In [3]:

```
import pandas

dict = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
       "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],
       "area": [8.516, 17.10, 3.286, 9.597, 1.221],
       "population": [200.4, 143.5, 1252, 1357, 52.98] }

import pandas as pd
brics = pd.DataFrame(dict)
print(brics)
```

```
        country    capital    area  population
0        Brazil   Brasilia   8.516      200.40
1        Russia     Moscow  17.100      143.50
2         India  New Dehli   3.286     1252.00
3         China    Beijing   9.597     1357.00
4  South Africa   Pretoria   1.221       52.98
```

In [4]:

```
# Set the index for brics
brics.index = ["BR", "RU", "IN", "CH", "SA"]

# Print out brics with new index values
print(brics)
```

```
         country    capital    area  population
BR        Brazil   Brasilia   8.516      200.40
RU        Russia     Moscow  17.100      143.50
IN         India  New Dehli   3.286     1252.00
CH         China    Beijing   9.597     1357.00
SA  South Africa   Pretoria   1.221       52.98
```

In [6]:

```
# Import pandas as pd
import pandas as pd

# Import the cars.csv data: cars
movies = pd.read_csv('movies.csv')

# Print out cars
print(movies)
```

```
---------------------------------------------------------------------------
UnicodeDecodeError                        Traceback (most recent call last)
pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._convert_tokens()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._convert_with_dty
pe()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._string_convert()

pandas\_libs\parsers.pyx in pandas._libs.parsers._string_box_utf8()

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe1 in position 2: inval
id continuation byte

During handling of the above exception, another exception occurred:

UnicodeDecodeError                        Traceback (most recent call last)
<ipython-input-6-21769f0d6261> in <module>
      3 
      4 # Import the cars.csv data: cars
----> 5 movies = pd.read_csv('movies.csv')
      6 
      7 # Print out cars

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\io\parsers.py in parser_f(filepath_or_buffer, sep, delimiter, header, n
ames, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, c
onverters, true_values, false_values, skipinitialspace, skiprows, skipfooter,
nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, pars
e_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_d
ates, iterator, chunksize, compression, thousands, decimal, lineterminator, q
uotechar, quoting, doublequote, escapechar, comment, encoding, dialect, error
_bad_lines, warn_bad_lines, delim_whitespace, low_memory, memory_map, float_p
recision)
    683         )
    684 
--> 685         return _read(filepath_or_buffer, kwds)
    686 
    687     parser_f.__name__ = name

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\io\parsers.py in _read(filepath_or_buffer, kwds)
    461 
    462     try:
--> 463         data = parser.read(nrows)
    464     finally:
    465         parser.close()

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\io\parsers.py in read(self, nrows)
   1152     def read(self, nrows=None):
   1153         nrows = _validate_integer("nrows", nrows)
-> 1154         ret = self._engine.read(nrows)
   1155 
   1156         # May alter columns / col_dict

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
```

```
andas\io\parsers.py in read(self, nrows)
   2057     def read(self, nrows=None):
   2058         try:
-> 2059             data = self._reader.read(nrows)
   2060         except StopIteration:
   2061             if self._first_chunk:

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader.read()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._read_low_memory
()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._read_rows()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._convert_column_d
ata()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._convert_tokens()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._convert_with_dty
pe()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._string_convert()

pandas\_libs\parsers.pyx in pandas._libs.parsers._string_box_utf8()

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe1 in position 2: inval
id continuation byte
```

In [8]:

```
import numpy as np
```

In [9]:

```
import pandas as pd
```

In [10]:

```
import numpy as np

a = np.array([1, 2, 3])   # Create a rank 1 array
print(type(a))            # Prints "<class 'numpy.ndarray'>"
print(a.shape)            # Prints "(3,)"
print(a[0], a[1], a[2])   # Prints "1 2 3"
a[0] = 5                  # Change an element of the array
print(a)
```

```
<class 'numpy.ndarray'>
(3,)
1 2 3
[5 2 3]
```

In [ ]:

In [11]:

```
b = np.array([[1,2,3],[4,5,6]])    # Create a rank 2 array
print(b.shape)                     # Prints "(2, 3)"
print(b[0, 0], b[0, 1], b[1, 0])   # Prints "1 2 4"
```

```
(2, 3)
1 2 4
```

In [12]:

```
import numpy as np

a = np.zeros((2,2))   # Create an array of all zeros
print(a)              # Prints "[[ 0.  0.]
                      #          [ 0.  0.]]"

b = np.ones((1,2))    # Create an array of all ones
print(b)              # Prints "[[ 1.  1.]]"

c = np.full((2,2), 7)  # Create a constant array
print(c)               # Prints "[[ 7.  7.]
                       #          [ 7.  7.]]"

d = np.eye(2)         # Create a 2x2 identity matrix
print(d)              # Prints "[[ 1.  0.]
                      #          [ 0.  1.]]"

e = np.random.random((2,2))  # Create an array filled with random values
print(e)                     # Might print "[[ 0.91940167  0.08143941]
                             #               [ 0.68744134  0.87236687]]"
```

```
[[0. 0.]
 [0. 0.]]
[[1. 1.]]
[[7 7]
 [7 7]]
[[1. 0.]
 [0. 1.]]
[[0.94848057 0.54973251]
 [0.13567348 0.00698117]]
```

In [13]:

```
import numpy as np

a = np.array([1, 2, 3,4,5])

print(a)
```

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

In [14]:

```
type(a)
```

Out[14]:

```
numpy.ndarray
```

In [15]:

```
a.shape
```

Out[15]:

```
(5,)
```

In [16]:
```
a.ndim
```

Out[16]:
```
1
```

In [19]:
```
a2 = np.array([[1, 2, 3,4,5],[7,8,9,10]])

a2.shape
```

Out[19]:
```
(2,)
```

a2.ndim

In [25]:
```
np.zeros((5,5,5))
```

Out[25]:
```
array([[[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]]])
```

In [26]:
```
np.ones((3,3,))
```

Out[26]:
```
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

```
In [148]: data =np.random.randint(1,10,(4,4))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-148-4019c63a9998> in <module>
----> 1 data =np.random.randint(size=(1,10,(4,4)))

mtrand.pyx in numpy.random.mtrand.RandomState.randint()

TypeError: randint() takes at least 1 positional argument (0 given)
```

```
In [30]: data
```

```
Out[30]: array([[8, 1, 7, 6],
                [1, 4, 5, 9],
                [6, 3, 3, 6],
                [9, 8, 2, 8]])
```

```
In [31]: data.reshape(8,2)
```

```
Out[31]: array([[8, 1],
                [7, 6],
                [1, 4],
                [5, 9],
                [6, 3],
                [3, 6],
                [9, 8],
                [2, 8]])
```

```
In [32]: data.reshape(-1,8)
```

```
Out[32]: array([[8, 1, 7, 6, 1, 4, 5, 9],
                [6, 3, 3, 6, 9, 8, 2, 8]])
```

```
In [33]: data.reshape(-1,1) #single cell
```

```
Out[33]: array([[8],
                [1],
                [7],
                [6],
                [1],
                [4],
                [5],
                [9],
                [6],
                [3],
                [3],
                [6],
                [9],
                [8],
                [2],
                [8]])
```

```
In [40]: data.reshape(-2,8).ndim #dimensional array

Out[40]: 2
```

```
In [41]: data[2:]

Out[41]: array([[6, 3, 3, 6],
                [9, 8, 2, 8]])
```

```
In [42]: data[1:3]

Out[42]: array([[1, 4, 5, 9],
                [6, 3, 3, 6]])
```

```
In [58]: data[:,2:4].ndim

Out[58]: 2
```

operations

```
In [59]: np.max(data)

Out[59]: 9
```

```
In [61]: np.max(data, axis=1) #axis 1 = rows ,axis=0 2 = colums

Out[61]: array([8, 9, 6, 9])
```

```
In [67]: np.max(data, axis=0)

Out[67]: array([9, 8, 7, 9])
```

```
In [72]: np.sort(data)

Out[72]: array([[1, 6, 7, 8],
                [1, 4, 5, 9],
                [3, 3, 6, 6],
                [2, 8, 8, 9]])
```

```
In [75]: np.log10(data)

Out[75]: array([[0.90308999, 0.        , 0.84509804, 0.77815125],
                [0.        , 0.60205999, 0.69897   , 0.95424251],
                [0.77815125, 0.47712125, 0.47712125, 0.77815125],
                [0.95424251, 0.90308999, 0.30103   , 0.90308999]])
```

In [78]:

```
np.matmul(data,data)
```

Out[78]:

```
array([[161,  81,  94, 147],
       [123, 104,  60, 144],
       [123,  75,  78, 129],
       [164, 111, 125, 202]])
```

In [82]:

```
np.linalg.inv(data)
```

Out[82]:

```
array([[ 0.01030928, -0.12371134,  0.18900344, -0.01030928],
       [ 0.15979381,  0.08247423, -0.7371134 ,  0.34020619],
       [ 0.32474227,  0.10309278, -0.71305842,  0.17525773],
       [-0.25257732,  0.03092784,  0.70274914, -0.24742268]])
```

In [94]:

```
#Using numpy.random.random
mat1 = np.random.random(size=(3,4))
print ('matrix generated from numpy.random.random is \n%s\n'%mat1)
mat2 = np.random.randint(low=0,high=2,size=(3,4))
print ('matrix generated from numpy.random.random is \n%s\n'%mat2)
mat3 = np.random.randn(3,4)
print ('matrix generated from numpy.random.randn is \n%s\n'%mat3)
```

```
matrix generated from numpy.random.random is
[[0.55817832 0.25657989 0.59702424 0.70015224]
 [0.74697253 0.94344625 0.86675727 0.01513514]
 [0.35020044 0.09751582 0.8381611  0.63273702]]

matrix generated from numpy.random.random is
[[0 1 1 0]
 [0 1 0 1]
 [0 1 1 1]]

matrix generated from numpy.random.randn is
[[ 1.02350398 -1.75664733 -1.86068041 -1.55995923]
 [ 0.15297584  2.37346288  0.72669258  0.08288785]
 [-1.97906297  0.09573856 -1.19682964  0.42197132]]
```

In [96]:

```
vec = np.arange(10)
vec1 = vec[:-3]
print ('Result of removing last 3 elements from range(10) : \n%s\n'%vec1)
```

```
Result of removing last 3 elements from range(10) :
[0 1 2 3 4 5 6]
```

In [97]:

```
mat = np.random.randint(0,6,(3,5))
# Create a submatrix with first 2 rows and last 2 columns
submat1 = mat[0:2,-2:]
print ('Original Matrix is \n%s\n'%mat)
print ('Sub Matrix with first 2 rows and last 2 columns is \n%s\n'%submat1)
submat2 = mat[:,3:0:-1]
print ('After flipping the columns of the matrix, it looks : \n%s\n'%submat2)
```

```
Original Matrix is
[[0 1 5 2 5]
 [0 3 1 0 2]
 [3 3 4 5 5]]

Sub Matrix with first 2 rows and last 2 columns is
[[2 5]
 [0 2]]

After flipping the columns of the matrix, it looks :
[[2 5 1]
 [0 1 3]
 [5 4 3]]
```

In [107]:

```
#Create a null array of size 10 but the fifth value which is 1


ndmat = np.zeros(shape=(5,5))

print(ndmat)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

In [119]:

```
#Reverse a above created array (first element becomes last)


a = np.array([1,2,3,4])


b = a[::-1]
print(b)
```

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

In [132]:

```
#Create a 3x3 matrix with values ranging from 0 to 8

import numpy as np
x =  np.arange(2, 11).reshape(3,3)
print(x)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

In [ ]:

```

```

In [140]:

```
#Create a 3x3x3 array with random values

y = np.random.random(size=(3,3))

print(y)
```

```
[[0.35242416 0.1734646  0.94903611]
 [0.77253706 0.28629444 0.55765066]
 [0.43828655 0.17630346 0.90036273]]
```

In [156]:

```
#Create a 10x10 array with random values and find the minimum and maximum valu
es

p = np.random.randint(10,10)
print(p)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-156-aa7d590d875a> in <module>
      1 #Create a 10x10 array with random values and find the minimum and max
imum values
      2 
----> 3 p = np.random.randint(10,10)
      4 print(p)

mtrand.pyx in numpy.random.mtrand.RandomState.randint()

bounded_integers.pyx in numpy.random.bounded_integers._rand_int32()

ValueError: low >= high
```

In [154]:
```
data =np.random.randint(1,10,(4,4))

print(data)
ind = np.where(data>0)

data[ind]
```

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

Out[154]:
```
array([3, 9, 7, 5, 7, 6, 6, 4, 7, 4, 3, 4, 7, 1, 8, 3])
```

In [158]:
```
#np.matmul(data,data)


data1 = np.random.random(5,3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-158-271fb8e86098> in <module>
      2
      3
----> 4 data1 = np.random.random(5,3)

mtrand.pyx in numpy.random.mtrand.RandomState.random()

TypeError: random() takes at most 1 positional argument (2 given)
```

In [161]:

```
import numpy as np

x = np.random.random((5,3))
print("First array:")
print(x)
y = np.random.random((3,2))
print("Second array:")
print(y)
z = np.dot(x, y)
print("Dot product of two arrays:")
print(z)
```

```
First array:
[[0.50724963 0.0557758  0.99490296]
 [0.67796839 0.48309573 0.98260946]
 [0.74541681 0.5163861  0.22542276]
 [0.36472589 0.25434333 0.77950322]
 [0.51412122 0.93823611 0.49344137]]
Second array:
[[0.57920981 0.73214692]
 [0.01861633 0.72403141]
 [0.38391906 0.66662398]]
Dot product of two arrays:
[[0.67680451 1.07499086]
 [0.77892191 1.50117997]
 [0.52791003 1.06990659]
 [0.51525389 0.97082103]
 [0.50469211 1.38466452]]
```

In [197]:

```
x1 = np.random.randint(0,10,(16,16))

x1

res = np.zeros((4,4))

res[0,0] = 1


print(res)
```

```
[[1. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

In [206]:

```
res[0,0] = np.sum(data[:4,:4])


res
```

Out[206]:

```
array([[84.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
```

In [207]:
```
for i in range(4):
    for j in range(4):
        res[i,j]= np.sum[res[:4]]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-207-fe6b929a3440> in <module>
      1 for i in range(4):
      2     for j in range(4):
----> 3         res[i,j]= np.sum[res[:4]]

TypeError: 'function' object is not subscriptable
```

In [217]:
```
res[0,1] = np.sum(data[:8,4:8])

res[0,8] = np.sum(data[:4,4:8])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-217-430f87945ada> in <module>
      1 res[0,1] = np.sum(data[4:8,4:8])
      2 
----> 3 res[0,8] = np.sum(data[:4,4:8])

IndexError: index 8 is out of bounds for axis 1 with size 4
```

In [222]:
```
res[0,2] = np.sum(x1[:4,4:8])

res
```

Out[222]:
```
array([[84.,  0., 68.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
```

In [ ]:
```

```

In [221]:
```
data
```

Out[221]:
```
array([[3, 9, 7, 5],
       [7, 6, 6, 4],
       [7, 4, 3, 4],
       [7, 1, 8, 3]])
```

In [225]:
```
res[0,2] = np.sum(x1[:4,4:6])

res
```

Out[225]:
```
array([[84.,  0., 39.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
```

In [231]:

```
for i in range(4):
    for j in range(4):
        res[i,j] = np.sum(x1[i *4: i*4+4,j*4: j*4+4])


print(res)
x1
```

```
[[71. 68. 69. 67.]
 [40. 71. 85. 65.]
 [88. 54. 84. 73.]
 [72. 80. 54. 74.]]
```

Out[231]:

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

In [239]:

```
x,y = np.hsplit(x1,[8])

x
```

Out[239]:

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

In [241]:

```
result = []
for e in res:
    result.append(np.vsplit(data,[4,8,12]))

result
```

Out[241]:

```
[[array([[3, 9, 7, 5],
         [7, 6, 6, 4],
         [7, 4, 3, 4],
         [7, 1, 8, 3]]),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32)],
 [array([[3, 9, 7, 5],
         [7, 6, 6, 4],
         [7, 4, 3, 4],
         [7, 1, 8, 3]]),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32)],
 [array([[3, 9, 7, 5],
         [7, 6, 6, 4],
         [7, 4, 3, 4],
         [7, 1, 8, 3]]),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32)],
 [array([[3, 9, 7, 5],
         [7, 6, 6, 4],
         [7, 4, 3, 4],
         [7, 1, 8, 3]]),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32),
  array([], shape=(0, 4), dtype=int32)]]
```

In [244]:

```
final = []
for r in result:
    final.append(np.sum(r))


final
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-244-9e16a896e300> in <module>
      1 final = []
      2 for r in result:
----> 3     final.append(np.sum(r))
      4 
      5 

<__array_function__ internals> in sum(*args, **kwargs)

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\n
umpy\core\fromnumeric.py in sum(a, axis, dtype, out, keepdims, initial, wher
e)
   2180 
   2181     return _wrapreduction(a, np.add, 'sum', axis, dtype, out, keepdim
s=keepdims,
-> 2182                           initial=initial, where=where)
   2183 
   2184 

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\n
umpy\core\fromnumeric.py in _wrapreduction(obj, ufunc, method, axis, dtype, o
ut, **kwargs)
     88                 return reduction(axis=axis, out=out, **passkwargs)
     89 
---> 90     return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
     91 
     92 

ValueError: operands could not be broadcast together with shapes (4,4) (0,4)
```

In [246]:

```
#Iterates over its argument and adding each element to the list and extending
 the list.
#The length of the list increases by number of elements in it’s argument.

my_list = ['geeks', 'for']
another_list = [6, 0, 4, 1]
my_list.extend(another_list)
print( my_list )
```

```
['geeks', 'for', 6, 0, 4, 1]
```

In [247]:

```
pd.__version__
```

Out[247]:

```
'0.25.1'
```

In [249]:

```
pd.Series(data=[1,2,3,4,5],index=['a','b','c','d','e'])
```

Out[249]:

```
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

In [252]:
```
df =pd.DataFrame({'s1':'s1', 's2':'s2'})

df
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-252-53f7be6bcac4> in <module>
----> 1 df =pd.DataFrame({'s1':'s1', 's2':'s2'})

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\frame.py in __init__(self, data, index, columns, dtype, copy)
    409             )
    410         elif isinstance(data, dict):
--> 411             mgr = init_dict(data, index, columns, dtype=dtype)
    412         elif isinstance(data, ma.MaskedArray):
    413             import numpy.ma.mrecords as mrecords

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\internals\construction.py in init_dict(data, index, columns, dtyp
e)
    255             arr if not is_datetime64tz_dtype(arr) else arr.copy() for
arr in arrays
    256         ]
--> 257     return arrays_to_mgr(arrays, data_names, index, columns, dtype=dt
ype)
    258 
    259 

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\internals\construction.py in arrays_to_mgr(arrays, arr_names, inde
x, columns, dtype)
     75     # figure out the index, if necessary
     76     if index is None:
---> 77         index = extract_index(arrays)
     78     else:
     79         index = ensure_index(index)

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\internals\construction.py in extract_index(data)
    356 
    357         if not indexes and not raw_lengths:
--> 358             raise ValueError("If using all scalar values, you must pa
ss an index")
    359 
    360         if have_series:

ValueError: If using all scalar values, you must pass an index
```

In [257]:

```
df = pd.DataFrame({'A':ser1, 'B':ser2})
df
```

Out[257]:

```
   A   B
a  1  11
b  2  22
c  3  33
d  4  44
e  5  55
```

In [ ]:

```

```

In [256]:

```
ser2 = pd.Series(data=[11,22,33,44,55], index=list('abcde'))

ser1 = pd.Series(data=[1,2,3,4,5], index=list('abcde'))
```

In [264]:

```
#https://raw.githubusercontent.com/zekelabs/data-science-complete-tutorial/master/Data/titanic-train.csv.txt


titanic = pd.read_csv("https://raw.githubusercontent.com/zekelabs/data-science-complete-tutorial/master/Data/titanic-train.csv.txt",index_col='PassengerId')

titanic
```

Out[264]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
PassengerId										
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na
...	...	...	...	...	...	...	...	...	...	.
887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	Na
888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B4
889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	Na
890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C14
891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	Na

891 rows × 11 columns

In [267]:
```
titanic[['Survived','Name','Sex','Age']].head()
```

Out[267]:

	Survived	Name	Sex	Age
PassengerId				
1	0	Braund, Mr. Owen Harris	male	22.0
2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
3	1	Heikkinen, Miss. Laina	female	26.0
4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
5	0	Allen, Mr. William Henry	male	35.0

In [268]:
```
titanic[['Name']]
```

Out[268]:

	Name
PassengerId	
1	Braund, Mr. Owen Harris
2	Cumings, Mrs. John Bradley (Florence Briggs Th...
3	Heikkinen, Miss. Laina
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)
5	Allen, Mr. William Henry
...	...
887	Montvila, Rev. Juozas
888	Graham, Miss. Margaret Edith
889	Johnston, Miss. Catherine Helen "Carrie"
890	Behr, Mr. Karl Howell
891	Dooley, Mr. Patrick

891 rows × 1 columns

In [ ]:
```

```

In [269]:

```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 891 entries, 1 to 891
Data columns (total 11 columns):
Survived    891 non-null int64
Pclass      891 non-null int64
Name        891 non-null object
Sex         891 non-null object
Age         714 non-null float64
SibSp       891 non-null int64
Parch       891 non-null int64
Ticket      891 non-null object
Fare        891 non-null float64
Cabin       204 non-null object
Embarked    889 non-null object
dtypes: float64(2), int64(4), object(5)
memory usage: 66.1+ KB
```

In [275]:
```
titanic[titanic.Age.isnull()]
```

Out[275]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN
18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN
20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN
27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7.2250	NaN
29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7.8792	NaN
...	...	...	...	...	...	...	...	...	...	...
860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2292	NaN
864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN
869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN
879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN
889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN

177 rows × 11 columns

In [ ]:
```
```

In [286]:
```
titanic[titanic['Embarked'].str.match('Dawson')]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-286-b18a2f2a09ba> in <module>
----> 1 titanic[titanic['Embarked'].str.match('Dawson')]

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\frame.py in __getitem__(self, key)
   2968
   2969         # Do we have a (boolean) 1d indexer?
-> 2970         if com.is_bool_indexer(key):
   2971             return self._getitem_bool_array(key)
   2972

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\common.py in is_bool_indexer(key)
    128             if not lib.is_bool_array(key):
    129                 if isna(key).any():
--> 130                     raise ValueError(na_msg)
    131                 return False
    132             return True

ValueError: cannot index with vector containing NA / NaN values
```

In [ ]:

In [ ]:

In [ ]:

In [276]:
```
titanic[titanic.Embarked.isnull()]
```

Out[276]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
PassengerId											
62	1	1	Icard, Miss. Amelie	female	38.0	0	0	113572	80.0	B28	
830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	113572	80.0	B28	

In [ ]:

In [296]:

```
titanic[titanic.Fare > 200000]
```

Out[296]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
PassengerId										

In [ ]:

In [ ]:

In [ ]:

In [280]:

