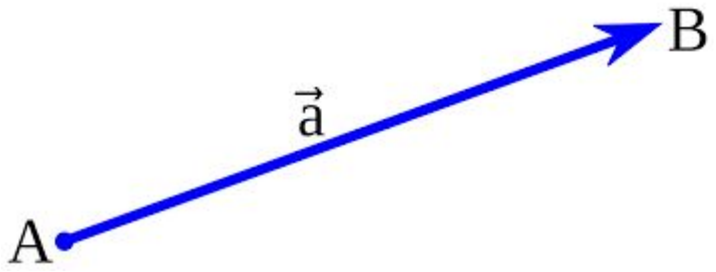
$$\begin{bmatrix} 9 & 13 & 5 & 2 \\ 1 & 11 & 7 & 6 \\ 3 & 7 & 4 & 1 \\ 6 & 0 & 7 & 10 \end{bmatrix}$$

Linear Algebra Introduction



Jessica Curley (credit Frank Burkholder and Taryn Heilman)

Data Science Immersive

Linear Algebra is for Everyone!

Objectives

Lecture

- Be able to define scalars, vectors, and matrices
- Translate real-world data into scalars, vectors and arrays
    - Featurizing your data
- Describe operations that define vector length and different distances between vectors
- Describe operations that multiply vectors and matrices

Self-guided Jupyter Notebook:

- Describe the Python library NumPy and define its fundamental data structure
- Use NumPy to:
    - Create **scalars**, **vectors**, and **matrices**
    - Find the **L1** and **L2 norms** of a vector
    - Find the **euclidean, manhattan**, and **cosine** distances between two vectors
    - Find the **dot product** of two matrices

Scalars

- A **scalar** is quantity that only has a magnitude (not a direction)
  - The markup was **$7** per item
  - The check was $**16.72** per person
  - The width is **1.5** times the height

- Typographically represented as a lowercase variable:

  - ρ (rho)
  - T
  - x

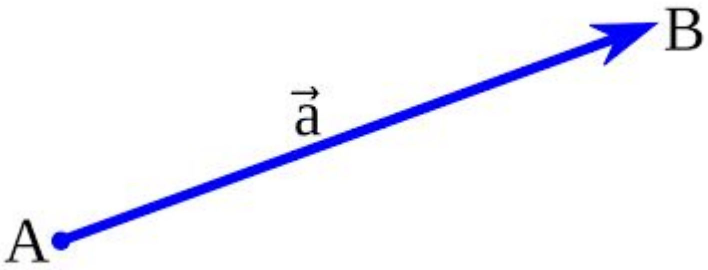
- Often used to **scale** objects (arrays) by a constant factor

Vectors

A **vector** is a quantity that has a magnitude and a direction. You can think of axes (columns) of data as directions that define the vector space of the data, and the values inside as the magnitude in each of those directions. Vectors are often represented as sequences of values.

Notation: Vectors are often seen written as either:

Bold, lowercase	lowercase italic with right pointed arrow
$\mathbf{x}$	x



(b1, b2, …, bn)

(a1, a2, …, an)
Or the origin

Vectors

A **vector** is a quantity that has a magnitude and a direction. You can think of axes (columns) of data as directions that define the vector space of the data, and the values inside as the magnitude in each of those directions. Vectors are often represented as sequences of values.

Notation: Vectors are often seen written as either:

Bold, lowercase	lowercase italic with right pointed arrow
$\mathbf{x}$	$\vec{x}$

Question: What other Python object would represent a vector?

Vectors

A **vector** is a quantity that has a magnitude and a direction. You can think of axes (columns) of data as directions that define the vector space of the data, and the values inside as the magnitude in each of those directions. Vectors are often represented as sequences of values.

Notation: Vectors are often seen written as either:

Bold, lowercase	lowercase italic with right pointed arrow
$\mathbf{x}$	$\vec{x}$

Question: What other Python object would represent a vector?

Answer: A 1 dimensional array!

Vectors

A **vector** is a quantity that has a magnitude and a direction. You can think of axes (columns) of data as directions that define the vector space of the data, and the values inside as the magnitude in each of those directions. Vectors are often represented as sequences of values.

