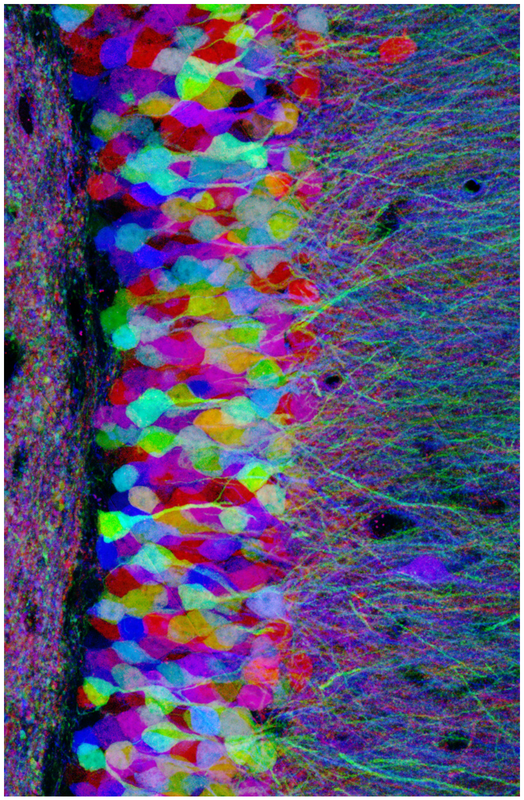
Lecture 1:
Introduction & Linear Algebra

April 1st, 2015

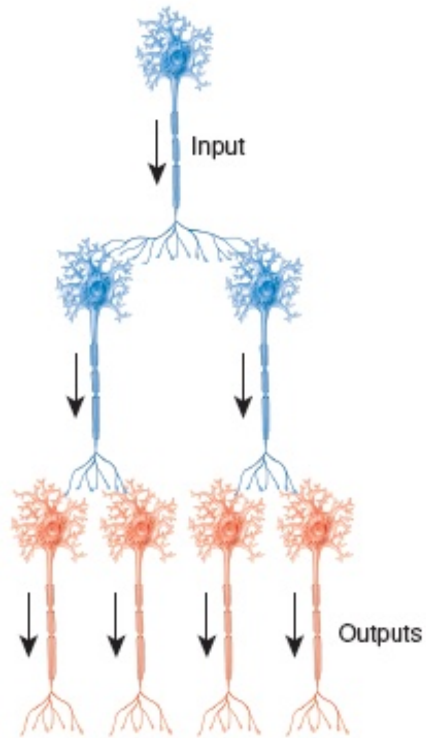
Lane McIntosh & Kiah Hardcastle

Math Tools for Neuroscience

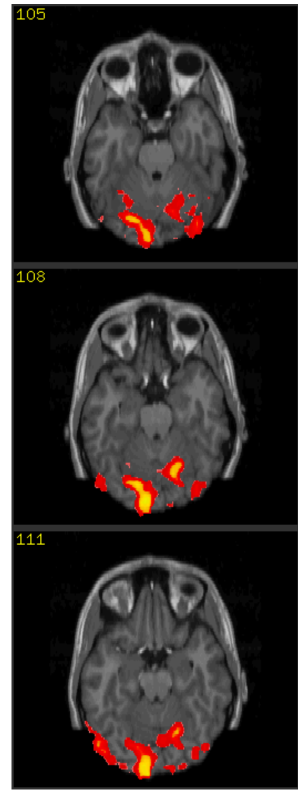
Welcome to NBIO 228!



cellular & molecular



circuits & systems

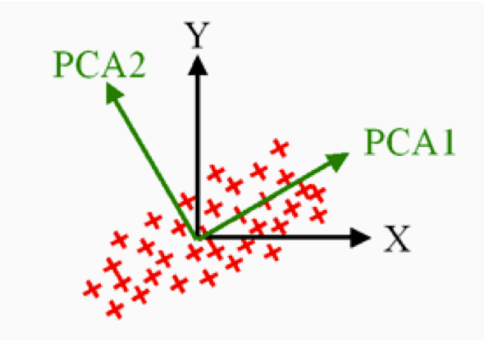


cognitive & behavioral

translational

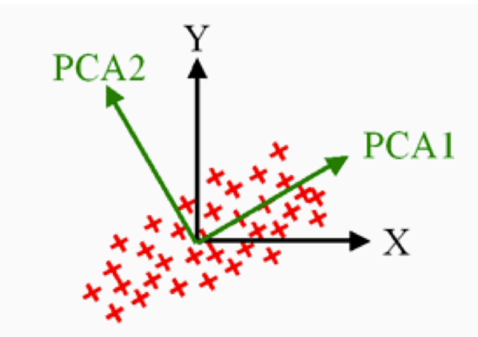
Who is this class designed for?

Topics we will cover this quarter

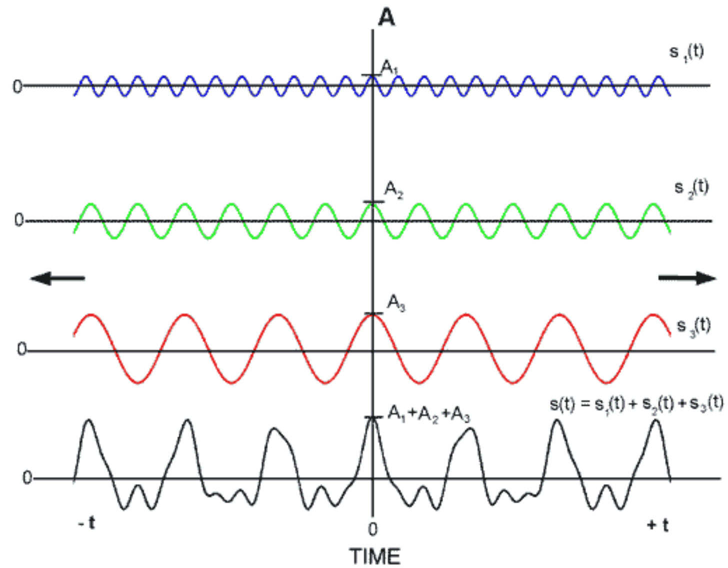


dimensionality reduction

Topics we will cover this quarter

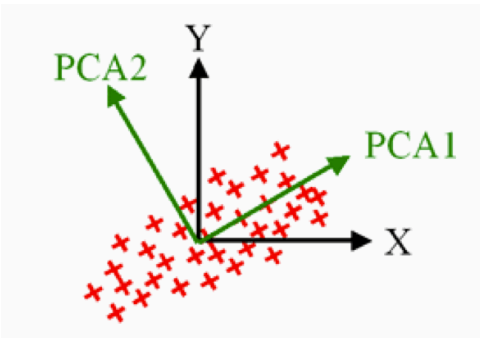


dimensionality reduction

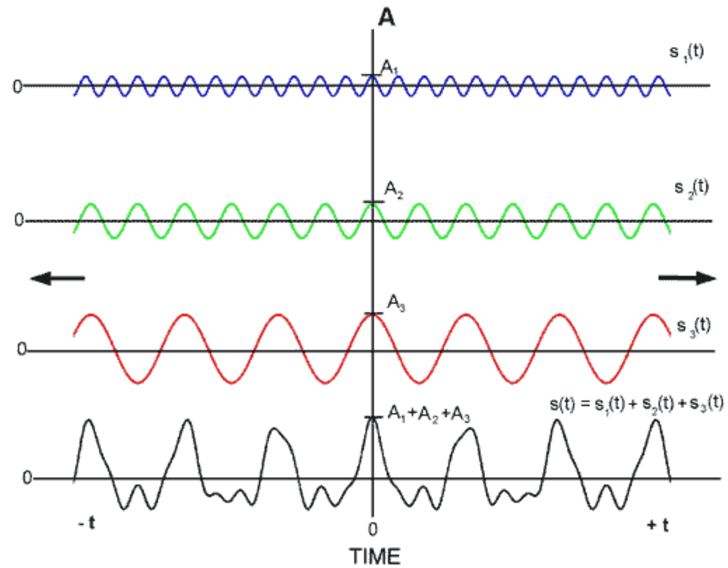


Fourier transforms, convolutions, and filtering out noise

Topics we will cover this quarter



dimensionality reduction



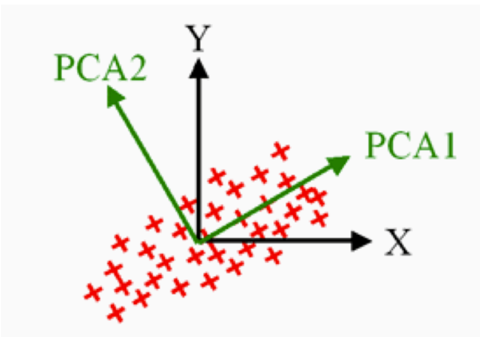
Fourier transforms, convolutions, and filtering out noise

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T e^{\frac{V - V_T}{\Delta_T}} - u + I$$

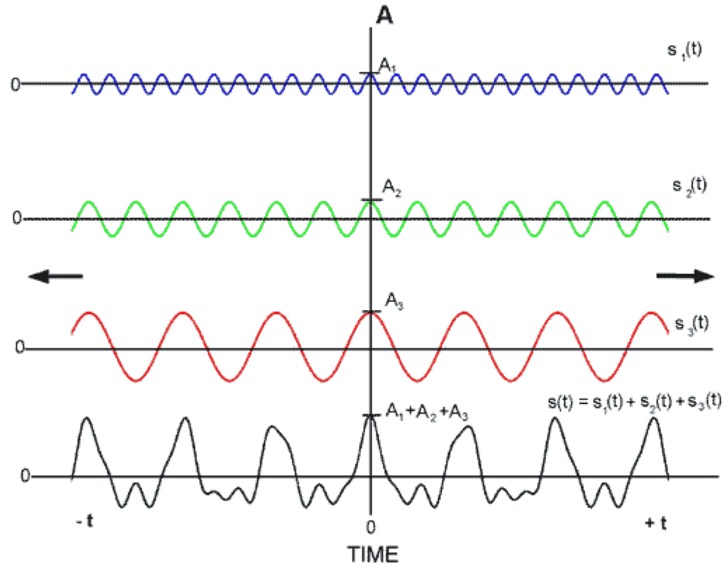
$$\tau_w \frac{du}{dt} = a(V - E_L) - u$$

differential equations and modeling

Topics we will cover this quarter



dimensionality reduction

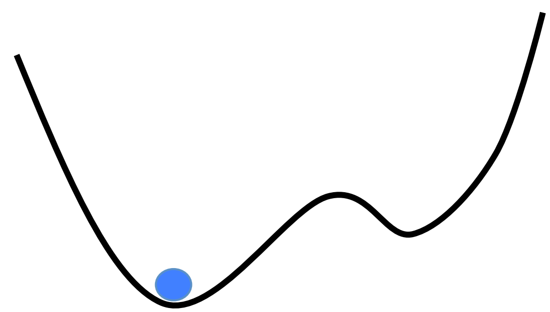


Fourier transforms, convolutions, and filtering out noise

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T e^{\frac{V-V_T}{\Delta_T}} - u + I$$

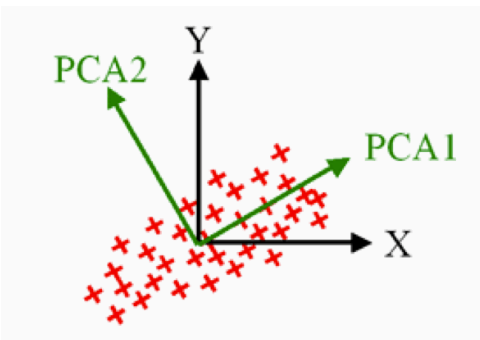
$$\tau_w \frac{du}{dt} = a(V - E_L) - u$$

differential equations and modeling

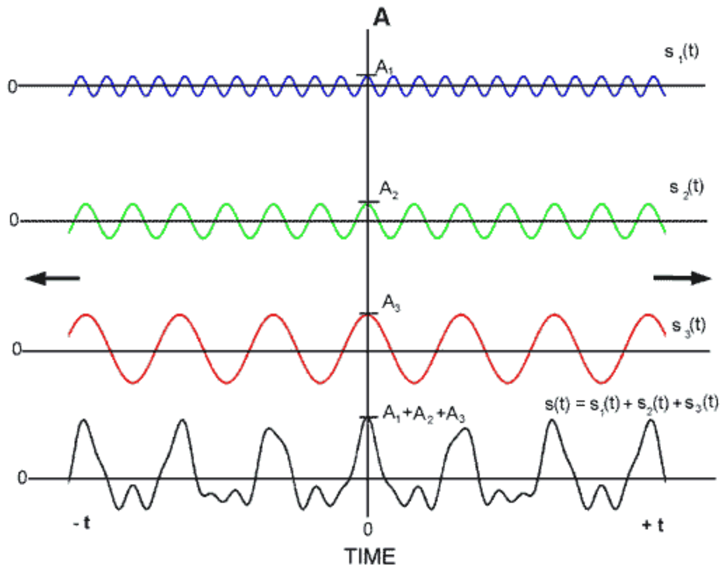


optimization

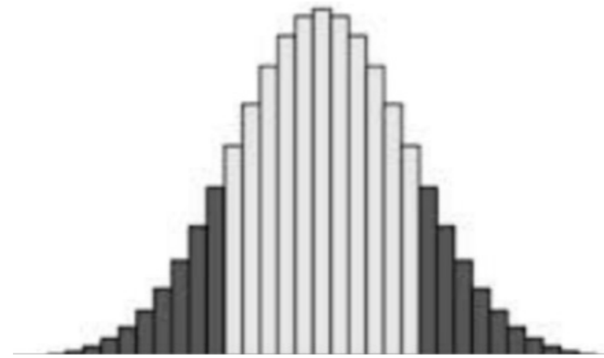
Topics we will cover this quarter



dimensionality reduction



Fourier transforms, convolutions, and filtering out noise

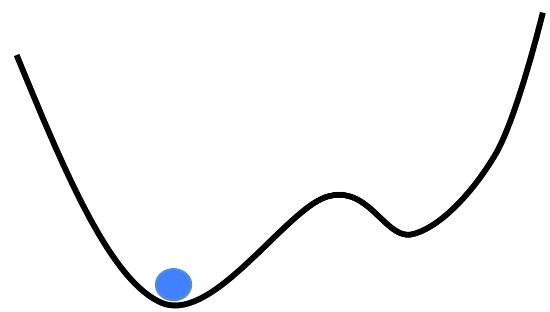


statistics, Bayesian probability, and information theory

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T e^{\frac{V-V_T}{\Delta_T}} - u + I$$

$$\tau_w \frac{du}{dt} = a(V - E_L) - u$$

differential equations and modeling



optimization

Today's lecture

NBIO 228 overview

Linear algebra

Who we are

Instructors	Contact
Kiah Hardcastle	khardcas@stanford.edu
Lane McIntosh	lmcintosh@stanford.edu

For announcements, lecture slides, syllabus, problem sets:

nbio228.stanford.edu

Projects

- Workshop your own data
- Apply a mathematical tool to a provided dataset
- Present a paper that uses a mathematical tool you want to understand better

Grading policy

- 3 or 4 Problem Sets: 0%
- Final Project: 100%
  - just show us that this course was useful!
  - either 5 min presentation or 1 page write-up

Today's lecture

NBIO 228 overview

Linear algebra

Today's lecture

NBIO 228 overview

Linear algebra

- Intuition about matrix operations
- Eigenvalues and eigenvectors
- Principal component analysis (PCA)

Why linear algebra?

Why linear algebra?

1.63	5.20	7.66	8.12	3.22
4.98	5.90	8.21	9.29	20.10
10.10	8.57	5.73	8.17	2.22
0.02	0.21	0.14	0.93	1.40
9.27	10.27	13.12	8.90	9.01
7.44	6.98	5.62	8.20	7.21
100.10	8.22	7.54	60.10	1.69
40.20	29.21	12.45	10.41	8.90
32.33	21.59	10.21	4.99	2.62
2.99	1.67	1.01	0.80	0.07

Datasets are matrices

time →

neuron 1	1.63	5.20	7.66	8.12	3.22
neuron 2	4.98	5.90	8.21	9.29	20.10
neuron 3	10.10	8.57	5.73	8.17	2.22
neuron 4	0.02	0.21	0.14	0.93	1.40
neuron 5	9.27	10.27	13.12	8.90	9.01
neuron 6	7.44	6.98	5.62	8.20	7.21
neuron 7	100.10	8.22	7.54	60.10	1.69
neuron 8	40.20	29.21	12.45	10.41	8.90
neuron 9	32.33	21.59	10.21	4.99	2.62
neuron 10	2.99	1.67	1.01	0.80	0.07

Datasets are matrices

time →

voxel 1	1.63	5.20	7.66	8.12	3.22
voxel 2	4.98	5.90	8.21	9.29	20.10
voxel 3	10.10	8.57	5.73	8.17	2.22
voxel 4	0.02	0.21	0.14	0.93	1.40
voxel 5	9.27	10.27	13.12	8.90	9.01
voxel 6	7.44	6.98	5.62	8.20	7.21
voxel 7	100.10	8.22	7.54	60.10	1.69
voxel 8	40.20	29.21	12.45	10.41	8.90
voxel 9	32.33	21.59	10.21	4.99	2.62
voxel 10	2.99	1.67	1.01	0.80	0.07

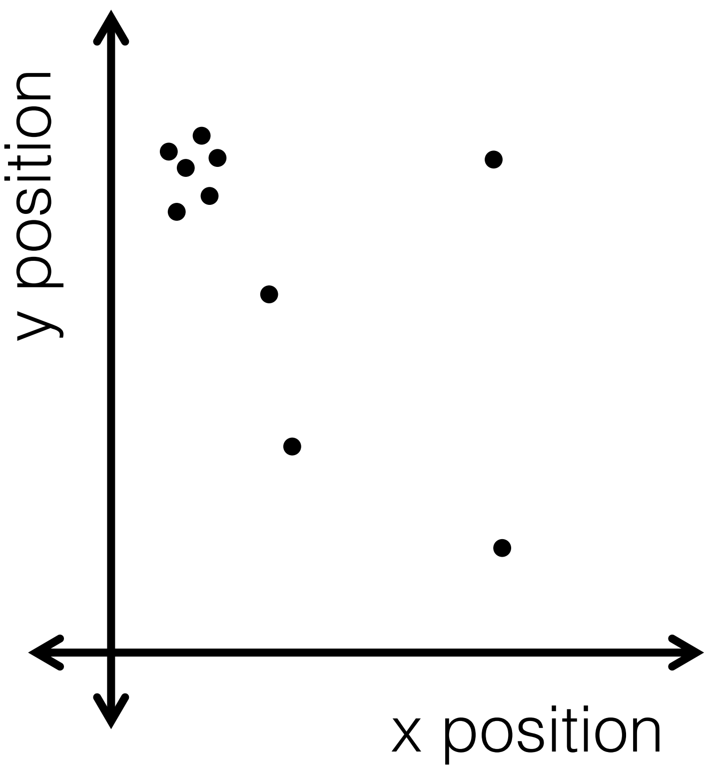
Datasets are matrices

patient →

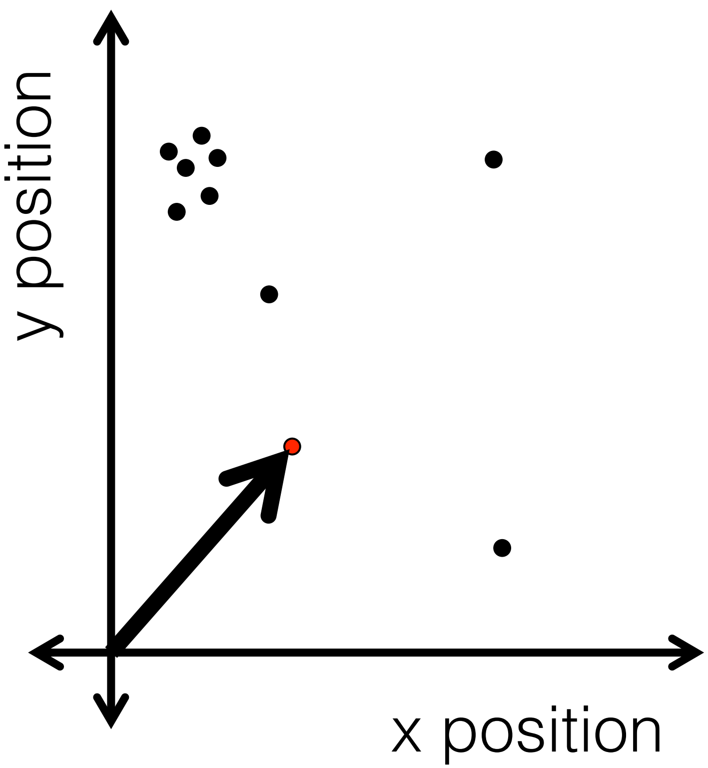
gene 1	1.63	5.20	7.66	8.12	3.22
gene 2	4.98	5.90	8.21	9.29	20.10
gene 3	10.10	8.57	5.73	8.17	2.22
gene 4	0.02	0.21	0.14	0.93	1.40
gene 5	9.27	10.27	13.12	8.90	9.01
gene 6	7.44	6.98	5.62	8.20	7.21
gene 7	100.10	8.22	7.54	60.10	1.69
gene 8	40.20	29.21	12.45	10.41	8.90
gene 9	32.33	21.59	10.21	4.99	2.62
gene 10	2.99	1.67	1.01	0.80	0.07

Part 1: Matrix Arithmetic
(w/ applications to an experiment)