```
df = titanic[titanic.Age > 60]

df.Survived.value_counts()
```

Out[280]:

```
0    17
1     5
Name: Survived, dtype: int64
```

In [281]:

```
titanic.describe()
```

Out[281]:

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [289]:
```
titanic.fillna({'Age':29.6,'Embarked':'$'})
```

Out[289]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
PassengerId										
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na
...	...	...	...	...	...	...	...	...	...	..
887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	Na
888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B4
889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.6	1	2	W./C. 6607	23.4500	Na
890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C14
891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	Na

891 rows × 11 columns

In [ ]:

In [ ]:

In [290]:

```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 891 entries, 1 to 891
Data columns (total 11 columns):
Survived    891 non-null int64
Pclass      891 non-null int64
Name        891 non-null object
Sex         891 non-null object
Age         714 non-null float64
SibSp       891 non-null int64
Parch       891 non-null int64
Ticket      891 non-null object
Fare        891 non-null float64
Cabin       204 non-null object
Embarked    889 non-null object
dtypes: float64(2), int64(4), object(5)
memory usage: 66.1+ KB
```

In [293]:
```
titanic.drop(['Cabin'],axis=1,inplace= True)

titanic.info()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-293-a081bed361eb> in <module>
----> 1 titanic.drop(['Cabin'],axis=1,inplace= True)
      2 
      3 titanic.info()

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\frame.py in drop(self, labels, axis, index, columns, level, inplac
e, errors)
   4100             level=level,
   4101             inplace=inplace,
-> 4102             errors=errors,
   4103         )
   4104 

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\generic.py in drop(self, labels, axis, index, columns, level, inpl
ace, errors)
   3912         for axis, labels in axes.items():
   3913             if labels is not None:
-> 3914                 obj = obj._drop_axis(labels, axis, level=level, error
s=errors)
   3915 
   3916         if inplace:

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\generic.py in _drop_axis(self, labels, axis, level, errors)
   3944                 new_axis = axis.drop(labels, level=level, errors=erro
rs)
   3945             else:
-> 3946                 new_axis = axis.drop(labels, errors=errors)
   3947             result = self.reindex(**{axis_name: new_axis})
   3948 

c:\users\crypto\appdata\local\programs\python\python37-32\lib\site-packages\p
andas\core\indexes\base.py in drop(self, labels, errors)
   5338         if mask.any():
   5339             if errors != "ignore":
-> 5340                 raise KeyError("{} not found in axis".format(labels[m
ask]))
   5341             indexer = indexer[~mask]
   5342         return self.delete(indexer)

KeyError: "['Cabin'] not found in axis"
```

In [294]:
```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 891 entries, 1 to 891
Data columns (total 10 columns):
Survived    891 non-null int64
Pclass      891 non-null int64
Name        891 non-null object
Sex         891 non-null object
Age         714 non-null float64
SibSp       891 non-null int64
Parch       891 non-null int64
Ticket      891 non-null object
Fare        891 non-null float64
Embarked    889 non-null object
dtypes: float64(2), int64(4), object(4)
memory usage: 62.6+ KB
```

In [295]:

