

INTRO TO DATA SCIENCE LESSON 1: BASIC PYTHON AND LINEAR ALGEBRA

LAST TIME...

2

WHAT IS DATA SCIENCE

DATA EXPLORATION AND WORKFLOW

PYTHON DATA STRUCTURES

QUESTIONS?

INTRO TO DATA SCIENCE

I. LINEAR ALGEBRA REVIEW

I. LINEAR ALGEBRA REVIEW

II. THE PYTHON CONTROL FLOW

LAB:

III. MATRIX OPERATIONS IN PYTHON

**IV. ADDING CONTROL FLOW TO CLICKS
AGGREGATION**

QUESTIONS?

In order to understand most machine learning algorithms, we first need to learn the basics of Linear Algebra.

What is Linear Algebra?

In order to understand most machine learning algorithms, we first need to learn the basics of Linear Algebra.

What is Linear Algebra?

Linear Algebra is defined as mathematics in multidimensional spaces and the mapping between those spaces.

$$y = mx + b$$

$$y = m_1x_1 + m_2x_2 + b$$

$$y = m_1x_1 + m_2x_2 + m_3x_3 \\ m_4x_4 + m_5x_5 + m_6x_6 + b$$

Matrices are arrays of real numbers with m rows and n columns

Each value in a matrix is called an entry

1	5	8	7
2	1	3	6
3	5	1	0
4	6	0	1

Matrices are arrays of real numbers with m rows and n columns

Each value in a matrix is called an entry

$$A_{2\ 1} \longrightarrow \begin{matrix} 1 & 5 & 8 & 7 \\ 2 & 1 & 3 & 6 \\ 3 & 5 & 1 & 0 \\ 4 & 6 & 0 & 1 \end{matrix}$$

Rule 1!

Matrices can be added together only when they are the same size. If they are not the same size, their sum is undefined.

$$[1 \ 3 \ 9 \ 2] + [2 \ 5 \ 9 \ 4] = [3 \ 8 \ 18 \ 6]$$

Rule 1!

Matrices can be added together only when they are the same size. If they are not the same size, their sum is undefined.

$$\begin{bmatrix} 1 & 3 & 9 & 2 \end{bmatrix} + \begin{bmatrix} 2 & 5 & 9 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 8 & 18 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 8 & 72 & 3 & 1 \end{bmatrix} + \begin{bmatrix} 17 & 55 & 3 & 10 \end{bmatrix} = ?$$

Rule 2!

Matrices can be multiplied by a scalar (single entity) value

Each value in the matrix is multiplied by the scalar value.

$$[1 \ 3 \ 9 \ 2] \times 3 = [3 \ 9 \ 27 \ 6]$$

$$[8 \ 72 \ 3 \ 1] \times 3 = ?$$

Rule 3!

Matrices and vectors can be multiplied together given that the matrix columns are as wide as the vector is long.

The result will always be a vector.

$$\begin{array}{cccc} 1 & 3 & 9 & 2 \\ 2 & 4 & 6 & 8 \end{array} \begin{array}{c} * \\ 3 \\ 6 \\ 5 \end{array} = \begin{array}{cc} 2+9+54+10 & 75 \\ 4+12+36+40 & 92 \end{array}$$

Rule 4!

Matrices can be multiplied together using the same rules that we have from matrix-vector multiplication.

The result will always be a matrix.

$$\begin{array}{cccc} & & 2 & 1 \\ 1 & 3 & 9 & 2 \\ 2 & 4 & 6 & 8 \end{array} * \begin{array}{cc} 3 & 2 \\ 6 & 0 \\ 5 & 4 \end{array} = \begin{array}{cc} 75 & 15 \\ 92 & 40 \end{array}$$

Matrices represent the multiple dimensions in our data! If we had a vector that suggested how important each dimension of our data was, we could use that to find our best **linear model**.

Matrices represent the multiple dimensions in our data! If we had a vector that suggested how important each dimension of our data was, we could use that to find our best **linear model**.

We will see matrices quite often in **all** of our data, so pay careful attention to how data is structured and how different algorithms interact with them

REVIEW

1. Complete the equations on the board.

II. PYTHON CONTROL FLOW

Python has a number of control flow tools that will be familiar from other languages. The first is the if-else statement, whose compound syntax looks like this:

```
>>> x, y = False, False
>>> if x:
...     print 'apple'
... elif y:
...     print 'banana'
... else:
...     print 'sandwich'
...
sandwich
```

Next is the while loop. **This executes while a given condition evaluates to True.**

```
>>> x = 0
>>> while True:
...     print 'HELLO!'
...     x += 1
...     if x >= 3:
...         break
...
HELLO!
HELLO!
HELLO!
```

Another familiar (and useful) construct is the for loop. This executes a block of code for a range of values.

```
>>> for k in range(4):  
...     print k**2  
...  
0  
1  
4  
9
```

The object that a for loop iterates over is called (appropriately) an iterable.

A useful but possibly unfamiliar construct is the try-except block:

```
>>> try:
...     print undef
... except:
...     print 'nice try'
...
nice try
```

This is useful for catching and dealing with errors, also called exception handling.

Python allows you to define custom functions as you would expect:

```
>>> def x_minus_3(x):  
...     return x - 3  
...  
>>> x_minus_3(12)  
9
```

Functions can optionally return a value with a return statement (as this example does).

Functions can take a number of arguments as inputs, and these arguments can be specified in two ways:

As positional arguments:

```
>>> def f(x, y):  
...     return x - y  
...  
>>> f(4,2)  
2  
>>> f(2,4)  
-2
```

Functions can take a number of arguments as inputs, and these arguments can be specified in two ways:

Or as keyword arguments:

```
>>> def g(arg1=x, arg2=y):  
...     return arg1 / float(arg2)  
...  
>>> g(arg1=10, arg2=5)  
2.0  
>>> g(arg2=100, arg1=10)  
0.1
```

INTRO TO DATA SCIENCE

LAB: MATRIX MANIPULATION IN PYTHON

IN CLASS WORK

1. Change our python script to also return minimum, maximum, and average age, and click through rate (clicks/impressions)
2. Homework: Update the script to write a new file instead of using standard out and save it to output