Notation: Vectors are often seen in textbooks:

Bold, lowercase lowercase italic with right pointed arrow

$\mathbf{x}$ $\vec{x}$

Example:

- We were driving **30 mph** going **North** on **I-25**.

Speed North	Speed East	On I25
30	0	True

Vectors

A **vector** is a quantity that has a magnitude and a direction. You can think of axes (columns) of data as directions that define the vector space of the data, and the values inside as the magnitude in each of those directions.Vectors are often represented as sequences of values.

Notation: Vectors are often seen in textbooks:

Bold, lowercase — $\mathbf{x}$

lowercase italic with right pointed arrow — $\vec{x}$

Example:

- We were driving **30 mph** going **North** on **I-25**.

Speed North	Speed East	On I25
30	0	True

Numerical

Categorical

Binary encoding
0 = False
1 = True
It’s still categorical (but a numerical value).

Vectors - examples

- The well in **Colorado** was **8000 ft deep** and **10 ft wide** with a **100 gpm** capacity.

State	Depth	Width	Capacity
CO	8000	10	100

Vectors - examples

- The well in **Colorado** was **8000 ft deep** and **10 ft wide** with a **100 gpm** capacity.

State	Depth	Width	Capacity
CO	8000	10	100

What do these values mean?

(If you came back to this data two weeks from now, would you remember?)

Vectors - examples

- The well in **Colorado** was **8000 ft deep** and **10 ft wide** with a **100 gpm** capacity.

State_abrv	Depth_ft	Width_ft	Capacity_gpm
CO	8000	10	100

Consider including units of measure in column names.

Vectors - examples

- I bought **2 boxes of oatmeal**, **1 box of crackers**, at the **supermarket** for only **$2.50**.

Oatmeal	Crackers	Location	Price
2	1	supermarket	2.5

Vectors - examples

- I bought **2 boxes of oatmeal**, **1 box of crackers**, at the **supermarket** for only **$2.50**.

Oatmeal	Crackers	Location	Price
2	1	supermarket	2.5

It’s a categorical variable, but models can’t train on text. How to make this a number?

Vectors - examples

- I bought **2 boxes of oatmeal**, **1 box of crackers**, at the **supermarket** for only **$2.50**.

Oatmeal	Crackers	Location	Price
2	1	supermarket	2.5

Let’s say options (categories) were:
supermarket, corner store, and on-line.
Use one-hot encoding.

One-hot encoding: When an **encoded** variable is removed and a new binary variable is added for each unique integer value.

Vectors - examples

- I bought **2 boxes of oatmeal**, **1 box of crackers**, at the **supermarket** for only **$2.50**.

Oatmeal	Crackers	supermarket	corner_store	on-line	Price
2	1	1	0	0	2.5

The purchase location has been
one-hot encoded.

One-hot encoding: When an **encoded** variable is removed and a new binary variable is added for each unique integer value.

Vectors - examples

- I bought **2 boxes of oatmeal**, **1 box of crackers**, at the **supermarket** for only **$2.50**.

Oatmeal	Crackers	supermarket	Price
2	1	1	2.5

Then again, maybe it only matters if the purchased happened at the supermarket or not...

Vectors - examples

- Can you make a vector of this data-science haiku?

  My profile is me
  A bunch of features - that’s all
  But where is my soul?

Vectors - examples

- Can you make a vector of this data-science haiku?

  My profile is me
  A bunch of features - that's all
  But where is my soul?

  Consider using a bag-of-words approach, where you count the number of times each word occurs (term frequency).

Vectors - examples

- Can you make a vector of this data-science haiku?

  My profile is me
  A bunch of features - that's all
  But where is my soul?

  Consider using a bag-of-words approach, where you count the number of times each word occurs (term frequency).

a	all	bunch	but	features	is	me	my	profile	soul	that's	where

Vectors - examples

- Can you make a vector of this data-science haiku?

  My profile is me
  A bunch of features - that's all
  But where is my soul?

  Consider using a bag-of-words approach, where you count the number of times each word occurs (term frequency).

a	all	bunch	but	features	is	me	my	profile	soul	that's	where

the vocabulary

Vectors - examples

- Can you make a vector of this data-science haiku?