```
titanic[titanic.Age > 60]
```

Out[295]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Em
PassengerId										
34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0	C.A. 24579	10.5000	
55	0	1	Ostby, Mr. Engelhart Cornelius	male	65.0	0	1	113509	61.9792	
97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	
117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7500	
171	0	1	Van der hoef, Mr. Wyckoff	male	61.0	0	0	111240	33.5000	
253	0	1	Stead, Mr. William Thomas	male	62.0	0	0	113514	26.5500	
276	1	1	Andrews, Miss. Kornelia Theodosia	female	63.0	1	0	13502	77.9583	
281	0	3	Duane, Mr. Frank	male	65.0	0	0	336439	7.7500	
327	0	3	Nysveen, Mr. Johan Hansen	male	61.0	0	0	345364	6.2375	
439	0	1	Fortune, Mr. Mark	male	64.0	1	4	19950	263.0000	
457	0	1	Millet, Mr. Francis Davis	male	65.0	0	0	13509	26.5500	
484	1	3	Turkula, Mrs. (Hedwig)	female	63.0	0	0	4134	9.5875	
494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609	49.5042	
546	0	1	Nicholson, Mr. Arthur Ernest	male	64.0	0	0	693	26.0000	
556	0	1	Wright, Mr. George	male	62.0	0	0	113807	26.5500	
571	1	2	Harris, Mr. George	male	62.0	0	0	S.W./PP 752	10.5000	
626	0	1	Sutton, Mr. Frederick	male	61.0	0	0	36963	32.3208	
631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	30.0000	

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Em
PassengerId										
673	0	2	Mitchell, Mr. Henry Michael	male	70.0	0	0	C.A. 24580	10.5000	
746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1	WE/P 5735	71.0000	
830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	113572	80.0000	
852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7750	

In [300]:
```
titanic[titanic.Fare > 500.000]
```

Out[300]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarke
PassengerId										
259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	
680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	
738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	

In [303]:
```
titanic.Fare.max()
```

Out[303]:
```
512.3292
```

In [308]:
```
titanic[titanic.Fare == titanic.Fare.max()]['Name'].values
```

Out[308]:
```
array(['Ward, Miss. Anna', 'Cardeza, Mr. Thomas Drake Martinez',
       'Lesurer, Mr. Gustave J'], dtype=object)
```

```
In [321]: titanic.groupby(['Survived','Pclass','Sex']).size()
```

```
Out[321]: Survived  Pclass  Sex   
          0         1       female      3
                            male       77
                    2       female      6
                            male       91
                    3       female     72
                            male      300
          1         1       female     91
                            male       45
                    2       female     70
                            male       17
                    3       female     72
                            male       47
          dtype: int64
```

```
In [322]: titanic.groupby(['Survived','Pclass','Sex']).Age.mean()
```

```
Out[322]: Survived  Pclass  Sex   
          0         1       female    25.666667
                            male      44.581967
                    2       female    36.000000
                            male      33.369048
                    3       female    23.818182
                            male      27.255814
          1         1       female    34.939024
                            male      36.248000
                    2       female    28.080882
                            male      16.022000
                    3       female    19.329787
                            male      22.274211
          Name: Age, dtype: float64
```

In [323]:

```
titanic
```

Out[323]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emb
PassengerId										
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	
...	...	...	...	...	...	...	...	...	...	
887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	
888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	
889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	
890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	
891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	

891 rows × 10 columns

In [325]:

```
def fun(a):
    if a<21:
        return "Kid"
    elif (a >=21) and (a<60):
        return "Adult"
    else:
        return "Old"

titanic['Age_Status'] = titanic.Age.map(fun)

titanic
```

Out[325]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emb
PassengerId										
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	
...	...	...	...	...	...	...	...	...	...	
887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	
888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	
889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	
890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	
891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	

891 rows × 11 columns

In [330]:

```
def incr_fare(r):
    if (r['Sex'] == 'male') and (r['Age_Status'] == 'Adult'):
        return 2*r['Fare']
    else:
        return r['Fare']

titanic['Fare'] = titanic.apply(incr_fare, axis=1)


titanic
```

Out[330]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Eml
PassengerId										
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	29.0000	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	32.2000	
...	...	...	...	...	...	...	...	...	...	
887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	52.0000	
888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	
889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	
890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	120.0000	
891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	31.0000	

891 rows × 11 columns

In [333]:

```
marks = pd.DataFrame({'Name':['A','B','C','D'], 'Marks':[[2,3,4],[5,4,6],[7,8,
9],[2,4,2,4]]})
```

In [336]:

```
marks.explode('Marks')
```

Out[336]:

```
   Name  Marks
0     A      2
0     A      3
0     A      4
1     B      5
1     B      4
1     B      6
2     C      7
2     C      8
2     C      9
3     D      2
3     D      4
3     D      2
3     D      4
```