

numpy-lecture-3-dec-batch

May 27, 2023

0.1 Numpy Lecture - 3

```
[3]: #Quiz 1
import numpy as np
a = np.array([0,1,2,3,4,5])
mask = (a%2 == 0)
a[mask] = -1
a
```

```
[3]: array([-1,  1, -1,  3, -1,  5])
```

```
[5]: arr = np.arange(12)

arr = arr.reshape(-1,4)
arr.shape
```

```
[5]: (3, 4)
```

```
[6]: a = np.array([[1, 2, 3], [4, 5, 6]])
a.shape
```

```
[6]: (2, 3)
```

```
[7]: a.flatten()
```

```
[7]: array([1, 2, 3, 4, 5, 6])
```

```
[8]: ### Indexing on 2D array
```

```
[9]: m1 = np.arange(1, 10).reshape(3, 3)
m1.ndim
```

```
[9]: 2
```

```
[10]: m1.shape
```

```
[10]: (3, 3)
```

```
[11]: m1
```

```
[11]: array([[1, 2, 3],  
          [4, 5, 6],  
          [7, 8, 9]])
```

```
[12]: m1[1][1]
```

```
[12]: 5
```

```
[13]: m1[1, 2]
```

```
[13]: 6
```

```
[14]: a = np.arange(12)  
      a[[0, 1, 2, 1, 2]]
```

```
[14]: array([0, 1, 2, 1, 2])
```

```
[15]: m1
```

```
[15]: array([[1, 2, 3],  
          [4, 5, 6],  
          [7, 8, 9]])
```

```
[16]: m1[[0, 1, 2], [0, 1, 2]]
```

```
[16]: array([1, 5, 9])
```

```
[18]: a = np.arange(10, 22).reshape(3, 4)  
      a
```

```
[18]: array([[10, 11, 12, 13],  
          [14, 15, 16, 17],  
          [18, 19, 20, 21]])
```

```
[19]: a[[0, 1], [2, 3]]
```

```
[19]: array([12, 17])
```

1 Slicing on 2D array

```
[20]: a = np.arange(1,13).reshape(3, 4)  
      a
```

```
[20]: array([[ 1,  2,  3,  4],
           [ 5,  6,  7,  8],
           [ 9, 10, 11, 12]])
```

```
[21]: a[:2]
```

```
[21]: array([[1, 2, 3, 4],
           [5, 6, 7, 8]])
```

```
[22]: a[:, :2]
```

```
[22]: array([[ 1,  2],
           [ 5,  6],
           [ 9, 10]])
```

```
[23]: a[1:, 1:3]
```

```
[23]: array([[ 6,  7],
           [10, 11]])
```

```
[24]: a[:, 1::2]
```

```
[24]: array([[ 2,  4],
           [ 6,  8],
           [10, 12]])
```

```
[26]: a[:, [1, 3]]
```

```
[26]: array([[ 2,  4],
           [ 6,  8],
           [10, 12]])
```

```
[30]: m1 = np.array([[0, 1, 2, 3], [4, 5, 6, 7],
                    [8, 9, 10, 11]])

m1[[0, 2], :2]
```

```
[30]: array([[0, 1],
           [8, 9]])
```

1.0.1 Fancy Indexing [Masking]

```
[31]: a = np.arange(12).reshape(3, 4)
a < 6
```

```
[31]: array([[ True,  True,  True,  True],
           [ True,  True, False, False],
```

```
[False, False, False, False]])
```

```
[32]: a[a<6]
```

```
[32]: array([0, 1, 2, 3, 4, 5])
```

1.1 2D - Axis

```
[33]: a = np.arange(1, 13).reshape(3, 4)
      np.min(a)
```

```
[33]: 1
```

```
[34]: np.min(a, axis=1)
```

```
[34]: array([1, 5, 9])
```

```
[35]: np.max(a, axis=0)
```

```
[35]: array([ 9, 10, 11, 12])
```

2 Fitbit use case

```
[36]: !lgdown 1vk1Pu0djiYcrdc85yUXZ_Rqq2oZNcohd
```

Downloading...

