In [84]:

```
import numpy

world_alcohol = numpy.genfromtxt("world_alcohol.txt", delimiter=",",dtype=str)
print (type(world_alcohol))
print (world_alcohol)
print (help(numpy.genfromtxt))
```

```
<class 'numpy.ndarray'>
[['Year' 'WHO region' 'Country' 'Beverage Types' 'Display Value']
 ['1986' 'Western Pacific' 'Viet Nam' 'Wine' '0']
 ['1986' 'Americas' 'Uruguay' 'Other' '0.5']
 ...
 ['1987' 'Africa' 'Malawi' 'Other' '0.75']
 ['1989' 'Americas' 'Bahamas' 'Wine' '1.5']
 ['1985' 'Africa' 'Malawi' 'Spirits' '0.31']]
Help on function genfromtxt in module numpy.lib.npyio:

genfromtxt(fname, dtype=<class 'float'>, comments='#', delimiter=None, skip_header
=0, skip_footer=0, converters=None, missing_values=None, filling_values=None, usec
ols=None, names=None, excludelist=None, deletechars=None, replace_space='_', autos
trip=False, case_sensitive=True, defaultfmt='f%i', unpack=None, usemask=False, loo
se=True, invalid_raise=True, max_rows=None, encoding='bytes')
    Load data from a text file, with missing values handled as specified.

    Each line past the first `skip_header` lines is split at the `delimiter`
    character, and characters following the `comments` character are discarded.
```

In [82]:

```
vector = numpy.array([5, 10, 15, 20])
matrix = numpy.array([[5, 10, 15], [20, 25, 30], [35, 40, 45]])
print (matrix)
print (vector)
```

```
[[ 5 10 15]
 [20 25 30]
 [35 40 45]]
[ 5 10 15 20]
```

In [83]:

```
vector = numpy.array([1, 2, 3, 4])
print(vector.shape)
matrix = numpy.array([[5, 10, 15], [20, 25, 30]])
print(matrix.shape)
```

```
(4,)
(2, 3)
```

In [20]:

```
import numpy
numbers = numpy.array([1, 2, 3, 4.])
print (numbers)
numbers.dtype
```

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

Out[20]:

```
dtype('float64')
```

In [22]:

```
world_alcohol
```

Out[22]:

```
array([['Year', 'WHO region', 'Country', 'Beverage Types',
        'Display Value'],
       ['1986', 'Western Pacific', 'Viet Nam', 'Wine', '0'],
       ['1986', 'Americas', 'Uruguay', 'Other', '0.5'],
       ...,
       ['1987', 'Africa', 'Malawi', 'Other', '0.75'],
       ['1989', 'Americas', 'Bahamas', 'Wine', '1.5'],
       ['1985', 'Africa', 'Malawi', 'Spirits', '0.31']], dtype='<U52')
```

```

```

In [24]:

```
world_alcohol = numpy.genfromtxt("world_alcohol.txt", delimiter="," ,dtype=str, skip_header=1)
print(world_alcohol)
```

```
[['1986' 'Western Pacific' 'Viet Nam' 'Wine' '0']
 ['1986' 'Americas' 'Uruguay' 'Other' '0.5']
 ['1985' 'Africa' "Cte d'Ivoire" 'Wine' '1.62']
 ...
 ['1987' 'Africa' 'Malawi' 'Other' '0.75']
 ['1989' 'Americas' 'Bahamas' 'Wine' '1.5']
 ['1985' 'Africa' 'Malawi' 'Spirits' '0.31']]
```

In [27]:

```
uruguay_other_1986 = world_alcohol[1,2]
third_country = world_alcohol[2,2]
print (uruguay_other_1986)
print (third_country)
```

```
Uruguay
Cte d'Ivoire
```

In [28]:

```
vector = numpy.array([5, 10, 15, 20])
print(vector[0:3])
```

```
[ 5 10 15]
```

In [30]:

```
matrix= numpy.array([
     [5, 10, 15],
    [20, 25, 30],
    [35, 40, 45],
])
print (matrix[:,0:2])
```

```
[[ 5 10]
 [20 25]
 [35 40]]
```

In [31]:

```
vector = numpy.array([5,10,15,20])
vector == 10
```

Out[31]:

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

In [34]:

```
matrix= numpy.array([
     [5, 10, 15],
    [20, 25, 30],
    [35, 40, 45],
])
matrix == 25
```

Out[34]:

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

In [36]:

```
vector = numpy.array([5,10,15,20])
equal_to_ten = (vector == 10)
print (equal_to_ten)
print(vector[equal_to_ten])
```

```
[False  True False False]
[10]
```

In [37]:

```
matrix= numpy.array([
     [5, 10, 15],
    [20, 25, 30],
    [35, 40, 45],
])
second_column_25 = (matrix[:,1] == 25)
print (second_column_25)
print (matrix[second_column_25, :])
```

```
[False  True False]
[[20 25 30]]
```

In [42]:

```
vector = numpy.array([5, 10, 15, 20])
equal_to_ten = (vector == 10) | (vector == 15)
print (equal_to_ten)
```

```
[False  True  True False]
```

In [43]:

```
vector = numpy.array(["1", "2", "3"])
print (vector.dtype)
print (vector)
vector = vector.astype(float)
print (vector.dtype)
print (vector)
```

```
<U1
['1' '2' '3']
float64
[1. 2. 3.]
```

In [45]:

```
vector = numpy.array([5, 10, 15, 20])
vector.min()
```

Out[45]:

```
5
```

In [46]:

```
matrix= numpy.array([
     [5, 10, 15],
    [20, 25, 30],
    [35, 40, 45],
])
matrix.sum(axis=1)
```

Out[46]:

```
array([ 30,  75, 120])
```

In [47]:

```
matrix= numpy.array([
     [5, 10, 15],
    [20, 25, 30],
    [35, 40, 45],
])
matrix.sum(axis=0)
```

Out[47]:

```
array([60, 75, 90])
```

In [50]:

```
import numpy as np
print (np.arange(15))
a = np.arange(15).reshape(3, 5)
a
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]
```

Out[50]:

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
```

In [51]:

```
a.shape
```

Out[51]:

```
(3, 5)
```

In [52]:

```
a.ndim
```

Out[52]:

```
2
```

In [53]:

```
a.dtype.name
```

Out[53]:

```
'int32'
```

In [54]:

```
a.size
```

Out[54]:

```
15
```

In [55]:

```
np.zeros((3,4))
```

Out[55]:

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

In [56]:

```
np.ones( (2,3,4), dtype=np.int32)
```

Out[56]:

```
array([[[1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1]],

       [[1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1]]])
```

In [57]:

```
np.arange( 10, 30, 5)
```

Out[57]:

```
array([10, 15, 20, 25])
```

In [58]:

```
np.arange(12).reshape(4,3)
```

Out[58]:

```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])
```

In [59]:

```
np.random.random((2,3))
```

Out[59]:

```
array([[0.45445318, 0.4115358 , 0.2275358 ],
       [0.06970885, 0.54709909, 0.59671975]])
```

In [60]:

```
from numpy import pi
np.linspace(0, 2*pi, 100)
```

Out[60]:

```
array([0.        , 0.06346652, 0.12693304, 0.19039955, 0.25386607,
       0.31733259, 0.38079911, 0.44426563, 0.50773215, 0.57119866,
       0.63466518, 0.6981317 , 0.76159822, 0.82506474, 0.88853126,
       0.95199777, 1.01546429, 1.07893081, 1.14239733, 1.20586385,
       1.26933037, 1.33279688, 1.3962634 , 1.45972992, 1.52319644,
       1.58666296, 1.65012947, 1.71359599, 1.77706251, 1.84052903,
       1.90399555, 1.96746207, 2.03092858, 2.0943951 , 2.15786162,
       2.22132814, 2.28479466, 2.34826118, 2.41172769, 2.47519421,
       2.53866073, 2.60212725, 2.66559377, 2.72906028, 2.7925268 ,
       2.85599332, 2.91945984, 2.98292636, 3.04639288, 3.10985939,
       3.17332591, 3.23679243, 3.30025895, 3.36372547, 3.42719199,
       3.4906585 , 3.55412502, 3.61759154, 3.68105806, 3.74452458,
       3.8079911 , 3.87145761, 3.93492413, 3.99839065, 4.06185717,
       4.12532369, 4.1887902 , 4.25225672, 4.31572324, 4.37918976,
       4.44265628, 4.5061228 , 4.56958931, 4.63305583, 4.69652235,
       4.75998887, 4.82345539, 4.88692191, 4.95038842, 5.01385494,
       5.07732146, 5.14078798, 5.2042545 , 5.26772102, 5.33118753,
       5.39465405, 5.45812057, 5.52158709, 5.58505361, 5.64852012,
       5.71198664, 5.77545316, 5.83891968, 5.9023862 , 5.96585272,
       6.02931923, 6.09278575, 6.15625227, 6.21971879, 6.28318531])
```

In [61]:

