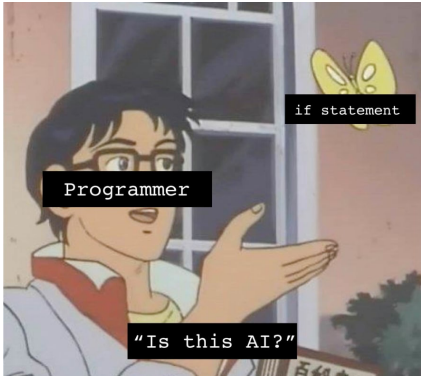
DATA 8
Spring 2021

Discussion 6



Control and Iteration

Materials: tinyurl.com/d8-disc06
or access through kevin-miao.com under teaching
Meme: Email me any fun Data Science memes!

Today

- Announcements
- Review: Control Flow and Iteration
- Worksheet
  - Link: www.tinyurl.com/d8-disc06

Announcements

- **Assignment deadlines**
  - **Vitamin 6** is due tonight
  - **Homework 5** is due Thursday
  - **Project 1** is due Friday
  - Submit homework & projects one day early for bonus point
- **Regrades** for homework 1 & 3 and lab 4 **due Friday**
  - Gradescope: Submit regrade via button
  - OkPy: Email me
- *Informal OH:* Feel free to stay after discussion, if you have homework/project/course related questions. I booked off time from 9-9:30 AM.

Conditionals

- **Objective:** We want to run different code depending on the value of a certain variable.
- **Example**: We want to know whether we need to wear a sweater based on the temperature
- **Usage:**

```
if temperature < 60:
        print('Sweater weather')
elif temperature < 70:
        print('Only if you are from SoCal😜')
else:
        print('Nope!')
```

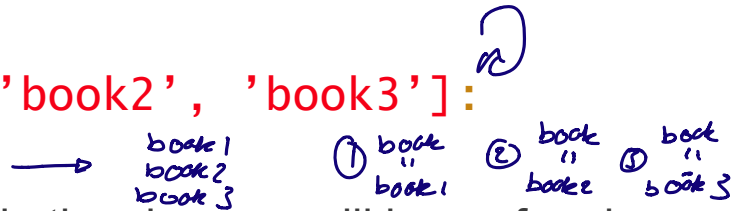
Mandatory part of the if statement!

optional

Iteration

- **Objective:** We want to do the same thing/call the same function for each item in a list/array
- **Usage:**

```
for book in ['book1', 'book2', 'book3']:
    print(book)
```



- **Why are we learning this?** Later in the class we will be performing simulations of chance experiments (i.e. rolling a die 200 times)

```
x = 0
for i in np.arange(0,100):
    x += 1
```

[0, ... , 99]

$x = x + 1$ ⟷ `x += 1`

(old value)

To the worksheet! ✍️

tinyurl.com/d8-disc06

Question 1

Question 1. What does the following function do? Fill out the docstring description for the function to include what the inputs should be and what the function does. *Hint: try to figure out what the function would do on different inputs.*

```
def mystery_function(n1, n2):
    """_____"""
    if n2 - n1 > 0:
        return n2 - n1
    elif n2 - n1 < 0:
        return n1 - n2
    else:
        return 0
```

n2 = 10 8
n1 = 8 10

10 - 8 = 2

10 - 8 = 2

Absolute difference between n1 and n2.

Question 2

Question 2. The instructor of a lower division statistics class has assigned you a task: make a function that takes in a student's score on a scale from 0 to 100 and assigns a letter grade based on the following grade boundaries.

Score	Letter Grade
0-69	F
70-79	C
80-89	B
90-100+	A

Complete the function compute_letter_grades. It takes in a student's score and returns the letter grade they should receive.

```
def compute_letter_grades(score):
    """
    compute_letter_grades(10)
    >>> "F"
    compute_letter_grades(99)
    >>> "A"
    """
    if Score <= 69:                    Score > 69
        return "F"
    elif Score <= 79:                  Score > 79
        return "C"
    elif Score <= 89:                  Score > 69
        return "B"
    else:
        return "A"
```

Question 3

Question 3. Skeleton code for the function count_evens is below. The function takes in an array of numbers and returns the number of even numbers in the array.

a. If a number n is odd, what will n%2 return?

an odd number

1 because we cannot divide cleanly by 2?

b. Use a combination of iteration and conditionals to complete the function below.

Hint: the % operator returns the remainder if you divide by a certain number! Example: 11 % 5 = 1

```
def count_evens(n_array):
    num_evens = ∅                                   make_array()

    for i in n_array:                               i → another num
        if i % 2 == ∅:
            num_evens += 1                          num_evens = np.append(num_evens, i)

    return num_evens                                len(num_evens)
```

Question 3

c. Use array operations to complete the function below. array([1, 2, 3, 4]) % 2

array([1, 0, 1, 0])

```
def count_evens(n_array):
    remainder_array = n_array % 2 == 0
    return sum(remainder_array)
```

[False, True, False, True]

np.count_nonzero

Question 4

Question 4. Complete the function `separate_numbers`, which takes in an array of numbers and a boolean value. It should return the number of even values in the array if the argument `return_even` is True, or the number of odd values in the array if `return_even` is False.

Hint: Use the `count_evens` function you defined above!

Already True False

```
def separate_numbers(n_array, return_even):
    num_evens = count_evens(n_array)
    if return_even:
        return num_evens
    else:
        return len(n_array) - num_evens
```

End of Section
How did I do?

https://tinyurl.com/kevind8feedback