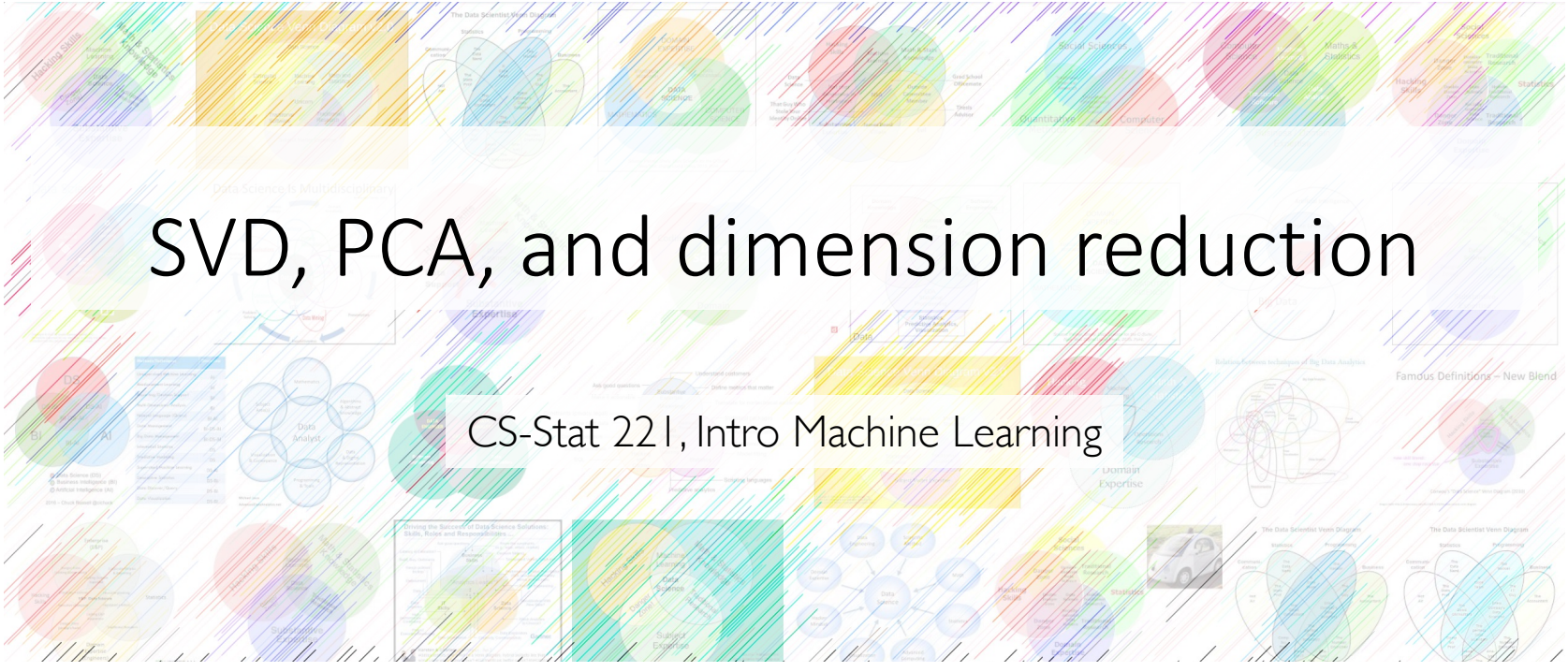


SVD, PCA, and dimension reduction

CS-Stat 221, Intro Machine Learning

THE UNIVERSITY OF CHICAGO

Consider genotype data

Individuals (few 1000s)

Genetic loci (few 100k)

	HGDP00448	HGDP00479	HGDP00985	HGDP00611	HGDP00623	HGDP00557	HGDP00569	HGDP00581
MitoA10045G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA10551G	AA	AA	AA	AA	AA	AA	--	--
MitoA11252G	AA	AA	AA	AA	GG	AA	AA	AA
MitoA11468G	AA	AA	AA	AA	AA	AA	GG	GG
MitoA11813G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA12309G	AA	AA	AA	AA	AA	AA	--	GG
MitoA13106G	GG	GG	GG	AA	AA	AA	AA	AA
MitoA13264G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA13781G	AA	AA	AA	AA	AA	AA	AA	AA
rs10000543	CC	CC	TC	CC	CC	CC	TT	CC
rs10000918	GG	AG	GG	AG	AG	AG	AA	AG
rs10000929	AG	AA	AA	AA	AA	AG	AA	AA
rs10001378	TT	CC	TT	TC	TC	TT	TT	TC
rs10001548	TC	TT	TC	TT	TC	TC	TT	CC
rs10002472	GG	AG	GG	GG	AG	GG	GG	GG
rs10004399	AA	AA	AA	AG	AG	AA	AA	AG
rs1000459	TT	TC	TC	CC	CC	CC		
rs10005550	GG	GG	GG	GG	GG	GG	GG	GG

https://hagsc.org/hgdp/files.html

Consider genotype data

Individuals (few 1000s)

Genetic loci (few 100k)

	HGDP00448	HGDP00479	HGDP00985	HGDP00611	HGDP00623	HGDP00557	HGDP00569	HGDP00581
MitoA10045G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA10551G	AA	AA	AA	AA	AA	AA	--	--
MitoA11252G	AA	AA	AA	AA	GG	AA	AA	AA
MitoA11468G	AA	AA	AA	AA	AA	AA	GG	GG
MitoA11813G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA12309G	AA	AA	AA	AA	AA	AA	--	GG
MitoA13106G	GG	GG	GG	AA	AA	AA	AA	AA
MitoA13264G	AA	AA	AA					
MitoA13781G	AA	AA	AA					
rs10000543	CC	CC	TC					
rs10000918	GG	AG	GG					
rs10000929	AG	AA	AA					
rs10001378	TT	CC	TT					
rs10001548	TC	TT	TC					
rs10002472	GG	AG	GG	GG	AG	GG	GG	GG
rs10004399	AA	AA	AA	AG	AG	AA	AA	AG
rs1000459	TT	TC	TC	CC	CC	CC	TC	CC
rs10005550	GG	GG	GG	GG	GG	GG	GG	GG

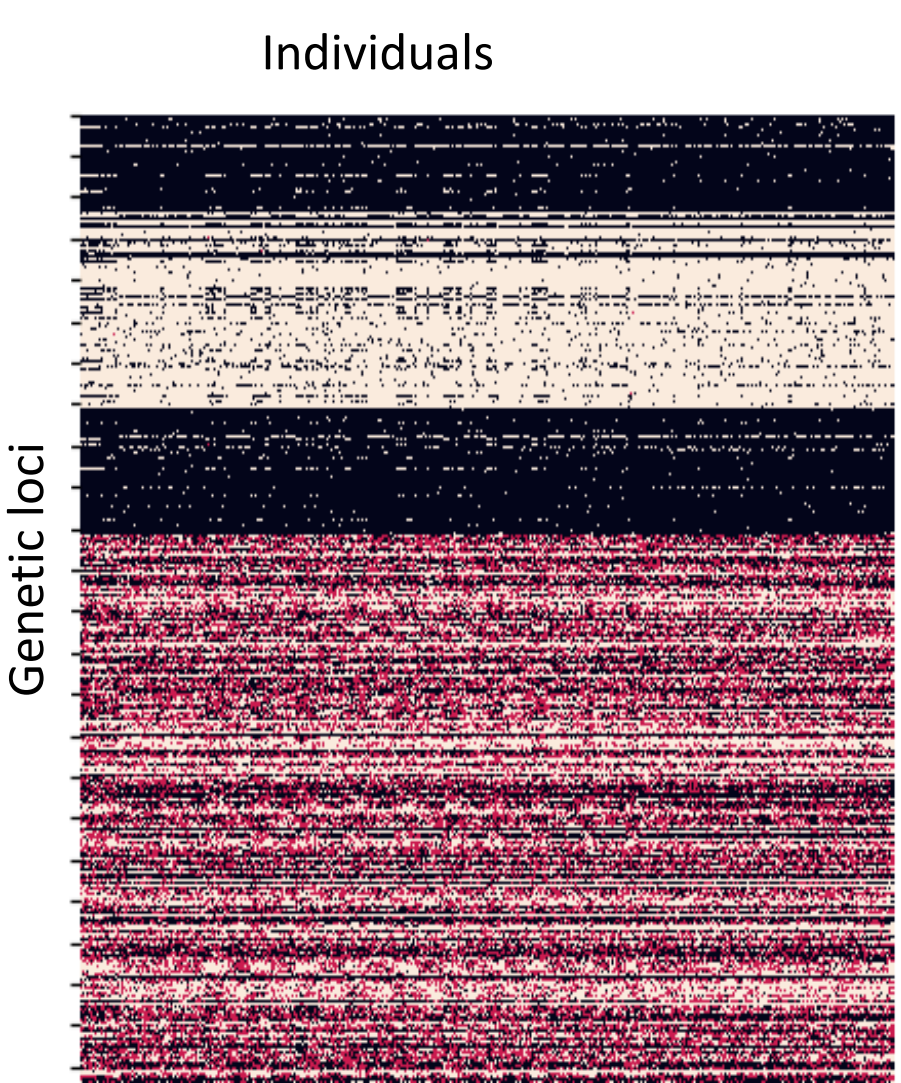
- Convert to numbers
- compare columns ?
- Would like to visualize in 2 dimesions:
  - 1000 x 2 ?

Suppose your data looks like genotype data

Individuals (few 1000s)

Genetic loci (few 100k)

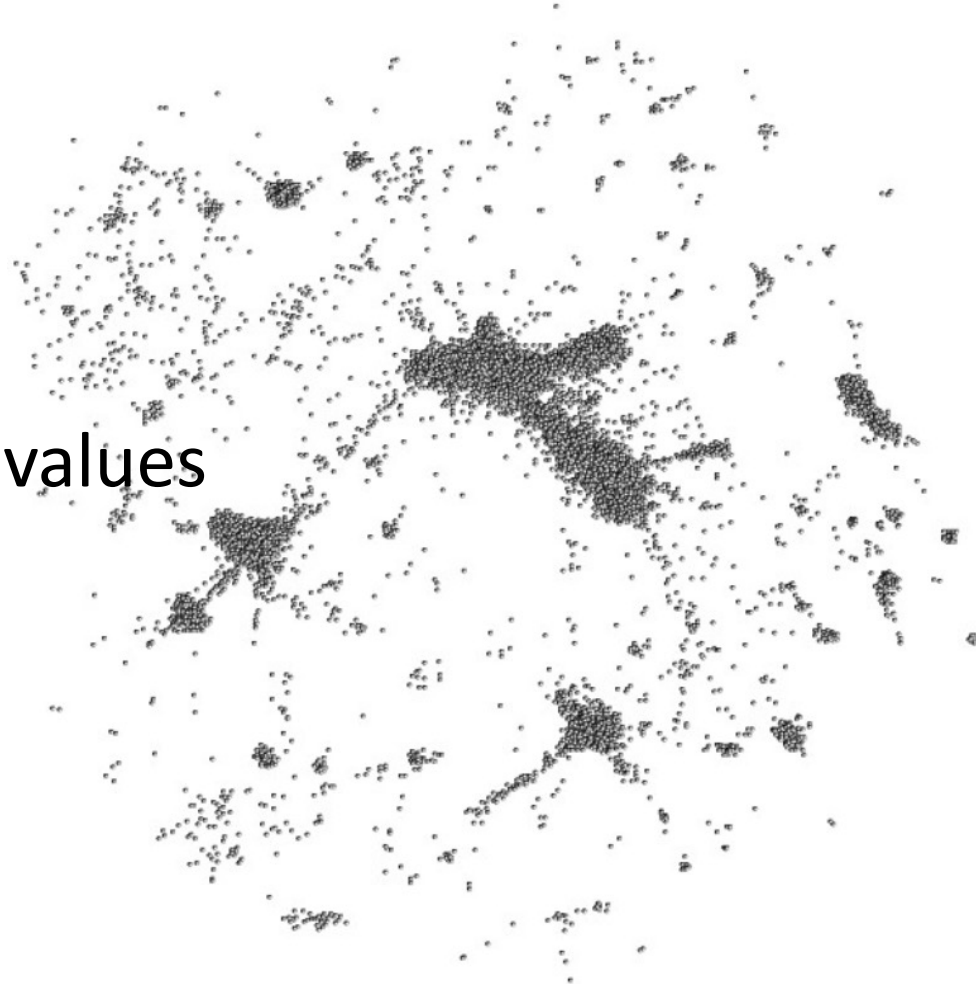
	HGDP00448	HGDP00479	HGDP00985	HGDP00611	HGDP00623	HGDP00557	HGDP00569	HGDP00581
MitoA10045G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA10551G	AA	AA	AA	AA	AA	AA	--	--
MitoA11252G	AA	AA	AA	AA	GG	AA	AA	AA
MitoA11468G	AA	AA	AA	AA	AA	AA	GG	GG
MitoA11813G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA12309G	AA	AA	AA	AA	AA	AA	--	GG
MitoA13106G	GG	GG	GG	AA	AA	AA	AA	AA
MitoA13264G	AA	AA	AA	AA	AA	AA	AA	AA
MitoA13781G	AA	AA	AA	AA	AA	AA	AA	AA
rs10000543	CC	CC	TC	CC	CC	CC	TT	CC
rs10000918	GG	AG	GG	AG	AG	AG	AA	AG
rs10000929	AG	AA	AA	AA	AA	AG	AA	AA
rs10001378	TT	CC	TT	TC	TC	TT	TT	TC
rs10001548	TC	TT	TC	TT	TC	TC	TT	CC
rs10002472	GG	AG	GG	GG	AG	GG	GG	GG
rs10004399	AA	AA	AA	AG	AG	AA	AA	AG
rs1000459	TT	TC	TC	CC	CC	CC	TC	CC
rs10005550	GG	GG	GG	GG	GG	GG	GG	GG