  My profile is me
  A bunch of features - that's all
  But where is my soul?

  Consider using a bag-of-words approach, where you count the number of times each word occurs (term frequency).

a	all	bunch	but	features	is	me	my	profile	soul	that's	where
1	1	1	1	1	2	1	2	1	1	1	1

the vocabulary

the counts

Vectors - examples

- Can you make a 1-D vector (a row) describing the status of this tic-tac-toe game?

X		
	O	O
		X

Vectors - examples

- Can you make a 1-D vector (a row) describing the status of this tic-tac-toe game?

X		
	O	O
		X

Consider flattening this 2-D array into 1-D.
Values: -1 (unplayed), 0 (O), and 1 (X).

Vectors - examples

- Can you make a 1-D vector (a row) describing the status of this tic-tac-toe game?

1	-1	-1
-1	0	0
-1	-1	1

Top Left	Top Middle	Top Right
Middle Left	Middle	Middle Right
Bottom Left	Bottom Middle	Bottom Right

Consider flattening this 2-D array into 1-D.
Values: -1 (unplayed), 0 (O), and 1 (X).

Vectors - examples

- Can you make a 1-D vector (a row) describing the status of this tic-tac-toe game?

X		
	O	O
		X

1	-1	-1
-1	0	0
-1	-1	1

TL	TM	TR
ML	M	MR
BL	BM	BR

The 1-D vector:

TL	TM	TR	ML	M	MR	BL	BM	BR
1	-1	-1	-1	0	0	-1	-1	1

Vectors - moral of the story

- Almost any type of data can be vectorized/featurized (doesn’t always have to be 1-D).
- Almost always, there is more than one way organize data as a vector.
- Your ability to featurize and feature engineer your data will be fundamental to the success of your machine learning algorithms and your career as a data scientist.
- Your intuition will grow with experience (and by reading books, papers, and blogs!)

Matrices

- A **matrix** is a rectangular array of values arranged in rows and columns. We can think of an array as **rows of vectors**, or **columns of vectors.**

Well	Depth	Width
1	8000	10
2	1000	3
3	10000	5

- We can also consider a the rows of a matrix as observations, or samples.
- The term frequency matrix of these data-science haikus:

My profile is me
A bunch of features - that's all
But where is my soul?

As data unfold
Alas the model won't fit
Still the curve bends right

haiku	a	all	alas	as	bends	bunch	but	curve	data	featur es	is	model	my	profile	right	soul	still	that's	the	unfold	where	won't
1	1	1	0	0	0	1	1	0	0	1	2	0	2	1	0	1	0	1	0	0	1	0
2	0	0	1	1	1	0	0	1	1	0	0	1	0	0	1	0	1	0	2	1	0	1

Breakout 1: Vectors

1. Pick at least three of these topics. Come up with how you would vectorize (*featurize*) this data with an example or description (bonus points for both!)

a) The location and orientation of an aerial drone
b) A black-and-white image
c) A gray-scale image
d) A movie review
e) Stock price
f) A color image
g) Music you hear at a concert

2. Recall the tic-toc game as an array or a 1D vector.

How would you rescale the values so every -1 becomes a -5 and 1 becomes a 5?

Vector operations

- Length of a vector
  - L1 & L2 norms
- Distance between vectors
  - euclidean distance (L2)
  - manhattan distance (L1)
  - cosine distance (which uses a dot product)
- Matrix multiplication
  - dot product

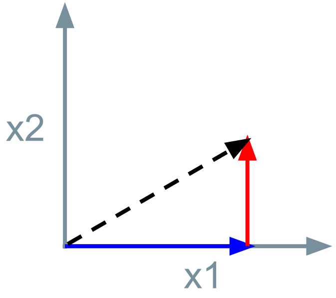
Length of a vector

Euclidean norm [edit]

Main article: Euclidean distance

On an n-dimensional Euclidean space $\mathbf{R}^n$, the intuitive notion of length of the vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is captured by the formula

$$\|\boldsymbol{x}\|_2 := \sqrt{x_1^2 + \cdots + x_n^2}.$$



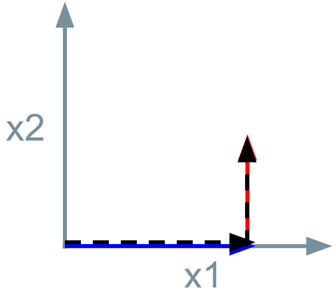
L2 norm: $\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2}$

Taxicab norm or Manhattan norm [edit]

Main article: Taxicab geometry

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i| .$$

The name relates to the distance a taxi has to drive in a rectangular street grid to get from the origin to the point x.