From: https://drive.google.com/uc?id=1vk1Pu0djiYcrdc85yUXZ_Rqq2oZNcohd

To: /Users/satish/Desktop/scaler/Dec Tue Batch - DAV-1/fit.txt

100% | 3.43k/3.43k [00:00<00:00, 4.10MB/s]

```
[38]: data = np.loadtxt('fit.txt', dtype='str')
```

```
[39]: data.ndim
```

```
[39]: 2
```

```
[40]: data.shape
```

```
[40]: (96, 6)
```

```
[41]: data[0]
```

```
[41]: array(['06-10-2017', '5464', 'Neutral', '181', '5', 'Inactive'],
      dtype='<U10')
```

```
[42]: data.T[0]
```

```
[42]: array(['06-10-2017', '07-10-2017', '08-10-2017', '09-10-2017',
          '10-10-2017', '11-10-2017', '12-10-2017', '13-10-2017',
          '14-10-2017', '15-10-2017', '16-10-2017', '17-10-2017',
          '18-10-2017', '19-10-2017', '20-10-2017', '21-10-2017',
          '22-10-2017', '23-10-2017', '24-10-2017', '25-10-2017',
          '26-10-2017', '27-10-2017', '28-10-2017', '29-10-2017',
          '30-10-2017', '31-10-2017', '01-11-2017', '02-11-2017',
          '03-11-2017', '04-11-2017', '05-11-2017', '06-11-2017',
          '07-11-2017', '08-11-2017', '09-11-2017', '10-11-2017',
          '11-11-2017', '12-11-2017', '13-11-2017', '14-11-2017',
          '15-11-2017', '16-11-2017', '17-11-2017', '18-11-2017',
          '19-11-2017', '20-11-2017', '21-11-2017', '22-11-2017',
          '23-11-2017', '24-11-2017', '25-11-2017', '26-11-2017',
          '27-11-2017', '28-11-2017', '29-11-2017', '30-11-2017',
          '01-12-2017', '02-12-2017', '03-12-2017', '04-12-2017',
          '05-12-2017', '06-12-2017', '07-12-2017', '08-12-2017',
          '09-12-2017', '10-12-2017', '11-12-2017', '12-12-2017',
          '13-12-2017', '14-12-2017', '15-12-2017', '16-12-2017',
          '17-12-2017', '18-12-2017', '19-12-2017', '20-12-2017',
          '21-12-2017', '22-12-2017', '23-12-2017', '24-12-2017',
          '25-12-2017', '26-12-2017', '27-12-2017', '28-12-2017',
          '29-12-2017', '30-12-2017', '31-12-2017', '01-01-2018',
          '02-01-2018', '03-01-2018', '04-01-2018', '05-01-2018',
          '06-01-2018', '07-01-2018', '08-01-2018', '09-01-2018'],
          dtype='<U10')
```

```
[43]: date, step_count, mood, calories_burned, hours_of_sleep, activity_status = data.
      ↪T
```

```
[44]: date
```

```
[44]: array(['06-10-2017', '07-10-2017', '08-10-2017', '09-10-2017',
          '10-10-2017', '11-10-2017', '12-10-2017', '13-10-2017',
          '14-10-2017', '15-10-2017', '16-10-2017', '17-10-2017',
          '18-10-2017', '19-10-2017', '20-10-2017', '21-10-2017',
          '22-10-2017', '23-10-2017', '24-10-2017', '25-10-2017',
          '26-10-2017', '27-10-2017', '28-10-2017', '29-10-2017',
          '30-10-2017', '31-10-2017', '01-11-2017', '02-11-2017',
          '03-11-2017', '04-11-2017', '05-11-2017', '06-11-2017',
          '07-11-2017', '08-11-2017', '09-11-2017', '10-11-2017',
          '11-11-2017', '12-11-2017', '13-11-2017', '14-11-2017',
          '15-11-2017', '16-11-2017', '17-11-2017', '18-11-2017',
          '19-11-2017', '20-11-2017', '21-11-2017', '22-11-2017',
          '23-11-2017', '24-11-2017', '25-11-2017', '26-11-2017',
          '27-11-2017', '28-11-2017', '29-11-2017', '30-11-2017',
          '01-12-2017', '02-12-2017', '03-12-2017', '04-12-2017',
          '05-12-2017', '06-12-2017', '07-12-2017', '08-12-2017',
```

```

'09-12-2017', '10-12-2017', '11-12-2017', '12-12-2017',
'13-12-2017', '14-12-2017', '15-12-2017', '16-12-2017',
'17-12-2017', '18-12-2017', '19-12-2017', '20-12-2017',
'21-12-2017', '22-12-2017', '23-12-2017', '24-12-2017',
'25-12-2017', '26-12-2017', '27-12-2017', '28-12-2017',
'29-12-2017', '30-12-2017', '31-12-2017', '01-01-2018',
'02-01-2018', '03-01-2018', '04-01-2018', '05-01-2018',
'06-01-2018', '07-01-2018', '08-01-2018', '09-01-2018'],
dtype='<U10')

```

```

[45]: step_count = np.array(step_count, dtype='int')
      step_count

```

```

[45]: array([5464, 6041,    25, 5461, 6915, 4545, 4340, 1230,    61, 1258, 3148,
          4687, 4732, 3519, 1580, 2822,   181, 3158, 4383, 3881, 4037,   202,
           292,   330, 2209, 4550, 4435, 4779, 1831, 2255,   539, 5464, 6041,
          4068, 4683, 4033, 6314,   614, 3149, 4005, 4880, 4136,   705,   570,
           269, 4275, 5999, 4421, 6930, 5195,   546,   493,   995, 1163, 6676,
          3608,   774, 1421, 4064, 2725, 5934, 1867, 3721, 2374, 2909, 1648,
           799, 7102, 3941, 7422,   437, 1231, 1696, 4921,   221, 6500, 3575,
          4061,   651,   753,   518, 5537, 4108, 5376, 3066,   177,    36,   299,
          1447, 2599,   702,   133,   153,   500, 2127, 2203])

```

```

[46]: calories_burned = np.array(calories_burned, dtype='int')
      hours_of_sleep = np.array(hours_of_sleep, dtype='int')

```

```

[47]: np.unique(mood)

```

```

[47]: array(['Happy', 'Neutral', 'Sad'], dtype='<U10')

```

```

[48]: np.unique(activity_status)

```

```

[48]: array(['Active', 'Inactive'], dtype='<U10')

```

```

[49]: #1. What is the average step count?

```

```

[50]: step_count.mean()

```

```

[50]: 2935.9375

```

```

[51]: #2 On which day the step count was highest?

```

```

[52]: a = np.array([10, 5, 20, 25, 30, 55])
      np.argmax(a)

```

```

[52]: 5

```

```
[53]: np.argmin(a)
```

```
[53]: 1
```

```
[54]: step_count.argmax()
```

```
[54]: 69
```

```
[55]: date[step_count.argmax()]
```

```
[55]: '14-12-2017'
```

```
[56]: calories_burned[step_count.argmax()]
```

```
[56]: 243
```

```
[57]: np.mean(step_count[mood=='Sad'])
```

```
[57]: 2103.0689655172414
```

```
[58]: np.mean(step_count[mood=='Happy'])
```

```
[58]: 3392.725
```

```
[59]: #Find mood, when step count is high/ low
```

```
[60]: np.unique(mood[step_count>4000], return_counts=True)
```

```
[60]: (array(['Happy', 'Neutral', 'Sad'], dtype='<U10'), array([22, 9, 7]))
```

```
[61]: np.unique(mood[step_count<2000], return_counts=True)
```

```
[61]: (array(['Happy', 'Neutral', 'Sad'], dtype='<U10'), array([13, 8, 18]))
```

```
[ ]:
```