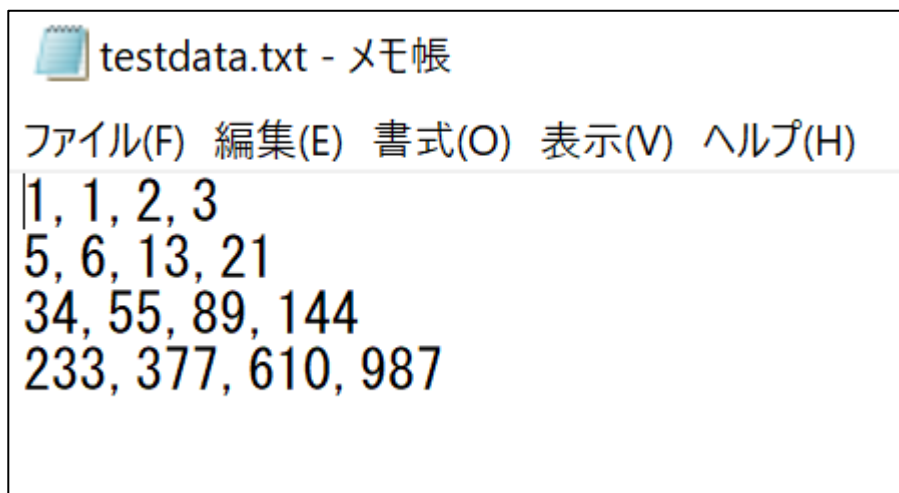


データの読み込み

- ① 予め以下のデータファイルを作成して
jupyter notebookのファイルと同じフォルダ
に入れておきます



np.loadやnp.saveでは、Excelデータなどの
使用が不便になるので、np.loadtxtや
np.savetxtが良い

② jupyter notebookにスクリプトを入力

```
import numpy as np
arr1=np.loadtxt('testdata.txt', delimiter=',')
print(arr1)
print()
print('平均'+str(np.mean(arr1, axis=0)))
print('平均'+str(np.mean(arr1, axis=1)))
print('中央'+str(np.median(arr1, axis=0)))
print('中央'+str(np.median(arr1, axis=1)))
print('偏差'+str(np.std(arr1, axis=0)))
print('偏差'+str(np.std(arr1, axis=1)))
```



```
[[ 1.  1.  2.  3.]
 [ 5.  6. 13. 21.]
 [34. 55. 89. 144.]
 [233. 377. 610. 987.]]
```

```
平均[ 68.25 109.75 178.5 288.75]
平均[ 1.75 11.25 80.5 551.75]
中央[19.5 30.5 51. 82.5]
中央[ 1.5 9.5 72. 493.5]
偏差[ 95.96711676 155.73274383 251.36875303 406.77043587]
偏差[ 0.8291562 6.41774883 41.58425183 285.0327832 ]
```

データ保存

```
import numpy as np
arr1=np.random.randint(0, 100, (10, 10))
print(arr1)
np.savetxt('testdata2.txt', arr1, delimiter=',')
print('データをファイルに保存しました')
```



```
[[ 2 39 77 58 80 41 63 35 17 94]
 [31 75 82 78 12 56 55 77 53 71]
 [39 71 64 15 22 81 29 34 48 62]
 [65  7 70 40 67 87 47  0 98 88]
 [85 54 75 74 88 49 70 45 80 18]
 [32 38 96 19 99 85 80 35 42 74]
 [55 23 98 96 11 18 82 62 74 11]
 [46 77 53 92 63  1  6 91 66 58]
 [53 38 66 99 86  0 57 53 90 90]
 [70 57 61 46 84 20 24 34 17 56]]
```

データをファイルに保存しました

testdata2.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
2. 0000000000000000e+00, 3. 9000000000000000e+01, 7. 7000000000000000e+01,
3. 1000000000000000e+01, 7. 5000000000000000e+01, 8. 2000000000000000e+01,
3. 9000000000000000e+01, 7. 1000000000000000e+01, 6. 4000000000000000e+01,
6. 5000000000000000e+01, 7. 0000000000000000e+00, 7. 0000000000000000e+01,
8. 5000000000000000e+01, 5. 4000000000000000e+01, 7. 5000000000000000e+01,
3. 2000000000000000e+01, 3. 8000000000000000e+01, 9. 6000000000000000e+01,
5. 5000000000000000e+01, 2. 3000000000000000e+01, 9. 8000000000000000e+01,
4. 6000000000000000e+01, 7. 7000000000000000e+01, 5. 3000000000000000e+01,
5. 3000000000000000e+01, 3. 8000000000000000e+01, 6. 6000000000000000e+01,
7. 0000000000000000e+01, 5. 7000000000000000e+01, 6. 1000000000000000e+01,
```

実数で保存

~

```
1. 7000000000000000e+01, 9. 4000000000000000e+01
5. 3000000000000000e+01, 7. 1000000000000000e+01
4. 8000000000000000e+01, 6. 2000000000000000e+01
9. 8000000000000000e+01, 8. 8000000000000000e+01
8. 0000000000000000e+01, 1. 8000000000000000e+01
4. 2000000000000000e+01, 7. 4000000000000000e+01
7. 4000000000000000e+01, 1. 1000000000000000e+01
6. 6000000000000000e+01, 5. 8000000000000000e+01
9. 0000000000000000e+01, 9. 0000000000000000e+01
1. 7000000000000000e+01, 5. 6000000000000000e+01
```

ソート

```
import numpy as np
arr1=np.random.randint(0,10,(3,3))
arr2=np.sort(arr1,axis=0)
arr3=np.sort(arr1,axis=1)
print(arr1)
print()
print(arr2)
print()
print(arr3)
```

axis=0 縦方向
axis=1 横方向



```
[[0 8 3]
 [2 0 0]
 [7 0 8]]
```

```
[[0 0 0]
 [2 0 3]
 [7 8 8]]
```

各列の小さい方から並べる

```
[[0 3 8]
 [0 0 2]
 [0 7 8]]
```

各行の小さい方から並べる

```
import numpy as np
arr1=np.random.randint(0,10,(3,3))
arr2=np.sort(arr1,axis=1)
print(arr2)
print()
print(arr2[::-1])
print()
print(arr2[:,::-1])
```



```
[[4 5 6]
 [2 6 8]
 [0 4 9]]
```

```
[[0 4 9]
 [2 6 8]
 [4 5 6]]
```

縦方向を逆に並べる

```
[[6 5 4]
 [8 6 2]
 [9 4 0]]
```

横方向を逆に並べる

スライス

```
import numpy as np
arr1=np.arange(100).reshape(10,10)
print(arr1)
print()
print(arr1[0:5:1,0:5:1])
print()
print(arr1[5:10:1,5:10:1])
print()
print(arr1[0:10:2,0:10:2])
```

左上から5×5で取り出す

中央から5×5で取り出す

1つおきに取り出す

[	0	1	2	3	4	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]

[	0	1	2	3	4]
[	10	11	12	13	14]
[	20	21	22	23	24]
[	30	31	32	33	34]
[	40	41	42	43	44]

[	55	56	57	58	59]
[	65	66	67	68	69]
[	75	76	77	78	79]
[	85	86	87	88	89]
[	95	96	97	98	99]

[	0	2	4	6	8]
[	20	22	24	26	28]
[	40	42	44	46	48]
[	60	62	64	66	68]
[	80	82	84	86	88]