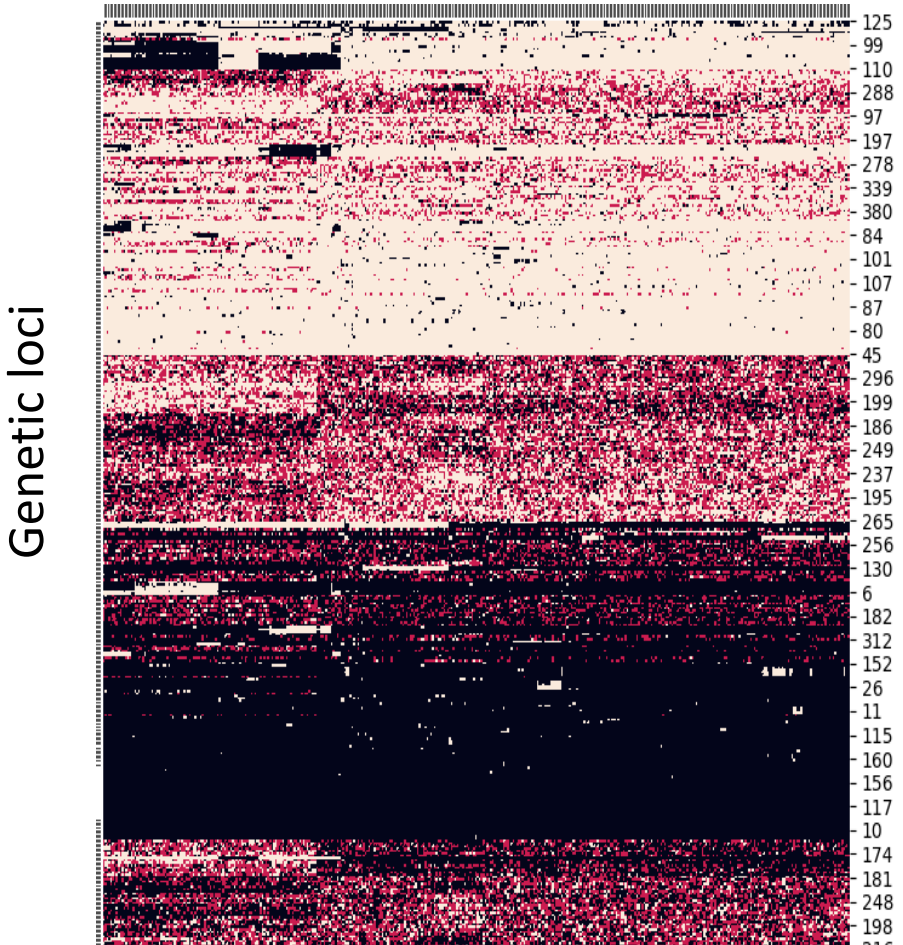
- Some rows are similar to each other, some columns are similar to each other
- The data are in some way “predictable”
- Can we build a low-complexity object that explains most of the data matrix.

Roadmap

- Dimension reduction
- Absolute essentials of eigenvectors and eigenvalues
- SVD and PCA
- Examples



Now that looks like structure



- Reorder rows and columns to put similar rows next to each other?
- Libraries to make exactly this sort of eye-candy are mature. This one is `sanborn.clustermap`
- Organizing data like this is deeply satisfying—it makes us think we have discovered truth.
- What if I told you linear algebra could do this?

Why Reduce dimensions?

m columns
“features”

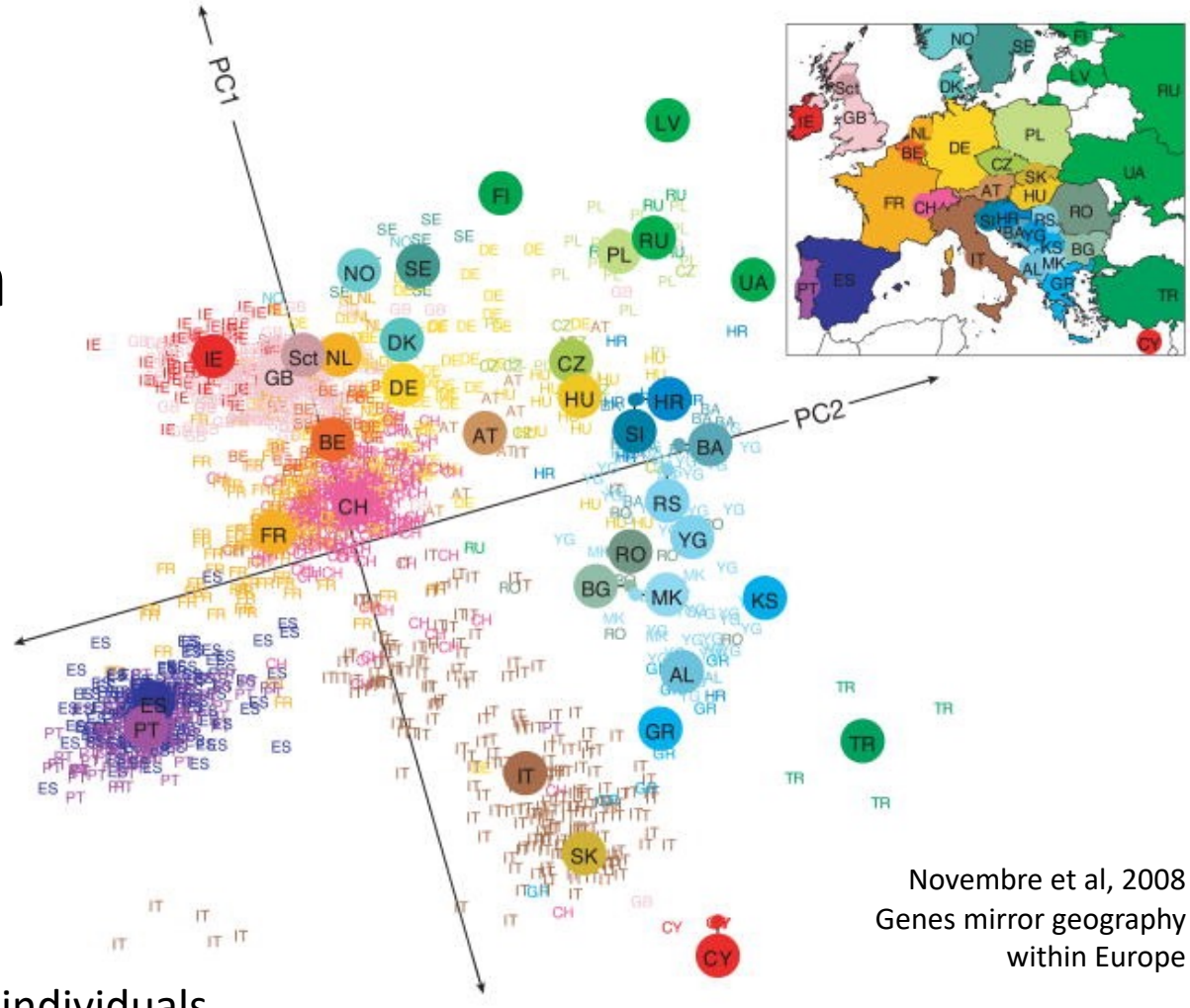
n rows
”samples”

- Reducing the number of dimensions can make ML easier
- May reduce noise.
- Can exploit unlabeled data to make nicer inputs for ML
- We need dimension reduction and categorization to understand high-dimensional data
- Useful for making visualizations of high-dimensional data.

r
“reduced
features”

n samples

Example: Genes mirror geography within Europe



m x n = 200k genetic loci x 1400 individuals

Novembre et al, 2008
Genes mirror geography within Europe

10.1038/nature07331

How many dimensions again?

n x m can be large (compared to our computers and our displays):

Twitter (500M tweets / day, May 2020)

Youtube (2.5 billion views / day)

Natural Language processing (10^4-10^6 words)

Netflix prize was 500K users x 20K movies

Biochemical or genetic assays (10^4-10^7 features of interest x sample size)

Image processing (10^7 pixels easy)

Definition of eigenvectors and eigenvalues

There exists a set of vectors $\{e_i\}$

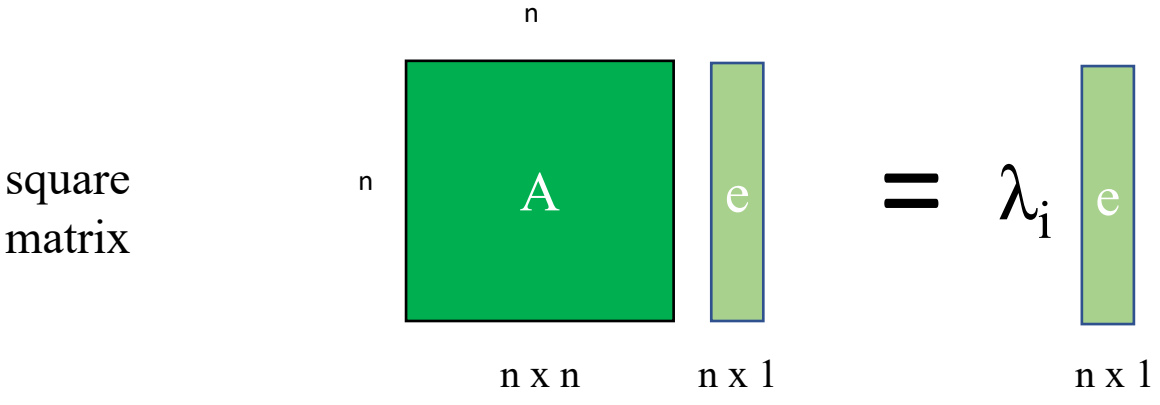
That has the following nice property

square matrix
n x n

ith eigenvalue

ith eigenvector
n x 1

$$\mathbf{A}\,\mathbf{e}_i = \lambda_i\,\mathbf{e}_i$$

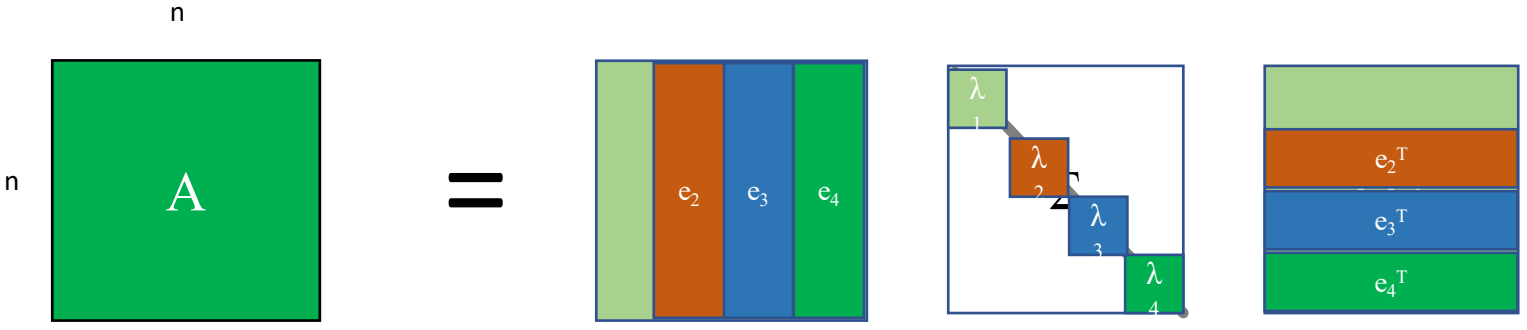


Eigenvalue decomposition

These vectors can be constructed orthogonal (zero inner products)

and normal (x dot x = 1)

$$\mathbf{A} = \text{Sum}_i \ \mathbf{e}_i^T \lambda_i \mathbf{e}_i$$



square matrix

eigenvectors (orthogonal)

eigenvalues (diagonal)

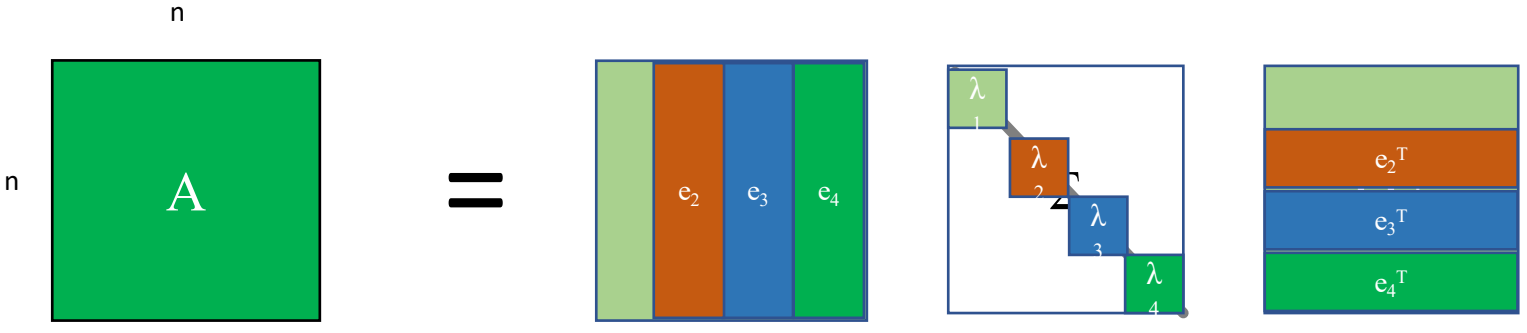
eigenvectors (orthogonal)

A $\quad Q^{-1} \quad \Sigma \quad Q$

Eigenvalue decomposition

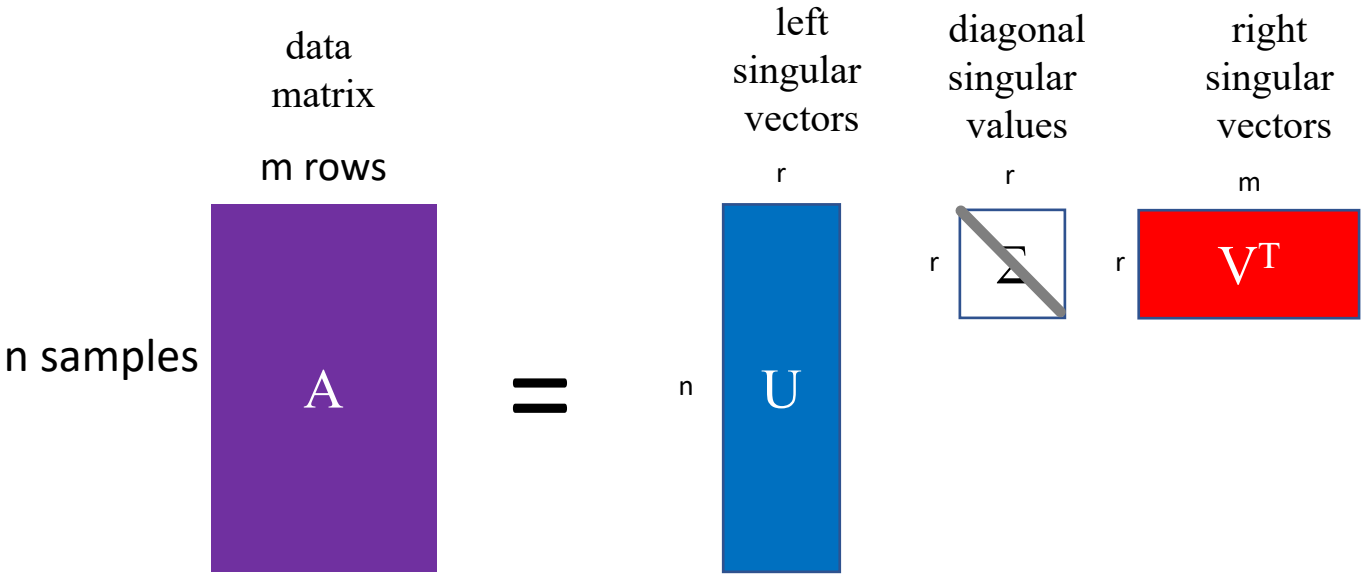
```
import numpy as np
from numpy import linalg as LA

# A library function will take a square matrix and give
# back a vector of eigenvalues and a matrix of eigenvectors.

A=np.array( [ [16, 2, 3, 13 ],[5, 11, 10, 8], [9, 7, 6, 12],
[4, 14, 15, 1] ])
evalues, evectors = LA.eig(A)
print(evalues)
print(evectors)
B=np.dot(np.dot(evectors , evalues * np.eye(4)) ,
LA.inv(evectors))
```

Singular Value Decomposition (SVD)



- Eigenvalue decomposition is very nice, but only applies to square data matrices.
- Singular Value Decomposition (SVD) is a generalization for rectangular matrices

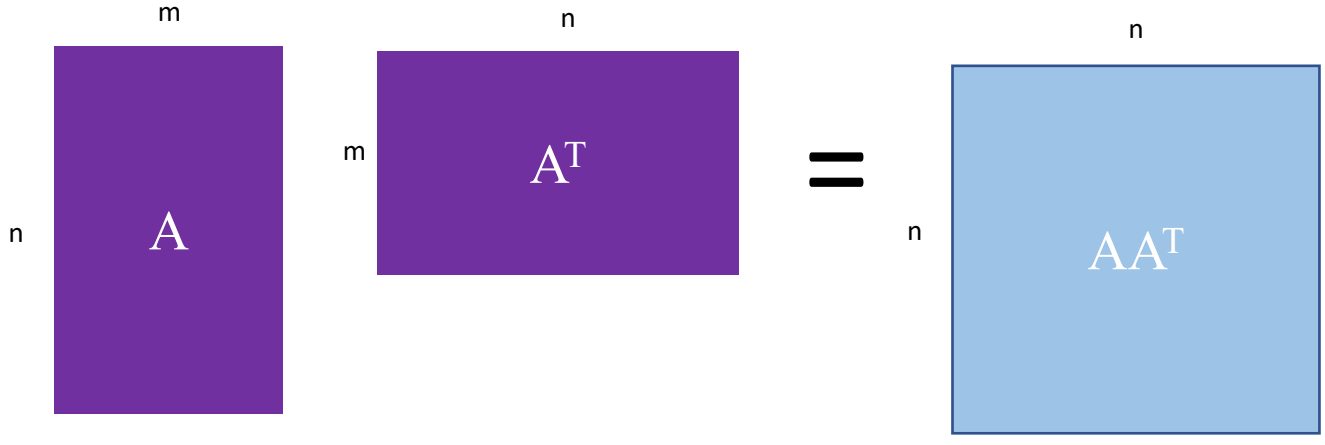
Singular Value Decomposition (SVD)



- Eigenvalue decomposition is very nice, but only applies to square data matrices.
- Singular Value Decomposition (SVD) is a generalization for rectangular matrices

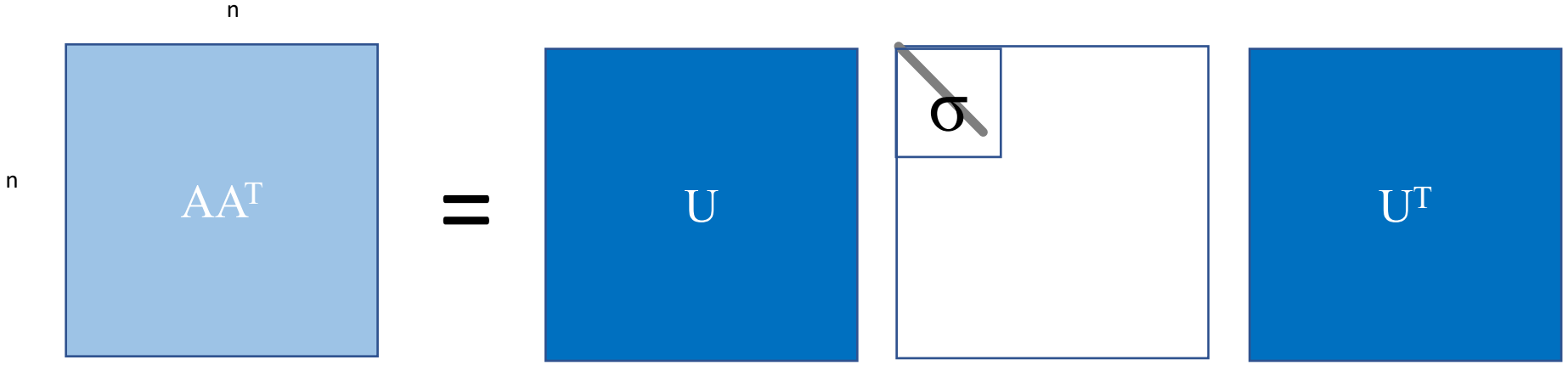
Constructing SVD - rows

- Recipe for building SVD:



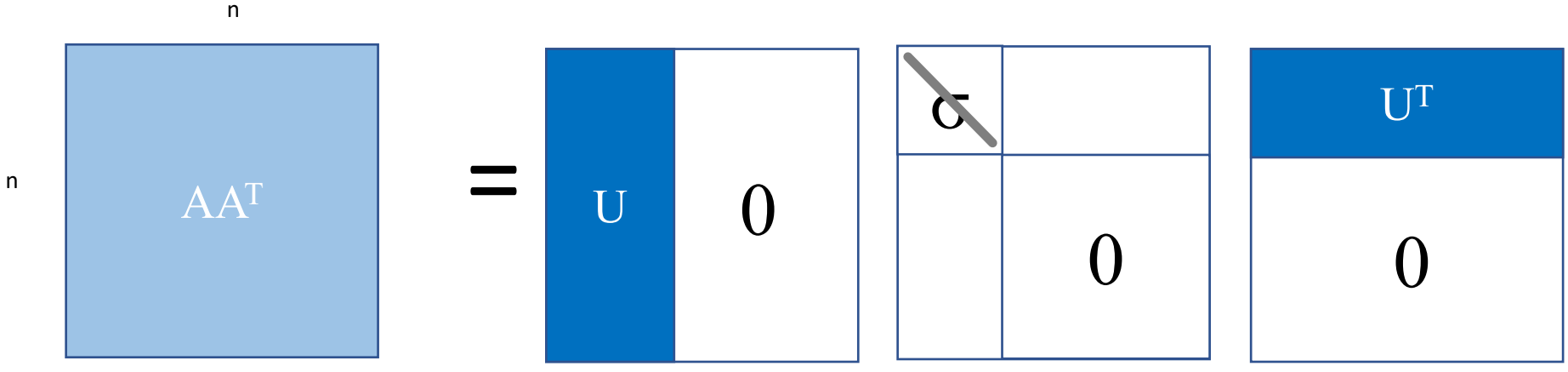
This is the matrix of inner products between rows; summed over all the samples

Constructing SVD - rows



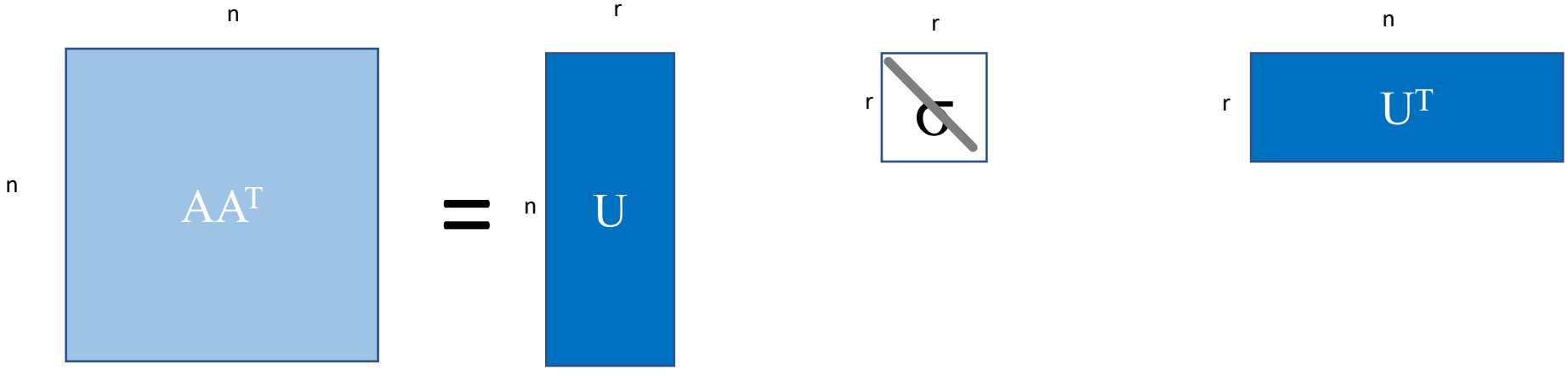
- Now take the eigenvalue decomposition of AA^T

Constructing SVD - rows



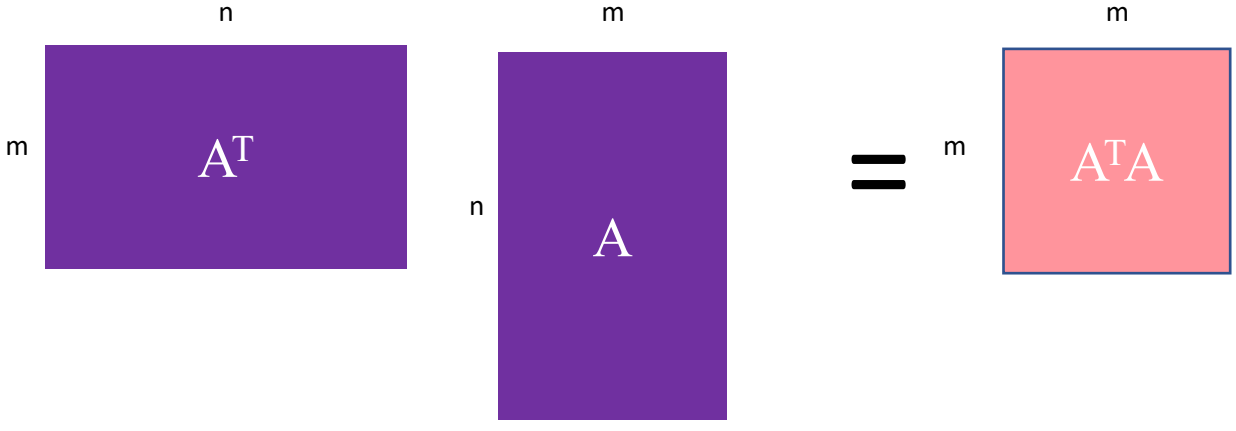
- Now take the eigenvalue decomposition of AA^T
- I have a bunch of zero eigenvalues, so I have a bunch of eigenvectors that can't contribute to the sum

Constructing SVD - rows



- Now take the eigenvalue decomposition of AA^T
- U (n x r) , σ (r), U^T (r x n)

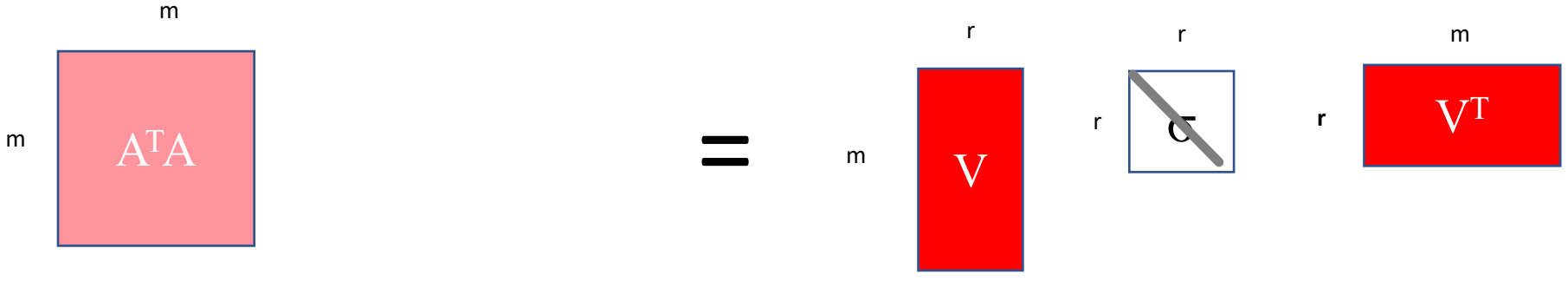
Constructing SVD - columns



- This matrix has dimensions m_features x m_features

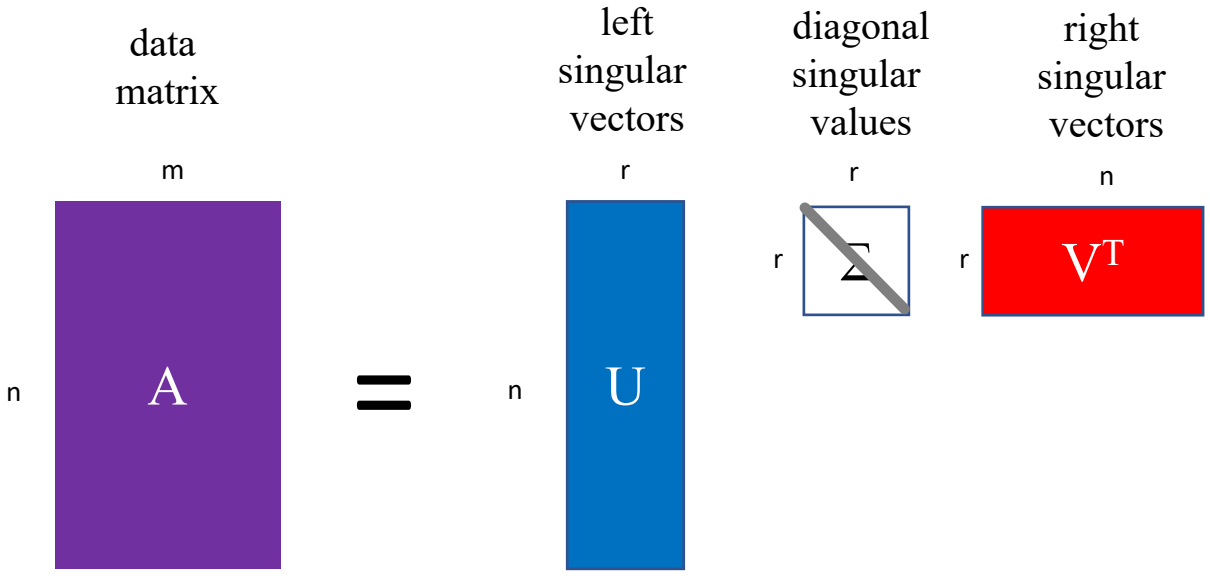
Constructing SVD - columns

Eigenvalue decomposition again



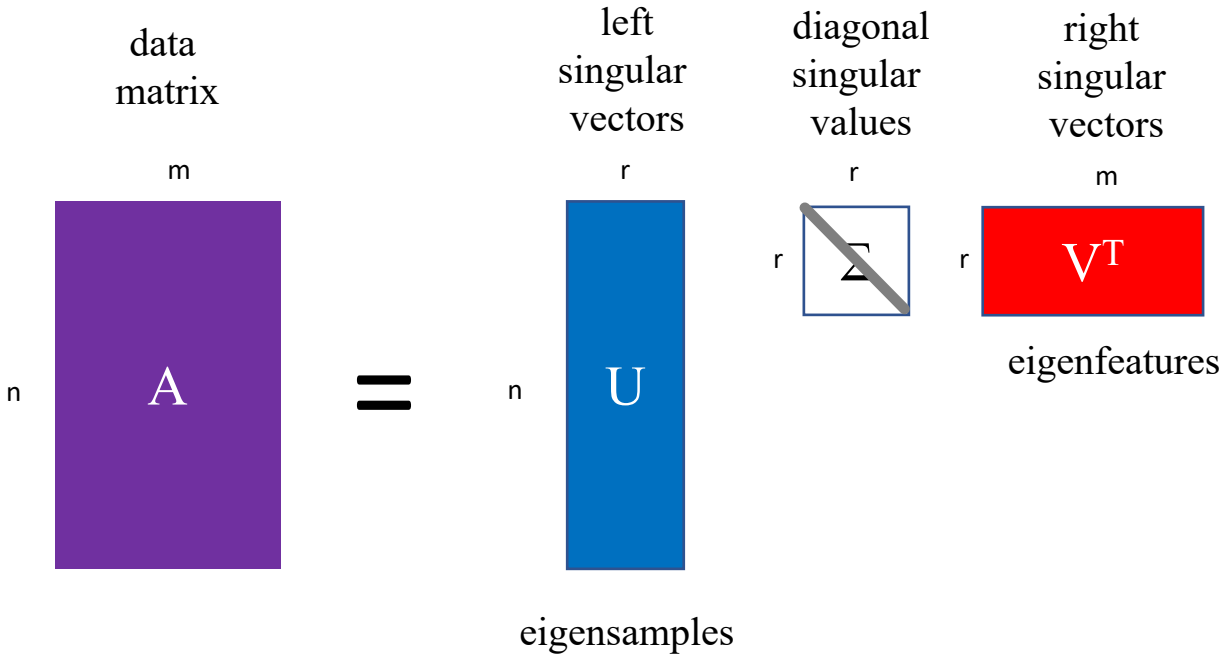
- Because of the relationship between A^TA and AA^T these two matrices have the same eigenvalues. (But one is m x m and one is n x n !!)
- Some of the eigenvalues have to be zero in the larger matrix

Singular Value Decomposition (SVD)



- I have two sets of basis vectors and one set of r singular values.
- Two powers of A -> elements of Σ are square roots of eigenvalues

Singular Value Decomposition (SVD)



- I have two sets of basis vectors and one set of r singular values.
- Two powers of A -> elements of Σ are square roots of eigenvalues

SVD

```
import numpy as np
from numpy import linalg as LA

#
https://numpy.org/doc/stable/refe
rence/generated/numpy.linalg.svd.
html

U, SIGMA VT = LA.svd(A)
```

We could calculate SVD ourselves in two lines –

but we shouldn't

Approximate, truncated SVD

```
import numpy as np
from numpy import linalg as LA
from sklearn.utils.extmath import
randomized_svd

# https://scikit-
learn.org/stable/modules/generate
d/sklearn.utils.extmath.randomize
d_svd.html


U, SIGMA, VT = randomized_svd(A,
n_components=50)
```

The library functions use convergence or optimization to find approximate matrix factorizations.

Does not actually build or store A^TA

Approximate and truncated decompositions are cheaper to compute.

Principal Component analysis

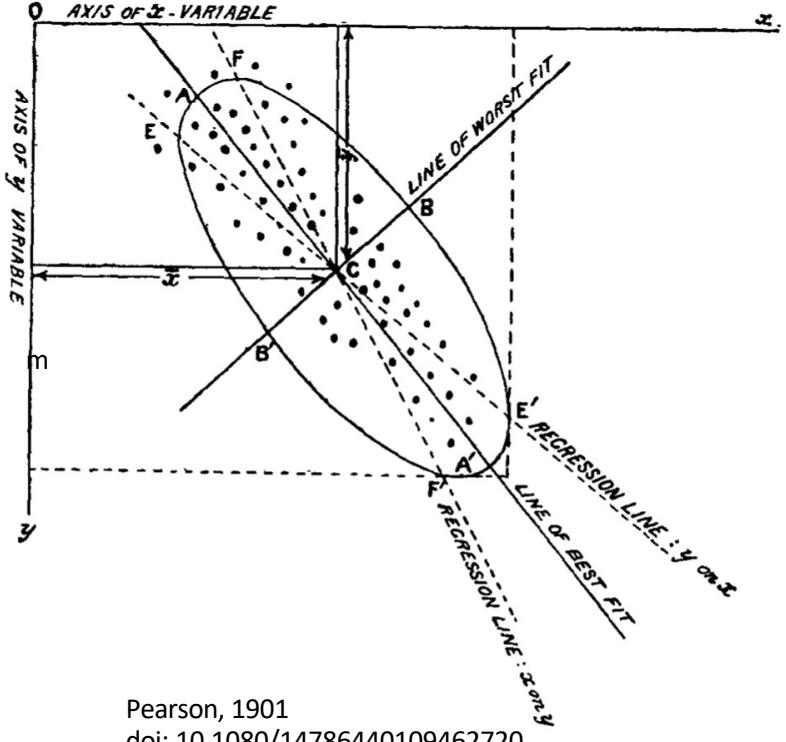
- SVD is a just linear algebra + geometry interpretation of the columnwise & rowwise inner products.
- PCA is the doctrine of using vectors derived from SVD to interpret data.
- The data points are rotated onto the singular vectors, renamed PCA coordinates
- $XV^T = T$ (new coordinates)

$(n \times m) * (m \times r) = (n \times r)$

- Can make pretty pictures

m

r



Pearson, 1901
doi: 10.1080/14786440109462720

Table 1
Principal Components Analysis of BIS-11 Items (Oblique Rotation)

BIS-11 items	First-order factors					
	1	2	3	4	5	6
11. I "squirm" at plays or lectures.	**.84**	.17	−.08	−.03	.03	.02
32. I am restless at the theater or lectures.	**.84**	.19	−.12	−.06	−.00	−.03
5. I don't "pay attention."	**.57**	.04	.16	−.02	.27	.02
17. I act "on impulse."	.15	**.74**	.08	−.02	−.20	.06
20. I act on the spur of the moment.	.12	**.72**	.19	−.10	−.19	.01
23. I buy things on impulse.	−.08	**.59**	−.04	.28	.10	.11
12. I am a careful thinker.[a]	.17	−.13	**.64**	.17	−.18	.05
1. I plan tasks carefully.[a]	−.05	.16	**.64**	−.04	.11	−.10
8. I am self-controlled.[a]	.10	.00	**.63**	−.24	.08	−.17

Patton et al. 1995 Factor structure of the Barratt Impulsiveness Scale

doi: 10.1002/1097-4679(199511)51:6<768::AID-JCLP2270510607>3.0.CO;2-1

Table 1
Principal Components Analysis of BIS-11 Items (Oblique Rotation)

BIS-11 items	First-order factors					
	1	2	3	4	5	6
11. I "squirm" at plays or lectures.	.84	.17	−.08	−.03	.03	.02
32. I am restless at the theater or lectures.	.84	.19	−.12	−.06	−.00	−.03
5. I don't "pay attention."	.57	.04	.16	−.02	.27	.02
17. I act "on impulse."	.15	.74	.08	−.02	−.20	.06
20. I act on the spur of the moment.	.12	.72	.19	−.10	−.19	.01
23. I buy things on impulse.	−.08	.59	−.04	.28	.10	.11
12. I am a careful thinker.[a]	.17	−.13	.64	.17	−.18	.05
1. I plan tasks carefully.[a]	−.05	.16	.64	−.04	.11	−.10
8. I am self-controlled.[a]	.10	.00	.63	−.24	.08	−.17

Patton et al. 1995 Factor structure of the Barratt Impulsiveness Scale

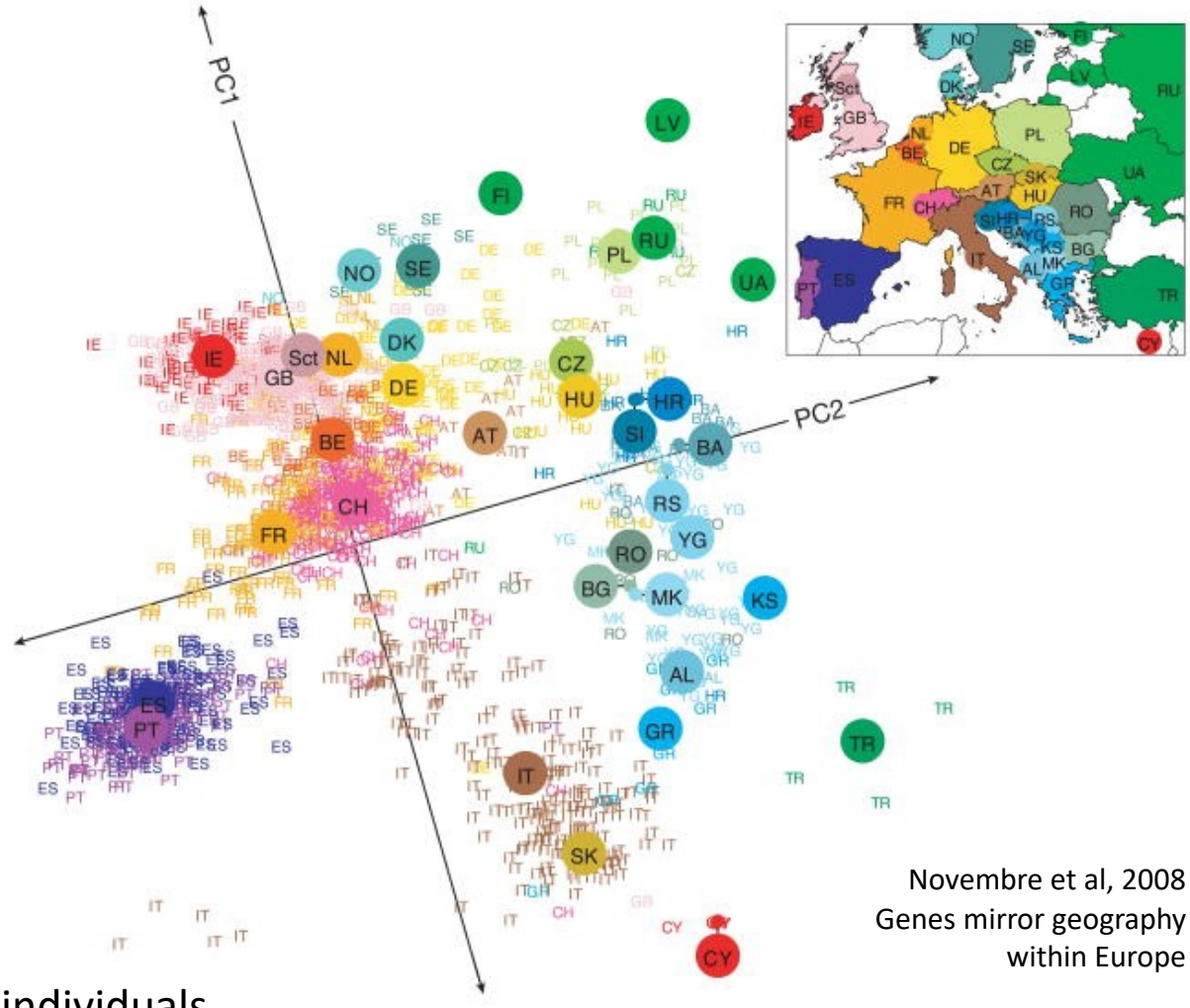
doi: 10.1002/1097-4679(199511)51:6<768::AID-JCLP2270510607>3.0.CO;2-1

Table 1
Principal Components Analysis of BIS-11 Items (Oblique Rotation)

BIS-11 items	First-order factors					
	1	2	3	4	5	6
11. I "squirm" at plays or lectures.	**.84**	.17	−.08	−.03	.03	.02
32. I am restless at the theater or lectures.	**.84**	.19	−.12	−.06	−.00	−.03
5. I don't "pay attention."	**.57**	.04	.16	−.02	.27	.02
17. I act "on impulse."	.15	**.74**	.08	−.02	−.20	.06
20. I act on the spur of the moment.	.12	**.72**	.19	−.10	−.19	.01
23. I buy things on impulse.	−.08	**.59**	−.04	.28	.10	.11
12. I am a careful thinker.[a]	.17	−.13	**.64**	.17	−.18	.05
1. I plan tasks carefully.[a]	−.05	.16	**.64**	−.04	.11	−.10
8. I am self-controlled.[a]	.10	.00	**.63**	−.24	.08	−.17

Data correlations here used to inform reduction of dimension from 35-question survey to 6 scores.

Genes mirror geography within Europe



m x n = 200k genetic loci x 1400 individuals

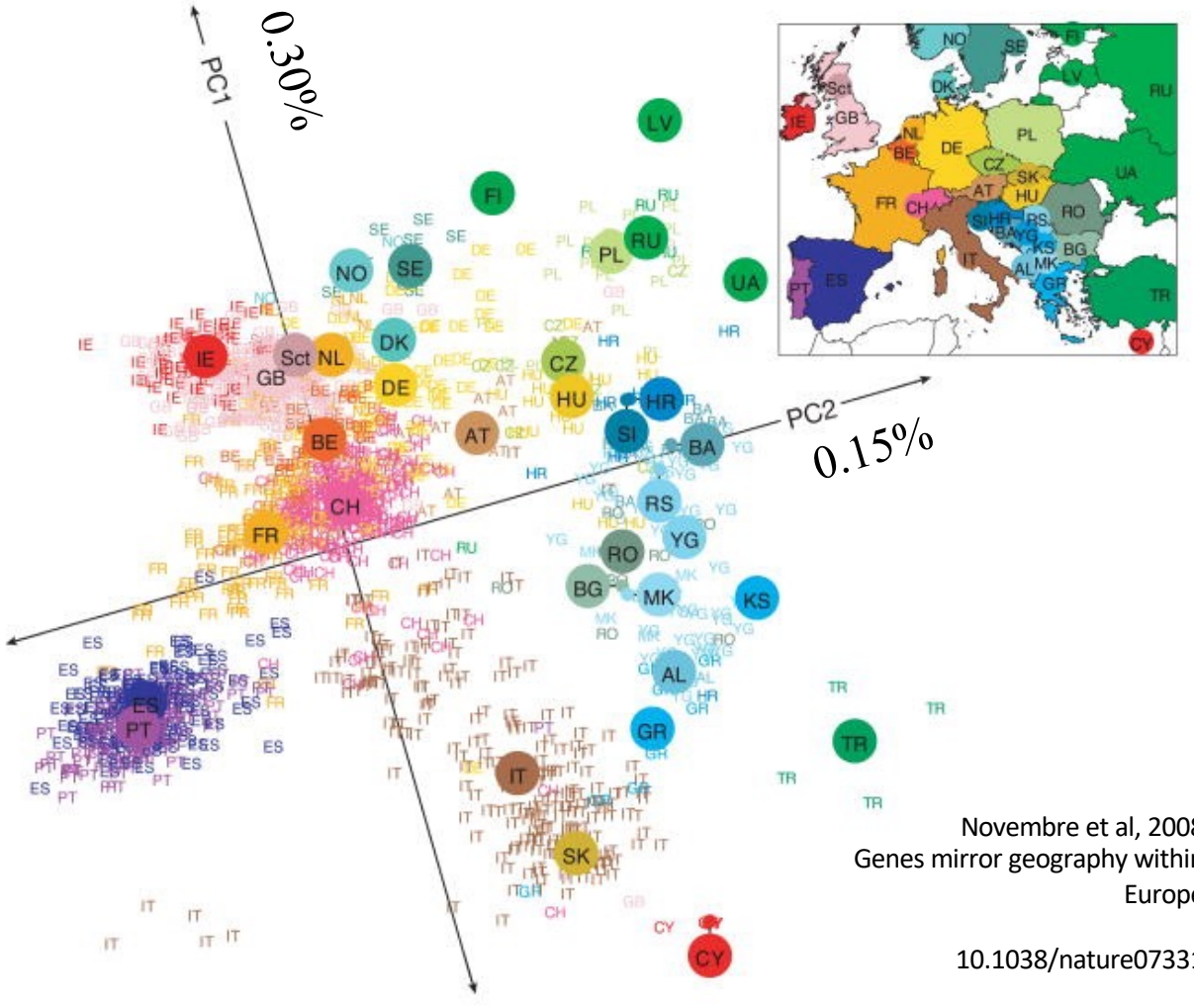
Novembre et al, 2008
Genes mirror geography within Europe

10.1038/nature07331

Genes mirror geography within Europe

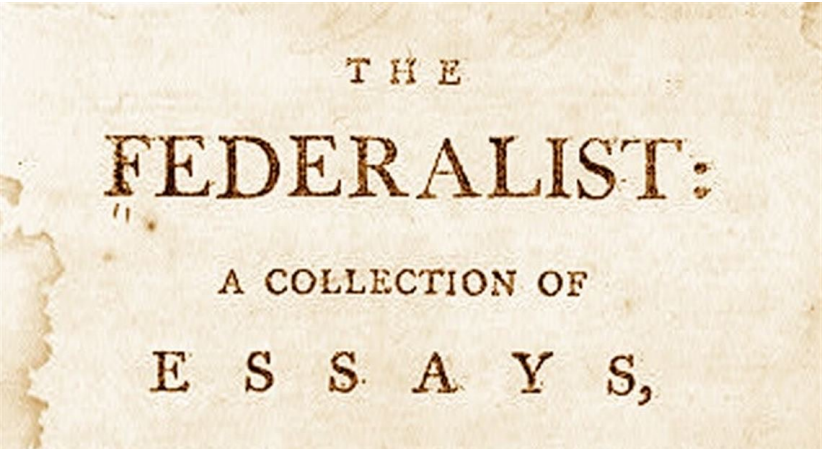
Principal components and eigenvalues come back from largest to smallest magnitude.

It is conventional to report principal components along with the fraction of variance explained by each



Novembre et al, 2008
Genes mirror geography within Europe

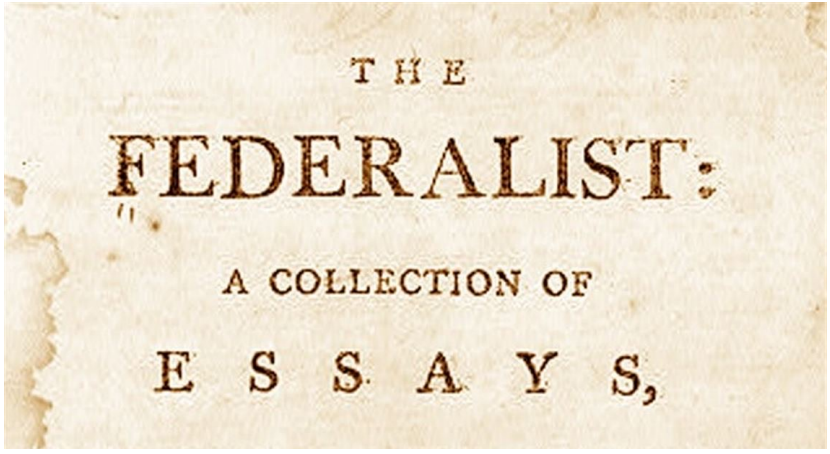
10.1038/nature07331



- 85 documents, initially published anonymously 1787-1788
- Authors' reminiscences disagree about authorship of some of the essays
- 85 documents parsed into vocabulary of ~10000 "words"

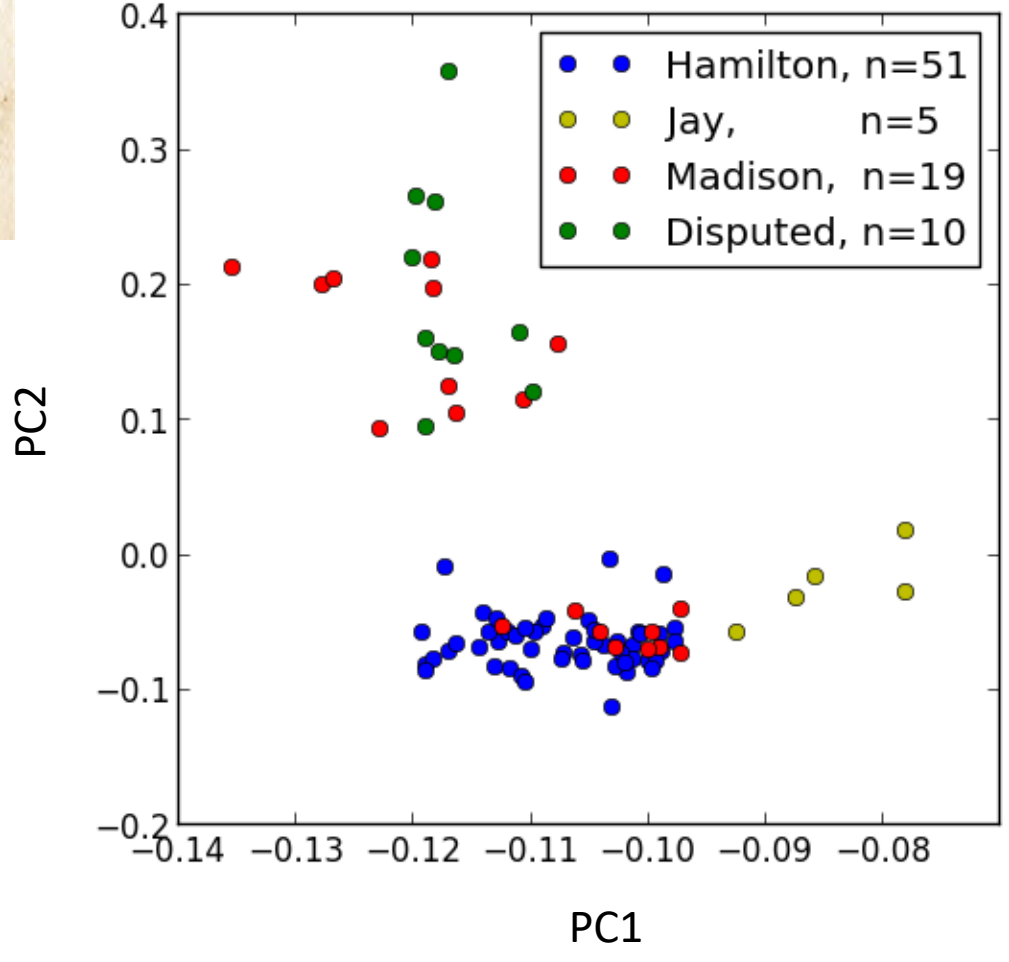
What if your data looks like this?

NEW	7	3	2	1	1	2	11	3	7	2	2	4	4	6	3	2	1	3	2	2
SO	3	4	4	5	3	4	4	5	5	4	4	6	3	6	2	3	4	2	4	2
THEY	6	22	5	17	11	11	7	13	20	11	9	8	3	17	15	14	7	12	9	7
AGAINST	2	1	3	6	5	4	1	5	5	6	1	3	1	4	4	6	2	4	6	0
ANY	6	1	4	5	3	0	6	2	4	4	4	7	3	2	3	11	2	2	6	0
EITHER	2	0	5	4	1	5	3	1	3	1	3	0	0	1	1	3	1	0	1	0
FOR	13	14	11	12	8	14	15	9	11	18	19	8	2	19	18	9	12	8	8	17
GOVERNMENT	5	4	5	11	2	1	1	1	8	4	2	6	7	7	4	10	4	3	1	2
GUARANTY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NATIONAL	1	3	14	7	0	4	2	2	2	2	4	4	4	2	9	9	8	0	2	0
NATURE	0	0	0	0	0	1	0	1	0	2	1	2	0	2	2	1	3	0	1	2
ON	9	8	6	11	5	2	13	11	9	18	5	12	3	17	10	4	2	16	17	7
OTHER	3	3	7	10	2	5	4	4	4	16	6	11	1	6	3	2	7	2	5	4
OWN	4	1	0	4	0	0	4	0	2	3	4	1	1	4	2	2	0	2	3	1
STATES,	0	0	6	0	1	3	4	4	4	1	5	1	3	3	1	2	1	1	1	2
UPON	6	1	0	0	0	4	11	3	4	0	6	7	2	0	10	6	6	1	0	1
COULD	1	1	1	2	0	0	2	3	3	2	3	2	0	1	3	12	1	2	2	1
KIND	0	2	1	0	1	0	0	0	3	0	1	1	0	0	5	4	1	0	0	0
SHALL	7	1	0	6	1	1	1	0	4	2	3	1	3	0	1	1	1	0	2	0
SHOULD	1	3	4	1	4	2	7	4	4	0	3	4	0	2	6	6	2	1	0	0
STATES	2	1	4	1	0	4	25	7	5	1	7	8	5	8	8	6	2	0	2	0
TOTAL	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
VALUE	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0
WANT	0	0	2	1	0	1	0	1	0	0	0	1	0	1	1	0	0	0	0	1
WHO	8	7	1	3	3	7	2	0	3	9	4	1	3	1	6	5	2	8	5	3
AT	8	9	1	2	4	6	11	11	10	8	11	13	1	7	24	11	7	4	5	8
BETWEEN	0	0	0	2	3	8	9	3	3	3	4	5	1	4	2	3	4	2	4	1
DUTIES	0	0	0	1	0	0	2	0	0	0	0	6	0	0	0	0	0	0	1	2
EACH	0	5	0	2	6	4	4	2	0	4	7	4	4	0	5	1	2	2	2	4
LANDS	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0



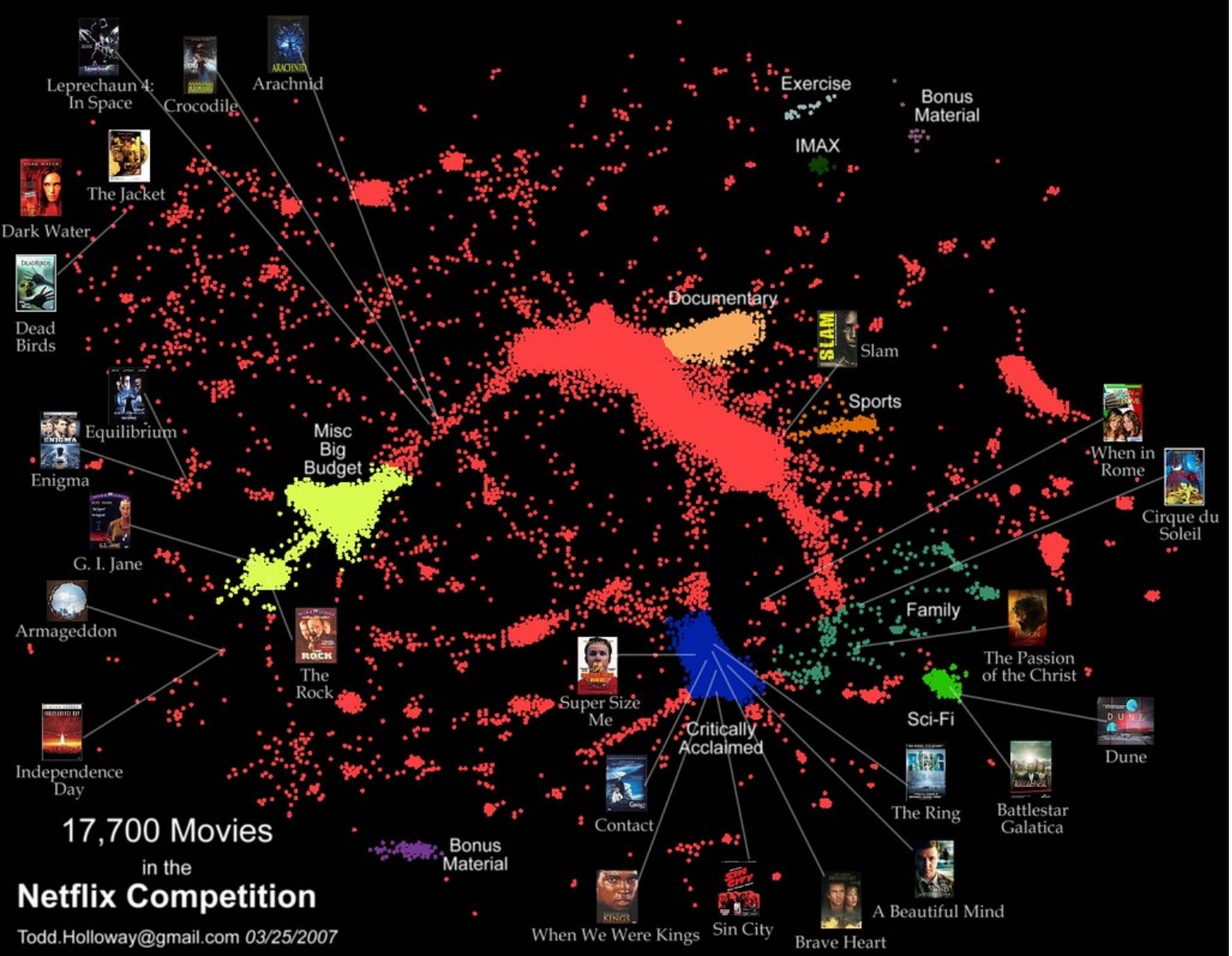
- Sometimes $\sum(x-\bar{x})(y-\bar{y})$ is not an appropriate distance score for your data.

- You can run PCA on an n x n distance matrix evaluated using a domain-specific columnwise comparison.
- Resulting low-dimensional coordinates visualize vocabulary/style relationships



Caveats

- SVD/PCA don’t know anything but the cloud of points and its correlations.
- Each new sample builds a different set of basis vectors
- Hard to interpret: positive and negative values in the eigenvectors
- Does not handle non-numerical values—have to change the objective function for that (Netflix)


Leprechaun 4: In Space
Crocodile
Arachnid
Exercise
Bonus Material
IMAX
The Jacket
Dark Water
Dead Birds
Documentary
Slam
Sports
Equilibrium
Misc Big Budget
Enigma
When in Rome
Cirque du Soleil
G. I. Jane
Armageddon
Family
The Rock
The Passion of the Christ
Super Size Me
Critically Acclaimed
Sci-Fi
Dune
Independence Day
Contact
The Ring
Battlestar Galatica
17,700 Movies
in the
Netflix Competition
Bonus Material
When We Were Kings
Sin City
Brave Heart
A Beautiful Mind
Todd.Holloway@gmail.com 03/25/2007

Dimension reduction approaches

- **Hierarchical clustering (built into clustermap):** takes pairwise distances as input; outputs a tree.
- PCA/SVD **(Principal Components Analysis, Singular Value Decomposition)** Linear algebra hammer. Finds vectors with maximum projection of covariance. This is a linear transformation of the data.
- MDS **Multidimensional Scaling**: Finds an embedding (low-dimensional projection) that optimally preserves pairwise distances.
- tSNE (**t-distributed stochastic neighbor embedding**) : more elaborate transformations into an embedding space optimized to put same-label points near each other.

- The APIs to find the three embeddings look the same, and the lore says for small problems the three approaches give similar results.

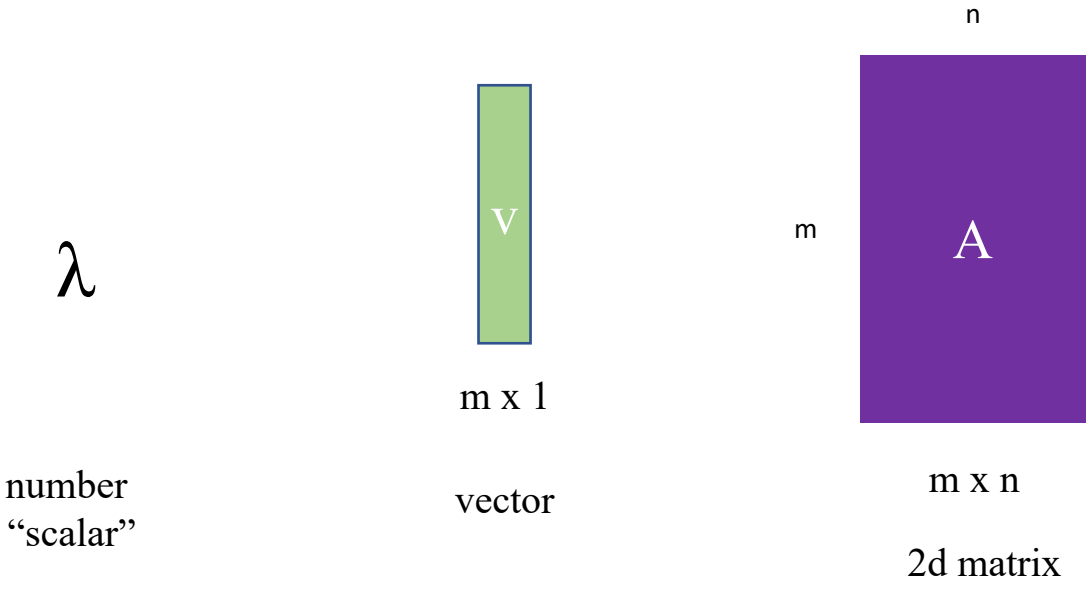
These library functions work by optimization

- Even though PCA is just math, the just-math calculation approach is often really expensive. (n^3 complexity, for instance, in finding determinant.)

- Particularly for large problems, close-enough algorithms can be made to work faster and on larger datasets than we could otherwise handle.

- These close-enough approximation algorithms often use optimization; don't be surprised when your matrix inversion says it is "searching for a good fit."

Linear algebra objects

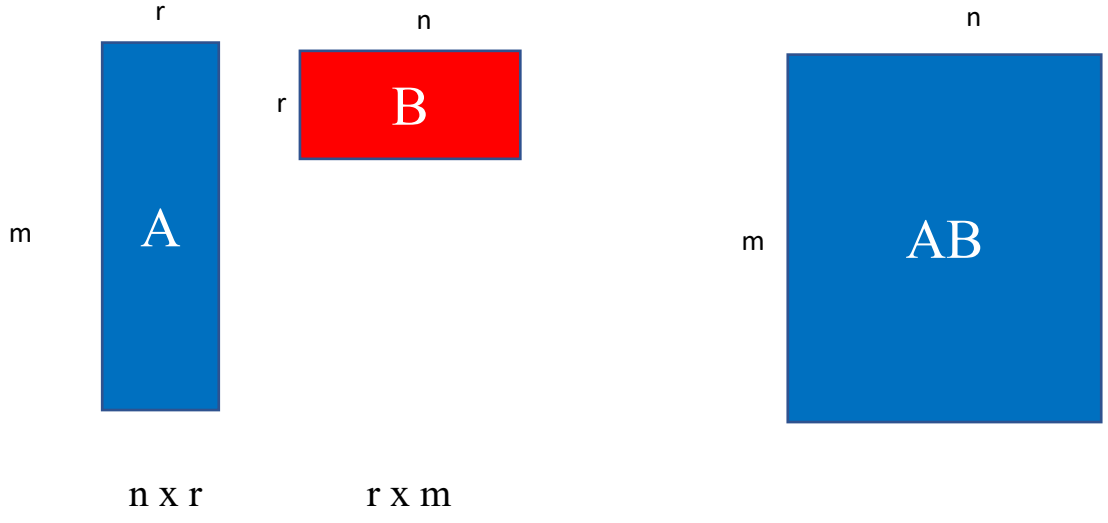


number
"scalar"

