SWE4012 - MACHINE LEARNING_LAB-1_NUMPY&PANDA

August 6, 2022

1 NAME: B DEVI PRASAD

2 REG NO: 19MIS1018

3 SLOT: L13+L14

4 Faculty: Prof. BHARADWAJA KUMAR

5 TOPIC : PYTHON_PACKAGES(NUMPY_PANDAS)

6 NUMPY

Arrays- Structured list of Numbers Vectors(1-D array), matrix(2-D,3-D array), Images(array of pixels value), Tensors(Combination of matrices value), convnets

[1]:
```
import numpy as np
d=[[45,32,12,89],[52,63,17,80]]
print(d)
print(type(d))
arr = np.array(d)
print(arr)
print(type(arr))
```

```
[[45, 32, 12, 89], [52, 63, 17, 80]]
<class 'list'>
[[45 32 12 89]
 [52 63 17 80]]
<class 'numpy.ndarray'>
```

[2]:
```
dimension=arr.shape
rows=arr.shape[0]
columns=arr.shape[1]
tot=len(arr)
print("Dimension: ",dimension)
print("Rows: ",rows)
print("Columns: ",columns)
#print("Channels: ",channels)
print("Length: ",tot)
```

```
Dimension:  (2, 4)
Rows:  2
Columns:  4
Length:  2
```

[3]:
```
print(arr.dtype)
```

```
int32
```

[4]:
```
arr2 = np.zeros((3,6))
print(arr2)
```

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

[5]:
```
arr3 = np.arange(10)
print(arr3)
```

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

[6]:
```
z=np.ones((2,5))
z
```

[6]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

[7]:
```
z=np.eye(5,5)
z
```

[7]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

[8]:
```
v = 89
z=np.full((3,3),v)
z
```

[8]:
```
array([[89, 89, 89],
       [89, 89, 89],
       [89, 89, 89]])
```

[9]:
```
z1=np.ones_like(z)
print(z1)
z2=np.zeros_like(z)
print(z2)
```

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

[10]:
```
z=np.random.rand(5,3)
print(z)
```

```
[[0.39958795 0.49727174 0.4681512 ]
 [0.93646492 0.93541213 0.88824998]
 [0.02316568 0.38261328 0.31388912]
 [0.19335246 0.87360255 0.89478479]
 [0.62916518 0.3856495  0.0267111 ]]
```

7 Array - Slicing

[11]:
```
arr=[11,22,43,42,64,91,62,74,60,57]
arr[6:10]
```

[11]:
```
[62, 74, 60, 57]
```

[12]:
```
arr=np.array([[12,29,31,43],[51,63,30,71],[7,8,10,6]])
arr
```

[12]:
```
array([[12, 29, 31, 43],
       [51, 63, 30, 71],
       [ 7,  8, 10,  6]])
```

[13]:
```
b=arr[1:3,:2]
b
```

[13]:
```
array([[51, 63],
       [ 7,  8]])
```

[14]:
```
b[0][0]=0
b
```

[14]:
```
array([[ 0, 63],
       [ 7,  8]])
```

[15]:
```
arr
```

[15]:
```
array([[12, 29, 31, 43],
       [ 0, 63, 30, 71],
       [ 7,  8, 10,  6]])
```

[16]:
```
arr3 = np.arange(10)
arr3[6:8]=19
arr3
```

[16]:
```
array([ 0,  1,  2,  3,  4,  5, 19, 19,  8,  9])
```

[17]:
```
ss = np.array([[arr3[1],arr3[2],arr3[3]]])
ss[0]=3
arr3
```

[17]:
```
array([ 0,  1,  2,  3,  4,  5, 19, 19,  8,  9])
```

8 Reshape, transpose & Sign

[18]:
```
re_arr= np.arange(10).reshape(2,5)
print(re_arr)
```

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

[19]:
```
re_arr[1:]
```

[19]:
```
array([[5, 6, 7, 8, 9]])
```

[20]:
```
re_arr.T
```

[20]:
```
array([[0, 5],
       [1, 6],
       [2, 7],
       [3, 8],
       [4, 9]])
```

[21]:
```
arr = np.array([[10,-33,40],[-61,73,10]])
np.sign(arr)
```

[21]:
```
array([[ 1, -1,  1],
       [-1,  1,  1]])
```

[22]:
```
arr = np.array([[18,23,42],[67,74,82]])
print(np.sqrt(arr))
arr2 =np.sqrt(arr)
```

```
[[4.24264069 4.79583152 6.4807407 ]
 [8.18535277 8.60232527 9.05538514]]
```

[23]:
```
print(np.floor(arr2))
print('\n')
print(np.ceil(arr2))
```

```
[[4. 4. 6.]
 [8. 8. 9.]]


[[ 5.  5.  7.]
 [ 9.  9. 10.]]
```

[24]:
```
print(np.round(arr2))
```

```
[[4. 5. 6.]
 [8. 9. 9.]]
```

[25]:
```
print(np.log(arr2))
```

```
[[1.44518588 1.56774711 1.86883481]
 [2.10234631 2.15203255 2.20335962]]
```

9 Binary Functions

[26]:
```
a = np.array([44,31,23])
b=  np.array([31,63,79])
print(np.add(a,b))
print(np.subtract(a,b))
print(np.multiply(a,b))
print(np.divide(a,b))
print(np.mod(a,b))
```

```
[ 75  94 102]
[ 13 -32 -56]
[1364 1953 1817]
[1.41935484 0.49206349 0.29113924]
[13 31 23]
```

[27]:
```
a = np.array([47,31,22])
b=  np.array([[33],[66],[77]])
print("Multiply: " ,np.multiply(a,b))
print("Dot: ",a.dot(b))
```

```
Multiply:  [[1551 1023  726]
 [3102 2046 1452]
 [3619 2387 1694]]
Dot:  [5291]
```

10 Logical Functions

[28]:
```
print(a)
print(b)
print(np.greater(a,b))
print(np.less(a,b))
print(np.greater_equal(a,b))
print(np.equal(a,b))
```

```
[47 31 22]
[[33]
 [66]
 [77]]
[[ True False False]
 [False False False]
 [False False False]]
[[False  True  True]
 [ True  True  True]
 [ True  True  True]]
[[ True False False]
 [False False False]
 [False False False]]
[[False False False]
 [False False False]
 [False False False]]
```

11 Statistics Functions

[29]:
```
arr = np.array([[13,21,35],[36,27,58],[51,12,62]])
print(np.mean(arr))
print(np.std(arr))
print(np.var(arr))
```

```
35.0
17.60050504325878
309.77777777777777
```

12 Append

[30]:
```
a= np.array([[12,2,3],[3,2,5]])
b= np.array([[5,26,32],[1,8,9]])
np.append(a,b,axis=0)
```

[30]:
```
array([[12,  2,  3],
       [ 3,  2,  5],
       [ 5, 26, 32],
       [ 1,  8,  9]])
```

[31]:
```
a= np.array([[51,21,36],[31,27,59]])
b= np.array([[50,26,32],[10,81,90]])
np.append(a,b,axis=1)
```

[31]:
```
array([[51, 21, 36, 50, 26, 32],
       [31, 27, 59, 10, 81, 90]])
```

13 Broadcasting

[32]:
```
x=np.array([[11,22,33],[44,55,66],[77,88,99]])
x
```

[32]:
```
array([[11, 22, 33],
       [44, 55, 66],
       [77, 88, 99]])
```

[33]:
```
v=np.array([[1,0,1]])
v
```

[33]:
```
array([[1, 0, 1]])
```

[34]:
```
y=np.empty_like(x)
for i in range(len(x)):
    y[i,:] = x[i,:]+v[:]
y
```

[34]:
```
array([[ 12,  22,  34],
       [ 45,  55,  67],
       [ 78,  88, 100]])
```

14 Combining Arrays

1)Vertical Stack -

Combines array vertically i.e places the second array below first array.

The two arrays should have same no of columns. 2) Horizontal stack-

Combines array horizontally i.e places the second array to the right of first array.

[35]:
```
a1=np.array([[1,2,3]])
a2=np.array([[9,22,36]])
np.vstack((a1,a2))
```

[35]:
```
array([[ 1,  2,  3],
       [ 9, 22, 36]])
```

[36]:
```
a1=np.array([[1,2,3]])
a2=np.array([[9,22,36]])
np.hstack((a1,a2))
```

[36]:
```
array([[ 1,  2,  3,  9, 22, 36]])
```

15 Pandas

is the standard python library to work with dataframes. Unlike in R, this is not a part of base python and must be imported separately. It is typically imported as pd:import pandas as pd

[37]:
```
import pandas as pd
list1=['CAT-1','CAT-2','DA-1','DA-2','DA-3','FAT']
Calculas=[45,23,10,9,10,56]
Ethics=[32,45,10,7,6.5,45]
SoftwareProjectManagent=[25,28,9,7,10,76.5]
titles=['ID','Calculas','Ethics','SoftwareProjectManagent']
values=[list1,Calculas,Ethics,SoftwareProjectManagent]
result=list(zip(titles,values))
result
```

[37]:
```
[('ID', ['CAT-1', 'CAT-2', 'DA-1', 'DA-2', 'DA-3', 'FAT']),
 ('Calculas', [45, 23, 10, 9, 10, 56]),
 ('Ethics', [32, 45, 10, 7, 6.5, 45]),
 ('SoftwareProjectManagent', [25, 28, 9, 7, 10, 76.5])]
```

[38]:
```
result1=dict(result)
df1=pd.DataFrame(result1)
print(df1)
```

```
     ID  Calculas  Ethics  SoftwareProjectManagent
0  CAT-1        45    32.0                     25.0
1  CAT-2        23    45.0                     28.0
2   DA-1        10    10.0                      9.0
3   DA-2         9     7.0                      7.0
4   DA-3        10     6.5                     10.0
5    FAT        56    45.0                     76.5
```

[39]:
```
df2=pd.read_csv(r'student.csv')
df2
```

[39]:

	Sl.No	StudnetName	CAT-1	CAT-2	DA-1	DA-2	DA-3	FAT
0	1	DeviPrasad	32.0	23.4	10.0	10	8.33	58.0
1	2	Dellip	25.0	33.0	7.0	9	10.00	67.0
2	3	Gayathri	32.6	44.0	7.6	10	10.00	92.0
3	4	Pandhu	17.0	43.0	9.0	9	8.00	72.5
4	5	Eswar	22.0	34.0	10.0	10	10.00	78.0

[40]:
```
file=pd.read_csv(r'student.csv')
print(file.head(5))
print("\ntail part\n")
print(file.tail(5))
#read_excel('file.xlsx')
#read_csv('file.txt',delimiter='\t')
```

```
   Sl.No StudnetName  CAT-1  CAT-2  DA-1  DA-2   DA-3   FAT
0      1  DeviPrasad   32.0   23.4  10.0    10   8.33  58.0
1      2      Dellip   25.0   33.0   7.0     9  10.00  67.0
2      3    Gayathri   32.6   44.0   7.6    10  10.00  92.0
3      4      Pandhu   17.0   43.0   9.0     9   8.00  72.5
4      5       Eswar   22.0   34.0  10.0    10  10.00  78.0

tail part

   Sl.No StudnetName  CAT-1  CAT-2  DA-1  DA-2   DA-3   FAT
0      1  DeviPrasad   32.0   23.4  10.0    10   8.33  58.0
1      2      Dellip   25.0   33.0   7.0     9  10.00  67.0
2      3    Gayathri   32.6   44.0   7.6    10  10.00  92.0
3      4      Pandhu   17.0   43.0   9.0     9   8.00  72.5
4      5       Eswar   22.0   34.0  10.0    10  10.00  78.0
```

[41]:
```
file.describe()
```

[41]:

	Sl.No	CAT-1	CAT-2	DA-1	DA-2	DA-3	FAT
count	5.000000	5.000000	5.000000	5.0000	5.000000	5.00000	5.000000
mean	3.000000	25.720000	35.480000	8.7200	9.600000	9.26600	73.500000
std	1.581139	6.655224	8.417363	1.3755	0.547723	1.01182	12.708265
min	1.000000	17.000000	23.400000	7.0000	9.000000	8.00000	58.000000
25%	2.000000	22.000000	33.000000	7.6000	9.000000	8.33000	67.000000
50%	3.000000	25.000000	34.000000	9.0000	10.000000	10.00000	72.500000
75%	4.000000	32.000000	43.000000	10.0000	10.000000	10.00000	78.000000
max	5.000000	32.600000	44.000000	10.0000	10.000000	10.00000	92.000000

[42]:
```
file.sort_values('FAT')
```

[42]:

	Sl.No	StudnetName	CAT-1	CAT-2	DA-1	DA-2	DA-3	FAT
0	1	DeviPrasad	32.0	23.4	10.0	10	8.33	58.0
1	2	Dellip	25.0	33.0	7.0	9	10.00	67.0
3	4	Pandhu	17.0	43.0	9.0	9	8.00	72.5
4	5	Eswar	22.0	34.0	10.0	10	10.00	78.0
2	3	Gayathri	32.6	44.0	7.6	10	10.00	92.0

[43]:
```
file.head(5)
```

[43]:

	Sl.No	StudnetName	CAT-1	CAT-2	DA-1	DA-2	DA-3	FAT
0	1	DeviPrasad	32.0	23.4	10.0	10	8.33	58.0

```
1      2      Dellip   25.0   33.0   7.0     9  10.00  67.0
2      3    Gayathri   32.6   44.0   7.6    10  10.00  92.0
3      4      Pandhu   17.0   43.0   9.0     9   8.00  72.5
4      5       Eswar   22.0   34.0  10.0    10  10.00  78.0
```

[44]:
```
file.drop(columns=['Sl.No'])
```

[44]:
```
  StudnetName  CAT-1  CAT-2  DA-1  DA-2   DA-3   FAT
0  DeviPrasad   32.0   23.4  10.0    10   8.33  58.0
1      Dellip   25.0   33.0   7.0     9  10.00  67.0
2    Gayathri   32.6   44.0   7.6    10  10.00  92.0
3      Pandhu   17.0   43.0   9.0     9   8.00  72.5
4       Eswar   22.0   34.0  10.0    10  10.00  78.0
```

[ ]:
```

```

[45]:
```
file.to_csv('Modified.csv')
file2=pd.read_csv('Modified.csv')
#file.to_excel('Modified.xlsx' ,index=False)
#file.to_csv('Modified.txt', index=False, sep='\t')
file2.head(5)
```

[45]:
```
   Unnamed: 0  Sl.No StudnetName  CAT-1  CAT-2  DA-1  DA-2   DA-3   FAT
0           0      1  DeviPrasad   32.0   23.4  10.0    10   8.33  58.0
1           1      2      Dellip   25.0   33.0   7.0     9  10.00  67.0
2           2      3    Gayathri   32.6   44.0   7.6    10  10.00  92.0
3           3      4      Pandhu   17.0   43.0   9.0     9   8.00  72.5
4           4      5       Eswar   22.0   34.0  10.0    10  10.00  78.0
```

[46]:
```
file.to_csv('Modified.csv')
file2=pd.read_csv('Modified.csv')
#file.to_excel('Modified.xlsx' ,index=False)
#file.to_csv('Modified.txt', index=False, sep='\t')
file2.head(5)
```

[46]:
```
   Unnamed: 0  Sl.No StudnetName  CAT-1  CAT-2  DA-1  DA-2   DA-3   FAT
0           0      1  DeviPrasad   32.0   23.4  10.0    10   8.33  58.0
1           1      2      Dellip   25.0   33.0   7.0     9  10.00  67.0
2           2      3    Gayathri   32.6   44.0   7.6    10  10.00  92.0
3           3      4      Pandhu   17.0   43.0   9.0     9   8.00  72.5
4           4      5       Eswar   22.0   34.0  10.0    10  10.00  78.0
```

[47]:
```
file.to_csv('Modified.csv')
file2=pd.read_csv('Modified.csv')
#file.to_excel('Modified.xlsx' ,index=False)
#file.to_csv('Modified.txt', index=False, sep='\t')
file2.head(5)
```

[47]:

	Unnamed: 0	Sl.No	StudnetName	CAT-1	CAT-2	DA-1	DA-2	DA-3	FAT
0	0	1	DeviPrasad	32.0	23.4	10.0	10	8.33	58.0
1	1	2	Dellip	25.0	33.0	7.0	9	10.00	67.0
2	2	3	Gayathri	32.6	44.0	7.6	10	10.00	92.0
3	3	4	Pandhu	17.0	43.0	9.0	9	8.00	72.5
4	4	5	Eswar	22.0	34.0	10.0	10	10.00	78.0

[48]:
```
file.to_csv('Modified.csv')
file2=pd.read_csv('Modified.csv')
#file.to_excel('Modified.xlsx' ,index=False)
#file.to_csv('Modified.txt', index=False, sep='\t')
file2.head(5)
```

[48]:

	Unnamed: 0	Sl.No	StudnetName	CAT-1	CAT-2	DA-1	DA-2	DA-3	FAT
0	0	1	DeviPrasad	32.0	23.4	10.0	10	8.33	58.0
1	1	2	Dellip	25.0	33.0	7.0	9	10.00	67.0
2	2	3	Gayathri	32.6	44.0	7.6	10	10.00	92.0
3	3	4	Pandhu	17.0	43.0	9.0	9	8.00	72.5
4	4	5	Eswar	22.0	34.0	10.0	10	10.00	78.0

[49]:
```
from pandas import read_csv
path = r"student.csv"
data = read_csv(path)
print(data.shape)
print(data[:3])
```

```
(5, 8)
   Sl.No StudnetName  CAT-1  CAT-2  DA-1  DA-2   DA-3   FAT
0      1  DeviPrasad   32.0   23.4  10.0    10   8.33  58.0
1      2      Dellip   25.0   33.0   7.0     9  10.00  67.0
2      3    Gayathri   32.6   44.0   7.6    10  10.00  92.0
```

[50]:
```
from pandas import read_csv
path = r"pima-indians-diabetes.csv"
headernames = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi',␣
 ↪'age','class']
data = read_csv(path, names=headernames)
print(data.head(50))
```

```
    preg  plas  pres  skin  test  mass   pedi  age  class
0      6   148    72    35     0  33.6  0.627   50      1
1      1    85    66    29     0  26.6  0.351   31      0
2      8   183    64     0     0  23.3  0.672   32      1
3      1    89    66    23    94  28.1  0.167   21      0
4      0   137    40    35   168  43.1  2.288   33      1
5      5   116    74     0     0  25.6  0.201   30      0
6      3    78    50    32    88  31.0  0.248   26      1
7     10   115     0     0     0  35.3  0.134   29      0
```

8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1
10	4	110	92	0	0	37.6	0.191	30	0
11	10	168	74	0	0	38.0	0.537	34	1
12	10	139	80	0	0	27.1	1.441	57	0
13	1	189	60	23	846	30.1	0.398	59	1
14	5	166	72	19	175	25.8	0.587	51	1
15	7	100	0	0	0	30.0	0.484	32	1
16	0	118	84	47	230	45.8	0.551	31	1
17	7	107	74	0	0	29.6	0.254	31	1
18	1	103	30	38	83	43.3	0.183	33	0
19	1	115	70	30	96	34.6	0.529	32	1
20	3	126	88	41	235	39.3	0.704	27	0
21	8	99	84	0	0	35.4	0.388	50	0
22	7	196	90	0	0	39.8	0.451	41	1
23	9	119	80	35	0	29.0	0.263	29	1
24	11	143	94	33	146	36.6	0.254	51	1
25	10	125	70	26	115	31.1	0.205	41	1
26	7	147	76	0	0	39.4	0.257	43	1
27	1	97	66	15	140	23.2	0.487	22	0
28	13	145	82	19	110	22.2	0.245	57	0
29	5	117	92	0	0	34.1	0.337	38	0
30	5	109	75	26	0	36.0	0.546	60	0
31	3	158	76	36	245	31.6	0.851	28	1
32	3	88	58	11	54	24.8	0.267	22	0
33	6	92	92	0	0	19.9	0.188	28	0
34	10	122	78	31	0	27.6	0.512	45	0
35	4	103	60	33	192	24.0	0.966	33	0
36	11	138	76	0	0	33.2	0.420	35	0
37	9	102	76	37	0	32.9	0.665	46	1
38	2	90	68	42	0	38.2	0.503	27	1
39	4	111	72	47	207	37.1	1.390	56	1
40	3	180	64	25	70	34.0	0.271	26	0
41	7	133	84	0	0	40.2	0.696	37	0
42	7	106	92	18	0	22.7	0.235	48	0
43	9	171	110	24	240	45.4	0.721	54	1
44	7	159	64	0	0	27.4	0.294	40	0
45	0	180	66	39	0	42.0	1.893	25	1
46	1	146	56	0	0	29.7	0.564	29	0
47	2	71	70	27	0	28.0	0.586	22	0
48	7	103	66	32	0	39.1	0.344	31	1
49	7	105	0	0	0	0.0	0.305	24	0

[51]:
```
from pandas import read_csv
path = r"pima-indians-diabetes.csv"
data = read_csv(path)
print(data.shape)
```

```
(767, 9)
```

[52]:
```
from pandas import read_csv
path = r"pima-indians-diabetes.csv"
data = read_csv(path)
print(data.dtypes)
```

```
6          int64
148        int64
72         int64
35         int64
0          int64
33.6     float64
0.627    float64
50         int64
1          int64
dtype: object
```

[53]:
```
from pandas import read_csv
from pandas import set_option
path = r"pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
data = read_csv(path, names=names)
set_option('display.width', 100)
set_option('precision', 2)
print(data.shape)
print(data.describe())
```

```
(768, 9)
         preg    plas    pres    skin    test    mass    pedi     age   class
count  768.00  768.00  768.00  768.00  768.00  768.00  768.00  768.00  768.00
mean     3.85  120.89   69.11   20.54   79.80   31.99    0.47   33.24    0.35
std      3.37   31.97   19.36   15.95  115.24    7.88    0.33   11.76    0.48
min      0.00    0.00    0.00    0.00    0.00    0.00    0.08   21.00    0.00
25%      1.00   99.00   62.00    0.00    0.00   27.30    0.24   24.00    0.00
50%      3.00  117.00   72.00   23.00   30.50   32.00    0.37   29.00    0.00
75%      6.00  140.25   80.00   32.00  127.25   36.60    0.63   41.00    1.00
max     17.00  199.00  122.00   99.00  846.00   67.10    2.42   81.00    1.00
```

[54]:
```
from pandas import read_csv
path = r"pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
data = read_csv(path, names=names)
count_class = data.groupby('class').size()
print(count_class)
```

```
class
```

```
0    500
1    268
dtype: int64
```

[55]:
```
from pandas import read_csv
from pandas import set_option
path = r"pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
data = read_csv(path, names=names)
set_option('display.width', 100)
set_option('precision', 2)
correlations = data.corr(method='pearson')
print(correlations)
```

```
        preg  plas  pres  skin  test  mass  pedi   age  class
preg    1.00  0.13  0.14 -0.08 -0.07  0.02 -0.03  0.54   0.22
plas    0.13  1.00  0.15  0.06  0.33  0.22  0.14  0.26   0.47
pres    0.14  0.15  1.00  0.21  0.09  0.28  0.04  0.24   0.07
skin   -0.08  0.06  0.21  1.00  0.44  0.39  0.18 -0.11   0.07
test   -0.07  0.33  0.09  0.44  1.00  0.20  0.19 -0.04   0.13
mass    0.02  0.22  0.28  0.39  0.20  1.00  0.14  0.04   0.29
pedi   -0.03  0.14  0.04  0.18  0.19  0.14  1.00  0.03   0.17
age     0.54  0.26  0.24 -0.11 -0.04  0.04  0.03  1.00   0.24
class   0.22  0.47  0.07  0.07  0.13  0.29  0.17  0.24   1.00
```

[56]:
```
from pandas import read_csv
path = r"pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
data = read_csv(path, names=names)
print(data.skew())
```

```
preg     0.90
plas     0.17
pres    -1.84
skin     0.11
test     2.27
mass    -0.43
pedi     1.92
age      1.13
class    0.64
dtype: float64
```

16 Creating data frames

[58]:
```
df = {'ca': [35, 37, 38], 'tx': [23, 24, 26], 'md': [5,5,6]}
pop = pd.DataFrame(df)
print('population:\n', pop, '\n')
```

```
population:
    ca  tx  md
0  35  23   5
1  37  24   5
2  38  26   6
```

[59]:
```
pop = pd.DataFrame(df, index = [2010,2012,2014])
print('population:\n', pop, '\n')
```

```
population:
       ca  tx  md
2010  35  23   5
2012  37  24   5
2014  38  26   6
```

17 Positional indexing of data frames

[61]:
```
data = pd.DataFrame({"capital":["Amaravathi", "Chennai", "Kerala"],
                     "population":[32.7, 26.7, 15.3]},  # in millions
                    index=["AP", "TN", "KERALA"])
data
```

[61]:

	capital	population
AP	Amaravathi	32.7
TN	Chennai	26.7
KERALA	Kerala	15.3