```
a = np.array( [20,30,40,50] )
b = np.arange( 4)
print (a)
print (b)
c = a-b
print (c)
c= c-1
print (c)
b**2
print (b**2)
print(a<35)
```

```
[20 30 40 50]
[0 1 2 3]
[20 29 38 47]
[19 28 37 46]
[0 1 4 9]
[ True  True False False]
```

In [65]:

```
a = np.array( [[1,1],
            [0,1]])
b = np.array( [[2,0],
              [3,4]])
print (a)
print ('---------')
print (b)
print('------')
print(a*b)
print('----')
print(a.dot(b))
print('------')
print (np.dot(a, b))
```

```
[[1 1]
 [0 1]]
---------
[[2 0]
 [3 4]]
------
[[2 0]
 [0 4]]
----
[[5 4]
 [3 4]]
------
[[5 4]
 [3 4]]
```

In [66]:

```
import numpy as np
b = np.arange(3)
print (b)
print (np.exp(b))
print (np.sqrt(b))
```

```
[0 1 2]
[1.         2.71828183 7.3890561 ]
[0.         1.         1.41421356]
```

In [69]:

```
a = np.floor(10*np.random.random((3,4)))
print (a)
print ("------------------")
print (a.ravel())
print ("------")
a.shape = (6,2)
print (a)
print ("-------")
print (a,-1)
```

```
[[0. 1. 8. 8.]
 [4. 5. 7. 0.]
 [7. 8. 1. 9.]]
------------------
[0. 1. 8. 8. 4. 5. 7. 0. 7. 8. 1. 9.]
------
[[0. 1.]
 [8. 8.]
 [4. 5.]
 [7. 0.]
 [7. 8.]
 [1. 9.]]
-------
[[0. 1.]
 [8. 8.]
 [4. 5.]
 [7. 0.]
 [7. 8.]
 [1. 9.]] -1
```

In [71]:

```
a = np.floor(10*np.random.random((2,2)))
b = np.floor(10*np.random.random((2,2)))
print (a)
print ("-----------")
print (b)
print("-------------")
print (np.hstack((a,b)))
```

```
[[6. 6.]
 [4. 4.]]
-----------
[[5. 4.]
 [5. 7.]]
-------------
[[6. 6. 5. 4.]
 [4. 4. 5. 7.]]
```

In [73]:

```
a = np.floor(10*np.random.random((2,12)))
print (a)
print ("---------")
print (np.hsplit(a,3))
print ("-----")
print (np.hsplit(a, (3,4)))
a = np.floor(10*np.random.random((12,2)))
print ("--------")
print (a)
np.vsplit(a,3)
```

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

Out[73]:

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

In [74]:

```
a = np.arange(12)
b = a
print (b is a)
b.shape = 3,4
print (a.shape)
print (id(a))
print (id(b))
```

```
True
(3, 4)
2119207634064
2119207634064
```

In [75]:

```
c = a.view()
print (c is a)
c.shape = 2,6
print (a.shape)
c[0,4] = 1234
print (a)
print (id(a))
print (id(c))
```

```
False
(3, 4)
[[   0    1    2    3]
 [1234    5    6    7]
 [   8    9   10   11]]
2119207634064
2119206929872
```

In [76]:

```
d = a.copy()
d is a
a[0,0] = 9999
print (d)
print (a)
```

```
[[   0    1    2    3]
 [1234    5    6    7]
 [   8    9   10   11]]
[[9999    1    2    3]
 [1234    5    6    7]
 [   8    9   10   11]]
```

In [77]:

```
import numpy as np
data = np.sin(np.arange(20)).reshape(5,4)
print (data)
ind = data.argmax(axis=0)
print (ind)
data_max = data[ind, range(data.shape[1])]
print (data_max)
```

```
[[ 0.          0.84147098  0.90929743  0.14112001]
 [-0.7568025  -0.95892427 -0.2794155   0.6569866 ]
 [ 0.98935825  0.41211849 -0.54402111 -0.99999021]
 [-0.53657292  0.42016704  0.99060736  0.65028784]
 [-0.28790332 -0.96139749 -0.75098725  0.14987721]]
[2 0 3 1]
[0.98935825 0.84147098 0.99060736 0.6569866 ]
```

In [80]:

```
a = np.arange(0, 40 ,10)
print (a)
b = np.tile(a, (2,2))
print (b)
```

```
[ 0 10 20 30]
[[ 0 10 20 30  0 10 20 30]
 [ 0 10 20 30  0 10 20 30]]
```

In [ ]: