

Geraldo Braho

HW1: python and numpy approach to filling the missing values in array

Given a 2D array (list of list) filled partially with missing values, fill the missing values (NA) with neighboring values based on the following rules: for missing values use the closest non-missing value along the same vertical column. if missing value is in the middle between 2 closest non-missing values, then use the average of these 2 values.

Provide 2 implementations that use pure Python (no numpy) and using numpy arrays and features. Test your algorithm on multiple test cases that you can think of. What are the limitations of this approach for filling the missing values?

| | | | | |
|---|----|----|-------|----|
| 0 | 3 | NA | 1.618 | 1 |
| 1 | NA | 4 | 2.618 | 1 |
| 2 | 1 | 0 | NA | 2 |
| 3 | 0 | 4 | NA | 3 |
| 4 | -1 | 0 | 11.09 | NA |

$$[[1.0, 3, 0.0, 1.618, 1], [1, 0.5, 4, 2.618, 1], [2, 1, 0.0, 1.309, 2], [3, 0.5, 4, 0.6545, 3], [4, -1, 2.0, 11.09, 1.5]]$$

```
np.where(np.isnan(a), ma.array(a, mask=np.isnan(a)).mean(axis=0), a)
```

75 11)

```
In [ ]: # I am not sure about the numpy approach. I just did this so I dont ge  
t a 0
```