

Note

2020年8月8日 4:01

各位同学老师大家下午好，这个第一节实验课就由我来给大家讲。我先简单介绍一下我自己，我叫张学臻，来自中山大学，目前本科在读，然后这节课的教程是萧鸿宇师兄编写的，我负责讲解。

1. 实验课的内容梗概：

这次暑期学校的实验课内容主要有几个部分，包括python、shell基础，可能的同学对此不是太熟悉，其次是噪声相关的数值实验、以及对一些真实地震数据的处理，用python对论文内容重现，最后是我们噪声程序包的介绍。今天由我来讲第一部分，python、shell、sac的基础部分，主要涉及到是一些程序上的基础内容，方便一些不太熟悉的同学对后续课程的理解，便于后续课程的进行，而且我也是在一个学习的过程，所以啊，大家只要有什么问题就尽管提出来，对我来说也是相互学习的过程。

我们正式进入第一部分，我先简单介绍一些这次课程主要讲的内容，我们主要介绍一下jupyter notebook，以及python的一些基础内容，shell 的一些关于目录、文件等等的常用操作命令，最后简单讲讲sac的基础命令。

1. Intro Jupyter Notebook

1. 简介

Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。——[Jupyter Notebook官方介绍](#)

简而言之，Jupyter Notebook是以网页的形式打开，可以在网页页面中直接编写代码和运行代码，代码的运行结果也会直接在代码块下显示。如在编程过程中需要编写说明文档，可在同一个页面中直接编写，便于作及时的说明和解释。

我们跑一个项目都会先编写代码，然后编译运行，进行相应的调试，查看程序的输出情况，如果我们的代码过程比较复杂的话，单从结果的输出是比较难对代码进行调试，因为可能是中间的步骤出了错误，导致没法直观的找出错误，所以我们希望能在中间设置断点，并进行输出，便于调试代码，jupyter notebook在这方面就做的很好，而且你也可以加上文字、数学公式进行说明，可以加深对程序的认识

- 用于查看输出情况
- 用于查看中间运行过程的输出，出错时便于定位出问题的位置

所以jupyter notebook的优势是非常明显的，因此我们接下来的教程中就会利用这种交互性，时时刻刻查看输出，从而可以加深对程序的认识。

然后后面的jupyter的安装我就不讲了，这几天大家应该都安装好了，都没问题了。

然后如何进入jupyter，如果大家都进入到这个页面了，那应该都没问题了，在终端输入jupyter notebook，就可以进去当前文件夹下的notebook

这个就是我们的notebook的界面，最上面这一行呢，是这个文件的名称，你可以双击它对文件进行重命名，后面这个是保存时间的相关信息。

下面这一栏是一些功能栏，可以download，还有查找和替代，清除所有输出等等，最下面的这一栏也是一样的，常用的就是保存、添加一个代码块、运行。后面这个markdown和代码说明了当前的代码块的格式，比如这里是文字说明部分，使用markdown语法写的，我们双击以下，这就是最原始的信息，然后跑一下，就会渲染成文字说明。然后这个左边带有In符号的就是代码格式了，是可以运行的部分。

这些大家自行去探索了

还有我们的code一般是用python的代码，当同时你也可以调用shell的命令，前提是你在这个代码块这里的开头加上%%bash，比如我这里调用shell，

conda - forge：

首先这是一个组织，维护了可供使用的conda recipes，字面上理解是菜谱，就是一份相互关联的可打包在一起的东西。具体使用起来，就是推荐红色圈中的命令，首先加上conda-forge，然后安装所需要的工具包。这是最推荐的安装包的方式，因为是社团首选的一个可以使用的包，版本关联性等都是经过测试的。

2. Python 学习

因为有部分的同学可能对python不太熟悉，所以这个教程我们会给大家简单介绍以下关于python的变量类型、还有查看数据类型；创建python的变量，查看值以及类型，同时也是最重要的一点：就是了解python的index索引规则，同时也会简单的讲讲for循环、ifelse等等控制结构，最后还会涉及定义函数以及面对对象Object和继承的一些内容，这些都是很基础的内容了。也方便与对后面课程的理解。后面的课程基本上都是python编写的。

OK, 我们的第一部分就是python的数据类型，python是动态语言，所以不需要指定变量的类型。

- List

列表是最常用的Python数据类型，它可以作为一个方括号内的逗号分隔值出现。

列表的数据项不需要具有相同的类型

创建一个列表，只要把逗号分隔的不同的数据项使用方括号括起来即可。

第二部分就是python里面的变量名还有值。

python的变量名的命名规则和其他的语言大同小异了，名称由字母数字下划线组成，数字不能当

开头，同时区分大小写。

在python里面，我们定义变量的时候，是不需要指定数据类型的，数据类型我们想怎么改就怎么改，虽然我们没有定义类型，但我们可以用type来查看变量的数据类型

还有一个点，python里面我们的变量名是会对应于一个内存地址，有点像指针，比如这里有一个例子。。。。。。他们两的变量对应的地址是完全一样的，所以你改其中一个的值，势必会影响到另一个，因为他们两是完全一样的。原因就是我们在赋值的时候它们两个指向的内存地址都是同一个。

第三部分就是我们的python的索引

这里就是直接用range函数构建一个数字的list，从10到19，不包括20，这里有一个很重要的地方，就是python的索引是从0开始，这个相当重要，重要的事情要说三遍，

然后我们可以用type函数来查看变量的类型，也可以对list进行索引操作，比如这里，用0来索引第一个值，1就是第2个值了，5就是第6个，如果是-1就是倒数第一个，-2就是倒数第2个了，也可以对list进行切片提取，这里3到7就是第4个到第7个元素，右端是不包括的。如果不指定端点值的话，就是默认从开头或者结尾

第四部分 就是for循环 还有一些ifelse控制结构

python的控制结构有三大结构，分别是顺序结构、循环结构、分支结构。顺序结构其实我们以及讲了，上面那些一条一条命令分别执行的就是顺序结构，然后接下是循环结构和分支结构，循环这里一般用for循环或者while循环。

这里还有python支持列表的缩略创建，也就是列表生成式，是Python内置的非常简单却强大的可以用来创建list的生成式，具体就是在生成list的是把一些for循环和ifelse结构放到生成list的过程中。比如这个例子。。。。同样你也可以嵌套更多结构，当不建议这样做，这样会降低代码的可读性，不方便查阅。

第五部分就是python中很常用的定义一个函数

用function可以很大程度减少重复代码，我们也称之为模块化编程，可以减少工作量，这里我们定义一个打招呼的函数

这一个部分就是选学部分，因为我们后面的课程代码里面也不涉及到对象，在python里面所有的东西都是对象，比如：数据类型就是一个对象，我们是这样定义对象的，用class类，这里class employee，后面的def function都被称之为这个对象的方法，这个对象会有自己的参数，这个init方法就是来把它自己构建起来的方法，比如这里的，这个对象有名字和工资两个属性，只要实例化这个对象，就会自动执行这个方法，然后就有这个属性了。比如说我们把这个对象实例化，变量名较老张，这个老张有两个属性，名字和工资，老张的名字是老张，工资是一万。

我们前面建立的时候也定义了一个方法，叫fire炒鱿鱼方法，只要执行这个，这个对象的工资属性就会变成0，然后这里我们把老张给炒了。

对象这里有一个继承概念，比如这里我们建立一个老板对象，把它要继承的对象放括号里，然

后老板对象的构建继承员工的构建，并增加监督员工这个属性。同时也定义一个firefunction,就是把员工都执行fire，工资变0，同时老板也继承fire并对老板执行。

3.Numpy部分

numpy在科学计算的时候是非常常用的库，这里我们也是简单介绍我们后面会用的一些numpy的属性，这个教程里面会简单介绍创建numpy数组，numpy数组的两个重要属性，分别是形状和数据类型，同时也会讲解numpy数组的索引、切片提取，还有一些筛选操作，最后还有就是我们之前也提到过的，内存地址的问题。还有numpy和一些其他计算库或者画图包的一些结合。

首先我们这里导入numpy库

最后面这里一定要对这个一维数组进行转置，但是正常的转置在python里面是不起作用的，我们要用到reshape把它变成二维数组来实现转置，然后这样乘完之后就会变成二维数组，这个plt.imshow可以画一个二维图，并指定颜色，这里我们用了默认的颜色。

4.shell部分

这个部分我们会简单介绍利用shell的命令来切换工作目录，以及查看工作目录，还有对文件夹或者文件的操作，比如创建一个文件夹，对文件进行复制移动删除等等操作。同时也会大致讲一些文本流的操作命令，这里面也会涉及到pipeline管道还有重定向的概念。

我们主要分成这几个部分来讲，分别是对目录操作、文件操作、文本操作。以及shell里面常用的命令，还有管道和重定向。

首先，可能有些同学对shell不太熟悉，Shell翻译过来就是壳，壳，是连接计算机系统核的桥梁，相当于我们在更高层的位置控制命令计算机。其实shell就是用来命令我们的操作系统的程序，一种脚本语言，在计算机终端或者控制台，用命令来操作，比如window里面的powershell，这些shell脚本语言都为我们，也就是系统管理员提供了对系统控制的功能。

shell里面有好多种类了,有Bash、Csh、Zsh等等，比较常用的就是bash，在大多数的类Unix系统上都可以操作bash, 包括linux和mac os都把它当成默认的shell，所以运用还是相当广泛的。我们后续的课程需要进入到服务器里面执行相关代码，所以提前讲解以下shell的命令还是很有必要的

首先ls命令，这个命令做的事情就是把相关文件给列出来，可以是任意地址的文件，比如这里后面什么都不加，就是把当前目录下的文件全部都列出来，ls也经常和其他一些命令结合在一起，可以达到跟高效率的运作，ls后面也有一些选项可选-al，列出文件权限等详细信息，也可以使用linux下的通配符，*号通配所有字符，-sh显示文件大小，方括号这里通配括号里面的字符，问号通配单个字符。

ls 有一大堆的选项，我们很难也没有必要全部记住，可以随时用man命令和--help来查看说明。也可以在网络上搜索。

pwd这个命令也十分常用，pwd会打印出来当前的目录位置，我们在复制文件到另一个比较远的位置的时候会经常用到pwd，

这里有一个点，就是相对地址和绝对地址的区别，绝对路径是从根目录开始的，相对路径是从当前的目录开始，也可以这么说：相对路径是相对当前的目录位置，绝对路径是相对根目录的位置。

绝对路径虽然准确无误，但是太长了；相对路径比较简便，但是它终究是相对于当前目录，有时候shell脚本里面对目录操作多了就会造成目录地位混乱。所以各有优缺点。

在相对地址里面，一个点就是当前目录，两个点就是上一级目录。

接下来cd就是切换当前工作目录，绝对路径和相对路径都可以使用。

我们一般都会使用相对路径会简便一点，如果需要调用一些数据库或者程序包的位置，我们会用绝对路径

这里是对文件进行操作，包括创建、复制、移动和删除。

这个mkdir就是maker directory，后面接你想创建的文件夹名

然后就是copy命令，复制文件或者文件夹都行。我们的copy命令经常用于脚本的修改和备份，对于运行脚本我们经常会该参数去跑，这个时候不想更改原脚本，所以可以用git或者在原脚本通过注释的方式保存原脚本，常用copy命令，直接复制一份来备份更方便。

cp命令格式是这样的，cp 后面接两个文件或者位置，第一个是被复制的文件，后面是目标地址，同样也可以带上文件名，这样复制过去之后就会用这个新的文件名。所以copy命令还是很灵活的，用cp命令可以复制任何文件到任何位置。正如我们之前讲的，cp命令也可以用于重命名，就是你复制完之后，把原文件给删掉。

然后就到move命令，就是对文件或文件夹进行移动，mv命令常用于重命名

remove

文本操作，主要三个cat、tail、head

后面两个命令看名字就可以知道什么意思，这个cat猫，其实是concatenate，应该连结的意思，它是用于把文本打印到终端。cat命令很方便我们查看一些数据文本等等，因为是直接在终端操作，不涉及占用系统资源。

然后是tail尾部，从最后一行开始打印文本内容，有时候一些输入参数数据文本，我们会从末尾去查看相应的值。

head，和tail一样，只不过是第一行开始打印文本

其实还有一些其他的常用命令，比如clear，清除屏幕，通常在服务器终端，我们得到我们要的结果后就可以clear清除一下终端页面，更简洁好看一点。

还有tar命令，用于压缩和解压文件，有主要的一些选项，-zcvf 创建压缩文件，-zxvf解压文件。创建压缩文件的时候把想要压缩的文件名放在后面，前面接压缩后的文件名称，同样也可以加上文件地址。解压直接接希望解压的文件就行了，然后会解压到当前文件目录。

最后就是我们的一个很重要的命令符号，管道符号，这个符号的作用就是把管道前端也就是左

边的命令输出流进后端命令也就是右边命令的输入，从左端流入右端的过程。

linux里面命令的输出都可以当成文本流，所以可以看到管道符号极大提高了shell脚本的效率。这是一个非常好用的命令符

重定向就是给文本流指定终点位置，把输出内容给一个文件或者设备，常用重定向符号有两个，一个大于号就是输出到后面的文件，并覆盖，两个大于号就是在文件末尾追加。

我们可以看到这个管道和重定向还是有点类似的，但是也是有很大的区别的，管道是把命令的输出转换成后面命令的输入，像流水一样向后传递。重定向则是命令的输出进入到文件，就像水流进了水池就结束了，不涉及到后续的命令的输入。

所以在这个教程里也看到了，shell脚本在linux系统是默认shell，是系统原生，不需要我们去安装或者配置，而且我们地球物理一些计算经常要跑到服务器里面去，这些服务器通常没有图形界面，所以我们需要用shell来帮我们对文件、或者脚本系统进行管理和操作。

同时我们也看到，利用一些简单的命令的组合，同时，结合管道、awk等等命令，可以执行一些非常复杂的操作，这些操作如果用python来写的话，可能要写很长的代码，而且在linux系统里面用shell脚本运行的效率也比运行python要高，所以shell的意义就会显得比较重要，会提高我们科研的效率。

最后一个部分就是SAC部分讲解

研究地震学的同学应该都知道SAC这个程序包，这里我们简单介绍一些SAC里面常用的操作，比如读写、查看地震数据文件，还有修改地震数据的头文件信息，还有一些去均值、去仪器响应等等的的数据预处理，最后还有通过脚本来调用SAC程序。

首先，SAC是用来处理天然地震数据的程序包，也是运用相当广泛的程序包，可以对地震数据进行处理、画图分析等等。同时sac也是一种地震学上地震数据存储的标准格式之一。由于考虑到sac安装上面比较繁琐，不同系统安装也不一样，所以这里我们只简单介绍一些相关操作，不做程序的演示了，后续的课程在地震学上数据处理上会用到python的obspy这个包来实现。

sac里面读数据就是read，简写一个r，写命令就是一个w，

这里的dg就是data generator。这里是写了三个地震数据在内存里，用w可以把它们2存在当前的目录下。同样，sac里面一样可以用绝对路径和相对路径，以及满足通配符，? 通配一个非空字符，还可以匹配方括号里面的字符，*号就是匹配任意字符。这里执行一次read就会把数据读进内存，执行另一次的话，就会把上次读的数据内存释放，读进新的数据，这里要注意一下。

后面是sac的画图，常用的有这几个，plot、plot1、plot2等等，这个poltok是用来拾取震相用的。画图的时候一般用简写，比如p，一个p就是一次只画一个数据，并用回车向后切换，p1就是同时把所有数据都画在一个窗口里，有时候我们检查波形的时候会一次性读很多个波形数据，这样密密麻麻在一张图里很难看，所以p1有个选项控制一次绘制波形数量，并用回车向后切换。P2就是把所有波形都画在一个坐标轴上。

还有可以查看数据里面的头段变量。

头文件里面包含了这个波形数据的元数据，包括台站信息、地震事件信息等等，一般用lh命令来执行，读进去数据之后，用lh看头段变量，比如这里看开始和结束时间。

这里kstnm是台站名称、台站维度、台站经度、台站高程、台站相对于地表的深度。

这些头段变量我们也可以用ch命令去更改，要先读进数据才可更改，把想更改的变量写在前面，并把改的值放在后面。这些操作都是对与所读进内存的数据而言的，所以我们可以指定修改内存里的第几个文件。

更多的关于头段变量的信息大家可以去看sac的手册或者点这里也可以。

ok,除去一些查看波形数据的操作，我们也可以对地震数据进行处理，比如一些很常规的地震数据的预处理：去均值、去趋势、波形尖灭等，这些操作都是一个命令就可以完成。还有去仪器响应。这里用到的响应文件时respond方法，同样也可用零极点文件方法，规定好对高频和低频压制的参数

SAC里面的编程和处理过程比较缺乏一些实用的功能，所以我们会在shell或者其他的一些脚本里面调用SAC，从而提高SAC的灵活性。

比如在bash里面调用SAC直接用sac就可以进入sac的环境里面，但同时要加上这个<<EOF，结束的时候一定要quit出来，并加上EOF,两个EOF中间的内容会被传递给sac并执行。结合bash的控制语句，比如判断还有循环，我们就可以方便对数据进行批量处理。

同样，不只是在bash可以调用，也可以在python还有Perl里面调用，但是，都不如bash调用sac好用，bash方便对于文本流的操作，以及一些文件的操作。