

```
In [2]: import numpy as np
```

```
In [5]: a = [1,2,3]
a = [i**2 for i in a]
print(a)
```

```
[1, 4, 9]
```

```
In [4]: b = np.array([1,2,3])
```

```
In [6]: print(b**2)
```

```
[1 4 9]
```

```
In [7]: l = range(1000000)
```

```
In [8]: %timeit [i**2 for i in l]
```

```
295 ms ± 19 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
In [9]: l = np.arange(1000000)
```

```
In [11]: %timeit l**2
```

```
1.96 ms ± 188 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

```
In [12]: arr1 = np.array([1, 2, 3])
print(arr1)
print(arr1 * 2)
```

```
[1 2 3]
[2 4 6]
```

```
In [13]: arr1
```

```
Out[13]: array([1, 2, 3])
```

```
In [14]: arr1.ndim
```

```
Out[14]: 1
```

```
In [15]: arr1.shape
```

```
Out[15]: (3,)
```

```
In [18]: np.arange(1,9,2)
```

```
Out[18]: array([1, 3, 5, 7])
```

```
In [19]: np.linspace(0,10,11)
```

```
Out[19]: array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
In [22]: np.linspace(0,100,25)
```

```
Out[22]: array([ 0.          ,  4.16666667,  8.33333333, 12.5         ,
 16.66666667, 20.83333333, 25.          , 29.16666667,
 33.33333333, 37.5          , 41.66666667, 45.83333333,
 50.          , 54.16666667, 58.33333333, 62.5         ,
 66.66666667, 70.83333333, 75.          , 79.16666667,
 83.33333333, 87.5          , 91.66666667, 95.83333333,
 100.         ])
```

```
In [21]: type(arr1)
```

```
Out[21]: numpy.ndarray
```

```
In [25]: arr4 = np.array([1,2,3,4,"a"])
```

```
In [26]: arr4
```

```
Out[26]: array(['1', '2', '3', '4', 'a'], dtype='<U11')
```

```
In [29]: a5 = np.array([1,2,3,4], dtype="float")
```

```
In [30]: a5
```

```
Out[30]: array([1., 2., 3., 4.])
```

```
In [31]: m1 = np.array([[1,2,3],[4,5,6]])
m1
```

```
Out[31]: array([[1, 2, 3],
 [4, 5, 6]])
```

```
In [32]: m1.shape
```

```
Out[32]: (2, 3)
```

```
In [33]: m1.ndim
```

```
Out[33]: 2
```

```
In [42]: m2 = np.arange(1,13)  
m2.shape
```

```
Out[42]: (12,)
```

```
In [43]: m2.reshape(1,12)
```

```
Out[43]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]])
```

```
In [36]: m2.reshape(4,3)
```

```
Out[36]: array([[ 1,  2,  3],  
                 [ 4,  5,  6],  
                 [ 7,  8,  9],  
                 [10, 11, 12]])
```

```
In [37]: m2.reshape(4,4)
```

```
-----  
ValueError                                     Traceback (most recent call last)  
<ipython-input-37-b0f1cc8c31b9> in <module>  
----> 1 m2.reshape(4,4)
```

```
ValueError: cannot reshape array of size 12 into shape (4,4)
```

```
In [38]: m2
```

```
Out[38]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

```
In [39]: m2.reshape(12,1)
```

```
Out[39]: array([[ 1],  
                 [ 2],  
                 [ 3],  
                 [ 4],  
                 [ 5],  
                 [ 6],  
                 [ 7],  
                 [ 8],  
                 [ 9],  
                 [10],  
                 [11],  
                 [12]])
```

```
In [40]: m2.resize(4,3)
```

```
In [41]: m2
```

```
Out[41]: array([[ 1,  2,  3],  
                 [ 4,  5,  6],  
                 [ 7,  8,  9],  
                 [10, 11, 12]])
```

```
In [44]: a = np.arange(3)
```

```
In [45]: a.shape
```

```
Out[45]: (3,)
```

```
In [46]: a.T
```

```
Out[46]: array([0, 1, 2])
```

```
In [47]: A = np.arange(12).reshape(3, 4)  
A
```

```
Out[47]: array([[ 0,  1,  2,  3],  
                 [ 4,  5,  6,  7],  
                 [ 8,  9, 10, 11]])
```

```
In [48]: A.flatten()
```

```
Out[48]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
In [49]: A
```

```
Out[49]: array([[ 0,  1,  2,  3],  
                 [ 4,  5,  6,  7],  
                 [ 8,  9, 10, 11]])
```

```
In [50]: A.reshape(1,12)
```

```
Out[50]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]])
```

```
In [52]: A.reshape(6,-1)
```

```
Out[52]: array([[ 0,  1],  
                 [ 2,  3],  
                 [ 4,  5],  
                 [ 6,  7],  
                 [ 8,  9],  
                 [10, 11]])
```

```
In [57]: A.reshape(1,-1)
```

```
Out[57]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]])
```

```
In [ ]:  
2*6  
1*12  
3*4
```

```
In [58]: np.zeros(3)
```

```
Out[58]: array([0., 0., 0.])
```

```
In [60]: np.zeros((2,3))
```

```
Out[60]: array([[0., 0., 0.],  
[0., 0., 0.]])
```

```
In [61]: np.ones((2,3))
```

```
Out[61]: array([[1., 1., 1.],  
[1., 1., 1.]])
```

```
In [62]: np.ones((2,3))*5
```

```
Out[62]: array([[5., 5., 5.],  
[5., 5., 5.]])
```

```
In [63]: np.diag([1,2,3])
```

```
Out[63]: array([[1, 0, 0],  
[0, 2, 0],  
[0, 0, 3]])
```

```
In [64]: np.identity(3)
```

```
Out[64]: array([[1., 0., 0.],  
[0., 1., 0.],  
[0., 0., 1.]])
```

```
In [ ]: # Break : 10 : 30 pm
```

```
In [65]: a = np.arange(10)
```

```
In [66]: a.shape
```

```
Out[66]: (10,)
```

```
In [67]: m1 = np.arange(12)  
m1
```

```
Out[67]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

```
In [68]: m1[0]
```

```
Out[68]: 0
```

In [69]: `m1[12]`

```
-----  
IndexError                                                 Traceback (most recent call last)  
<ipython-input-69-0abd94d7097d> in <module>  
----> 1 m1[12]  
  
IndexError: index 12 is out of bounds for axis 0 with size 12
```

In [70]: `m1 = np.arange(1,10).reshape((3,3))`In [71]: `m1`Out[71]: `array([[1, 2, 3],
 [4, 5, 6],
 [7, 8, 9]])`In [72]: `m1[1][2]`Out[72]: `6`In [73]: `m1[1,2]`Out[73]: `6`In [74]: `m1 = np.array([100,200,300,400,500,600])`In [75]: `m1[[2,3,4,1,2,2]]`Out[75]: `array([300, 400, 500, 200, 300, 300])`In [76]: `m1 = np.arange(12)`
`m1`Out[76]: `array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])`In [77]: `m1 = np.arange(9).reshape((3,3))`In [78]: `m1`Out[78]: `array([[0, 1, 2],
 [3, 4, 5],
 [6, 7, 8]])`In [79]: `m1[1:,:1:]`Out[79]: `array([[4, 5],
 [7, 8]])`

```
In [80]: m1[1:][1:]
```

```
Out[80]: array([[6, 7, 8]])
```

```
In [85]: m1[1:]
```

```
Out[85]: array([[3, 4, 5],  
                 [6, 7, 8]])
```

```
In [86]: m1 = np.arange(12).reshape(3, 4)
```

```
In [87]: m1
```

```
Out[87]: array([[ 0,  1,  2,  3],  
                 [ 4,  5,  6,  7],  
                 [ 8,  9, 10, 11]])
```

```
In [88]: m1 < 6
```

```
Out[88]: array([[ True,  True,  True,  True],  
                 [ True,  True, False, False],  
                 [False, False, False, False]])
```

```
In [89]: m1[m1<6]
```

```
Out[89]: array([0, 1, 2, 3, 4, 5])
```

```
In [90]: a = np.array([1,2,3,4])  
b = np.array([5,6,7,8])  
a+b
```

```
Out[90]: array([ 6,  8, 10, 12])
```

```
In [91]: np.add(a,b)
```

```
Out[91]: array([ 6,  8, 10, 12])
```

```
In [92]: np.sum(a)
```

```
Out[92]: 10
```

```
In [ ]: axis = 0 -> vertical  
axis = 1 -> horizontal
```

```
In [93]: a = np.arange(12).reshape(3, 4)
```

In [94]: a

Out[94]: array([[0, 1, 2, 3],
 [4, 5, 6, 7],
 [8, 9, 10, 11]])

In [95]: np.sum(a)

Out[95]: 66

In [96]: np.sum(a, axis=1)

Out[96]: array([6, 22, 38])

In [97]: np.sum(a, axis=0)

Out[97]: array([12, 15, 18, 21])

In [98]: np.any([True, True, False])

Out[98]: True

In [101...]: a = np.array([0,0,1]) # atleast 1 element is non-zero
np.any(a)

Out[101...]: True

In [102...]: np.all(a)

Out[102...]: False

In [103...]: a = np.array([2,30,41,7,17,52])
a

Out[103...]: array([2, 30, 41, 7, 17, 52])

In [104...]: np.sort(a)

Out[104...]: array([2, 7, 17, 30, 41, 52])

In [105...]: np.argsort(a)

Out[105...]: array([0, 3, 4, 1, 2, 5], dtype=int64)

In [106...]: data = np.loadtxt('fitness.txt', dtype='str')

In [107...]: data[0:5]

```
Out[107...]: array([['06-10-2017', '5464', '200', '181', '5', '0', '66'],
       ['07-10-2017', '6041', '100', '197', '8', '0', '66'],
       ['08-10-2017', '25', '100', '0', '5', '0', '66'],
       ['09-10-2017', '5461', '100', '174', '4', '0', '66'],
       ['10-10-2017', '6915', '200', '223', '5', '500', '66']],  
      dtype='<U10')
```

In [108...]: data.shape

Out[108...]: (96, 7)

In [109...]: data[0]

Out[109...]: array(['06-10-2017', '5464', '200', '181', '5', '0', '66'], dtype='<U10')

In [110...]: data.T

```
Out[110...]: 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'],
      ['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'],
      ['200', '100', '100', '100', '200', '100', '100', '100',
       '100', '100', '100', '300', '100', '100', '100', '100',
       '200', '200', '200', '200', '300', '200', '300', '300',
       '300', '300', '300', '300', '200', '300', '300', '300']])
```

Numpy1

In [111...]

```
data[0]
```

```
Out[111]: array(['06-10-2017', '5464', '200', '181', '5', '0', '66'], dtype='|<U10')
```

In [112...]

```
data.T[0]
```

```
Out[112... 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' )
```

In [113...]

```
date = data.T[0]
```

In [114...]

```
date, step_count, mood, calories_burned, hours_of_sleep, activity_status, weight = data
```

In [115...]

```
date
```

```
Out[115... 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' )
```

In [116...]

```
mood
```

```
Out[116... array(['200', '100', '100', '100', '200', '100', '100', '100',
'100', '100', '100', '300', '100', '100', '100', '200',
'200', '200', '200', '200', '300', '200', '200', '300',
'300', '300', '300', '300', '200', '200', '300', '300',
'300', '300', '300', '300', '300', '300', '200', '300',
'300', '300', '300', '300', '300', '300', '300', '200',
'100', '300', '300', '300', '300', '300', '300', '300',
'100', '300', '300', '300', '300', '300', '300', '300',
'200', '200', '100', '100', '200', '200', '300', '200',
'100', '200', '100', '200', '200', '100', '100', '100',
'300', '200', '300', '200', '100', '100', '100', '200',
'100', '100', '300', '200', '200', '300'], dtype='<U10' )
```

```
In [117]: step_count
```

```
Out[117]: 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'], dtype='<U10')
```

```
In [118]: step_count.dtype
```

```
Out[118]: dtype('<U10')
```

```
In [119]: step_count = np.array(step_count, dtype = 'int')
step_count.dtype
```

```
Out[119]: dtype('int32')
```

```
In [120]: step_count
```

```
Out[120]: 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])
```

```
In [121]: calories_burned = np.array(calories_burned, dtype = 'int')
calories_burned.dtype
```

```
Out[121]: dtype('int32')
```

```
In [122]: hours_of_sleep = np.array(hours_of_sleep, dtype = 'int')
hours_of_sleep.dtype
```

```
Out[122]: dtype('int32')
```

```
In [123]: weight = np.array(weight, dtype = 'int')
weight.dtype
```

```
Out[123]: dtype('int32')
```

```
In [124]:
```

mood

```
Out[124... array(['200', '100', '100', '100', '200', '100', '100', '100', '100',
       '100', '100', '100', '300', '100', '100', '100', '200',
       '200', '200', '200', '200', '200', '300', '200', '200', '300',
       '300', '300', '300', '300', '300', '300', '300', '300',
       '300', '300', '300', '300', '300', '300', '300', '300',
       '300', '300', '300', '300', '300', '300', '300', '300',
       '100', '300', '300', '300', '300', '300', '300', '300',
       '200', '200', '100', '100', '200', '200', '300', '200',
       '100', '200', '100', '200', '200', '100', '100', '100',
       '300', '200', '300', '200', '100', '100', '200', '200',
       '100', '100', '300', '200', '200', '300'], dtype='<U10')
```

```
In [125... mood[mood == '300'] = 'Happy'
```

mood

```
Out[126... array(['200', '100', '100', '100', '200', '100', '100', '100', '100',
       '100', '100', '100', 'Happy', '100', '100', '100', '200',
       '200', '200', '200', '200', '200', 'Happy', '200', 'Happy',
       'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy'],
      dtype='<U10')
```

```
In [127... mood[mood == '200'] = 'Neutral'
mood[mood == '100'] = 'Sad'
```

mood

```
Out[128... array(['Neutral', 'Sad', 'Sad', 'Sad', 'Neutral', 'Sad', 'Sad', 'Sad',
       'Sad', 'Sad', 'Sad', 'Sad', 'Happy', 'Sad', 'Sad', 'Sad',
       'Neutral', 'Neutral', 'Neutral', 'Neutral', 'Neutral', 'Neutral',
       'Happy', 'Neutral', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy',
       'Happy', 'Happy', 'Neutral', 'Happy', 'Happy', 'Happy', 'Happy',
       'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy',
       'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy',
       'Happy', 'Happy', 'Neutral', 'Sad', 'Happy', 'Happy', 'Happy',
       'Happy', 'Happy', 'Happy', 'Happy', 'Sad', 'Neutral', 'Neutral',
       'Sad', 'Sad', 'Neutral', 'Neutral', 'Happy', 'Neutral', 'Neutral',
       'Sad', 'Neutral', 'Sad', 'Neutral', 'Neutral', 'Sad', 'Sad', 'Sad',
       'Sad', 'Happy', 'Neutral', 'Happy', 'Neutral', 'Sad', 'Sad', 'Sad',
       'Neutral', 'Neutral', 'Sad', 'Sad', 'Happy', 'Neutral', 'Neutral',
       'Happy'], dtype='<U10')
```

```
In [129... np.unique(mood)
```

```
Out[129... array(['Happy', 'Neutral', 'Sad'], dtype='<U10')
```

activity_status

```
Out[130... array(['0', '0', '0', '0', '500', '0', '0', '0', '0', '0', '0', '0', '0', '0',  
      '500', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0',  
      '500', '0', '0', '0', '0', '0', '500', '0', '0', '0', '0', '0', '0', '0', '0', '500',  
      '500', '500', '500', '500', '500', '500', '500', '500', '500', '500', '500', '0', '0',  
      '0', '0', '0', '0', '500', '500', '500', '500', '500', '500', '500', '500', '500',  
      '500', '500', '500', '500', '500', '500', '500', '0', '500', '500', '500', '500', '0',  
      '500', '500', '500', '500', '500', '500', '500', '0', '500', '500', '500', '500', '500',  
      '0', '0', '0', '0', '500', '500', '500', '500', '500', '500', '500', '0', '0', '0', '0',  
      '0', '0', '0', '0', '500', '0', '500'], dtype='<U10')
```

```
In [131...]: activity_status[activity_status == '500'] = 'Active'  
        activity_status[activity_status == '0'] = 'Inactive'
```

```
In [132...]: activity status
```

```
Out[132... array(['Inactive', 'Inactive', 'Inactive', 'Inactive', 'Active',
   'Inactive', 'Inactive', 'Inactive', 'Inactive', 'Inactive',
   'Inactive', 'Inactive', 'Active', 'Inactive', 'Inactive',
   'Inactive', 'Inactive', 'Inactive', 'Inactive', 'Inactive',
   'Inactive', 'Inactive', 'Inactive', 'Inactive', 'Inactive',
   'Inactive', 'Inactive', 'Inactive', 'Inactive', 'Inactive',
   'Active', 'Inactive', 'Inactive', 'Inactive', 'Inactive', 'Active',
   'Inactive', 'Inactive', 'Inactive', 'Inactive', 'Inactive',
   'Active', 'Active', 'Active', 'Active', 'Active', 'Active',
   'Active', 'Active', 'Active', 'Inactive', 'Inactive', 'Inactive',
   'Inactive', 'Inactive', 'Inactive', 'Active', 'Active', 'Active',
   'Active', 'Active', 'Active', 'Active', 'Active', 'Active',
   'Active', 'Active', 'Active', 'Active', 'Active', 'Active',
   'Active', 'Active', 'Active', 'Active', 'Active', 'Active',
   'Inactive', 'Active', 'Active', 'Active', 'Active', 'Active',
   'Inactive', 'Active', 'Active', 'Active', 'Active', 'Active',
   'Inactive', 'Active', 'Active', 'Active', 'Active', 'Inactive',
   'Inactive', 'Inactive', 'Inactive', 'Active', 'Active', 'Active',
   'Active', 'Inactive', 'Inactive', 'Inactive', 'Inactive',
   'Inactive', 'Inactive', 'Inactive', 'Inactive', 'Active',
   'Inactive', 'Active'], dtype='<U10')
```

In [133...]: step_count.mean()

Out[133... 2935.9375

```
Out[134... 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])
```

In [135...]: step_count.argmax()

Out[135... 69

```
In [136]: date[step].count.argmax()
```

```
Out[136... '14-12-2017'
```

```
In [137... calories_burned[step_count.argmax()]
```

```
Out[137... 243
```

```
In [138... step_count.max()
```

```
Out[138... 7422
```

```
In [139... mood[mood == 'Sad'].shape
```

```
Out[139... (29,)
```

```
In [140... np.unique(mood)
```

```
Out[140... array(['Happy', 'Neutral', 'Sad'], dtype='<U10')
```

```
In [141... np.unique(mood, return_counts = True)
```

```
Out[141... (array(['Happy', 'Neutral', 'Sad'], dtype='<U10'),  
array([40, 27, 29], dtype=int64))
```

```
In [142... np.mean(step_count[mood == 'Sad'])
```

```
Out[142... 2103.0689655172414
```

```
In [143... step_count[mood == 'Sad']
```

```
Out[143... array([6041,    25,  5461,  4545,  4340, 1230,    61, 1258,  3148,  4687,  3519,  
1580, 2822,   181, 6676, 3721, 1648,  799, 1696,   221, 4061,   651,  
753, 518,   177,    36, 299,   702,  133])
```

```
In [144... np.mean(step_count[mood == 'Happy'])
```

```
Out[144... 3392.725
```

```
In [147... np.unique(mood[step_count > 4000], return_counts = True)
```

```
Out[147... (array(['Happy', 'Neutral', 'Sad'], dtype='<U10'),  
array([22,  9,   7], dtype=int64))
```

```
In [148... step_count[mood == 'Happy']
```

```
Out[148... array([4732,  330, 4550, 4435, 4779, 1831, 2255,  539, 5464, 4068, 4683,  
4033, 6314,  614, 3149, 4005, 4880, 4136,  705, 269, 4275, 5999,
```

```
4421, 6930, 5195, 546, 493, 995, 3608, 774, 1421, 4064, 2725,  
5934, 1867, 7422, 5537, 5376, 153, 2203])
```

In [150...]

```
mood[step_count > 4000]
```

```
Out[150... array(['Neutral', 'Sad', 'Sad', 'Neutral', 'Sad', 'Sad', 'Happy',  
'Neutral', 'Neutral', 'Happy', 'Happy', 'Happy', 'Happy',  
'Neutral', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy',  
'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Happy', 'Sad',  
'Happy', 'Happy', 'Neutral', 'Happy', 'Neutral', 'Neutral', 'Sad',  
'Happy', 'Neutral', 'Happy'], dtype='<U10')
```

In [151...]

```
mood == 'Happy'
```

```
Out[151... array([False, False, False, False, False, False, False, False,  
False, False, False, True, False, False, False, False,  
False, False, False, False, True, False, True, True,  
True, True, True, True, False, True, True, True,  
True, True, True, True, True, True, True, False,  
False, True, True, True, True, True, True, True, False,  
False, False, False, False, False, True, False, False,  
False, False, False, False, False, False, False, False,  
True, False, True, False, False, False, False, False, False,  
False, False, True, False, True], dtype='<U10')]
```

In []: