



# 第01章 读取文件、 计算和绘图

顾立平

# 基本介绍



## 计算处理

对读取数据进行数学计算或逻辑处理，如Python的NumPy库。



## 编程三步曲

读取文件：从存储设备读取数据，如Python的open()函数。

## 结果可视化

使用Matplotlib等库将计算结果绘制成图形，便于理解分析。



## 应用领域

科学实验用于数据展示和趋势分析，辅助解释实验结果。商业领域中，分析销售趋势、预测股票、评估网站流量，以支持决策。



# 商业领域的应用介绍



## 市场分析的统计摘要

快速评估市场趋势，识别机会与挑战，支持策略制定。



## 销售预测中的数据摘要

计算历史销售趋势，评估稳定性，构建准确预测模型。



## 客户分析的统计应用

解析客户特征，购买行为，制定个性化营销策略。



## 商业决策的数据支持

概括数据关键信息，辅助科学决策，提升商业效能。





# 读取数据介绍



## 数据读取原理

从硬盘、数据库等导入数据到内存，作为程序处理的第一步，确保获取所需信息。



## 数据来源多样性

涵盖文件系统、数据库和网络资源，涉及数据的序列化和反序列化过程。



## 数据处理前提

读取数据是程序执行分析前的必备步骤，对各种形式和来源的数据进行处理。



# 科学领域的应用介绍



## 01

### 科学领域数据读取 关键性

涉及实验分析、大规模数据处理、科学模拟与建模，为科研活动提供基础，确保数据准确性和分析效率。

## 02

### 商业领域数据应用

包括市场分析、客户关系管理、金融风险管理与供应链管理，支持企业决策，提升运营效率和市场竞争力。

## 03

### 移除变量的重要性

科学领域运用特征选择和降维技术提升模型性能，商业领域通过数据清洗和预测分析优化决策，确保业务效果。



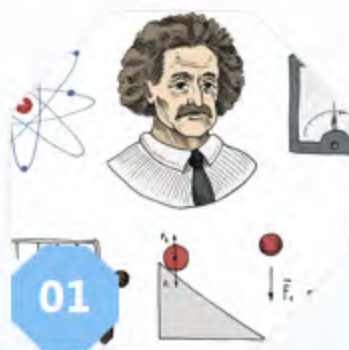


# 在科学领域的应用



## 物理学中的函数应用

牛顿第二定律、速度与加速度，通过函数揭示力、质量与加速度间的关系。



## 工程学中的函数应用

电路分析中的欧姆定律，电流、电压和电阻的关系以函数形式表达。

## 生物学中的函数应用

酶促反应速率与底物浓度的米氏方程，非线性函数解析生物化学过程。



## 经济学中的函数应用

供给需求函数分析市场趋势，成本收益函数帮助企业决策和财务评估。



# 在商业领域的应用



数据分析中，利用线性回归、指数平滑函数揭示数据关系，预测销售趋势。

## 商业领域函数应用

通过成本函数、收益函数比较定价策略，构建满意度函数评估服务策略效果。

## 市场策略制定



## 预测模型构建

基于函数的ARIMA模型用于时间序列分析，预测未来市场走向，支持决策制定。





# 在商业领域的应用



## 计算矩形面积应用

在房地产中，计算矩形面积确定房产价值，为定价提供依据。



## 建筑与工程设计

计算矩形面积帮助工程师规划结构，计算材料需求，确保项目顺利进行。



## 制造业生产优化

通过计算原材料面积，优化生产流程，提高资源利用率，降低生产成本。







## 三角形面积计算基础

介绍几何学基本任务，理解二维空间中三角形所占区域，涵盖底乘高、海伦公式和三角函数计算方法。

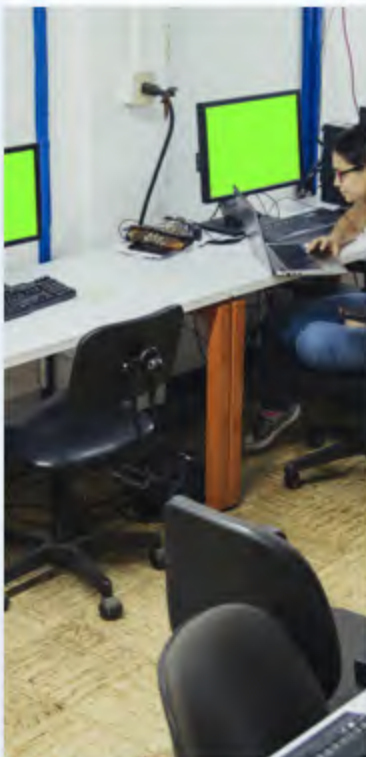
# 计算三角形面积

# 在科学领域的应用



## 三角形面积 计算应用

涵盖物理学的流体力学计算，工程学结构设计，地理学地图制作，体现其科学价值。



## 商业领域中的 应用

在房地产评估地块，建筑学材料计算，制造业优化生产中，计算三角形面积发挥关键作用。



## 分段函数概念 解析

分段函数是根据 $x$ 值范围使用不同解析式的函数，用于描述和解决涉及多个不同规则的数学或实际问题。





# 001 绘制曲线图

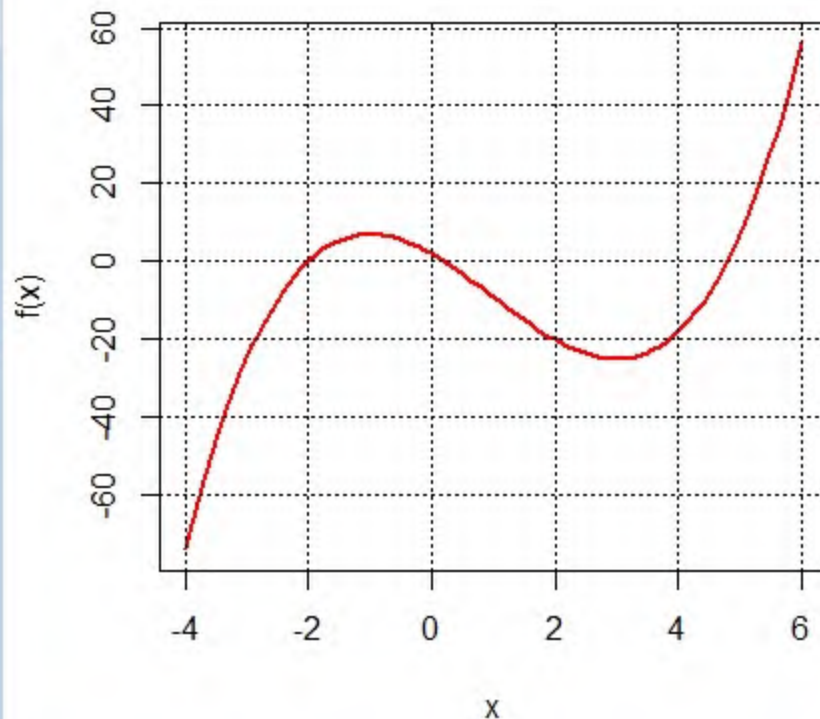


R Console

R是个合作计划，有许多人为之做出了贡献。  
用'contributors()'来看合作者的详细情况  
用'citation()'会告诉你如何在出版物中正确地引用R或R程序包。  
  
用'demo()'来看一些示范程序，用'help()'来阅读在线帮助文件，或  
用'help.start()'通过HTML浏览器来看帮助文件。  
用'q()'退出R。

```
> setwd("C:/Users/user/Desktop/code")
> 2^6*3*643
[1] 123456
>
> #####
> # 定义一个函数f，该函数接收一个参数x，并返回x的三次方减去3倍的x的平方减去9倍的x
> f=function(x){x^3-3*x^2-9*x+2}
>
> # 使用curve函数绘制函数f在-4到6之间的图像，线条宽度为2，颜色为红色。
> curve(f,-4,6,lwd=2,col="red") # lwd表示线条宽度，col表示颜色
>
> # 使用grid函数在图上添加网格线，网格线颜色为黑色。
> # 这里的两个NULL表示在x和y方向上都使用默认的网格间隔。
> grid(NULL,NULL,col="black")
>
>
> |
```

R Graphics: Device 2 (ACTIVE)