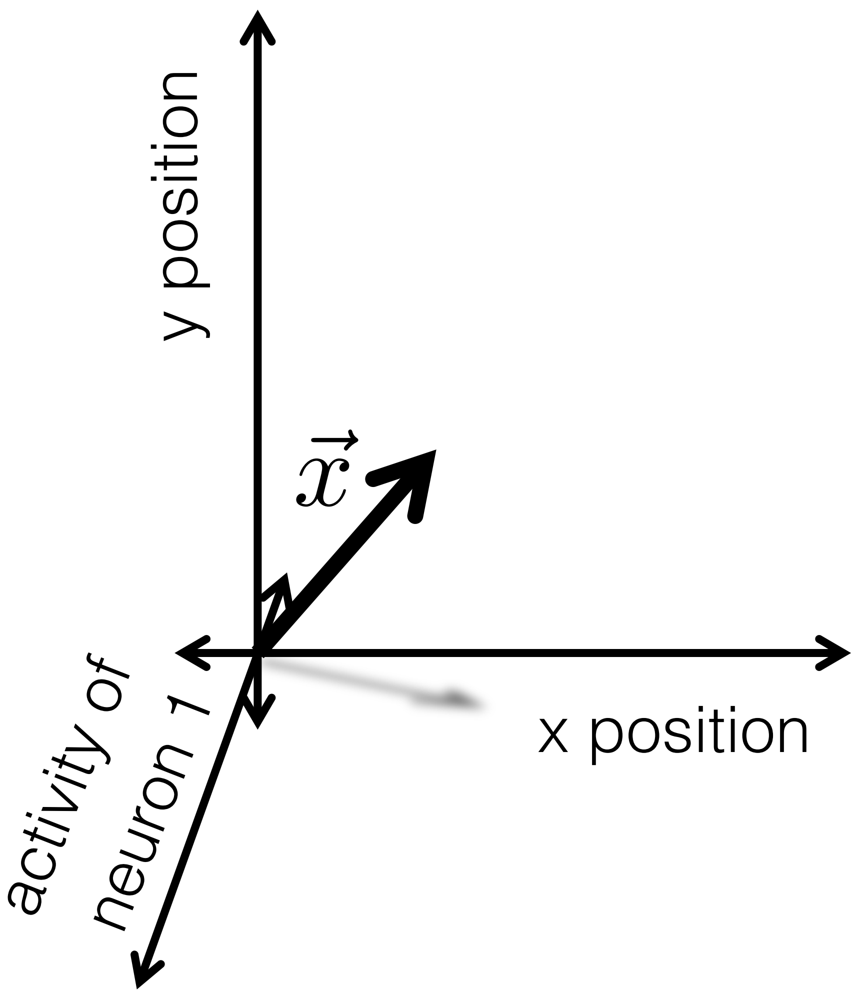
Measure a mouse’s location



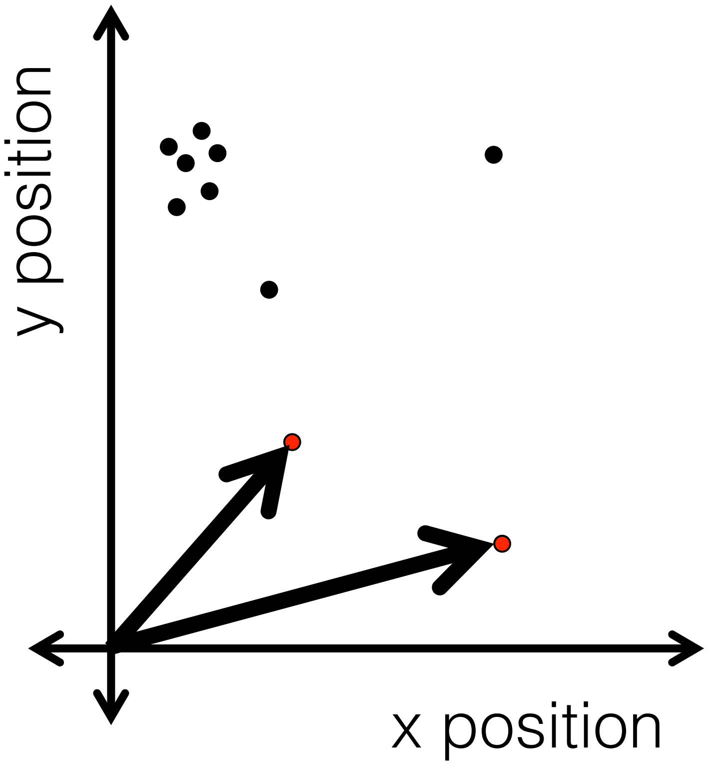
Measure a mouse's location



Measure a mouse's location and 10 of its neurons

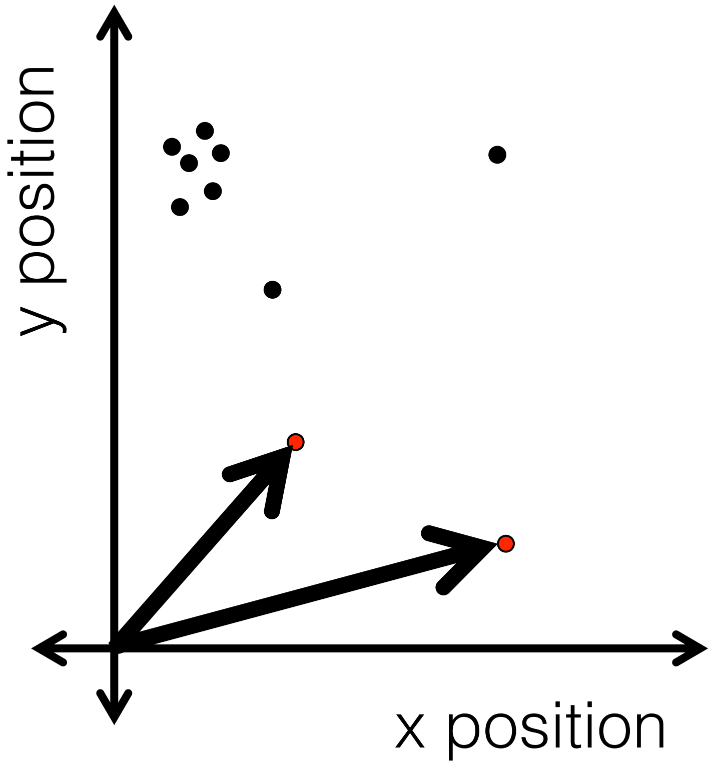


Matrix addition and subtraction



$$\begin{array}{r c}
 & \begin{array}{cc} \text{measurement 1} & \text{measurement 2} \end{array} \\
\begin{array}{r} \text{feature 1} \\ \text{feature 2} \end{array} & \begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix}
\end{array}$$

Matrix addition and subtraction

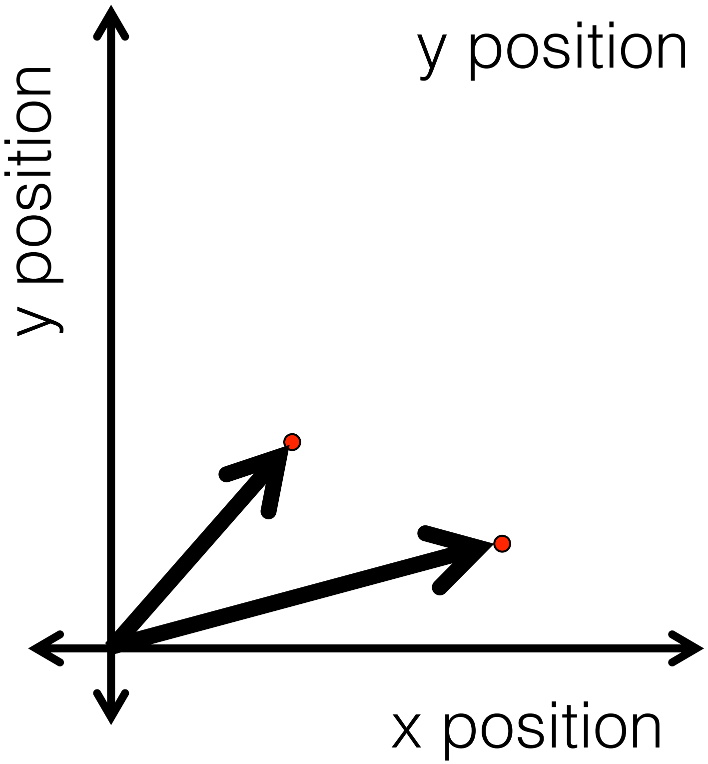


$$\begin{array}{lcc} & \text{measurement 1} & \text{measurement 2} \\ \text{x position} & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix}$}} \\ \text{y position} & & \end{array}$$

Matrix addition and subtraction

$$\begin{array}{l} \text{x position} \\ \text{y position} \end{array} \begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix} - \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$$

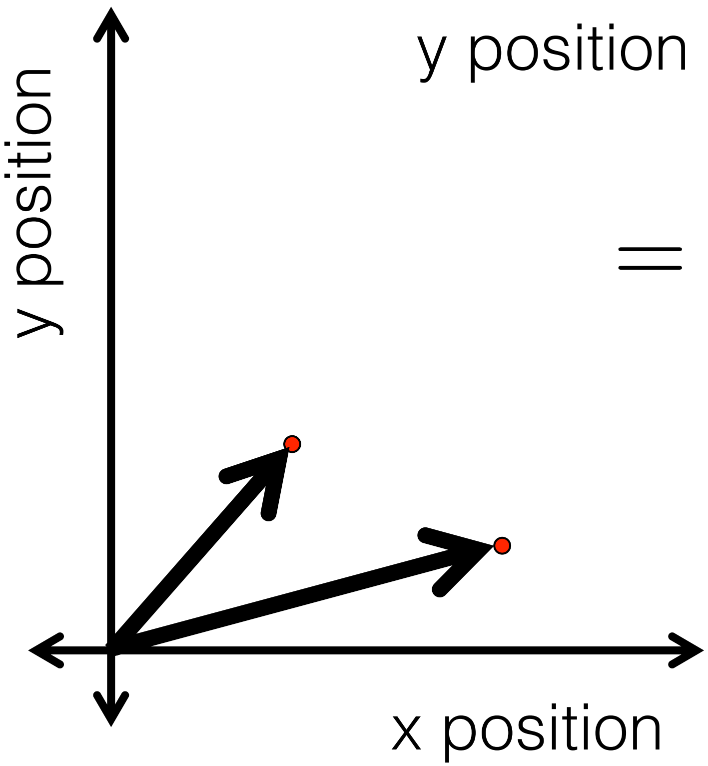
MATLAB: `mean(A, 2)`



Matrix addition and subtraction

$$\begin{array}{r c} \text{x position} & \multirow{2}{*}{} \\ \text{y position} & \end{array} \begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix} - \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix} - \begin{pmatrix} 3 & 3 \\ 1.5 & 1.5 \end{pmatrix}$$

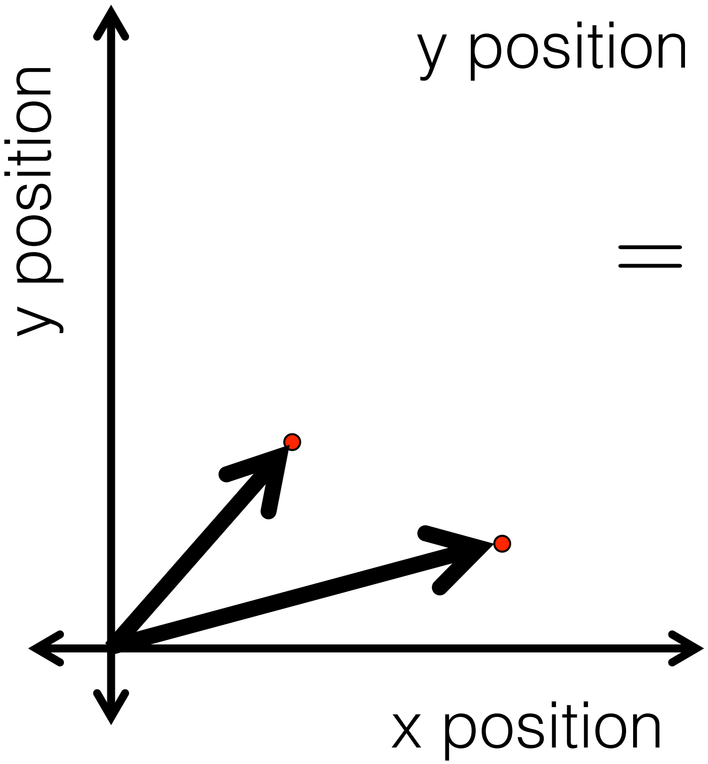


Matrix addition and subtraction

$$\begin{array}{l} \text{x position} \\ \text{y position} \end{array} \begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix} - \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix} - \begin{pmatrix} 3 & 3 \\ 1.5 & 1.5 \end{pmatrix}$$

$$= \begin{pmatrix} -1 & 1 \\ 0.5 & -0.5 \end{pmatrix}$$

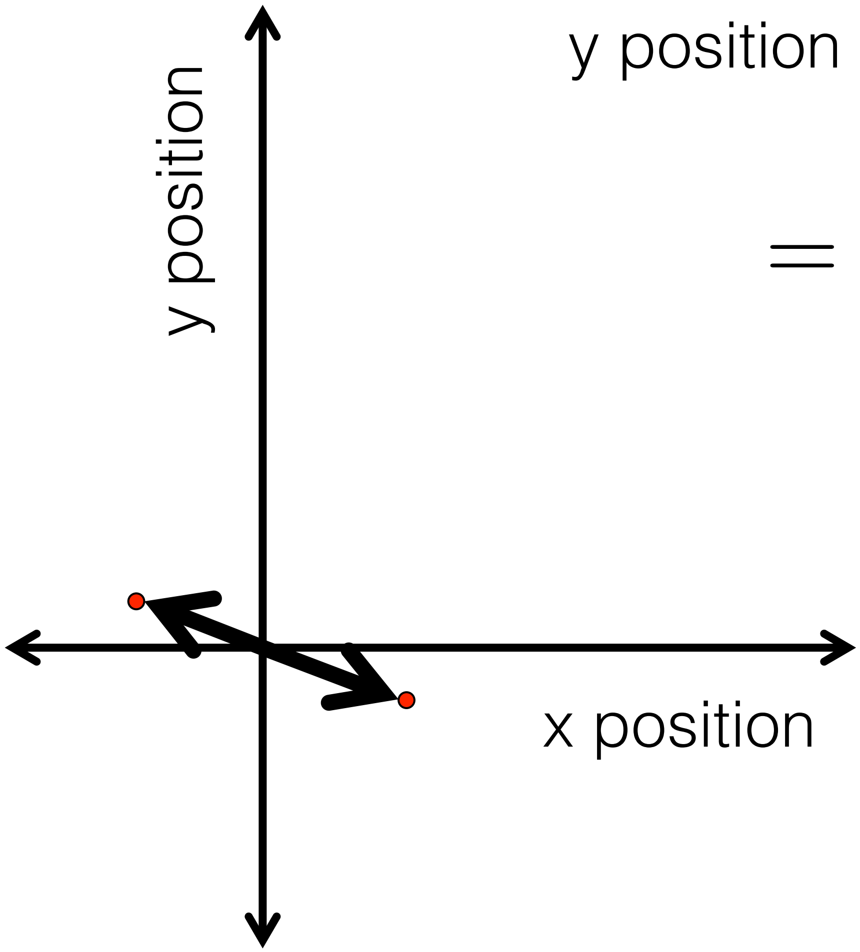


Matrix addition and subtraction

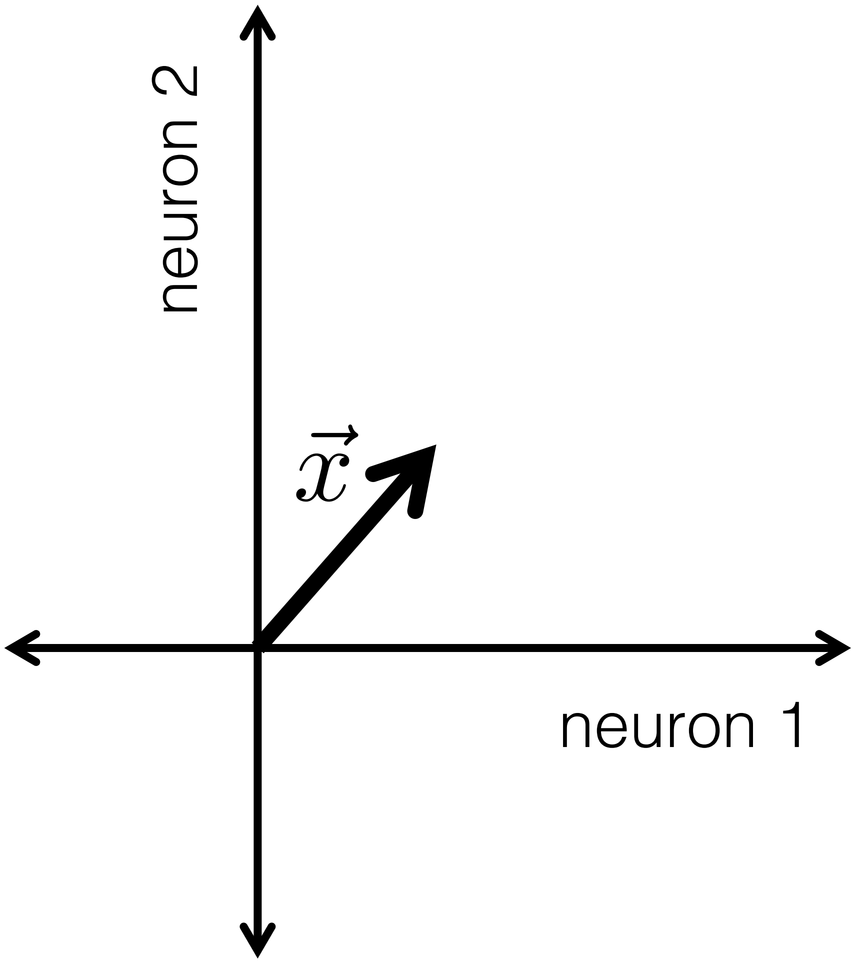
$$
\begin{array}{l}
\text{x position} \\
\text{y position}
\end{array}
\begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix} - \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}
$$

$$
= \begin{pmatrix} 2 & 4 \\ 2 & 1 \end{pmatrix} - \begin{pmatrix} 3 & 3 \\ 1.5 & 1.5 \end{pmatrix}
$$

$$
= \begin{pmatrix} -1 & 1 \\ 0.5 & -0.5 \end{pmatrix}
$$

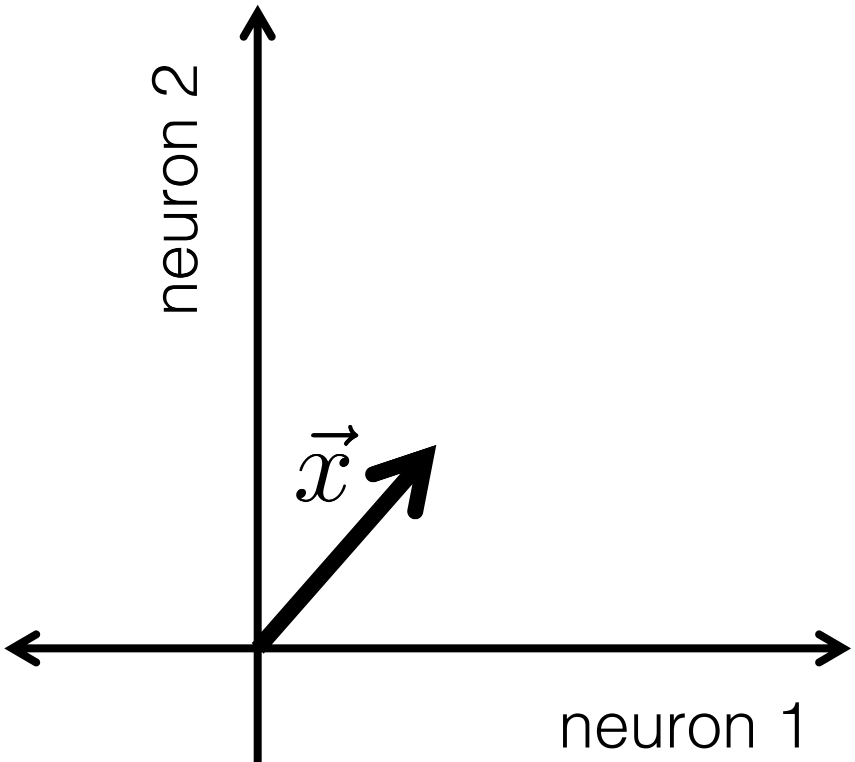


Scalar times vector



Measure 10 neurons at 100 HZ

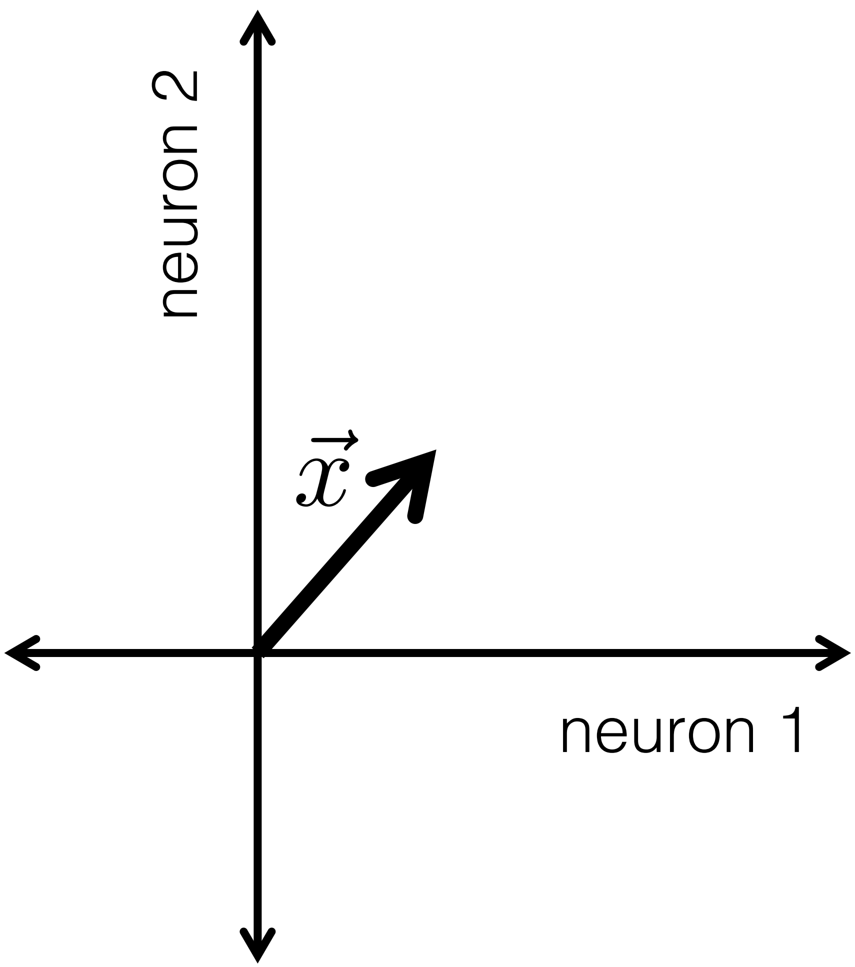
Scalar times vector



Measure 10 neurons at 100 HZ

Number of spikes per 10ms bin

Scalar times vector

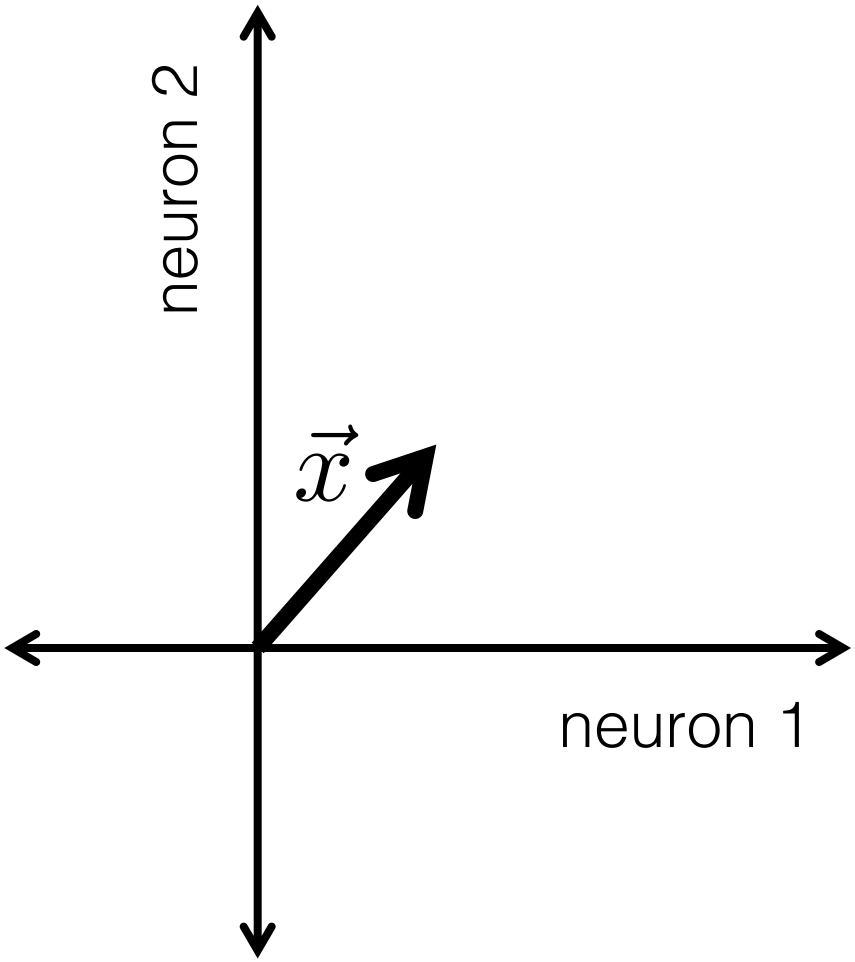


Measure 10 neurons at 100 HZ

Number of spikes per 10ms bin

How to convert to a firing rate in spikes/sec?

