FSE598 前沿计算技术

模块 2 数据与数据处理
单元 5 文件操作与大数据处理
第 2 讲 大数据处理

讲座的英文版内容基于本书：
Y. Chen 《编程语言入门：C、C++、Scheme、Prolog、C# 和 Python 编程》(Introduction to Programming Languages: Programming in C, C++, Scheme, Prolog, C#, and Python)，第 6 版，Kendall Hunt Publishing Company，2019 年。
https://www.public.asu.edu/~ychen10/book/IntroPl.html

本讲大纲

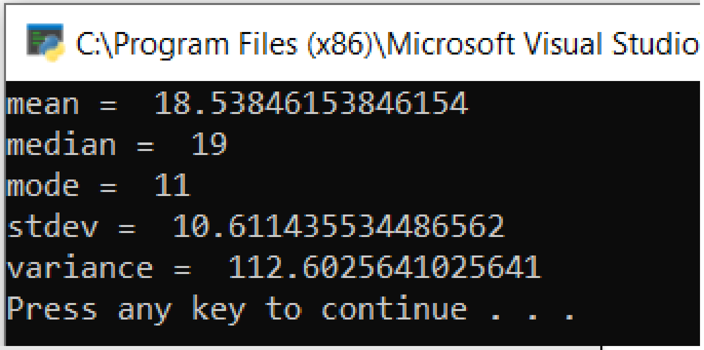
学习

- ❑ 基本统计库函数
- ❑ Python大数据处理和 AI 库
- ❑ NumPy数组和矩阵处理
- ❑ 将CSV文件加载到数组中进行处理

基本统计库函数

- 我们在此程序中使用“列表”。
- 列表是一种动态数据结构，它使用“对象”来存储列表的每个元素，因此，它可以灵活地存储不同类型的数据：int、float、str 等。
- 灵活性不得不以牺牲效率为代价。如果列表扩展到包含数百万元素的“大数据”，程序执行将非常缓慢。
- 我们如何有效地处理大数据？

```
import statistics
data = [11, 21, 11, 19, 46, 21, 19, 29, 21, 18, 3, 11, 11]

def main():
    a = statistics.mean(data)
    print("mean = ", a)
    b = statistics.median(data)
    print("median = ", b)
    c = statistics.mode(data)
    print("mode = ", c)
    d = statistics.stdev(data)
    print("stdev = ", d)
    e = statistics.variance(data)
    print("variance = ", e)
main()
```

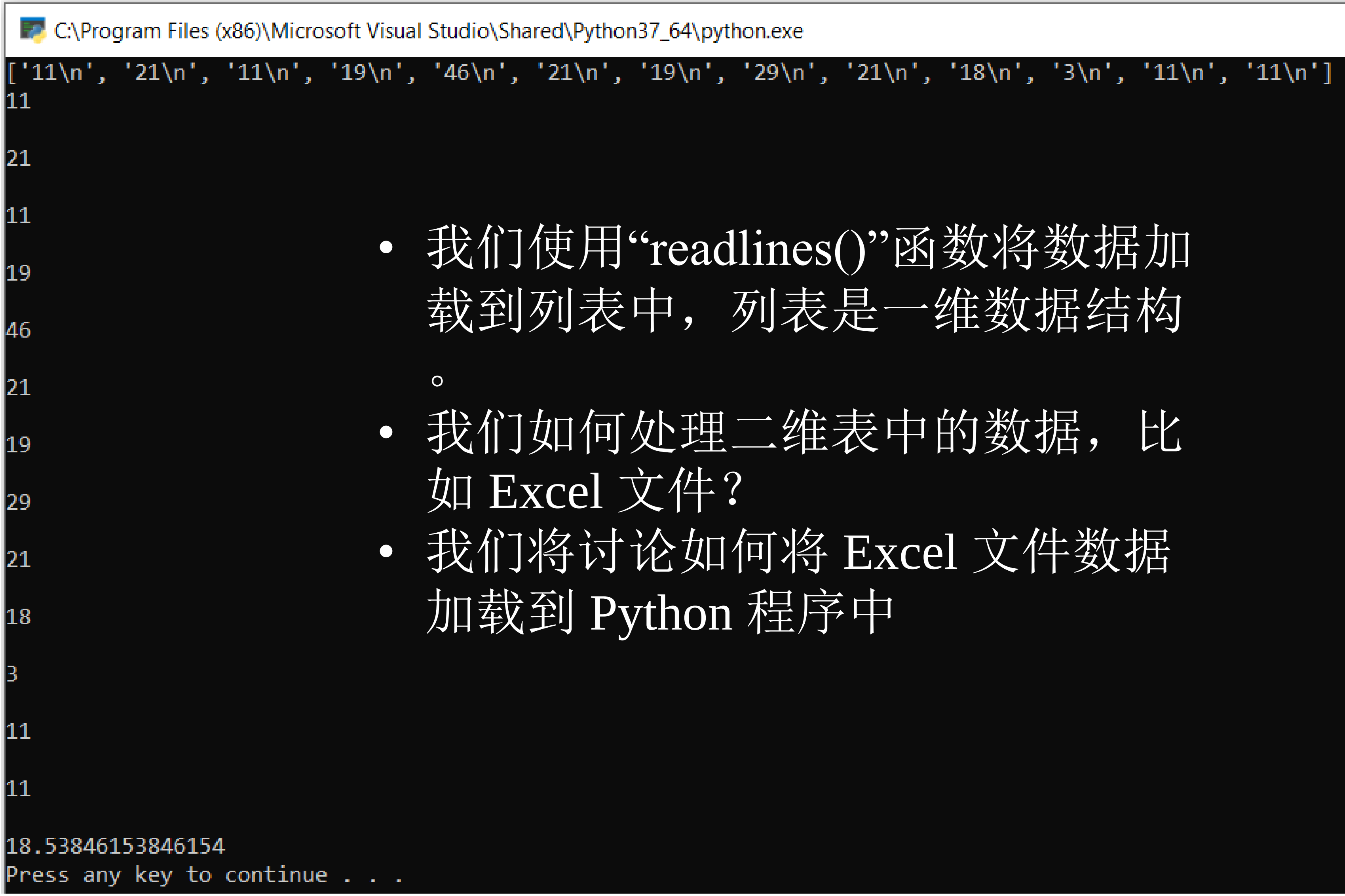


```
mean =  18.53846153846154
median =  19
mode =  11
stdev =  10.611435534486562
variance =  112.6025641025641
Press any key to continue . . .
```

回顾：从文件中读取数据

打开文件

```
data = []
sum = 0
n = 0
with open('mydata.txt') as f:
    data = f.readlines()
    print(data)
    for d in data:
        if (d):
            sum = sum + int(d)
            n += 1
            print(d)
f.close()
average = sum/n
print (average)
```



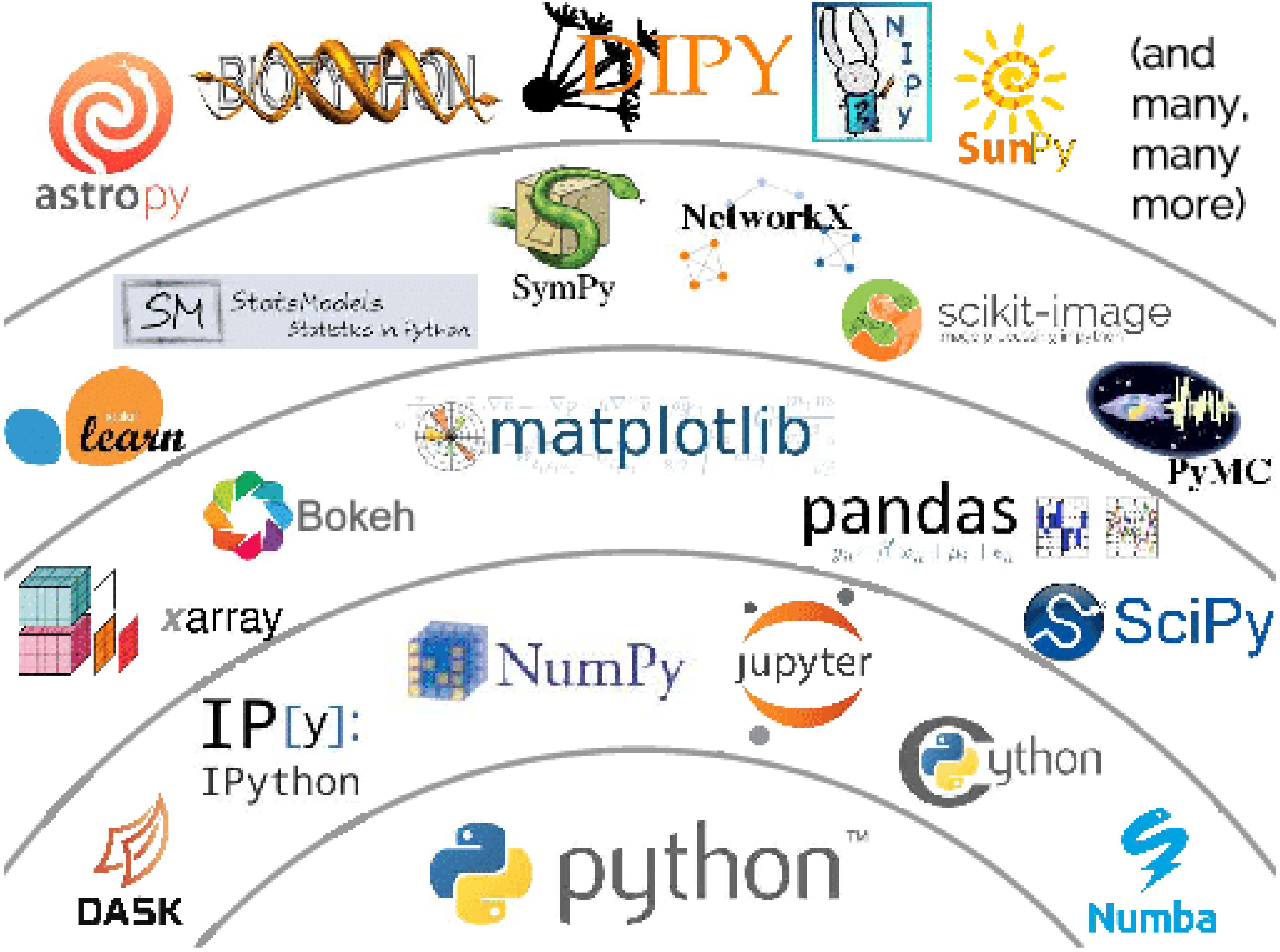
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe

```
['11\n', '21\n', '11\n', '19\n', '46\n', '21\n', '19\n', '29\n', '21\n', '18\n', '3\n', '11\n', '11\n']
11

21

11

19

46

21

19

29

21

18

3

11

11

18.538461538461554
Press any key to continue . . .
```

- 我们使用“readlines()”函数将数据加载到列表中，列表是一维数据结构。
- 我们如何处理二维表中的数据，比如 Excel 文件？
- 我们将讨论如何将 Excel 文件数据加载到 Python 程序中

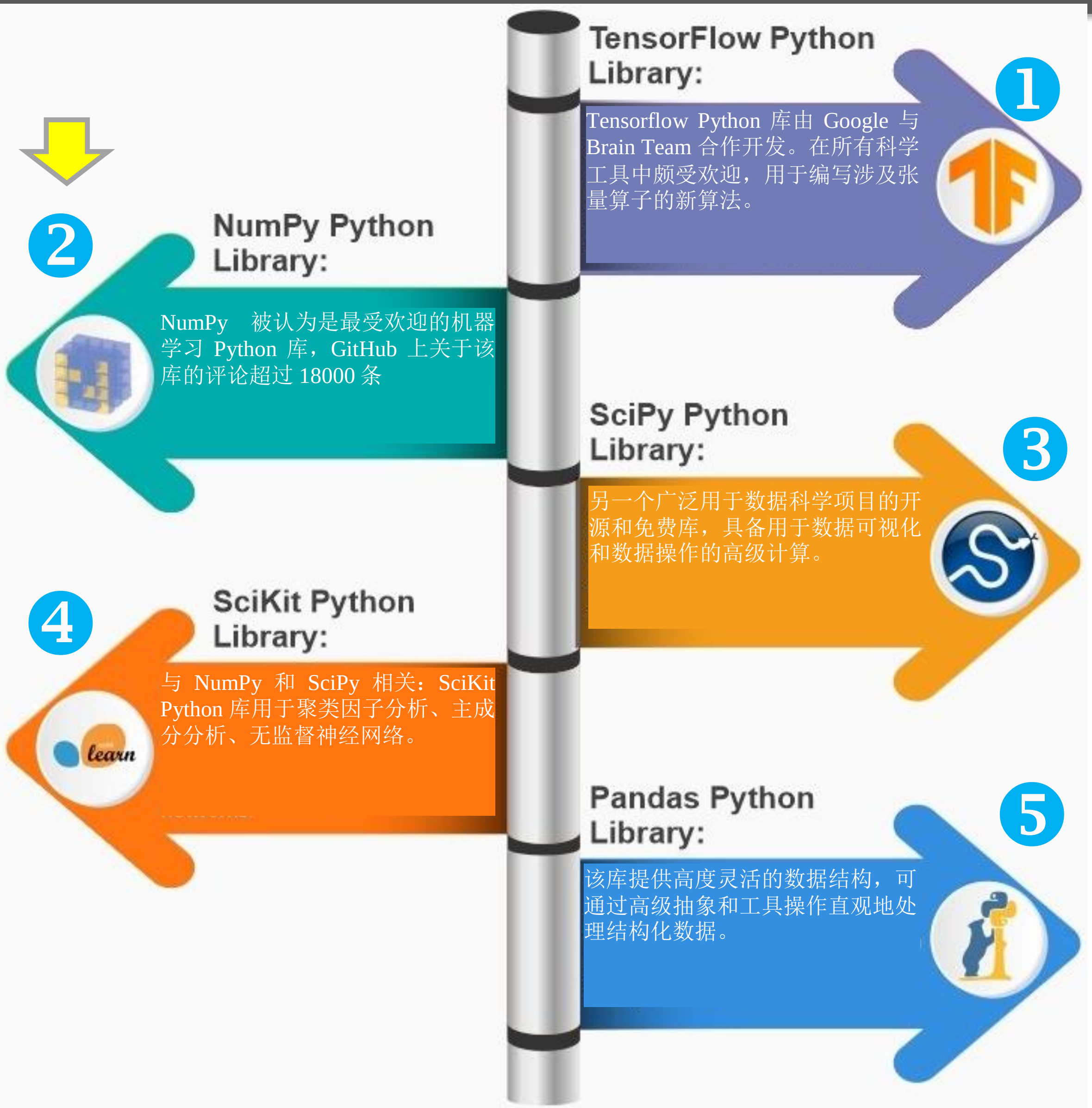
Python 大数据处理和人工智能库概述

用于科学计算的 Python 库



资料来源：https://www.researchgate.net/publication/332799309_The_Scikit-HEP_Project

用于人工智能和数据科学的 10 大 Python 库



1 TensorFlow Python Library:

Tensorflow Python 库由 Google 与 Brain Team 合作开发。在所有科学工具中颇受欢迎，用于编写涉及张量算子的新算法。

2 NumPy Python Library:

NumPy 被认为是最受欢迎的机器学习 Python 库，GitHub 上关于该库的评论超过 18000 条

3 SciPy Python Library:

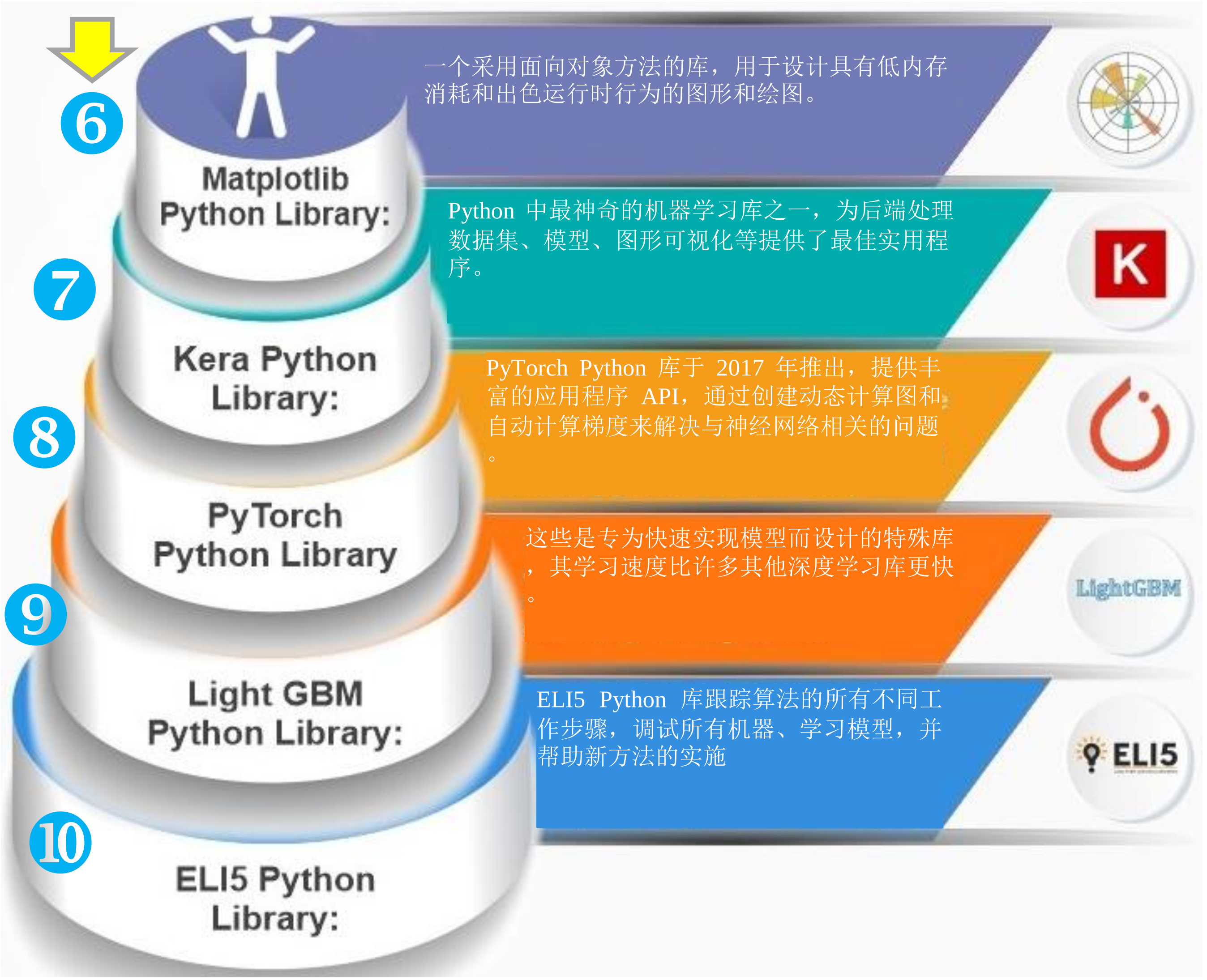
另一个广泛用于数据科学项目的开源和免费库，具备用于数据可视化和数据操作的高级计算。

4 SciKit Python Library:

与 NumPy 和 SciPy 相关：SciKit Python 库用于聚类因子分析、主成分分析、无监督神经网络。

5 Pandas Python Library:

该库提供高度灵活的数据结构，可通过高级抽象和工具操作直观地处理结构化数据。



6 Matplotlib Python Library:

一个采用面向对象方法的库，用于设计具有低内存消耗和出色运行时行为的图形和绘图。

7 Kera Python Library:

Python 中最神奇的机器学习库之一，为后端处理数据集、模型、图形可视化等提供了最佳实用程序。

8 PyTorch Python Library

PyTorch Python 库于 2017 年推出，提供丰富的应用程序 API，通过创建动态计算图和自动计算梯度来解决与神经网络相关的问题。

9 Light GBM Python Library:

这些是专为快速实现模型而设计的特殊库，其学习速度比许多其他深度学习库更快。

10 ELI5 Python Library:

ELI5 Python 库跟踪算法的所有不同工作步骤，调试所有机器、学习模型，并帮助新方法的实施

资料来源：https://www.janbasktraining.com/blog/python-libraries/

数据科学家需要了解的顶级 Python 库

01 Pandas

02 NumPy

03 SciPy

04 Scrapy

05 Matplotlib

06 Seaborn

07 Scikit-Learn

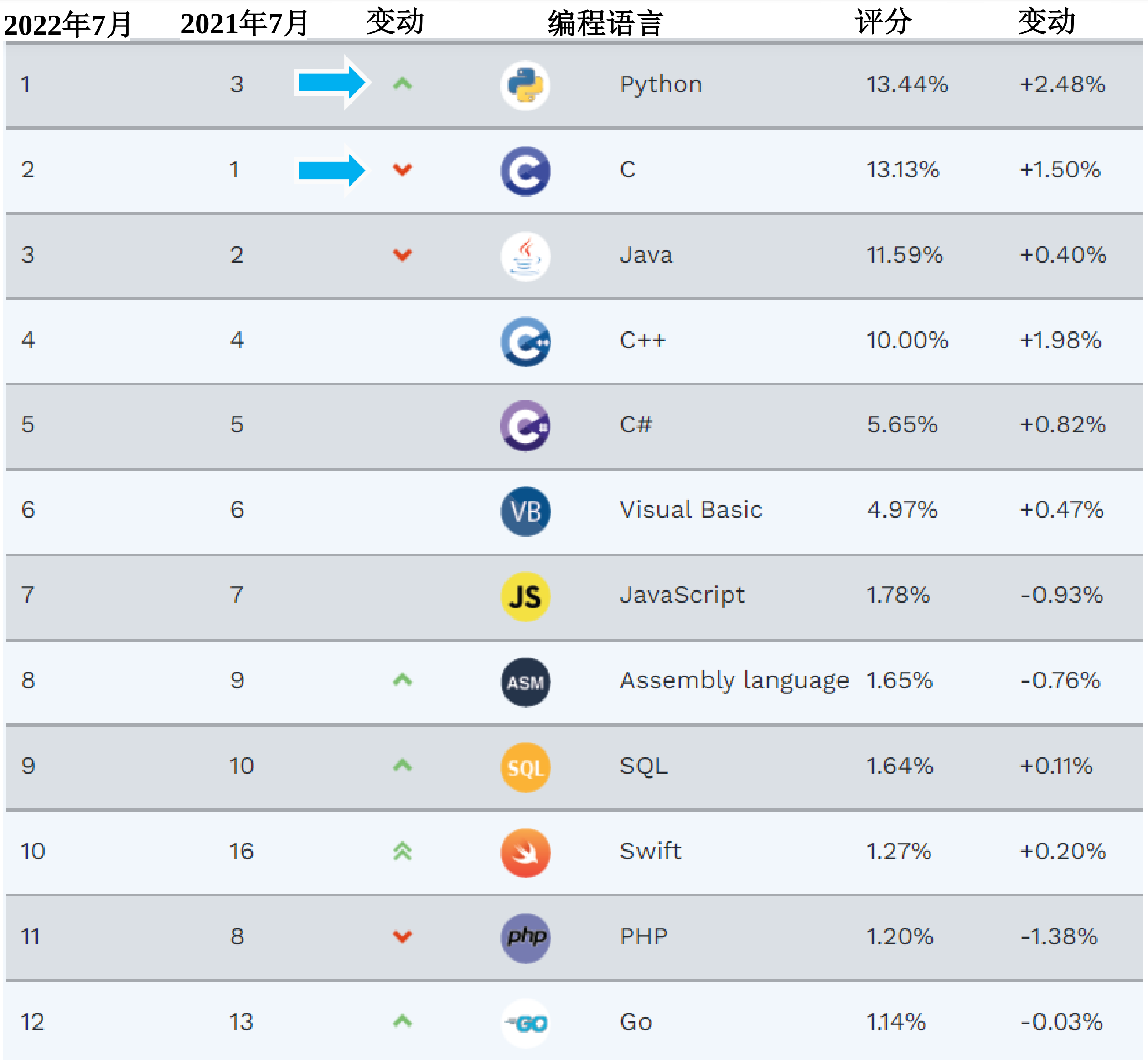
08 Tensor Flow

09 Scikit-Image

10 Librosa

资料来源：https://techvidvan.com/tutorials/python-libraries-for-data-scientist/

C vs. Python NumPy



2022年7月	2021年7月	变动	编程语言	评分	变动
1	3	➡ ↑	Python	13.44%	+2.48%
2	1	➡ ↓	C	13.13%	+1.50%
3	2	↓	Java	11.59%	+0.40%
4	4		C++	10.00%	+1.98%
5	5		C#	5.65%	+0.82%
6	6		Visual Basic	4.97%	+0.47%
7	7		JavaScript	1.78%	-0.93%
8	9	↑	Assembly language	1.65%	-0.76%
9	10	↑	SQL	1.64%	+0.11%
10	16	⇈	Swift	1.27%	+0.20%
11	8	↓	PHP	1.20%	-1.38%
12	13	↑	Go	1.14%	-0.03%

- C 是一种接近硬件的低级编程语言
- C 的可理解性和编程都很难
- C 在数据处理方面效率很高

- Python是一种更接近人类的高级编程语言
- Python 更易于理解和编程
- Python 和 Python List 在大数据处理方面效率不高

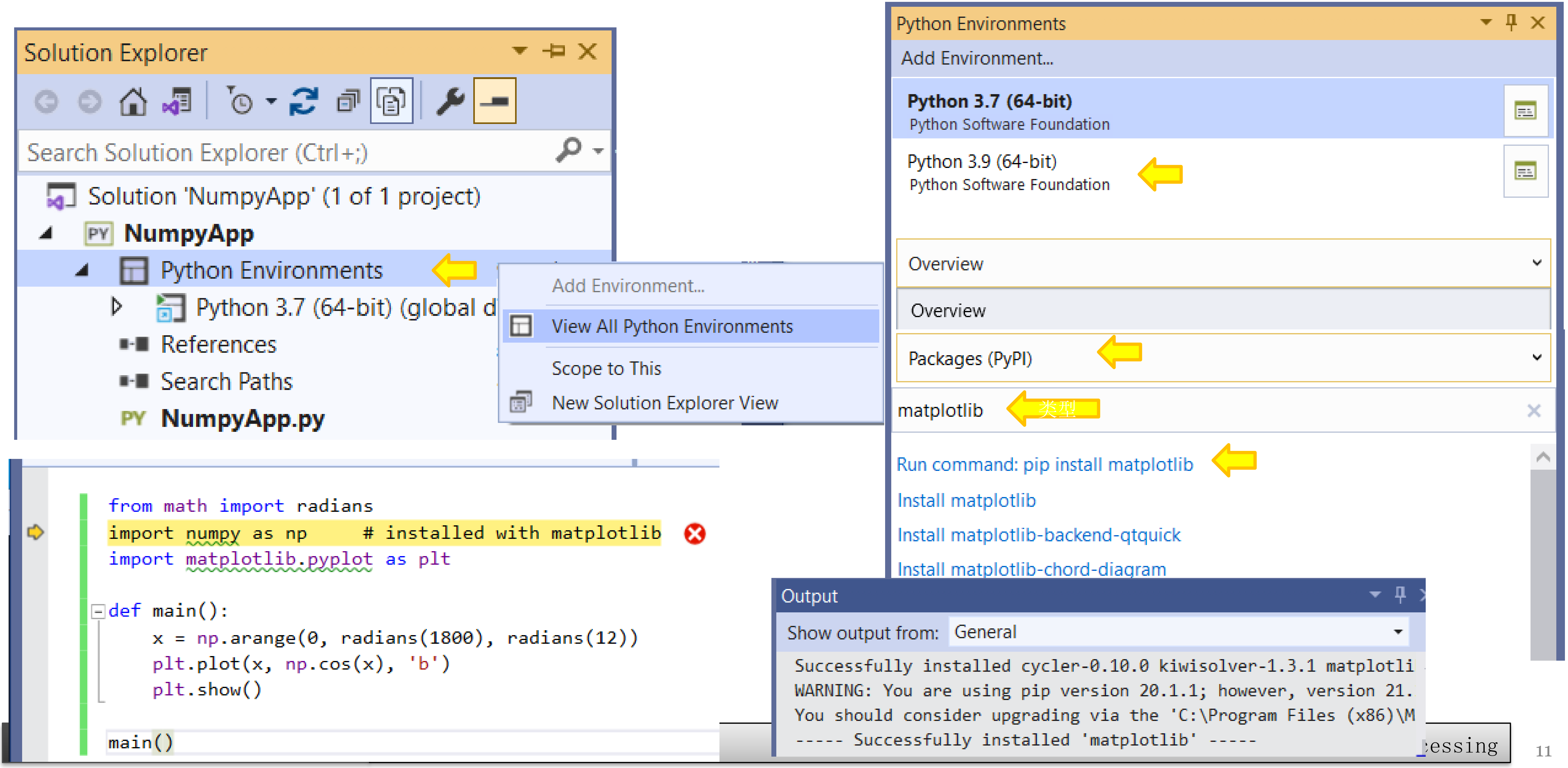
- NumPy 使用 C- 风格数组并使用 C 实现 NumPy 库
- 因此，NumPy 易于使用且高效

资料来源： http://www.tiobe.com/tiobe-index/

安装 Python 库

Python 拥有大量库。并不是所有库都会预先安装

创建新 Python 项目


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'NumpyApp' (1 of 1 project)
NumpyApp
Python Environments
Python 3.7 (64-bit) (global d
References
Search Paths
NumpyApp.py
Add Environment...
View All Python Environments
Scope to This
New Solution Explorer View
Python Environments
Add Environment...
Python 3.7 (64-bit)
Python Software Foundation
Python 3.9 (64-bit)
Python Software Foundation
Overview
Overview
Packages (PyPI)
类型
Run command: pip install matplotlib
Install matplotlib
Install matplotlib-backend-qtquick
Install matplotlib-chord-diagram
Output

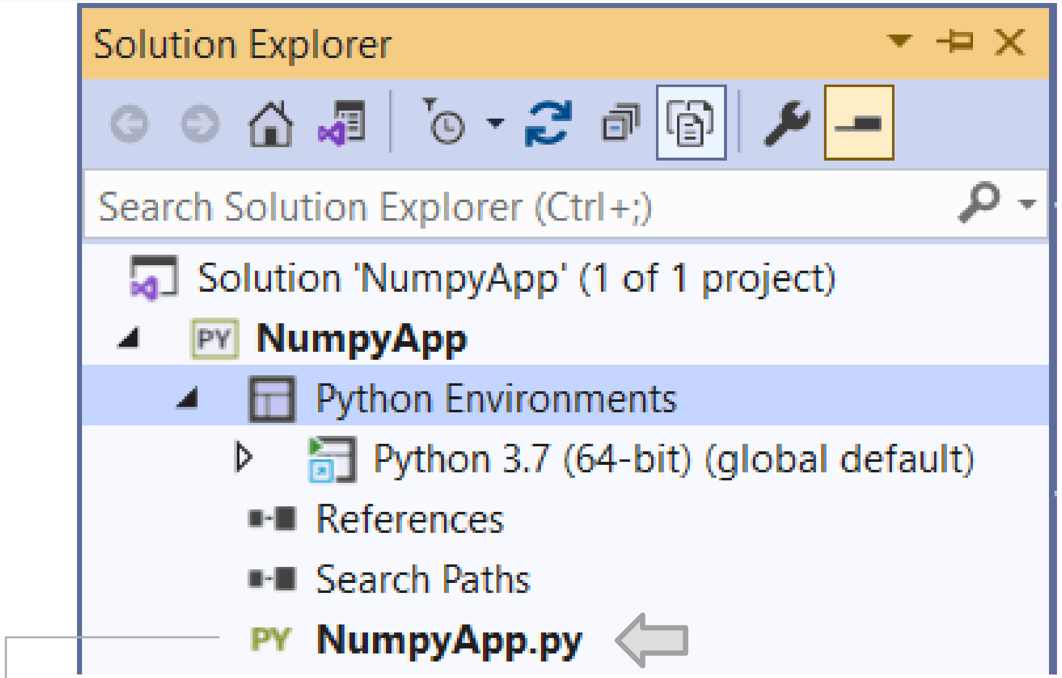
matplotlib

```
from math import radians
import numpy as np      # installed with matplotlib
import matplotlib.pyplot as plt

def main():
    x = np.arange(0, radians(1800), radians(12))
    plt.plot(x, np.cos(x), 'b')
    plt.show()

main()
```

Show output from: General

```
Successfully installed cycler-0.10.0 kiwisolver-1.3.1 matplotli
WARNING: You are using pip version 20.1.1; however, version 21.
You should consider upgrading via the 'C:\Program Files (x86)\M
----- Successfully installed 'matplotlib' -----
```

安装 matpolitlib 后



Solution Explorer

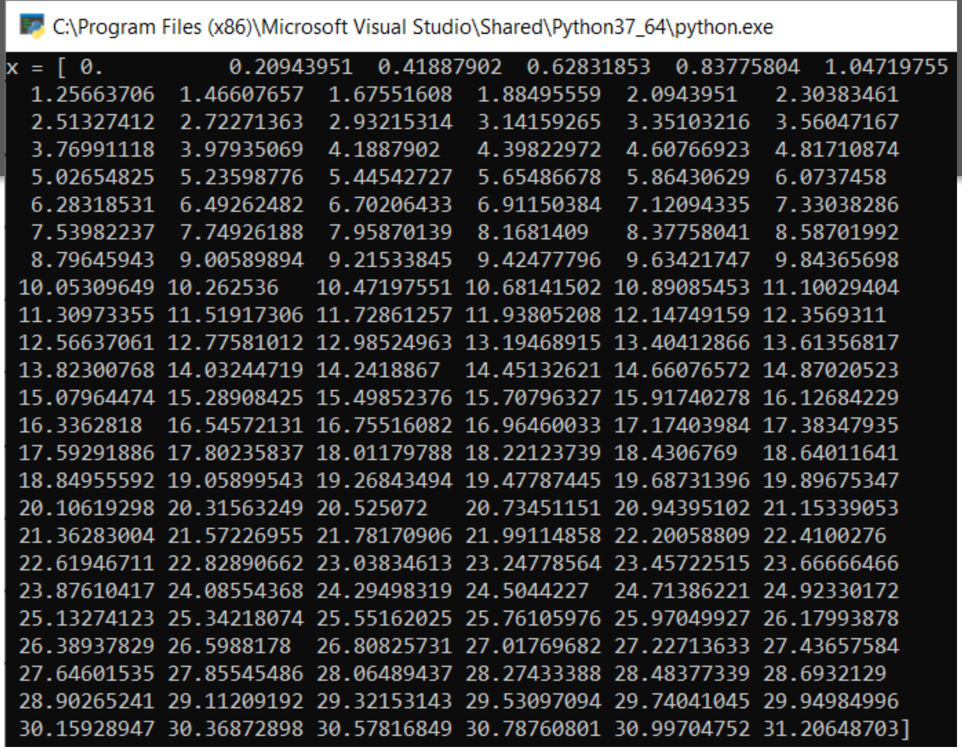
Search Solution Explorer (Ctrl+;)

Solution 'NumpyApp' (1 of 1 project)

- NumpyApp
  - Python Environments
    - Python 3.7 (64-bit) (global default)
  - References
  - Search Paths
  - NumpyApp.py

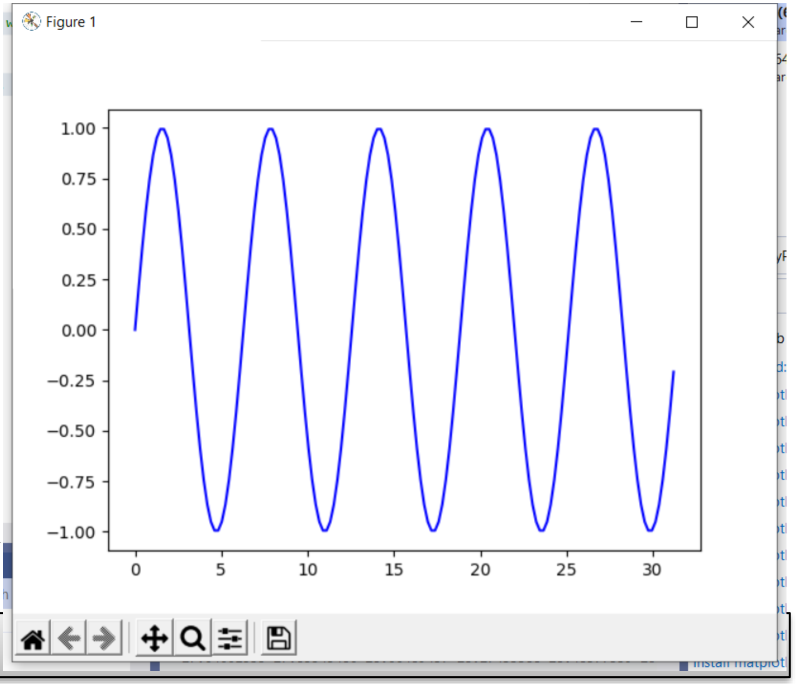
```
from math import radians
import numpy as np      # installed with matplotlib
import matplotlib.pyplot as plt
def main():
    x = np.arange(0, radians(1800), radians(12))
    print(F'x = {x}')
    plt.plot(x, np.sin(x), 'b')
    plt.show()
main()
```

弧度



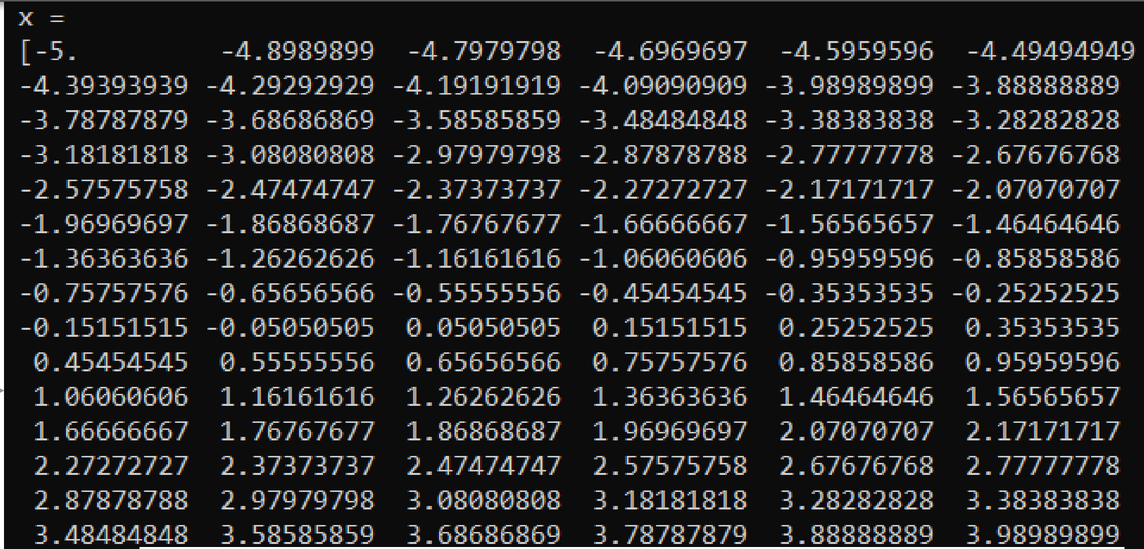
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe

```
x = [ 0.          0.20943951  0.41887902  0.62831853  0.83775804  1.04719755
  1.25663706  1.46607657  1.67551608  1.88495559  2.0943951   2.30383461
  2.51327412  2.72271363  2.93215314  3.14159265  3.35103216  3.56047167
  3.76991118  3.97935069  4.1887902   4.39822972  4.60766923  4.81710874
  5.02654825  5.23598776  5.44542727  5.65486678  5.86430629  6.0737458
  6.28318531  6.49262482  6.70206433  6.91150384  7.12094335  7.33038286
  7.53982237  7.74926188  7.95870139  8.1681409   8.37758041  8.58701992
  8.79645943  9.00589894  9.21533845  9.42477796  9.63421747  9.84365698
 10.05309649 10.262536   10.47197551 10.68141502 10.89085453 11.10029404
 11.30973355 11.51917306 11.72861257 11.93805208 12.14749159 12.3569311
 12.56637061 12.77581012 12.98524963 13.19468915 13.40412866 13.61356817
 13.82300768 14.03244719 14.2418867  14.45132621 14.66076572 14.87020523
 15.07964474 15.28908425 15.49852376 15.70796327 15.91740278 16.12684229
 16.3362818  16.54572131 16.75516082 16.96460033 17.17403984 17.38347935
 17.59291886 17.80235837 18.01179788 18.22123739 18.4306769  18.64011641
 18.84955592 19.05899543 19.26843494 19.47787445 19.68731396 19.89675347
 20.10619298 20.31563249 20.525072   20.73451151 20.94395102 21.15339053
 21.36283004 21.57226955 21.78170906 21.99114858 22.20058809 22.4100276
 22.61946711 22.82890662 23.03834613 23.24778564 23.45722515 23.66666466
 23.87610417 24.08554368 24.29498319 24.5044227  24.71386221 24.92330172
 25.13274123 25.34218074 25.55162025 25.76105976 25.97049927 26.17993878
 26.38937829 26.5988178  26.80825731 27.01769682 27.22713633 27.43657584
 27.64601535 27.85545486 28.06489437 28.27433388 28.48377339 28.6932129
 28.90265241 29.11209192 29.32153143 29.53097094 29.74041045 29.94984996
 30.15928947 30.36872898 30.57816849 30.78760801 30.99704752 31.20648703]
```

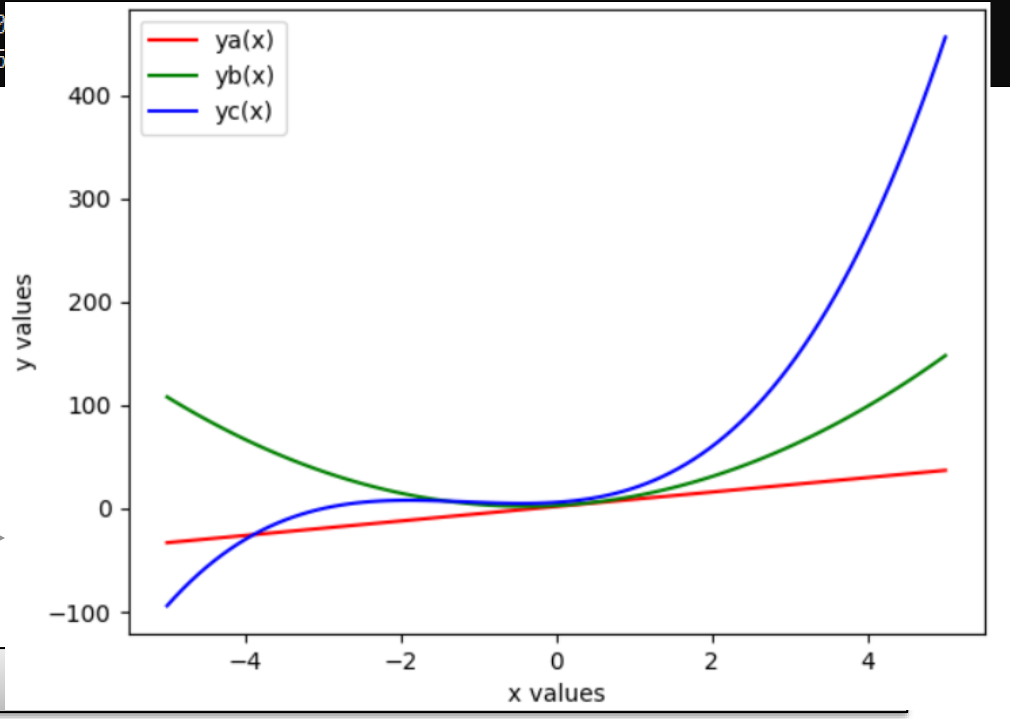


matpolitlib 中的图形绘制

```
import numpy as np
import matplotlib.pyplot as plt
def main():
       # ny.linspace(start, stop, num=50)
    x = np.linspace(-5, 2, 100)
    print(F' x = \n {x}')
    ya = 7*x + 2
    yb = 5*x**2 + 4*x + 3
    yc = 2*x**3 + 7*x**2 + 5*x +6
    fig, ax = plt.subplots()
    ax.plot(x, ya, color = "red", label = "ya(x)")
    ax.plot(x, yb, color = "green", label = "yb(x)")
    ax.plot(x, yc, color = "blue", label = "yc(x)")
    ax.set_xlabel("x values")
    ax.set_ylabel("y values")
    ax.legend()
    plt.show()
main()
```



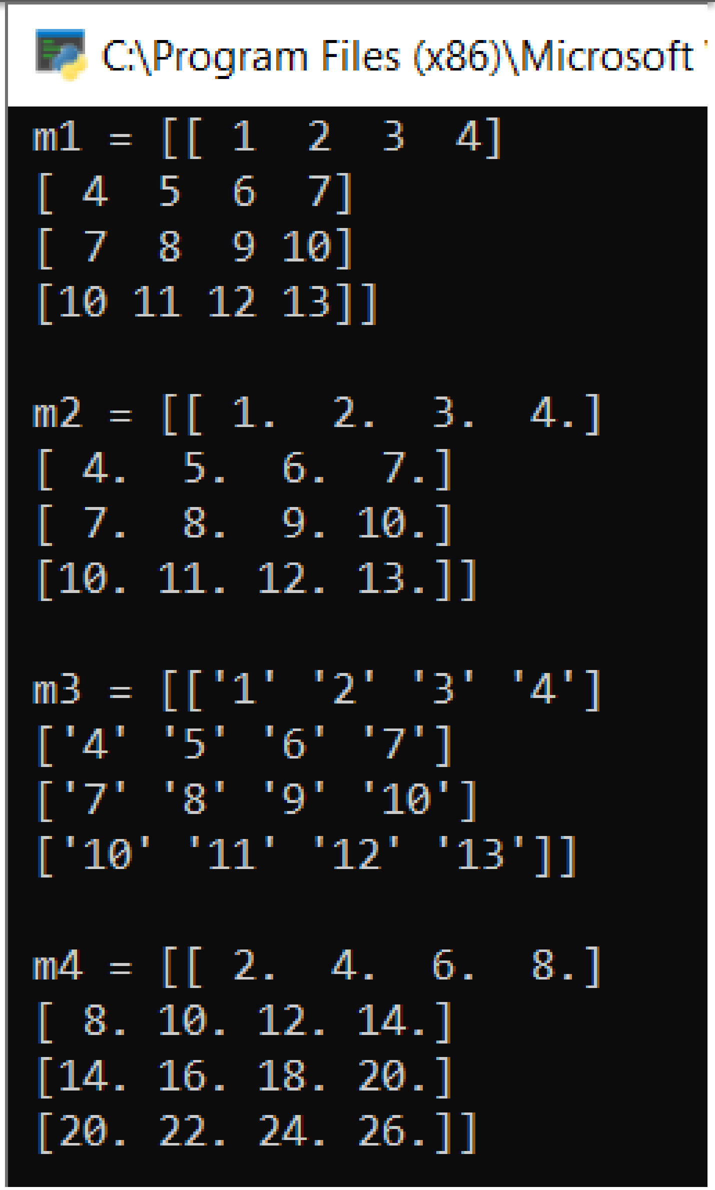
```
x =
[-5.         -4.8989899  -4.7979798  -4.6969697  -4.5959596  -4.49494949
-4.39393939 -4.29292929 -4.19191919 -4.09090909 -3.98989899 -3.88888889
-3.78787879 -3.68686869 -3.58585859 -3.48484848 -3.38383838 -3.28282828
-3.18181818 -3.08080808 -2.97979798 -2.87878788 -2.77777778 -2.67676768
-2.57575758 -2.47474747 -2.37373737 -2.27272727 -2.17171717 -2.07070707
-1.96969697 -1.86868687 -1.76767677 -1.66666667 -1.56565657 -1.46464646
-1.36363636 -1.26262626 -1.16161616 -1.06060606 -0.95959596 -0.85858586
-0.75757576 -0.65656566 -0.55555556 -0.45454545 -0.35353535 -0.25252525
-0.15151515 -0.05050505  0.05050505  0.15151515  0.25252525  0.35353535
 0.45454545  0.55555556  0.65656566  0.75757576  0.85858586  0.95959596
 1.06060606  1.16161616  1.26262626  1.36363636  1.46464646  1.56565657
 1.66666667  1.76767677  1.86868687  1.96969697  2.07070707  2.17171717
 2.27272727  2.37373737  2.47474747  2.57575758  2.67676768  2.77777778
 2.87878788  2.97979798  3.08080808  3.18181818  3.28282828  3.38383838
 3.48484848  3.58585859  3.68686869  3.78787879  3.88888889  3.98989899
 4.09090
 4.69696
```



NumPy 矩阵处理

```
import numpy as np     # installed with matplotlib
import matplotlib.pyplot as plt

def main():
       m1 = np.array([
        [1, 2, 3, 4],
        [4, 5, 6, 7],
        [7, 8, 9, 10],
        [10, 11, 12, 13]
    ]) # no type is default
    m2 = np.array([
        [1, 2, 3, 4],
        [4, 5, 6, 7],
        [7, 8, 9, 10],
        [10, 11, 12, 13]
    ], dtype = float)
```

```
m3 = np.array([
        [1, 2, 3, 4],
        [4, 5, 6, 7],
        [7, 8, 9, 10],
        [10, 11, 12, 13]
    ], dtype = str)
    m4 = m1+m2
    print(F' m1 = {m1}\n\n m2 =
{m2}\n\n m3 = {m3}\n\n m4 = {m4}\n')
main()
```

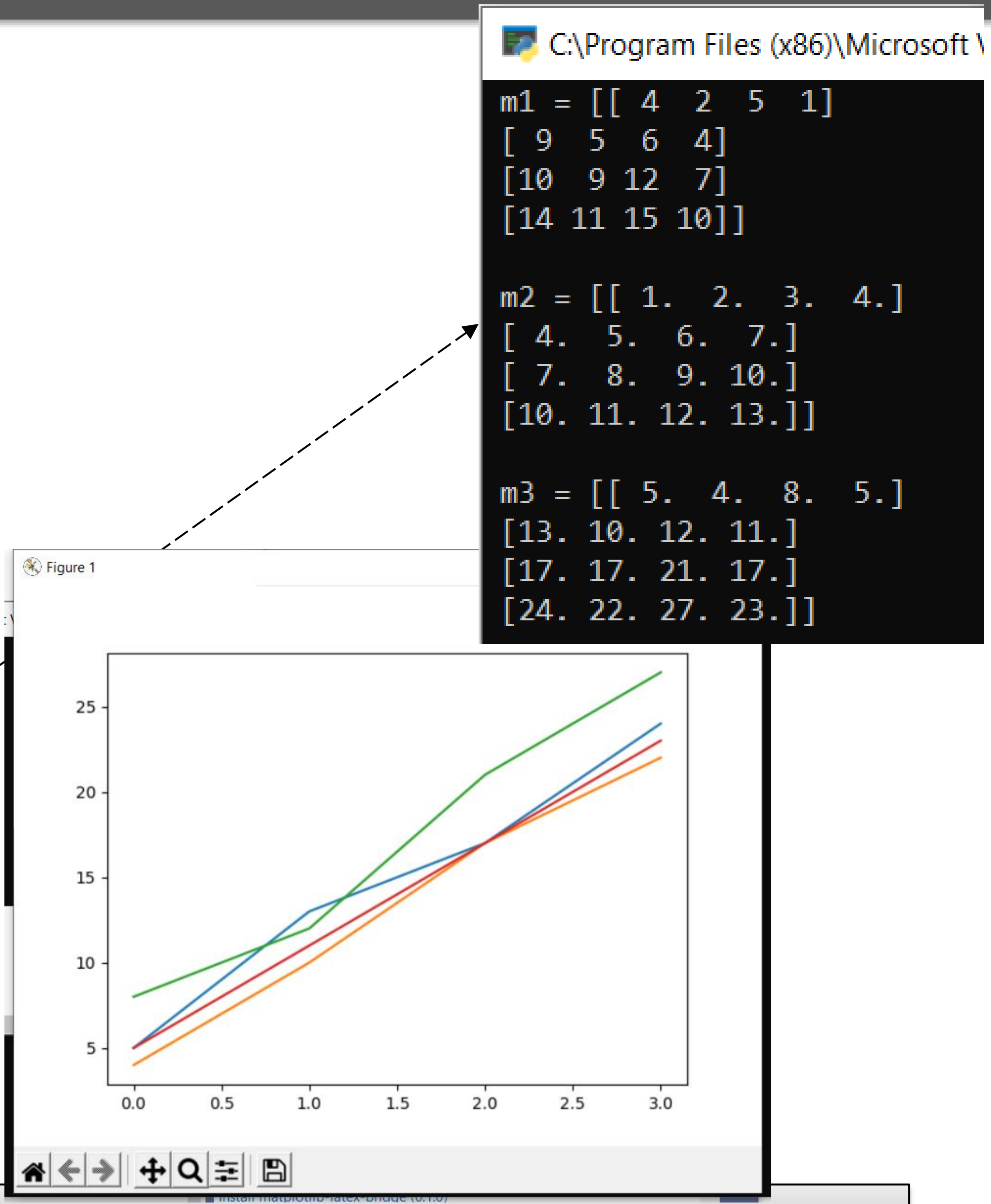


C:\Program Files (x86)\Microsoft

```
m1 = [[ 1  2  3  4]
[ 4  5  6  7]
[ 7  8  9 10]
[10 11 12 13]]

m2 = [[ 1.  2.  3.  4.]
[ 4.  5.  6.  7.]
[ 7.  8.  9. 10.]
[10. 11. 12. 13.]]

m3 = [['1' '2' '3' '4']
['4' '5' '6' '7']
['7' '8' '9' '10']
['10' '11' '12' '13']]

m4 = [[ 2.  4.  6.  8.]
[ 8. 10. 12. 14.]
[14. 16. 18. 20.]
[20. 22. 24. 26.]]
```

NumPy 矩阵处理和绘图

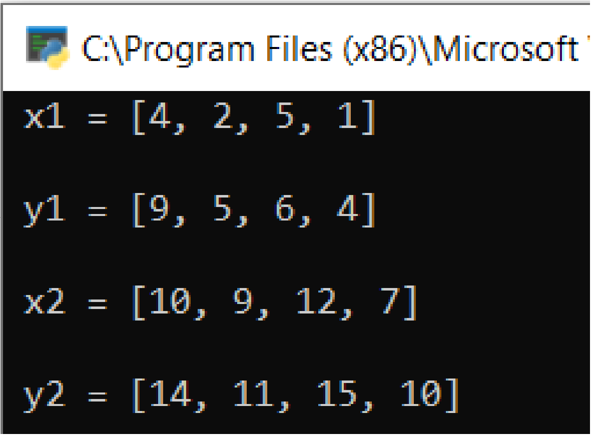
```
import numpy as np      # installed with matplotlib
import matplotlib.pyplot as plt
def main():
    #x = np.arange(0, radians(1800), radians(12))
    #plt.plot(x, np.cos(x), 'b')
    #plt.show()
    m1 = np.array([
        [4, 2, 5, 1], [9, 5, 6, 4],
        [10, 9, 12, 7], [14, 11, 15, 10]
    ]) # no type is default
    m2 = np.array([
        [1, 2, 3, 4], [4, 5, 6, 7],
        [7, 8, 9, 10], [10, 11, 12, 13]
    ], dtype = float)
    m3 = m1+m2
    print(F' m1 = {m1}\n\n m2 = {m2}\n\n m3 = {m3}\n')
    plt.plot(m3)
    plt.show()
main()
```



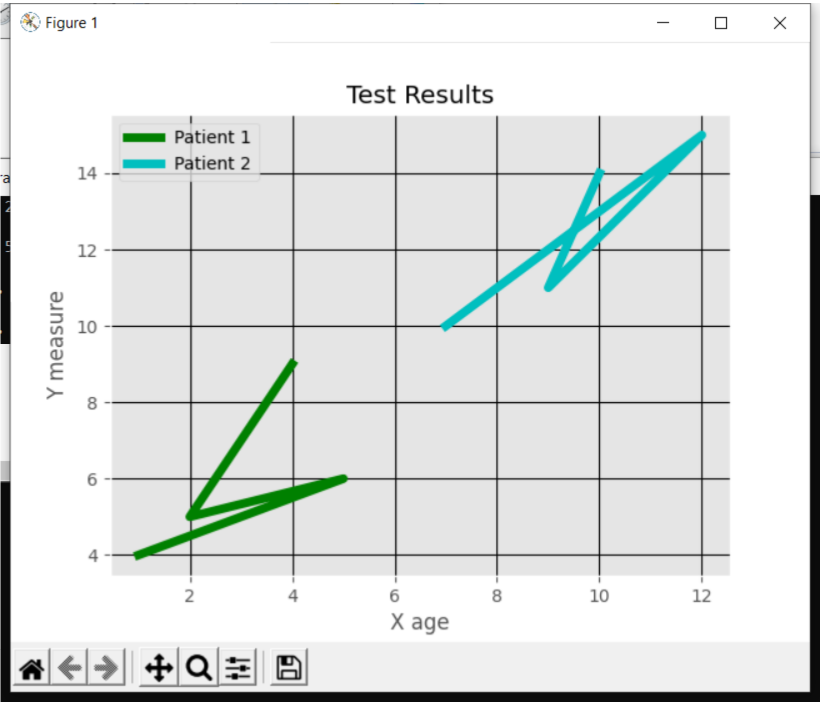
```
m1 = [[ 4  2  5  1]
[ 9  5  6  4]
[10  9 12  7]
[14 11 15 10]]

m2 = [[ 1.  2.  3.  4.]
[ 4.  5.  6.  7.]
[ 7.  8.  9. 10.]
[10. 11. 12. 13.]]

m3 = [[ 5.  4.  8.  5.]
[13. 10. 12. 11.]
[17. 17. 21. 17.]
[24. 22. 27. 23.]]
```

matpolitlib 绘图

```
import numpy as np     # installed with matplotlib
import matplotlib.pyplot as plt
from matplotlib import style
def main():
    style.use('ggplot')
    x1 = [4, 2, 5, 1]
    y1 = [9, 5, 6, 4]
    x2 = [10, 9, 12, 7]
    y2 = [14, 11, 15, 10]
    print(F' x1 = {x1}\n\n y1 = {y1}\n\n x2 = {x2}\n\n y2
= {y2}\n')
    plt.plot(x1,y1,'g',label='Patient 1', linewidth=5)
    plt.plot(x2,y2,'c',label='Patient 2',linewidth=5)
    plt.title('Test Results')
    plt.ylabel('Y measure')
    plt.xlabel('X age')
    plt.legend()
    plt.grid(True,color='k')
    plt.show()
main()
```



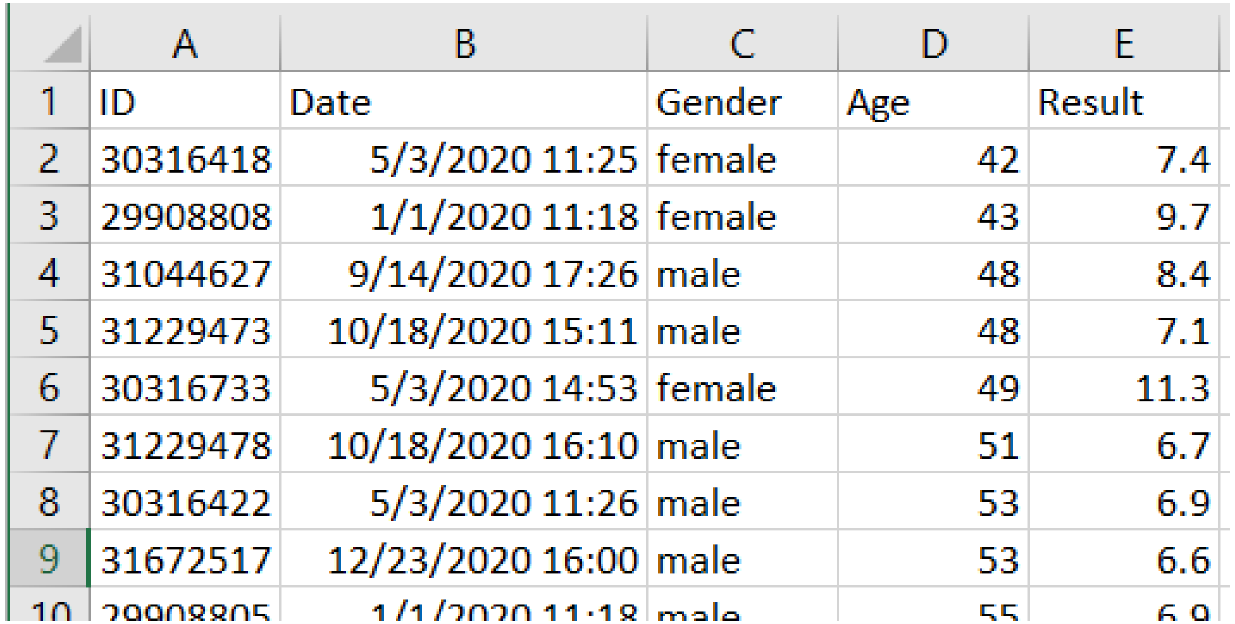
```
x1 = [4, 2, 5, 1]

y1 = [9, 5, 6, 4]

x2 = [10, 9, 12, 7]

y2 = [14, 11, 15, 10]
```



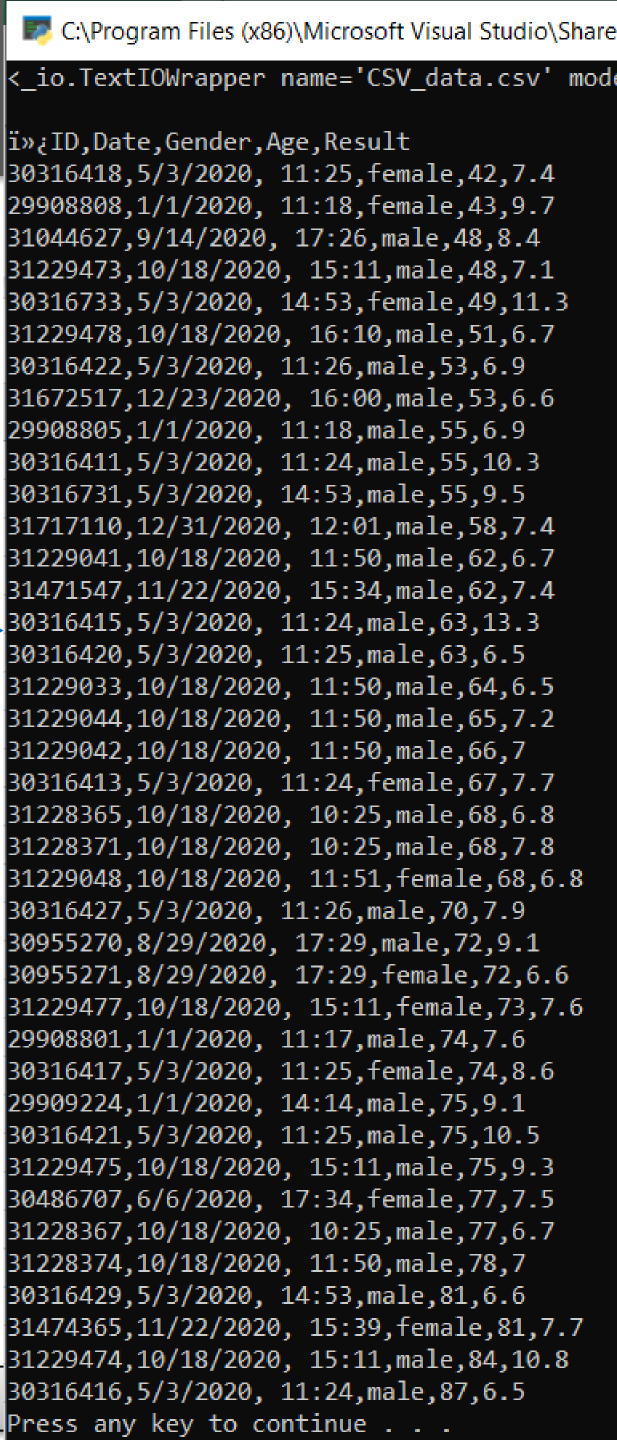
在 python 中使用 CSV 模块读取 CSV 文件

```
import csv     # CSV module in python
def main():
    with open('CSV_data.csv', newline='') as csvfile:
        print(csvfile, '\n')
        myData = csv.reader(csvfile, delimiter=' ', quotechar='|')
        for row in myData:
            print(row)
main()
```

CSV_data.csv



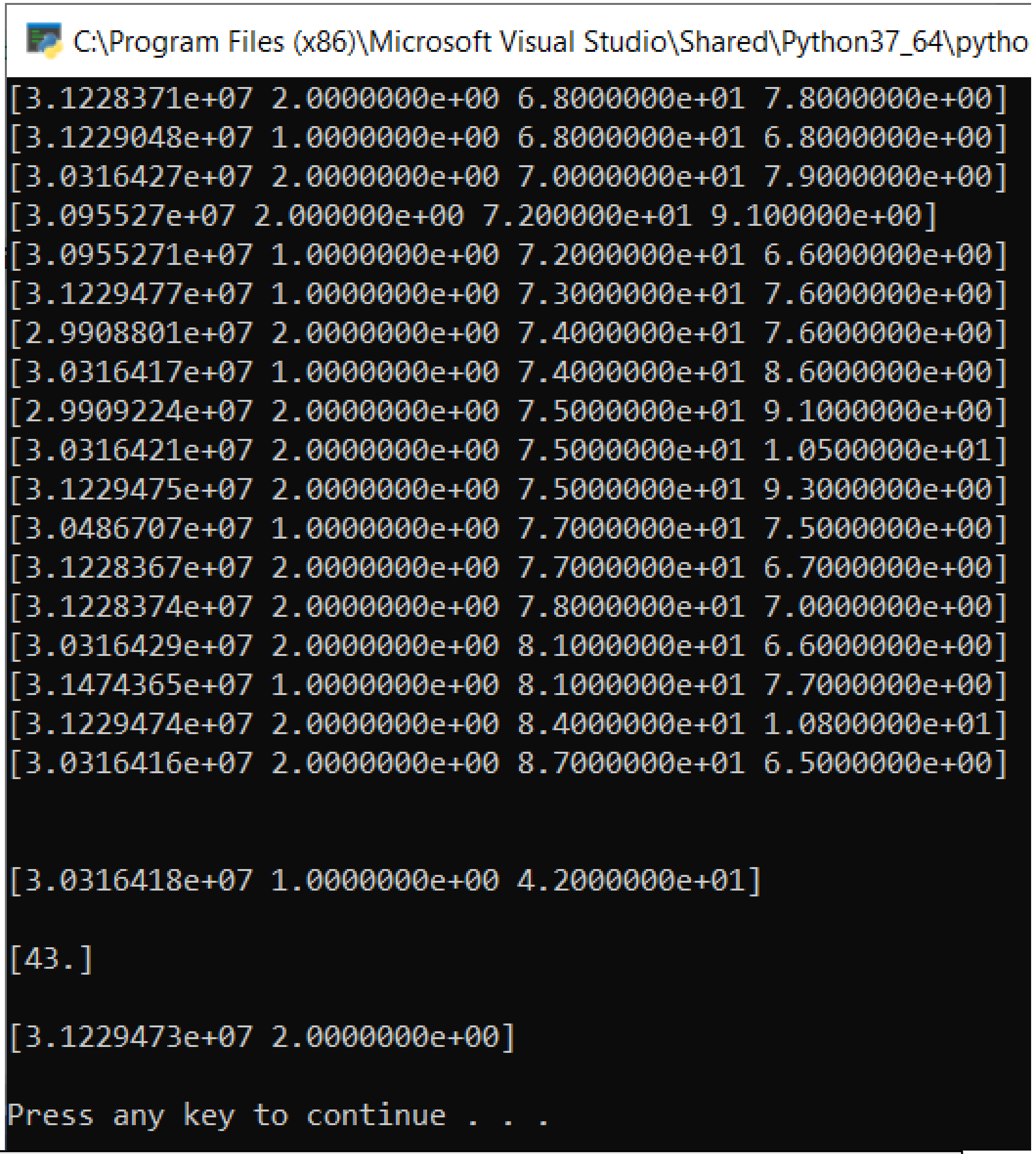
	A	B	C	D	E
1	ID	Date	Gender	Age	Result
2	30316418	5/3/2020 11:25	female	42	7.4
3	29908808	1/1/2020 11:18	female	43	9.7
4	31044627	9/14/2020 17:26	male	48	8.4
5	31229473	10/18/2020 15:11	male	48	7.1
6	30316733	5/3/2020 14:53	female	49	11.3
7	31229478	10/18/2020 16:10	male	51	6.7
8	30316422	5/3/2020 11:26	male	53	6.9
9	31672517	12/23/2020 16:00	male	53	6.6
10	29908805	1/1/2020 11:18	male	55	6.9



```
<_io.TextIOWrapper name='CSV_data.csv' mode

ï»¿ID,Date,Gender,Age,Result
30316418,5/3/2020, 11:25,female,42,7.4
29908808,1/1/2020, 11:18,female,43,9.7
31044627,9/14/2020, 17:26,male,48,8.4
31229473,10/18/2020, 15:11,male,48,7.1
30316733,5/3/2020, 14:53,female,49,11.3
31229478,10/18/2020, 16:10,male,51,6.7
30316422,5/3/2020, 11:26,male,53,6.9
31672517,12/23/2020, 16:00,male,53,6.6
29908805,1/1/2020, 11:18,male,55,6.9
30316411,5/3/2020, 11:24,male,55,10.3
30316731,5/3/2020, 14:53,male,55,9.5
31717110,12/31/2020, 12:01,male,58,7.4
31229041,10/18/2020, 11:50,male,62,6.7
31471547,11/22/2020, 15:34,male,62,7.4
30316415,5/3/2020, 11:24,male,63,13.3
30316420,5/3/2020, 11:25,male,63,6.5
31229033,10/18/2020, 11:50,male,64,6.5
31229044,10/18/2020, 11:50,male,65,7.2
31229042,10/18/2020, 11:50,male,66,7
30316413,5/3/2020, 11:24,female,67,7.7
31228365,10/18/2020, 10:25,male,68,6.8
31228371,10/18/2020, 10:25,male,68,7.8
31229048,10/18/2020, 11:51,female,68,6.8
30316427,5/3/2020, 11:26,male,70,7.9
30955270,8/29/2020, 17:29,male,72,9.1
30955271,8/29/2020, 17:29,female,72,6.6
31229477,10/18/2020, 15:11,female,73,7.6
29908801,1/1/2020, 11:17,male,74,7.6
30316417,5/3/2020, 11:25,female,74,8.6
29909224,1/1/2020, 14:14,male,75,9.1
30316421,5/3/2020, 11:25,male,75,10.5
31229475,10/18/2020, 15:11,male,75,9.3
30486707,6/6/2020, 17:34,female,77,7.5
31228367,10/18/2020, 10:25,male,77,6.7
31228374,10/18/2020, 11:50,male,78,7
30316429,5/3/2020, 14:53,male,81,6.6
31474365,11/22/2020, 15:39,female,81,7.7
31229474,10/18/2020, 15:11,male,84,10.8
30316416,5/3/2020, 11:24,male,87,6.5
Press any key to continue . . .
```

将 CSV 文件加载到数组中

```
import numpy as np      # installed with matplotlib
import csv
import matplotlib.pyplot as plt

def main():
    data = np.genfromtxt('CSV_Floats.csv',
delimiter=',', skip_header = 1)
    # lib func includes open file before reading
    print(data)
    print('\n')
    for r in data:
            print(r)
    print('\n')
    row = [fields for fields in data]
    print(row[0][0:3], '\n')
    print(row[1][2:3], '\n')
    print(row[3][0:2], '\n')
main()
```



C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\pytho

```
[3.1228371e+07 2.0000000e+00 6.8000000e+01 7.8000000e+00]
[3.1229048e+07 1.0000000e+00 6.8000000e+01 6.8000000e+00]
[3.0316427e+07 2.0000000e+00 7.0000000e+01 7.9000000e+00]
[3.095527e+07 2.000000e+00 7.200000e+01 9.100000e+00]
[3.0955271e+07 1.0000000e+00 7.2000000e+01 6.6000000e+00]
[3.1229477e+07 1.0000000e+00 7.3000000e+01 7.6000000e+00]
[2.9908801e+07 2.0000000e+00 7.4000000e+01 7.6000000e+00]
[3.0316417e+07 1.0000000e+00 7.4000000e+01 8.6000000e+00]
[2.9909224e+07 2.0000000e+00 7.5000000e+01 9.1000000e+00]
[3.0316421e+07 2.0000000e+00 7.5000000e+01 1.0500000e+01]
[3.1229475e+07 2.0000000e+00 7.5000000e+01 9.3000000e+00]
[3.0486707e+07 1.0000000e+00 7.7000000e+01 7.5000000e+00]
[3.1228367e+07 2.0000000e+00 7.7000000e+01 6.7000000e+00]
[3.1228374e+07 2.0000000e+00 7.8000000e+01 7.0000000e+00]
[3.0316429e+07 2.0000000e+00 8.1000000e+01 6.6000000e+00]
[3.1474365e+07 1.0000000e+00 8.1000000e+01 7.7000000e+00]
[3.1229474e+07 2.0000000e+00 8.4000000e+01 1.0800000e+01]
[3.0316416e+07 2.0000000e+00 8.7000000e+01 6.5000000e+00]


[3.0316418e+07 1.0000000e+00 4.2000000e+01]

[43.]

[3.1229473e+07 2.0000000e+00]

Press any key to continue . . .
```