Data Analytics and Machine Learning using Python

Lab Exercise Week-2 Submission Date :- 28-05-2020 (Time -9:00AM)

1. Import the numpy package under the name np

In [1]:

```
import numpy as np
```

2. Create a null vector of size 20

In [3]:

```
np.zeros(20)
```

Out[3]:

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0.,
       0., 0., 0.])
```

3. Create a Ones Vector of size 20

In [4]:

```
np.ones(20)
```

Out[4]:

```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1.,
       1., 1., 1.])
```

4. Create a boolean array of 3X4.

In [6]:

```
np.ones((3, 4), dtype = bool)
```

Out[6]:

```
array([[ True,  True,  True,  True],
       [ True,  True,  True,  True],
       [ True,  True,  True,  True]])
```

5. Create a vector with values ranging from 100 to 200 of float64 data type

In [8]:

```
range_0x01 = np.arange(100, 200, 1.0)
range_0x01
```

Out[8]:

```
array([100., 101., 102., 103., 104., 105., 106., 107., 108., 109., 11
0.,
       111., 112., 113., 114., 115., 116., 117., 118., 119., 120., 12
1.,
       122., 123., 124., 125., 126., 127., 128., 129., 130., 131., 13
2.,
       133., 134., 135., 136., 137., 138., 139., 140., 141., 142., 14
3.,
       144., 145., 146., 147., 148., 149., 150., 151., 152., 153., 15
4.,
       155., 156., 157., 158., 159., 160., 161., 162., 163., 164., 16
5.,
       166., 167., 168., 169., 170., 171., 172., 173., 174., 175., 17
6.,
       177., 178., 179., 180., 181., 182., 183., 184., 185., 186., 18
7.,
       188., 189., 190., 191., 192., 193., 194., 195., 196., 197., 19
8.,
       199.])
```

In [9]:

```
type(range_0x01[0])
```

Out[9]:

```
numpy.float64
```

6. Create an array of five values evenly spaced between 0 and 1

In [10]:

```
np.linspace(0, 1, 5)
```

Out[10]:

```
array([0.  , 0.25, 0.5 , 0.75, 1.  ])
```

7. Reverse a given Vector

In [13]:

```
myarray = np.array([9, 8, 7, 6, 5, 4, 3, 2, 1, 0])


myarray = np.flipud(myarray)
myarray
```

Out[13]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

8. Find indices of non-zero elements from [12,34,0,4,0,2,3,0,123]

In [14]:

```
arr_0x02 = [12,34,0,4,0,2,3,0,123]
np.array([i for i, v in enumerate(arr_0x02) if v != 0])
```

Out[14]:

```
array([0, 1, 3, 5, 6, 8])
```

In [16]:

```
arr_0x02[arr_0x02 != 0]
arr_0x02
```

Out[16]:

```
[12, 34, 0, 4, 0, 2, 3, 0, 123]
```

9. Replace all even numbers in given arr vector with -1

In [18]:

```
arr = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14])

arr[arr % 2 == 0] = -1
arr
```

Out[18]:

```
array([ 1, -1,  3, -1,  5, -1,  7, -1,  9, -1, 11, -1, 13, -1])
```

In [19]:

```
```

```
---------------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
 last)
<ipython-input-19-fb9d7992aa36> in <module>
----> 1 arr[not(arr & 1)] = -1
      2 arr

ValueError: The truth value of an array with more than one element is
 ambiguous. Use a.any() or a.all()
```

10. Replace all odd numbers in arr with -1 without changing arr

In [ ]:

```
arr = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14])
z = np.where(arr % 2==1 , -1,arr)
z
```

11. Create a 5x3 array with random values (In - between 100 to 300) and find the minimum and maximum values (Hints : Use np.random.random)

In [25]:

```
r = np.random.randint(100,300,size=(5,3))
print(r)
print(min(min(i) for i in r))
print(max(max(i) for i in r))
print(r.min())
print(r.max())
```

```
[[108 180 188]
 [120 202 277]
 [222 206 263]
 [172 183 232]
 [205 140 289]]
108
289
108
289
```

12. Create a random vector of size 30 and find the mean value

In [22]:

```
arr_0x03 = np.array(np.random.randint(1, 300, size = 30))
print(arr_0x03)
np.mean(arr_0x03)
```

```
[181  28 233 269 281 171  40   4 127 208 109  20   7 127  34 260  11
33
 230 118 112  90 205 165 186 297  59 159 207 142]
```

Out[22]:

```
137.1
```

17. What is the result of the following expression?

```
0 * np.nan
np.nan == np.nan
np.inf > np.nan
np.nan - np.nan
np.nan in set([np.nan])
0.3 == 3 * 0.1
```

In [23]:

```
print(0 * np.nan)
print(np.nan == np.nan)
print(np.inf > np.nan)
print(np.nan - np.nan)
print(np.nan in set([np.nan]))
print(0.3 == 3 * 0.1)
```

```
nan
False
False
nan
True
False
```

18. Normalize a 5x5 random matrix (Hints - fourmula (x - mean) / std)

In [27]:

```
Z = np.random.random((5,5))
print(Z)
Z = (Z - Z.mean()) / Z.std()
Z
```

```
[[0.20608506 0.53568867 0.88902779 0.13781639 0.76054299]
 [0.75685501 0.5042531  0.69231653 0.56783593 0.9349615 ]
 [0.96449477 0.72177922 0.40692956 0.96402142 0.47609018]
 [0.23580509 0.72811963 0.54077535 0.71899994 0.80530001]
 [0.38873003 0.85035258 0.59435801 0.78661256 0.02194984]]
```

Out[27]:

```
array([[-1.56080446, -0.27950195,  1.09407006, -1.82619235,  0.5945977
],
       [ 0.58026103, -0.40170474,  0.32937389, -0.15453255,  1.2726329
8],
       [ 1.38744075,  0.44390727, -0.78004067,  1.38560063, -0.5111853
8],
       [-1.44527067,  0.46855501, -0.25972797,  0.43310306,  0.7685863
6],
       [-0.85078959,  0.94372391, -0.05143049,  0.69594066, -2.2766124
8]])
```

19. Multiply a 5x3 matrix by a 3x2 matrix (real matrix product)

In [33]:

```
mat_0x01 = np.random.randint(1, 3, size=(5, 3))
mat_0x02 = np.random.randint(1, 4, size=(3, 2))
print(mat_0x01, mat_0x02, sep="\n")
mat_r_0x01 = np.matmul(mat_0x01, mat_0x02)
mat_r_0x01

assert((len(mat_0x01), len(mat_0x02[0])) ==  mat_r_0x01.shape)
```

```
[[1 1 2]
 [1 1 1]
 [2 2 1]
 [2 1 2]
 [2 2 1]]
[[3 3]
 [2 3]
 [2 3]]
```

20. How to find common values between two arrays?

In [37]:

```
arr_0x2001 = np.random.randint(1, 10, size = 10)
arr_0x2002 = np.random.randint(1, 10, size = 10)
print(arr_0x2001, arr_0x2002, sep="\n")
np.intersect1d(arr_0x2001, arr_0x2002)
```

```
[7 3 4 7 6 4 1 6 3 7]
[5 3 1 8 9 2 8 8 2 9]
```

Out[37]:

```
array([1, 3])
```

21. How to get the dates of yesterday, today and tomorrow?

In [48]:

```
today = np.datetime64('today', 'D')
print("Today: ", today)

yesterday = np.datetime64('today', 'D') - np.timedelta64(1, 'D')
print("Yestraday: ", yesterday)

tomorrow = np.datetime64('today', 'D') + np.timedelta64(1, 'D')
print("Tomorrow: ", tomorrow)
```

```
Today:  2021-06-12
Yestraday:  2021-06-11
Tomorrow:  2021-06-13
```

22. How to get all the dates corresponding to the month of July 2016?

In [ ]:

```

```

23. Extract Integer part from vector

In [56]:

```
Z = np.array([1.234,2.345,5.678,666.543,123.99])

np.array(list(map(int, Z)))
```

Out[56]:

```
array([  1,   2,   5, 666, 123])
```

24. Convert a 1D array to a 2D array with 4 rows

In [47]:

```
one = np.arange(2,22)
np.reshape(one, (4, -1))
```

Out[47]:

```
array([[ 2,  3,  4,  5,  6],
       [ 7,  8,  9, 10, 11],
       [12, 13, 14, 15, 16],
       [17, 18, 19, 20, 21]])
```

25. Create two array (a and b) and stack them vertically?.(concatenate vertically?)

In [50]:

```
a_0x2501 = np.random.randint(1, 5, size = 4)
b_0x2501 = np.random.randint(1, 5, size = 4)
print(a_0x2501, b_0x2501, sep = "\n")
np.vstack((a_0x2501, b_0x2501))
```

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

Out[50]:

```
array([[4, 3, 2, 1],
       [2, 2, 4, 1]])
```

26. Create two 2Darray (a and b) and stack them horizontally.(concatenate horizontally)

In [52]:

```
print(a_0x2501, b_0x2501, sep = "\n")
np.hstack((a_0x2501, b_0x2501))
```

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

Out[52]:

```
array([4, 3, 2, 1, 2, 2, 4, 1])
```

27. Create two 2Darray (a and b) and stack last tow colums of b into a horizontally.(concatenate horizontally)

In [ ]:

```

```

28. Create a 2darray of 4X4 and swap 2nd and 4th column .

In [57]:

```
mat_0x2801 = np.random.randint(1, 10, size=(4, 4))
print(mat_0x2801)
mat_0x2801[:, 1], mat_0x2801[:, 3] = mat_0x2801[:, 3], mat_0x2801[:, 1].copy()
print(mat_0x2801)
```

```
[[2 7 8 6]
 [3 2 9 2]
 [1 1 7 9]
 [2 8 3 8]]
[[2 6 8 7]
 [3 2 9 2]
 [1 9 7 1]
 [2 8 3 8]]
```

29. Create a 2darray of 4X4 and swap 2nd and 4th rows

In [58]:

```
mat_0x2801 = np.random.randint(1, 10, size=(4, 4))
print(mat_0x2801)
mat_0x2801[1], mat_0x2801[3] = mat_0x2801[3], mat_0x2801[1].copy()
print(mat_0x2801)
```

```
[[6 4 4 5]
 [3 3 1 5]
 [5 3 4 3]
 [7 9 9 9]]
[[6 4 4 5]
 [7 9 9 9]
 [5 3 4 3]
 [3 3 1 5]]
```

In [ ]:

```

```