L1 norm: $\|\boldsymbol{x}\|_1 = |x_1| + |x_2|$

source: Wikipedia

Distance between vectors: Euclidean

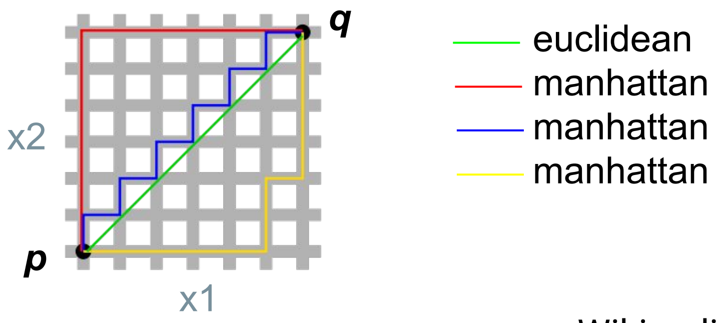
Euclidean Distance: Most often thought as "the shortest distance between two points"

The **Euclidean distance** between points **p** and **q** is the length of the line segment connecting them ($\overline{\mathbf{pq}}$).

In Cartesian coordinates, if **p** = (p_1, p_2,..., p_n) and **q** = (q_1, q_2,..., q_n) are two points in Euclidean *n*-space, then the distance (d) from **p** to **q**, or from **q** to **p** is given by the Pythagorean formula:[1]

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$



source: Wikipedia

Distance between vectors: Manhattan

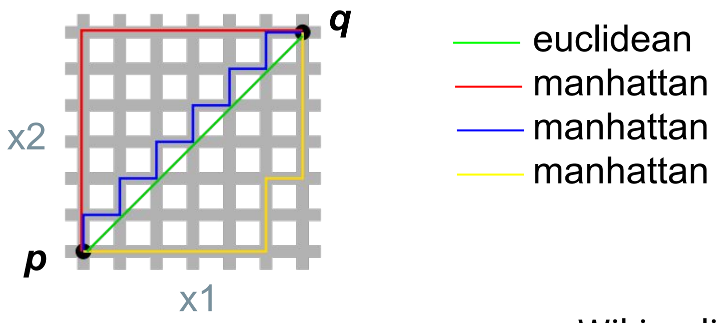
Manhattan, or taxicab distance: The distance between two points measured along axes at right angles.

The taxicab distance, d_1, between two vectors $\mathbf{p}, \mathbf{q}$ in an n-dimensional real vector space with fixed Cartesian coordinate system, is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes. More formally,

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

where $(\mathbf{p}, \mathbf{q})$ are vectors

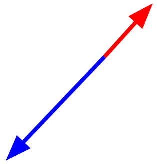
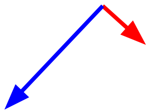
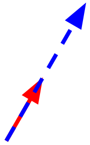
$$\mathbf{p} = (p_1, p_2, \dots, p_n) \text{ and } \mathbf{q} = (q_1, q_2, \dots, q_n)$$



source: Wikipedia

Distance between vectors: Cosine

- Cosine distance = 1 - cosine similarity
- We want to know: are the vectors pointing in the same direction? If so, they're similar!
- Typically used in higher dimensional vector spaces like text analysis because we care about similarity rather than distance

Metric	Vectors pointing in opposite directions	Vectors orthogonal	Vectors pointing in the same direction
Image			
Cosine similarity	-1	0	1
Cosine distance	2	1	0

Calculation of Cosine Similarity

The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \|\mathbf{B}\| \cos\theta$$

Given two vectors of attributes, *A* and *B*, the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

where A_i and B_i are components of vector A and B respectively.

source: Wikipedia

Dot product

In mathematics, the **dot product** or **scalar product**[note 1] is an algebraic operation that takes two equal-length sequences of numbers (usually coordinate vectors) and returns a single number.

