第十二章 数据处理工具使用实验

本章主要介绍 R 语言的基本语法等知识，也简要介绍了一些主流数据处理软件，包括 Matlab 和 Python。Python 的学习可以参考文献[17,18]，Matlab 的深入学习可以参考文献[19]。Matlab 软件的下载较为困难，但是由于是商用软件，更容易学习、掌握和使用。本章的编写主要参考了文献[20]，对统计分析等理解和掌握可以仔细阅读一下该文献资料。

第一节 主要数据处理工具软件介绍

当前社会随着信息化程度的加深，各行各业都会产生大量的数据，从这些数据中分析和挖掘出有用信息为管理决策和生产制造服务已经成为基本手段，而这些数据的处理显然需要借助基本的工具软件来进行。目前比较流行的数据处理软件有 Python、R 语言、Matlab 和 SAS 等。本教程主要是给大学四年级非计算机专业学生使用的，因此仅仅介绍一些最基本知识，具体深入了解需要上网查找或者阅读这些数据处理工具软件的书籍。

一、matlab 语言

Matlab 是 matrix 和 laboratory 两个词的组合，从这个命名可以看出来，它主要为数学计算服务，也就是为数据处理而发明的语言。我们都知道，既然是计算机语言，必须有存储数据的变量，基本的表达式形式，循环和控制语句，函数和常用的函数库。Matlab 设计目的是为了数据处理，因此其代码编写特别简洁，比如两个向量 a 和 b 相加，就直接 a+b 即可，也就是采用向量运算形式。此外还提供大量的 toolbox，涉及到几乎所有的学科门类，用户直接调用这些 toolbox 的函数就可以进行相关数据处理了。图 12.1 就是 matlab 语言的开发集成环境。左侧是命令行窗口，所有的 matlab 语句可以直接在提示符后面写出，回车立刻运行，运行的变量值显示在右侧的工作区。对数据的操纵非常容易，直接写出数据变量名称即可使用。由于 Matlab 是付费的，因此有专业人员维护，各个工具包之间函数库命名统一规范，故没有函数命名冲突的困扰，直接使用工具包的函数名即可。正是由于付费软件，提供的工具箱帮助文档以及 demo 示例非常完善，十分容易学习应用。Matlab 一个文件对应一个函数，只要这个函数设置在 matlab 搜索的路径内，就可以直接调用，图 12.2 是调用一个类的静态函数 fitDrugRelease.finalModelVersion，实现对一维样本数据多数学模型的非线性拟合图形，可以看出图形精美，函数调用等十分简单。

应该说 Matlab 相比较 R 语言和 Python 最适合数据处理，这无论从代码编写的简洁性，方便性和效率方面都是如此，但是 Matlab 最大缺点就是付费，由 Matlab 公司开发发布，因此使用成本很高，目前国内很多个人用户使用 matlab 都不是正版的。另外一个缺点就是由于不是免费的，造成用户群体相对较小，因此新兴学科和新出现的算法等等一般不会使用 matlab 语言编写，造成 matlab 的工具函数包与 R 语言以及 Python 相比不够丰富。

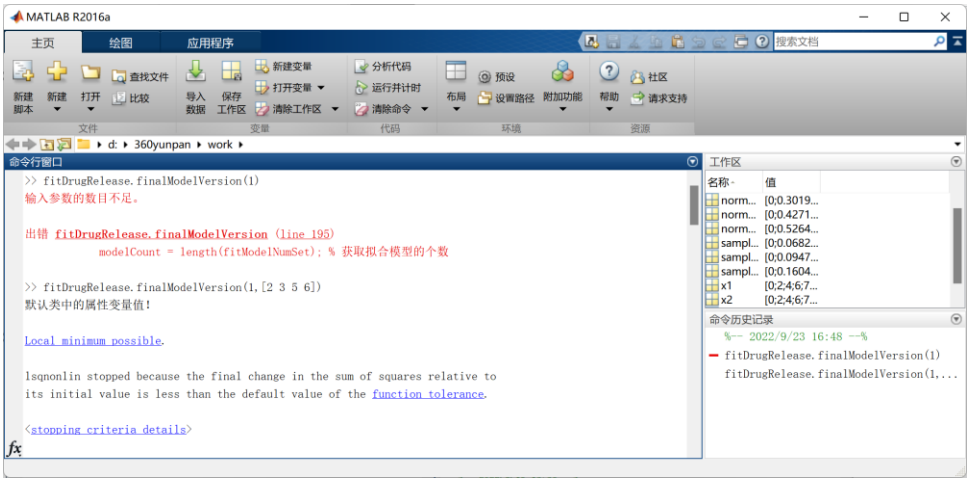

图 12.1 Matlab 开发集成环境

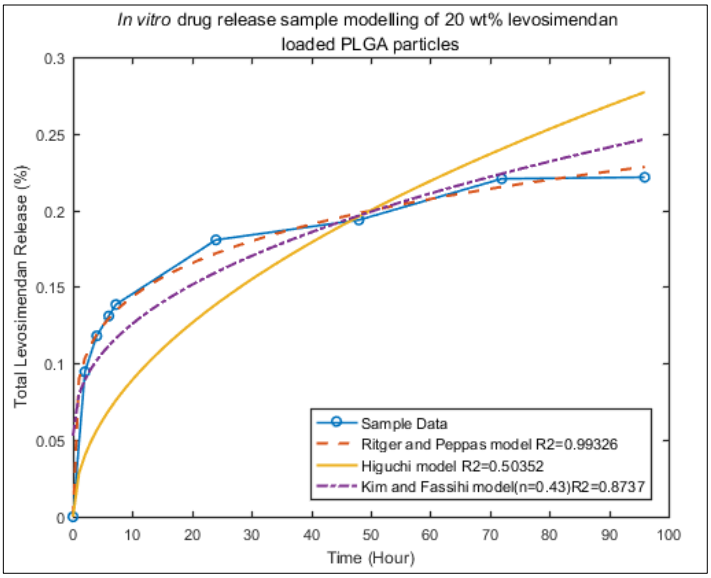

图 12.2 一维数据三个数学模型的非线性拟合

二、R 语言

Matlab 是付费的，由此带来一系列问题，于是一个免费的，有 Matlab 同样功能的数据处理软件越来越受欢迎，这就是 R 语言。R 语言也是专门用来数据处理的语言，因此它的数据类型，数据运算都是为数学计算而设计的，故和 Matlab 一样对数据处理具有方便、简洁和实用的特点，另外一方面，由于它是免费的，全世界各行各业的研究和开发人员都为其开发工具包，故其工具包十分丰富，而且几乎每周都有很多工具包生成。事实上,R 语言就是专门为统计分析设计的，因此对于经济统计学等专业的学生来说，掌握 R 语言是基本技能。图 12.3 是 R 语言的 Rstudio 集成开发环境，这个环境和 Matlab 集成开发环境一样，console 可以运行 R 语言语句，运行期间的变量值显示在右上的环境中。

正是由于 R 语言是免费的，工具包都是由爱好者生成和发布，因此十分庞杂，学习和应用起来困难，没有像 Matlab 那样由专门的技术人员编写使用手册，另外就是使用到的工具包，都要进行临时加载。

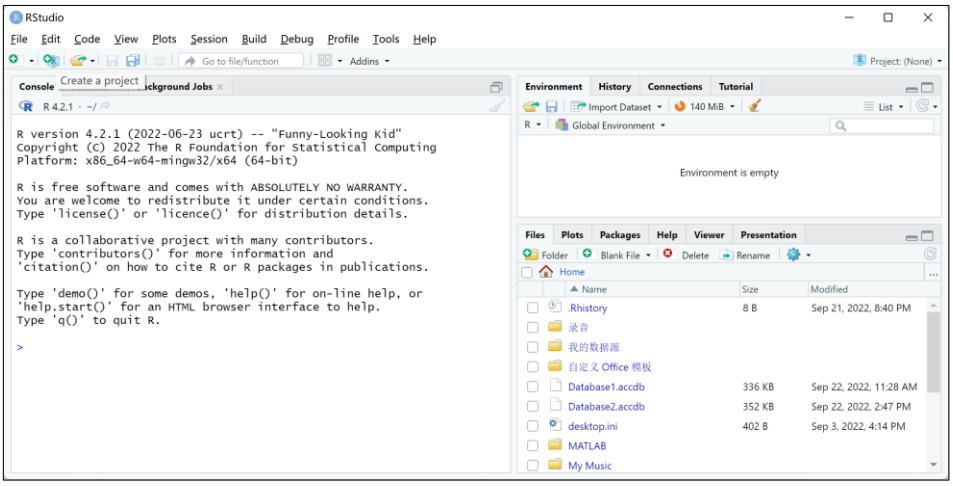

图 12.3 Rstudio 集成开发环境

三、Python 语言

Python 语言和 Matlab 以及 R 语言完全不同了，是真正意义的高级计算机程序语言，它的初衷不是为了数学计算，本身就是为了一般的程序设计而开发的，因此它和 Java 以及 C 语言一样，都能够进行网络编程，信息系统开发等等。由于它的易用性，很多开发人员使用 Python 开发了一些工具包，使得它也能够进行数据处理，这里面有三个为数学计算和数据处理而开发的工具包，分别是：numpy、pandas 和 matplotlib。 这三个包加载应用以后，就具有了类似 Matlab 和 R 语言的数据计算和数据处理的功能。当然 Python 还有大量的其它各个领域的工具包，可以直接应用，这就使 Python 成为一个应用很广泛的数据处理软件。同样的，Python 也有类似 Matlab 和 R 语言的开发环境。比如安装 Anaconda 就会自动安装 conda 和 Python 以及涉及数据处理的工具包，其中就有 Spyder 环境开发工具，它模仿 Matlab 和 R 语言的开发环境，可以编写数据处理的脚本文件等，图 12.4 就是 Spyder 开发环境的主界面，它同样具有命令行窗口(console)，工作区(workspace)和停靠 IDE 的代码编辑器。

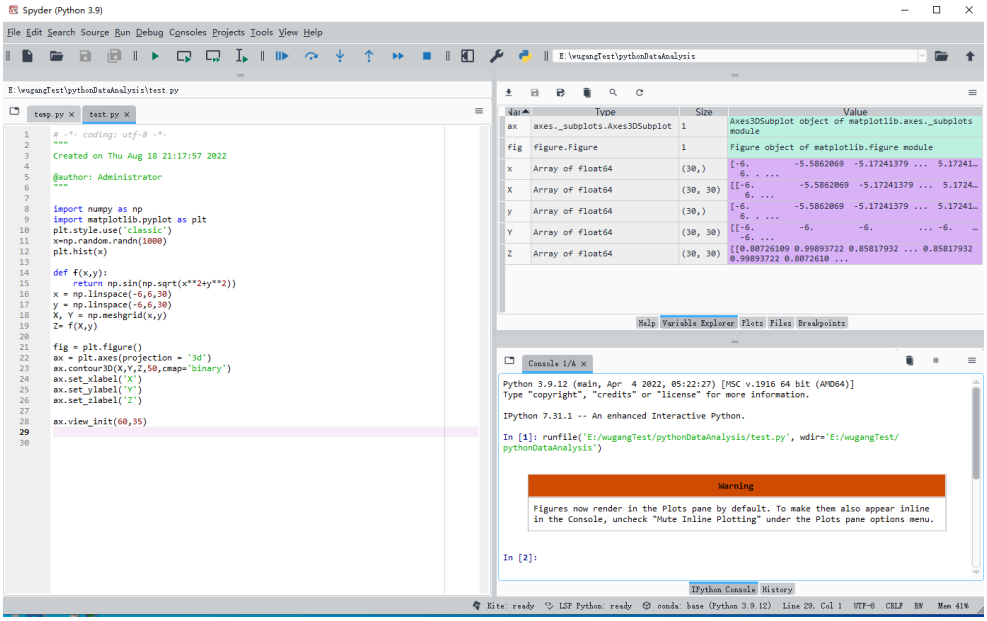

图 12.4 Spyder 集成开发环境

正是由于 Python 本身是一门高级语言，因此其语法习惯和语法格式，以及数据处理的代码编写和 Matlab 以及 R 语言相比较更为难记，难学和不好使用。比如要生成一个数组，Python 则要使用 numpy.array()来生成，这是因为 Python 任何命令都需要调用工具包里面一个对象的函数，因此十

分不方便，对初学者而言，Matlab 和 R 语言更容易使用，不需要太多面向对象的思想和知识。我们此后几章主要使用 R 语言来进行一些数据处理的实验。

第二节 R 语言基本语法介绍

任何语言的语法都包括数据结构(变量，向量，数组，结构)，表达式，输入输出，循环和逻辑判断，函数定义这几方面，掌握这些语法格式，就能完成基本的逻辑判断和数据处理了，而这些语言的区别也仅仅是采用的关键字形式不同，但是它们的思想和格式几乎都是一样的。一般来说掌握一门计算机语言，其它的计算机语言基本上简单浏览一下即可，而对于 R 这类数据处理语言，它们的特点就在于数据结构和表达式对数据的存储和运算更为灵活方便，下面简单介绍上面的内容。

一、Rstudio 的使用

图 12.3 是 R 语言集成开发环境的主页面，通过该界面，可以编写 R 语言脚本，也可以运行或者动态调试 R 语言脚本，另外可以通过命令行窗口运行 R 语言命令，查看环境变量等等。在 console 中可以直接调用函数。Rstudio 菜单 Tools 中选项 Global options 可以设置该集成开发环境的各种配置，包含工作路径和界面风格等等。

R 语言工具包是 R 函数、数据以及预编译代码等组成的集合。计算机上存储包的目录称为库，运行 library()可以查看当前系统安装了哪些包。

第一次安装包，可以使用 install.packages()来安装，可以使用 update.packages()来更新安装包。也可以使用 Rstudio 中 tool 菜单里面的 install packages 来安装包。比如安装 gclus 包，形式如下: >install.packages("gclus")，必须使用双引号，否则会报错。特别需要注意的是在使用 Rstudio 时候，选择安装包的地址时候，选择 CRAN 镜像的时候，尽量选择 Global(CDN)，选择国内镜像不是很稳定。

安装包仅仅是将包下载下来，放入 R 语言系统对应的目录下面，如果要使用它，还要加载包，一般使用 library()来加载。这时候加载的程序包，直接写包的名称，不需要使用双引号将包名称括起来。

二、数据结构

1. 向量

向量是存放数值、字符和逻辑性数据的一维数组，用函数 c()来创建。例如 a = c(1, 2, 5, 3, 6, -2, 4)、b <- c("one", "two", "three")和 c <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)。使用 a[]来访问。比如 a[3:5]就是访问向量 a 下标为 3，4 和 5 的向量值。这体现了 R 语言专门为数据处理设计的语言。当然向量中的元素的数据类型是一致的。

2. 矩阵

矩阵是一个二维数组，使用 matrix()函数来创建。矩阵中元素类型都是一样的，即要么是数值型，要么是逻辑或者字符型，不能是混合在一起，具体使用方法只要举一个例子即可。

先创建一个向量：cells = c(1,26,24,68)， myMatrix = matrix(cells,2,2)就自动生成一个 2*2 的矩阵，默认按列排列，访问时和向量一样使用中括号以及里面的下标来进行。如果 cells=1 是一个单个数值，则 myMatrix 就是一个 2*2 矩阵，每一个矩阵元素的值就是 1。注意矩阵和数组都可以有行名称和列名称。

3. 数组

数组和矩阵类似，只是维数可以大于 2，使用函数 array()来创建。例如 z = array(1,c(2,3,4))即可创建一个三维的数组，由 4 个 2*3 的二维数组组成，每一个元素都是 1。

4. 数据框(dataframe)

数据框是 R 语言是最常用的数据结构，因为上面三种数据结构的元素只能是单一类型数据，但是数据框可以存放各种类型数据。使用函数 data.frame()创建。列表 12.1 是 dataframe 创建数据

结构的例子。

列表 12.1 dataframe 数据结构的创建

```
patientID = c(1, 2, 3, 4)
age = c(25, 34, 28, 52)
diabetes = c("Type1", "Type2", "Type1", "Type1")
status = c("Poor", "Improved", "Excellent", "Poor")
patientdata = data.frame(patientID, age, diabetes, status)
patientdata
```

```
  patientID age diabetes  status
1         1  25    Type1    Poor
2         2  34    Type2 Improved
3         3  28    Type1 Excellent
4         4  52    Type1    Poor
```

列表 12.1 表明,dataframe 的创建是通过将不同类型的向量作为该数据结构的元素创建的集合，当然向量中元素的个数要相同，比如上面的病人 ID、年龄、糖尿病以及状态这四个向量都是有四个元素，那么在一起创建成一个列表。在 dataframe 中，向量元素是按列排列的，同时 dataframe 还将向量的变量名称也保留下来，成了表头。和前面讨论的向量和数组一样，使用中括号和下标来访问其中的元素。这里面还有一个与其它语言不同的概念，也就是在创建的 patientdata 中，也存在变量的概念，即 patientID、age、diabetes 和 status 都是 patientdata 中的变量，其实学习过面向对象的同学都知道 patientdata 就是一个对象，而 patientID、age、diabetes 和 status 就是属性变量，那么当然也可以使用变量名访问其值，具体的操作符不是 Java 那样的点操作符，而是$，也就是 patientdata$age 这种形式，返回的结果就是 age 这个向量，其效果和 patientdata[,2]一样的。但是使用 patientdata[2]返回的是一个只含 age 向量的 dataframe 类型的数据结构，还是有区别的。patientdata[[2]]也可以返回第 2 列向量。

总结一下就是 patientdata[n,m]返回的是第 n 行第 m 列的元素，patientdata[,m]返回的是第 m 个向量，patientdata[n,]返回的是只含第 n 行元素组成的 dataframe；patientdata[m]返回的是 dataframe 只含第 m 行向量的 dataframe 数据结构；patientdata[[m]]返回第 m 个向量。还有一个就是 attach 和 detach 的应用,访问 dataframe 里面的向量,每次都要带上 dataframe 的变量名称，因此很麻烦，如果使用 attach(patientdata)，以后可以直接使用 patientdata 里面的向量名称访问对应的数据了。detach()函数是解除这种绑定的。

因为 dataframe 类型数据是按列排放的，也就是一列是一个数据类型，如果希望提取其中的一行数据，将这行数据转换为向量也是经常用到的。例如 patientdata 中的 patientID 和 age 都是数值类型，假如取出第一行的这两个数值成为一个向量：myVector = c(t(patientdata[1,1:2]))。这个 t()函数不可少。如果是 myVector=matrix((patientdata[1,1:2]),则可以直接转化为数值矩阵。

5. 因子(factor)

因子数据类型就是将没有顺序的枚举型向量按照首字母或者指定顺序给以整数编号。比如上面例子中的 status 就是一个枚举型的向量，生成 factor 变量: test = factor(status)，test 就是 factor 变量，按照字母顺序排序了

6. 列表(list)

列表也是一种数据结构，它类似 dataframe，但是比 dataframe 包容性更大，dataframe 只能存放向量的集合，而 list 可以存放任何数据结构或者对象的变量。其访问存放的各种数据结构或者对象方式和 dataframe 一样，可以使用下标访问，也可以使用变量名称访问。具体示例如列表 12.2 所示，可以将任何向量，矩阵都存放到 mylist 这个列表变量中，存放时候还可以给它取名字，比如 title=g，age=k，而对于 j，k 都没有取名称，访问的时候特别注意，比如 mylist[2]返回的是一个含有向量 h 的 list 数据结构，而 mylist[[2]]才返回 h 这个向量。

列表 12.2 list 数据类型使用例子

```
> g = "My First List"
> h = c(25, 26, 18, 39)
> j = matrix(1:10, nrow=5)
> k = c("one", "two", "three")
> mylist = list(title=g, ages=h, j, k)
```

三、数据读取

R 语言将数据读入内存提供了很多方式，此处只介绍简单的读取文本。该读取函数的语法格式就是：mydataframe = read.table(file,option)。可以观察到，读取到的数据放入 dataframe 类型的数据结构中。Option 是选项，表示读取的文本使用了什么分隔符，文件第一行是否包含了变量名等等。假设文件名称为 studentgrades.csv，文件内容如列表 12.3 所示。

列表 12.3 读取数据文件示例

```
StudentID,  First,Last,  Math,  Science,  Social Studies
011,        Bob,  Smith,  90,    80,       67
012,        Jane, Weary,  75,     ,        80
010,        Dan,"Thornton, III",65, 75, 70
040,        Mary,  "O'Leary",90,  95,       92
```

则读取语句为

```
grades = read.table("studentgrades.csv", header=TRUE, sep="," , row.names="StudentID")
```

返回结果显示得到的结果为

```
     First         Last  Math  Science  Social.Studies
11    Bob         Smith   90       80              67
12   Jane          Weary   75       NA              80
10    Dan Thornton, III   65       75              70
40   Mary       O'Leary   90       95              92
```

可以发现读取的数据呈现一个二维表，就是一般 dataframe 的数据格式，其中缺少数据直接使用 NA 填写。StudentID 作为了序号。如果数据文件里面没有表头，就可以直接读取即可了。具体语法：

```
read.table("studentgrades.csv",,",")
```

注意逗号分隔符前面有一个缺省项，这是因为 read.table 参数表中，option 取值第一选项是 header，第二个选项是 sep，因此如果缺省 header,则要添加一个缺省项，当然如果给出选项名称，则就没有顺序限制了。特别需要说明的是在数据文件中，最后一行是空行，比如列表 12.3 中，要保持第 6 行为空行，也就是第 5 行后面要有回车键，否则读取时会有 incomplete final line found by readTableHeader on…这种错误信息

四、数据运算

1. 表达式运算

算数运算符和一般的语言类似，可以查找 R 语言手册即可。求幂的是^，也可以是**。需要说明的是向量可以直接使用算数运算符进行计算，结果是对应元素的计算。逻辑运算符也和一般计算机语言一样，不过运算结果是 TRUE 和 FALSE。

2. 数据类型转换

数据类型转换在数据处理中也很有用，用的最多的是将数值型转换为字符串，这样方便输出显示，可是使用 as.character()将其它类型的数据转换为字符型。检测某个数据是否为字符型，可以使用 is.character()，是否为数值型为：is.numeric()等等，都可以在通过百度搜索引擎随时查看。

3. 数据集合并

数据集合并也经常用到，可以使用 cbind(A,B)将对象 A 和 B 按列横向合并，比如 A=c(1,2,3)，B=c(4,5,6)。那么 D=cbind(A,B)生成的 D 就是两个向量按列依次排列，生成一个 3 行 2 列的数组。E=rbind(A,B)则按行纵向排列，也就是生成了 2 行 3 列的数组，如果合并成一个向量，则 F=c(c1,c2)，这就生成了一个具有 6 个元素的向量。可以使用 merge()合并 dataframe 类型的数据结构。上面这些函数都不需要记忆，用的时候通过百度搜索引擎搜索一下即可。

4. 数据集的子集处理

在数据处理中，经常需要从数据结构中取一些数据，一般可以直接使用下标范围来获取，例如一个 10 行 10 列的矩阵 A，我们现在希望取 2 行到第 5 行的数据，则 A[2:5,]即可以取出，也就是给

出下标范围即可。如果要去掉 A 中第 2 行到第 5 行元素，可以使用 A[-2:-5]去除。

5. 数据集的整合和重构

一个数据集的转置，可以使用函数 t()等，还有一些其它诸如 aggregate()函数负责数据集的重构和整合。

6. 常用的数学函数

字符串处理函数很常用，比如取一个字符串的子串函数 substr(x,start,stop)。在具体应用的时候可以查找。还有其它常用的三角函数和对数函数等等都可以通过百度搜索引擎查找使用的函数名称和使用的方法以及示例。

五、控制流

控制语句最重要就两种类型，一种是循环语句，一种是判断语句。这对任何语言都必须具备的，思想也是一样的，只是具体关键词不一样而已。

1. for 循环语句

for(var in seq) statement。比如语句：for (i in 1:10) print("hello")，如果循环体 statement 有多条语句,就是用大括号将它们括起来即可。如果有步长 2,则是 for (i in seq(1,10,2)) print("hello")。注意 seq()函数的使用，它表示从 1 开始，以 10 结尾，步长为 2 的数列集合。

2. if 判断语句

if(cond) statement1 else statement2, 这种形式和大多数计算机语言一致。如果 statement1 语句有多行，也需要使用大括号括起来。

六、函数定义和调用

函数创建的语法格式

```
myfunction = function(arg1,arg2,…){
      statements
      return(object)
}
```

其中 myfunction 就是函数名称。如果要创建多个函数，可以将所有创建的函数都存入一个以.R 为扩展名的文件中，比如 test.R。需要调用这些函数时候，先运行 source("test.R")，即表示将 test.R 文件中定义的所有函数都装载进内存，然后就可以直接调用 test.R 中定义的函数了。如果 test.R 存放的仅仅是脚本，不是函数，则 source("test.R")就会直接执行脚本。需要注意的是文件必须在默认当前路径，否则要将路径写全。

七、图形绘制

R 语言绘制图形中绘制曲线最为常用，可以使用函数 plot 完成曲线绘制，使用 help(plot)来理解 plot 函数的各个参数含义。下面举一些例子。 运行下面三行 R 语句

```
dose = c(20, 30, 40, 45, 60)
drugA = c(16, 20, 27, 40, 60)
plot(dose, drugA, type="b")
```

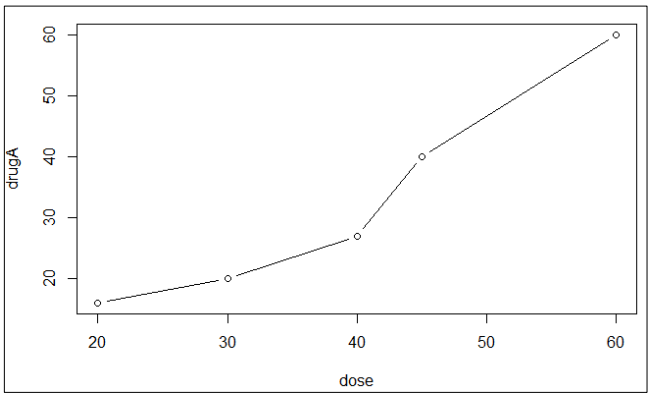

图 12.5 plot 绘制一幅曲线图

plot(dose, drugA, type="b")中，dose 表示横坐标，drugA 表示纵坐标，type="b"表示既绘制点也绘制相邻点的连线，绘制结果如图 12.5 所示。

可以通过 par()函数来修改 plot 中的属性。opar = par(no.readonly=TRUE)表示将当前的属性保存到 opar，修改完 plot 的属性以后，可以通过 par(opar)再将修改前的属性重新设置回来。具体设置那些属性，根据用户的需求进行。比如 par(lty=2, pch=17)以后，重新绘制得到的图形如图 12.6 所示。

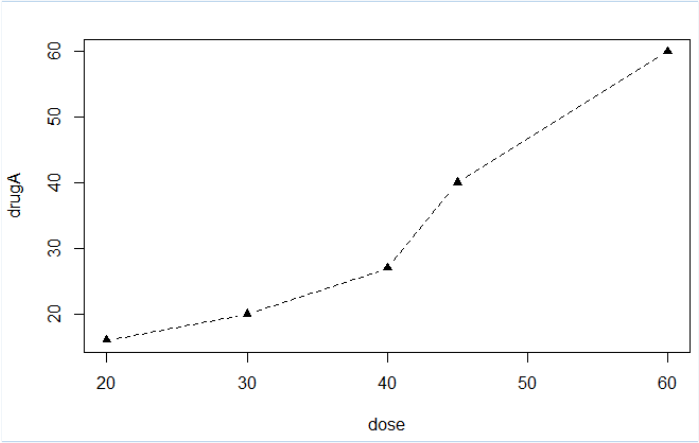

图 12.6 plot 使用 par 设置属性后绘制的图形

从图 12.6 可以看出 lty 属性表示线型，pch 表示点的形状。显然 lty=2 表示虚线，pch=17 表示实体三角形。具体还有哪些线型和点的形状，可以使用 help 查看，也可以到网上查找。具体参数含义如图 12.7 所示。

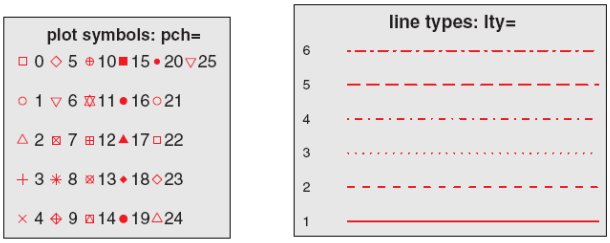

图 12.7 plot 线型和点型的参数设置

运行下面的代码，会生成图 12.8 所显示的图形。该图较为全面的显示图标题，plot(dose, drugA, type="b",col="red", lty=2, pch=2, lwd=2,main="Clinical Trials for Drug A",sub="This is hypothetical data", xlab="Dosage", ylab="Drug Response",xlim=c(0, 60), ylim=c(0, 70))。

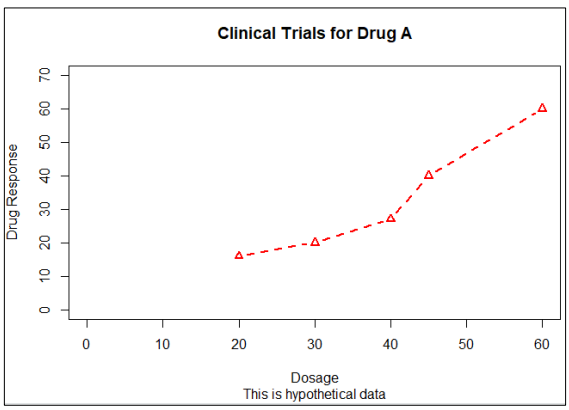

图 12.8 plot 坐标轴以及题目等的设置

观察图 12.8 很容易发现 main,sub,xlab,ylab,xlim 以及 ylim 的含义以及相应设置对图形的控制。而其中 lwd 指的是线的宽度。注意也可以通过函数 title()和 axis()对 plot 所绘制图形进行单独设置。至于图例等等设置，使用的时候，随时到百度搜索引擎上搜索即可。特别注意，如果要画多条线，可以使用 lines，不能再使用 plot，否则会创建一个新的画板，重新画。lines 是在 plot 的画板上继续画线，但是不能自动调节坐标范围，如果 lines 画的线超出 plot 的图形坐标显示范围,将无法显示 lines 画的线。使用函数 dev.new()、dev.next()、dev.prev()、dev.set()、dev.off()同时打开多个图形窗口，并决定将哪个图形输出发送到对应窗口。

第三节 综合实验十二

一、实验目的

1. 了解 R 语言、Matlab 和 Python 等数据处理工具软件基本知识；
2. 掌握 R 语言的基础语法；
3. 具有使用 R 语言实现基本数据处理的能力

二、实验平台和软件

1. 互联网环境
2. 计算机系统
3. R 语言编译系统，Rstudio ,Rtool,Matlab 和 Anaconda

三、实验内容和步骤

1. Rstudio 的使用

(1)下载 R 语言编译器，Rstudio 以及 Rtools

(2)依次安装相关 R 语言开发环境

(3)打开 Rstudio，熟悉菜单和界面

(4)打开菜单 tools，依次熟悉该菜单下的各子菜单功能以及含义

(5)熟悉 help 命令的使用

(6)使用百度搜索引擎，搜索关键字 Rstudio 查看相关网页，进一步了解 R 语言和 Rstudio

2. R 语言基本语法等实验

(1)将下面数据保存在一个 studentGrades.txt 文件中

```
StudentID,First,Last,Math,Science,Social Studies
011,Bob,Smith,90,80,67
012,Jane,Weary,75,,80
010,Dan,"Thornton, III",65,75,70
040,Mary,"O'Leary",90,95,92
```

(2) 在 Rstudio 环境中读取该文件

(3) 将该文件中的 math 和 social studies 成绩抽取出来生成两个 vector 变量，变量名分别是 mathGrades 和 socialGrades.再生成一个 vector，变量名为 studentsNo，其值为 1，2，3，4。

(4) 使用 for 循环分别显示每一个 studentNo 以及 mathGrades 和 socialGrades.

(5) 使用 plot 语句，绘制出 mathGrades 和 SocialGrades 的成绩图形，横坐标是 studentNo。

四、实验报告 (总分 8 分)

1. 编写三个函数，函数名分别为 circleArea, circlePerimeter，circleDraw，这三个函数都只有一个参数，即半径。circleArea, circlePerimeter 函数分别计算圆的面积和周长并返回计算的结果值。circleDraw 功能是根据给定的半径参数绘制圆的图形。这三个函数同时存入一个以.R 为扩展名的文件中并存放到 Rstudio 的工作路径下面；(4 分)

2.创建一个 radiusValue.txt 文件，文件里面第一行是 2,4,8,10 四个值，并分别以西文逗号分割，第二行为空行，保存后放入 Rstudio 的工作路径下面，然后使用 R 语言读取这个文件；(1 分)

3.编写一个脚本文件实现下面步骤:将读取到的数值作为半径,分别调用已经创建的 circleArea, circlePerimeter 和 circleDraw 三个函数，计算圆的面积和周长并显示其数值，最后将圆的图形绘制出来，要求使用循环语句编写。(2 分)

4.记录实验时的错误信息，分析原因，给出最后解决办法。(1 分)