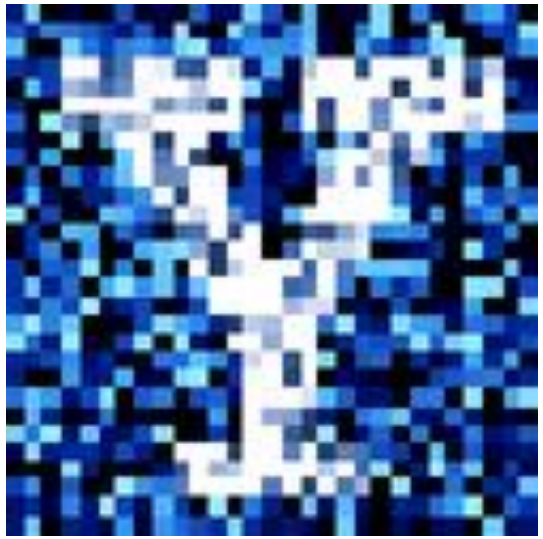
YData: Introduction to Data Science



Lecture 13: Iteration

Overview

Quick review of Comparisons

Conditional statements

Appending arrays

Randomly selecting elements

If there is time: for loops

Announcements

Homework 4 has been posted

- Due Sunday February 27th at 11pm

Project 1 has been posted

- Due Friday March 4th
- You are allowed to work with one other person on the project
  - You can not discuss the project with anyone else part from the TAs
  - If you do not have a partner to work on the project with, you can sign up using this link: https://bit.ly/YPartner

Comparisons

Comparisons

We can use mathematical operators to compare numbers and strings

- Results return Boolean values True and False

Comparison	Operator	True example	False Example
Less than	<	2 < 3	2 < 2
Greater than	>	3 > 2	3 > 3
Less than or equal	<=	2 <= 2	3 <= 2
Greater or equal	>=	3 >= 3	2 >= 3
Equal	==	3 == 3	3 == 2
Not equal	!=	3 != 2	2 != 2

We can also make comparisons across elements in an array

Let's explore this in Jupyter!

Conditional statements

Conditional statements

Conditional statements control the sequence of computations that are performed in a program

We use the keyword if to begin a conditional statement to only execute lines of code if a particular condition is met.

We can use elif to test additional conditions

We can use an else statement to run code if none of the if or elif conditions have been met.

```
num = 5
if num == 1:
    print("Monday")
elif num == 2:
    print("Tuesday")
elif num == 3:
    print("Wednesday")
elif num == 4:
    print("Thursday")
elif num == 5:
    print("Friday")
elif num == 6:
    print("Saturday")
elif num == 7:
    print("Sunday")
else:
    print("Invalid input")
```

Let's explore this in Jupyter!

Appending arrays

Appending arrays

We can append values or another array onto an existing array

np.append(array1, value)

- value appended to array1
- the value has to be of the same type as the elements of array1

np.append(array1, array2)

- array2 appended to array1
- array2 elements must have the same type as array1 elements

Let's explore this in Jupyter!

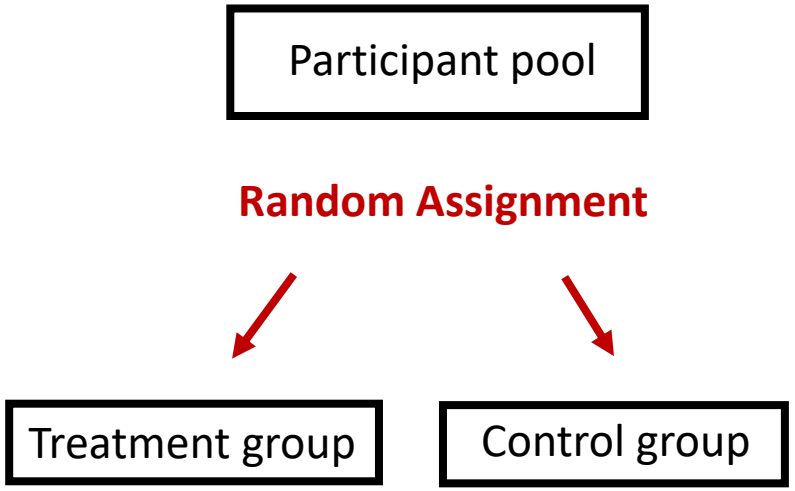
Randomly selecting elements

Randomly selecting elements

We can randomly select elements from an array using np.random.choice()

np.random.choice(array, size)

- array: an array to select random elements from
  - Elements selected uniformly random ***with replacement***
- size: number of elements to randomly select



Have we seen any examples of randomly choosing items in this class?

Let's explore this in Jupyter!

For loops

For loops

For loops repeat a process many times, iterating over a sequence of items

- Often we are iterating over an array of sequential numbers

```
animals = make_array("cat", "dog", "bat")
for creature in animals:
    print(creature)

for i in np.arange(4):
    print(i**2)
```

Let's explore this in Jupyter!