# 分段函数在科学领域的应用介绍

在科学领域，分段函数因其能够精确描述复杂、非线性关系的能力而被广泛应用。以下是一些具体的应用场景

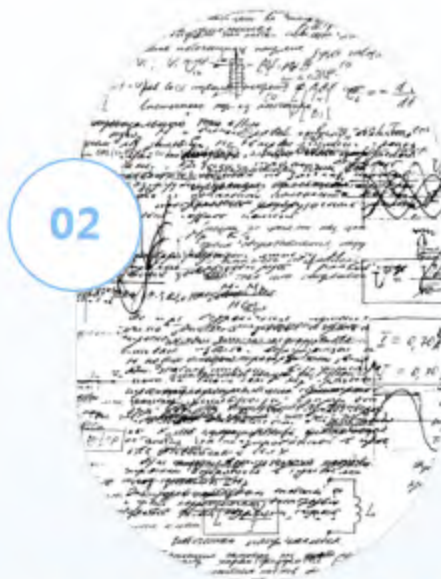


# 物理学：



## 物理现象的数学描述

分段函数用于描述物理量的变化范围，如弹簧振子振动时位移随时间的关系，不同区间对应不同函数关系，体现约束条件。

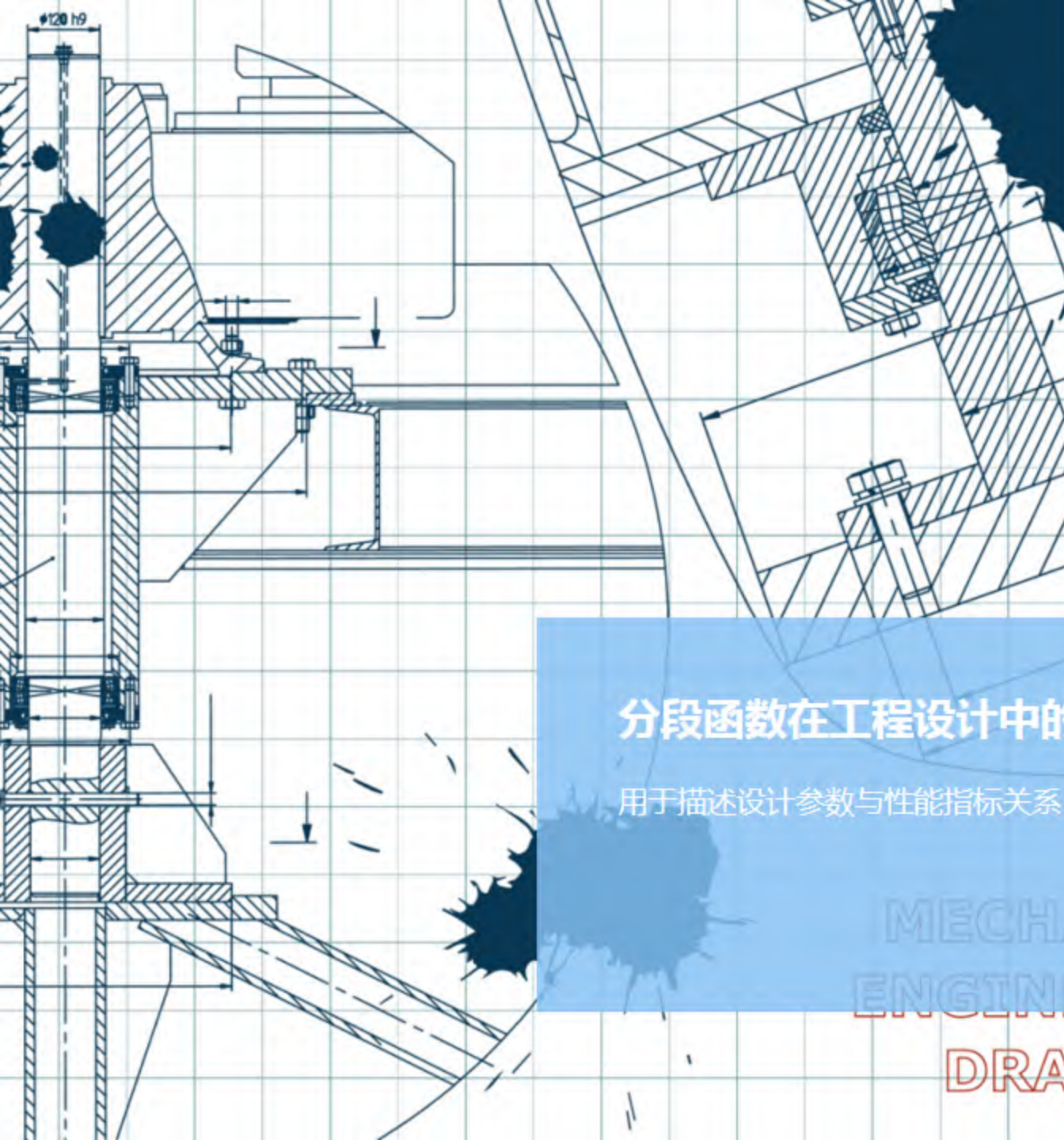


## 函数在物理研究中的应用

在研究机械振动、电磁波传播等现象时，通过定义函数的定义域确定物理量的变化规律，分段函数能精确刻画其状态转变。







# 工程学：

## 分段函数在工程设计中的应用

用于描述设计参数与性能指标关系，灵活调整以优化设计，确保可行性和稳定性。

MECH  
ENGINE  
DRA





# 生物学与医学：



01

## 分段函数在生物学中的应用

描述生理状态变化，如药物浓度与药效的关系，当浓度超过阈值时，药效增长可能不再线性。

02

## 药物浓度与药效分析

利用分段函数，体现浓度不同区间内，药效的不同响应模式，可能在高浓度时出现饱和或峰值。



## 003 读取数据



```
>
> # 注意：应该下载Arctic-Ice-Data-R.csv文件并将其放置在一个文件夹中。
> # 更改目录，使其指向包含该文件的文件夹。
> # （注意：此行为注释，实际代码中并未包含更改目录的操作，通常需要使用setwd()函数$
>
> # 读取名为"Arctic-Ice-Data-R.csv"的CSV文件，该文件包含表头，并且使用逗号作为分$
> # 读取的数据被存储在变量Ice中。
> Ice = read.table("Arctic-Ice-Data-R.csv", header=TRUE, sep=",")
>
> # 输出变量Ice中的列名，以查看数据的各个字段。
> names(Ice)
[1] "Year"                "Years.after.1970"      "March.Extent.in.MSK"
[4] "June.Extent.in.MSK"   "September.Extent.in.MSK"
```





# 分段函数在商业领域的应用介绍



## 销售策略制定

分段打折策略，购买数量不同阶段对应不同折扣，如100件内不打折，101-500件打8折，超过500件打6折。



## 业绩奖金计算

销售团队根据业绩分段获取奖金，如销售额100万内5%奖金，100-200万8%，超过200万10%。



## 信用评级与贷款

银行利用分段函数评估客户信用，不同收入和还款记录等级对应不同贷款额度和利率。



## 阶梯函数在商业应用

阶梯函数用于计算销售提成、定价策略和市场分析，如根据购买数量或客户类型设定不同奖励或价格水平。





# 什么是极坐标



01

## 极坐标概念

由牛顿创立，定点 $O$ ，射线 $Ox$ 为极轴， $\rho$ 为极径， $\theta$ 为极角， $(\rho, \theta)$ 代表平面点的位置。

02

## 极坐标发展

经伯努利、皮科克等人推动完善，定点、极轴、长度单位和角度方向构成的坐标系统。

03

## 坐标表示方式

用极径和极角描述平面内点的位置，有序数对 $(\rho, \theta)$ 即为极坐标表示。



## 物理学：



01

### 极坐标在物理学中的应用

涵盖旋转对称性现象，如磁场中粒子轨迹、电磁波传播，直观表示，便于计算。



02

### 量子力学中的极坐标

波函数常用极坐标描述，因动量、角动量等物理量呈现极坐标特性。



## 工程学：

### 极坐标系统应用

涵盖工程学多个领域，如建筑设计、机械工程和航空航天，便于描述旋转对称结构。

### 信号处理中的极坐标

在音频信号处理中，极坐标用于表示声音幅度和相位信息，简化数据分析。

### 地理学中的极坐标应用

在地图制作和导航中发挥作用，特别是在极地地区的特殊导航需求。



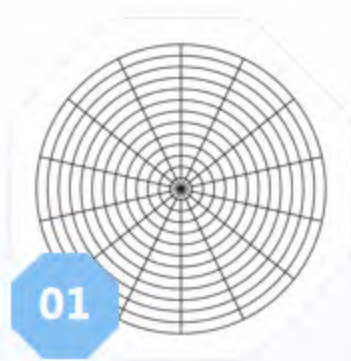


# 工程学：



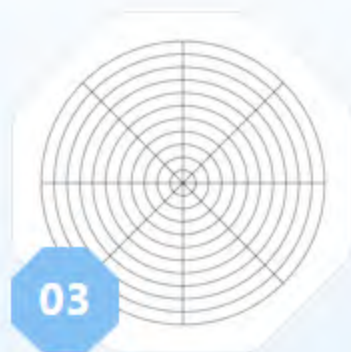
## 数学中的极坐标工具

解析几何、复变函数等领域不可或缺，  
简化圆锥曲线等复杂问题的求解。



## 市场分析中的极坐标应用

分析市场份额和消费者偏好，以极径  
和极角展示市场结构和竞争态势。



## 金融分析中的极坐标

用于描绘资产价格波动，帮助分析市  
场趋势，优化投资组合选择。



## 002 显示统计摘要



```
R Console
> # 生成150个来自正态分布的随机数，均值为0，标准差为1，并将这些随机数存储在变量ex$
> example=rnorm(150,0,1)
>
> # 显示example变量的摘要统计信息，包括最小值、第一四分位数、中位数、均值、第三四分位数
> summary(example)
      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
-2.33347 -0.70751 -0.01564  0.01917  0.70244  2.37927
>
> # 对example变量进行t检验，检验其均值是否显著小于0（单侧检验）
> # mu=0表示假设的均值为0
> # alternative="less"表示进行的是单侧检验，并且备择假设是example的均值小于0
> # conf.level=0.90表示置信水平为90%
> t.test(example,mu=0,alternative="less",conf.level=0.90)

One Sample t-test

data:  example
t = 0.23486, df = 149, p-value = 0.5927
alternative hypothesis: true mean is less than 0
90 percent confidence interval:
 -Inf 0.1242492
sample estimates:
mean of x
0.01917155
>
```





# 商业绘图与设计：



01

## 极坐标系统应用

用于创建旋转对称图案，如螺旋线、花瓣，  
常见于广告和包装设计，简化复杂图形描述。



02

## 商业设计创新

借助极坐标绘制独特视觉图形，增强美感和吸引力，有效应用于广告和产品包装，  
提升设计效果。

# 什么是参数方程



数学概念，曲线上点的 $x, y$ 由 $t$ 函数决定， $x=f(t)$ ， $y=g(t)$ ， $t$ 为参数， $x, y$ 为因变量。



参数方程定义

与直接给出坐标关系的普通方程不同，参数方程用参数表示点，某些情况下更方便直观。



与普通方程对比

在处理复杂曲线和曲面时，参数方程能更直观地描述，提供了一种有效的表示方法。



参数方程应用





## 004 移除变量

```
R Console

> 
> #####
> 
> # 移除名为Ice的变量（如果它之前存在的话），释放内存空间
> rm(Ice)
> # header=TRUE 表示该文件的第一行是列名（表头）
> # sep="," 表示数据中的字段是用逗号分隔的
> # 读取后的数据框（data frame）赋值给变量Ice
> Ice = read.table("Arctic-Ice-Data-R.csv", header=TRUE, sep=",")
> head(Ice)
  Year Years.after.1970 March.Extent.in.MSK June.Extent.in.MSK
1 1979                9          16.34          12.53
2 1980               10          16.04          12.20
3 1981               11          15.63          12.43
4 1982               12          16.04          12.48
5 1983               13          16.09          12.30
6 1984               14          15.58          12.15
  September.Extent.in.MSK
1              7.05
2              7.67
3              7.14
4              7.30
5              7.39
6              6.81
> |
```





## 参数方程在科学领域的应用介绍

参数方程在科学领域有着广泛的应用，以下是几个具体的应用场景



# 物理学：



## 物理学中的参数方程应用

涵盖行星运动轨迹、摆线、电磁波传播、机械设计、航空航天等领域，描述物体动态和结构特性。



## 商业领域的参数方程应用

产品设计中调整参数生成不同模型，市场分析预测趋势，金融产品建模辅助决策，展现多行业价值。



## 返回函数的应用于科学

数据处理定义动态流程，模拟仿真构建灵活系统，算法设计实现定制化策略，促进科学计算的灵活性和效率。



# 商业领域的应用介绍

01

## 软件开发中的返回函数

常见于框架和库设计，如Web开发中的路由处理，返回响应处理函数，增强代码灵活性和可重用性。

02

## 业务逻辑处理与返回函数

用于封装复杂业务规则，返回的函数使得逻辑组合、重用和测试更为简便。

03

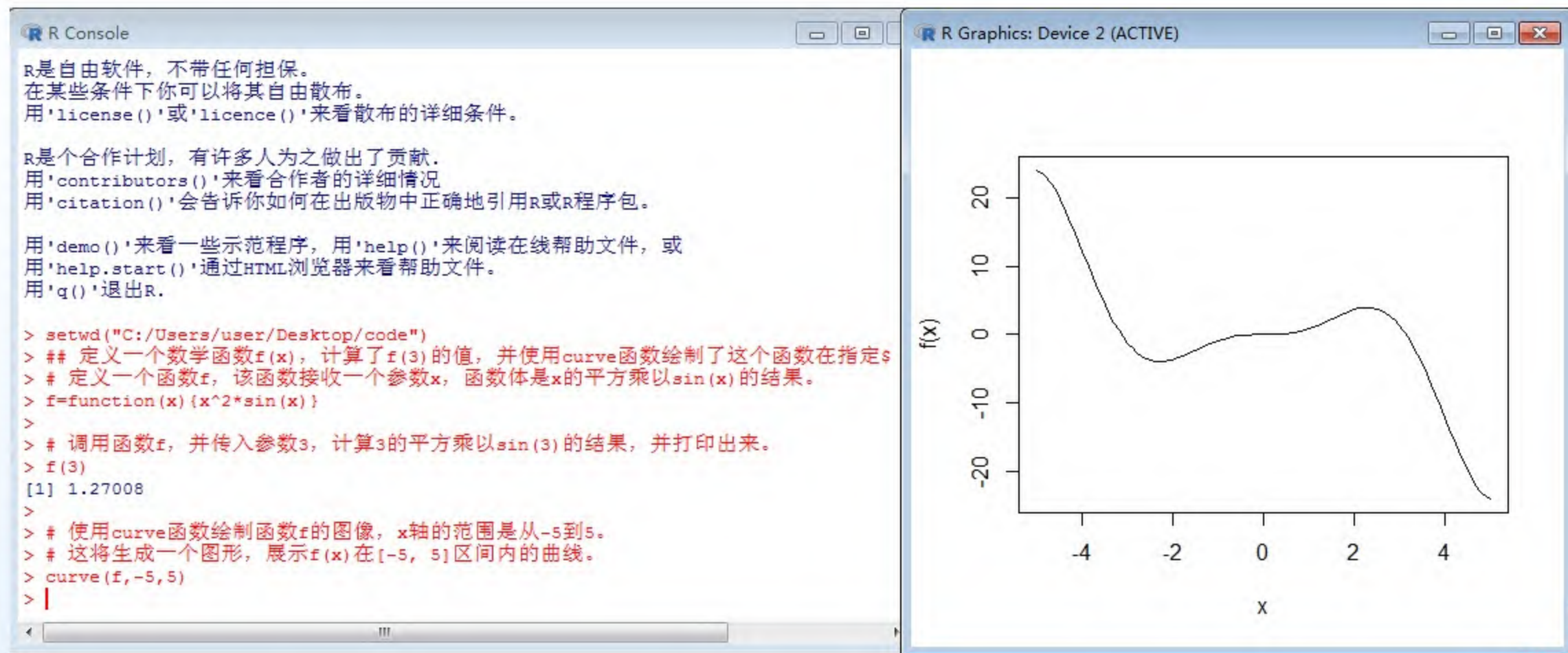
## API返回函数设计

实现回调和中间件模式，通过函数作为参数或返回值，创建更灵活、可扩展的API接口。





# 005 定义函数



## 006 计算矩形面积



```
R Console

5          7.39
6          6.81
> ## 定义一个数学函数f(x)，计算了f(3)的值，并使用curve函数绘制了这个函数在指定$
> # 定义一个函数f，该函数接收一个参数x，函数体是x的平方乘以sin(x)的结果。
> f=function(x){x^2*sin(x)}
>
> # 调用函数f，并传入参数3，计算3的平方乘以sin(3)的结果，并打印出来。
> f(3)
[1] 1.27008
>
> # 使用curve函数绘制函数f的图像，x轴的范围是从-5到5。
> # 这将生成一个图形，展示f(x)在[-5, 5]区间内的曲线。
> curve(f,-5,5)
>
>
> ## 定义一个计算矩形面积的函数，并通过具体参数调用了这个函数，得到了一个4x6矩$
> # 定义一个名为rectangle.area的函数，该函数接收两个参数l（长度）和w（宽度）
> # 函数的功能是计算矩形的面积，通过将长度l与宽度w相乘得到结果
> rectangle.area=function(l,w){l*w}
>
> # 调用上面定义的rectangle.area函数，并传入长度4和宽度6作为参数
> # 函数将返回这两个参数的乘积，即矩形的面积，此处为4乘以6等于24
> # 该结果将被打印到控制台
> rectangle.area(4,6)
[1] 24
```





## 007 计算三角形面积



```
R Console
> # 函数的功能是计算矩形的面积，通过将长度l与宽度w相乘得到结果
> rectangle.area=function(l,w){l*w}
>
> # 调用上面定义的rectangle.area函数，并传入长度4和宽度6作为参数
> # 函数将返回这两个参数的乘积，即矩形的面积，此处为4乘以6等于24
> # 该结果将被打印到控制台
> rectangle.area(4,6)
[1] 24
> ## 定义一个根据三角形的三边长来计算其面积的函数，并通过具体参数（边长3, 4, 5）
> # 定义一个名为area.triangle.sides的函数，用于计算给定三边长的三角形的面积
> # 函数接收三个参数a, b, c，分别代表三角形的三条边长
> # 函数内部使用海伦公式（Heron's formula）来计算三角形的面积
> area.triangle.sides=function(a,b,c){
+   # 计算半周长p，即三条边长之和的一半
+   p=(a+b+c)/2
+   # 使用海伦公式计算三角形的面积，并返回该值
+   # 海伦公式为：面积 = sqrt[p * (p - a) * (p - b) * (p - c)]
+   sqrt((p)*(p-a)*(p-b)*(p-c))
+ }
>
> # 调用上面定义的area.triangle.sides函数，并传入边长3, 4, 5作为参数
> # 函数将返回这个三角形的面积，对于边长3, 4, 5的直角三角形，面积应为6
> area.triangle.sides(3,4,5)
[1] 6
> |
```





# 总结



## 返回函数功能

编程工具，用于代码抽象、模块化、延迟执行和组合，提升开发效率与软件质量。



## 应用领域

返回函数广泛应用于科学计算和商业逻辑，实现复杂任务的简化和高效处理。



# 计算毕达哥拉斯三元组的函数



01

## 编程实现

通过穷举法或数学公式法生成满足 $a^2 + b^2 = c^2$ 的正整数三元组，实现毕达哥拉斯三元组计算。

02

## 几何与数论应用

在几何学中描述直角三角形边长关系，数论中探究生成规律和与其他定理的联系。

03

## 软件与游戏开发

软件开发中用于特定算法，如图形处理和数据分析；游戏开发中构建直角相关元素。

04

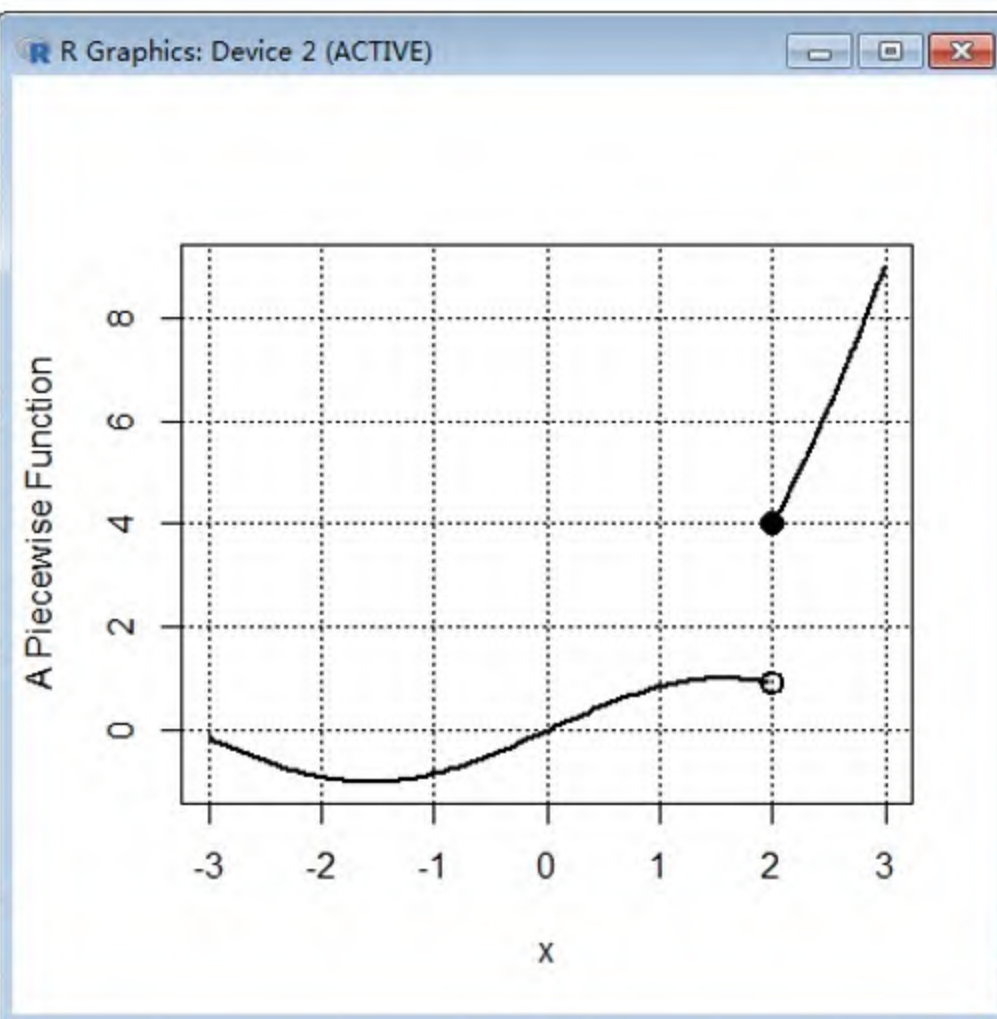
## 教育实践

教育工具中帮助学生理解和验证毕达哥拉斯定理，提升数学原理的掌握。



## 008 分段函数

```
R Console
> ## 绘制一个分段函数在[-3,3]区间的图像，并用点来表示函数值，并且清晰
> # 定义一个分段函数f.piecewise，该函数根据x的值返回不同的结果
> # 如果x小于2，则返回sin(x)；否则返回x的平方
> f.piecewise=function(x){ifelse(x<2,sin(x),x^2)}
>
> # 调用f.piecewise函数，并传入参数-pi/4，计算并返回该点的函数值
> f.piecewise(-pi/4)
[1] -0.7071068
>
> # 使用curve函数绘制f.piecewise函数在[-3,3]区间的图像，使用点来表示函数值
> # 设置y轴标签为"A Piecewise Function"
> curve(f.piecewise,-3,3,type="p",ylab="A Piecewise Function")
>
> # 在当前图形上添加网格线，网格线颜色为黑色
> grid(NULL,NULL,col="black")
>
> # 再次使用curve函数，但这次不绘制函数值（type="n"），仅为后续的绘图
> # 设置y轴标签为"A Piecewise Function"
> curve(f.piecewise,-3,3,type="n",ylab="A Piecewise Function")
>
> # 在当前图形上添加网格线，网格线颜色为黑色
> grid(NULL,NULL,col="black")
>
> # 在[-3,1.95]区间上绘制f.piecewise函数的图像，线条宽度为2，添加到当前
> curve(f.piecewise,-3,1.95,add=TRUE,lwd=2)
```





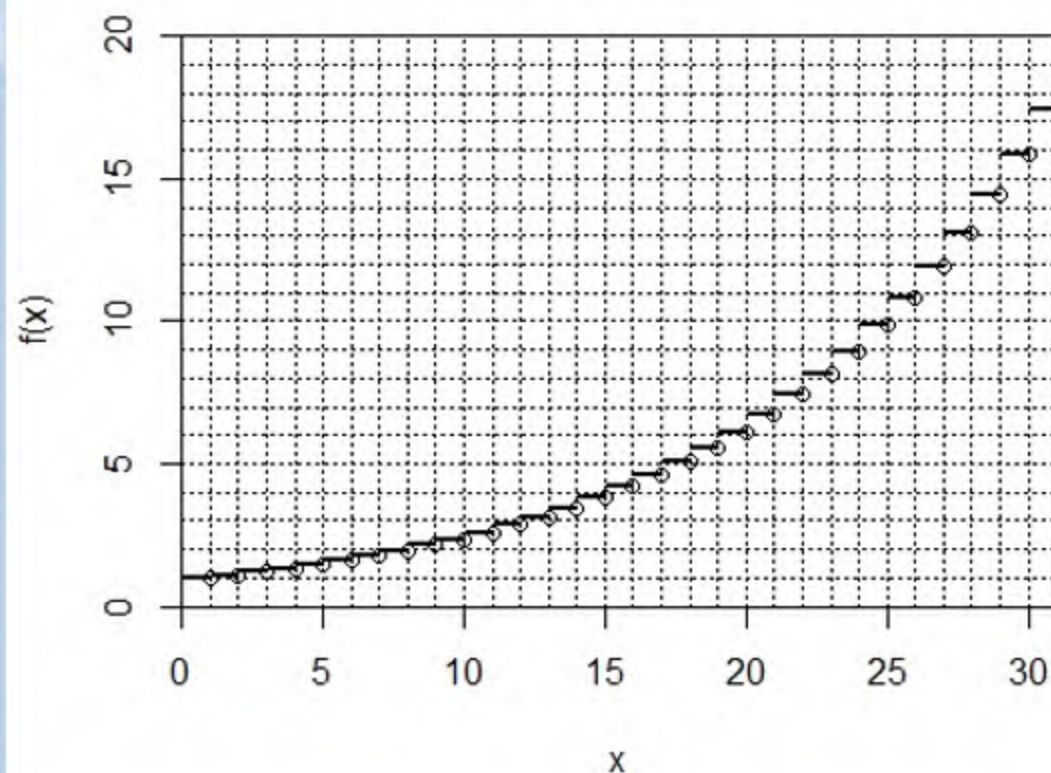
## 009 阶梯函数

R Console

```
> ## 创建一个阶梯函数，并使用plot函数将其绘制出来。  
> # 创建一个从1到30的整数序列，并将其赋值给变量x  
> x = 1:30  
>  
> # 使用sapply函数对0到x中的最大值（即30）的每个整数进行迭代  
> # 对于每个整数，计算1.10的该整数次幂，并将结果赋值给变量y  
> # 这里实际上生成了一个指数增长的序列  
> y = sapply(0:max(x), function(x) (1.10^x))  
>  
> # 使用stepfun函数创建一个阶梯函数  
> # x和y作为函数的输入和输出值  
> # right = TRUE表示阶梯函数的右连续性  
> step = stepfun(x, y, right = TRUE)  
>  
> # 使用plot函数绘制阶梯函数  
> # xlim和ylim设置了x轴和y轴的范围  
> # lwd设置了线的宽度  
> # xaxs和yaxs设置了轴线的扩展方式，"i"表示轴线恰好与数据的  
> # verticals设置为FALSE表示在绘图时不显示垂直线  
> # main设置了图表的标题  
> plot(step, xlim = c(0, max(x) + 1), ylim = c(0, 20), lwd  
+       verticals = FALSE, main = "A Step Function")  
>  
> # 使用abline函数在图中添加辅助线  
> # h参数设置了水平线的y值，这里添加了从0到20的整数对应的水
```

R Graphics: Device 2 (ACTIVE)

A Step Function





# 010 极坐标

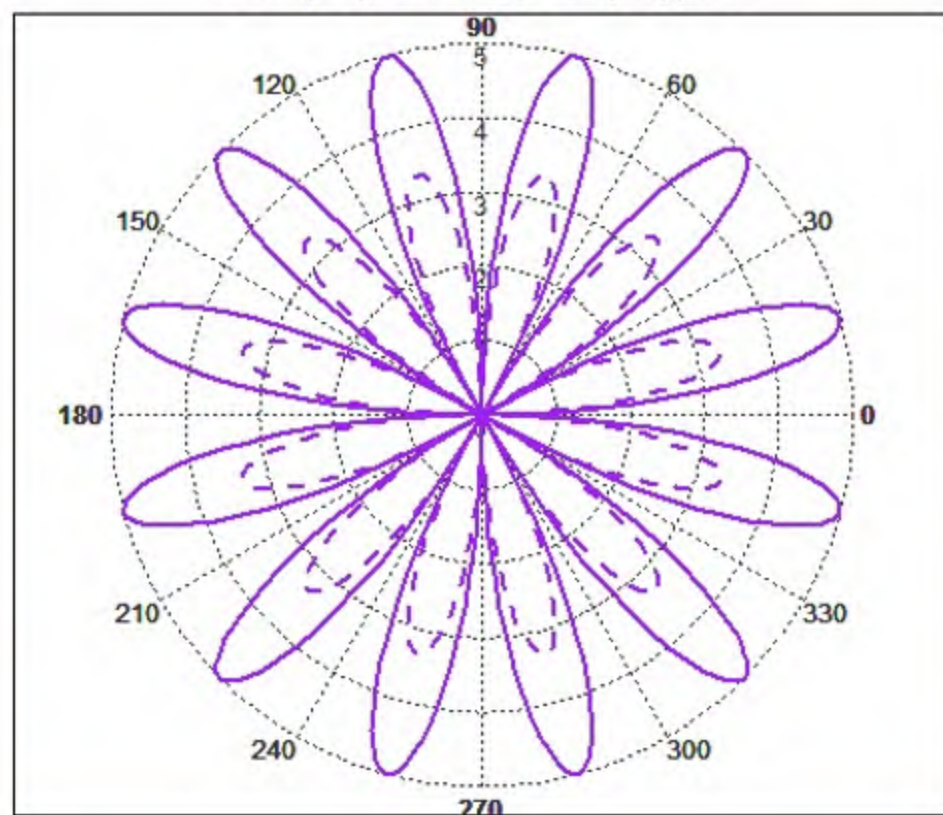


R Console

```
>
> # 创建一个从0到2π的序列，长度为360，用于表示角度。
> # 这里t实际上表示的是极坐标中的角度。
> t = seq(0, 2*pi, length=360)
>
> # 定义一个函数f，输入是t（角度），输出是5*sin(6*t)。
> # 这个函数描述了一个在极坐标系中的曲线。
> f = function(t) {5*sin(6*t)}
>
> # 设置图形参数，mar参数用于设置图形区域的边距。
> # c(1,1,2,1)分别表示图形区域下方、左侧、上方和右侧的边距。
> par(mar=c(1,1,2,1))
>
> # 使用polar函数绘制极坐标图。
> # t是角度，f(t)是对应的半径。
> # grcol设置网格线的颜色，bxcol设置坐标轴的颜色。
> # col设置曲线的颜色，lwd设置线宽。
> # main设置图形的主标题，使用expression来支持数学表达式。
> polar(t, f(t), grcol="gray20", bxcol="black", col="purple", lwd=2,
+       main=expression(5*sin(6*theta) ~ ~"and" ~ ~2*(5*sin(6*theta)/3))
>
> # 在同一个极坐标图中，再绘制一条曲线，其半径是原曲线的2/3。
> # col设置曲线的颜色，lty设置线型（2表示虚线），lwd设置线宽。
> # add=TRUE表示在当前图形上添加这条曲线，而不是创建新的图形。
> polar(t, 2*f(t)/3, col="purple", lty=2, lwd=2, add=TRUE)
```

R Graphics: Device 2 (ACTIVE)

$5\sin(6\theta)$  and  $2(5\sin(6\theta))/3$





## 011 参数方程



R Console

```
> ## 两个函数x和y都接受一个参数t，并根据给定的公式计算结果
> # 定义一个函数x，其输入是t，输出是根据给定公式计算的结果
> x=function(t){sin(t)*(exp(cos(t))-2*cos(4*t)-(sin(t/12)))}
>
> # 定义一个函数y，其输入是t，输出是根据给定公式计算的结果
> y=function(t){cos(t)*(exp(cos(t))-2*cos(4*t)-(sin(t/12)))}
>
> # 定义一个函数Butterfly，其输入是t
> # 该函数会返回一个由x(t)和y(t)组成的向量
> Butterfly=function(t){return(c(x(t),y(t)))}
>
> # 调用Butterfly函数，并传入参数2*pi，但此处并未对返回的
> Butterfly(2*pi)
[1] -1.682687e-16 6.870318e-01
>
> # 设置图形参数，mar参数用于定义图形区域的边距，c(1,1,1,
> par(mar=c(1,1,1,1))
>
> # 生成一个从0到12*pi的序列，步长为0.001，并将这个序列赋
> t.values=seq(0,12*pi, 0.001)
>
> # 使用plot函数绘制图形
> # x轴使用x(t.values)的结果，y轴使用y(t.values)的结果
> # type="l"表示绘制线图
> # col="purple"表示线的颜色是紫色
```

R Graphics: Device 2 (ACTIVE)

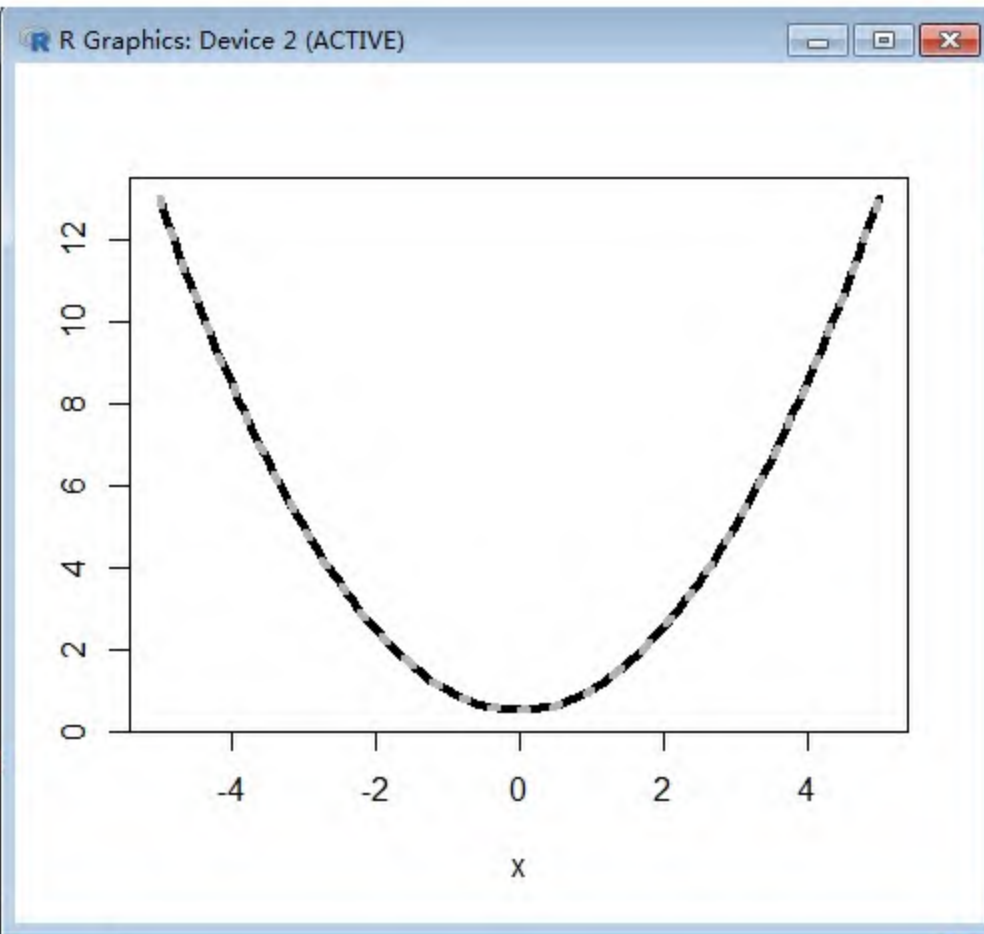




# 012 抛物线的几何定义



```
R Console
> y.max = 20
>
> # 定义一个函数g, 它接受一个参数x
> g = function(x) {
+   # 在函数g内部定义另一个函数f, 它接受一个参数y
+   f = function(y) {
+     # 函数f计算表达式sqrt(x^2 + (1-y)^2) - y的值
+     sqrt(x^2 + (1 - y)^2) - y
+   }
+   # 使用uniroot.all函数从rootSolve包中寻找函数f在区间[0, y.max]内的所有根
+   # 并返回这些根的信息
+   return(uniroot.all(f, c(0, y.max)))
+ }
>
> # 调用函数g, 并传入参数0, 得到x=0时函数g的结果
> g(0)
[1] 0.5
>
> # 使用Vectorize函数将g函数向量化, 使其能够处理向量输入
> h = Vectorize(g)
>
> # 设置图形参数, mar设置图形四边的边距, c(5,3,3,2)分别代表图形下、左、上、右
> par(mar = c(5, 3, 3, 2))
>
> # 使用curve函数绘制函数h的图像, x的范围是[-5,5], 线宽为4, y轴标签为空
```



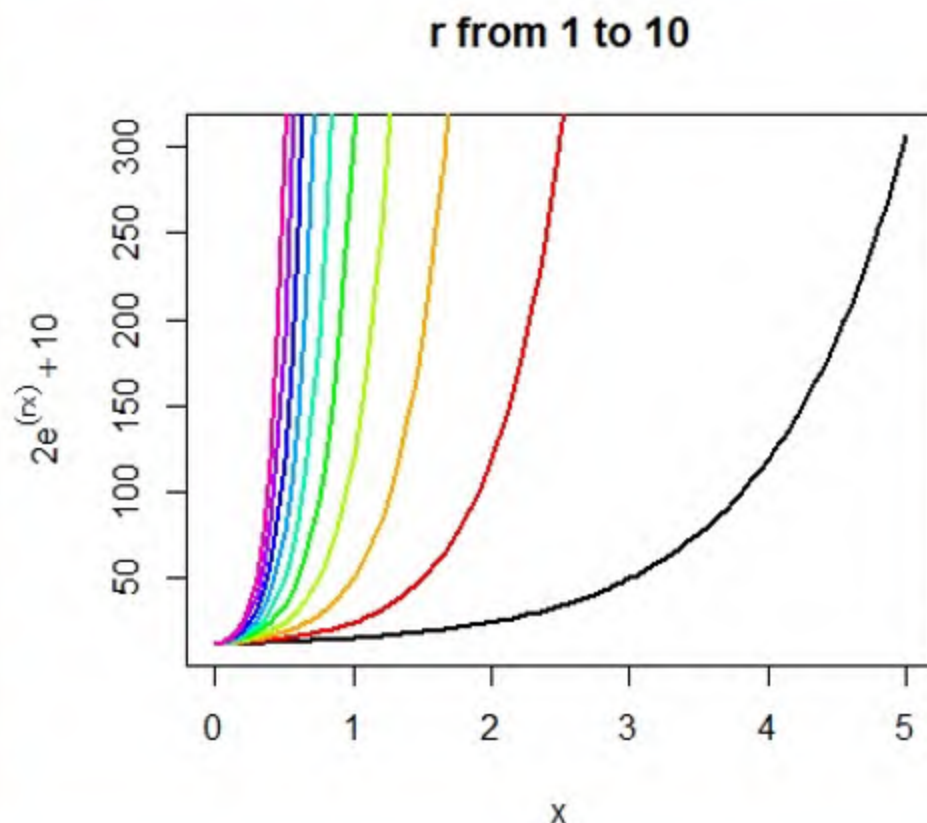
## 013 返回函数的函数



R Console

```
> ## 定义函数、绘制函数图形以及使用循环来绘制多个函数的图形。  
> # 定义一个函数Reflect.origin, 它接受一个函数h作为参数, 并返回一个新的  
> # 这个新函数接受一个参数x, 并返回-h(-x)的结果, 即函数h关于原点的反射。  
> Reflect.origin=function(h){return(function(x){-h(-x)})}  
>  
> # 定义一个函数f, 它接受一个参数x, 并返回x^5 - 3*x^2 - 3*x + 2的结果。  
> f=function(x){x^5-3*x^2-3*x+2}  
>  
> # 判断f(2)的值是否等于f函数关于原点反射后在2处的值, 即-f(-2)的值, 并  
> f(2)==Reflect.origin(f)(2)  
[1] FALSE  
>  
> # 设置图形参数, 其中mar参数设置图形区域的边距。  
> par(mar=c(5.1,5.1,4.1,2.1))  
>  
> # 使用curve函数绘制函数f在区间[-2,2]上的图形, 并设置y轴的范围为[-5,5]  
> # y轴标签为f(x)的表达式, 图形的主标题为"Graph of f(x) and its Reflect  
> curve(f,-2,2,ylim=c(-5,5),ylab=expression(f(x)==x^5-3*x^2-3*x+2),  
+       main="Graph of f(x) and its Reflection (in red) about the Origin")  
>  
> # 使用curve函数在同一张图上绘制函数f关于原点的反射图形, 颜色设置为红  
> curve(Reflect.origin(f)(x),-2,2,col="red",add=TRUE)  
>  
> # 在图形中添加坐标轴的辅助线, v=0和h=0分别表示在x=0和y=0处添加垂直和  
> abline(v=0,h=0,lty=2)
```

R Graphics: Device 2 (ACTIVE)





## 014 计算毕达哥拉斯三元组的函数



```
R Console

> ## 计算毕达哥拉斯三元组的函数
> # 定义一个名为Euclid1的函数，它接受两个参数n和m
> Euclid1=function(n,m){
+   # 初始化一个长度为3的数值型向量triple
+   triple=numeric(3)
+
+   # 使用毕达哥拉斯三元组的公式来计算triple的三个元素
+   triple[1]=m^2-n^2          # triple的第一个元素是m的平方减去n的平方$
+   triple[2]=2*m*n           # triple的第二个元素是2乘以m和n的乘积
+   triple[3]=m^2+n^2         # triple的第三个元素是m的平方加上n的平方$
+
+   # 返回计算好的triple向量
+   return(triple)
+ }
>
> # 调用Euclid1函数，并传入参数2和3，然后打印返回的结果
> Euclid1(2,3)
[1]  5 12 13
>
> # 定义一个名为Euclid2的函数，它也接受两个参数n和m
> Euclid2=function(n,m){
+   # 检查m是否大于n
+   if (m > n){
+     # 如果m大于n，则执行以下操作：
+   }
```





## 015 生成毕达哥拉斯三元组

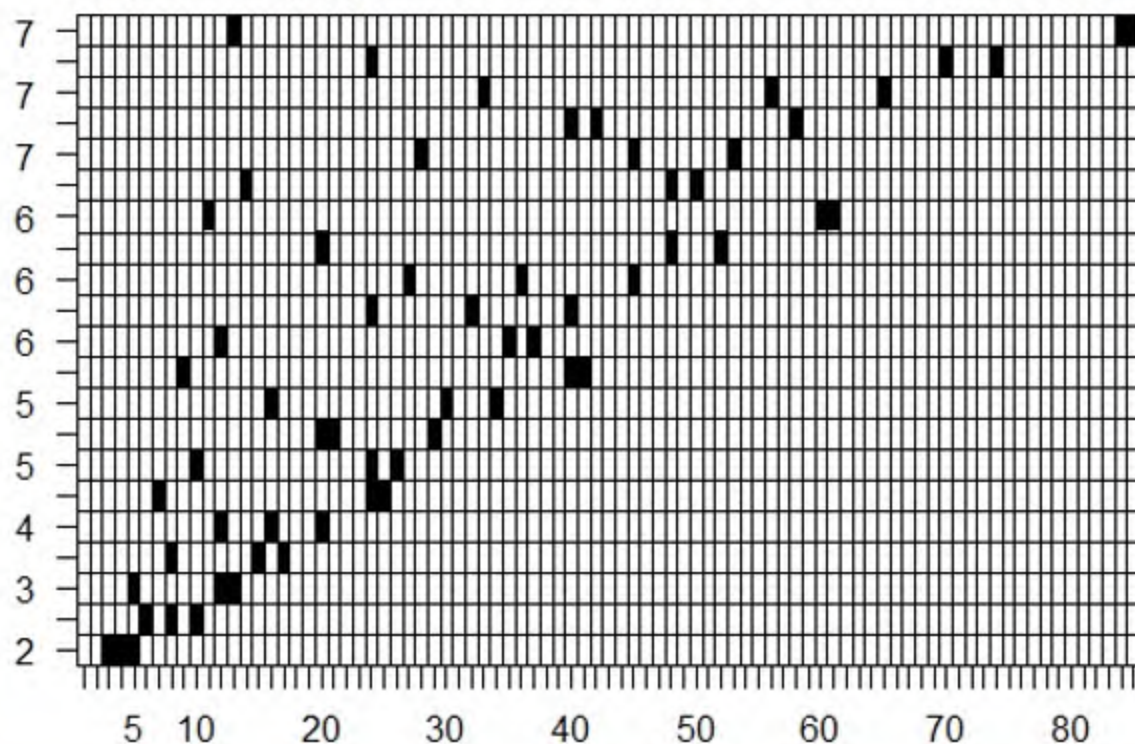


R Console

```
> Two.Columns[1:10,]  
      [,1] [,2]  
[1,]    1    3  
[2,]    1    4  
[3,]    1    5  
[4,]    2    8  
[5,]    2    6  
[6,]    2   10  
[7,]    3    5  
[8,]    3   12  
[9,]    3   13  
[10,]   4   15  
>  
> # 创建一个85列、21行的零矩阵B  
> B=matrix(0,ncol=85,nrow=21)  
>  
> # 根据Two.Columns中的索引，将矩阵B中对应位置的值设为1  
> B[Two.Columns]=1  
>  
> # 输出矩阵B的前2行、前10列  
> B[1:2,1:10]  
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]  
[1,]    0    0    1    1    1    0    0    0    0    0  
[2,]    0    0    0    0    0    1    0    1    0    1  
>
```

R Graphics: Device 2 (ACTIVE)

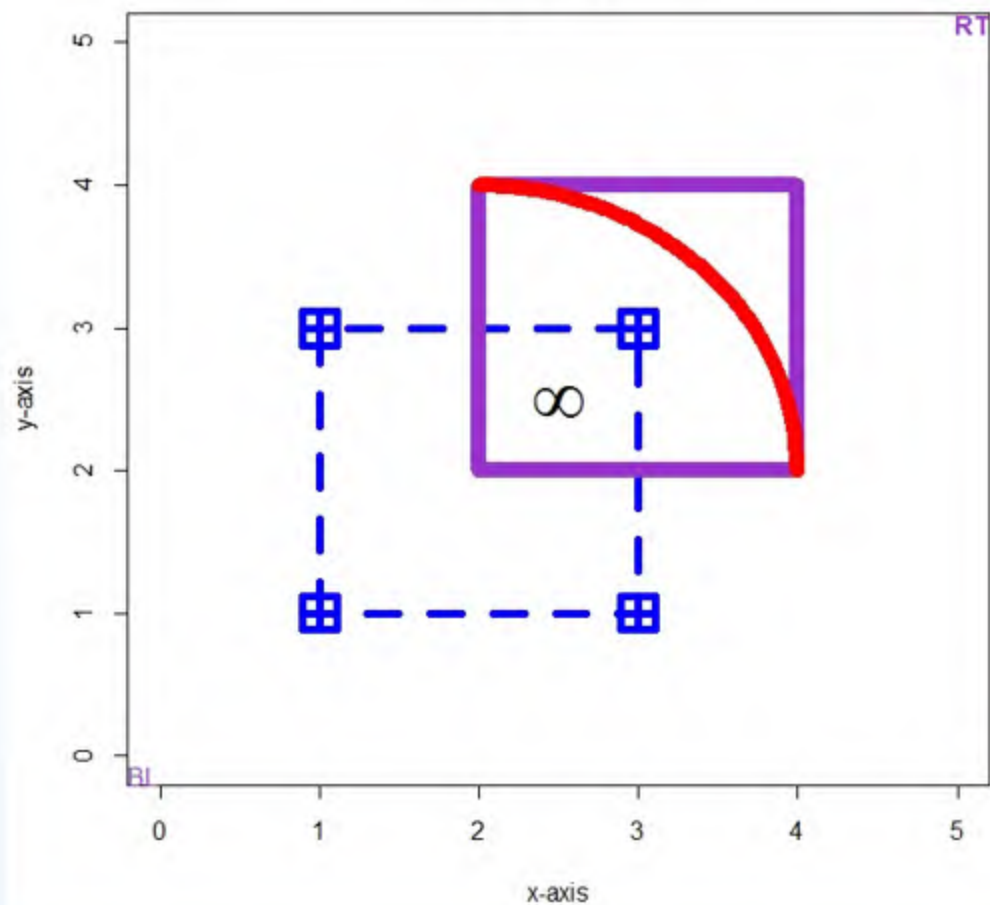
Pythagorean Triples Generated by  $m^2 - n^2$ ,  $2mn$ , and  $m^2 + n^2$ .  
The y-axis displays  $m$  with the values of  $n$  taken from 1 to  $m-1$ .



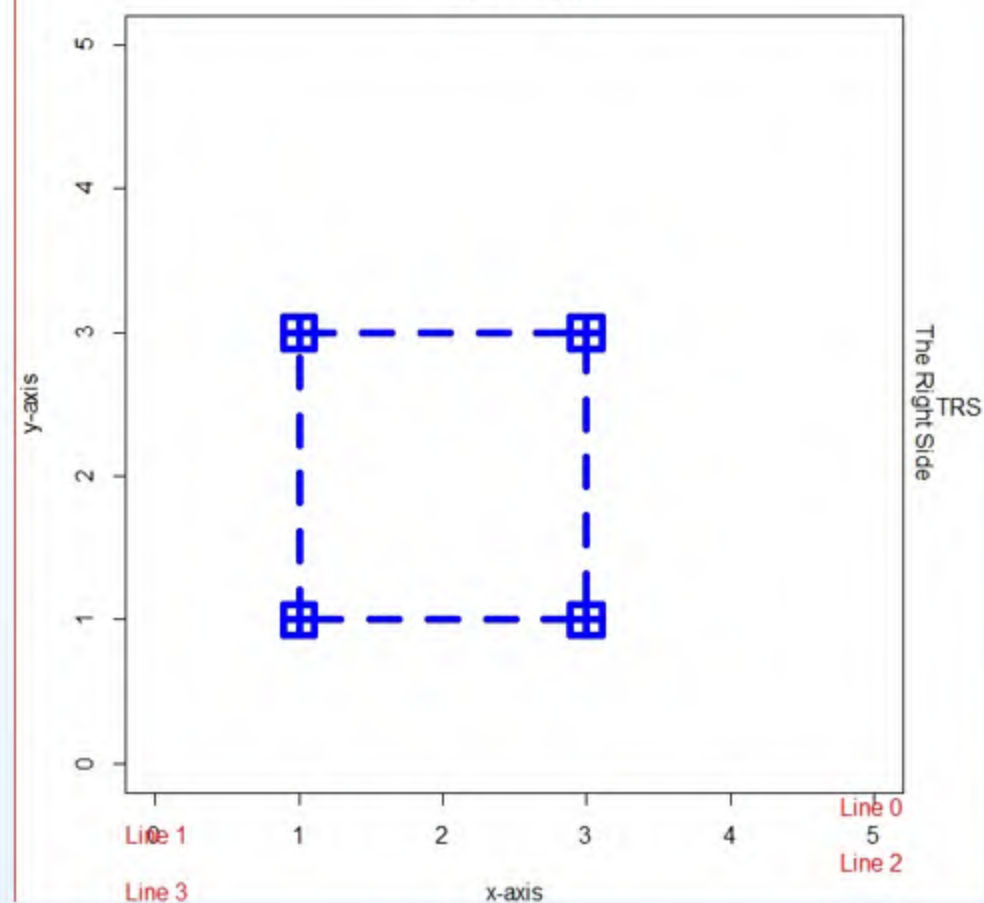
# 016 -017绘图



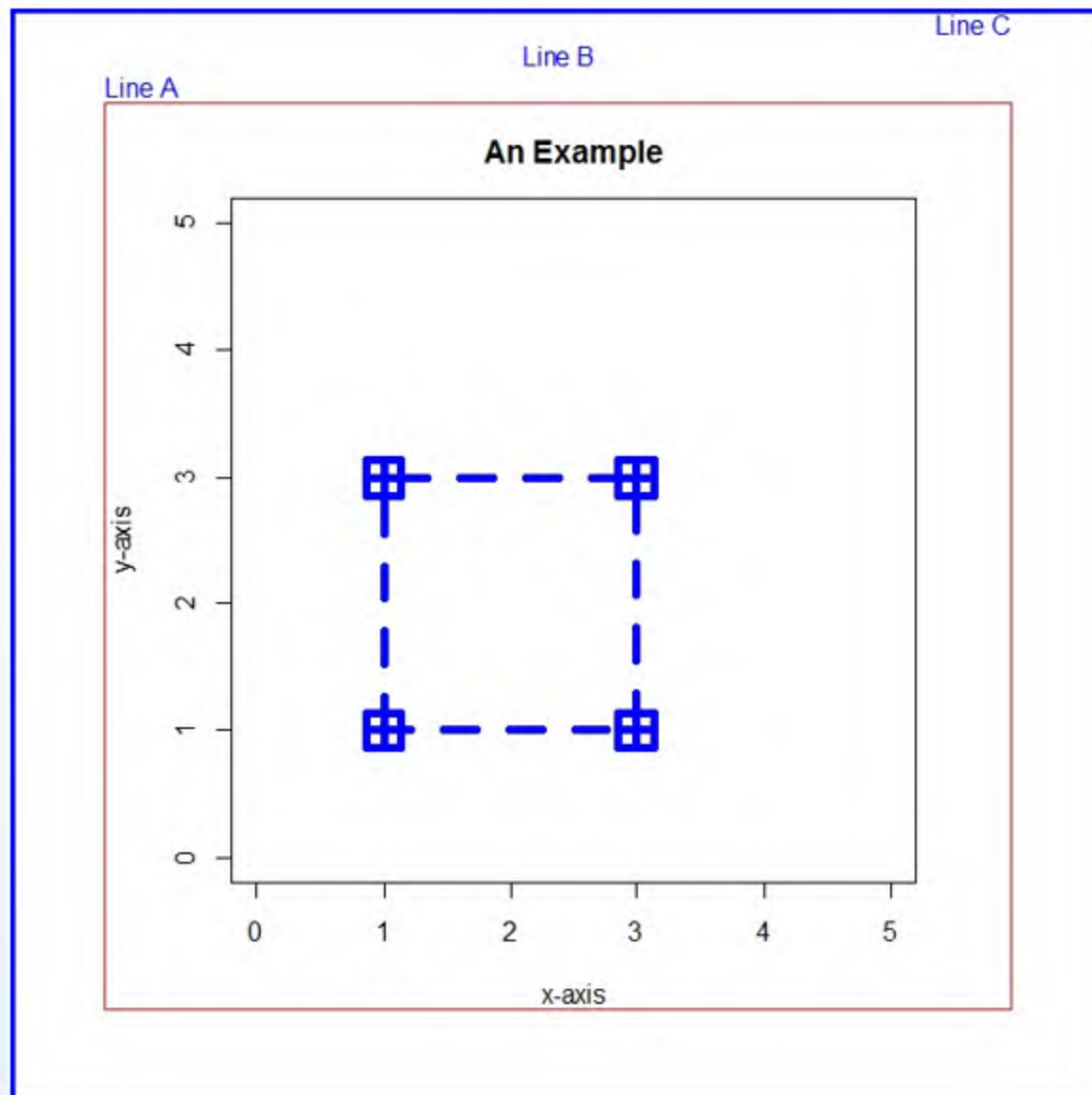
An Example



An Example



# 018 绘图







# 谢谢

[gulp@mail.las.ac.cn](mailto:gulp@mail.las.ac.cn)