Data Types & Tables

In lecture, you have been introduced to various data *types* in Python such as integers, strings, and arrays. These data types are particularly important for manipulating and extracting useful information out of data, an important skill for data science. In this section, we’ll be analyzing some of the behavior that Python displays when dealing with particular data types.

# ”Fun” with Arrays

Suppose we have executed the following lines of code. Answer each question with the appropriate output associated with each line of code, or write ERROR if you think the operation is not possible.

odd\_array = make\_array(1, 3, 5, 7)

even\_array = np.arange(2, 10, 2)

an\_array = make\_array(‘1’, ‘2’, ‘3’, ‘4’)

1. odd\_array + even\_array
2. odd\_array + an\_array
3. even\_array.item(3) \* odd\_array.item(1)
4. odd\_arry \* 3
5. (odd\_array + 1) == even\_array
6. an array.item(3) + ‘abcd’

In this next section, we will practice working with tables. In particular, we’ll be focusing on table methods and what data types they return. This will help in understanding how to effectively manipulate tables. Remember to make use of the Python Reference guide when working through these questions – A similar guide will be provided on exams.

# Drought

Your friend Sarah is interested in the level of drought experienced in California over time. She obtained data on the percentage of the population experiencing each level of drought (D0 being the lowest severity, D4 being the highest), as well as the DSCI (Drought Severity and Coverage Index) score (higher = more prevalence and severity of drought). The [table](https://droughtmonitor.unl.edu/DmData/DataTables.aspx)[[1]](#footnote-1) below is called drought and was obtained from the National Drought Mitigation Center at the University of Nebraska-Lincoln.

A table with numbers and a number of numbers

Description automatically generated with medium confidence

Unfortunately, the code Sarah wrote to analyze the data has some bugs. Below are some error messages that appeared, along with what Sarah was trying to calculate; describe the bugs and how you would fix them.

1. A close up of a yellow line

   Description automatically generatedThe proportion of weeks with less than 20% of people experiencing D0 drought
2. A yellow and black text

   Description automatically generatedThe difference each week of the percent of people in D1 drought and the percent in D4 drought
3. The week with the lowest DSCI score out of weeks with more than 10% experiencing D4 drought  
   A screen shot of a computer

   Description automatically generated

# Helpdesk Forum Statistics

The table staff contains information about the staff’s statistics. There are 5 columns:

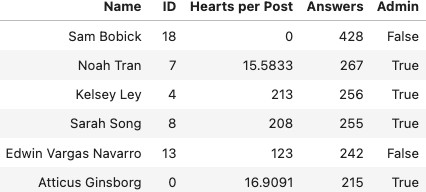
• Name: string, name of the staff member

• ID: int, the ID of the staff member

• Hearts per Post: float, the amount of hearts/likes awarded per post made by a staff member (0 is also given to those with 0 posts)

• Answers: int, number of answers to the Helpdesk questions and comments

• Admin: boolean, whether or not the staff member was an admin

Some rows are shown below:  


1. For each of the columns in staff, identify if the data contained in that column is numerical or categorical.
2. Professor Chen wants to award a very lucky staff member a prize based on their Helpdesk Forum performance; however, he has a few things he’s looking for in them: The staff member have more than 0 hearts per post, they must be an admin, and then out of all the staff members left, they must have the highest number of answers. Fill in the code blanks to find out who gets a prize.

good staff = staff.where( ).where( )

lucky staff name = .sort( ).column( ).item(0)

# Attendance Form

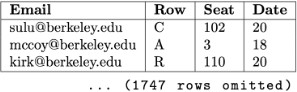
A table named seat contains a row for each time a student submitted the attendance form in lecture on September 18th, 20th, or 22nd. The table contains four columns.

• Email: a string, the email address of the student

• Row: a string, the letter of the row in which they claim to be seated

• Seat: an int, the number of the seat in which they claim to be seated

• Date: an int, the date of the submission, either 18, 20, or 22.



**Fill in the blanks of the Python expressions to compute the described values.** You must use *all* and *only* the lines provided. The last (or only) line of each answer should evaluate to the value described.

1. The largest seat number in the seat table.

Method 1:

max( )

Method 2:

.sort( , ).(

1. The total number of attendance submissions for September 20th in rows A, B, C, D, or E.

Hint: You can use Table.where predicates to compare letters lexicographically (e.g. A is below B), or use a different Table.where predicate

u = seat. ( , )

u. ( , ).

1. <https://droughtmonitor.unl.edu/DmData/DataTables.aspx> filtered with Statistical Population Percent and data type USDM. [↑](#footnote-ref-1)