vector

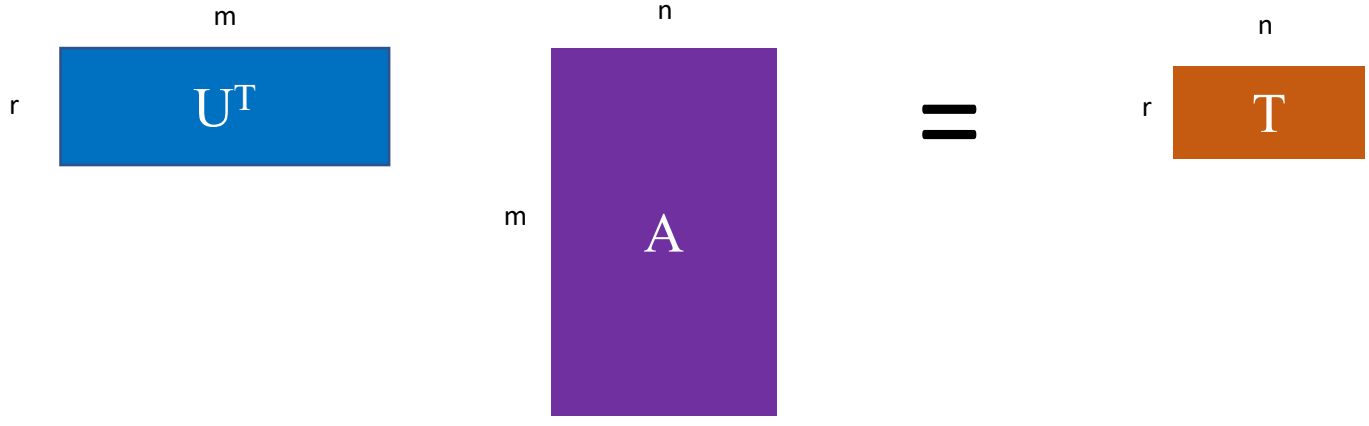
2d matrix

Matrix multiplication



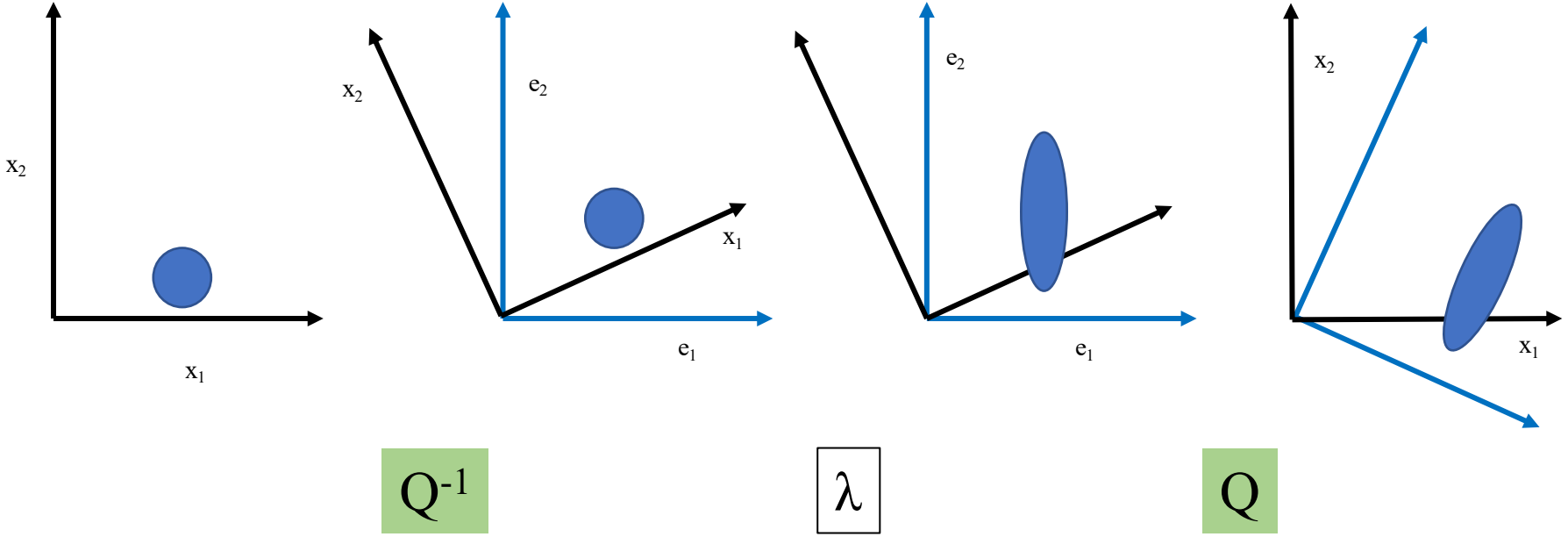
2d matrix

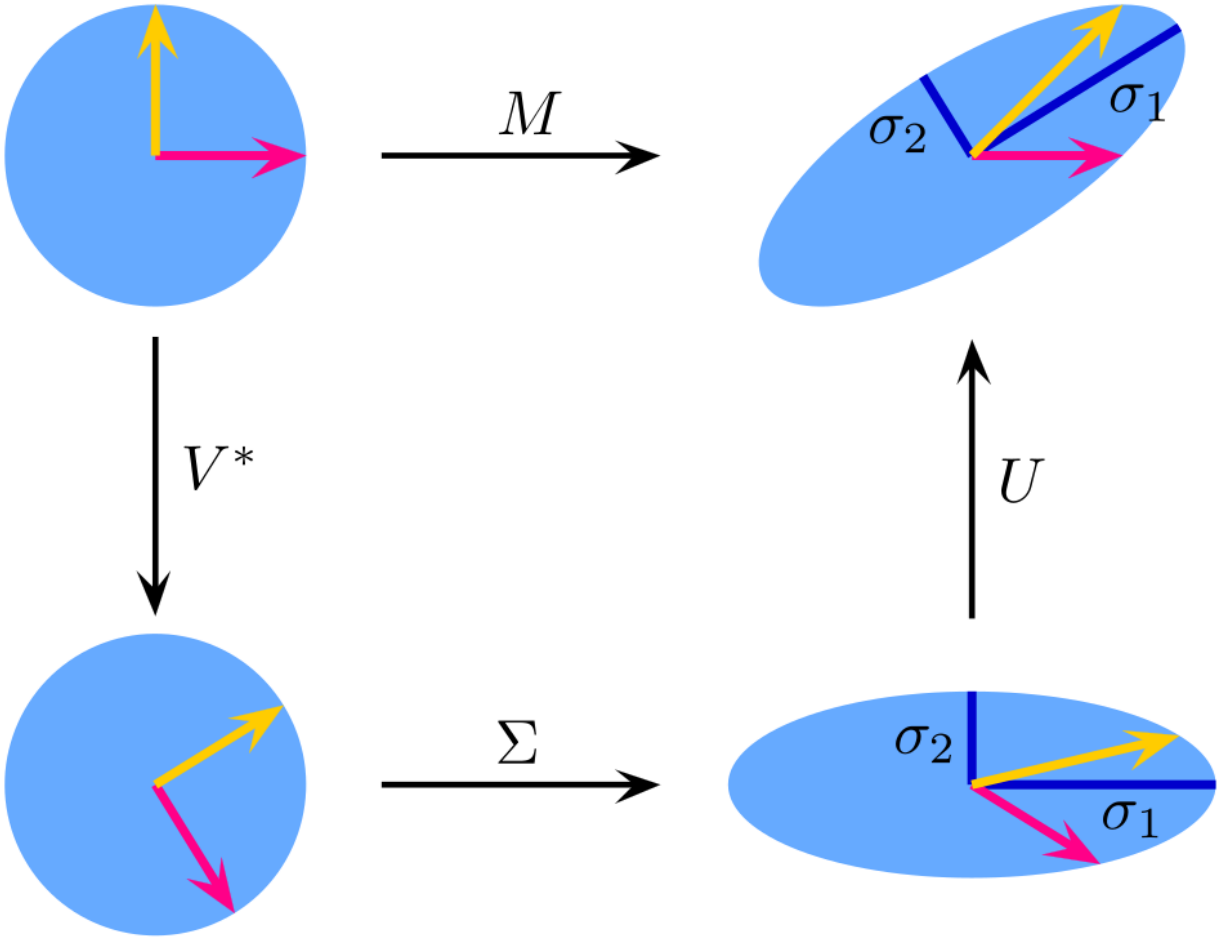
Principal Component analysis



This is the dimension reduction we might have asked for

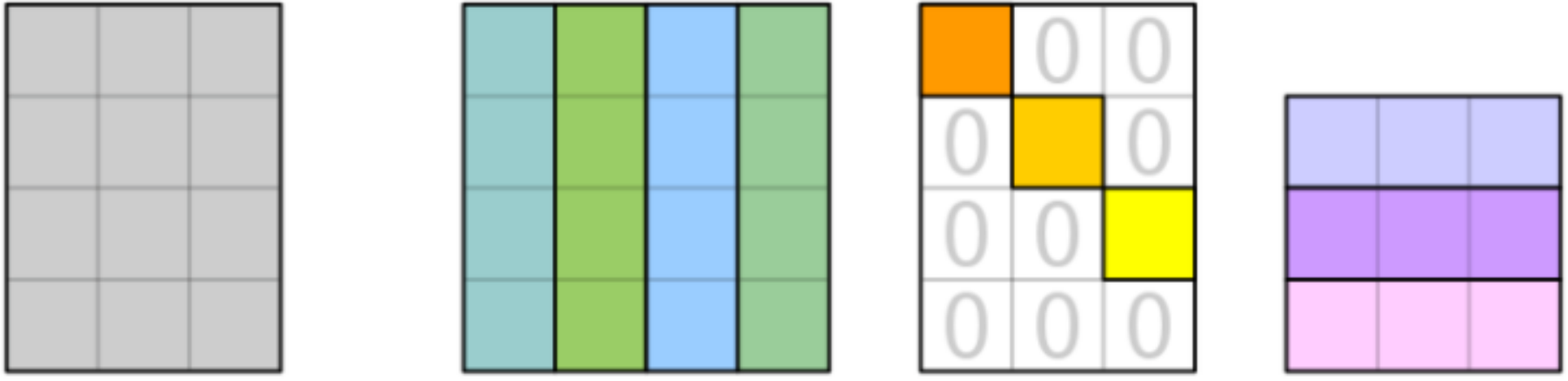
Transformation of x1, x2 using A





$$M = U \cdot \Sigma \cdot V^*$$

https://towardsdatascience.com/simple-svd-algorithms-13291ad2eef2



$$\mathbf{M} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{*}$$

$$m \times n \qquad m \times m \qquad m \times n \qquad n \times n$$

https://towardsdatascience.com/simple-svd-algorithms-13291ad2eef2

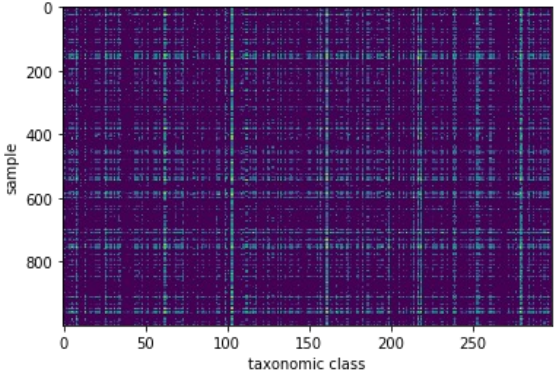
Dimension reduction

For a general matrix representing n observations of m traits:

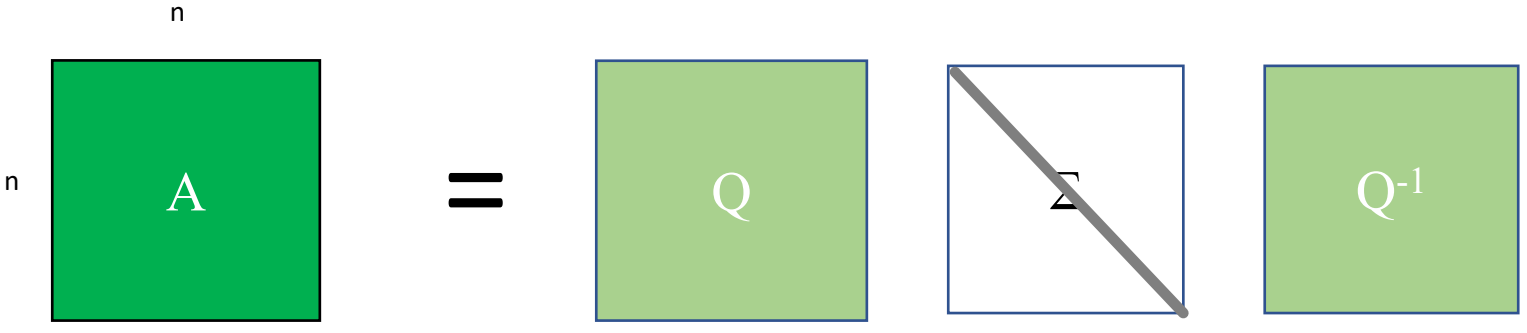
A (m x n)

It is unlikely that all of the elements in this matrix are independent; there is likely some “structure”

If I could replace A with an approximation of A that has fewer than n x m numbers, A will be cheaper to store, faster to copy, can run on computers with less memory.



Eigenvalue decomposition



square matrix

eigenvectors (orthogonal)

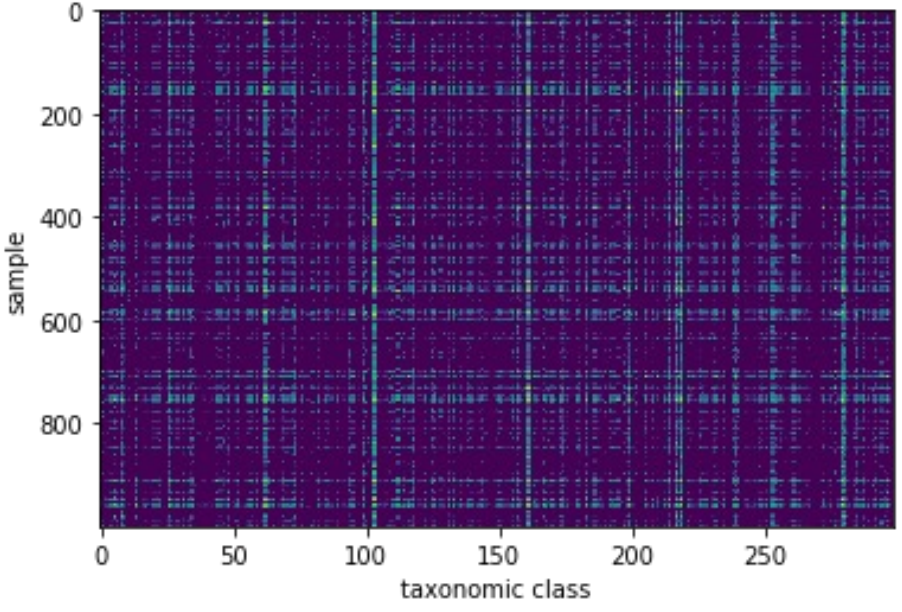
eigenvalues (diagonal)

eigenvectors (orthogonal)

A (n x m)

(rows x columns)

(n “samples” by m “features”)



(Index,)	Hydrozoa	Sphingobacteriia	Marattiopsida	Pedinophyceae	Chrysiogenetes (class)	Fusobacteria (class)	Amphibia
mgm4703051.3.mg	2	354	0	0	0	0	0
mgm4755207.3.mg	3430	0	25	14	33111	101765	5094
mgm4473196.3.mg	150	0	0	10	1265	487614	11598
mgm4679127.3.mg	0	245	0	0	0	0	0
mgm4529798.3.mg	2	4588	0	0	25	0	56

Now this looks like “structure”

Organizing data like this is deeply satisfying—it makes us think we have discovered truth.

Mature libraries to make exactly this sort of eye-candy graphs. This one is `sanborn.clustermap`

What if I told you linear algebra could do this?