Assignment1

February 21, 2022

1 Problem 1

Import the numpy package under the name np
Create a vector or 1D array with 10 zeros and print it
Find the memory size of this array

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

[ ]:
```
arr = np.zeros(10)
```

[ ]:
```
arr
```

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

[ ]:
```
print("The Size of the array is", arr.itemsize*arr.size, "Bytes")
```

```
The Size of the array is 80 Bytes
```

2 Problem 2:

Create another vector or 1D array with values ranging from 10 to 20
Reverse the created vector (first element becomes last) -- Is there any NumPy method that you can use?

[ ]:
```
arr = np.arange(10,21,1)
```

[ ]:
```
arr
```

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

[ ]:
```
np.flip(arr)
```

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

3 Problem 3:

Create a 3x4 array with random values (standard normal distribution) and find the minimum and maximum values

[ ]:
```
from numpy import random as rand
```

[ ]:
```
arr3 = rand.randint(10000,size=([4,3]))
```

[ ]:
```
arr3
```

[ ]:
```
array([[8672, 9272, 6342],
       [3784, 5232, 7887],
       [5001, 4270, 3926],
       [6843, 5154, 7836]])
```

[ ]:
```
arr3.max()
```

[ ]:
```
9272
```

[ ]:
```
arr3.min()
```

[ ]:
```
3784
```

4 Problem 4:

Given the following 1D array, negate all elements which are between 3 and 8, in place. (include both 3 and 8 in conditional statements)

[ ]:
```
#note this will not run without completing the first step of problem 1
Z = np.arange(11)
```

[ ]:
```
Z
```

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

[ ]:
```
for i in range(0,Z.size):
  if(Z[i]>=3 and Z[i]<=8): Z[i]=-Z[i]
```

[ ]:
```
Z
```

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

Given the 1D array Z, find the closest value to the given scalar v?

[ ]:
```
Z = np.arange(5,100)
v = np.array(33.2)
```

[ ]:
```
Z
```

[ ]:
```
array([ 5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
       22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
       39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55,
       56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72,
       73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89,
       90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

[ ]:
```
Z.sort()
for i in range(0,Z.size):
  if(Z[i] >= v):
    small = Z[i-1];
    large = Z[i];
    break;
if(abs(small-v) > abs(large-v)): print("Closet Value is %d", large)
```

```
else: print("Closet Value is %d", small)
```

```
Closet Value is %d 33
```

Subtract the mean of each row of the following matrix

[ ]:
```
np.random.seed(2)
X = np.random.rand(3, 4)
print(X)
```

```
␣
↪---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call␣
↪last)

        <ipython-input-1-3b30b7c6267b> in <module>()
    ----> 1 np.random.seed(2)
          2 X = np.random.rand(3, 4)
          3 print(X)


        NameError: name 'np' is not defined
```

[ ]:
```
r1 = np.full(4, np.mean(X[0]))
r2 = np.full(4, np.mean(X[1]))
r3 = np.full(4, np.mean(X[2]))
```

[ ]:
```
Mean = np.array([r1,r2,r3])
Mean
```

[ ]:
```
array([[0.3617265 , 0.3617265 , 0.3617265 , 0.3617265 ],
       [0.39365556, 0.39365556, 0.39365556, 0.39365556],
       [0.42918947, 0.42918947, 0.42918947, 0.42918947]])
```

[ ]:
```
# complete the following:
Y = X - Mean
print(Y)
```

```
[[ 0.0742684  -0.33580027  0.18793598  0.07359589]
 [ 0.02671225 -0.06332073 -0.18900692  0.22561541]
 [-0.1295348  -0.16236219  0.19194436  0.09995263]]
```

Timing comparison for multiplication of 4 arrays. Find the fastest way to compute the multiplication ABCD. Make sure you report the elapsed time. (hint: you can find relevant information at https://youtu.be/SeBRHg9ZrSs)

Complete the following:

[ ]:
```
A = np.random.random((10000,1000))
B = np.random.random((1000,10000))
C = np.random.random((10000,5))
D = np.random.random((5,1000))
```

[ ]:
```
import time

start = time.time()
np.linalg.multi_dot([A, B, C, D])
end = time.time()

elapsed_time = end-start
print(elapsed_time)
```

```
0.19710588455200195
```

5 Problem 5

Import and print the file 'parks.csv' (Park Code should be the index column)

[ ]:
```
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>


Saving parks.csv to parks.csv
```

[ ]:
```
import pandas as pd
```

[ ]:
```
parks = pd.read_csv("parks.csv",index_col=0)
parks
```

[ ]:

Park Code	Park Name	...	Longitude
		...	
ACAD	Acadia National Park	...	-68.21
ARCH	Arches National Park	...	-109.57
BADL	Badlands National Park	...	-102.50
BIBE	Big Bend National Park	...	-103.25
BISC	Biscayne National Park	...	-80.08
BLCA	Black Canyon of the Gunnison National Park	...	-107.72
BRCA	Bryce Canyon National Park	...	-112.18
CANY	Canyonlands National Park	...	-109.93
CARE	Capitol Reef National Park	...	-111.17
CAVE	Carlsbad Caverns National Park	...	-104.44
CHIS	Channel Islands National Park	...	-119.42
CONG	Congaree National Park	...	-80.78
CRLA	Crater Lake National Park	...	-122.10
CUVA	Cuyahoga Valley National Park	...	-81.55

```
DENA                    Denali National Park and Preserve   ...    -150.50
DEVA                           Death Valley National Park   ...    -116.82
DRTO                           Dry Tortugas National Park   ...     -82.87
EVER                             Everglades National Park   ...     -80.93
GAAR       Gates Of The Arctic National Park and Preserve   ...    -153.30
GLAC                                 Glacier National Park   ...    -114.00
GLBA                Glacier Bay National Park and Preserve   ...    -137.00
GRBA                             Great Basin National Park   ...    -114.30
GRCA                           Grand Canyon National Park   ...    -112.14
GRSA          Great Sand Dunes National Park and Preserve   ...    -105.51
GRSM                   Great Smoky Mountains National Park   ...     -83.53
GRTE                             Grand Teton National Park   ...    -110.80
GUMO                    Guadalupe Mountains National Park   ...    -104.87
HALE                              Haleakala National Park   ...    -156.17
HAVO                         Hawaii Volcanoes National Park   ...    -155.20
HOSP                             Hot Springs National Park   ...     -93.05
ISRO                             Isle Royale National Park   ...     -88.55
JOTR                             Joshua Tree National Park   ...    -115.90
KATM                    Katmai National Park and Preserve   ...    -155.00
KEFJ                           Kenai Fjords National Park   ...    -149.65
KOVA                           Kobuk Valley National Park   ...    -159.28
LACL                Lake Clark National Park and Preserve   ...    -153.42
LAVO                         Lassen Volcanic National Park   ...    -121.51
MACA                            Mammoth Cave National Park   ...     -86.10
MEVE                              Mesa Verde National Park   ...    -108.49
MORA                           Mount Rainier National Park   ...    -121.75
NOCA                          North Cascades National Park   ...    -121.20
OLYM                                Olympic National Park   ...    -123.50
PEFO                         Petrified Forest National Park   ...    -109.78
PINN                               Pinnacles National Park   ...    -121.16
REDW                                Redwood National Park   ...    -124.00
ROMO                           Rocky Mountain National Park   ...    -105.58
SAGU                                Saguaro National Park   ...    -110.50
SEKI             Sequoia and Kings Canyon National Parks   ...    -118.68
SHEN                             Shenandoah National Park   ...     -78.35
THRO                     Theodore Roosevelt National Park   ...    -103.45
VOYA                               Voyageurs National Park   ...     -92.88
WICA                               Wind Cave National Park   ...    -103.48
WRST       Wrangell - St Elias National Park and Preserve   ...    -142.00
YELL                             Yellowstone National Park   ...    -110.50
YOSE                                Yosemite National Park   ...    -119.50
ZION                                    Zion National Park   ...    -113.05