Scalar times vector



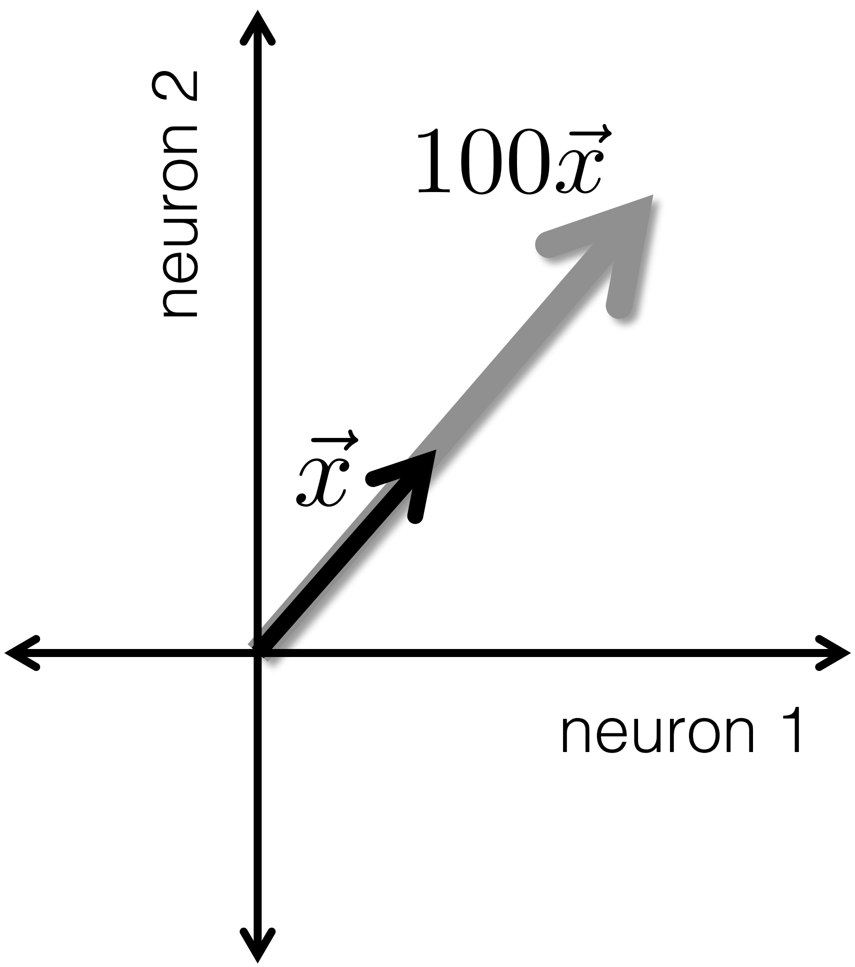
Measure 10 neurons at 100 HZ

Number of spikes per 10ms bin

How to convert to a firing rate in spikes/sec?

➡ Multiply each measurement by 100

Scalar times vector



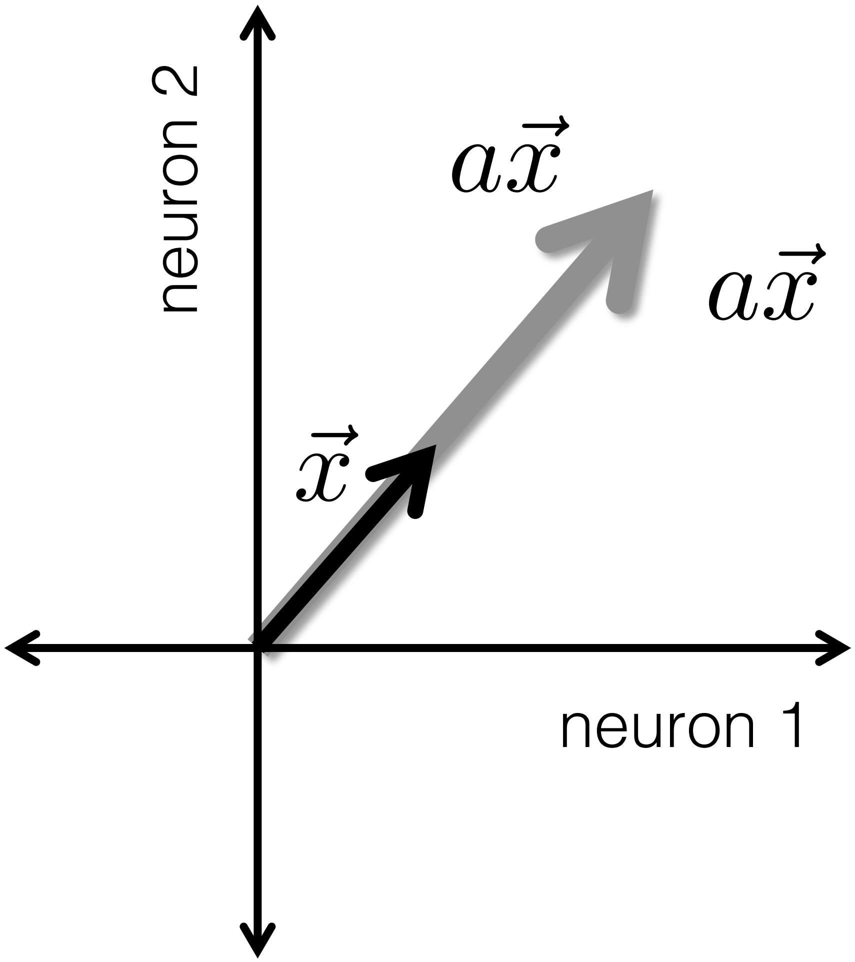
Measure 10 neurons at 100 HZ

Number of spikes per 10ms bin

How to convert to a firing rate in spikes/sec?

 Multiply each measurement by 100

Scalar times vector



$$a\vec{x} = a\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} ax_1 \\ ax_2 \\ \vdots \\ ax_N \end{pmatrix}$$

Product of Two Vectors

Three ways to multiply

- Element-by-element
- Inner product
- Outer product

Element-by-element product

(Hadamard product)

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} .* \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_2 b_2 \end{pmatrix}$$

- Element-wise multiplication (`.*` in MATLAB)

Element-by-element product

(Hadamard product)

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} .* \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_2 b_2 \end{pmatrix}$$

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} .^{*} \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_2 b_2 \\ a_3 b_3 & a_4 b_4 \end{pmatrix}$$

- Element-wise multiplication (`.*` in MATLAB)

Element-by-element product

Example

$$\begin{array}{l} \text{neuron 1} \\ \text{neuron 2} \end{array} \begin{pmatrix} 102 & 34 \\ 2 & 3 \end{pmatrix}$$

Element-by-element product

Example

$$
\begin{array}{lc}
\text{neuron 1} & \multirow{2}{*}{$\begin{pmatrix} 102 & 34 \\ 2 & 3 \end{pmatrix}$} \\
\text{neuron 2} &
\end{array}
$$

high standard deviation (~48)

low standard deviation (~0.7)

Element-by-element product

Example

$$\begin{array}{l} \text{neuron 1} \\ \text{neuron 2} \end{array} \begin{pmatrix} 102 & 34 \\ 2 & 3 \end{pmatrix} \begin{array}{l} \text{high standard deviation (~48)} \\ \text{low standard deviation (~0.7)} \end{array}$$

$$\begin{pmatrix} 102 & 34 \\ 2 & 3 \end{pmatrix} ./ \begin{pmatrix} \sigma_1 & \sigma_1 \\ \sigma_2 & \sigma_2 \end{pmatrix} = \begin{pmatrix} 102/\sigma_1 & 34/\sigma_1 \\ 2/\sigma_2 & 3/\sigma_2 \end{pmatrix}$$

- Element-wise division (`./` in MATLAB)

Element-by-element product

Example

$$\begin{array}{l} \text{neuron 1} \\ \text{neuron 2} \end{array} \begin{pmatrix} 102 & 34 \\ 2 & 3 \end{pmatrix} \begin{array}{l} \text{high standard deviation (~48)} \\ \text{low standard deviation (~0.7)} \end{array}$$

$$\begin{pmatrix} 102 & 34 \\ 2 & 3 \end{pmatrix} ./ \begin{pmatrix} \sigma_1 & \sigma_1 \\ \sigma_2 & \sigma_2 \end{pmatrix} = \begin{pmatrix} 102/\sigma_1 & 34/\sigma_1 \\ 2/\sigma_2 & 3/\sigma_2 \end{pmatrix}$$

$$= \begin{pmatrix} 2.1 & 0.7 \\ 2.9 & 4.2 \end{pmatrix}$$

- Element-wise division (`./` in MATLAB)

Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

$$= \sum_{i=1}^{N} x_i y_i$$

Multiplication:
Dot product (inner product)

$$\vec{x} \cdot \vec{y} =$$

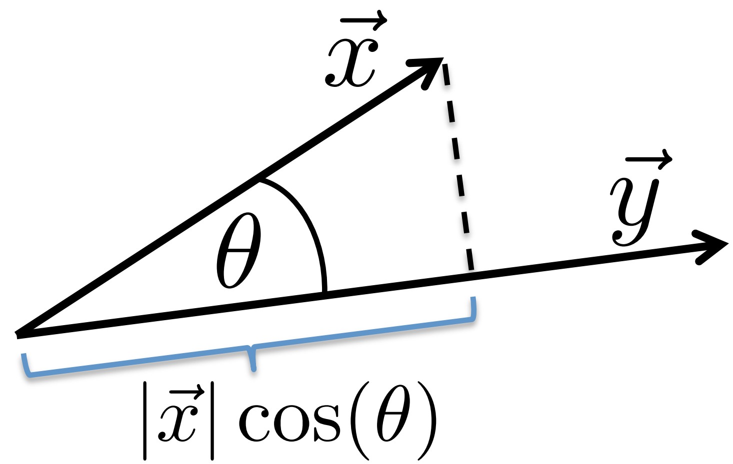
$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

1 X N　　　　N X 1　　　　1 X 1

- MATLAB: ‘inner matrix dimensions must agree’

Outer dimensions give size of resulting matrix

Dot product geometric intuition: "Overlap" of 2 vectors

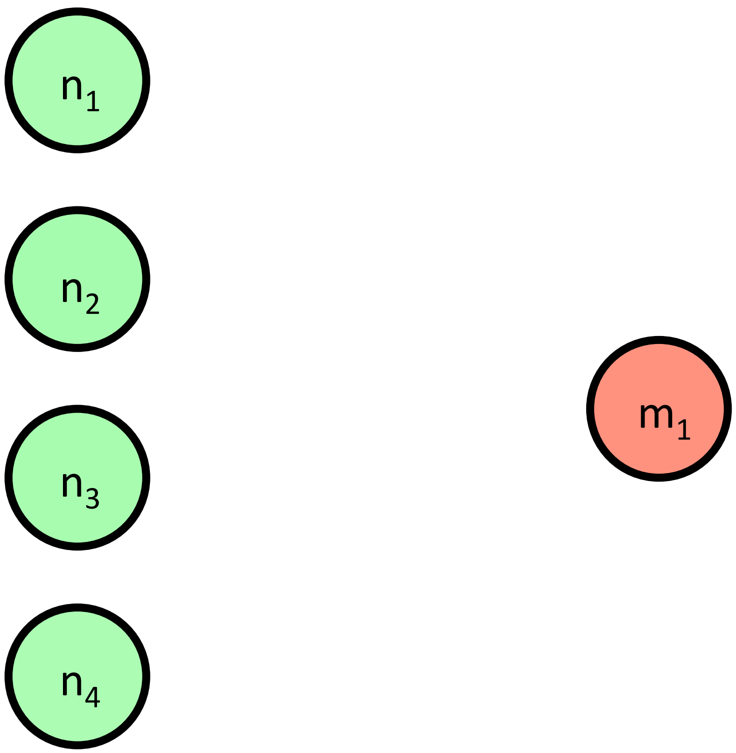


$$\vec{x} \cdot \vec{y} = |\vec{x}||\vec{y}|\cos(\theta)$$

Multiplication:

Dot product (inner product) Example 1

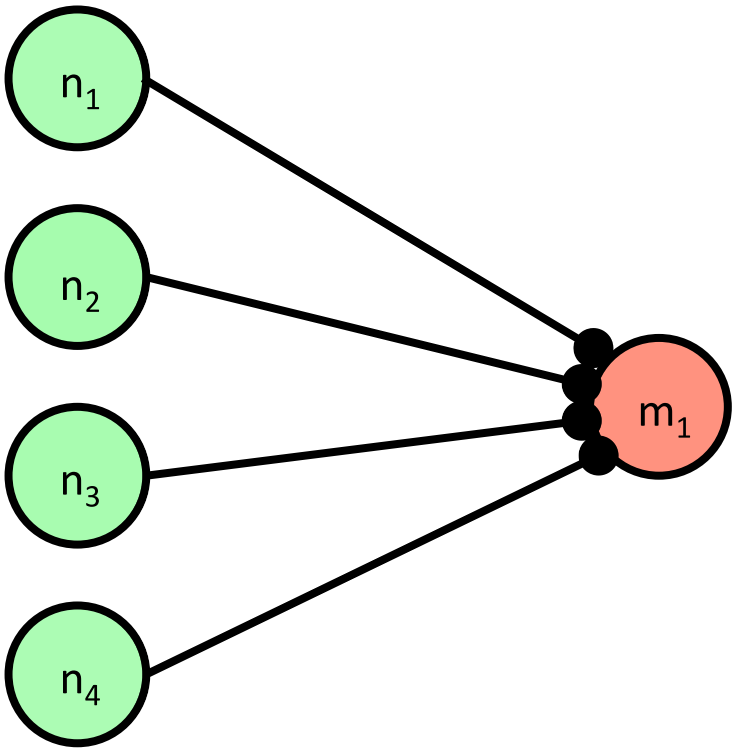
weighted average



Multiplication:

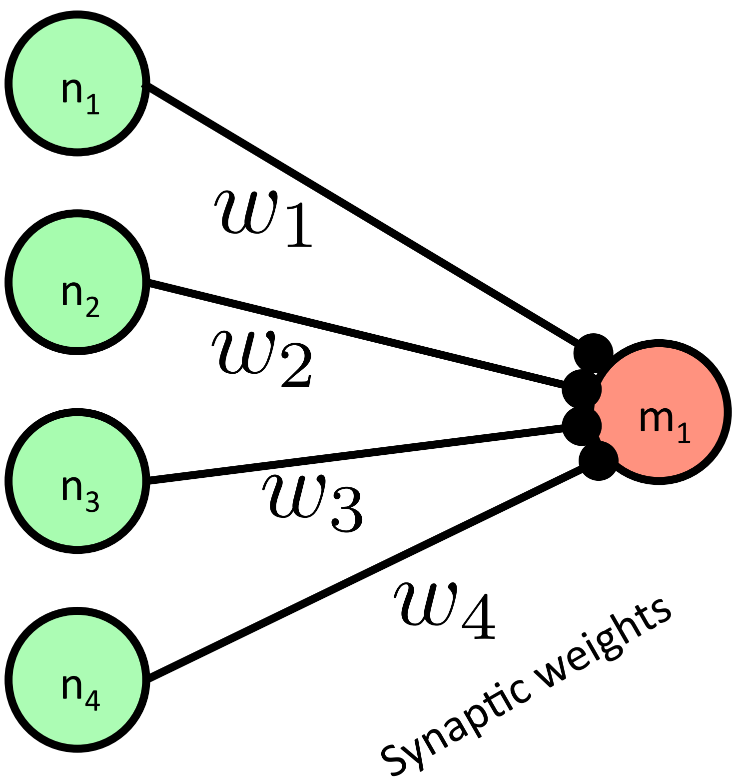
Dot product (inner product) Example 1

weighted average



Multiplication:
Dot product (inner product) Example 1

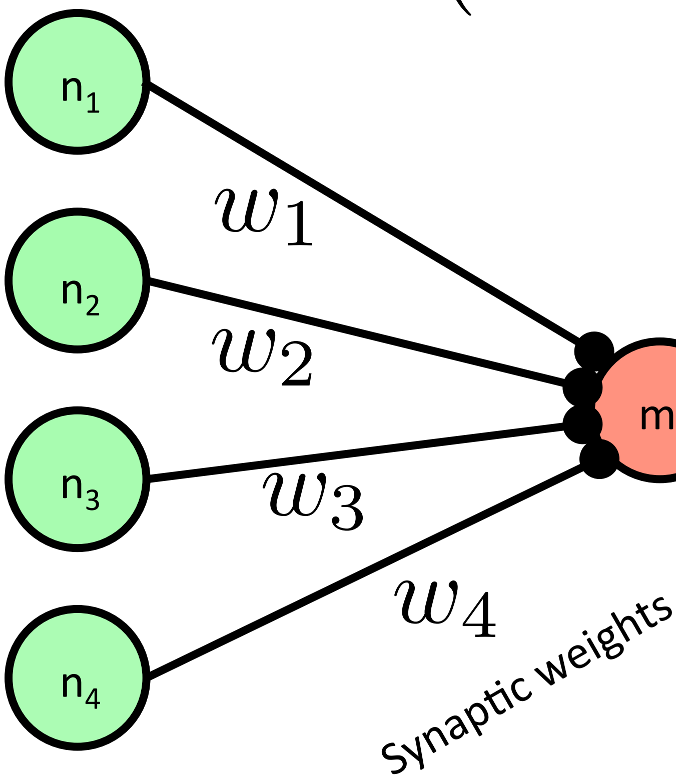
weighted average



Multiplication:
Dot product (inner product) Example 1

weighted average

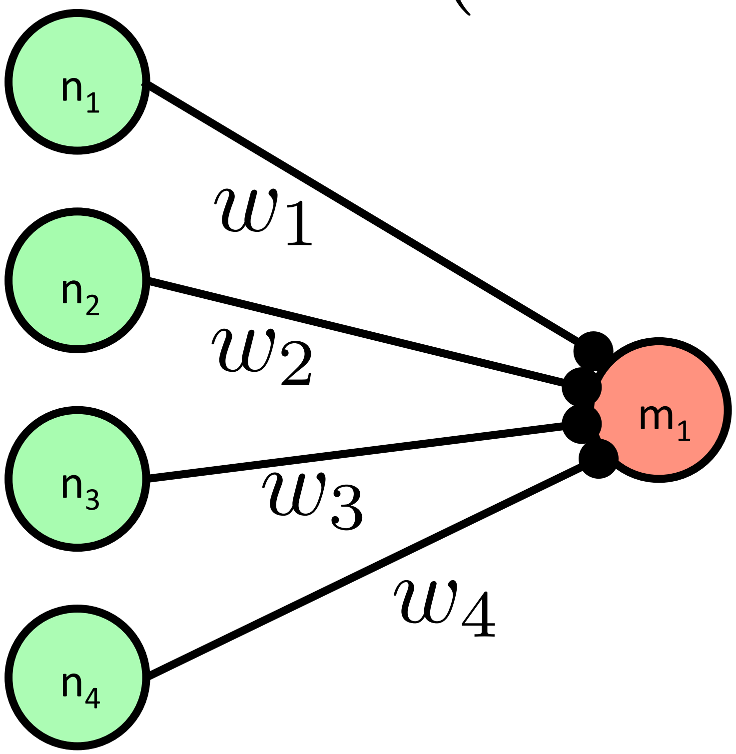
$$m_1 = \begin{pmatrix} w_1 & w_2 & w_3 & w_4 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix}$$



Multiplication:
Dot product (inner product) Example 1

weighted average

$$m_1 = \begin{pmatrix} w_1 & w_2 & w_3 & w_4 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix}$$



$$= w_1 n_1 + w_2 n_2 + w_3 n_3 + w_4 n_4$$

$$= w^T n$$ (`w'n` in MATLAB)

Multiplication:
Dot product (inner product) Example 2

linear regression

patient $\longrightarrow$

gene 1	1.63	5.20	7.66	8.12	3.22
gene 2	4.98	5.90	8.21	9.29	20.10
gene 3	10.10	8.57	5.73	8.17	2.22
gene 4	0.02	0.21	0.14	0.93	1.40
gene 5	9.27	10.27	13.12	8.90	9.01
gene 6	7.44	6.98	5.62	8.20	7.21

Multiplication:
Dot product (inner product) Example 2

linear regression

patient $\longrightarrow$

gene 1	1.63	5.20	7.66	8.12	3.22
gene 2	4.98	5.90	8.21	9.29	20.10
gene 3	10.10	8.57	5.73	8.17	2.22
gene 4	0.02	0.21	0.14	0.93	1.40
gene 5	9.27	10.27	13.12	8.90	9.01
gene 6	7.44	6.98	5.62	8.20	7.21

For each patient, you also measure their Asperger’s disorder quotient

Multiplication:
Dot product (inner product) Example 2

linear regression

patient $\longrightarrow$

gene 1	1.63	5.20	7.66	8.12	3.22
gene 2	4.98	5.90	8.21	9.29	20.10
gene 3	10.10	8.57	5.73	8.17	2.22
gene 4	0.02	0.21	0.14	0.93	1.40
gene 5	9.27	10.27	13.12	8.90	9.01
gene 6	7.44	6.98	5.62	8.20	7.21
score	2	0	9	44	48

Multiplication:
Dot product (inner product) Example 2

linear regression

patient $\longrightarrow$

gene 1	1.63	5.20	7.66	8.12	3.22
gene 2	4.98	5.90	8.21	9.29	20.10
gene 3	10.10	8.57	5.73	8.17	2.22
gene 4	0.02	0.21	0.14	0.93	1.40
gene 5	9.27	10.27	13.12	8.90	9.01
gene 6	7.44	6.98	5.62	8.20	7.21
score	2	0	9	44	48

$$\text{score} = w_1 \text{gene}_1 + w_2 \text{gene}_2 + \cdots + w_6 \text{gene}_6$$

Multiplication:
Dot product (inner product) Example 2

linear regression

patient $\longrightarrow$

gene 1	1.63	5.20	7.66	8.12	3.22
gene 2	4.98	5.90	8.21	9.29	20.10
gene 3	10.10	8.57	5.73	8.17	2.22
gene 4	0.02	0.21	0.14	0.93	1.40
gene 5	9.27	10.27	13.12	8.90	9.01
gene 6	7.44	6.98	5.62	8.20	7.21
score	2	0	9	44	48

$$\text{score} = w^T \vec{\text{genes}}$$

Multiplication:
Dot product (inner product) Example 2

linear regression

$$\begin{pmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{pmatrix} \begin{array}{c} \begin{pmatrix} 8.12 \\ 9.29 \\ 8.17 \\ 0.93 \\ 8.90 \\ 8.20 \end{pmatrix} \end{array} \begin{array}{l} \text{gene 1} \\ \text{gene 2} \\ \text{gene 3} \\ \text{gene 4} \\ \text{gene 5} \\ \text{gene 6} \end{array}$$

$$\text{score} = w^T \vec{\text{genes}}$$

Multiplication:
Dot product (inner product) Example 2

linear regression

patient →

gene 1	1.63	5.20	7.66	8.12	3.22
gene 2	4.98	5.90	8.21	9.29	20.10
gene 3	10.10	8.57	5.73	8.17	2.22
gene 4	0.02	0.21	0.14	0.93	1.40
gene 5	9.27	10.27	13.12	8.90	9.01
gene 6	7.44	6.98	5.62	8.20	7.21
score	2	0	9	44	48

$$\text{score} = w^T \vec{\text{genes}}$$

(`regress(score',X')` in MATLAB to get w's, where `X` is the full genes by patients matrix)

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_M \\ x_2y_1 & x_2y_2 & \cdots & x_2y_M \\ \vdots & \vdots & \ddots & \dots \\ x_Ny_1 & x_Ny_2 & \cdots & x_Ny_M \end{pmatrix}$$

N X 1 1 X M N X M

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \dots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \ldots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \vdots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

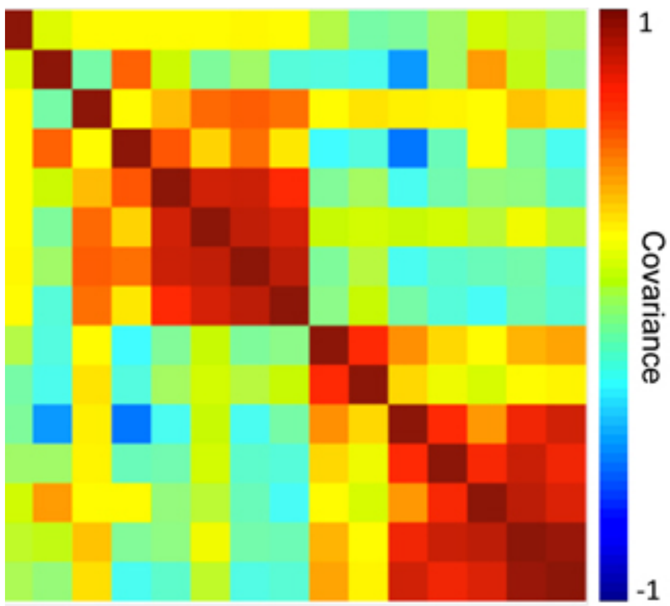
Multiplication: **Outer product**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_M \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \vdots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

- Note: each column or each row is a multiple of the others

Multiplication: **Outer product**

Example: Covariance Matrices



$$= \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_M \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_M \\ \vdots & \vdots & \ddots & \vdots \\ x_N y_1 & x_N y_2 & \cdots & x_N y_M \end{pmatrix}$$

- When $\vec{x} = \vec{y}$ and $\vec{x}$ has an average of zero, this outer product is called the covariance matrix

Matrix times vector

$$\overrightarrow{y} = \overleftrightarrow{W}\,\overrightarrow{x}$$

Matrix times vector

$$\overrightarrow{y} = \overleftrightarrow{W}\,\overrightarrow{x}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

