实验一： 了解数据处理工具

本实验旨在让同学们了解常用的数据处理工具Numpy、Pandas，以及常用的绘图工具Matplotlib。

1. Numpy

Numpy这个名字其实是Number+Python，其在数学计算，科学计算方面有突出贡献。

1.1 安装

1.1.1 pip

`pip` 的方法很简单，你只需要在终端里面输入下面这样：

```
pip install numpy
```

如果你是 Python3.+ 的版本，用下面这种方式：

```
pip3 install numpy
```

怎么确认自己已经安装好了？首先，它打印出正确安装的信息，然后你再输入这句话，如果没有提示任何信息，则安装好。

```
python3 -c "import numpy"
```

如果提示下面这样的信息，就意味着你的安装失败，请再尝试一下前面的流程。

```
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'numpy'
```

1.1.2 conda

如果你是 conda 来管理 Python 环境，你可以先创建一个 Python 环境，然后再装 numpy。

```
# 创建 Python 环境，如果你已经有一个环境，就不用创建了
conda create -n my-env
conda active my-env

# 在这个 my-env 的Python环境中安装 numpy
conda install numpy

# 或者直接用 pip 安装也能用
pip install numpy
```

怎么确认自己已经安装好了？首先，它打印出正确安装的信息，然后你再输入这句话，如果没有提示任何信息，则安装好。

```
python3 -c "import numpy"
```

如果提示下面这样的信息，就意味着你的安装失败，请再尝试一下前面的流程。

```
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'numpy'
```

1.2 写 Numpy 程序

当你安装好了，你就可以在自己的文件中写 Numpy 代码了。一般的流程是你先 import numpy。为了后续调用numpy更方便，我们通常在 import完之后， 还给它一个缩写形式，as np。接下来你就能用 np.xxx 写 numpy 的代码了，在下面尝试一下吧。

```
import numpy as np

print(np.array([1,2,3])) # [1 2 3]
```

1.3 Numpy与Python List的区别

Numpy的核心优势：运算快。用专业的语言描述的话，Numpy喜欢用电脑内存中连续的一块物理地址存储数据，因为都是连号的嘛，找到前后的号，不用跑很远， 非常迅速。而 Python的List并不是连续存储的，它的数据是分散在不同的物理空间，在批量计算的时候，连号的肯定比不连号的算起来更快。因为找他们的时间更少了。

而且 Numpy Array 存储的数据格式也有限制，尽量都是同一种数据格式，这样也有利于批量的数据计算。 所以只要是处理大规模数据的批量计算，Numpy肯定会比Python的原生 List要快。

```
import time

t0 = time.time()
# python list
```

```
l = list(range(100))
for _ in range(10000):
    for i in range(len(l)):
        l[i] += 1

t1 = time.time()
# numpy array
a = np.array(l)
for _ in range(10000):
    a += 1

print("Python list spend {:.3f}s".format(t1-t0))
print("Numpy array spend {:.3f}s".format(time.time()-t1))
```

得到输出：

```
Python list spend 0.068s
Numpy array spend 0.006s
```

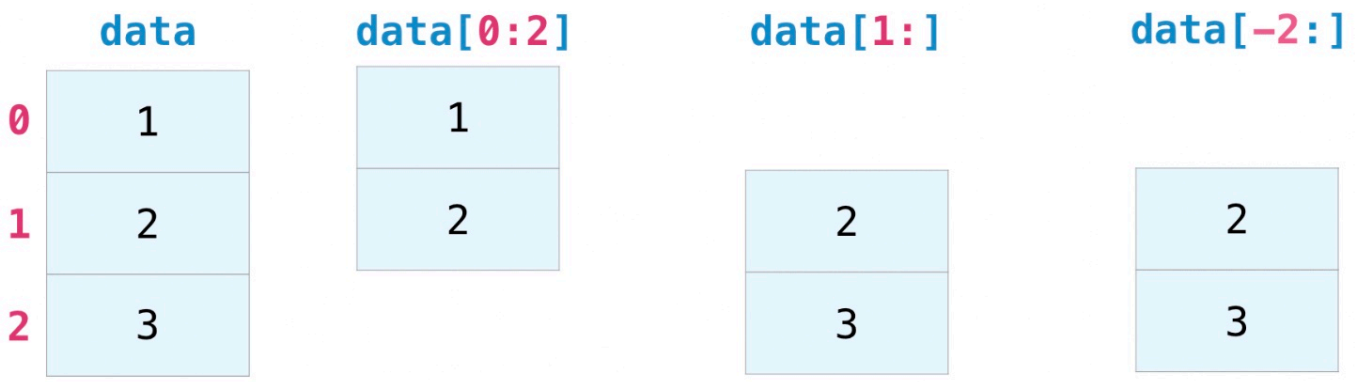
Numpy Array 和 Python List 在很多使用场景上是可以互换的，不过在大数据处理的场景下，而且你的数据类型又高度统一， 那么 Numpy 绝对是你不二的人选，能提升的运算速度也是杠杠的~

1.4 切片

如果我要在 100 个数中，从第一个一直找到第 50 个。我要写成这样 `a[1,2,3,4....50]`，这样太累赘了。Numpy 也考虑得非常周全，它有一个更取巧的方式。

```
a = np.array([1, 2, 3])
print("a[0:2]：\n", a[0:2])
print("a[1:]：\n", a[1:])
print("a[-2:]：\n", a[-2:])

#输出：
#a[0:2]：
# [1 2]
#a[1:]：
# [2 3]
#a[-2:]：
# [2 3]
```



使用 `:` 就能让你跨着取数字，而且一次取一批。**注意，在 Numpy 中：一次取一批和一个个拎起来，拎了一批，是不同的概念哦** 一次取一批来的更快， 因为它不用去一个个查看，一个个数了。

在多维上，也可以进行切片划分。

```
b = np.array([
[1,2,3,4],
[5,6,7,8],
[9,10,11,12]
])

print("b[:2]:\n", b[:2])
print("b[:2, :3]:\n", b[:2, :3])
print("b[1:3, -2:]:\n", b[1:3, -2:])

#输出:
#b[:2]:
# [[1 2 3 4]
# [5 6 7 8]]
#b[:2, :3]:
# [[1 2 3]
# [5 6 7]]
#b[1:3, -2:]:
# [[ 7  8]
# [11 12]]
```

2. Pandas

Pandas 是 Python 中一个比较常用的第三方库，里面集成了很多和数据相关的功能组件。它承接了 Numpy 的能力，使用的底层也是 Numpy。

2.1 安装

不管你是用什么途径安装的 Python，你都可以用这个 Python 自带的一个叫 Pip 的包管理工具来安装 Pandas 库，或者其他第三方库。 直接在 Windows 的 cmd 工具中，或者 Mac 的 Terminal 工具中，输入下面的指令：

```
pip install pandas
```

如果你是 Python3 的版本，也可以输入

```
pip3 install pandas
```

安装好之后，提示成功安装之后，请打开你的 Python 编辑器，在编辑器中输入下列指令并运行，你应该能看到和我一样的运行结果。

```
import pandas as pd
print(pd.Series([1,2,3]))

#输出
#0    1
#1    2
#2    3
#dtype: int64
```

2.2 与Numpy的区别

Pandas 是在 Numpy 上的封装。 继承了 Numpy 的所有优点，但是这种封装有好有坏，我们在这节内容中就先来阐述一下 Pandas 和 Numpy 的对比。

用过Python，你肯定熟悉里面的 List 和 Dictionary, 拿这两种形态来对比 Numpy 和 Pandas 的关系。

```
a_list = [1,2,3]
a_dict = {"a": 1, "b": 2, "c": 3}
print("list:", a_list)
print("dict:", a_dict)

#输出:
#list: [1, 2, 3]
#dict: {'a': 1, 'b': 2, 'c': 3}
```

上面就是一种最常见的 Python 列表和字典表达方式。而下面，我们展示的就是 Numpy 和 Pandas 的一种构建方式。

```
import pandas as pd
import numpy as np

a_array = np.array([
    [1,2],
    [3,4]
])
a_df = pd.DataFrame(
    {"a": [1,3],
     "b": [2,4]}
)

print("numpy array:\n", a_array)
print("\npandas df:\n", a_df)

#输出:
#numpy array:
# [[1 2]
# [3 4]]
#
#pandas df:
#    a  b
#0  1  2
#1  3  4
```

你会发现，我们看到的结果中，Numpy的是没有任何数据标签信息的，你可以认为它是纯数据。而 Pandas 就像字典一样，还记录着数据的外围信息， 比如标签（Column 名）和索引（Row index）。 这也是为什么说 Numpy 是 Python 里的列表，而 Pandas 是 Python 里的字典。

还是回到之前提的问题，对于数据运算，既然我们有了 Numpy，为什么还要用 Pandas？。对比列表和字典，我们很容易感受到其中的一种原因。 就是 Pandas 帮我们记录的信息量变多了。

在 Numpy 中， 如果你不特别在其他地方标注，你是不清楚记录的这里边记录的是什么信息的。**而 Pandas 记录的信息可以特别丰富， 你给别人使用传播数据的时，这些信息也会一起传递过去。或者你自己处理数据时对照着信息来加工数据，也会更加友善。**

这就是在我看来 Pandas 对比 Numpy 的一个最直观的好处。

另外 Pandas 用于处理数据的功能也比较多，信息种类也更丰富，特别是你有一些包含字符的表格，Pandas 可以帮你处理分析这些字符型的数据表。 当然还有很多其它功能，比如处理丢失信息，多种合并数据方式，读取和保存为更可读的形式等等。

这些都让 Pandas 绽放光彩。**但是，Pandas 也有不足的地方：运算速度稍微比 Numpy 慢。**

因为 Pandas 是在 Numpy 之上的一层封装，所以肯定在处理数据的时候要多几层处理，小数据量的处理不要紧，慢一点就慢一点， 你也感受不到处理速度的变化。但当数据量变大，用 Numpy 要处理 1 小时的数据，你可能用 Pandas 要花两小时。 所以你得依据自己的实际需求来选择到底是用 Numpy 还是 Pandas。

如果在做少量数据的分析时，因为不涉及到机器学习的模型运算等，都可以用 Pandas，但如果要模型训练， 训练过程中还一直要调用数据处理的功能， 肯定毫不犹豫都用 Numpy 来做。

Pandas 是 Numpy 的封装库，继承了 Numpy 的很多优良传统，也具备丰富的功能组件，但是你还是得分情况来酌情选择要使用的工具。

3. Matplotlib

Matplotlib 是一个非常强大的 Python 画图工具;

手中有很多数据, 可是不知道该怎么呈现这些数据.

所以就找到了 Matplotlib. 它能帮你画出美丽的:

- 线图;
- 散点图;
- 等高线图;
- 条形图;
- 柱状图;
- 3D 图形,
- 甚至是图形动画等等.

3.1 基础应用

使用`import`导入模块`matplotlib.pyplot`，并简写成`plt` 使用`import`导入模块`numpy`，并简写成`np`

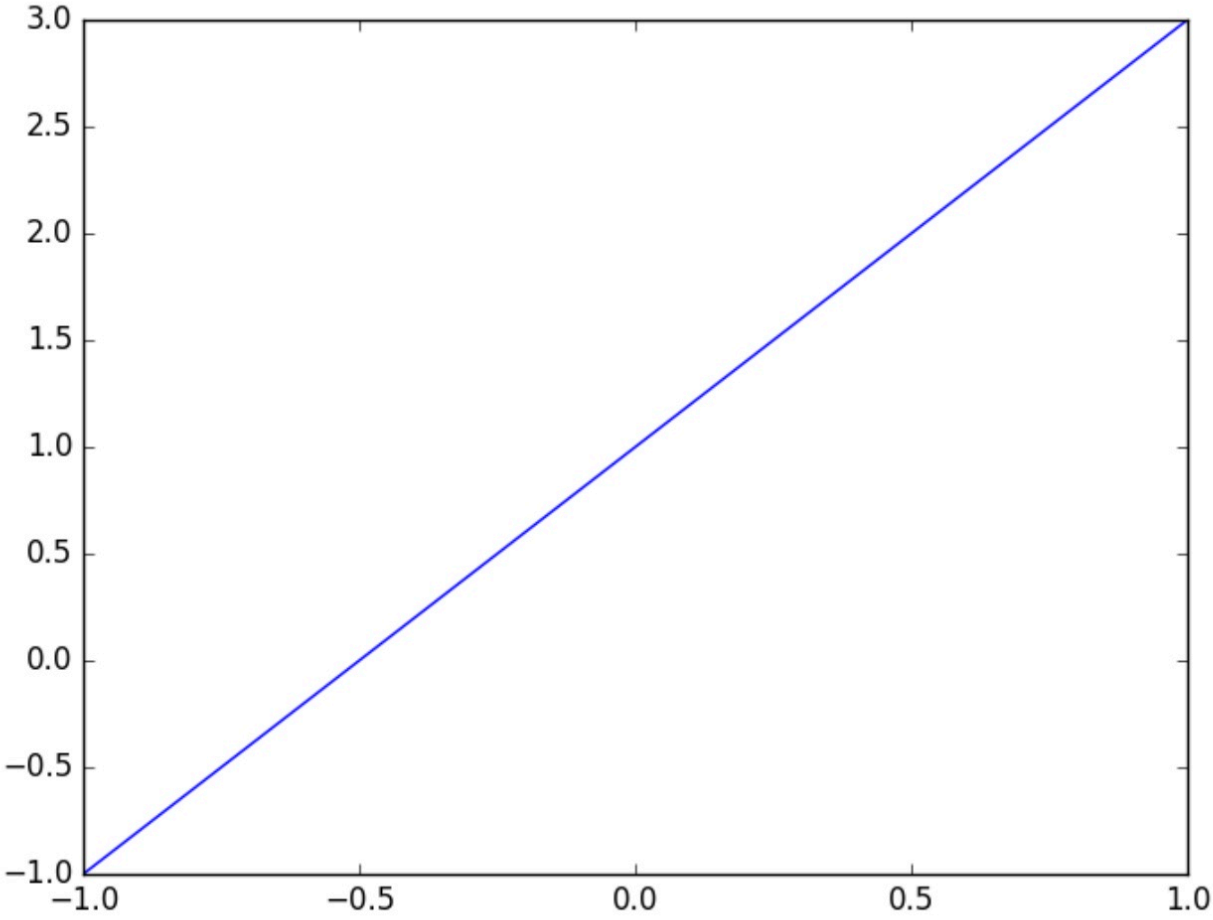
```
import matplotlib.pyplot as plt
import numpy as np
```

使用`np.linspace`定义x：范围是(-1,1);个数是50. 仿真一维数据组(`x` ,`y`)表示曲线1.

```
x = np.linspace(-1, 1, 50)
y = 2*x + 1
```

