Exemplar_ Arrays and vectors with NumPy

December 7, 2023

Exemplar: Vectors and arrays with NumPy

0.1 Introduction

Your work as a data professional for the U.S. Environmental Protection Agency (EPA) requires you to analyze air quality index data collected from the United States and Mexico.

The air quality index (AQI) is a number that runs from 0 to 500. The higher the AQI value, the greater the level of air pollution and the greater the health concern. For example, an AQI value of 50 or below represents good air quality, while an AQI value over 300 represents hazardous air quality. Refer to this guide from AirNow.gov for more information.

In this lab, you will work with NumPy arrays to perform calculations and evaluations with data they contain. Specifically, you'll be working with just the data from the numerical AQI readings.

0.2 Tips for completing this lab

As you navigate this lab, keep the following tips in mind:

- `### YOUR CODE HERE ###` indicates where you should write code. Be sure to replace this with your own code before running the code cell.
- Feel free to open the hints for additional guidance as you work on each task.
- To enter your answer to a question, double-click the markdown cell to edit. Be sure to replace the "[Double-click to enter your responses here.]" with your own answer.
- You can save your work manually by clicking File and then Save in the menu bar at the top of the notebook.
- You can download your work locally by clicking File and then Download and then specifying your preferred file format in the menu bar at the top of the notebook.

1 Task 1: Create an array using NumPy

The EPA has compiled some AQI data where each AQI report has the state name, county name, and AQI. Refer to the table below as an example.

state_name	county_name	aqi
Arizona	Maricopa	18
California	Alameda	11

state_name	county_name	aqi
California	Butte	6
Texas	El Paso	40
Florida	Duval	15

1.1 1a: Import NumPy

Import NumPy using its standard alias.

[1]:

```
### YOUR CODE HERE ###
import numpy as np
```

Hint 1

You can refer to what you've learned about importing packages.

Hint 2

Begin with the `import` statement.

Hint 3

The conventional alias for NumPy is `np`.

1.2 1b: Create an array of AQI data

You are given an ordered `list` of AQI readings called `aqi_list`.

1. Use a NumPy function to convert the list to an `ndarray`. Assign the result to a variable called `aqi_array`.
2. Print the length of `aqi_array`.
3. Print the first five elements of `aqi_array`.

Expected result:

```
[OUT] 1725
      [18.  9. 20. 11.  6.]
```

[2]:

```
### RUN THIS CELL TO IMPORT YOUR DATA
import ada_c2_labs as lab
aqi_list = lab.fetch_epa('aqi')
```

[3]:

```
# 1. ### YOUR CODE HERE
aqi_array = np.array(aqi_list)

# 2. ### YOUR CODE HERE
print(len(aqi_array))

# 3. ### YOUR CODE HERE
print(aqi_array[:5])
```

```
1725
[18.  9. 20. 11.  6.]
```

Hint 1

You can refer to what you've learned about creating and slicing arrays.

Hint 2

- Use the NumPy library you imported earlier. NumPy functions must begin with the alias—`np`—that was chosen when you imported the library.
- Built-in Python functions don't need to be prefaced with an alias.
- Use bracket notation for slicing.

Hint 3

- To cast `aqi_list` as an `ndarray`, pass the list as an argument to the `np.array()` NumPy function.
- Use the `len()` function to calculate the length of the array. This function is versatile and works on most iterable data structures in Python.

2 Task 2: Calculate summary statistics

Now that you have the AQI data stored in an array, use NumPy functions to calculate some summary statistics about it.

- Use built-in NumPy functions to print the following values from `aqi_array`:
  1. Maximum value
  2. Minimum value
  3. Median value
  4. Standard deviation

Expected result:

```
[OUT] Max = 93.0
      Min = 0.0
      Median = 8.0
      Std = 10.382982538847708
```

[4]:

```
### YOUR CODE HERE ###
print('Max =', np.max(aqi_array))
print('Min =', np.min(aqi_array))
print('Median =', np.median(aqi_array))
print('Std =', np.std(aqi_array))
```

```
Max = 93.0
Min = 0.0
Median = 8.0
Std = 10.382982538847708
```

Hint 1

Refer to what you've learned about built-in NumPy functions.

Hint 2

- Remember, to use a function from the NumPy library, it must begin with the alias you used for NumPy when you imported it.
- The function names for these operations are the same as they are for the same built-in Python functions, and they work the same way for 1-dimensional arrays.

Hint 3

Use `np.max()`, `np.min()`, `np.median()`, `np.std()` to calculate these statistics.

3 Task 3: Calculate percentage of readings with cleanest AQI

You are interested in how many air quality readings in the data represent the cleanest air, which we'll consider **readings of 5 or less.**

To perform this calculation, you'll make use of one of the properties of arrays that make them so powerful: their element-wise operability. For example, when you add an integer to an `ndarray` using the `+` operator, it performs an element-wise addition on the whole array.

```
[IN]  my_array = np.array([1, 2, 3])
      my_array = my_array + 10
      print(my_array)

[OUT] [11, 12, 13]
```

The same concept applies to comparison operators used on an `ndarray`. With this in mind:

- Calculate the percentage of AQI readings that are considered cleanest:
  1. Use a comparison statement to get an array of Boolean values that is the same length as `aqi_array`. Assign the result to variable called `boolean_aqi`.
  2. Calculate the number of `True` values in the `boolean_aqi` and divide this number by the total number of values in the array. Assign the result to a variable named `percent_under_6` and print it.

Expected result:

```
[OUT] 0.3194202898550725
```

[5]:

```
# 1. ### YOUR CODE HERE ###
boolean_aqi = (aqi_array <= 5)

# 2. ### YOUR CODE HERE ###
percent_under_6 = boolean_aqi.sum() / len(boolean_aqi)
percent_under_6
```

```
[5]: 0.3194202898550725
```

Hint 1

1. To create `boolean_aqi`, apply the appropriate comparison expression to `aqi_array`.

Hint 2

To calculate `percent_under_6`, consider the fact that in Python 3, `True` values always equate with `1`, and `False` values always equate with `0`.

Hint 3

- Because `True` values always equate with `1` and `False` values always equate with `0`, you can sum a Boolean sequence to find the number of `True` values.
- To calculate the length of the full array, remember that the `len()` function works with most iterable objects in Python, including NumPy arrays.

4 Conclusion

What are your key takeaways from this lab?

- Python packages contain functions to perform specific tasks.
  - The NumPy package has functions used for working with arrays and performing mathematical operations
- Arrays are similar to lists, but only store one type of data per array.
  - Processing data stored in an array is much quicker than processing data stored in traditional lists.
- Arrays are useful for performing element-wise operations, including arithmetic and comparisons.

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.