Matrix times vector

$$\overrightarrow{y} = \overleftrightarrow{W}\,\overrightarrow{x}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

M X 1   M X N   N X 1

Matrix times vector: inner product interpretation

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Rule: the i^{th} element of **y** is the dot product of the i^{th} row of W with **x**

Matrix times vector: inner product interpretation

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Rule: the i^{th} element of **y** is the dot product of the i^{th} row of W with **x**

Matrix times vector:
inner product interpretation

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Rule: the i^{th} element of **y** is the dot product of the i^{th} row of W with **x**

Matrix times vector:
inner product interpretation

$$
\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}
$$

- Rule: the i^{th} element of **y** is the dot product of the i^{th} row of W with **x**

Matrix times vector:
inner product interpretation

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{i1} & W_{i2} & \cdots & W_{iN} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Rule: the i^{th} element of **y** is the dot product of the i^{th} row of W with **x**

Matrix times vector:
outer product interpretation

$$
\begin{array}{l}
\overrightarrow{W}^{(1)} \\
\downarrow \\
\begin{pmatrix}
W_{11} & W_{12} & \cdots & W_{1N} \\
W_{21} & W_{22} & \cdots & W_{2N} \\
\vdots & \vdots & & \vdots \\
W_{M1} & W_{M2} & \cdots & W_{MN}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_N
\end{pmatrix}
=
\end{array}
$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Matrix times vector:
outer product interpretation

$$\begin{array}{l} \quad\overrightarrow{W}^{(1)} \\ \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \end{array} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = x_1 \overrightarrow{W}^{(1)} +$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Matrix times vector:
outer product interpretation

$$\begin{array}{c}
\begin{array}{cccc} \overrightarrow{W}^{(1)} \downarrow & \quad & \quad & \quad \end{array} \\
\begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix}
\end{array}
\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = x_1 \overrightarrow{W}^{(1)} + x_2 \overrightarrow{W}^{(2)}$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Matrix times vector:
outer product interpretation

$$\begin{array}{l} \quad\overrightarrow{W}^{(1)} \\ \quad\;\downarrow \\ \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = x_1\overrightarrow{W}^{(1)} + x_2\overrightarrow{W}^{(2)} + \cdots + x_N\overrightarrow{W}^{(N)} \end{array}$$

- The product is a weighted sum of the columns of W, weighted by the entries of x

Example of the outer product method

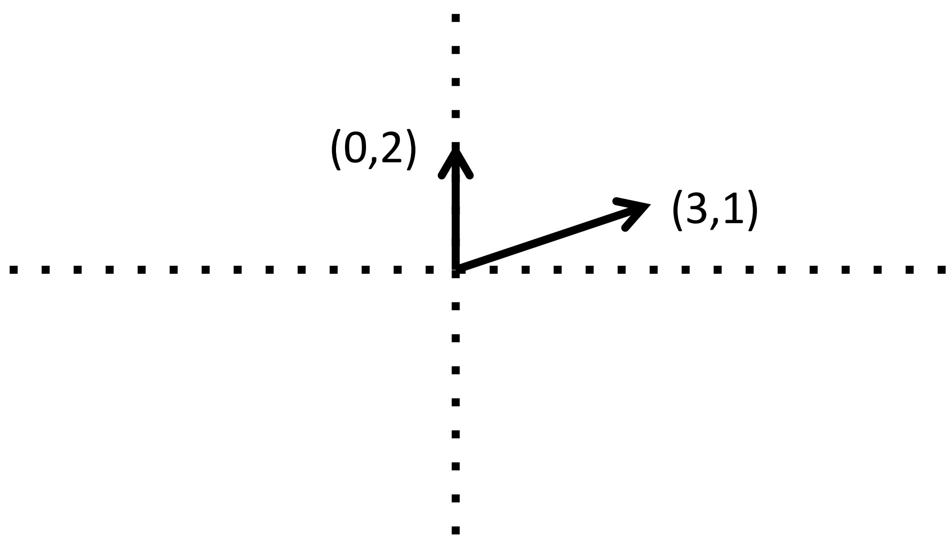
$\overleftrightarrow{M}$

$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} =$$

Example of the outer product method

$$\overleftrightarrow{M}$$

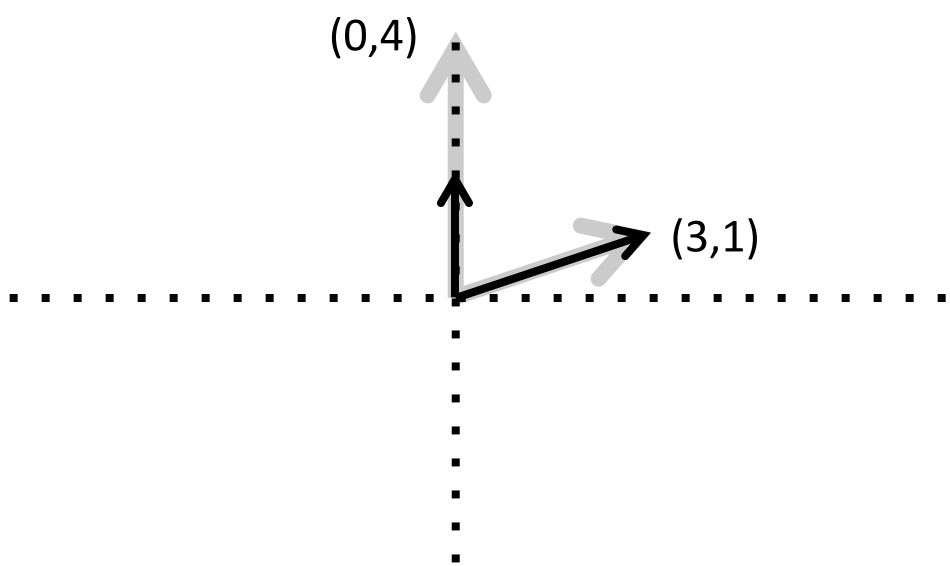
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$



Example of the outer product method

$\overleftrightarrow{M}$

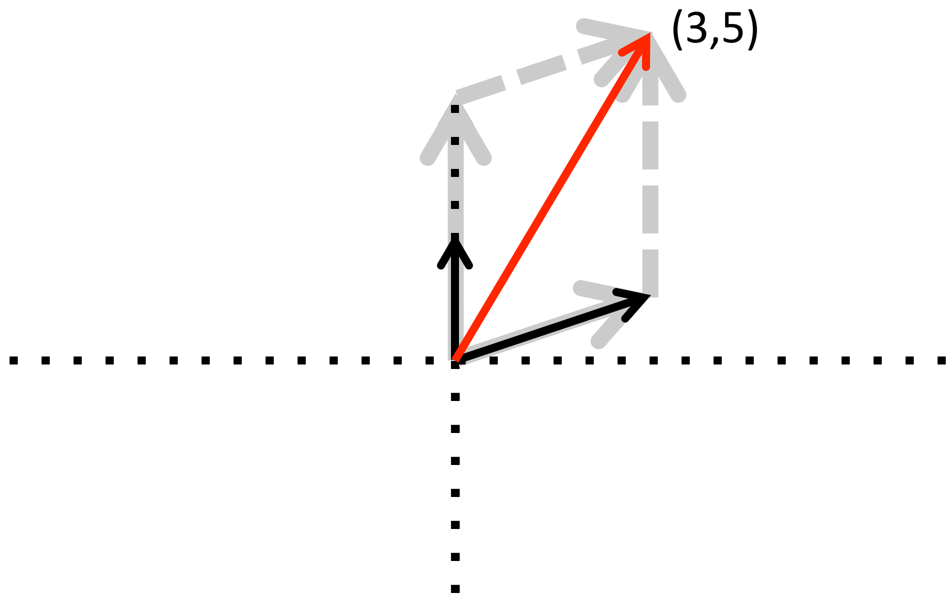
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$



Example of the outer product method

$$\overleftrightarrow{M}$$

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



- Note: different combinations of the columns of **M** can give you any vector in the plane

(we say the columns of **M** "span" the plane)

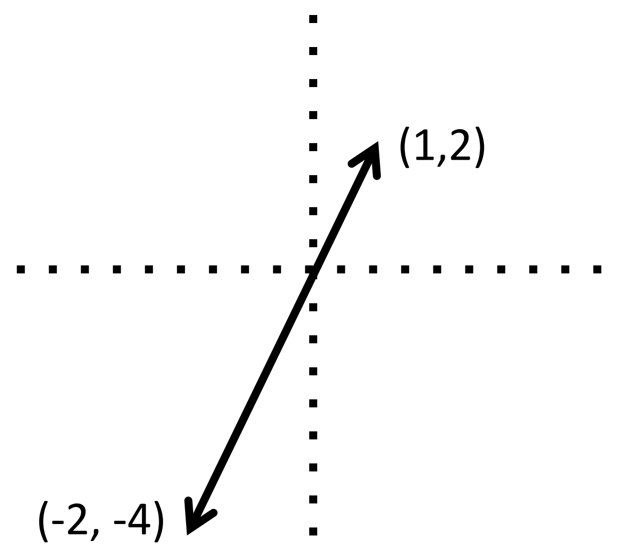
Rank of a Matrix

- Are there special matrices whose columns don’t span the full plane?

Rank of a Matrix

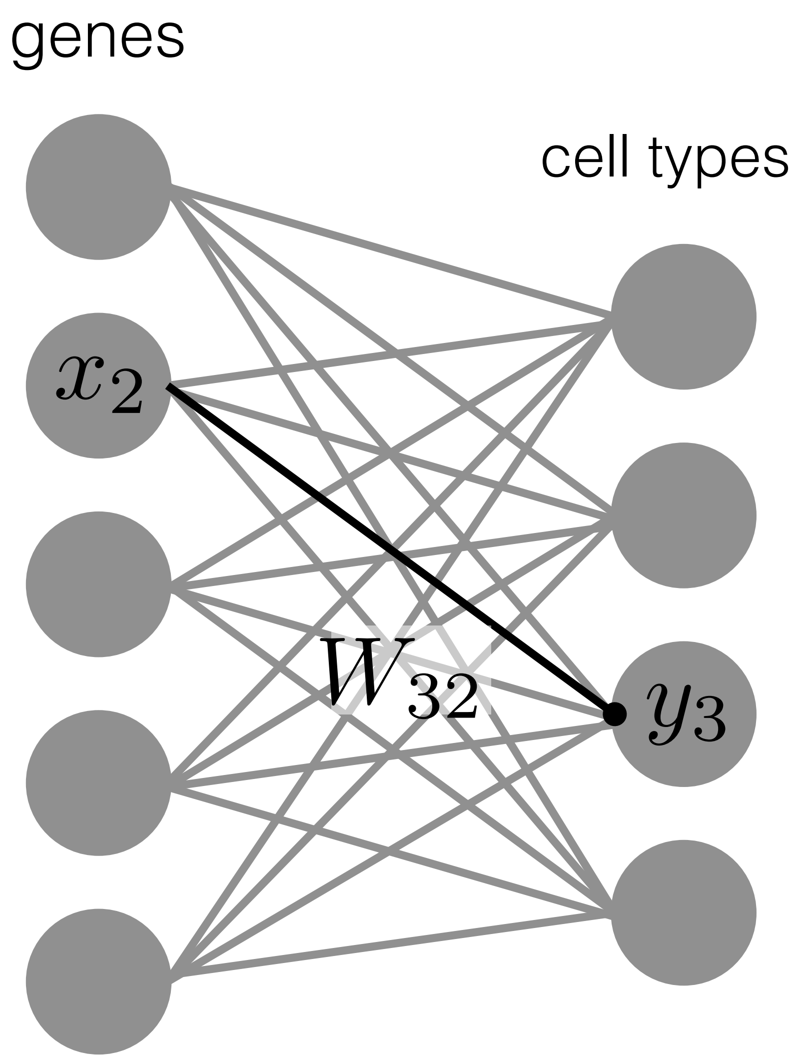
- Are there special matrices whose columns don't span the full plane?

$$\begin{pmatrix} 1 & -2 \\ 2 & -4 \end{pmatrix}$$



- You can only get vectors along the (1,2) direction (i.e. outputs live in 1 dimension, so we call the matrix *rank 1*)

Example: Development of cell types

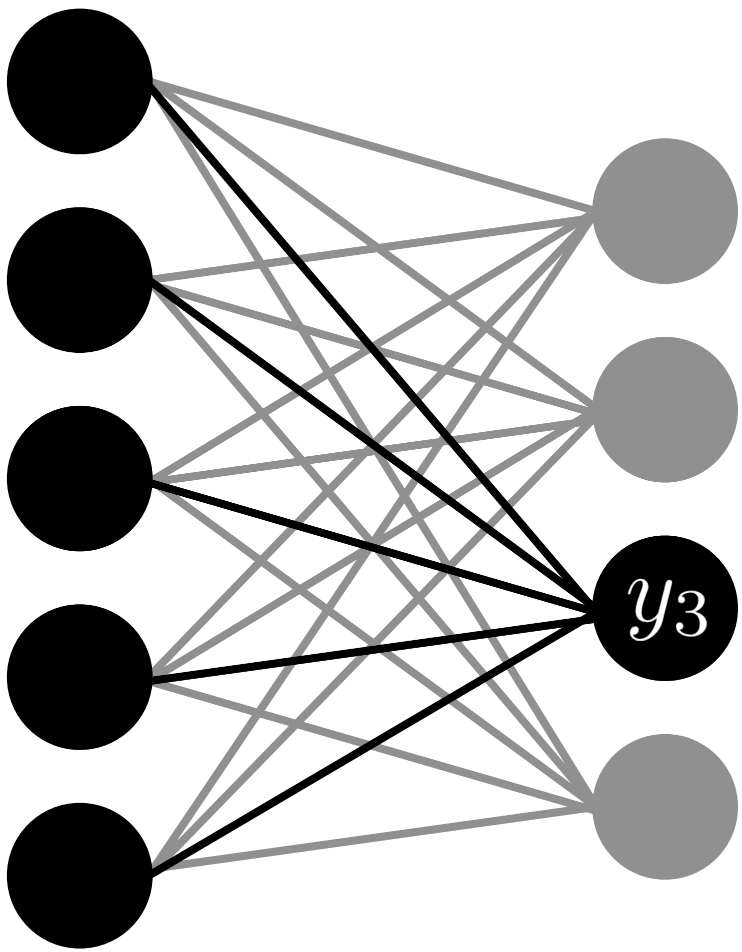


$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- W_{32} is the influence of gene 2 on developing cell type 3

Example: Development of cell types
inner product point of view

- *How many cells of type 3 will be created?*

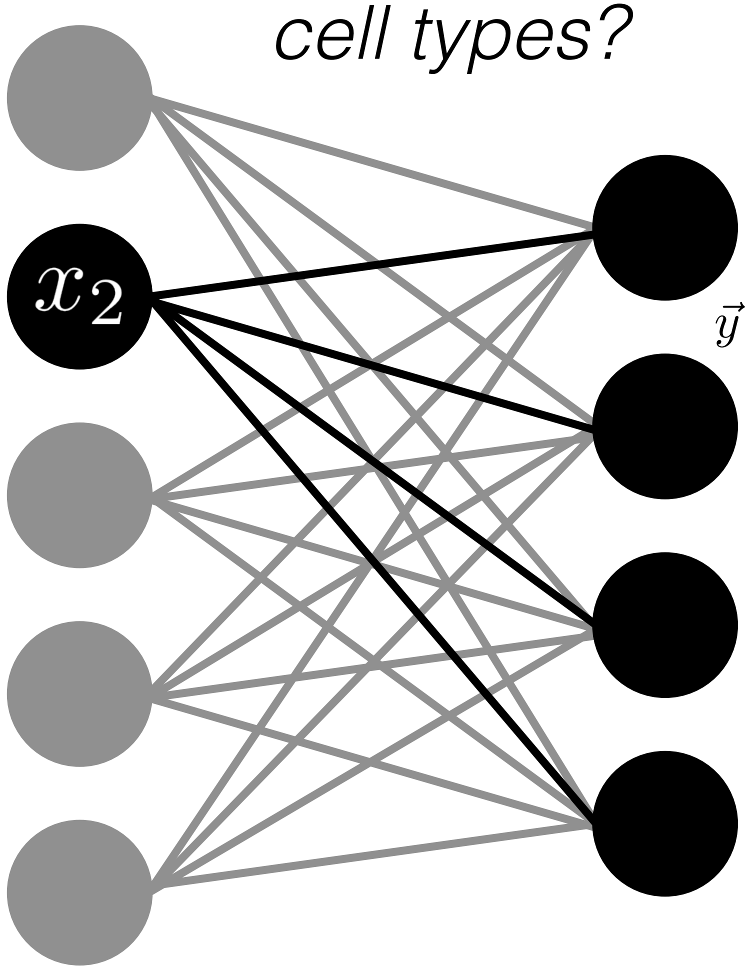


$$y_3 = \sum_{j=1}^{5} W_{3j} x_j$$

- The response is the dot product of the 3rd row of W with the vector x (gene expressions)

Example: Development of cell types: **outer product point of view**

- *How does gene 2 contribute to the distribution of cell types?*



$$\vec{y} = x_1 \overbrace{\overrightarrow{W}^{(1)}}^{\text{1}^{\text{st}}\text{ column of W}} + x_2\overrightarrow{W}^{(2)} + \cdots + \underbrace{x_j\overrightarrow{W}^{(j)}}_{\text{Contribution of } x_j \textit{ to cell types}} + \cdots + x_N\overrightarrow{W}^{(N)}$$

Product of 2 Matrices

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

N X P   P X M   N X M

- MATLAB: 'inner matrix dimensions must agree'
- **Note:** Matrix multiplication doesn't (generally) commute, AB ≠ BA

Matrix times Matrix: **by inner products**

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

- C_{ij} is the inner product of the i^{th} row with the j^{th} column

Matrix times Matrix:
by inner products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

- C_{ij} is the inner product of the i^{th} row with the j^{th} column

Matrix times Matrix:
by inner products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

- C_{ij} is the inner product of the i^{th} row with the j^{th} column

Matrix times Matrix: **by inner products**

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{iP} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1j} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2j} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{Pj} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & C_{ij} & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$C_{ij} = \sum_{k=1}^{P} A_{ik} B_{kj}$$

- C_{ij} is the inner product of the i^{th} row of **A** with the j^{th} column of **B**

Matrix times Matrix:
by outer products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} =$$

Matrix times Matrix:
by outer products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} = \begin{pmatrix} \\ A^{\mathrm{c}1} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}1} \quad \end{pmatrix} +$$

Matrix times Matrix: **by outer products**

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} = \begin{pmatrix} \\ A^{\mathrm{c}1} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}1} \quad \end{pmatrix} + \begin{pmatrix} \\ A^{\mathrm{c}2} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}2} \quad \end{pmatrix} +$$

Matrix times Matrix:
by outer products

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1P} \\ A_{21} & A_{22} & \cdots & A_{2P} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NP} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{P1} & B_{P2} & \cdots & B_{PM} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NM} \end{pmatrix}$$

$$\overleftrightarrow{C} = \begin{pmatrix} \\ A^{\mathrm{c}1} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}1} \quad \end{pmatrix} + \begin{pmatrix} \\ A^{\mathrm{c}2} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{r}2} \quad \end{pmatrix} + \cdots + \begin{pmatrix} \\ A^{\mathrm{cP}} \\ \\ \end{pmatrix} \begin{pmatrix} \quad B^{\mathrm{rP}} \quad \end{pmatrix}$$

- C is a sum of outer products of the columns of A with the rows of B

BREAK

Part 2: Matrix Properties

- (A few) special matrices
- The determinant and eigenvalues/vectors
- Principal component analysis (PCA)

Special matrices: **diagonal matrix**

$$\overleftrightarrow{D} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix}$$

$$\overleftrightarrow{D}\,\overrightarrow{x} = \begin{pmatrix} d_1 x_1 \\ d_2 x_2 \\ \vdots \\ d_n x_n \end{pmatrix}$$

- This acts like scalar multiplication

Special matrices: **identity matrix**

$$\overleftrightarrow{1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

for all $\overleftrightarrow{A}, \quad \overleftrightarrow{1}\,\overleftrightarrow{A} = \overleftrightarrow{A}\,\overleftrightarrow{1} = \overleftrightarrow{A}$

Special matrices: **inverse matrix**

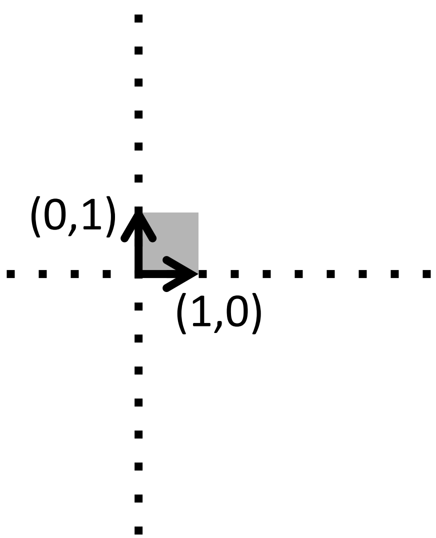
$$\overleftrightarrow{A}\,\overleftrightarrow{A}^{-1} = \overleftrightarrow{A}^{-1}\overleftrightarrow{A} = \overleftrightarrow{1}$$

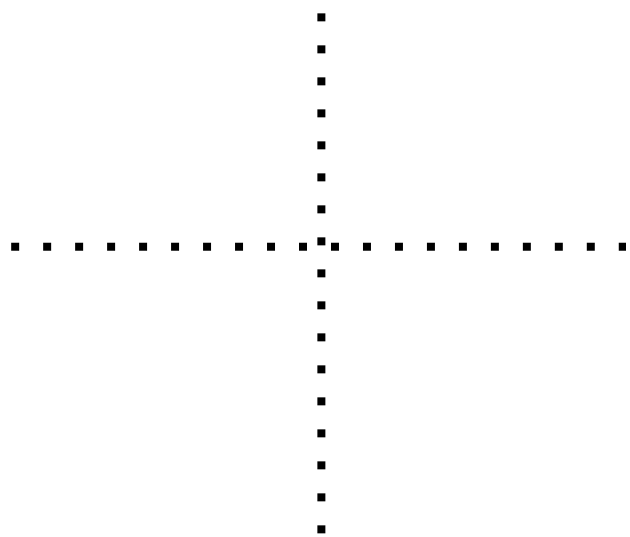
- Does the inverse always exist?

Part 2: Matrix Properties

- (A few) special matrices
- The determinant and eigenvalues/vectors
- Principal component analysis (PCA)

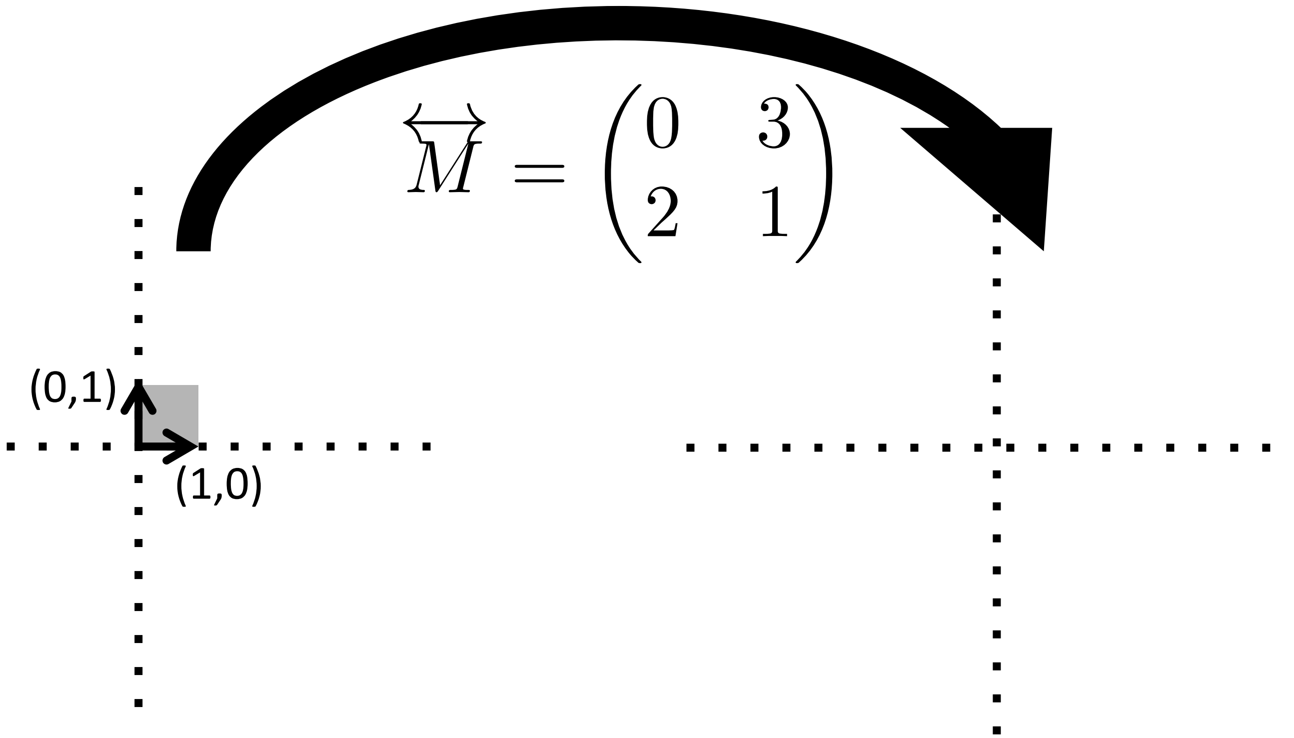
How does a matrix transform a square?





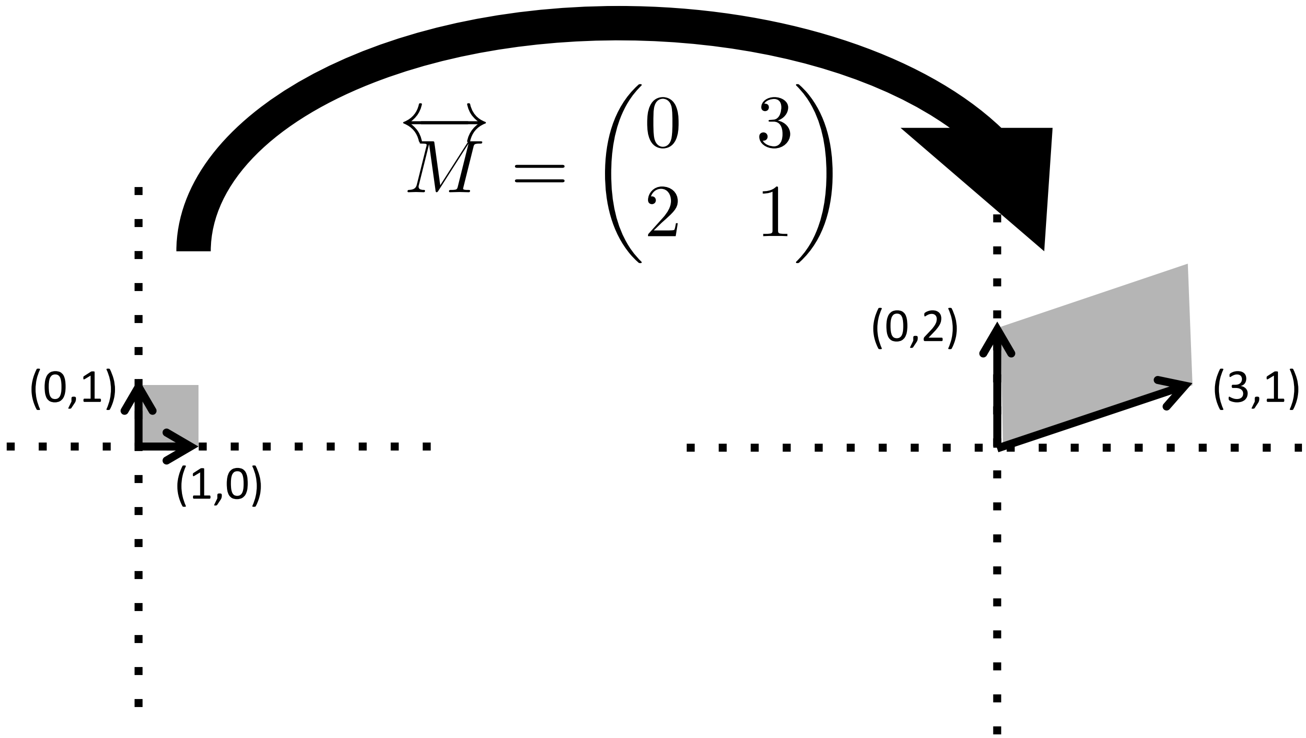
How does a matrix transform a square?

$$\overleftrightarrow{M} = \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}$$

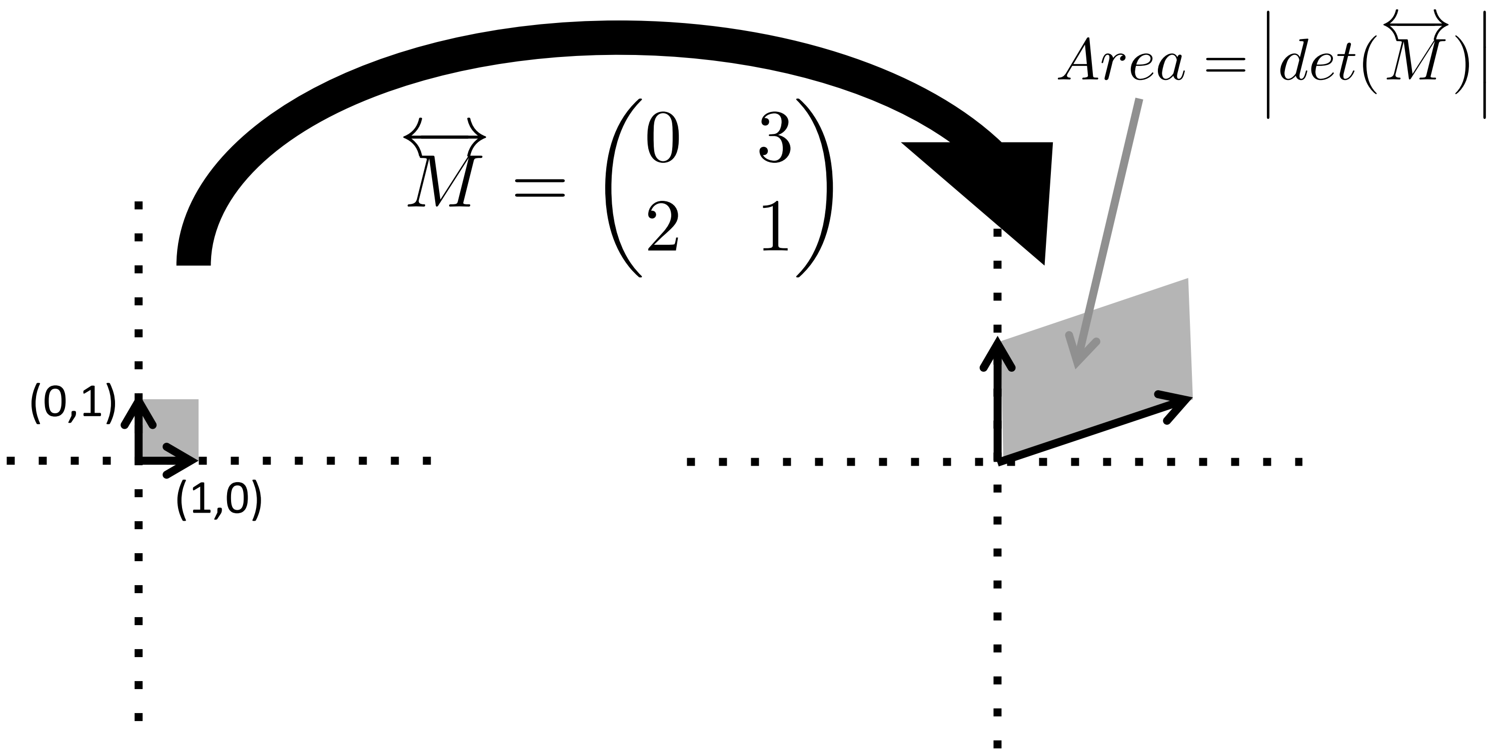


How does a matrix transform a square?

$$\overleftrightarrow{M} = \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}$$



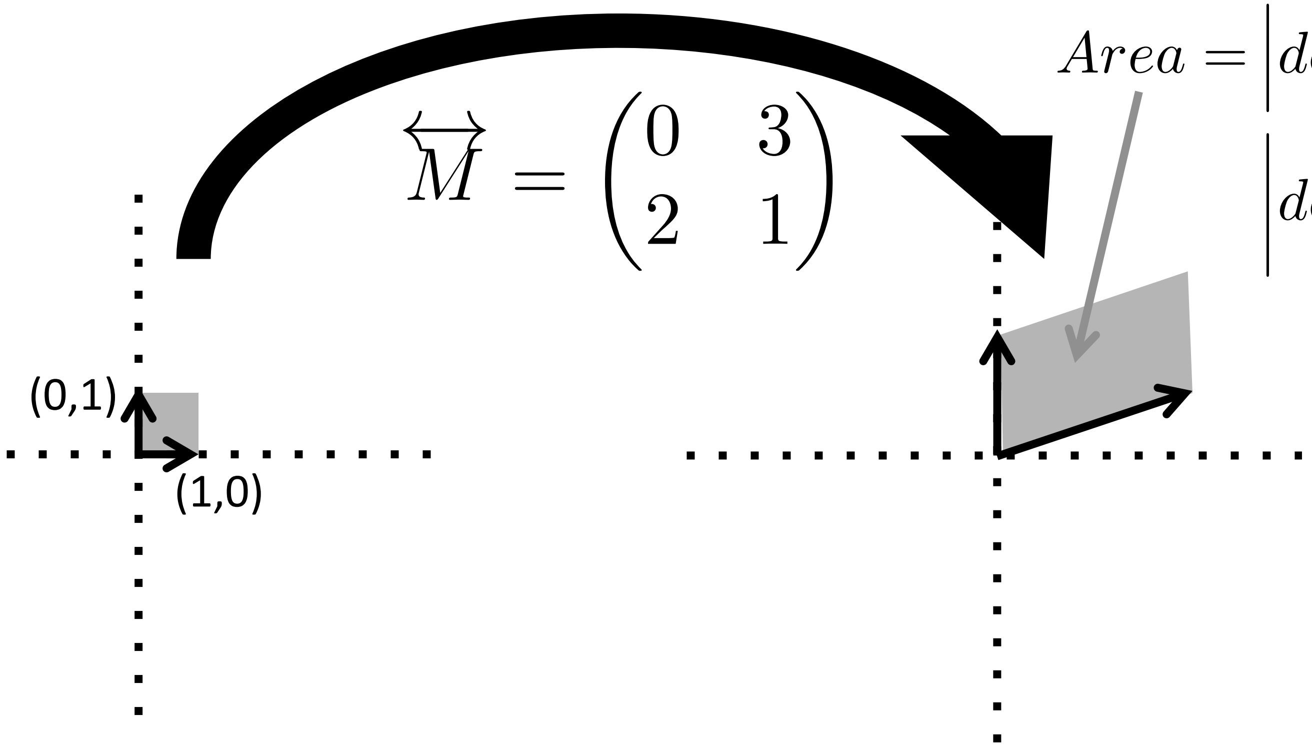
Geometric definition of the determinant: How does a matrix transform a square?



$$Mx?$$

MATLAB: `det(M)`

Geometric definition of the determinant: How does a matrix transform a square?



$$\overleftrightarrow{M} = \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}$$

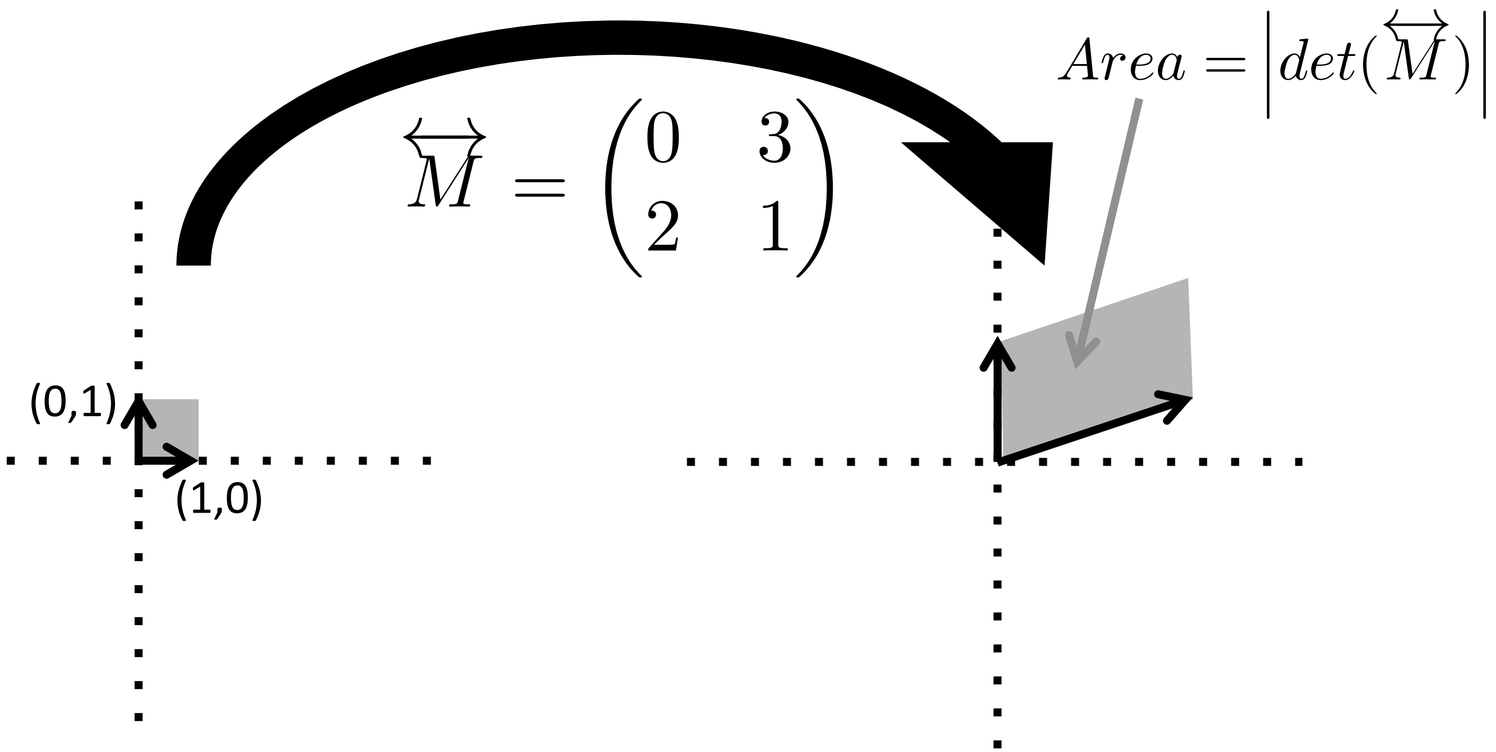
$$Area = \left| det(\overleftrightarrow{M}) \right|$$

$$\left| det \begin{pmatrix} a & b \\ c & d \end{pmatrix} \right| = |ad - bc|$$

$$Mx?$$

MATLAB: `det(M)`

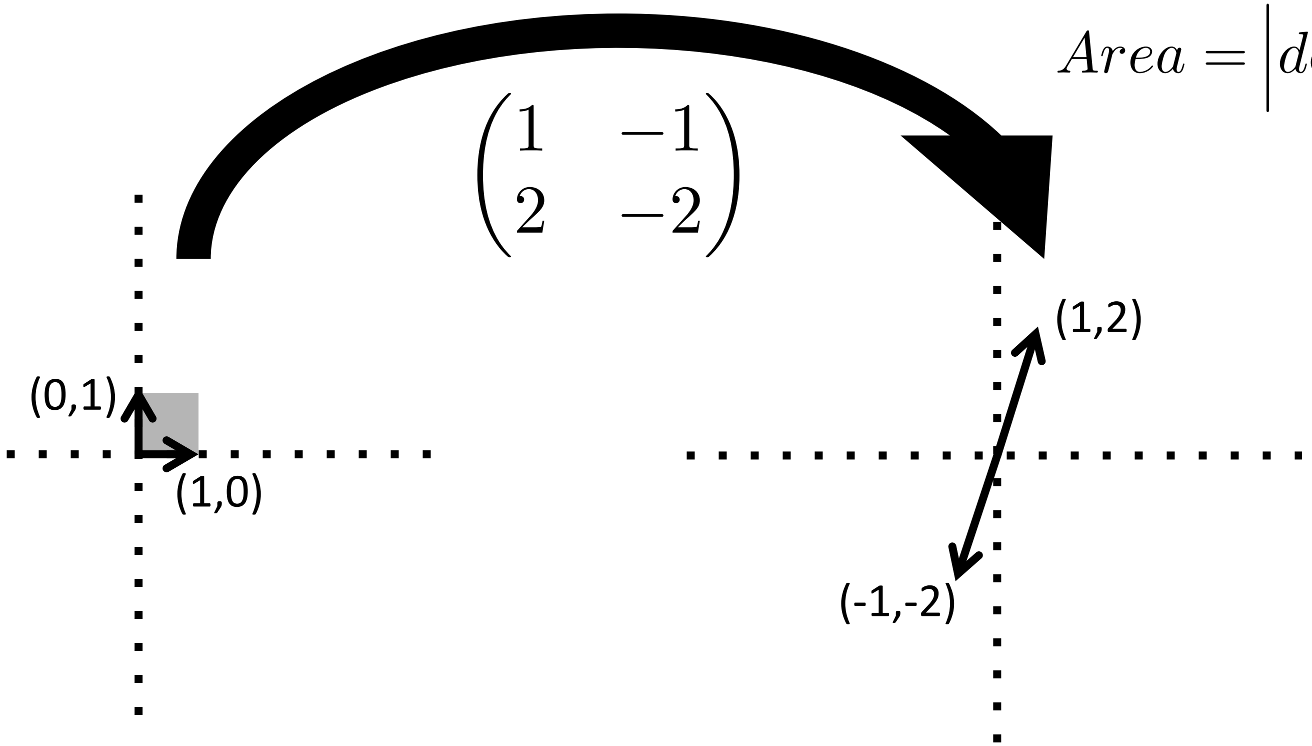
Geometric definition of the determinant: How does a matrix transform a square?



what about $\begin{pmatrix} 1 & -1 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$?

Geometric definition of the determinant: How does a matrix transform a square?

$$Area = \left| det(\overleftrightarrow{M}) \right|$$



no area!

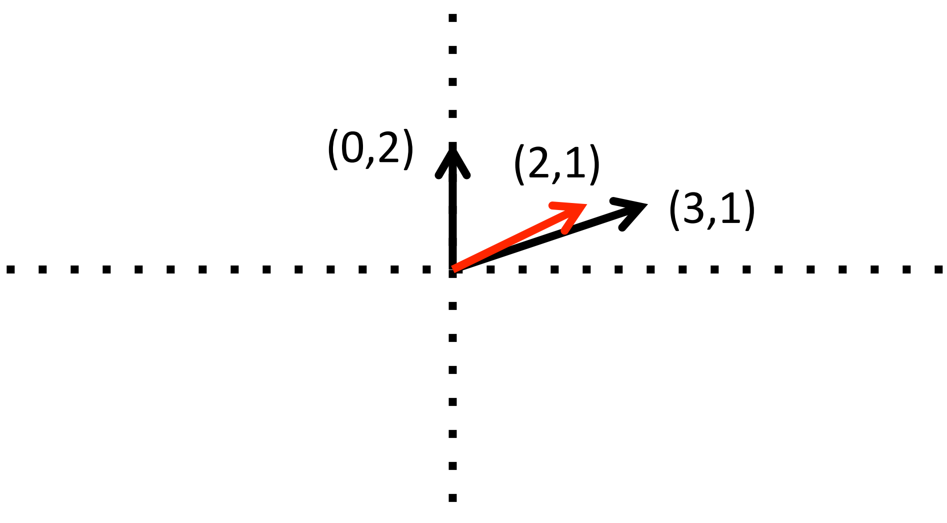
this means the determinant is zero

what about $\begin{pmatrix} 1 & -1 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$?

What do matrices do to vectors?

$$\overleftrightarrow{M}$$

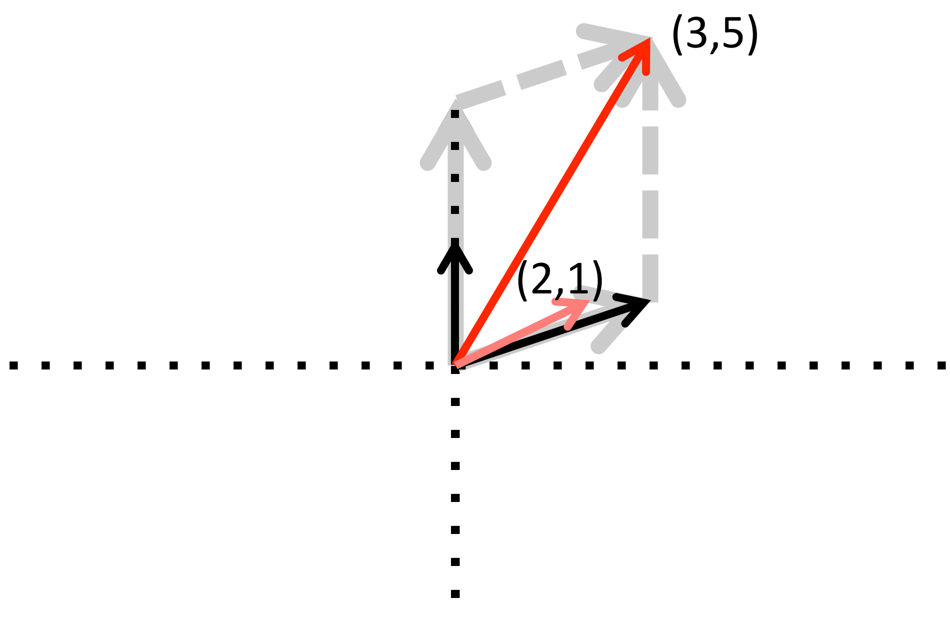
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} =$$



Recall

$$\overleftrightarrow{M}$$

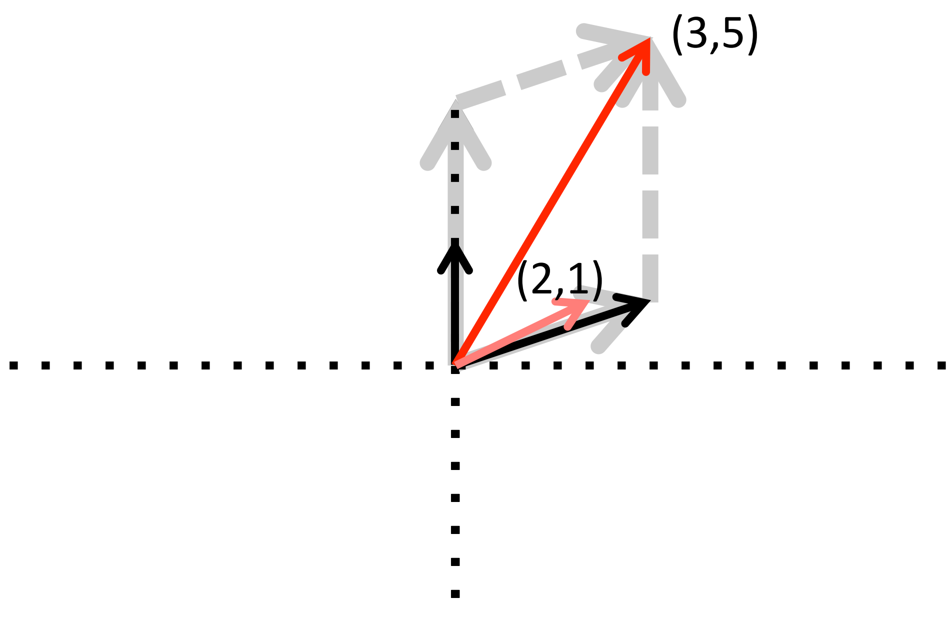
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$



What do matrices do to vectors?

$$\overleftrightarrow{M}$$

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



- The new vector is:
  1) rotated
  2) scaled

Are there any special vectors that **only get scaled**?

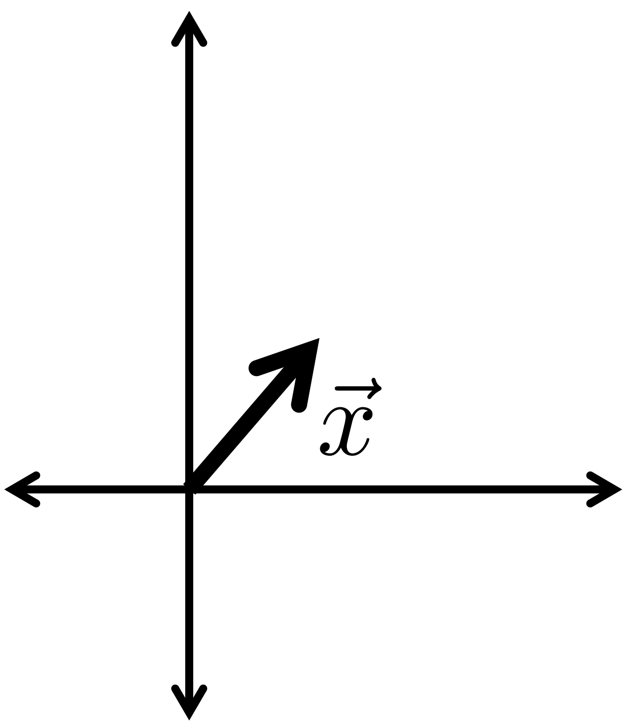
$$\overleftrightarrow{M}$$

$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}$$

Are there any special vectors
that **only get scaled**?

$\overleftrightarrow{M}$

$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

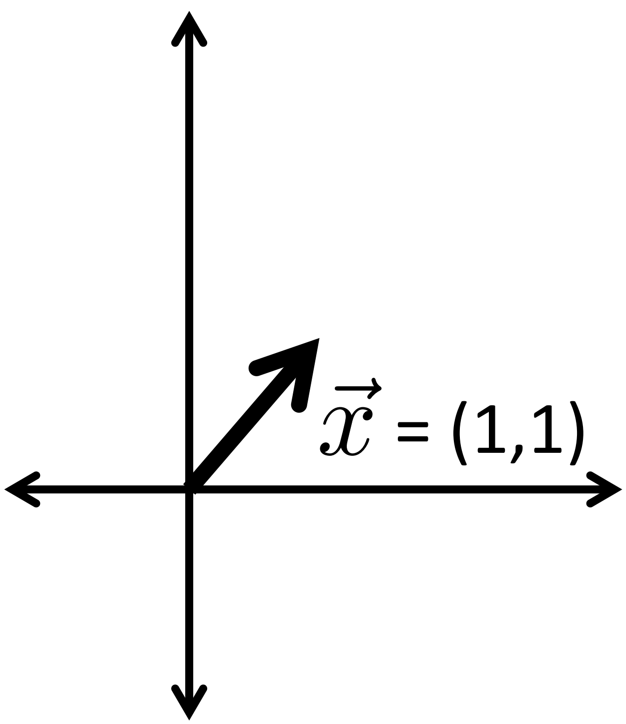


Try (1,1)

Are there any special vectors that **only get scaled**?

$\overleftrightarrow{M}$

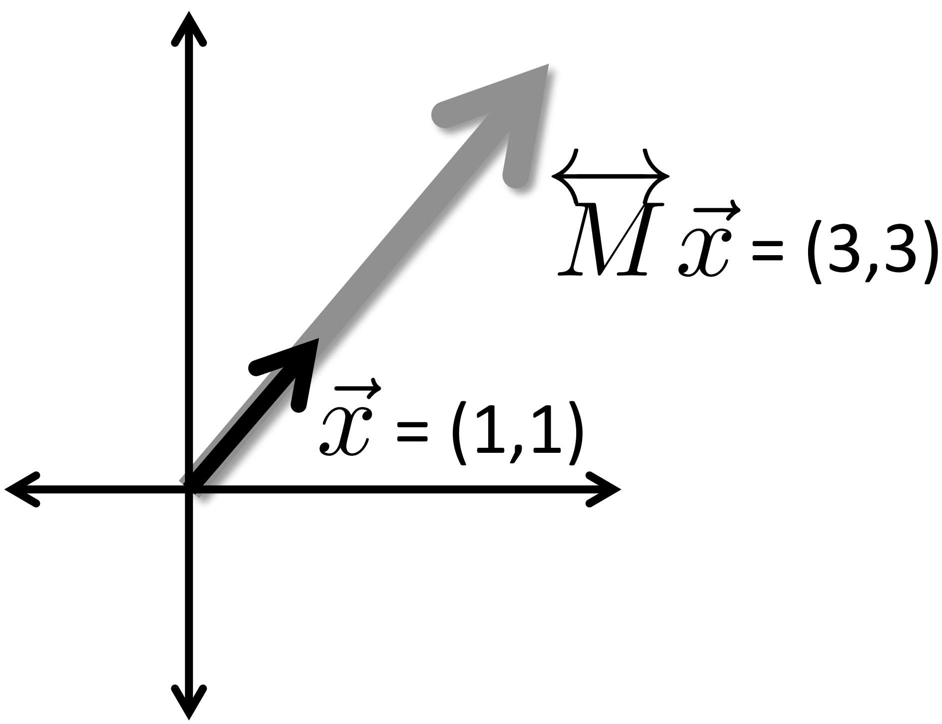
$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$



Are there any special vectors that **only get scaled**?

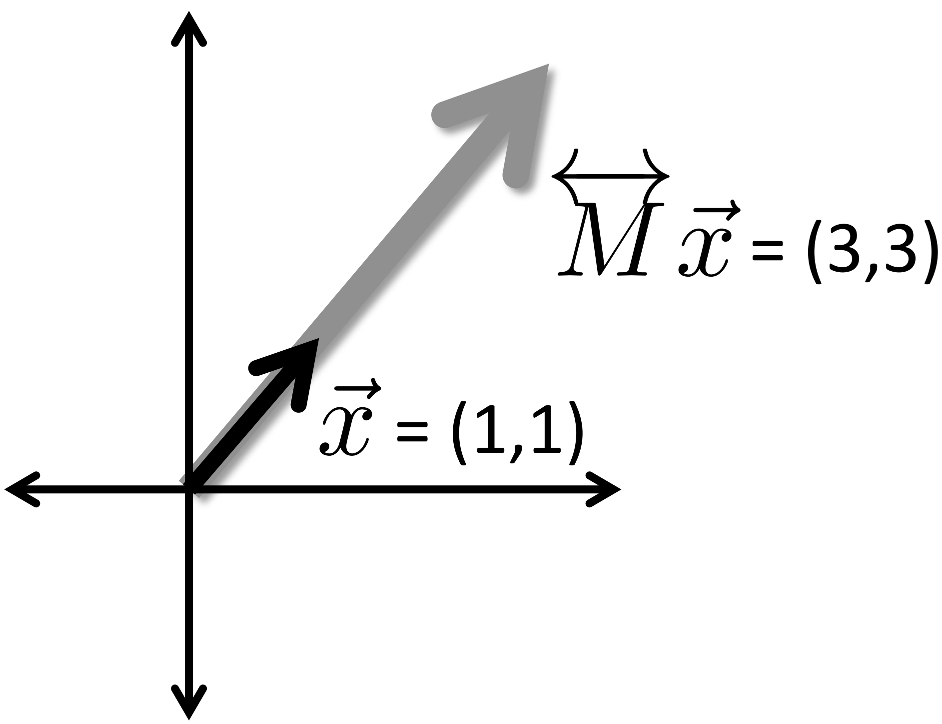
$\overleftrightarrow{M}$

$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



Are there any special vectors that **only get scaled**?

$$\underbrace{\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}}_{\overleftrightarrow{M}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

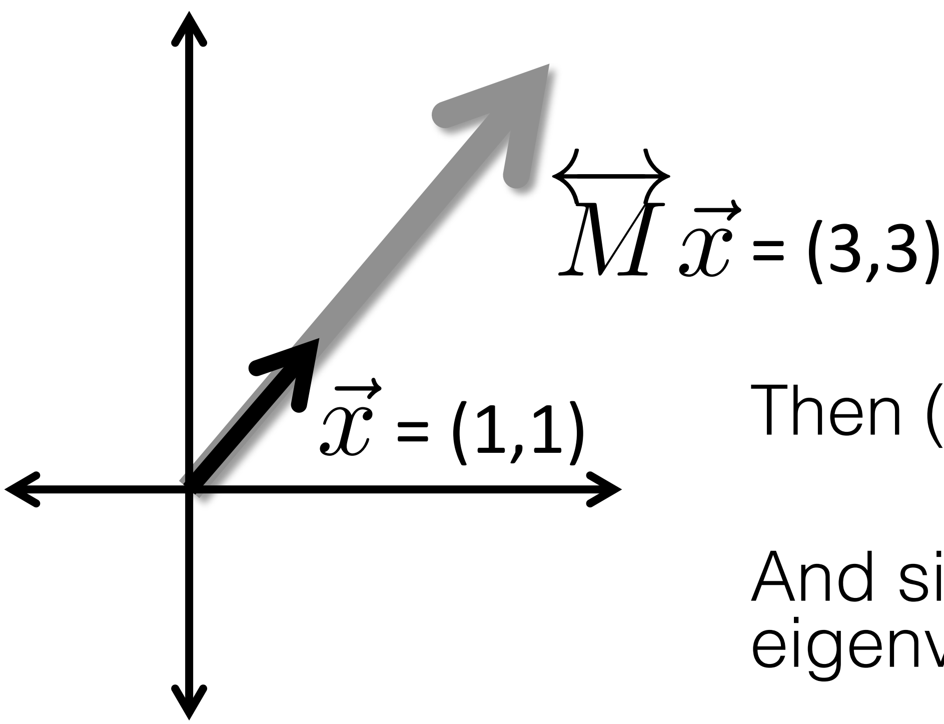


- For this special vector, multiplying by M is like multiplying by a scalar.

Are there any special vectors that **only get scaled**?

$\overleftrightarrow{M}$

$$\begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



- For this special vector, multiplying by M is like multiplying by a scalar.

Then (1,1) is an eigenvector!

And since it was scaled by 3, its eigenvalue is 3.

Are there any other eigenvectors?

Are there any other eigenvectors?

- Yes! The easiest way to find is with MATLAB's **`eig`** command.

$$\vec{e}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \ \vec{e}^{(2)} = \begin{pmatrix} -1.5 \\ 1 \end{pmatrix}$$

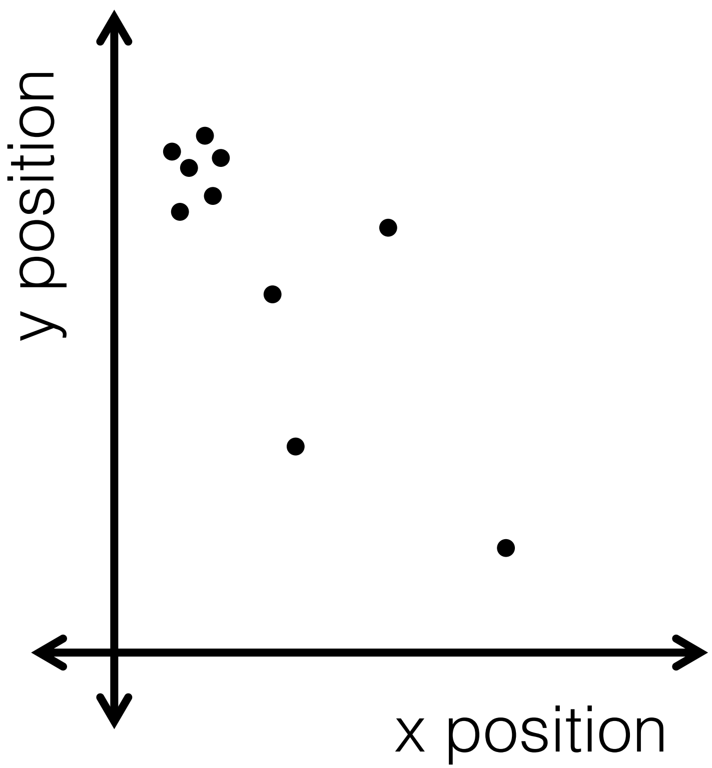
- Exercise: verify that (-1.5, 1) is also an eigenvector of M.

(MATLAB returns unit norm eigenvectors, which means that $\sqrt{e_1^2 + e_2^2 = 1}$)

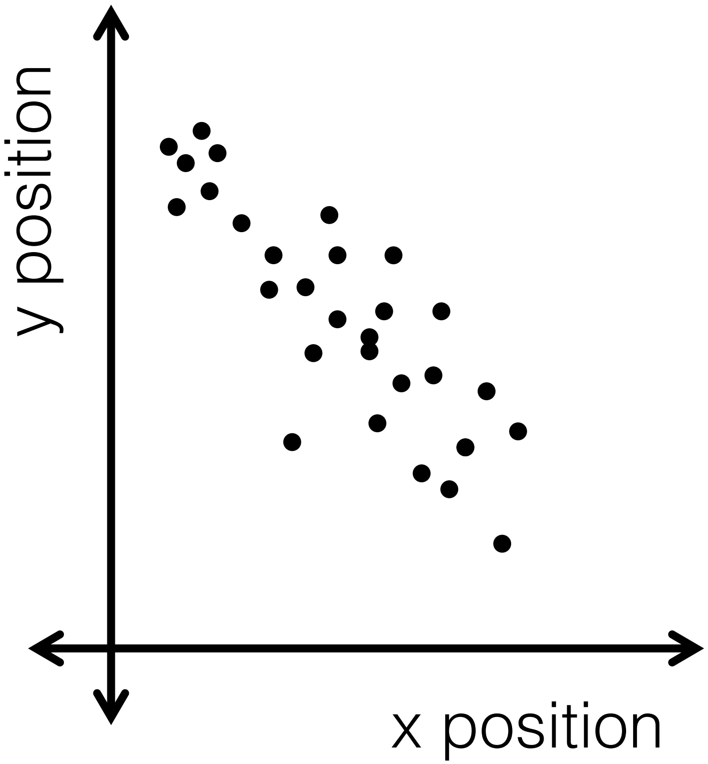
Part 2: Matrix Properties

- (A few) special matrices
- The determinant and eigenvalues/vectors
- Principal component analysis (PCA)

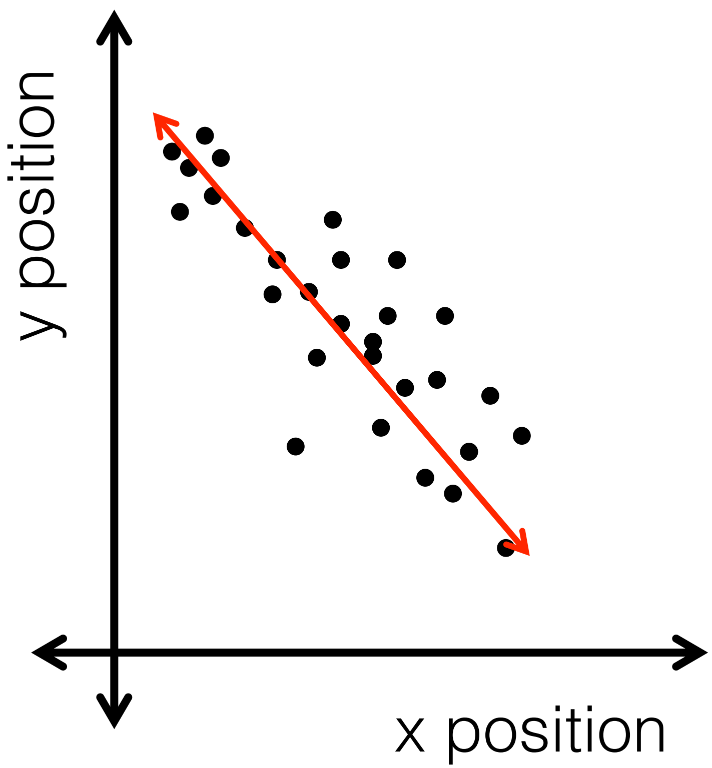
Why PCA?



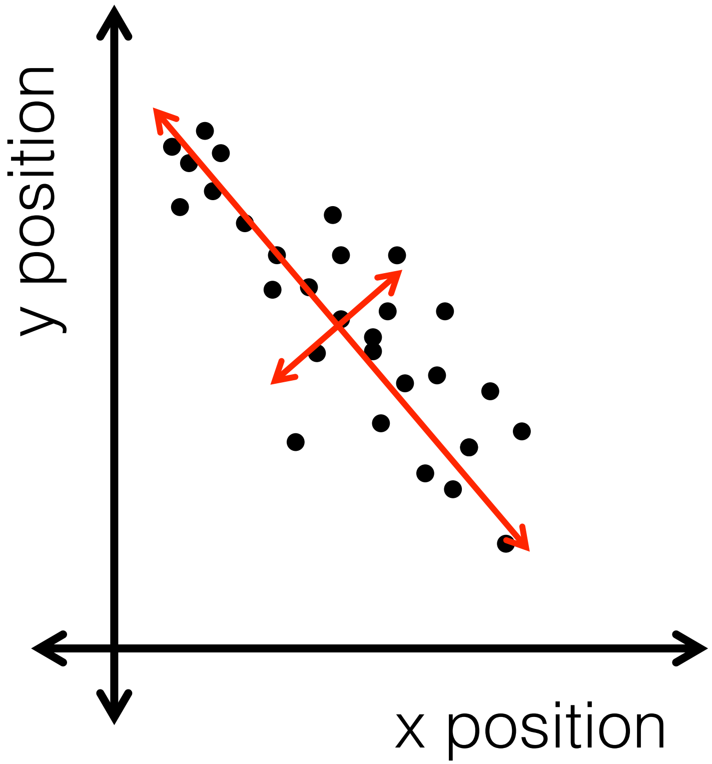
Why PCA?



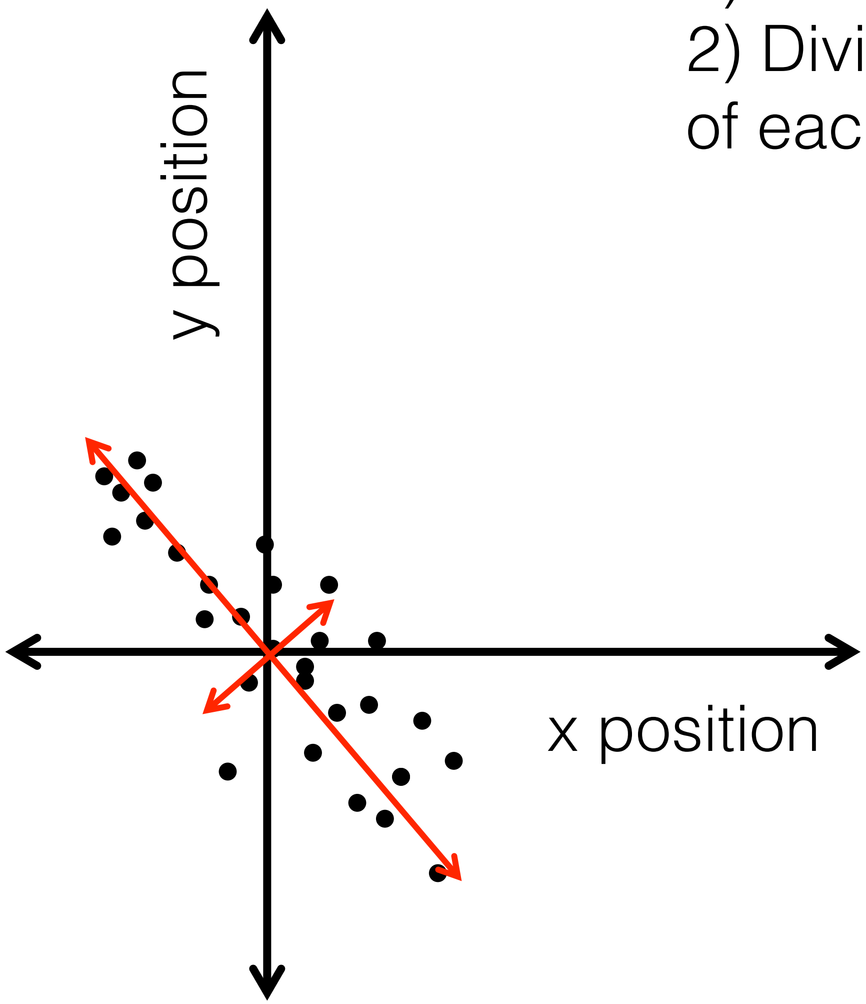
Why PCA?



Why PCA?



Before you can do PCA

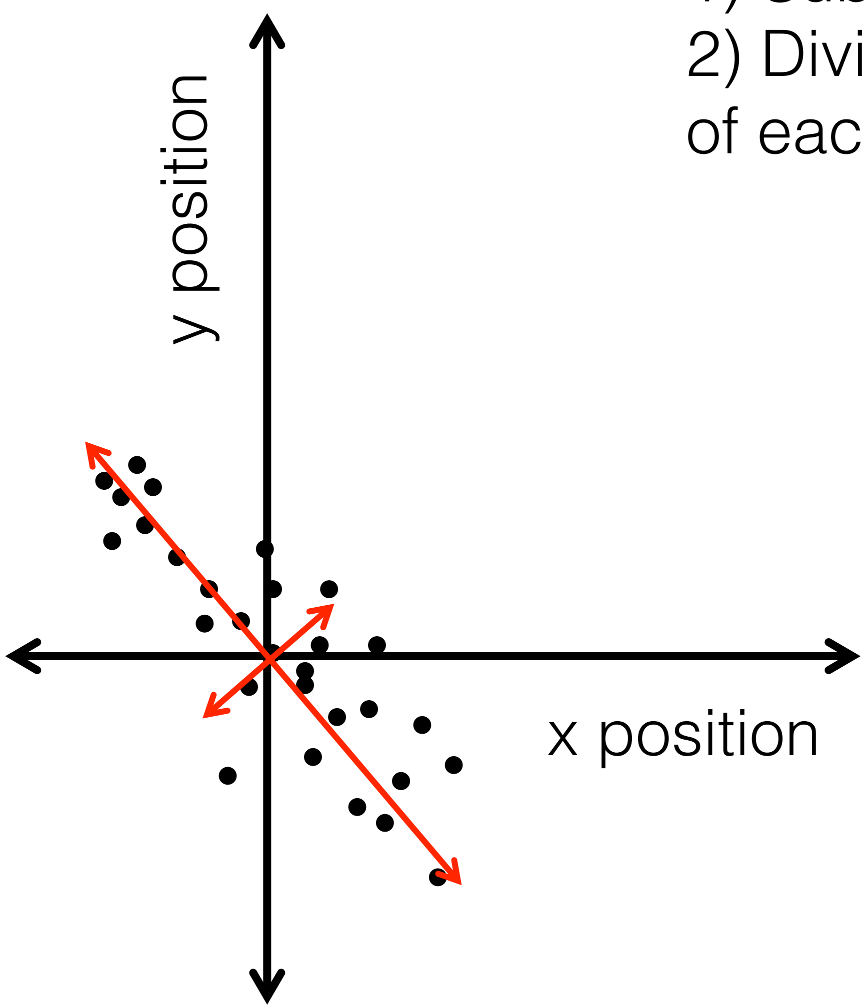


1) Subtract mean from data
2) Divide by the standard deviation of each variable

MATLAB: `zscore(X)`

Before you can do PCA

1) Subtract mean from data
2) Divide by the standard deviation of each variable

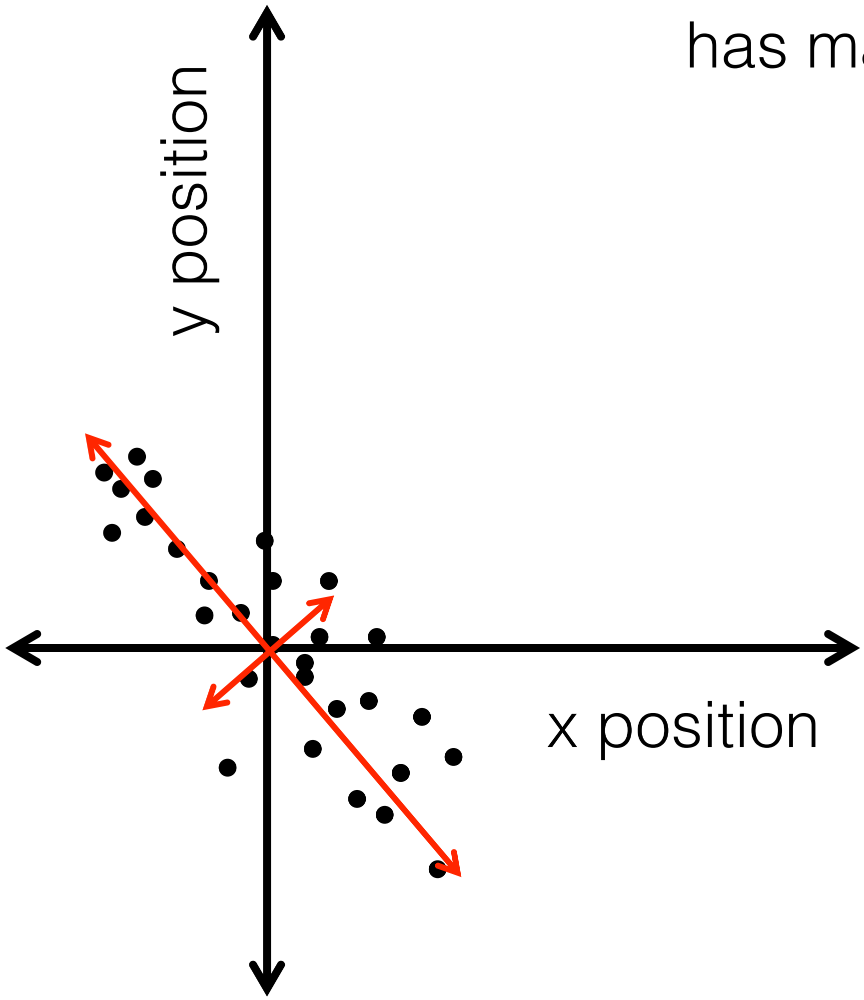


MATLAB: `zscore(X)`

Why?

Exercise: What would happen if x position was measured in millimeters and y position was measured in meters?

Ok, so now what is PCA?

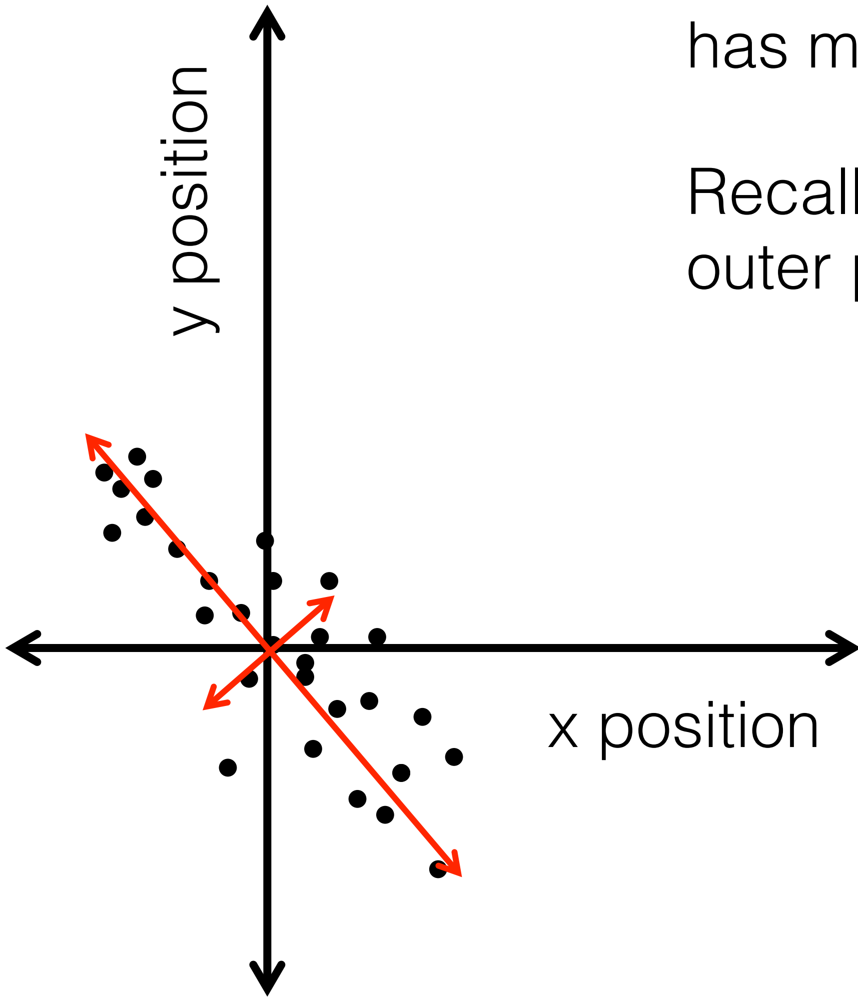


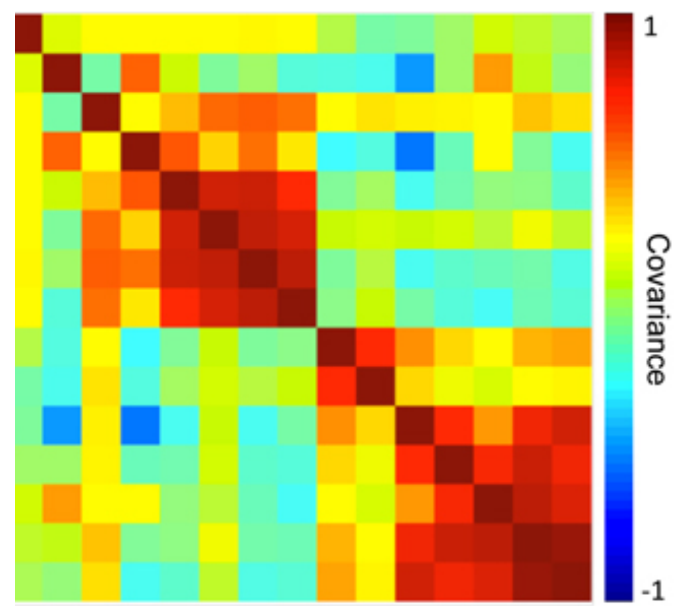
Find a combination of your variables that has maximum variance.

Ok, so now what is PCA?

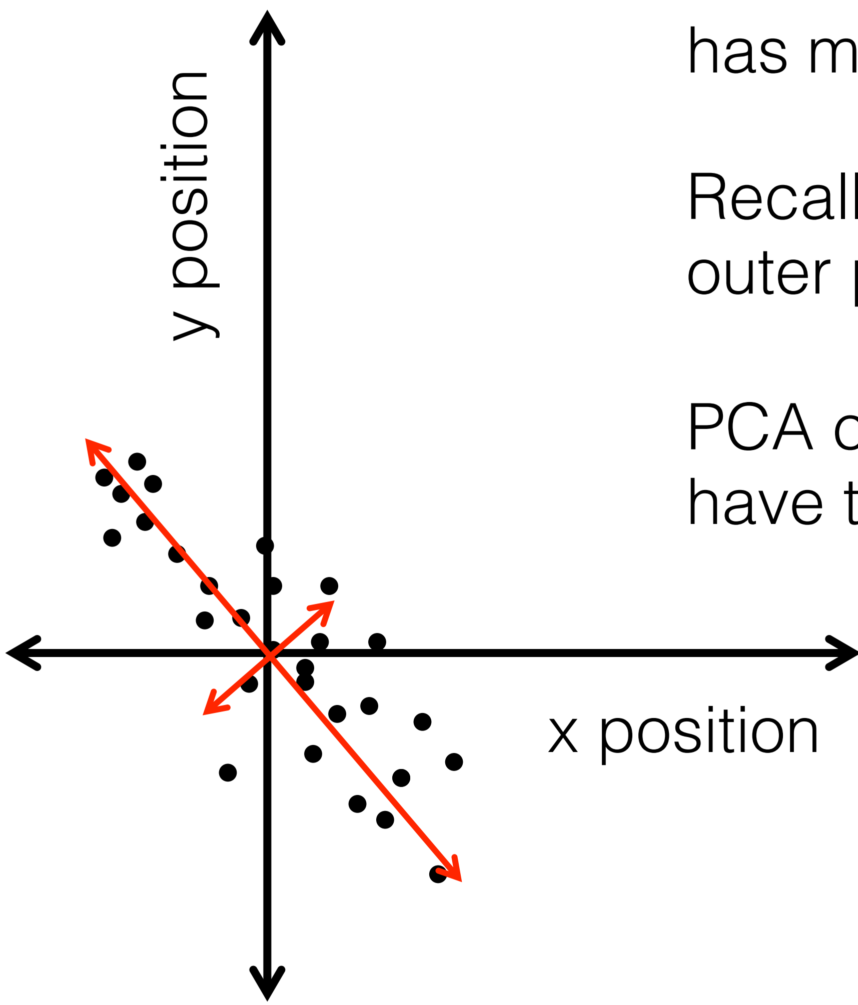
Find a combination of your variables that has maximum variance.

Recall that when x has zero mean, the outer product xx^T is the covariance matrix.





Ok, so now what is PCA?

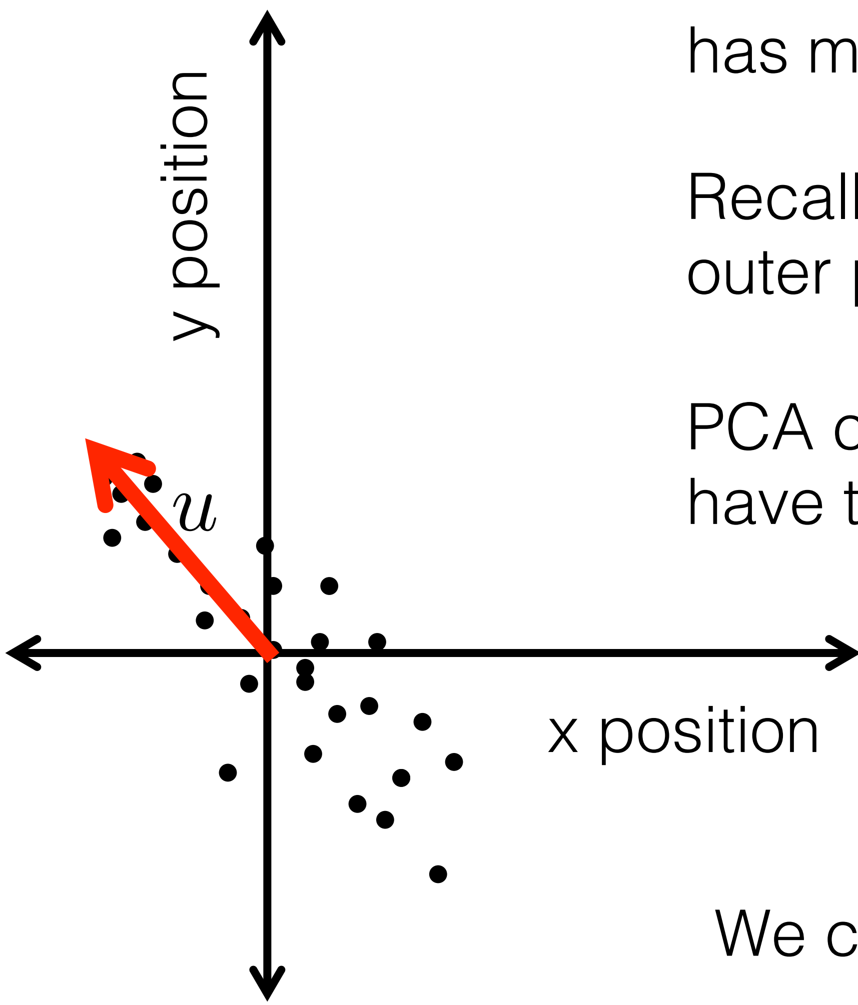


Find a combination of your variables that has maximum variance

Recall that when x has zero mean, the outer product xx^T is the covariance matrix

PCA chooses a vector u such that we have the largest possible $u^T x x^T u$

Ok, so now what is PCA?



Find a combination of your variables that has maximum variance

Recall that when x has zero mean, the outer product xx^T is the covariance matrix

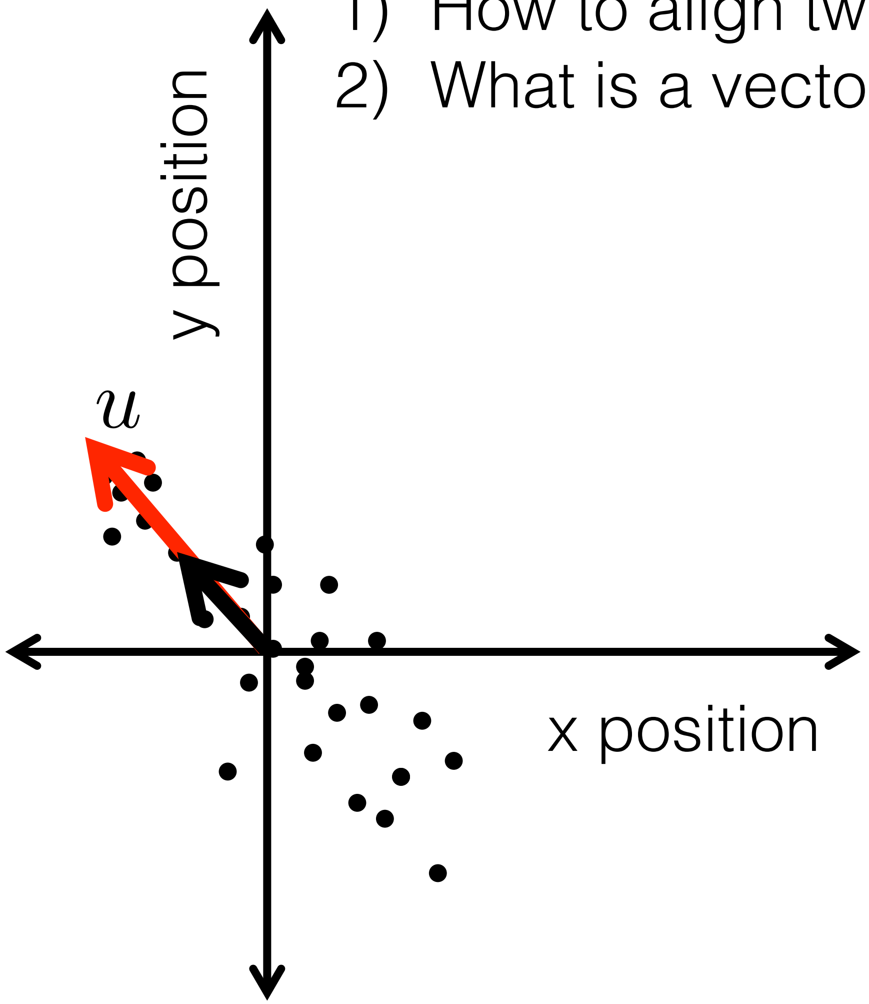
PCA chooses a vector u such that we have the largest possible $u^T x x^T u$

We call u the first principal component

How do we choose u?

Recall:

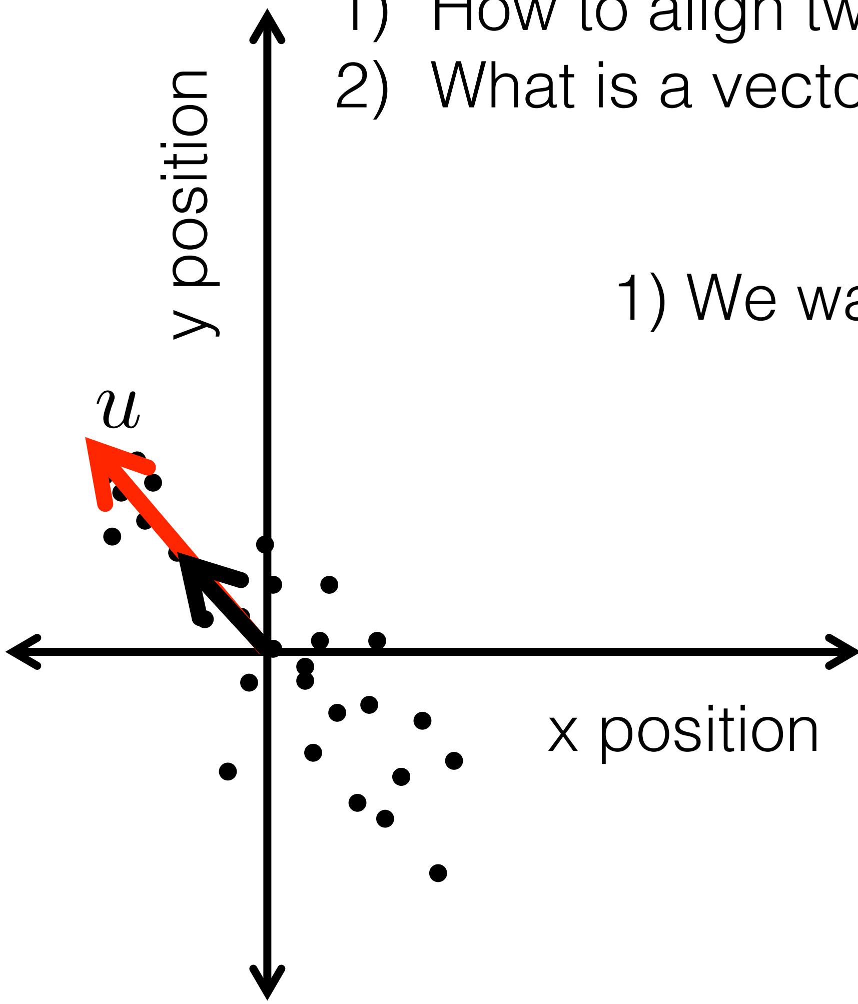
1) How to align two vectors to maximize dot product?
2) What is a vector that only gets scaled by a matrix?



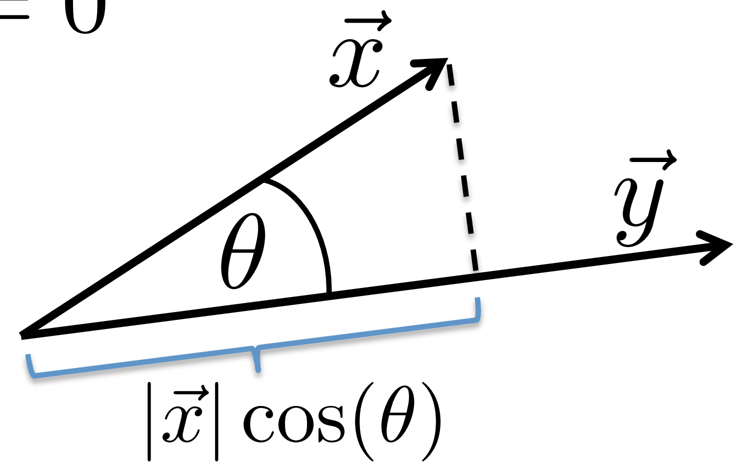
How do we choose u?

Recall:

1) How to align two vectors to maximize dot product?
2) What is a vector that only gets scaled by a matrix?



1) We want $\theta = 0$

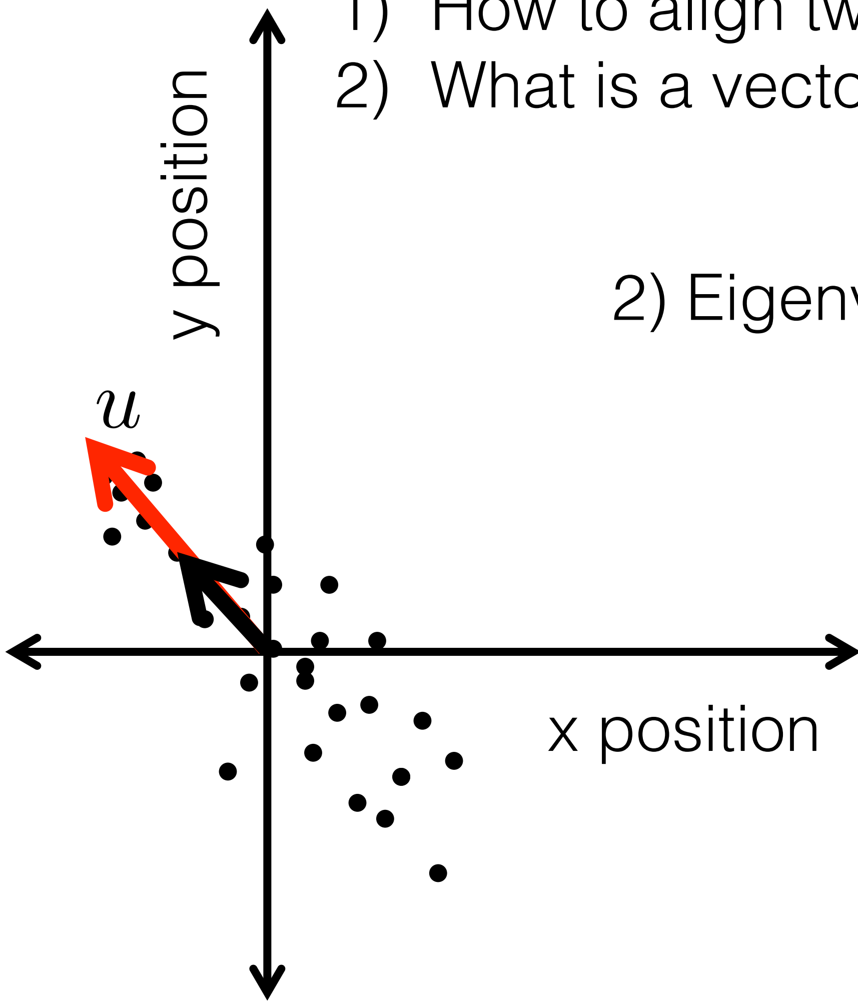


$$\vec{x} \cdot \vec{y} = |\vec{x}||\vec{y}|\cos(\theta)$$

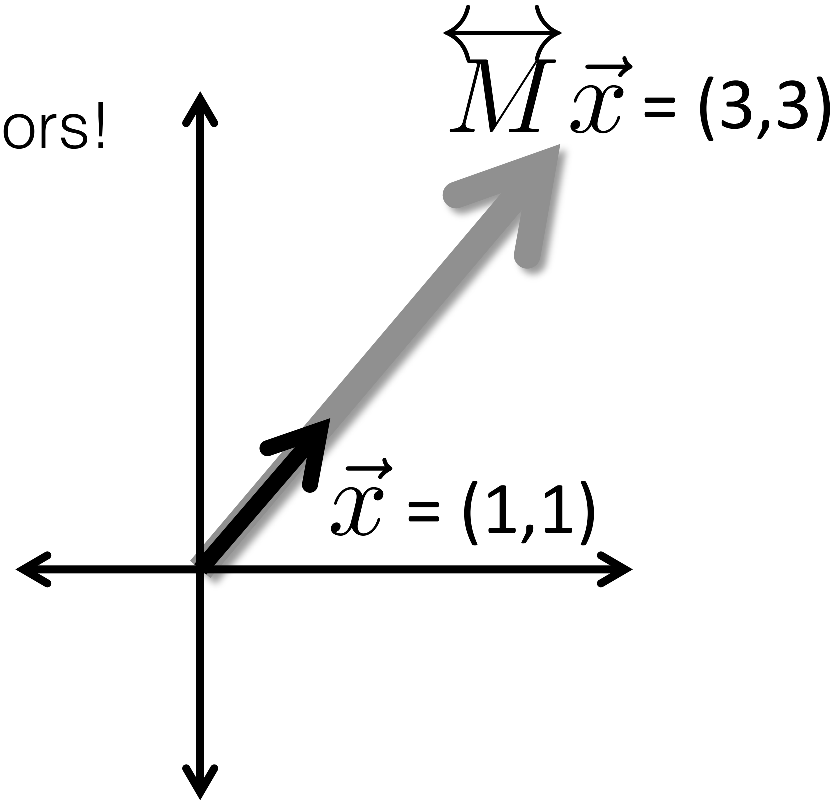
How do we choose u?

Recall:

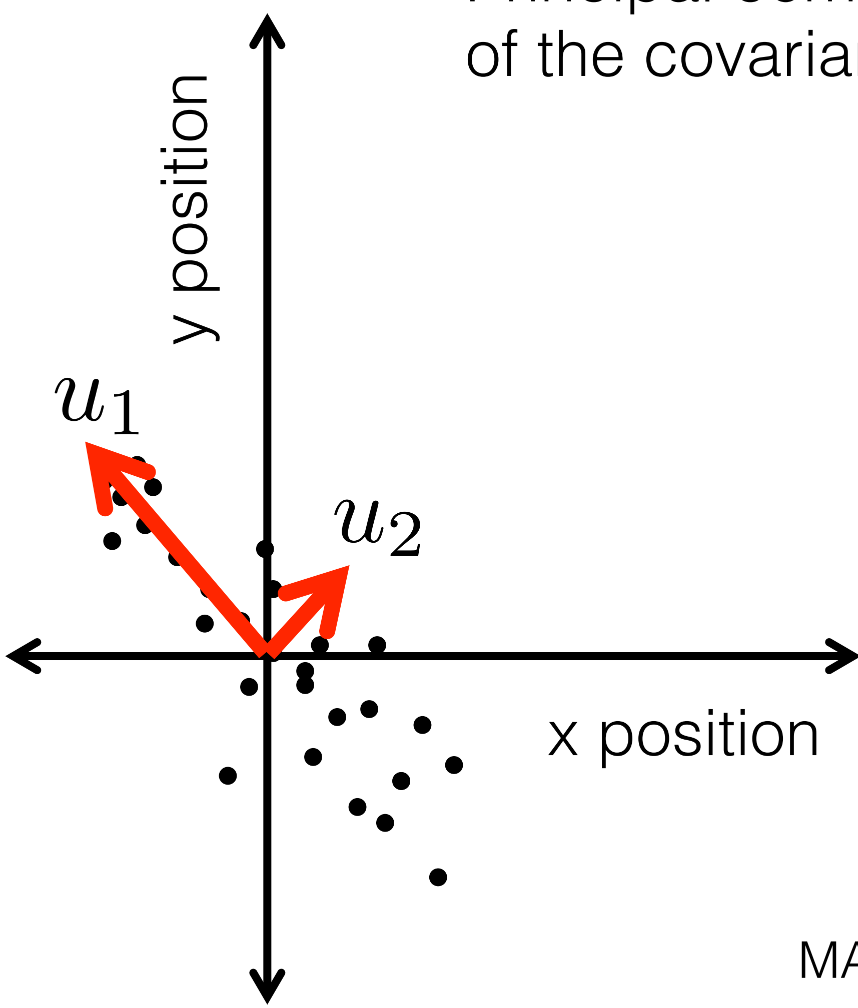
1) How to align two vectors to maximize dot product?
2) What is a vector that only gets scaled by a matrix?



2) Eigenvectors!



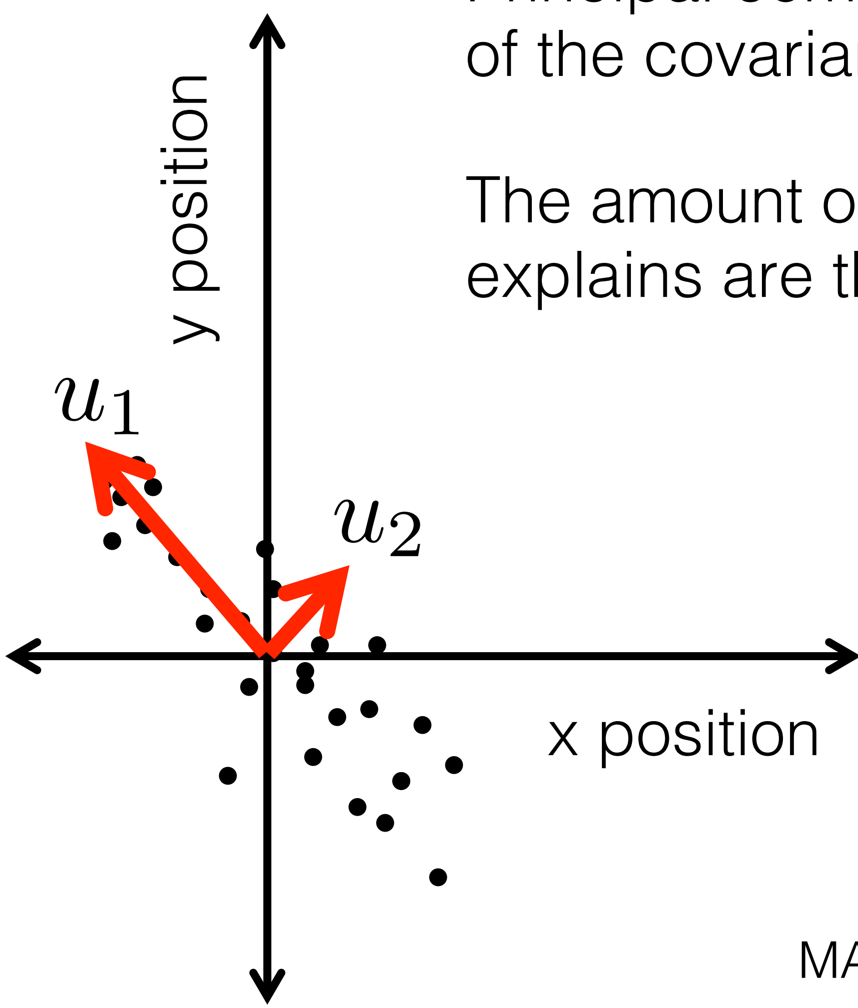
Ok, so now what is PCA?



Principal components **U** are the eigenvectors of the covariance matrix.

MATLAB: `U, W, lambda = princomp(X)`

Ok, so now what is PCA?

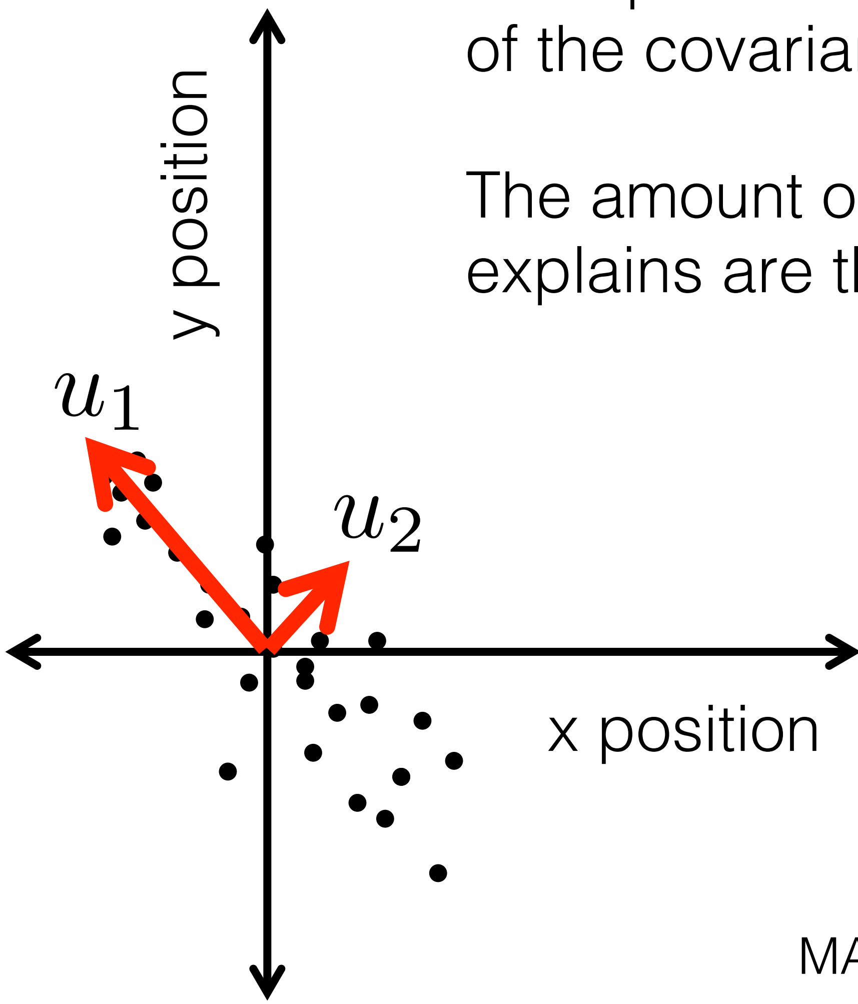


Principal components `U` are the eigenvectors of the covariance matrix.

The amount of variance each component explains are the normalized eigenvalues `lambda`

MATLAB: `U, W, lambda = princomp(X)`

Ok, so now what is PCA?



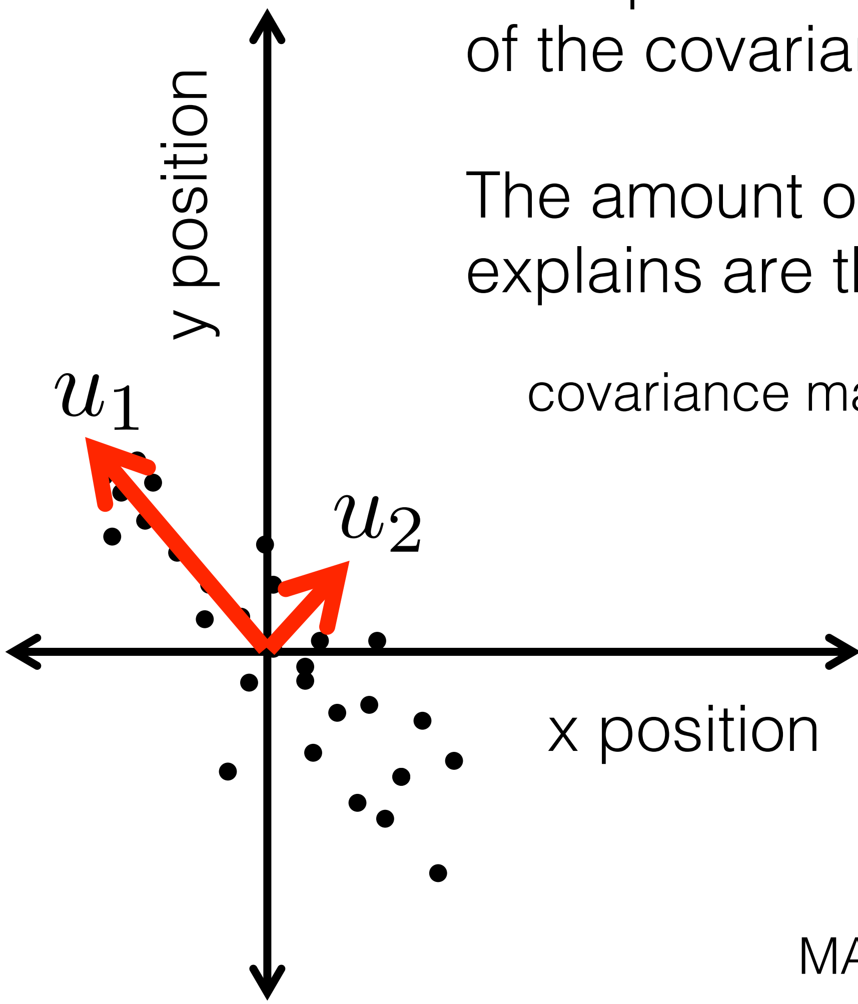
Principal components `U` are the eigenvectors of the covariance matrix.

The amount of variance each component explains are the normalized eigenvalues `lambda`

$$XX^T = U\Lambda U^T$$

MATLAB: `U, W, lambda = princomp(X)`

Ok, so now what is PCA?



Principal components **U** are the eigenvectors of the covariance matrix.

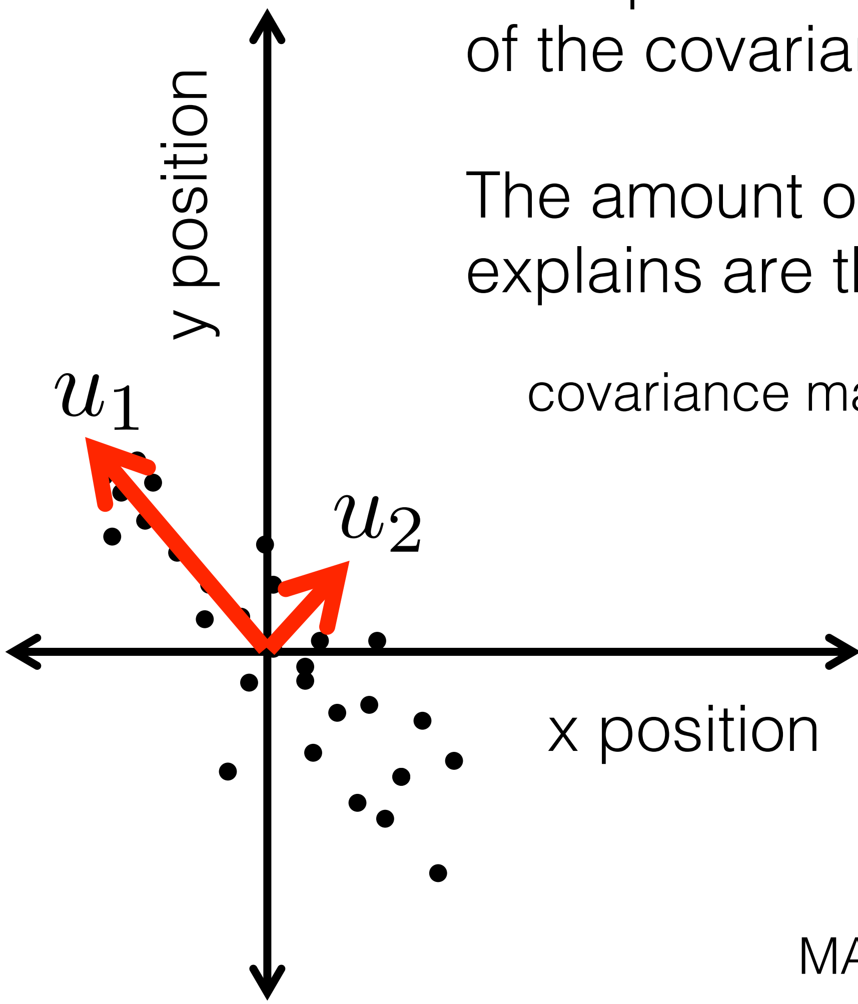
The amount of variance each component explains are the normalized eigenvalues `lambda`

covariance matrix of your dataset **X**

$$XX^T = U\Lambda U^T$$

MATLAB: `U, W, lambda = princomp(X)`

Ok, so now what is PCA?



Principal components **U** are the eigenvectors of the covariance matrix.

The amount of variance each component explains are the normalized eigenvalues `lambda`

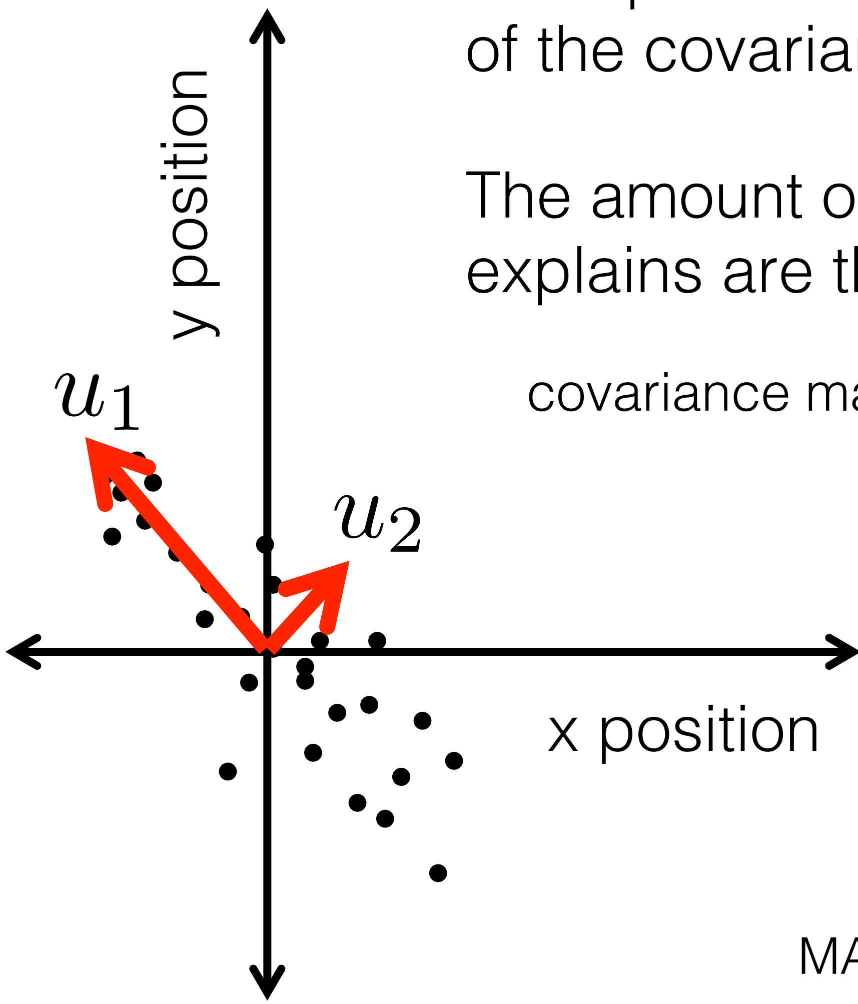
covariance matrix of your dataset **X**

$$XX^T = U\Lambda U^T$$

each column is an eigenvector

MATLAB: `U, W, lambda = princomp(X)`

Ok, so now what is PCA?



Principal components `U` are the eigenvectors of the covariance matrix.

The amount of variance each component explains are the normalized eigenvalues `lambda`

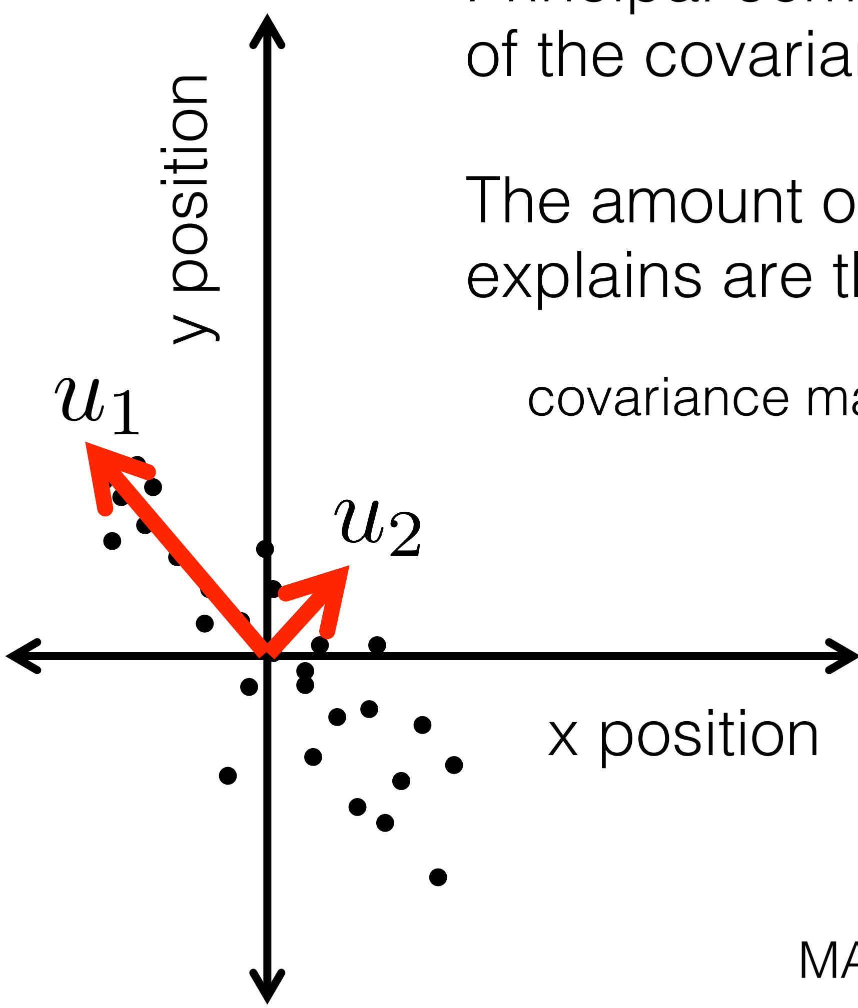
covariance matrix of your dataset `X`

$$XX^T = U\Lambda U^T$$

each column is an eigenvector

MATLAB: `U, W, lambda = princomp(X)`

Ok, so now what is PCA?



Principal components **U** are the eigenvectors of the covariance matrix.

The amount of variance each component explains are the normalized eigenvalues `lambda`

covariance matrix of your dataset **X**

$$XX^T = U\Lambda U^T$$

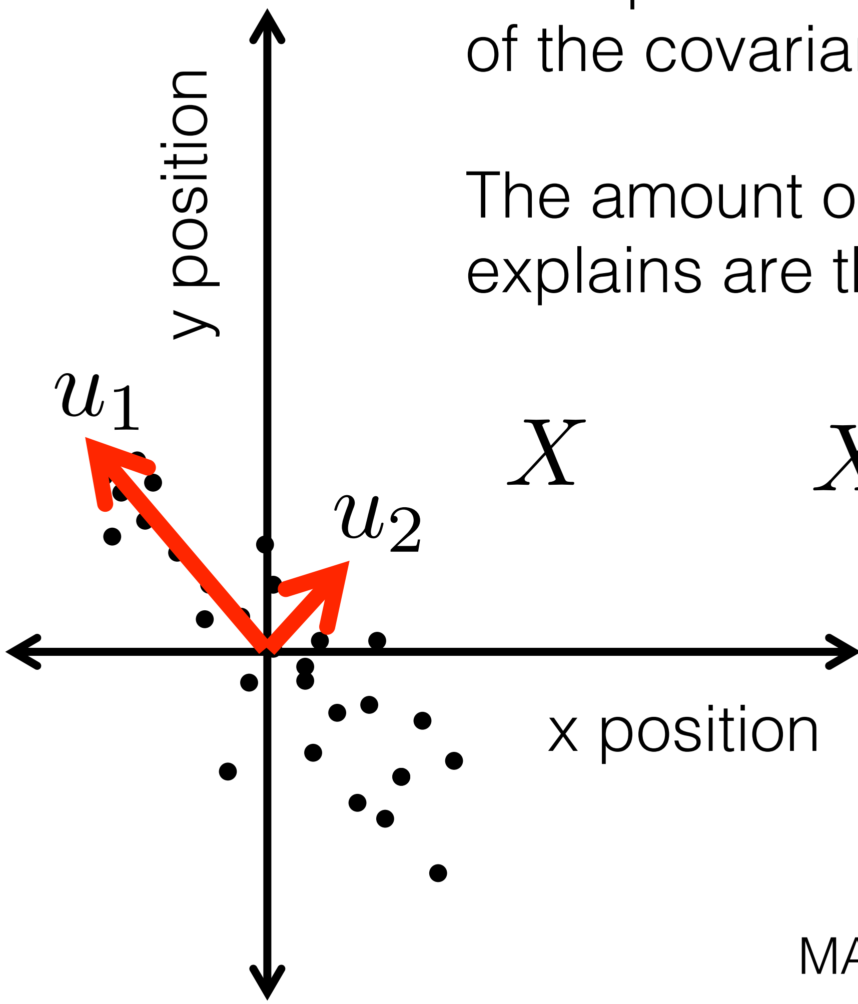
each row is an eigenvector

MATLAB: `U, W, lambda = princomp(X)`

Ok, so now what is PCA?

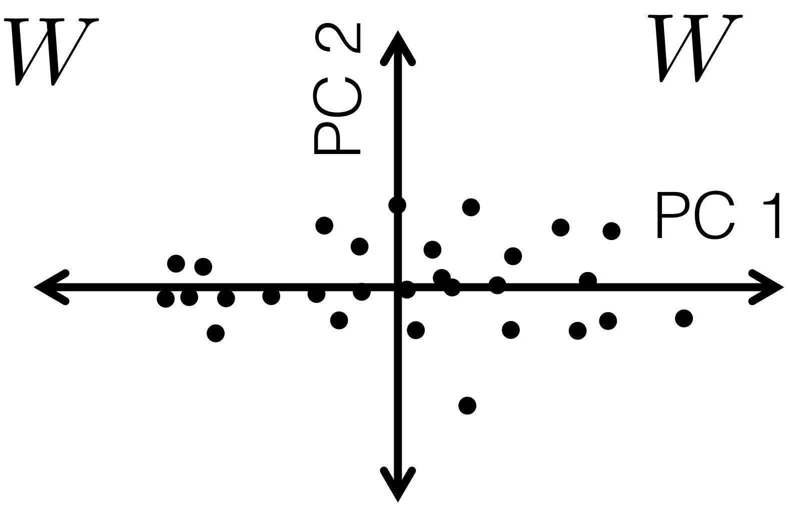
Principal components **U** are the eigenvectors of the covariance matrix.

The amount of variance each component explains are the normalized eigenvalues `lambda`



X

$$XU = W$$



MATLAB: `U, W, lambda = princomp(X)`

The End

Acknowledgements

Some of these slides were adapted from a Linear Algebra lecture presented at the 2014 Woods Hole *Methods in Computational Neuroscience* course by Mark Goldman and Emily Mackevicius.