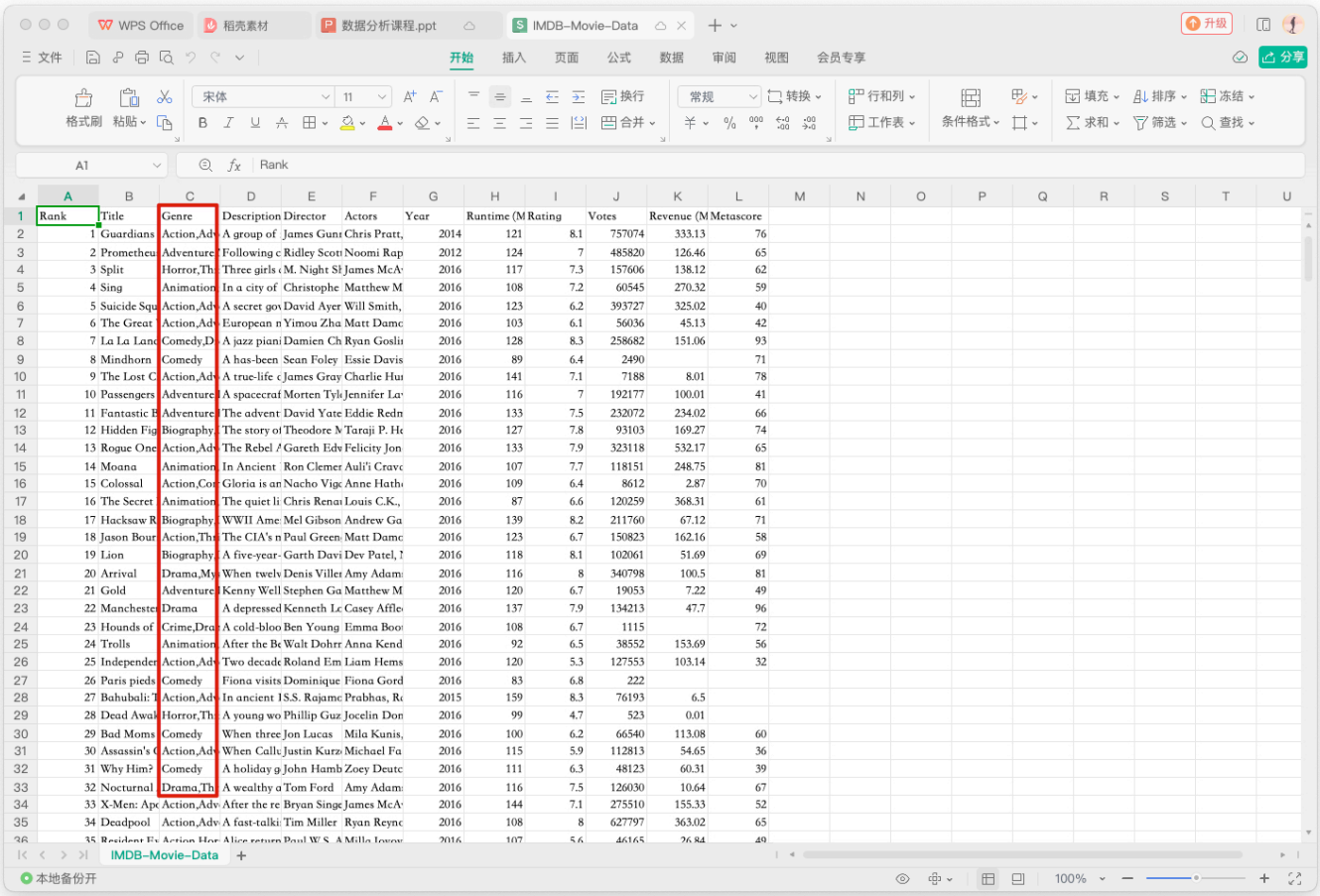
使用`plt.figure`定义一个图像窗口. 使用`plt.plot`画(`x` ,`y`)曲线. 使用`plt.show`显示图像.

```
plt.figure()
plt.plot(x, y)
plt.show()
```



4. 动手实践

IMDB-Movie-Data.csv文件中有1000部电影的信息，其中每部电影都有对应的类型：



Rank	Title	Genre	Description	Director	Actors	Year	Runtime (M	Rating	Votes	Revenue (M	Metascore
1	Guardians	Action,Adv	A group of	James Gunı	Chris Pratt,	2014	121	8.1	757074	333.13	76
2	Prometheu	Adventure	Following c	Ridley Scot	Noomi Rap	2012	124	7	485820	126.46	65
3	Split	Horror,Thr	Three girls	M. Night Sl	James McA	2016	117	7.3	157606	138.12	62
4	Sing	Animation	In a city of	Christophe	Matthew M	2016	108	7.2	60545	270.32	59
5	Suicide Squ	Action,Adv	A secret gov	David Ayer	Will Smith,	2016	123	6.2	393727	325.02	40
6	The Great	Action,Adv	European n	Yimou Zha	Matt Damc	2016	103	6.1	56036	45.13	42
7	La La Land	Comedy,D	A jazz piani	Damien Ch	Ryan Goslir	2016	128	8.3	258682	151.06	93
8	Mindhorn	Comedy	A has-been	Sean Foley	Essie Davis	2016	89	6.4	2490		71
9	The Lost C	Action,Adv	A true-life	James Gray	Charlie Hur	2016	141	7.1	7188	8.01	78
10	Passengers	Adventure	A spacecraf	Morten Tyld	Jennifer Lav	2016	116	7	192177	100.01	41
11	Fantastic B	Adventure	The advent	David Yate	Eddie Redn	2016	133	7.5	232072	234.02	66
12	Hidden Figı	Biography	The story of	Theodore M	Taraji P. He	2016	127	7.8	93103	169.27	74
13	Rogue One	Action,Adv	The Rebel A	Gareth Edv	Felicity Jon	2016	133	7.9	323118	532.17	65
14	Moana	Animation	In Ancient	Ron Clemer	Auli'i Cravc	2016	107	7.7	118151	248.75	81
15	Colossal	Action,Cor	Gloria is an	Nacho Vigc	Anne Hathe	2016	109	6.4	8612	2.87	70
16	The Secret	Animation	The quiet li	Chris Renai	Louis C.K.,	2016	87	6.6	120259	368.31	61
17	Hacksaw R	Biography	WWII Ame	Mel Gibson	Andrew Ga	2016	139	8.2	211760	67.12	71
18	Jason Bour	Action,Thr	The CIA's n	Paul Green	Matt Damc	2016	123	6.7	150823	162.16	58
19	Lion	Biography	A five-year-	Garth Davi	Dev Patel, I	2016	118	8.1	102061	51.69	69
20	Arrival	Drama,My	When twelv	Denis Ville	Amy Adam	2016	116	8	340798	100.5	81
21	Gold	Adventure	Kenny Well	Stephen Ga	Matthew M	2016	120	6.7	19053	7.22	49
22	Manchester	Drama	A depressed	Kenneth Lo	Casey Affle	2016	137	7.9	134213	47.7	96
23	Hounds of	Crime,Dra	A cold-bloo	Ben Young	Emma Boo	2016	108	6.7	1115		72
24	Trolls	Animation	After the Be	Walt Dohrr	Anna Kend	2016	92	6.5	38552	153.69	56
25	Independen	Action,Adv	Two decade	Roland Em	Liam Hems	2016	120	5.3	127553	103.14	32
26	Paris pieds	Comedy	Fiona visits	Dominique	Fiona Gord	2016	83	6.8	222		
27	Bahubali: T	Action,Adv	In ancient I	S.S. Rajamc	Prabhas, Ra	2015	159	8.3	76193	6.5	
28	Dead Awak	Horror,Thr	A young wo	Phillip Guz	Jocelin Don	2016	99	4.7	523	0.01	
29	Bad Moms	Comedy	When three	Jon Lucas	Mila Kunis,	2016	100	6.2	66540	113.08	60
30	Assassin's C	Action,Adv	When Callu	Justin Kurz	Michael Fa	2016	115	5.9	112813	54.65	36
31	Why Him?	Comedy	A holiday g	John Hamb	Zoey Deutc	2016	111	6.3	48123	60.31	39
32	Nocturnal	Drama,Thr	A wealthy a	Tom Ford	Amy Adam	2016	116	7.5	126030	10.64	67
33	X-Men: Apc	Action,Adv	After the re	Bryan Singe	James McA	2016	144	7.1	275510	155.33	52
34	Deadpool	Action,Adv	A fast-talki	Tim Miller	Ryan Reync	2016	108	8	627797	363.02	65
35	Resident Ev	Action,Hor	Alice return	Paul W.S. A	Milla Jovov	2016	107	5.6	46165	26.84	49

我们希望统计电影分类(genre)的情况，应该如何处理数据？需要绘制出图像，直观展示。

如：