

2018 年春季《大学计算机基础》（理科）实验指导书

实验 7 科学计算与绘图

1. 实验目的

- （1）学会安装 Numpy、Matplotlib 等 Python 扩展模块
- （2）利用 Numpy 和 Matplotlib 进行科学计算和绘制基本图像

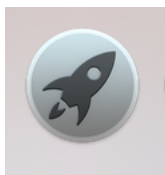
2. 实验任务

Numpy 以及 Matplotlib 的安装方法（之后的 tkinter 等扩展包同理）

Mac 系统：在终端使用 pip 进行安装

步骤：

1. 打开屏幕下方 Launchpad

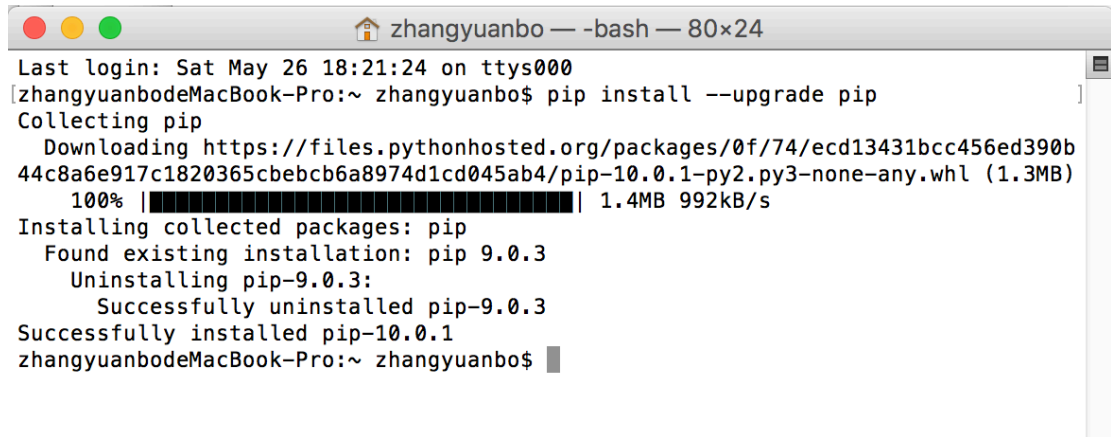


2. 搜索“终端”



3. 升级 pip

输入: `pip install --upgrade pip`

A terminal window titled 'zhangyuanbo — -bash — 80x24'. The output shows the command 'pip install --upgrade pip' being executed. It reports downloading pip from a PythonHosted.org URL, showing a progress bar at 100%, and then upgrading from pip 9.0.3 to pip 10.0.1. The final prompt is 'zhangyuanbodeMacBook-Pro:~ zhangyuanbo\$'.

4. 安装扩展包

格式: `pip install 扩展包名称`

例如: `pip install matplotlib`

程序将自动进行 Numpy、Six 等相关依赖包的安装, 然后安装 Matplotlib

Windows 系统: 在 cmd 使用 pip 进行安装

以课程中心提供的 Python3.4 版本为例

步骤:

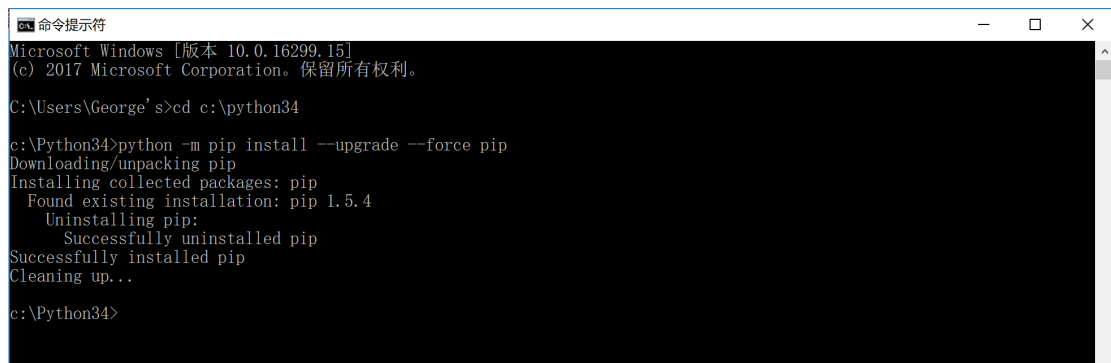
1. 升级 pip

首先找到 Python 的安装目录, 例如 C:\Python34

进入开始菜单, 输入: cmd, 回车, 打开命令提示符

输入: `cd python 安装目录`, 然后回车

例如: `cd C:\Python34`

A Windows command prompt window titled '命令提示符'. It shows the command 'python -m pip install --upgrade --force pip' being executed in the directory 'C:\Python34'. The output shows pip 1.5.4 being uninstalled and pip being successfully installed. The prompt ends with 'c:\Python34>'.

右键粘贴以下内容 (不可 `ctrl+V`): `python -m pip install --upgrade --force pip`

2. 安装扩展包

在前一步的基础上，输入：`cd Scripts`

```
c:\Python34>cd scripts  
c:\Python34\Scripts>
```

然后输入：`pip install 扩展包名称`

例如：`pip install matplotlib`

程序将自动进行 Numpy、Six 等相关依赖包的安装，然后安装 Matplotlib

如果速度过慢，可以改用清华的镜像库安装，将上面蓝色字体替换为：

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple matplotlib
```



3. 检测是否安装成功的方法

如果上述步骤完成后没有红字报错（有时可能会有黄字提示，属正常现象）大概率说明安装成功

成功与否的标志：进入 IDLE，输入：`import matplotlib`，如果没有报错，则安装成功

```
>>> import matplotlib  
>>> |
```

实验任务 7-1 函数图像绘制练习

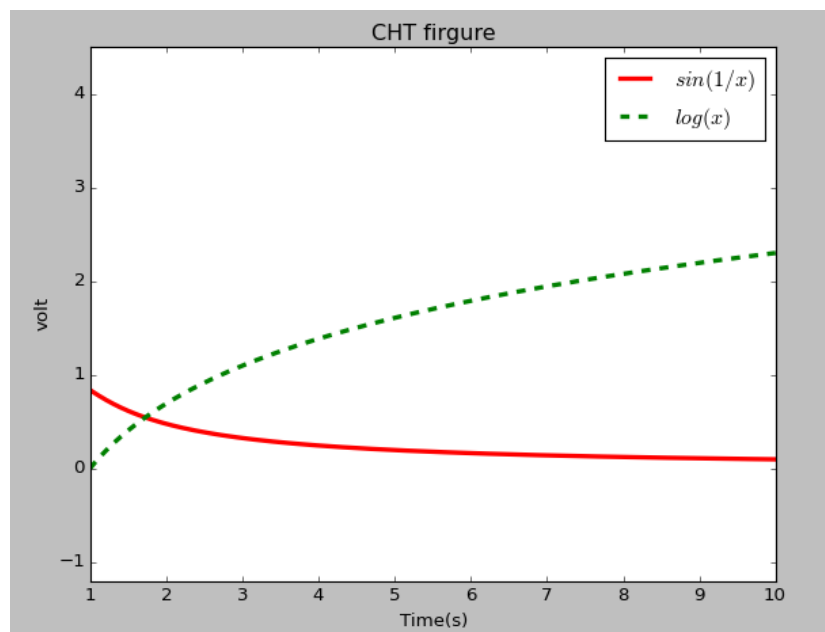
题目描述：

我们在平时学习生活中总是会遇到各种各样的函数，并非每个函数都可以手动绘制图像，有时可以借助计算机帮助我们绘制图像，今天需要同学们绘制 $\sin(1/x)$ 和 $\log(x)$ 的图像。

输入格式：

无输入数据

输出格式：



要求横坐标区间从 1 到 10，纵坐标区间从-1.2 到 4.5。 $\sin(1/x)$ 为实线，线为红色，线宽为 3。 $\log(x)$ 为虚线，线为绿色，线宽为 3。需要有每个图像的 label, xlabel, ylabel 和 title(有即可，不要求文字一样)，画布大小 8*6。（以上要求可不满足，只要画出函数图形即可得到绝大部分的分数）

实验指导：

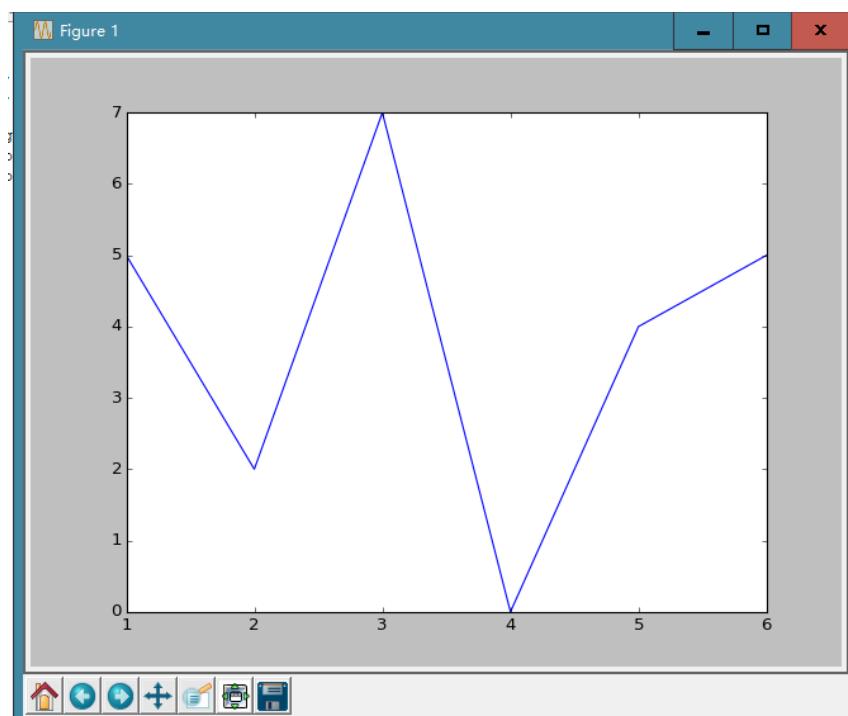
1. `plt.plot(a, b)` 其中 a, b 是两个列表，`plt.plot` 工作的原理是把 a, b 两个列表中对对应点画在直角坐标系中，再用直线连接。

例如：

```
import matplotlib.pyplot as plt

a = [1,2,3,4,5,6]
b = [5,2,7,0,4,5]
|
plt.figure()
plt.plot(a,b)
plt.show()
```

输出图像：



2. 所以我们要绘制一个图像，必须要得到足够多的点，对于横坐标的生成我们可以用 numpy 包中的 arange 函数。arange(a,b,c) 中 a 为起始点，b 为终止点，c 为步长。返回一个列表（实际类型是 numpy.ndarray）。（也可以使用 linspace 函数）

```
>>> import numpy as np
>>> x = np.arange(0,5,0.1)
>>> print(x)
[ 0.  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.  1.1  1.2  1.3  1.4
  1.5  1.6  1.7  1.8  1.9  2.  2.1  2.2  2.3  2.4  2.5  2.6  2.7  2.8  2.9
  3.  3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.  4.1  4.2  4.3  4.4
  4.5  4.6  4.7  4.8  4.9]
```

3. np.sin 函数会作用于一个列表，对该列表中每个元素进行 sin 运算。（math.sin 函数只能作用于一个数）。
例如：

```

import numpy as np
import matplotlib.pyplot as plt
import math
x = np.arange(0,5,0.1)
print(x)
y = np.sin(x)
print(y)

z = []
for i in x:
    z.append(math.sin(i))
print(z)

```

输出结果:

```

>>> ===== RESTART =====
>>>
[ 0.  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.  1.1  1.2  1.3  1.4
 1.5  1.6  1.7  1.8  1.9  2.  2.1  2.2  2.3  2.4  2.5  2.6  2.7  2.8  2.9
 3.  3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9  4.  4.1  4.2  4.3  4.4
 4.5  4.6  4.7  4.8  4.9]
[ 0.  0.09983342  0.19866933  0.29552021  0.38941834  0.47942554
 0.56464247  0.64421769  0.71735609  0.78332691  0.84147098  0.89120736
 0.93203909  0.96355819  0.98544973  0.99749499  0.9995736  0.99166481
 0.97384763  0.94630009  0.90929743  0.86320937  0.8084964  0.74570521
 0.67546318  0.59847214  0.51550137  0.42737988  0.33498815  0.23924933
 0.14112001  0.04158066 -0.05837414 -0.15774569 -0.2555411 -0.35078323
-0.44252044 -0.52983614 -0.61185789 -0.68776616 -0.7568025 -0.81827711
-0.87157577 -0.91616594 -0.95160207 -0.97753012 -0.993691 -0.99992326
-0.99616461 -0.98245261]
[0.0, 0.09983341664682815, 0.19866933079506122, 0.2955202066613396, 0.3894183423
086505, 0.479425538604203, 0.5646424733950355, 0.6442176872376911, 0.71735609089
95228, 0.7833269096274834, 0.8414709848078965, 0.8912073600614354, 0.93203908596
72264, 0.963558185417193, 0.9854497299884603, 0.9974949866040544, 0.999573603041
5051, 0.9916648104524686, 0.9738476308781951, 0.9463000876874145, 0.909297426825
6817, 0.8632093666488737, 0.8084964038195901, 0.74570521217672, 0.67546318055115
06, 0.5984721441039564, 0.5155013718214642, 0.4273798802338298, 0.33498815015590
466, 0.23924932921398198, 0.1411200080598672, 0.04158066243329049, -0.0583741434
27580086, -0.15774569414324865, -0.25554110202683167, -0.35078322768961984, -0.4
4252044329485246, -0.5298361409084934, -0.6118578909427193, -0.6877661591839741,
-0.7568024953079282, -0.8182771110644108, -0.8715757724135882, -0.9161659367494
549, -0.951602073889516, -0.977530117665097, -0.9936910036334645, -0.99992325756
41008, -0.9961646088358406, -0.9824526126243325]
>>>

```

4. 更为具体的，我们可以在 `plt.plot(a,b,···)` 中 `···` 部分，加入参数替换缺省值，实现标注 `label (label =)`，更换曲线的颜色 (`color =`)，改变曲线宽度 (`linewidth =`)，线形等。
5. 还可以使用 `plt` 一些其他函数设置 `title`，`label`，对图像大小限制等。

实验任务 7-2 制造和管理科学数据

题目描述:

泊松分布和正态分布在我们生活中十分常见。

其中泊松分布有很多应用，比如互联网的流量模式服从泊松分布，操作系统中的进程到达时间符合泊松分布，在给定时间内股票价格的下降次数也符合泊松分布，而泊松分布的算法实现在高德纳的《计算机程序设计艺术(卷 2)》中给出了，其代码如下：

```
1  algorithm poisson_random_number(Knuth):
2      initializations:
3          L = e - λ
4          k = 0
5          p = 1
6      do:
7          k = k + 1
8          u = uniform_random_number(0,1)
9          p = p × u
10         while p > L:
11             return k - 1
```

而正态分布在统计学上也有十分重要的意义，它的各种数学特征一直以来都是我们学习的重点。

实验内容分为两部分：

1. 泊松分布

请你通过查阅相关资料，编写程序随机生成 8 个符合泊松分布（其中 $\lambda=8$ ）的数

2. 正态分布

通过 SciPy 从标准正态分布 $N(0, 1)$ 中返回十个样本值，分别输出它们的数量，具体数值，最小值，最大值，均值，方差，偏度和峰度。（最小值，最大值，均值，方差，偏度和峰度都需要小数点后保留五位小数）

输入格式:

无输入数据

输出格式:

```
===== RESTART: E:\WinPython-64bit-3.4.4.2\notebooks\1.py =====
Poisson distribution:
8 6 9 8 8 3 11 7

Normal distribution:
Number of elements:10
Elements:
[-1.65494575 -2.57859553  0.38382772 -0.34034051  1.2371887  -0.31227661
  0.0151612  1.83956318 -0.46608373  0.97625382]
Minimum:-2.57860 Maximum:1.83956
Mean:-0.09002
Variance:1.75031
Skewness:-0.42899
Kurtosis:-0.47387
>>> |
```

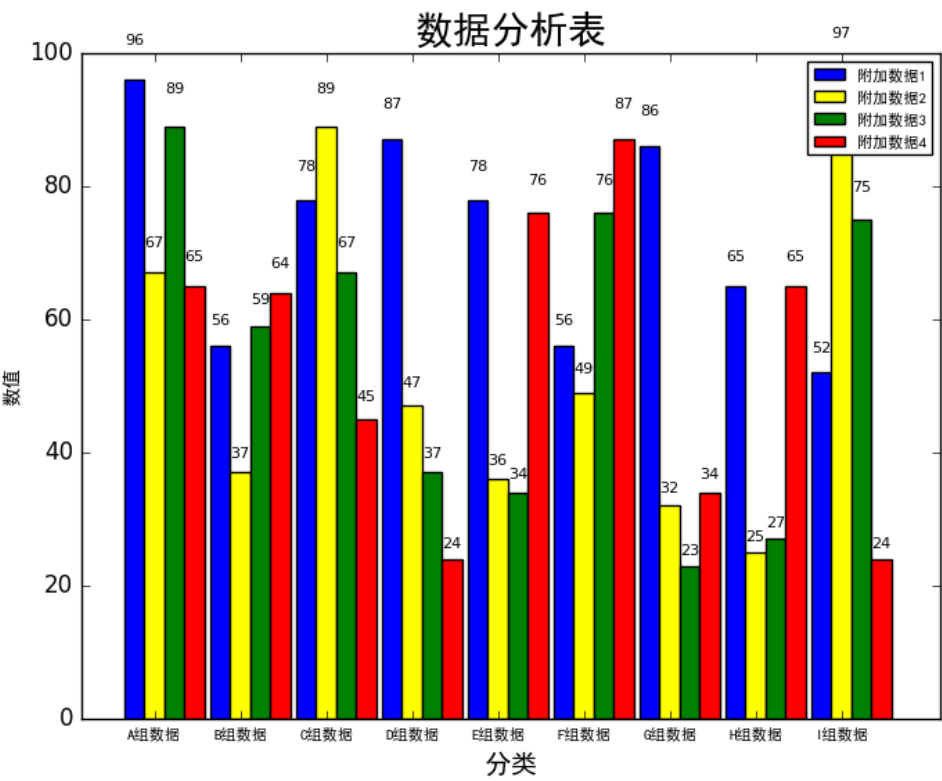
实验指导:

1. 本题可能需要用到的部分相关资料已给出, 请自行甄选
2. 泊松分布部分需要用到 math 和 random 模块, 具体算法可检索, 重点在于让大家熟悉用编程解决数学问题的过程
3. 正态分布部分需要使用 SciPy 模块, 所需要的各个数学特征都可通过 scipy.randn() 以及 scipy.stats.describe() 完成, 需要大家注意按照样例输出的格式, 最小值, 最大值, 均值, 方差, 偏度和峰度都需要小数点后保留五位小数

实验任务 7-3 图表绘制练习

题目描述：

柱形图是处理大量数据并进行比较的一个较为优秀的图形，现有数据文件 `check.py`，请根据其中的列表中的数据，编写程序构建一个柱形图。柱形图形式要求与下图相同（字号类似即可，不做要求）。



输入格式：

无输入数据，具体数据参见 `check.py`

输出格式：

形如上图柱形图一张

实验指导：

1. 柱形图绘制参考书 p285。
2. 注意 `bar` 函数可进行缺省。
3. 替换 `x_tick` 利用 `ax.setticklabels(labels, fontproperties, size)`
4. 函数可能不稳定，多进行尝试。