

```

In [2]: # INTRODUCTION
# NumPy参考《利用Python进行数据分析》
import numpy as np
# 创建一个ndarray类的对象，是一个二维NumPy数组，也叫数组。
x = np.array([[1, 2, 3], [4, 5, 6]])
print("\nx:\n{}".format(x))

# SciPy参考《Scipy Lecture Notes》(http://scipy-lectures.org/)
from scipy import sparse
# 创建一个二维NumPy数组，对角线全为1，其余为0，
# 稀疏矩阵（sparse matrix）的稠密表示（Dense Representation）
eye = np.eye(4)
print('\nNumPy array:\n{}'.format(eye))

# SciPy的稀疏矩阵有好几种形式
# CSR格式比COO格式保存要多30%的内存占用率。
# COO格式一经定义后shape就不可再修改，但是data, row, col还可以修改。
# 转换稀疏数组的表示方式为SciPy的CSR格式
sparse_csr_matrix = sparse.csr_matrix(eye)
print("\nSciPy sparse CSR matrix:\n{}".format(sparse_csr_matrix))
# 转换稀疏数组的表示方式为SciPy的COO格式
sparse_coo_matrix = sparse.coo_matrix(eye)
print('\nSciPy sparse COO matrix:\n{}'.format(sparse_coo_matrix))
print('-----')
# 输出一个全1的4*1向量
data = np.ones(4)
print('\nRow Vector:\n{}'.format(data))

# 输出一个全1的4*4矩阵
data = np.ones((4,4))
print('\nNumPy array:\n{}'.format(data))

# 输出一个全1的4*1向量，将之转化为COO格式的稀疏矩阵
data = np.ones(4)
row_indices = np.arange(4)
col_indices = np.arange(4)
eye_coo = sparse.coo_matrix((data, (row_indices, col_indices)))
print("\nCOO representation:\n{}".format(eye_coo))

# matplotlib
import matplotlib.pyplot as plt
# 生成一个序列，序列范围是-10到10，总共100个数
x = np.linspace(-10,10,100)
#使用sin函数生成第二个序列
y = np.sin(x)
#用 plot函数生成曲线图像
plt.plot(x, y, marker="x")
plt.show()

# pandas
import pandas as pd
# 生成一个简单的人群数据集
data = {'Name' : ["John", "Anna", "Peter", "Linda"],
        'Location' : ["New York", "Paris", "Berlin", "London"],
        'Age' : [24, 13, 53, 33]}
data_pandas = pd.DataFrame(data)
print('-----')

```

```

print(data_pandas)
# 输出年龄大于30的数据
print('-----')
print(data_pandas[data_pandas.Age > 30])
print('-----')
# 在jupyter notebook中运行，画出图表
display(data_pandas)
print('-----')
display(data_pandas[data_pandas.Age > 30])

```

```

x:
[[1 2 3]
 [4 5 6]]

```

```

NumPy array:
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]

```

```

SciPy sparse CSR matrix:
(0, 0)      1.0
(1, 1)      1.0
(2, 2)      1.0
(3, 3)      1.0

```

```

SciPy sparse COO matrix:
(0, 0)      1.0
(1, 1)      1.0
(2, 2)      1.0
(3, 3)      1.0

```

```

Row Vector:
[1. 1. 1. 1.]

```

```

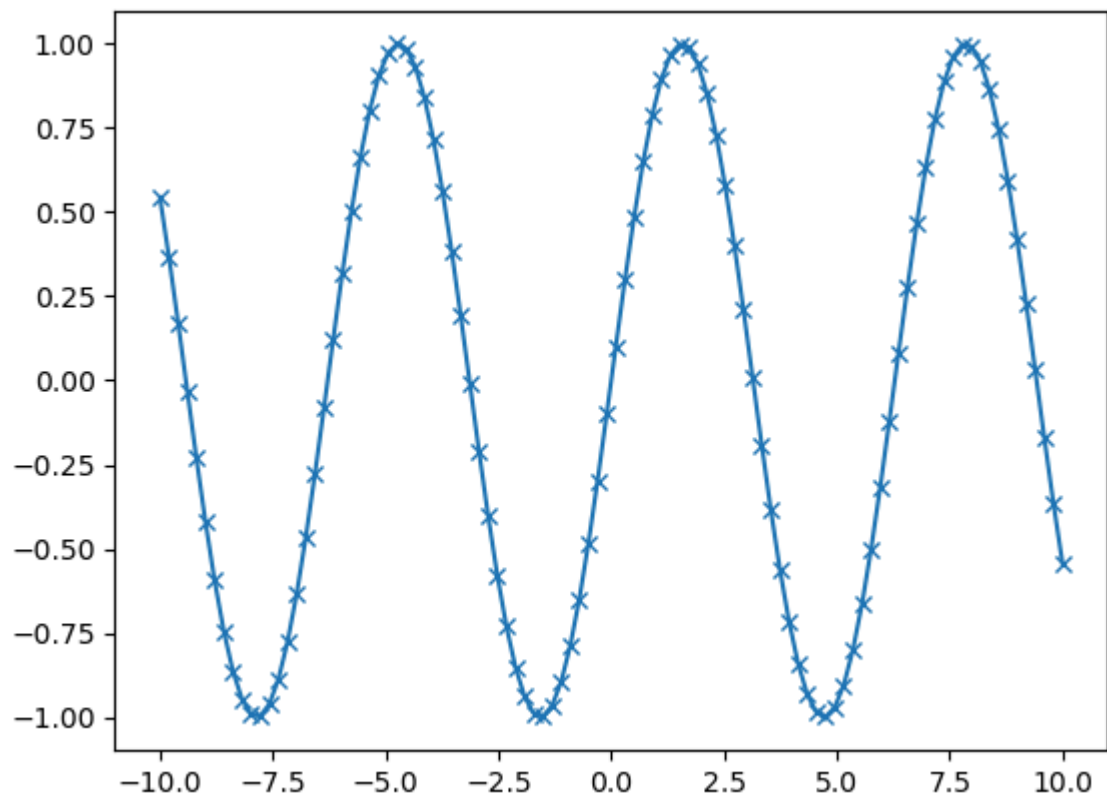
NumPy array:
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]

```

```

COO representation:
(0, 0)      1.0
(1, 1)      1.0
(2, 2)      1.0
(3, 3)      1.0

```



	Name	Location	Age
0	John	New York	24
1	Anna	Paris	13
2	Peter	Berlin	53
3	Linda	London	33

	Name	Location	Age
2	Peter	Berlin	53
3	Linda	London	33

	Name	Location	Age
0	John	New York	24
1	Anna	Paris	13
2	Peter	Berlin	53
3	Linda	London	33

	Name	Location	Age
2	Peter	Berlin	53
3	Linda	London	33