The dot product of two vectors **a** = $[a_1, a_2, \ldots, a_n]$ and **b** = $[b_1, b_2, \ldots, b_n]$ is defined as:[1]

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

where Σ denotes summation and n is the dimension of the vector space. For instance, in three-dimensional space, the dot product of vectors [1, 3, −5] and [4, −2, −1] is:

$$\begin{aligned} [1, 3, -5] \cdot [4, -2, -1] &= (1 \times 4) + (3 \times -2) + (-5 \times -1) \\ &= 4 - 6 + 5 \\ &= 3 \end{aligned}$$

source: Wikipedia

Dot product - extrapolating to matrix multiplication

If vectors are identified with row matrices, the dot product can also be written as a matrix product

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}\mathbf{b}^\top,$$

where $\mathbf{b}^\top$ denotes the transpose of $\mathbf{b}$.

Expressing the above example in this way, a 1 × 3 matrix (row vector) is multiplied by a 3 × 1 matrix (column vector) to get the a 1 × 1 matrix that is identified with its unique entry:

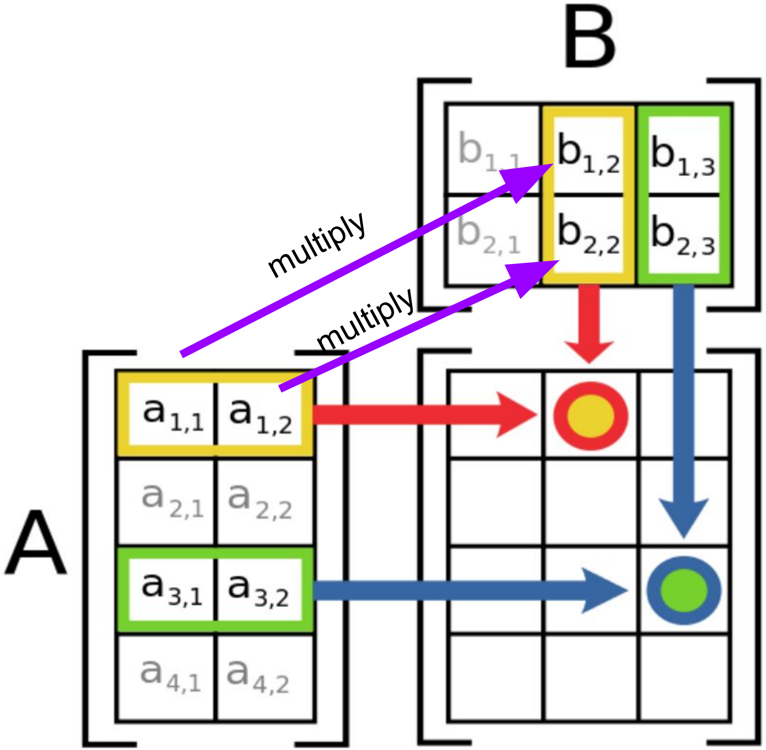
$$\begin{bmatrix} 1 & 3 & -5 \end{bmatrix} \begin{bmatrix} 4 \\ -2 \\ -1 \end{bmatrix} = 3.$$

source: Wikipedia

Matrix Multiplication

The figure to the right illustrates diagrammatically the product of two matrices $\mathbf{A}$ and $\mathbf{B}$, showing how each intersection in the product matrix corresponds to a row of $\mathbf{A}$ and a column of $\mathbf{B}$.

$$\overset{4\times 2 \text{ matrix}}{\begin{bmatrix} a_{11} & a_{12} \\ \cdot & \cdot \\ a_{31} & a_{32} \\ \cdot & \cdot \end{bmatrix}} \overset{2\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & b_{12} & b_{13} \\ \cdot & b_{22} & b_{23} \end{bmatrix}} = \overset{4\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & x_{12} & x_{13} \\ \cdot & \cdot & \cdot \\ \cdot & x_{32} & x_{33} \\ \cdot & \cdot & \cdot \end{bmatrix}}$$



source: Wikipedia

Matrix Multiplication: Solving Equations

Given matrices A, B and C.
If A • B = C, how would you solve for B?

$$\underbrace{\begin{bmatrix} 5 & -4 \\ 1 & 2 \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 8 \\ 6 \end{bmatrix}}_{C}$$

Matrix Multiplication: Solving Equations

Given matrices A, B and C.
If A • B = C, how would you solve for B?

$A^{-1} \bullet C = B$

$$\underbrace{\begin{bmatrix} 5 & -4 \\ 1 & 2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 8 \\ 6 \end{bmatrix}}_{C} \longrightarrow \underbrace{\begin{bmatrix} 5 & -4 \\ 1 & 2 \end{bmatrix}^{-1}}_{A^{-1}} \underbrace{\begin{bmatrix} 8 \\ 6 \end{bmatrix}}_{C} = \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{B}$$

Breakout 2 - Basic Matrix Multiplication

A =

1	2
-1	0

B =

1	2	0
1	-1	3

1) What is A • B?
2) What is B • A?
3) Can you do B • B?
4) How about B • B^T?

Breakout 3 - Matrix Multiplication Word Problem

We want to find the final grades for 3 students, and we know what their averages are for tests, projects, homework, and quizzes. We also know that tests are 40% of the final grade, projects 15%, homework is 25%, and quizzes are 20%. The student data is in the chart below:

Student	**Tests**	**Projects**	**Homework**	**Quizzes**
Sam	92	100	89	80
Li	72	85	80	75
Alex	88	78	85	90

Question: What are the final grades for Sam, Li and Alex? Use matrix multiplication - on paper or with NumPy.

Credit: shelovesmath.com

Breakout 4 - Solve a Linear Algebra Problem

Question: Solve for x and y using matrix multiplication

$$-3x + 4y = 5$$

$$2x - y = -10$$

$$\begin{bmatrix} -3 & 4 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ -10 \end{bmatrix}$$

source: slideplayer.com

Objectives

Lecture

- Be able to define scalars, vectors, and matrices
- Translate real-world data into scalars, vectors and arrays
  - Featurizing your data
- Describe operations that define vector length and different distances between vectors
- Describe operations that multiply vectors and matrices

Self-guided Jupyter Notebook:

- Describe the Python library NumPy and define its fundamental data structure
- Use NumPy to:
  - Create **scalars**, **vectors**, and **matrices**
  - Find the **L1** and **L2 norms** of a vector
  - Find the **euclidean, manhattan**, and **cosine** distances between two vectors
  - Find the **dot product** of two matrices

Some notes before you go

- Do the self-guided `linear-algebra-with-numpy.ipynb` for NumPy applications
- Markov chains are very cool and shouldn't be intimidating
- There's Matplotlib and Pandas in the assignment. The solutions are there for help with that

Self Guided Jupyter Notebook

`linear-algebra-with-numpy.ipynb`

Some Breakout Answers

Breakout 1:

2) Multiply the vector by a scalar of 5

Breakout 2:

1) [3 0 6
   -1 -2 0]
2) Impossible!
3) Nope. Not square…
4) [-1 2
   -1, -2]

Breakout 3:

Matrix	Multiplication
$\begin{bmatrix}92 & 100 & 89 & 80\\ 72 & 85 & 80 & 75\\ 88 & 78 & 85 & 92\end{bmatrix} \times \begin{bmatrix}.4\\ .15\\ .25\\ .2\end{bmatrix} = \begin{bmatrix}90\\ 77\\ 87\end{bmatrix}$	Think of turning the first matrix to the right and sideways, multiplying each number by the numbers in the second matrix, and then adding them together. For example, $(92 \times .4) + (100 \times .15) + (89 \times .25) + (80 \times .2) = 90.05$ $(72 \times .4) + (85 \times .15) + (80 \times .25) + (75 \times .2) = 76.55$ $(88 \times .4) + (78 \times .15) + (85 \times .25) + (92 \times .2) = 86.55$ She Loves Math

Breakout 4:

x = - 7

y = -4