[56 rows x 5 columns]
```

Print all column names

[ ]:
```
list(parks.columns)
```

[ ]:
```
['Park Name ', 'State ', 'Acres ', 'Latitude ', 'Longitude ']
```

Make sure tha all letters are lower case and replace space with _

[ ]:
```
parks = parks.astype(str).apply(lambda x: x.str.lower())
```

[ ]:
```
parks = parks.astype(str).apply(lambda x: x.str.rstrip())
```

[ ]:
```
parks = parks.astype(str).apply(lambda x: x.str.replace(' ','_'))
```

[ ]:
```
parks["State "] = parks["State "].apply(lambda state: state.replace('_',''))
```

Which state has the smallest national park?

[ ]:
```
parks = parks.astype({'Acres ': int})
```

[ ]:
```
parks["Acres "].min()
```

[ ]:
```
5550
```

[ ]:
```
parks.loc[parks['Acres ']==parks["Acres "].min()]
```

[ ]:

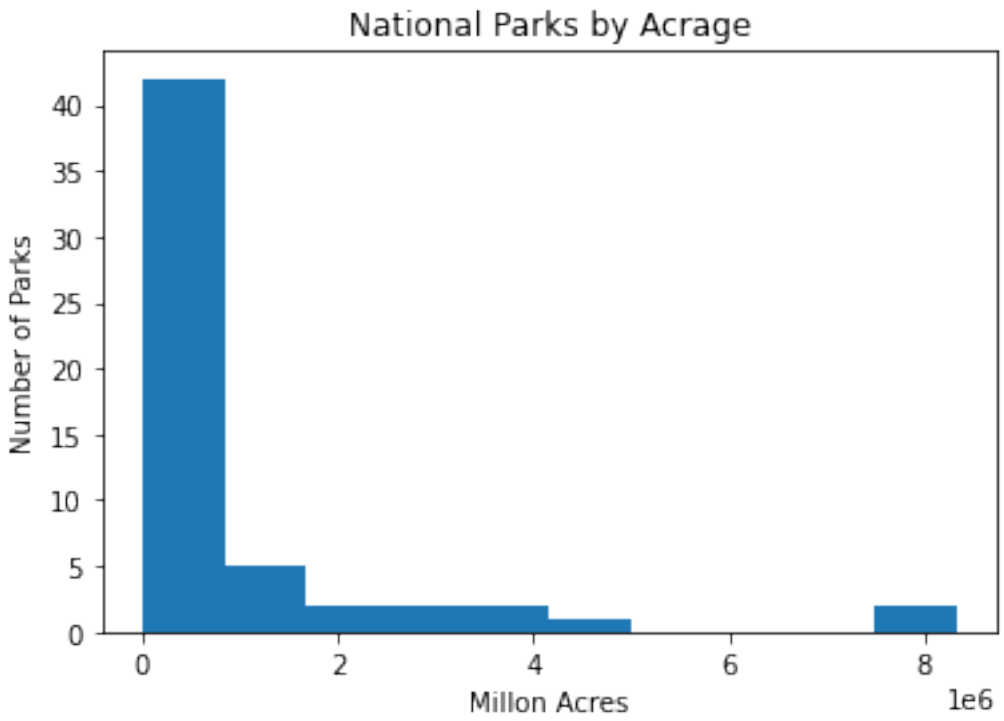
Park Code	Park Name	State	Acres	Latitude	Longitude
HOSP	hot_springs_national_park	ar	5550	34.51	-93.05

State is Arkansas

Produce a histogram plot that shows the distribution of 'acres'.

[ ]:
```
import matplotlib.pyplot as plt
```

[ ]:
```
plt.hist(parks['Acres '])
plt.title("National Parks by Acrage")
plt.xlabel("Millon Acres")
plt.ylabel("Number of Parks")
plt.show()
```


National Parks by Acrage
Number of Parks
40
35
30
25
20
15
10
5
0
0
2
4
6
8
Millon Acres
1e6

[ ]: