```
from pandas import *
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib notebook
from scipy.stats import mode

import os
print(os.listdir("E:/input"))
['bike-sharing-dataset', 'pima-diabetes']
```

30

```
################################读入 diabetes.csv 文件到
diabetes###################################
diabetes = pd.read_csv('E:/input/pima-diabetes/diabetes.csv')
################################读入 diabetes.csv 文件到
diabetes###################################
```

No output

31

```
################################显示 diabetes 的前 5 行###################################
diabetes.head(5)
################################显示 diabetes 的前 5 行###################################
```

31

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
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

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Check for Summary Statistics

32

##################################显示 diabetes 的详细信息，包括样本数、均值、方差等等##################################

diabetes.describe()

##################################显示 diabetes 的详细信息，包括样本数、均值、方差等等##################################

32

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

##################################显示 diabetes 中 Outcome 列中每种值出现的次数

##################################

diabetes["Outcome"].value_counts()

##################################显示 diabetes 中 Outcome 列中每种值出现的次数

##################################

0 500

1 268

Name: Outcome, dtype: int64

Check missing values

34

```
diabetes.head()
```

34

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
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

35

```
##################################显示 diabetes 中所有列的标题名
##################################
diabetes.columns
```

```
################################显示 diabetes 中所有列的标题名
################################
```

35

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

36

```
################################显示 diabetes 中所有列的值里面为 0 的个数
################################
print((diabetes[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age']] == 0).sum())
################################显示 diabetes 中所有列的值里面为 0 的个数
################################
Pregnancies                 111
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
dtype: int64
```

37

```
################################将 diabetes 中所有列的值中为 0 的值替换为
NaN################################
diabetes.loc[:,['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI']] = diabetes[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin',
       'BMI']].replace(0, np.NaN)

################################将 diabetes 中所有列的值中为 0 的值替换为
NaN################################
diabetes.head()
```

37

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6.0	148.0	72.0	35.0	NaN	33	0.627	50	1

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
						.6			
1	1.0	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	8.0	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	NaN	137.0	40.0	35.0	168.0	43.1	2.288	33	1

38

##################################统计 diabetes 中所有列的值中为 null 的个数
###################################

```
diabetes.isnull().sum()
```

##################################统计 diabetes 中所有列的值中为 null 的个数
###################################

38

```
Pregnancies                 111
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
```

```
Age                         0
Outcome                     0
dtype: int64
```

Dealing with missing values

A. Drop rows having NaN

39

```
print("Size before dropping NaN rows",diabetes.shape,"\n")

################################将 diabetes 中包含 NaN 的行删掉
################################
nan_dropped = diabetes.dropna()

################################将 diabetes 中包含 NaN 的行删掉
################################
print(nan_dropped.isnull().sum())
print("\nSize after dropping NaN rows",nan_dropped.shape)
```

```
Size before dropping NaN rows (768, 9)

Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64

Size after dropping NaN rows (336, 9)
```

Project2、数据分析的一系列问题

40

```
vector = np.random.chisquare(1,500)
print(vector)
################################打印 vector 的均值##################################
print("Mean",np.mean(vector))
```

```
################################打印 vector 的均值###################################

print("SD",np.std(vector))
print("Range",max(vector)-min(vector))
[2.37790160e-03 4.94903951e-05 6.46828766e-01 1.32283727e-01
 2.28800215e-01 8.06225934e-01 2.67146169e-04 1.11378924e-01
 7.97062954e-01 8.89663700e-01 2.35863991e-01 1.00758560e+00
 1.92579421e+00 1.90843549e-01 1.40523131e+00 8.25794809e-01
 9.48739614e-02 6.19796205e-02 6.34424155e-03 2.46532919e-01
 5.26347411e-01 1.18722896e-01 2.02085434e+00 4.27865652e-01
 1.36179133e+00 1.39358778e-01 3.96333372e+00 1.95011587e+00
 6.88951450e-02 1.12314811e-01 5.12783599e+00 7.90283979e-01
 2.04495142e-02 2.24568002e+00 1.08157986e-01 3.38975536e-05
 2.76337249e-01 8.59476519e-02 6.77722695e-01 1.97194784e-01
 2.40451410e+00 3.51015977e-01 4.86116234e-01 2.23372339e+00
 1.74109052e-01 6.65692316e-06 4.47359583e-03 8.61652531e-02
 6.22560091e-03 6.73274991e+00 3.76331129e-01 1.13856776e-03
 1.54576159e-02 8.36454362e-01 5.47257784e-01 5.68271466e-01
 4.52913898e-01 1.70309378e-01 1.90711915e+00 1.63948919e-01
 1.12388909e+00 1.10627177e-01 4.90558320e-02 1.64202501e+00
 6.30588647e-01 1.00493323e+00 1.96593874e-04 1.94783378e+00
 8.43551120e-02 1.41212618e-01 8.15312695e-01 6.04784841e-01
 2.14301426e-01 1.94722419e-01 4.97310889e-01 1.21359563e+00
 6.86396661e-02 1.05296379e+00 2.96981344e+00 1.90212674e+00
 2.31027123e-02 1.36552531e-01 6.88541185e-01 7.42094410e-02
 1.01249333e+00 2.90206583e-01 1.02850072e-01 1.04645377e+00
 4.74865824e-01 2.45005803e-04 5.44798880e-02 1.53853919e-02
 7.90168169e-01 1.21178431e-02 2.85774669e-01 1.65538355e+00
 1.07965113e+00 1.91906782e+00 1.72688371e+00 1.75988305e+00
 4.70933174e-01 5.76560779e-01 2.59998955e-01 2.87258934e+00
 7.02377802e-01 9.42114975e-01 1.70236609e-02 2.83710929e+00
 3.75716750e+00 3.55809119e-02 1.18648540e-02 8.80217505e-02
 8.33554394e+00 4.17393749e-01 7.37596096e-01 3.52481401e-01
 1.60718214e-02 6.40427715e-01 1.05193848e-02 2.74838193e-01
 2.68250776e-01 8.22414638e-03 1.63269644e-02 1.35334023e-01
 1.80695300e-02 4.77506700e+00 1.61294725e-01 2.35231546e+00
 1.10141922e+00 2.77513675e-01 9.69641879e-01 2.92082391e+00
 1.59761993e+00 9.26813514e-01 3.00618602e-01 4.27712133e-01
 2.90371144e+00 2.34497869e-02 1.85462497e+00 1.57388651e+00
 1.64677735e+00 5.02453788e-01 3.97718794e-01 1.02191533e+00
 7.82549643e+00 1.03166297e+00 2.43534867e-01 1.16237786e-01
 2.71638344e+00 1.21998058e-02 5.93949585e-01 3.86101769e-01
 7.69992991e-02 2.20379086e-01 4.23049392e+00 3.33563127e-01
 2.97198775e-01 1.72572934e+00 1.67021244e+00 5.48724160e-01
```

2.14227439e-01 1.08588506e-02 2.50548255e+00 1.11990992e-02
8.42950191e-01 2.12088673e-04 1.67104048e-01 2.50662760e-01
1.74342715e+00 1.75717422e-02 9.71739447e-01 3.81077082e-01
1.18272638e+00 1.19243667e+00 5.68947128e-01 9.33037919e-02
8.78437298e-01 3.26519562e-01 8.60222337e-01 1.29431562e+00
2.19337034e+00 8.70830030e-01 1.59515618e+00 8.54300068e-04
3.46282799e-01 7.23943333e-05 7.34242795e-01 1.75842046e+00
5.82905030e+00 2.93957005e-01 7.47695904e-02 1.08094020e+00
7.19813022e-01 4.36998695e-01 3.22108987e-02 6.23736560e+00
4.10494975e-01 2.88119055e+00 2.23964765e-01 2.92707206e-05
1.09164697e+00 4.05071201e-03 1.64460301e-01 2.15421175e-02
2.38016387e-01 2.09642926e+00 5.92057219e-03 1.39379762e-01
2.90980795e+00 1.22018846e-02 8.57505382e-01 2.38157278e+00
5.37322174e-01 3.02114479e-01 3.26894316e-01 2.22054572e-01
2.80880700e+00 5.55721835e-02 1.66836298e+00 5.38966480e-01
5.50203170e+00 4.41695153e-01 1.79011772e+00 1.20815878e+00
2.75556517e-02 6.22983679e-01 3.12143942e-01 3.01638635e-01
3.65999508e-01 2.37269470e+00 2.18811014e-02 1.41047063e+00
2.58242104e+00 3.35882970e-02 2.26502333e+00 2.32996775e-02
4.07400122e-01 2.07454626e+00 3.23325728e-01 5.32419073e-02
2.60294003e-01 2.38374823e+00 6.12985618e-01 1.70942521e-02
7.07257486e-02 1.26661262e-01 4.37560045e-01 3.02546062e-01
7.61021043e-01 9.41971278e-01 1.65602668e+00 2.58986416e+00
5.90180102e-02 4.66583952e-02 3.79749679e-01 1.58156691e+00
8.51999568e+00 9.67048939e-01 8.32150174e-01 2.29011822e-02
1.74855485e+00 4.14926087e-01 1.82352657e+00 6.92839529e-01
3.95173042e-01 4.93651703e-01 2.26055370e-04 2.67983350e+00
5.76825318e-01 1.33734032e-01 2.59228433e-01 1.52841842e+00
1.01730736e-01 2.50487232e+00 3.88334420e+00 4.96768177e-02
1.17334818e+00 1.57574497e+00 4.65639904e-02 7.08764438e-01
1.13211482e+00 1.02291344e-01 1.72094561e+00 9.53048084e-01
5.88861022e+00 3.36048826e+00 6.80250712e+00 1.82579825e-03
8.91097044e-02 3.61904405e+00 1.31361280e+00 1.49871359e-01
9.68535964e-01 1.88949109e+00 6.78736997e-01 9.23548794e-01
7.85293610e-01 1.09311200e-01 1.81799609e-01 3.01298365e+00
8.66892152e-01 1.87512350e+00 2.08764021e-02 1.14593043e+00
3.58834157e-01 1.53755303e+00 6.85259897e-01 7.52440403e-01
4.80659575e-02 3.54155481e-01 8.57795332e-02 1.52755641e+00
1.23736035e+00 9.74002860e-01 3.49967696e+00 3.08833743e-01
8.52821138e-02 1.24184214e+00 4.17457853e-01 1.94818157e-01
3.84660000e+00 1.35409497e+00 9.68736524e-04 1.99963374e+00
1.03565191e+00 4.05878444e-01 8.20557476e-02 1.86630034e+00
7.33311854e-01 1.94871199e+00 2.33539877e-02 2.12130054e+00
3.04323694e+00 7.24755341e-01 6.97398573e-01 1.26438461e+00

1.64409896e-01 5.26009796e-02 1.16719788e+00 3.45244613e-01
6.14672172e-01 3.28896757e-01 3.16473113e-01 4.17812275e+00
2.24370079e-02 8.71360626e-01 3.76441776e-02 8.20722432e-01
6.69965470e-01 2.37137898e+00 1.18944057e-01 9.28480822e+00
9.01452923e-01 1.61540852e+00 4.87676595e-01 4.62620480e-03
1.34766423e-01 6.33514251e-02 6.40813307e+00 1.88626147e-01
3.98012005e+00 1.67755999e+00 9.29436402e-01 6.94290187e-02
4.51217736e-02 4.61307737e-04 1.21306640e-02 8.54943872e-02
4.05402353e-03 1.66826668e-02 1.41790572e-02 4.08940728e+00
7.42960706e-01 5.99728138e-01 8.98808417e-01 1.70682697e-01
3.53580924e-01 4.96003749e-02 2.93539995e+00 2.42203520e-03
2.33303616e+00 2.01371012e+00 9.66879404e-01 8.70853646e-02
6.45334575e-01 7.16524734e-01 1.41402768e-02 5.12961385e-01
2.17882373e-01 1.39441158e+00 7.17246208e-01 8.34297957e-01
5.93822743e-03 2.76649788e-01 6.79316803e-01 4.09675073e-01
3.44250209e-02 2.48331327e-01 7.93915647e-01 1.66628008e-01
3.86844661e-02 8.84515370e-01 9.93621792e-01 6.35411971e-01
3.94326739e-01 3.07964883e-01 5.59213898e-01 1.09856602e+00
9.99437172e-01 4.02472618e-01 2.98214456e+00 1.59292766e+00
2.21751232e+00 7.30618543e-01 6.48327268e-01 5.13154852e+00
1.10096430e+00 3.51775710e+00 6.17711035e+00 1.38090401e+00
9.50434889e-02 2.01060950e-01 1.86007589e-01 6.53446836e-01
9.38017548e-02 2.08716965e+00 8.95613984e-02 2.43164747e-01
3.27648055e-02 8.58474434e-01 8.44644292e-02 6.52178860e-01
1.70487640e+00 8.36077850e-01 6.33757034e-02 6.27768268e-05
4.28594603e+00 2.83141841e-01 8.20788125e-01 1.64369765e+00
3.99631267e-01 5.94759283e-01 3.56194814e-01 3.69837229e-01
5.84394202e-01 1.73608776e+00 3.07376653e-01 6.70633077e+00
3.58270972e-02 2.42650697e-02 3.67963023e+00 5.39121367e-02
3.61013277e-02 8.22086549e-01 2.66737083e+00 6.73433018e-01
1.13077160e-01 2.22731565e+00 6.81665618e-02 1.16016279e+00
6.36668487e-03 4.56414610e-01 3.89833560e-03 1.32082155e+00
7.56962602e-02 3.11698022e-02 2.63805793e+00 1.02462842e+00
2.59120577e-01 3.02986991e+00 2.76265863e+00 7.94190137e-02
1.09194090e+00 5.44720327e-02 1.77263443e-01 5.20879667e-01
2.87056244e+00 1.77500925e-01 1.66454339e-02 1.76174153e-01
3.84027347e-03 2.45620522e-01 5.40047820e-01 8.15131070e+00
3.93227915e-01 4.24389323e-04 3.60489868e-01 6.09029881e-04
2.73699847e-01 7.36690592e-05 2.48425217e-01 3.14281113e-01
9.17856416e+00 1.46289884e+00 2.90063470e-01 8.03041455e-01
1.20284956e+00 3.60870545e+00 1.80261958e+00 5.62397449e+00]

Mean 1.0460399866410892

SD 1.4848661656480682

Range 9.284801563672533

Reshape

41

```
vector.shape
```

41

```
(500,)
```

42

```
################################将一维向量 vector 转成大小为 500*1 的二维向量并赋值给
row_vector#################################
row_vector = vector.reshape(-1,1)
################################将一维向量 vector 转成大小为 500*1 的二维向量并赋值给
row_vector#################################
row_vector.shape
print(row_vector)
[[2.37790160e-03]
 [4.94903951e-05]
 [6.46828766e-01]
 [1.32283727e-01]
 [2.28800215e-01]
 [8.06225934e-01]
 [2.67146169e-04]
 [1.11378924e-01]
 [7.97062954e-01]
 [8.89663700e-01]
 [2.35863991e-01]
 [1.00758560e+00]
 [1.92579421e+00]
 [1.90843549e-01]
 [1.40523131e+00]
 [8.25794809e-01]
 [9.48739614e-02]
 [6.19796205e-02]
 [6.34424155e-03]
 [2.46532919e-01]
 [5.26347411e-01]
 [1.18722896e-01]
 [2.02085434e+00]
 [4.27865652e-01]
 [1.36179133e+00]
 [1.39358778e-01]
 [3.96333372e+00]
 [1.95011587e+00]
 [6.88951450e-02]
```

[1.12314811e-01]
[5.12783599e+00]
[7.90283979e-01]
[2.04495142e-02]
[2.24568002e+00]
[1.08157986e-01]
[3.38975536e-05]
[2.76337249e-01]
[8.59476519e-02]
[6.77722695e-01]
[1.97194784e-01]
[2.40451410e+00]
[3.51015977e-01]
[4.86116234e-01]
[2.23372339e+00]
[1.74109052e-01]
[6.65692316e-06]
[4.47359583e-03]
[8.61652531e-02]
[6.22560091e-03]
[6.73274991e+00]
[3.76331129e-01]
[1.13856776e-03]
[1.54576159e-02]
[8.36454362e-01]
[5.47257784e-01]
[5.68271466e-01]
[4.52913898e-01]
[1.70309378e-01]
[1.90711915e+00]
[1.63948919e-01]
[1.12388909e+00]
[1.10627177e-01]
[4.90558320e-02]
[1.64202501e+00]
[6.30588647e-01]
[1.00493323e+00]
[1.96593874e-04]
[1.94783378e+00]
[8.43551120e-02]
[1.41212618e-01]
[8.15312695e-01]
[6.04784841e-01]
[2.14301426e-01]

[1.94722419e-01]
[4.97310889e-01]
[1.21359563e+00]
[6.86396661e-02]
[1.05296379e+00]
[2.96981344e+00]
[1.90212674e+00]
[2.31027123e-02]
[1.36552531e-01]
[6.88541185e-01]
[7.42094410e-02]
[1.01249333e+00]
[2.90206583e-01]
[1.02850072e-01]
[1.04645377e+00]
[4.74865824e-01]
[2.45005803e-04]
[5.44798880e-02]
[1.53853919e-02]
[7.90168169e-01]
[1.21178431e-02]
[2.85774669e-01]
[1.65538355e+00]
[1.07965113e+00]
[1.91906782e+00]
[1.72688371e+00]
[1.75988305e+00]
[4.70933174e-01]
[5.76560779e-01]
[2.59998955e-01]
[2.87258934e+00]
[7.02377802e-01]
[9.42114975e-01]
[1.70236609e-02]
[2.83710929e+00]
[3.75716750e+00]
[3.55809119e-02]
[1.18648540e-02]
[8.80217505e-02]
[8.33554394e+00]
[4.17393749e-01]
[7.37596096e-01]
[3.52481401e-01]
[1.60718214e-02]

[6.40427715e-01]
[1.05193848e-02]
[2.74838193e-01]
[2.68250776e-01]
[8.22414638e-03]
[1.63269644e-02]
[1.35334023e-01]
[1.80695300e-02]
[4.77506700e+00]
[1.61294725e-01]
[2.35231546e+00]
[1.10141922e+00]
[2.77513675e-01]
[9.69641879e-01]
[2.92082391e+00]
[1.59761993e+00]
[9.26813514e-01]
[3.00618602e-01]
[4.27712133e-01]
[2.90371144e+00]
[2.34497869e-02]
[1.85462497e+00]
[1.57388651e+00]
[1.64677735e+00]
[5.02453788e-01]
[3.97718794e-01]
[1.02191533e+00]
[7.82549643e+00]
[1.03166297e+00]
[2.43534867e-01]
[1.16237786e-01]
[2.71638344e+00]
[1.21998058e-02]
[5.93949585e-01]
[3.86101769e-01]
[7.69992991e-02]
[2.20379086e-01]
[4.23049392e+00]
[3.33563127e-01]
[2.97198775e-01]
[1.72572934e+00]
[1.67021244e+00]
[5.48724160e-01]
[2.14227439e-01]

[1.08588506e-02]
[2.50548255e+00]
[1.11990992e-02]
[8.42950191e-01]
[2.12088673e-04]
[1.67104048e-01]
[2.50662760e-01]
[1.74342715e+00]
[1.75717422e-02]
[9.71739447e-01]
[3.81077082e-01]
[1.18272638e+00]
[1.19243667e+00]
[5.68947128e-01]
[9.33037919e-02]
[8.78437298e-01]
[3.26519562e-01]
[8.60222337e-01]
[1.29431562e+00]
[2.19337034e+00]
[8.70830030e-01]
[1.59515618e+00]
[8.54300068e-04]
[3.46282799e-01]
[7.23943333e-05]
[7.34242795e-01]
[1.75842046e+00]
[5.82905030e+00]
[2.93957005e-01]
[7.47695904e-02]
[1.08094020e+00]
[7.19813022e-01]
[4.36998695e-01]
[3.22108987e-02]
[6.23736560e+00]
[4.10494975e-01]
[2.88119055e+00]
[2.23964765e-01]
[2.92707206e-05]
[1.09164697e+00]
[4.05071201e-03]
[1.64460301e-01]
[2.15421175e-02]
[2.38016387e-01]

[2.09642926e+00]
[5.92057219e-03]
[1.39379762e-01]
[2.90980795e+00]
[1.22018846e-02]
[8.57505382e-01]
[2.38157278e+00]
[5.37322174e-01]
[3.02114479e-01]
[3.26894316e-01]
[2.22054572e-01]
[2.80880700e+00]
[5.55721835e-02]
[1.66836298e+00]
[5.38966480e-01]
[5.50203170e+00]
[4.41695153e-01]
[1.79011772e+00]
[1.20815878e+00]
[2.75556517e-02]
[6.22983679e-01]
[3.12143942e-01]
[3.01638635e-01]
[3.65999508e-01]
[2.37269470e+00]
[2.18811014e-02]
[1.41047063e+00]
[2.58242104e+00]
[3.35882970e-02]
[2.26502333e+00]
[2.32996775e-02]
[4.07400122e-01]
[2.07454626e+00]
[3.23325728e-01]
[5.32419073e-02]
[2.60294003e-01]
[2.38374823e+00]
[6.12985618e-01]
[1.70942521e-02]
[7.07257486e-02]
[1.26661262e-01]
[4.37560045e-01]
[3.02546062e-01]
[7.61021043e-01]

[9.41971278e-01]
[1.65602668e+00]
[2.58986416e+00]
[5.90180102e-02]
[4.66583952e-02]
[3.79749679e-01]
[1.58156691e+00]
[8.51999568e+00]
[9.67048939e-01]
[8.32150174e-01]
[2.29011822e-02]
[1.74855485e+00]
[4.14926087e-01]
[1.82352657e+00]
[6.92839529e-01]
[3.95173042e-01]
[4.93651703e-01]
[2.26055370e-04]
[2.67983350e+00]
[5.76825318e-01]
[1.33734032e-01]
[2.59228433e-01]
[1.52841842e+00]
[1.01730736e-01]
[2.50487232e+00]
[3.88334420e+00]
[4.96768177e-02]
[1.17334818e+00]
[1.57574497e+00]
[4.65639904e-02]
[7.08764438e-01]
[1.13211482e+00]
[1.02291344e-01]
[1.72094561e+00]
[9.53048084e-01]
[5.88861022e+00]
[3.36048826e+00]
[6.80250712e+00]
[1.82579825e-03]
[8.91097044e-02]
[3.61904405e+00]
[1.31361280e+00]
[1.49871359e-01]
[9.68535964e-01]

[1.88949109e+00]
[6.78736997e-01]
[9.23548794e-01]
[7.85293610e-01]
[1.09311200e-01]
[1.81799609e-01]
[3.01298365e+00]
[8.66892152e-01]
[1.87512350e+00]
[2.08764021e-02]
[1.14593043e+00]
[3.58834157e-01]
[1.53755303e+00]
[6.85259897e-01]
[7.52440403e-01]
[4.80659575e-02]
[3.54155481e-01]
[8.57795332e-02]
[1.52755641e+00]
[1.23736035e+00]
[9.74002860e-01]
[3.49967696e+00]
[3.08833743e-01]
[8.52821138e-02]
[1.24184214e+00]
[4.17457853e-01]
[1.94818157e-01]
[3.84660000e+00]
[1.35409497e+00]
[9.68736524e-04]
[1.99963374e+00]
[1.03565191e+00]
[4.05878444e-01]
[8.20557476e-02]
[1.86630034e+00]
[7.33311854e-01]
[1.94871199e+00]
[2.33539877e-02]
[2.12130054e+00]
[3.04323694e+00]
[7.24755341e-01]
[6.97398573e-01]
[1.26438461e+00]
[1.64409896e-01]

[5.26009796e-02]
[1.16719788e+00]
[3.45244613e-01]
[6.14672172e-01]
[3.28896757e-01]
[3.16473113e-01]
[4.17812275e+00]
[2.24370079e-02]
[8.71360626e-01]
[3.76441776e-02]
[8.20722432e-01]
[6.69965470e-01]
[2.37137898e+00]
[1.18944057e-01]
[9.28480822e+00]
[9.01452923e-01]
[1.61540852e+00]
[4.87676595e-01]
[4.62620480e-03]
[1.34766423e-01]
[6.33514251e-02]
[6.40813307e+00]
[1.88626147e-01]
[3.98012005e+00]
[1.67755999e+00]
[9.29436402e-01]
[6.94290187e-02]
[4.51217736e-02]
[4.61307737e-04]
[1.21306640e-02]
[8.54943872e-02]
[4.05402353e-03]
[1.66826668e-02]
[1.41790572e-02]
[4.08940728e+00]
[7.42960706e-01]
[5.99728138e-01]
[8.98808417e-01]
[1.70682697e-01]
[3.53580924e-01]
[4.96003749e-02]
[2.93539995e+00]
[2.42203520e-03]
[2.33303616e+00]

[2.01371012e+00]
[9.66879404e-01]
[8.70853646e-02]
[6.45334575e-01]
[7.16524734e-01]
[1.41402768e-02]
[5.12961385e-01]
[2.17882373e-01]
[1.39441158e+00]
[7.17246208e-01]
[8.34297957e-01]
[5.93822743e-03]
[2.76649788e-01]
[6.79316803e-01]
[4.09675073e-01]
[3.44250209e-02]
[2.48331327e-01]
[7.93915647e-01]
[1.66628008e-01]
[3.86844661e-02]
[8.84515370e-01]
[9.93621792e-01]
[6.35411971e-01]
[3.94326739e-01]
[3.07964883e-01]
[5.59213898e-01]
[1.09856602e+00]
[9.99437172e-01]
[4.02472618e-01]
[2.98214456e+00]
[1.59292766e+00]
[2.21751232e+00]
[7.30618543e-01]
[6.48327268e-01]
[5.13154852e+00]
[1.10096430e+00]
[3.51775710e+00]
[6.17711035e+00]
[1.38090401e+00]
[9.50434889e-02]
[2.01060950e-01]
[1.86007589e-01]
[6.53446836e-01]
[9.38017548e-02]

[2.08716965e+00]
[8.95613984e-02]
[2.43164747e-01]
[3.27648055e-02]
[8.58474434e-01]
[8.44644292e-02]
[6.52178860e-01]
[1.70487640e+00]
[8.36077850e-01]
[6.33757034e-02]
[6.27768268e-05]
[4.28594603e+00]
[2.83141841e-01]
[8.20788125e-01]
[1.64369765e+00]
[3.99631267e-01]
[5.94759283e-01]
[3.56194814e-01]
[3.69837229e-01]
[5.84394202e-01]
[1.73608776e+00]
[3.07376653e-01]
[6.70633077e+00]
[3.58270972e-02]
[2.42650697e-02]
[3.67963023e+00]
[5.39121367e-02]
[3.61013277e-02]
[8.22086549e-01]
[2.66737083e+00]
[6.73433018e-01]
[1.13077160e-01]
[2.22731565e+00]
[6.81665618e-02]
[1.16016279e+00]
[6.36668487e-03]
[4.56414610e-01]
[3.89833560e-03]
[1.32082155e+00]
[7.56962602e-02]
[3.11698022e-02]
[2.63805793e+00]
[1.02462842e+00]
[2.59120577e-01]

```
 [3.02986991e+00]
 [2.76265863e+00]
 [7.94190137e-02]
 [1.09194090e+00]
 [5.44720327e-02]
 [1.77263443e-01]
 [5.20879667e-01]
 [2.87056244e+00]
 [1.77500925e-01]
 [1.66454339e-02]
 [1.76174153e-01]
 [3.84027347e-03]
 [2.45620522e-01]
 [5.40047820e-01]
 [8.15131070e+00]
 [3.93227915e-01]
 [4.24389323e-04]
 [3.60489868e-01]
 [6.09029881e-04]
 [2.73699847e-01]
 [7.36690592e-05]
 [2.48425217e-01]
 [3.14281113e-01]
 [9.17856416e+00]
 [1.46289884e+00]
 [2.90063470e-01]
 [8.03041455e-01]
 [1.20284956e+00]
 [3.60870545e+00]
 [1.80261958e+00]
 [5.62397449e+00]]
```

43

```
col_vector = vector.reshape(1,-1)
col_vector.shape
```

43

```
(1, 500)
```

44

```
################################将一维向量 vector 转成大小为 10*50 的二维向量并赋值给
maxtrix##################################
matrix = vector.reshape(10,50)
################################将一维向量 vector 转成大小为 10*50 的二维向量并赋值给
maxtrix##################################
matrix.shape
print(matrix)
```

```
[[2.37790160e-03 4.94903951e-05 6.46828766e-01 1.32283727e-01
  2.28800215e-01 8.06225934e-01 2.67146169e-04 1.11378924e-01
  7.97062954e-01 8.89663700e-01 2.35863991e-01 1.00758560e+00
  1.92579421e+00 1.90843549e-01 1.40523131e+00 8.25794809e-01
  9.48739614e-02 6.19796205e-02 6.34424155e-03 2.46532919e-01
  5.26347411e-01 1.18722896e-01 2.02085434e+00 4.27865652e-01
  1.36179133e+00 1.39358778e-01 3.96333372e+00 1.95011587e+00
  6.88951450e-02 1.12314811e-01 5.12783599e+00 7.90283979e-01
  2.04495142e-02 2.24568002e+00 1.08157986e-01 3.38975536e-05
  2.76337249e-01 8.59476519e-02 6.77722695e-01 1.97194784e-01
  2.40451410e+00 3.51015977e-01 4.86116234e-01 2.23372339e+00
  1.74109052e-01 6.65692316e-06 4.47359583e-03 8.61652531e-02
  6.22560091e-03 6.73274991e+00]
 [3.76331129e-01 1.13856776e-03 1.54576159e-02 8.36454362e-01
  5.47257784e-01 5.68271466e-01 4.52913898e-01 1.70309378e-01
  1.90711915e+00 1.63948919e-01 1.12388909e+00 1.10627177e-01
  4.90558320e-02 1.64202501e+00 6.30588647e-01 1.00493323e+00
  1.96593874e-04 1.94783378e+00 8.43551120e-02 1.41212618e-01
  8.15312695e-01 6.04784841e-01 2.14301426e-01 1.94722419e-01
  4.97310889e-01 1.21359563e+00 6.86396661e-02 1.05296379e+00
  2.96981344e+00 1.90212674e+00 2.31027123e-02 1.36552531e-01
  6.88541185e-01 7.42094410e-02 1.01249333e+00 2.90206583e-01
  1.02850072e-01 1.04645377e+00 4.74865824e-01 2.45005803e-04
  5.44798880e-02 1.53853919e-02 7.90168169e-01 1.21178431e-02
  2.85774669e-01 1.65538355e+00 1.07965113e+00 1.91906782e+00
  1.72688371e+00 1.75988305e+00]
 [4.70933174e-01 5.76560779e-01 2.59998955e-01 2.87258934e+00
  7.02377802e-01 9.42114975e-01 1.70236609e-02 2.83710929e+00
  3.75716750e+00 3.55809119e-02 1.18648540e-02 8.80217505e-02
  8.33554394e+00 4.17393749e-01 7.37596096e-01 3.52481401e-01
  1.60718214e-02 6.40427715e-01 1.05193848e-02 2.74838193e-01
  2.68250776e-01 8.22414638e-03 1.63269644e-02 1.35334023e-01
  1.80695300e-02 4.77506700e+00 1.61294725e-01 2.35231546e+00
  1.10141922e+00 2.77513675e-01 9.69641879e-01 2.92082391e+00
  1.59761993e+00 9.26813514e-01 3.00618602e-01 4.27712133e-01
  2.90371144e+00 2.34497869e-02 1.85462497e+00 1.57388651e+00
  1.64677735e+00 5.02453788e-01 3.97718794e-01 1.02191533e+00
  7.82549643e+00 1.03166297e+00 2.43534867e-01 1.16237786e-01
  2.71638344e+00 1.21998058e-02]
 [5.93949585e-01 3.86101769e-01 7.69992991e-02 2.20379086e-01
  4.23049392e+00 3.33563127e-01 2.97198775e-01 1.72572934e+00
  1.67021244e+00 5.48724160e-01 2.14227439e-01 1.08588506e-02
  2.50548255e+00 1.11990992e-02 8.42950191e-01 2.12088673e-04
  1.67104048e-01 2.50662760e-01 1.74342715e+00 1.75717422e-02
```

9.71739447e-01 3.81077082e-01 1.18272638e+00 1.19243667e+00
5.68947128e-01 9.33037919e-02 8.78437298e-01 3.26519562e-01
8.60222337e-01 1.29431562e+00 2.19337034e+00 8.70830030e-01
1.59515618e+00 8.54300068e-04 3.46282799e-01 7.23943333e-05
7.34242795e-01 1.75842046e+00 5.82905030e+00 2.93957005e-01
7.47695904e-02 1.08094020e+00 7.19813022e-01 4.36998695e-01
3.22108987e-02 6.23736560e+00 4.10494975e-01 2.88119055e+00
2.23964765e-01 2.92707206e-05]
[1.09164697e+00 4.05071201e-03 1.64460301e-01 2.15421175e-02
2.38016387e-01 2.09642926e+00 5.92057219e-03 1.39379762e-01
2.90980795e+00 1.22018846e-02 8.57505382e-01 2.38157278e+00
5.37322174e-01 3.02114479e-01 3.26894316e-01 2.22054572e-01
2.80880700e+00 5.55721835e-02 1.66836298e+00 5.38966480e-01
5.50203170e+00 4.41695153e-01 1.79011772e+00 1.20815878e+00
2.75556517e-02 6.22983679e-01 3.12143942e-01 3.01638635e-01
3.65999508e-01 2.37269470e+00 2.18811014e-02 1.41047063e+00
2.58242104e+00 3.35882970e-02 2.26502333e+00 2.32996775e-02
4.07400122e-01 2.07454626e+00 3.23325728e-01 5.32419073e-02
2.60294003e-01 2.38374823e+00 6.12985618e-01 1.70942521e-02
7.07257486e-02 1.26661262e-01 4.37560045e-01 3.02546062e-01
7.61021043e-01 9.41971278e-01]
[1.65602668e+00 2.58986416e+00 5.90180102e-02 4.66583952e-02
3.79749679e-01 1.58156691e+00 8.51999568e+00 9.67048939e-01
8.32150174e-01 2.29011822e-02 1.74855485e+00 4.14926087e-01
1.82352657e+00 6.92839529e-01 3.95173042e-01 4.93651703e-01
2.26055370e-04 2.67983350e+00 5.76825318e-01 1.33734032e-01
2.59228433e-01 1.52841842e+00 1.01730736e-01 2.50487232e+00
3.88334420e+00 4.96768177e-02 1.17334818e+00 1.57574497e+00
4.65639904e-02 7.08764438e-01 1.13211482e+00 1.02291344e-01
1.72094561e+00 9.53048084e-01 5.88861022e+00 3.36048826e+00
6.80250712e+00 1.82579825e-03 8.91097044e-02 3.61904405e+00
1.31361280e+00 1.49871359e-01 9.68535964e-01 1.88949109e+00
6.78736997e-01 9.23548794e-01 7.85293610e-01 1.09311200e-01
1.81799609e-01 3.01298365e+00]
[8.66892152e-01 1.87512350e+00 2.08764021e-02 1.14593043e+00
3.58834157e-01 1.53755303e+00 6.85259897e-01 7.52440403e-01
4.80659575e-02 3.54155481e-01 8.57795332e-02 1.52755641e+00
1.23736035e+00 9.74002860e-01 3.49967696e+00 3.08833743e-01
8.52821138e-02 1.24184214e+00 4.17457853e-01 1.94818157e-01
3.84660000e+00 1.35409497e+00 9.68736524e-04 1.99963374e+00
1.03565191e+00 4.05878444e-01 8.20557476e-02 1.86630034e+00
7.33311854e-01 1.94871199e+00 2.33539877e-02 2.12130054e+00
3.04323694e+00 7.24755341e-01 6.97398573e-01 1.26438461e+00
1.64409896e-01 5.26009796e-02 1.16719788e+00 3.45244613e-01

```
 6.14672172e-01 3.28896757e-01 3.16473113e-01 4.17812275e+00
 2.24370079e-02 8.71360626e-01 3.76441776e-02 8.20722432e-01
 6.69965470e-01 2.37137898e+00]
[1.18944057e-01 9.28480822e+00 9.01452923e-01 1.61540852e+00
 4.87676595e-01 4.62620480e-03 1.34766423e-01 6.33514251e-02
 6.40813307e+00 1.88626147e-01 3.98012005e+00 1.67755999e+00
 9.29436402e-01 6.94290187e-02 4.51217736e-02 4.61307737e-04
 1.21306640e-02 8.54943872e-02 4.05402353e-03 1.66826668e-02
 1.41790572e-02 4.08940728e+00 7.42960706e-01 5.99728138e-01
 8.98808417e-01 1.70682697e-01 3.53580924e-01 4.96003749e-02
 2.93539995e+00 2.42203520e-03 2.33303616e+00 2.01371012e+00
 9.66879404e-01 8.70853646e-02 6.45334575e-01 7.16524734e-01
 1.41402768e-02 5.12961385e-01 2.17882373e-01 1.39441158e+00
 7.17246208e-01 8.34297957e-01 5.93822743e-03 2.76649788e-01
 6.79316803e-01 4.09675073e-01 3.44250209e-02 2.48331327e-01
 7.93915647e-01 1.66628008e-01]
[3.86844661e-02 8.84515370e-01 9.93621792e-01 6.35411971e-01
 3.94326739e-01 3.07964883e-01 5.59213898e-01 1.09856602e+00
 9.99437172e-01 4.02472618e-01 2.98214456e+00 1.59292766e+00
 2.21751232e+00 7.30618543e-01 6.48327268e-01 5.13154852e+00
 1.10096430e+00 3.51775710e+00 6.17711035e+00 1.38090401e+00
 9.50434889e-02 2.01060950e-01 1.86007589e-01 6.53446836e-01
 9.38017548e-02 2.08716965e+00 8.95613984e-02 2.43164747e-01
 3.27648055e-02 8.58474434e-01 8.44644292e-02 6.52178860e-01
 1.70487640e+00 8.36077850e-01 6.33757034e-02 6.27768268e-05
 4.28594603e+00 2.83141841e-01 8.20788125e-01 1.64369765e+00
 3.99631267e-01 5.94759283e-01 3.56194814e-01 3.69837229e-01
 5.84394202e-01 1.73608776e+00 3.07376653e-01 6.70633077e+00
 3.58270972e-02 2.42650697e-02]
[3.67963023e+00 5.39121367e-02 3.61013277e-02 8.22086549e-01
 2.66737083e+00 6.73433018e-01 1.13077160e-01 2.22731565e+00
 6.81665618e-02 1.16016279e+00 6.36668487e-03 4.56414610e-01
 3.89833560e-03 1.32082155e+00 7.56962602e-02 3.11698022e-02
 2.63805793e+00 1.02462842e+00 2.59120577e-01 3.02986991e+00
 2.76265863e+00 7.94190137e-02 1.09194090e+00 5.44720327e-02
 1.77263443e-01 5.20879667e-01 2.87056244e+00 1.77500925e-01
 1.66454339e-02 1.76174153e-01 3.84027347e-03 2.45620522e-01
 5.40047820e-01 8.15131070e+00 3.93227915e-01 4.24389323e-04
 3.60489868e-01 6.09029881e-04 2.73699847e-01 7.36690592e-05
 2.48425217e-01 3.14281113e-01 9.17856416e+00 1.46289884e+00
 2.90063470e-01 8.03041455e-01 1.20284956e+00 3.60870545e+00
 1.80261958e+00 5.62397449e+00]]
```

Pivot Table

45

```
#Determine pivot table
df = pd.DataFrame({"A": ["foo", "foo", "foo", "foo", "foo",
                         "bar", "bar", "bar", "bar"],
                   "B": ["one", "one", "one", "two", "two",
                         "one", "one", "two", "two"],
                   "C": ["small", "large", "large", "small",
                         "small", "large", "small", "small",
                         "large"],
                   "D": [1, 2, 2, 3, 3, 4, 5, 6, 7],
                   "E": [2, 4, 5, 5, 6, 6, 8, 9, 9]})
```

No output

46

```
df.head(9)
```

46

	A	B	C	D	E
0	foo	one	small	1	2
1	foo	one	large	2	4
2	foo	one	large	2	5
3	foo	two	small	3	5
4	foo	two	small	3	6
5	bar	one	large	4	6
6	bar	one	small	5	8
7	bar	two	small	6	9
8	bar	two	large	7	9

47

```
###################################想办法生成以下表格，行为 A、B，列为
C#################################
table = pivot_table(df, values='D', index=['A', 'B'],
                    columns=['C'], aggfunc=np.sum)
```

####################################想办法生成以下表格，行为 A、B，列为 C###################################

table

47

	C	large	small
A	B		
bar	one	4.0	5.0
	two	7.0	6.0
foo	one	4.0	1.0
	two	NaN	6.0

Merging dataframes

48

```
df1 = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],
                    'B': ['B0', 'B1', 'B2', 'B3'],
                    'C': ['C0', 'C1', 'C2', 'C3'],
                    'D': ['D0', 'D1', 'D2', 'D3']},
                   index=[0, 1, 2, 3])


df2 = pd.DataFrame({'A': ['A4', 'A5', 'A6', 'A7'],
                    'B': ['B4', 'B5', 'B6', 'B7'],
                    'C': ['C4', 'C5', 'C6', 'C7'],
                    'D': ['D4', 'D5', 'D6', 'D7']},
                   index=[4, 5, 6, 7])


df3 = pd.DataFrame({'A': ['A8', 'A9', 'A10', 'A11'],
                    'B': ['B8', 'B9', 'B10', 'B11'],
                    'C': ['C8', 'C9', 'C10', 'C11'],
                    'D': ['D8', 'D9', 'D10', 'D11']},
                   index=[8, 9, 10, 11])
```

```
###################################将 df1,df2,df3 按行拼接到一起，并赋给
result###################################
frames = [df1, df2, df3]

result = pd.concat(frames)

###################################将 df1,df2,df3 按行拼接到一起，并赋给
result###################################

print(result)
```

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11