

```
# NumPy Exercise  
# Richa Patel  
# week 06
```

Type *Markdown* and LaTeX:  $\alpha^2$

```
In [23]: import numpy as np;  
np.random.seed(21)  
random_integers = np.random.randint(1,high=500000, size=(20, 5))  
random_integers
```

```
Out[23]: array([[ 80842, 333008, 202553, 140037,  81969],  
                [ 63857, 42105, 261540, 481981, 176739],  
                [489984, 326386, 110795, 394863,  25024],  
                [ 38317, 49982, 408830, 485118,  16119],  
                [407675, 231729, 265455, 109413, 103399],  
                [174677, 343356, 301717, 224120, 401101],  
                [140473, 254634, 112262,  25063, 108262],  
                [375059, 406983, 208947, 115641, 296685],  
                [444899, 129585, 171318, 313094, 425041],  
                [188411, 335140, 141681,  59641, 211420],  
                [287650,   8973, 477425, 382803, 465168],  
                [ 3975,  32213, 160603, 275485, 388234],  
                [246225,  56174, 244097,   9350, 496966],  
                [225516, 273338,  73335, 283013, 212813],  
                [ 38175, 282399, 318413, 337639, 379802],  
                [198049, 101115, 419547, 260219, 325793],  
                [148593, 425024, 348570, 117968, 107007],  
                [ 52547, 180346, 178760, 305186, 262153],  
                [ 11835, 449971, 494184, 472031, 353049],  
                [476442,  35455, 191553, 384154,  29917]])
```

1.Find average value of the second column

```
In [24]: a = np.average(random_integers, axis = 0)  
  
print("All column of average value: ", a)  
  
print("Average value of the second column :", a[1])
```

```
All column of average value:  [204660.05 214895.8  254579.25 258840.9  
5 243333.05]
```

```
Average value of the second column : 214895.8
```

2.The average value of the first 5 rows of the third and fourth columns.

```
In [25]: print("Average value of the first 5 rows of the third and fourth column")

arrayT = random_integers[:5]
print(arrayT)
z = np.average(arrayT, axis = 0)

print(z)

print("Average value of the third column :", z[2])

print("Average value of the fourth column :", z[3])
```

```
Average value of the first 5 rows of the third and fourth columns
[[ 80842 333008 202553 140037  81969]
 [ 63857  42105 261540 481981 176739]
 [489984 326386 110795 394863  25024]
 [ 38317  49982 408830 485118  16119]
 [407675 231729 265455 109413 103399]]
[216135. 196642. 249834.6 322282.4  80650. ]
Average value of the third column : 249834.6
Average value of the fourth column : 322282.4
```

3. 10 x 10 slice from of random\_integers array, named arrayTenByTen

```
In [26]: arrayTenByTen = np.reshape(random_integers, (10,10))
print(arrayTenByTen)
```

```
[[ 80842 333008 202553 140037  81969  63857  42105 261540 481981 1767
39]
 [489984 326386 110795 394863  25024  38317  49982 408830 485118  161
19]
 [407675 231729 265455 109413 103399 174677 343356 301717 224120 4011
01]
 [140473 254634 112262  25063 108262 375059 406983 208947 115641 2966
85]
 [444899 129585 171318 313094 425041 188411 335140 141681  59641 2114
20]
 [287650   8973 477425 382803 465168   3975  32213 160603 275485 3882
34]
 [246225  56174 244097   9350 496966 225516 273338  73335 283013 2128
13]
 [ 38175 282399 318413 337639 379802 198049 101115 419547 260219 3257
93]
 [148593 425024 348570 117968 107007  52547 180346 178760 305186 2621
53]
 [ 11835 449971 494184 472031 353049 476442  35455 191553 384154  299
17]]
```

4. new array from the arrayTenByTen that is one rank

```
In [27]: arrayTenFlat = arrayTenByTen.flatten()
print(arrayTenFlat)
```

```
[ 80842 333008 202553 140037  81969  63857  42105 261540 481981 17673
9
489984 326386 110795 394863  25024  38317  49982 408830 485118  1611
9
407675 231729 265455 109413 103399 174677 343356 301717 224120 40110
1
140473 254634 112262  25063 108262 375059 406983 208947 115641 29668
5
444899 129585 171318 313094 425041 188411 335140 141681  59641 21142
0
287650   8973 477425 382803 465168   3975  32213 160603 275485 38823
4
246225  56174 244097   9350 496966 225516 273338  73335 283013 21281
3
38175 282399 318413 337639 379802 198049 101115 419547 260219 32579
3
148593 425024 348570 117968 107007  52547 180346 178760 305186 26215
3
11835 449971 494184 472031 353049 476442  35455 191553 384154  2991
7]
```

5. What is the sum of arrayTenFlat

```
In [28]: print(arrayTenFlat.sum())
```

```
23526182
```

6. Iterate through arrayTenFlat

```
In [29]: for x in np.nditer(arrayTenFlat):
print(x)
```

```
80842
333008
202553
140037
81969
63857
42105
261540
481981
176739
489984
```

326386  
110795  
394863  
25024  
38317  
49982  
408830  
485118  
16119  
407675  
231729  
265455  
109413  
103399  
174677  
343356  
301717  
224120  
401101  
140473  
254634  
112262  
25063  
108262  
375059  
406983  
208947  
115641  
296685  
444899  
129585  
171318  
313094  
425041  
188411  
335140  
141681  
59641  
211420  
287650  
8973  
477425  
382803  
465168  
3975  
32213  
160603  
275485  
388234  
246225

56174  
244097  
9350  
496966  
225516  
273338  
73335  
283013  
212813  
38175  
282399  
318413  
337639  
379802  
198049  
101115  
419547  
260219  
325793  
148593  
425024  
348570  
117968  
107007  
52547  
180346  
178760  
305186  
262153  
11835  
449971  
494184  
472031  
353049  
476442  
35455  
191553  
384154  
29917

7. What is the value of the element (9,2) in random\_integers array?

In [30]: `print(random_integers[8,1])`

129585

8. What is the data type of arrayTenFlat?

```
In [31]: print(arrayTenFlat.dtype)

int64
```

9. In arrayTenFlat replace the value in index 5 with 42

```
In [32]: arrayTenFlat[4] = 42
print(arrayTenFlat[4])

42
```

10. Save the array random\_integers to a file. List the directory showing the saved array.

```
In [33]: np.save('savemydata', random_integers)
final = np.load('savemydata.npy')
final

Out[33]: array([[ 80842,  333008,  202553,  140037,   81969],
 [  63857,   42105,  261540,  481981,  176739],
 [489984,  326386,  110795,  394863,   25024],
 [  38317,   49982,  408830,  485118,   16119],
 [407675,  231729,  265455,  109413,  103399],
 [174677,  343356,  301717,  224120,  401101],
 [140473,  254634,  112262,   25063,  108262],
 [375059,  406983,  208947,  115641,  296685],
 [444899,  129585,  171318,  313094,  425041],
 [188411,  335140,  141681,   59641,  211420],
 [287650,    8973,  477425,  382803,  465168],
 [   3975,   32213,  160603,  275485,  388234],
 [246225,   56174,  244097,    9350,  496966],
 [225516,  273338,   73335,  283013,  212813],
 [  38175,  282399,  318413,  337639,  379802],
 [198049,  101115,  419547,  260219,  325793],
 [148593,  425024,  348570,  117968,  107007],
 [  52547,  180346,  178760,  305186,  262153],
 [  11835,  449971,  494184,  472031,  353049],
 [476442,   35455,  191553,  384154,   29917]])
```

```
In [ ]:
```

