a perfect place to show an application!

$$\begin{pmatrix} 1 & 2 & 3 \\ 6 & -1 & 4 \end{pmatrix} + \begin{pmatrix} 0 & 4 & 8 \\ -5 & 1 & 2 \end{pmatrix}$$

$$= \begin{pmatrix} 1+0 & 2+4 & 3+8 \\ 6-5 & -1+1 & 4+2 \end{pmatrix} = \begin{pmatrix} 1 & 6 & 11 \\ 1 & 0 & 6 \end{pmatrix}$$

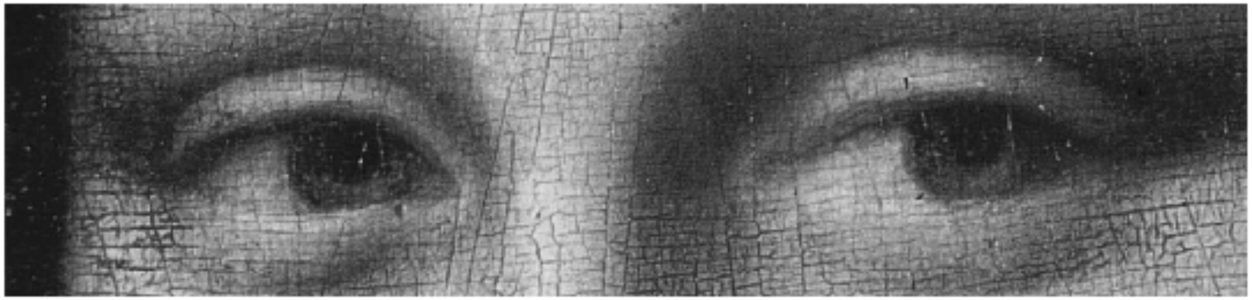
matrix dimmer

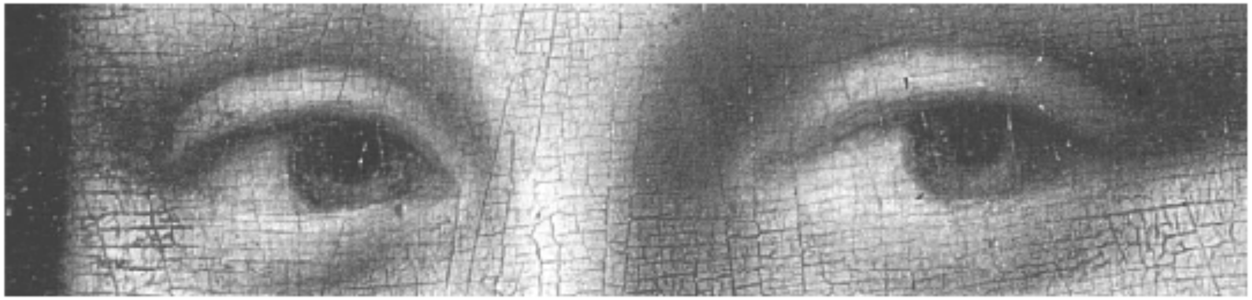
- This simple process can lighten an image.
- Let’s use the matrix of grayscale values for the following image of Mona Lisa’s eyes:



brighten your day

Suppose we create another matrix of the same size where every entry is 40. If we add this to the Mona Lisa matrix, what happens?





scalar multiplication

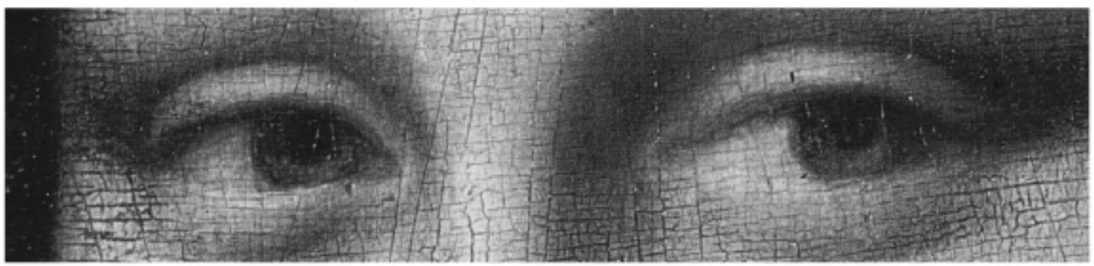
We can multiply a single number, called a scalar, by a matrix. For instance,

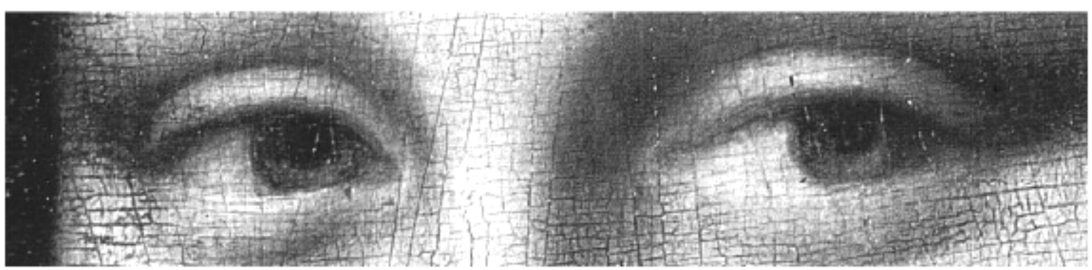
$$3\begin{pmatrix} 1 & 2 & 3 \\ 6 & -1 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 9 \\ 18 & -3 & 12 \end{pmatrix}$$

Again, easy enough. We almost immediately ask, "What can we do with it?"

lighten up

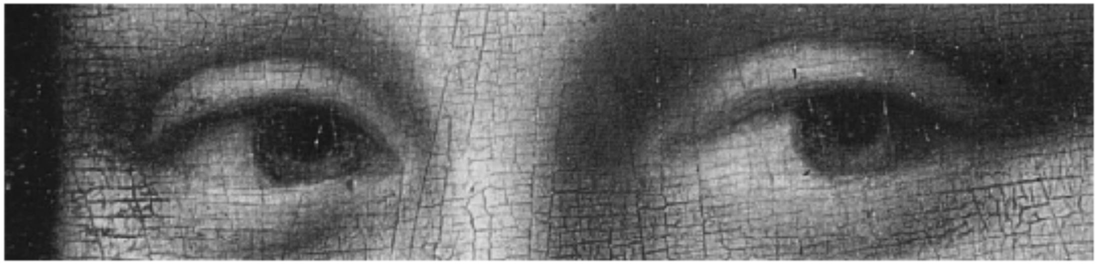
We can use scalar multiplication to lighten or dim an image. Consider multiplying the matrix of pixels for Mona Lisa by 1.4. Then, we get:

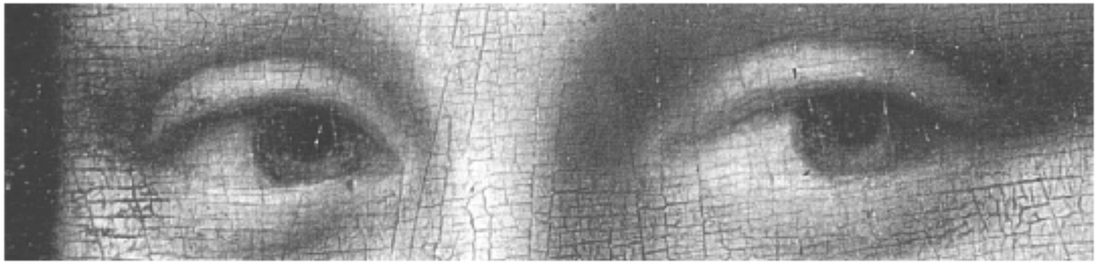


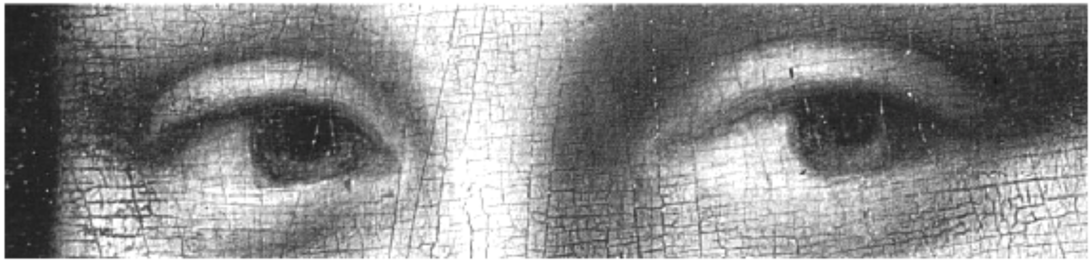


What's the difference?

What differences do you see in these?







drawing the line

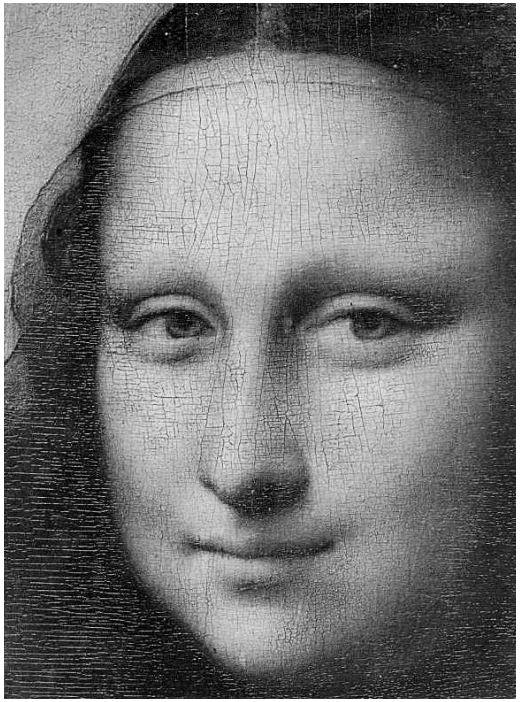
- We can also invert the colors of a grayscale image.
- A white pixel (255) becomes black (0).
- And, black (0) becomes white (255).
- Want the points (255,0) (0,255), which corresponds to the line:

$$y = -x+255$$

invert

Let *M* be the Mona Lisa and **255** the matrix with every element equaling 255, then we want

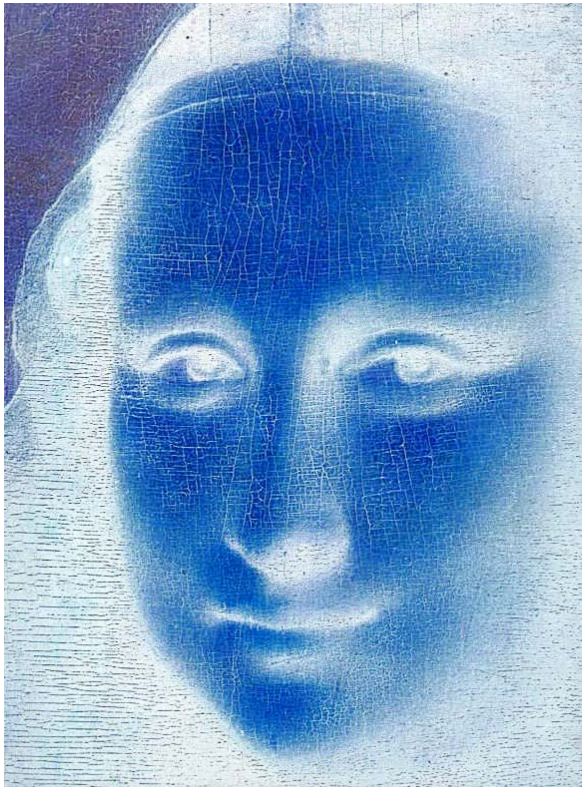
$$G = -M + \mathbf{255}.$$



Your Turn

Try it yourself (in color) at:

https://lifeislinear.davidson.edu/inversion.html



FOCUS

- How do you think this is coded?
- Any improvements?

Code it!

Let’s see this in python code.

? on programming

Submatrix

If we have

$$\begin{pmatrix} 5 & 10 & 8 & 10 & 10 \\ 8 & 4 & 5 & 9 & 1 \\ 4 & 2 & 5 & 7 & 9 \\ 2 & 9 & 9 & 5 & 7 \\ 2 & 9 & 2 & 9 & 5 \end{pmatrix}$$

then a submatrix is: $\begin{pmatrix} 5 & 7 \\ 9 & 5 \end{pmatrix}$

Diagon Alley

And? In linear you'd start with

$$\left(\begin{array}{rrr|r} 2 & 1 & -1 & 1 \\ 1 & -2 & 1 & 3 \\ 2 & 1 & -2 & 2 \end{array}\right)$$

And use row reduction to get:

$$\left(\begin{array}{rrr|r} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -2 \end{array}\right)$$

What's the connection?

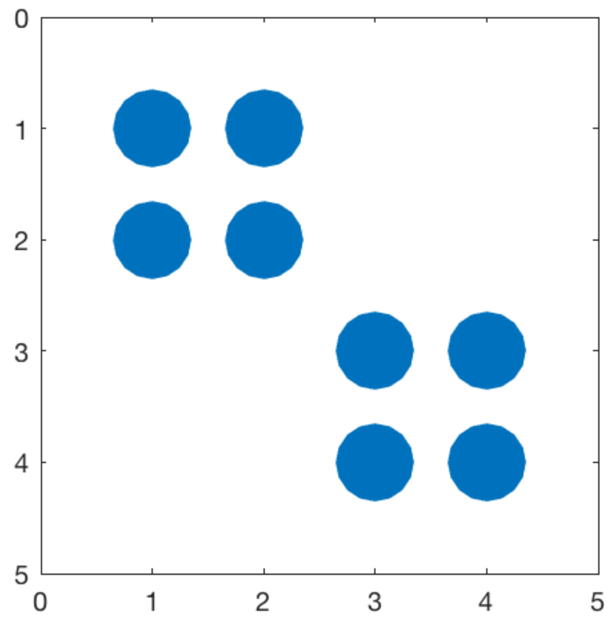
Block party

In linear, a 4x4 could seem like a recipe for hand cramps. What about this system?

$$\left(\begin{array}{cccc|c} 1 & 2 & 0 & 0 & 1 \\ 3 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 1 & 3 \\ 0 & 0 & 2 & 3 & 1 \end{array}\right)$$

Spying

If we make a spy plot of A, for Ax = b, we find:



On and off

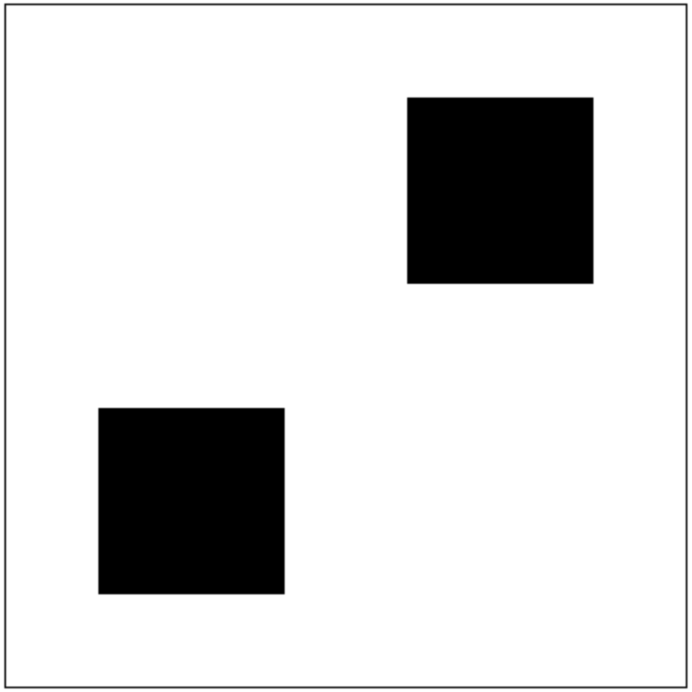
Consider the following matrix:

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Bit of graphics

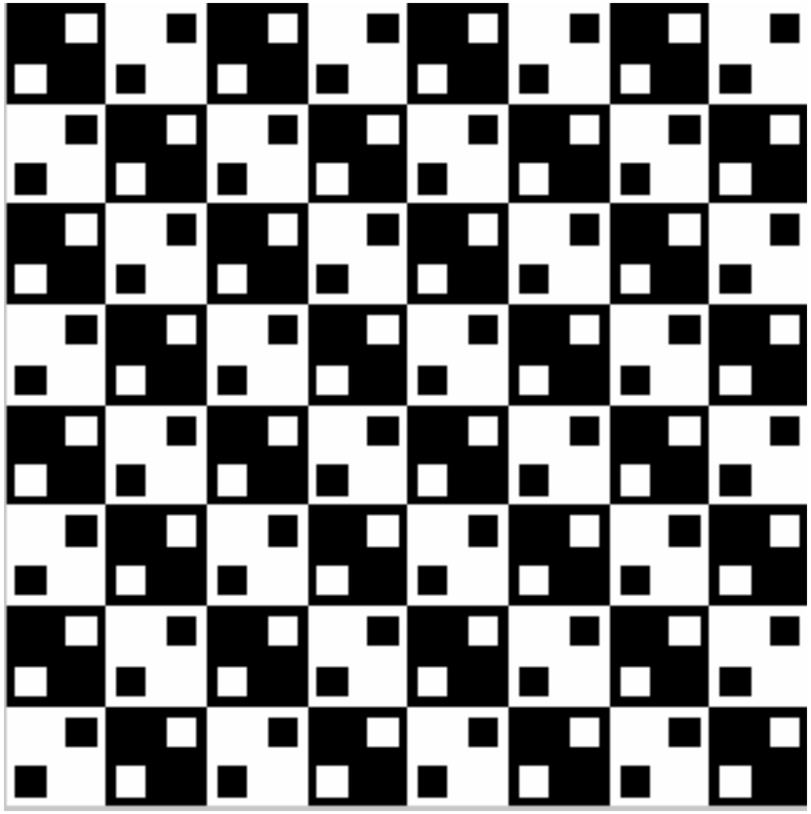
Let's visualize where 0 is white and 1 is black.

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$



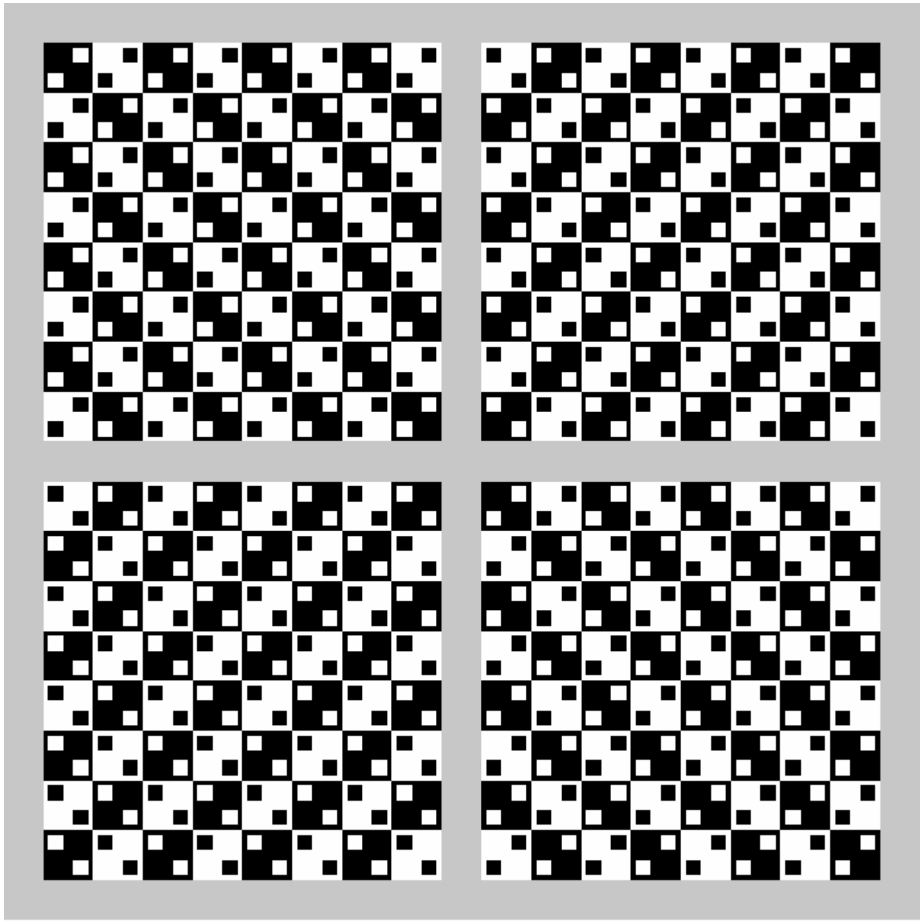
Visualize structure

What structure do you see now?



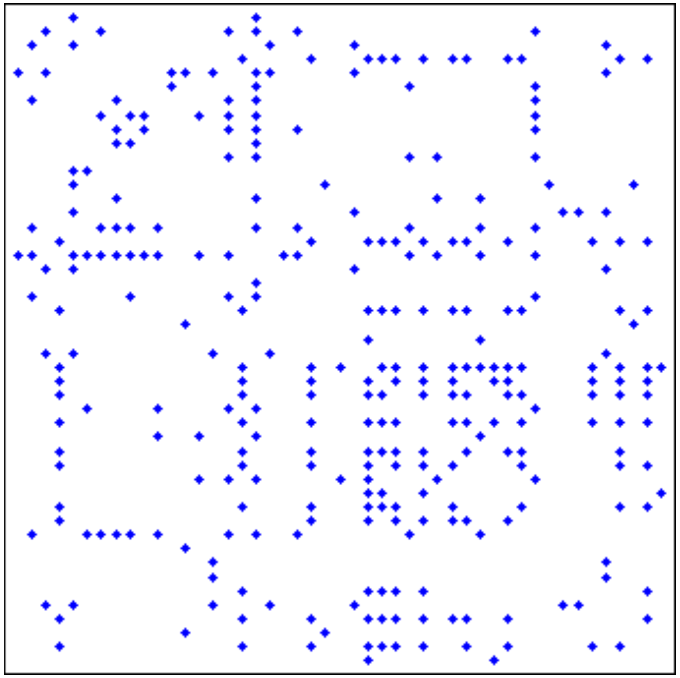
Mathematically bent

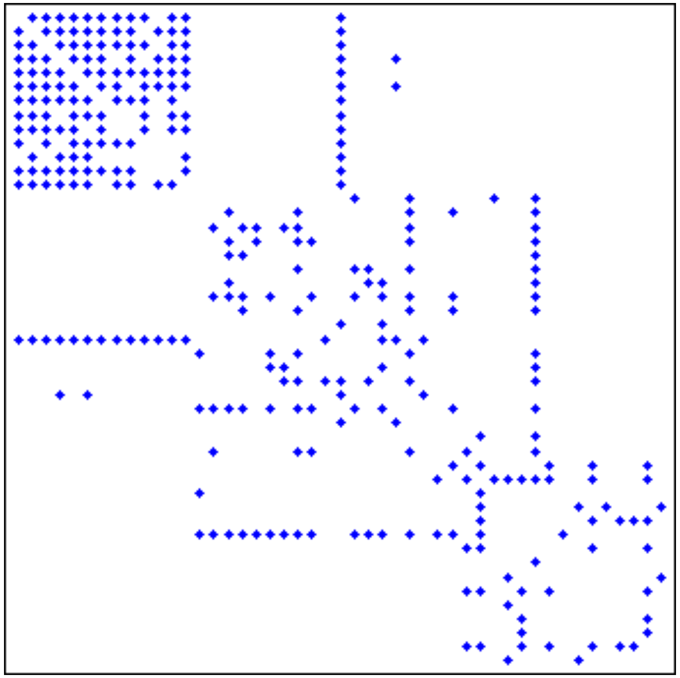
Let’s look at 4 matrices.



matrix friends

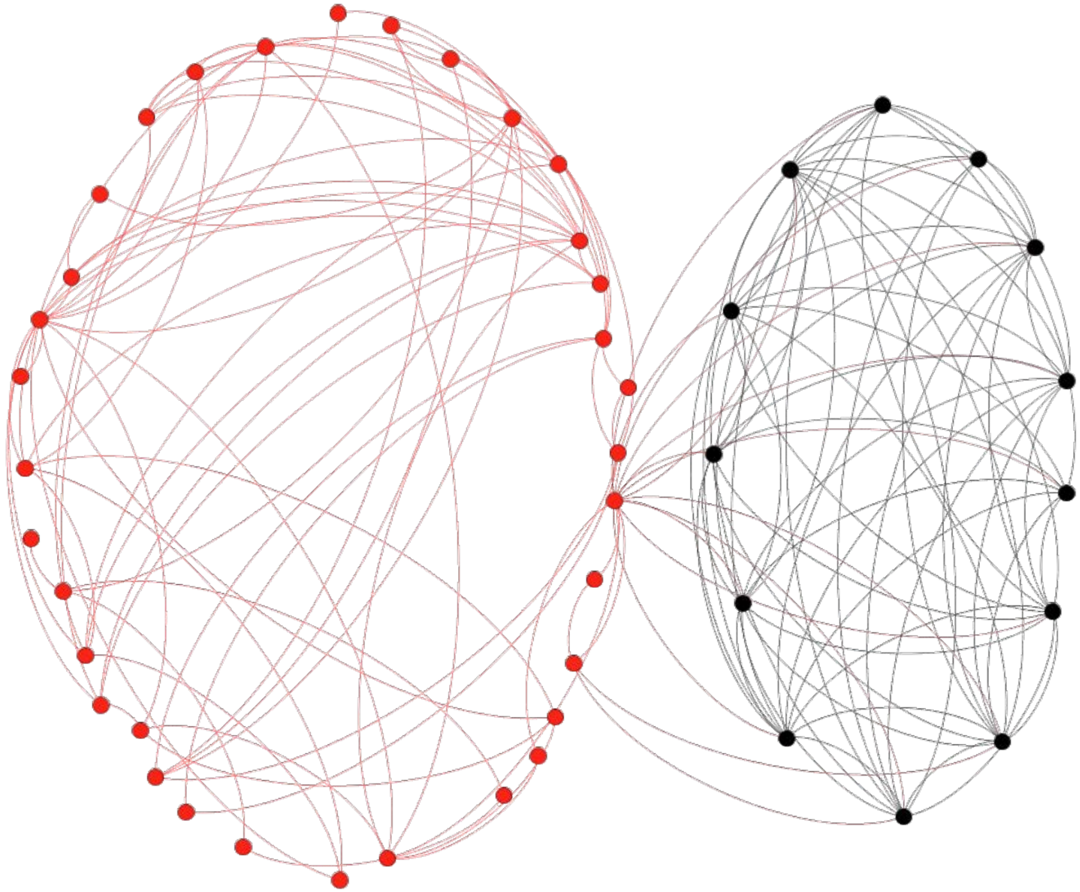
Let's create a spy plot of the Facebook adjacency matrix. Notice how the reordering gives us information.





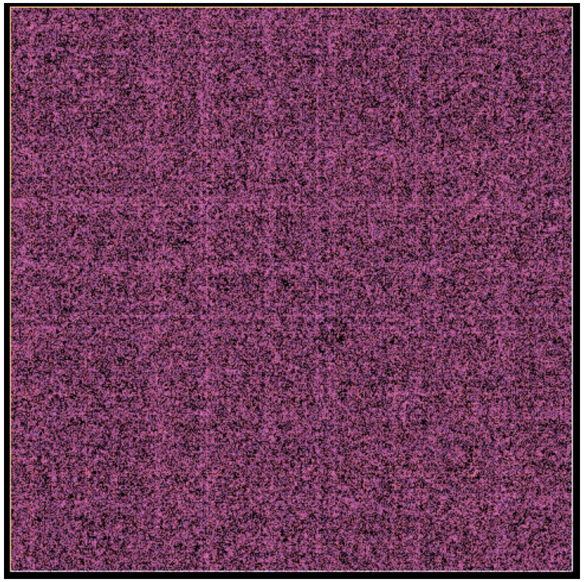
groups of friends

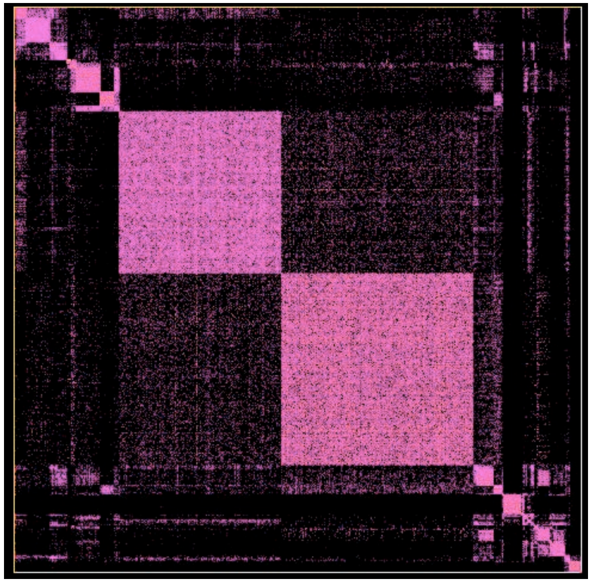
Reordering the graph uncovers a similar dynamic.



Blockbuster movie

A similar idea can be adapted to movies.



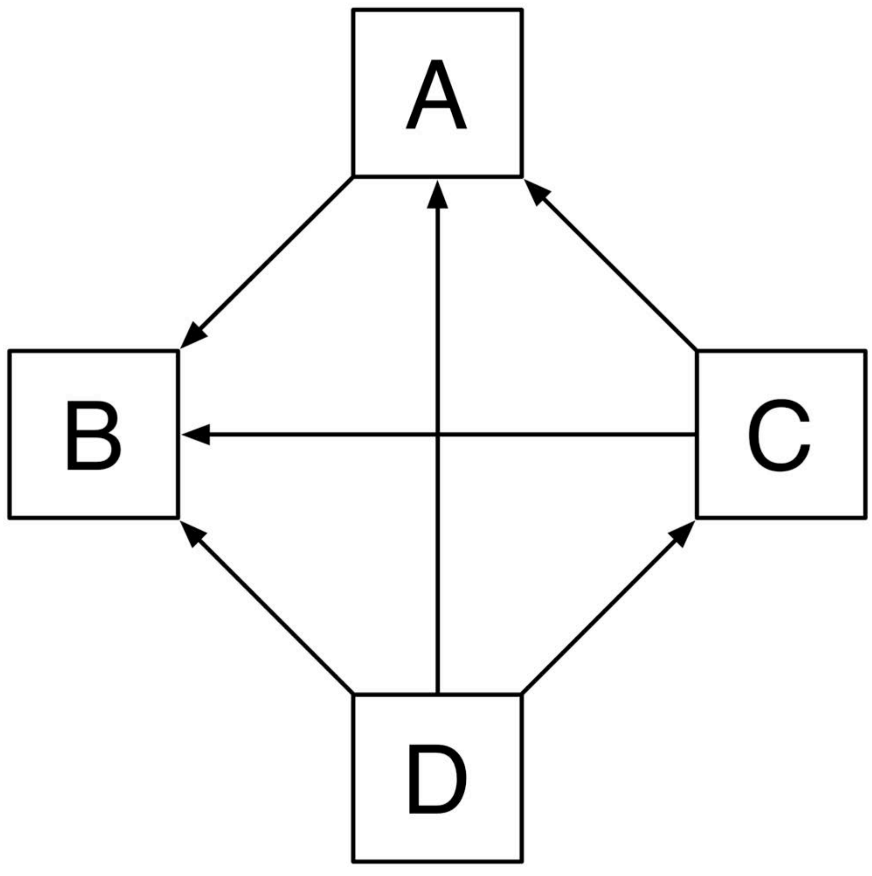


NETFLIX

Reordering

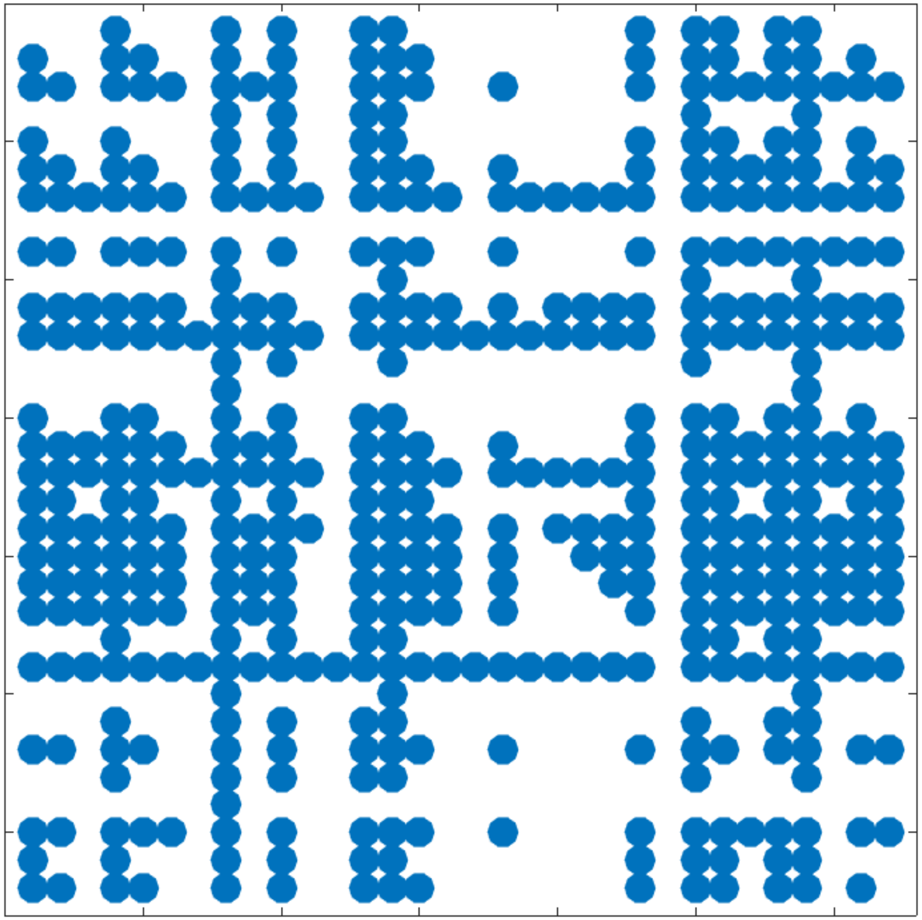
Reordering can uncover structure within matrices.

Rank these teams.



Rank

How would you rank these teams?



block by block

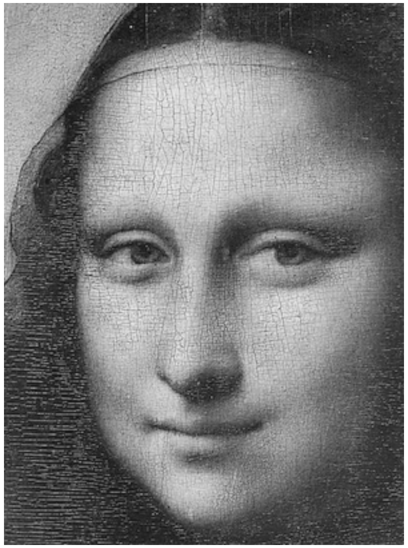
Let's see code to build this matrix.

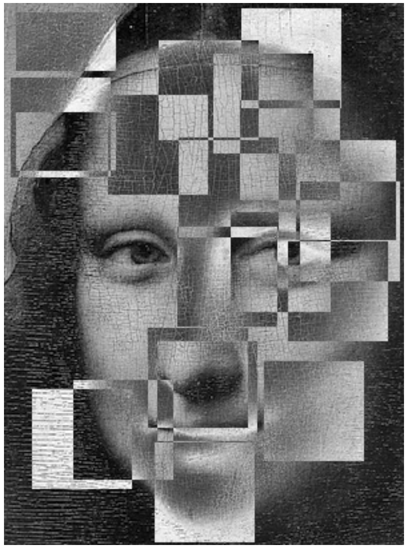
$$\left(\begin{array}{cccc|c} 1 & 2 & 0 & 0 & 1 \\ 3 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 1 & 3 \\ 0 & 0 & 2 & 3 & 1 \end{array}\right)$$

```
import numpy as np
A = np.zeros((4,4))
A[0:2,0:2] = np.array([[1,2],[3,5]])
A[2:4,2:4] = np.array([[4,1],[2,3]])
rhsVector = np.array([[1],[0],[3],[1]])
np.append(A,rhsVector,1)
```

Swap submatrices

Suppose we swap submatrices randomly in an image between the image and the inverted image. Can you see how this happened?



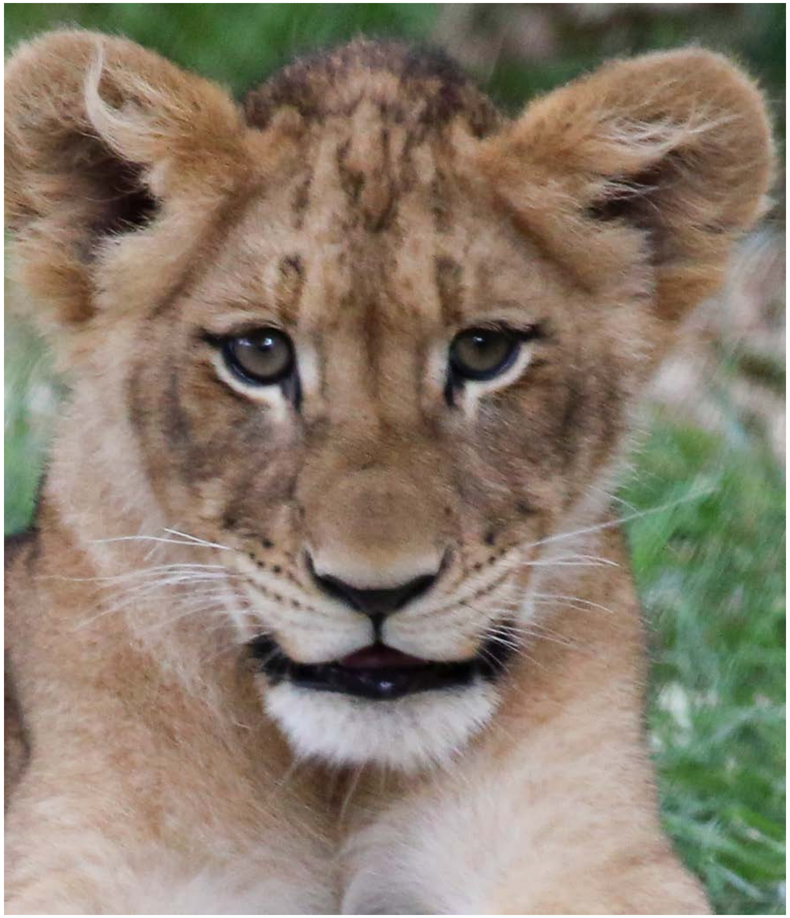


Code it!

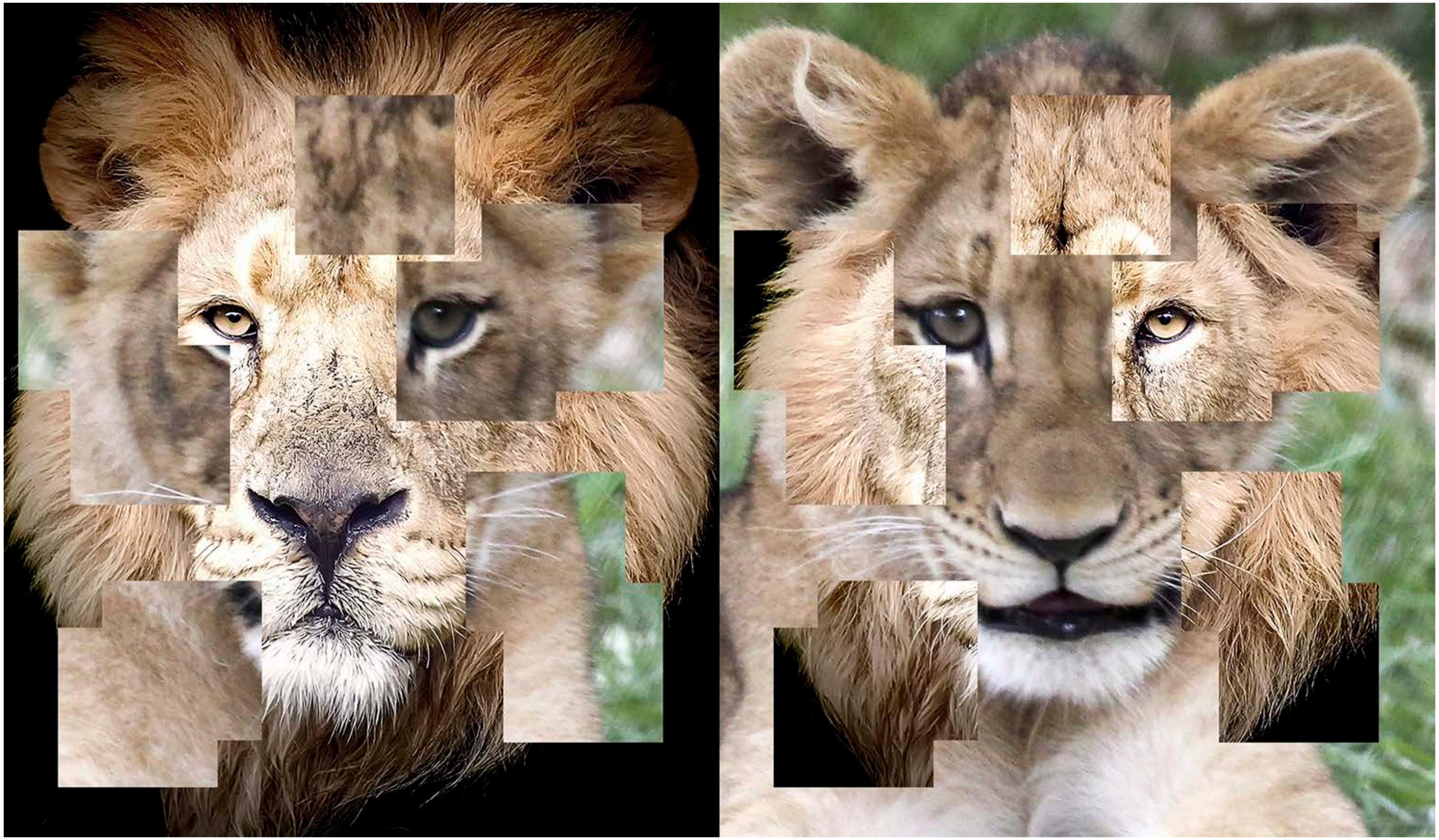
Let’s see this in python code.

? on programming

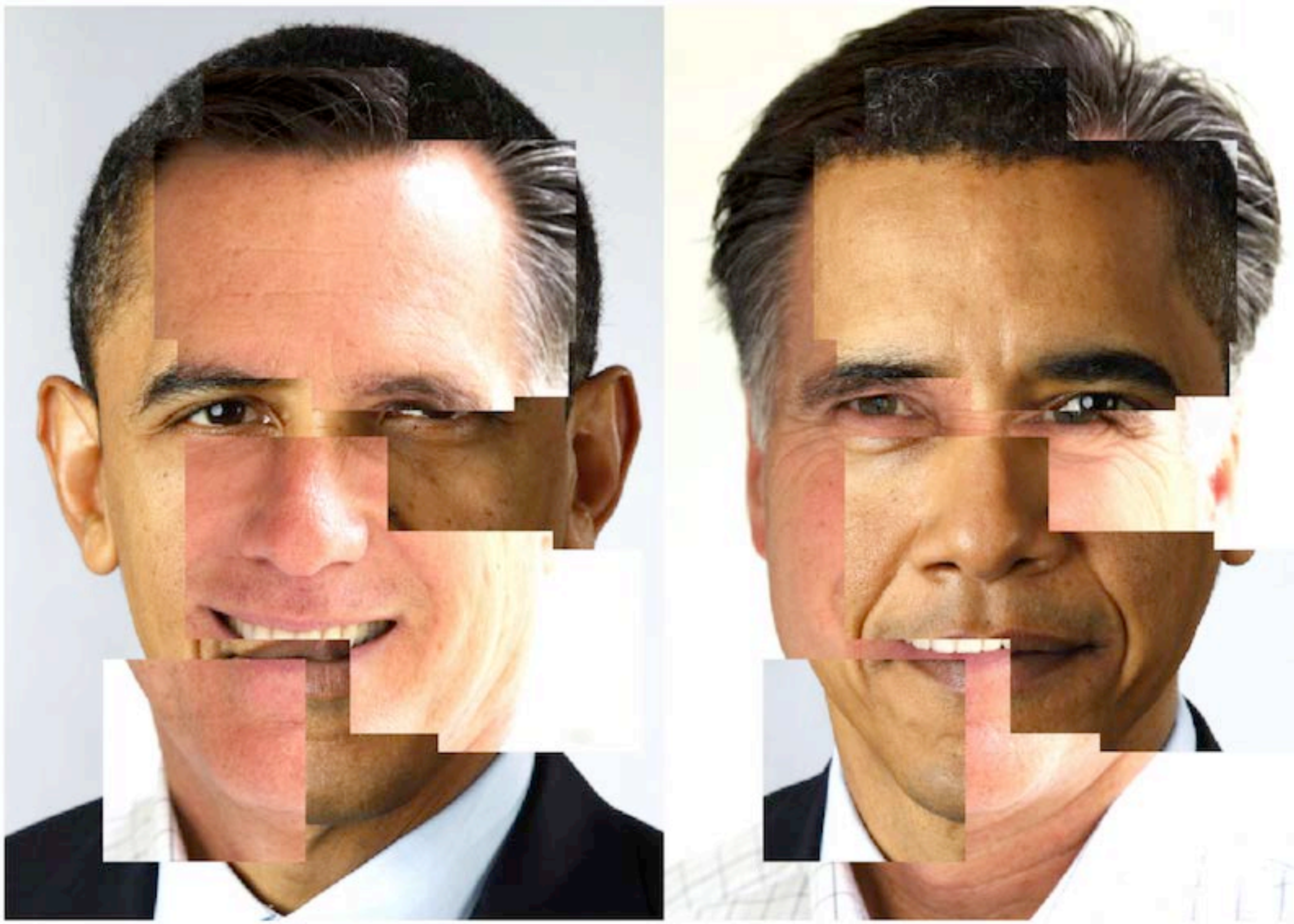
swapping in artistry



swapping submatrices



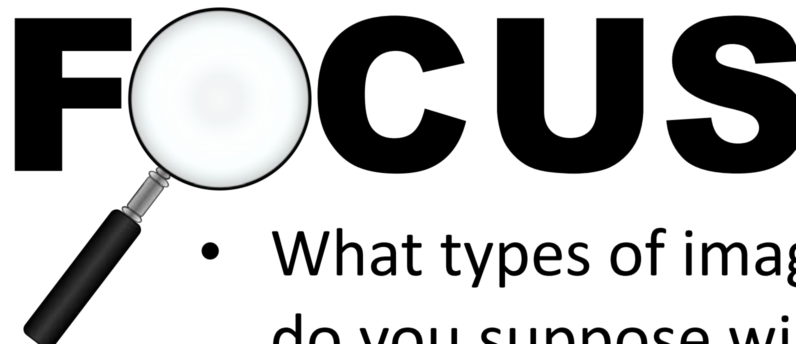
Obomney vs Rombama



Your Turn

Try it yourself at:

https://lifeislinear.davidson.edu/picasso.html



- What types of images do you suppose will work well or struggle with this algorithm?
- Any improvements?

Homework – 1

- Reading: Google PageRank, Simplified: A Guide for SEO Beginners: https://bit.ly/2NXZK3U (if you don't have access, let me know and I can print it)
- You will be asked for 2-3 paragraphs on your thoughts on this article. We will learn the mathematics of PageRank the next session.

Homework – 2

Install what is needed for you to begin coding!

Homework – 3

Use a swapping submatrix program on Schoology.

- Create a swapped submatrix image using your own images. Give an example that worked well and another example that didn't.
- Give an explanation, suitable to someone who has never worked with swapping submatrices, how to think what will and won't work well with such a program.

Homework – 4

The last part of the homework will ask your ideas on what application area attracts you to ranking. Do you already have something in mind to rank? If not, that's fine. If so, I'd like to know.

Note on submission

- All your responses should be contained in one document saved as a PDF.
- Update the PDF and your programming files to Schoology.

if time….

- A huge part of your project will depend on finding data on what you will rank.
- Can you find season data if it’s a sport?
- Can you find other data if you rank Twitter or movies?
- For now, begin to look around at options and let’s share with each other and ask questions.
- We’ll talk more about this in future lessons.