lec2_step1

October 6, 2022

[1]:
```
## Python basics for novice data scientists, supported by Wagatsuma Lab@Kyutech
#
# The MIT License (MIT): Copyright (c) 2020 Hiroaki Wagatsuma and Wagatsuma␣
 ↪Lab@Kyutech
#
# Permission is hereby granted, free of charge, to any person obtaining a copy␣
 ↪of this software and associated documentation files (the "Software"), to␣
 ↪deal in the Software without restriction, including without limitation the␣
 ↪rights to use, copy, modify, merge, publish, distribute, sublicense, and/or␣
 ↪sell copies of the Software, and to permit persons to whom the Software is␣
 ↪furnished to do so, subject to the following conditions:
# The above copyright notice and this permission notice shall be included in␣
 ↪all copies or substantial portions of the Software.
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR␣
 ↪IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,␣
 ↪FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE␣
 ↪AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER␣
 ↪LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING␣
 ↪FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS␣
 ↪IN THE SOFTWARE. */
#
# # @Time    : 2020-10-14
# # @Author  : Hiroaki Wagatsuma
# # @Site    : https://github.com/hirowgit/2A_python_basic_course
# # @IDE     : Python 3.7.7 (default, Mar 10 2020, 15:43:27) [Clang 10.0.0␣
 ↪(clang-1000.11.45.5)] on darwin
# # @File    : lec1_step5.py
```

[3]:
```
# first idea
OpenList=[1,2,3,4]
OpenList
```

[3]:
```
[1, 2, 3, 4]
```

[5]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[8]:
```
a = np.array([4,2,5,5,9,4])
print(a)
```

```
[4 2 5 5 9 4]
```

[15]:
```
itemD = np.array([4,2,5,5,9,4])
print(itemD)
```

```
[4 2 5 5 9 4]
```

[9]:
```
np.sum(a)
```

[9]:
```
29
```

[ ]:
```
spList=1:spN-1;

filltSum=cell(length(itemD),1);
filltSum(:)={itemD};

shiftS=0:length(itemD)-1;
shiftCell=num2cell(shiftS)';
sMat=cell2mat(cellfun(@(x,y) circshift(x,[0␣
 ↪y]),filltSum,shiftCell,'UniformOutput',false));
sMat2=triu(sMat);

itemDAll=sum(sMat2);
```

[ ]:
```
np.full(itemD)
filltSum=np.array

import numpy.matlib
a0 = np.array(1)
np.matlib.repmat(a0, 2, 3)
array([[1, 1, 1],
       [1, 1, 1]])
```

[27]:
```
import numpy.matlib
a0 = np.array(1)
filltSum=np.matlib.repmat(itemD, 10,1)
print(filltSum)
```

```
[[4 2 5 5 9 4]
 [4 2 5 5 9 4]
 [4 2 5 5 9 4]
 [4 2 5 5 9 4]
 [4 2 5 5 9 4]
 [4 2 5 5 9 4]
 [4 2 5 5 9 4]
```

```
 [4 2 5 5 9 4]
 [4 2 5 5 9 4]
 [4 2 5 5 9 4]]
```

[44]:
```
print(itemD)
print('')
shiftS=np.arange(0,len(itemD))
print(shiftS)
# np.roll(filltSum[0],shiftS[0])
# np.roll(filltSum[1],shiftS[1])
print('')
sMat=[np.roll(filltSum[i],shiftS[i]) for i in range(0,len(shiftS)) ]
print(sMat)
print('')
sMat2=np.triu(sMat)
print(sMat2)
print('')
itemDAll=np.sum(sMat2,axis=0)
print(itemDAll)
```

```
[4 2 5 5 9 4]

[0 1 2 3 4 5]

[array([4, 2, 5, 5, 9, 4]), array([4, 4, 2, 5, 5, 9]), array([9, 4, 4, 2, 5,
5]), array([5, 9, 4, 4, 2, 5]), array([5, 5, 9, 4, 4, 2]), array([2, 5, 5, 9, 4,
4])]

[[4 2 5 5 9 4]
 [0 4 2 5 5 9]
 [0 0 4 2 5 5]
 [0 0 0 4 2 5]
 [0 0 0 0 4 2]
 [0 0 0 0 0 4]]

[ 4  6 11 16 25 29]
```

[35]:
```
for i in range(shiftS):
    print(i)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/var/folders/mg/w5t8lkhc8xj79f001s7kzpfh0000gp/T/ipykernel_28141/1789563555.py␣
 ↪in <module>
----> 1 for i in range(shiftS):
      2     print(i)
```

```
TypeError: only integer scalar arrays can be converted to a scalar index
```

[13]:
```
x = np.arange(10)
print(x)
np.roll(x, 2)
```

```
[0 1 2 3 4 5 6 7 8 9]
```

[13]:
```
array([8, 9, 0, 1, 2, 3, 4, 5, 6, 7])
```

[12]:
```
np.roll(x, -2)
```

[12]:
```
array([2, 3, 4, 5, 6, 7, 8, 9, 0, 1])
```

[14]:
```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if "a" in x]

print(newlist)
```

```
['apple', 'banana', 'mango']
```