第二章 R语言与数据科学

2.1 R语言发展简史



R是一个面向统计计算和统计制图的语言和编程环境。R语言作为一种开源软件，拥有活跃的使用及开发团体。R语言最早诞生于S语言。1976年，S语言是由贝尔实验室首先开发用于统计研究。1992，Ross Ihaka 和Robert Gentleman二人着手基于S语言的句法，创造了R语言，一开始的目的是用于统计入门的教学。由于两人的名字首字母均为R，所以称之为R语言。1994，R语言的雏形完成，两人决定通过网络将R语言公开发布并免费供他人使用。1997，Kurt Hornik和 Fritz Leisch 建立CRAN 档案库，让R语言的使用者可以共享自己开发的包。

R提供了丰富多样的统计和制图工具，并且这些工具具有极好的拓展性。R语言允许用户自定义新函数，并且可以通过包（Package）轻松的扩展功能。

R语言的历史可以追溯到20世纪90年代。R语言最初是由奥克兰大学的罗斯·伊哈卡（Ross Ihaka）和滑铁卢大学的罗斯·伊哈卡（Robert Gentleman）和纽约大学的共同开发的。他们二人的名字都以R开头所以将其发明的语言成为R语言。起初是为了解决自己在统计学研究中遇到的问题而创建了R语言。(Ihaka, R. (2011). The R Project: A brief history and thoughts about the future. Presentation given at the University of Otago.)

R语言最早的版本基于S语言句法实现。R语言从S语言继承了很多特性，但也进行了许多改进和扩展，使其变得更加强大和灵活。

R语言为统计学家和数据分析师提供一个强大的计算和可视化工具。它提供了丰富的统计分析函数和图形绘制函数，使用户可以在一个集成的环境中进行数据的处理和分析。R语言是一个跨平台的编程语言，可以在不同操作系统上运行，如Windows、Mac和Linux。

R语言在统计学界和数据科学领域得到了广泛的应用和认可，并且在金融、医药、社会科学等领域中都有大量使用者。同时，随着人工智能和机器学习的发展，R语言在机器学习和人工智能等领域也有着广泛的应用。

由于R语言的开源特性，R语言的社区也非常活跃，有很多开源的扩展包和工具可以使用，用户还可以贡献自己的包或作为志愿者参与R语言项目的发展。

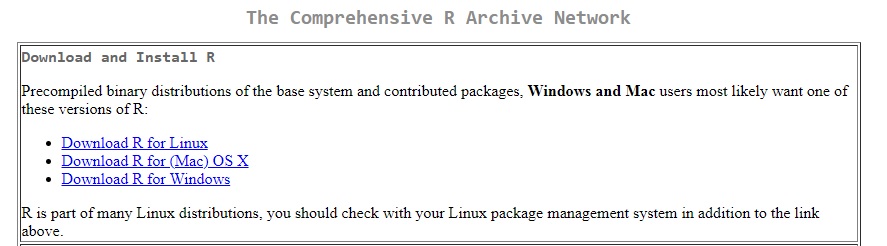
值得一提的是，R语言的学习门槛并不高。它的语法简洁易懂，对于初学者来说也比较友好。因此，R语言在大学和研究机构的教学中得到了广泛应用。它拥有丰富的学习资源和社区支持（例如，R语言在线手册：https://cran.r-project.org/manuals.html； R语言图书：https://www.r-project.org/doc/bib/R-books.html ）。即使是没有编程经验的初学者也能够快速上手。通过学习R语言，人们可以更好地理解和应用数据，从而为决策提供更多的支持和建议。因此，无论你是研究者、数据分析师还是爱好者，学习和掌握R语言都是一个值得投入的宝贵的时间和精力。

R语言可以提供数据科学中数据分析各个环节所需要的各种功能。换句话说，R语言提供了一个数据收集、数据整理、数据计算、数据建模、数据可视化、成果展示的集中的编程环境。R语言是开源的免费软件并且在Windows、Mac和Linux系统上都可以运行。

2.2 R语言/R-studio及软件包的安装

2.2.1 R语言的安装

使用这个链接<https://cloud.r-project.org/>来下载R语言软件。根据你的电脑操作系统，选择相应的版本。



2.2.2 R-Studio安装

R语言可以在计算机上通过终端命令直接运行，也有自己的图形界面。但是RStudio 是最适用于初学者的R 编程的集成开发环境(integrated development environment，IDE)。使用者可以从 http://www.rstudio.com/download 下载并安装。其中免费版已经可以满足数据科学的需求了。

R-Studio支持多种操作系统，包括Windows、Mac和Linux。它具有快速和高效的数据处理能力，可以处理大型数据集并提供交互式可视化。此外，R-Studio还允许用户通过插件和扩展来自定义其功能，以满足不同的需求。R-Studio还提供了一些方便的功能，如代码自动补全和语法高亮等，使得编程变得更加轻松和流畅。通过R-Studio，用户可以快速编写和测试代码，并进行数据可视化和统计分析。

此外，R-Studio还提供了丰富的教程和文档，可以帮助用户快速上手并解决问题。它还有一个活跃的社区，用户可以在这里交流经验和技巧，获取支持和建议。

总之，R-Studio是一款功能强大、易于使用、高度可定制、跨平台的数据科学工具，可以帮助用户轻松地掌握R语言并进行高效的数据处理和分析，为用户提供了全面的数据分析解决方案。无论是进行统计分析、数据可视化还是机器学习，R-Studio都能够帮助用户实现他们的目标。

2.2.3 软件包安装

R package是一种用于R语言的扩展工具，它可以包含数据、代码、文档等，用于实现特定的数据分析、可视化或模型建立任务。R语言允许用户自由地创建、发布和分享自己的package。这种开放性不仅促进了R语言的发展，也为用户提供了更多的选择和灵活性。在R社区中，有很多的R package可以供用户使用，这些package不仅可以提高数据分析的效率，还可以促进不同用户之间的知识交流和共享。

此外，R package还有助于提高代码的可重复性和可维护性，通过将代码和数据封装在一个统一的包中，用户可以更好地管理和维护自己的代码库，使得数据分析的过程更加规范化和标准化。无论是在学术研究、商业应用还是社会活动中，R package都是一个非常有用的工具。

R 语言的开源社区上有各种软件包，一般在The Comprehensive R Archive Network (CRAN)下载安装。其中值得一提的是tidyverse软件包， tidyverse将常用的数据科学软件包整合在一起，是一个重要的数据科学相关包的集成体，提供了数据整理数据计算数据建模数据可视化等一系列的功能包。是R语言入门 学习的首选。

A close up of a map

Description automatically generated

首先，数据导入由readr完成。readr是一个用于读取数据的程序包，它能够快速有效地读取CSV、TSV和其他类型的文本文件。你可以使用read\_csv函数读取CSV文件（类似于使用R基础包中的read.csv函数）。

随后的数据整理（data wrangling）由dplyr和tidyr完成。其中dplyr侧重数据操纵，它提供了一系列函数，用于对数据集进行筛选、排序、修改和汇总等操作。tidyr侧重将数据转换成整洁数据（tidy data）的形式。

所谓“整洁数据”是指每个变量（variable）形成一列，每个观察（observation）形成一行，每个观测单元形成一张表）的形式组织数据。例如，你可以将宽格式的数据转换为长格式，或者将长格式的数据转换为宽格式。

对于字符串数据，stringr包，它提供了一系列简单易用的函数，用于处理字符串。例如，你可以使用str\_detect函数检测字符串中是否含有某个子串，或者用str\_replace函数替换字符串中的内容。

数据可视化由ggplot2包完成。ggplot2是一个基于图形语法（Grammar of Graphics）的设计理念而开发的强大且灵活的可视化工具。你可以使用ggplot2创建各种复杂的图形，如散点图、箱线图、直方图等，以展示数据分布、关系或比较差异等。

此外，purrr用于函数式编程的包，它提供了一系列高级工具，用于处理列表和函数。而tibble用于创建和操作tibble的包，tibble是一种现代化的数据框（data frame），比基础R的数据框更为优化。

由Hadley Wickham和他的团队开发的tidyverse包可以说是R语言数据科学工具中的瑞士军刀。数据处理流程的各个部分均可以由其中不同的工具完成。更重要的是这些工具包的设计和风格都十分一致，能够很好地协同工作。这些包都被设计成能够很好地协同工作，并且它们的函数和语法都十分一致。这使得你可以将它们串联起来，形成一个数据科学工作流，包括数据的导入、整理、变换、可视化和模型拟合等步骤。tidyverse中所有这些包通过管道操作符（%>%）链接起来，使得数据分析的工作流程更加清晰和一致。例如，你可能先使用readr读取数据，然后用dplyr和tidyr进行数据清洗和转换，接着使用ggplot2进行可视化，最后使用purrr进行结果处理。此外，tidyverse生态系统还在不断扩展和改进，包括不断添加新的包和功能，以适应数据科学的新需求和挑战。

软件包的安装可以使用代码，也可以使用R-Studio的界面。

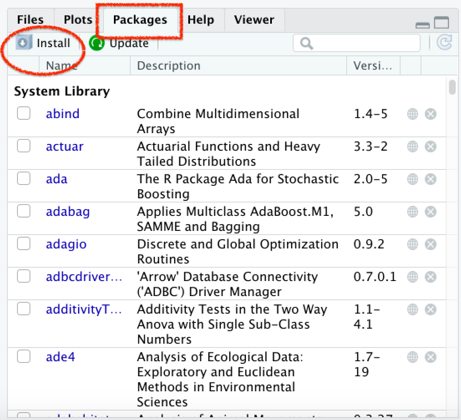
使用命令安装软件包:

install.packages("tidyverse")

在使用软件包的功能时我们需要先

library(tidyverse)

使用R-studio界面中的Package功能也能安装软件包。



2.3 R-studio界面及R脚本和R-markdown文件创建

2.3.1R-Studio界面简介

A screenshot of a social media post

Description automatically generated

首先，初始界面有3个区域。在图中左侧区域是输入和运行代码区域，右侧上部的区域目前选中的是environment也就是环境变量的显示窗口。当前并没有输入任何代码，所以环境变量的窗口是空白。在右侧下部，对窗口有5个不同的按键：文件（File），制图（Plots），程序包（Packages），帮助（Help），视图（Viewer）。目前选中的是作图窗口。

在代码运行窗口，我们可以输入命令，然后按回车键即可运行命令。如果我们想要将命令写成脚本，方便重复使用，修改，那需要新建一个脚本文件。

A screenshot of a cell phone

Description automatically generated

我们推荐两种方式书写脚本，一种是把脚本文件格式，另外一种是Rmarkdown文件格式。

**2.3.2 脚本文件格式**

当我们新建了一个脚本文件，就会出现一个窗口。原先的代码运行窗口，变换到了左边的下部，而在左边上部出现了一个新的窗口。这就是脚本书写窗口。在这里书写的脚本，我们可以通过选中来运行。

A screenshot of a social media post

Description automatically generated

**2.3.3 R-Markdown文件**

A screenshot of a cell phone

Description automatically generated

r-markdown是一种可复制性更好的数据分析记录报告文件格式，符合可读性统计编程（literate statistical programming）的理念。它可以将数据分析的代码，统计建模的细节和解释说明的文字以及可视化的图片统一集成在一个文件当中。这个文件可以输出网页格式、PDF或者文档。

可读性统计编程（literate statistical programming）是基于可读性编程（Literate programming）创造的新的一个概念。可读性统计编程将统计分析过程的代码和相关描述融合在一个文件当中。这样读者既可以了解到分析过程，同时也可以通过作者的数据，运行代码得到相同的结果。这样最大程度的保证研究数据分析的可再生性（reproducibility）。

从学科发展的角度，研究可再生性可以提高研究的科学性。一个研究能否成为人类知识的一个部分，取决于他人是否能够复制验证这个研究的结论。当然，不同学科的可复制性本身有差异。比如，复制一个化学实验比复制一个心理学实验要容易的多，因为化学实验的对象是物质，物质本身的特性是稳定，化学反应的条件是可控的。而心理学实验的对象是人的心理，人本身是存在变异性的，心理活动产生的条件控制难度远远大于化学实验。但是，对于一项心理学研究而言，至少可以实现同行使用此项研究的数据，按照研究者的方法可以得到相同的结论，即可再生性。可复制性最低的门槛就是可再生性。

其次，提高研究的可再生性，公开研究分析过程和代码，可以减少重复研究的浪费。研究者可以利用别人现有的研究过程和代码，在此基础上进一步进行研究。而不需要从头开发。此外，研究的过程不是线性的，一蹴而就的。合作者、期刊编辑、外审专家都会对研究数据处理过程提出一些意见。难免要进行反复修改，模型调试，数据增减，需要不断重复某些数据分析过程。通过编程高效重复数据处理过程，会使研究者将精力放在研究问题上，而不是重复机械劳动上。同时也保证研究结果准确。

对于个人来说，研究者如果有很强的可读性统计编程和可再生性意识，并且准备在研究中实践。这种观念本身就会影响研究数据的管理和组织。使得研究工作更为高效。其次，研究者通常会长期从事某一领域的研究，其研究具有一定的继承性。通过可读性统计编程和脚本管理，研究者可以轻松利用自己过去某个研究的分析过程，实现复利迭代。

对于研究团队而言，可再生研究可以提高团队协作的效率。如果数据分析过程非常透明易懂，那么合作者更加容易融入团队，分工协作。

可再生的研究通常能够让别人更加相信。同行会使用你的数据分析中某些步骤或者代码，并且引用你的论文可以增加你的研究影响力。

首先，点击新建文件，然后选择R-markdown。

A screenshot of a cell phone

Description automatically generated

在弹出的对话框中，你需要选择文件输出的类型，比如这里我们选择网页格式。同时你可以填写相应的文件标题和文件作者名称。完成以后点击OK就可以了。

在R-studio里面会出现一个新的文件即R-markdown文件。这个文件最开头，从第二行到第五行，是一些相关属性设置。比如，文件名称、作者、时间和输出格式。

在图中第八行到第十行之间，有一个代码块区域。在这里书写代码，完成之后，可以选中，进行运行。也可以点击右面绿色小三角，运行整个代码块。

A screenshot of a social media post

Description automatically generated

编辑完成以后，需要将R-markdown文件先命名并且保存。然后，我们可以点击文件上面有一个knit按钮（配有一个织毛衣图示），R-studio就会将文件转换成之前设置的格式。这里我们将文件输出为网页格式。

A screenshot of a social media post

Description automatically generated

建立R-project

在数据处理中，我们通常会将输入和输出的数据放在一个文件夹里，每次导入数据的时候，我们需要输入绝对或者相对的路径。但是当你的合作者在另外一台电脑上打开你的文件夹时，这个绝对或者相对的路径就会发生错乱。因此，为了保持编程环境的相对稳定，我们可以为自己的数据处理项目建立一个project文件夹。当我们有一个project文件夹以后，R-studio里面就只需要输入相对路径。特别值得注意的是，无论是在云端同步，还是通过移动硬盘U盘等复制，这个文件夹是一个独立的编程环境。

首先我们点击文件新建项目。

A screenshot of a cell phone

Description automatically generated

然后我们可以选择新建项目或者使用已有的文件夹。

A screenshot of a cell phone

Description automatically generated

如果选择新建项目，那么需要给项目文件夹一个名称，并设置文件夹的位置。

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

新建项目以后，在所指定的路径，就会发现一个新的文件夹。在该文件夹内部，会有一个R-project文件。以后每次开始项目时，双击这个文件就会自动打开一个项目编程环境，里面所有路径都可以是相对路径。

后接相应的R-markdown

3.3 R语言数据结构

在R语言中，有四种存储数据的结构，分别为向量、列表、矩阵、数据框。

|  |  |  |
| --- | --- | --- |
| 维度 | 同质 | 异质 |
| 1D | 向量 | 列表 |
| 2D | 矩阵 | 数据库 |
| nD | Array |  |

向量

首先，向量是一个一维的数据结构。它可以看作是一个集合。集合元素必须是同一数据类型，比如数值、字符串、逻辑值。如果数据类型不同，数值型数据会被改写成字符串数据；逻辑值数据也会被改写为字符串数据。

创建向量

# “：”表示从数值8到数值17.  
a = 8:17  
  
a1 = 8:17 + 1  
  
# c()是一个函数,表示将括号内元素合并在一起。  
b <- c(9, 10, 100, 38)  
  
#逻辑值数据  
c1 = c (TRUE, FALSE, TRUE, FALSE)  
  
c2 = c (T, F, T, F)  
  
# 双引号表示数据为字符串  
d = c ("TRUE", "FALSE", "FALSE")  
  
# 将数值数据改变为字符串数据  
as.vector(b, mode = "character")

## [1] "9" "10" "100" "38"

# 如果数据类型不同，数值型数据会被改写成字符串数据；逻辑值数据也会被改写为字符串数据。  
  
e = c(9,10, "ab", "cd")  
  
f = c(10, 11, T, F)

列表也是一种一维数据结构，和向量不同的是，它可以包含很多不同元素的集合。 创建列表的方式，就是将不同向量集合在一起。在实际数据分析过程中，会产生数据框内，嵌套列表的情况，并且很多统计模型的输出结果也是列表。

### 

### 创建列表

list1 = list(a, b, c, d, e, f)  
  
list1

## [[1]]  
## [1] 8 9 10 11 12 13 14 15 16 17  
##   
## [[2]]  
## [1] 9 10 100 38  
##   
## [[3]]  
## function (...) .Primitive("c")  
##   
## [[4]]  
## [1] "TRUE" "FALSE" "FALSE"  
##   
## [[5]]  
## [1] "9" "10" "ab" "cd"  
##   
## [[6]]  
## [1] 10 11 1 0

## 二维数据结构

我们大部分人都使用过Excel表格。在Excel表格当中有行和列。这就是一个典型的二维数据结构。 在R中，我们有两个对应的数据结构：矩阵和数据框。

矩阵是一系列数据元素的二维矩形排列。所有元素必须是同样的数据类型。比如，同为数值型或者同为字符串型。

数据框和矩阵的形式非常相近，也是一种二维数据结构。区别在于，数据框中数据，列与列之间，可以是不同的数据类型。因此，在实际的数据科学数据分析当中，我们使用数据框的频率更高。

下面我们尝试模拟一个数据框。这个数据框模拟的内容是一个班的学生，语文、数学、英语、体育、音乐的成绩。

# 如何模拟数据？

# 模拟连续数值  
9:20

## [1] 9 10 11 12 13 14 15 16 17 18 19 20

seq (1, 10)

## [1] 1 2 3 4 5 6 7 8 9 10

#连续数值间隔为2  
seq (1, 20, by= 2)

## [1] 1 3 5 7 9 11 13 15 17 19

#重复数值，重复4次5  
rep (5, 4)

## [1] 5 5 5 5

#重复4次123  
rep (c(1,2,3), 4)

## [1] 1 2 3 1 2 3 1 2 3 1 2 3

#重复4次1，4次2，4次3  
rep (c(1,2,3), each = 4)

## [1] 1 1 1 1 2 2 2 2 3 3 3 3  
  
#产生40个数，均值是70，标准差是5，并符合正态分布。  
rnorm(40, mean = 70, sd = 5)

## [1] 70.23895 65.96219 61.79326 74.80064 63.37901 74.95880 71.51027 72.10169  
## [9] 64.58556 63.99798 67.32969 61.33770 64.76354 70.24638 68.66829 72.61847  
## [17] 67.29309 57.58353 66.39143 65.96490 71.13346 71.15491 76.27109 69.41313  
## [25] 65.90121 64.98780 69.29191 77.32458 76.47522 74.25527 64.20388 72.51400  
## [33] 67.81757 69.28027 67.00122 60.11639 59.22925 70.07005 70.73027 73.95155

# 生成40个学号  
  
id = seq(1, 40)  
  
# 生成性别各20个  
gender = rep(c("male", "female"), 20)  
  
# 生成数学成绩  
maths = rnorm(40, mean = 70, sd = 5)  
  
#生成英语成绩  
english = rnorm(40, mean = 80, sd = 9)  
  
# 生成音乐成绩  
music = rnorm(40, mean = 75, sd = 10)  
  
#生成体育成绩  
pe = rnorm(40, mean = 86, sd = 12)

## 创建数据框

# 创建数据框：学号、性别、数学、英语成绩  
df1 = data.frame (id, gender, maths, english)  
  
#数据框结构  
str(df1)

## 'data.frame': 40 obs. of 4 variables:  
## $ id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ gender : chr "male" "female" "male" "female" ...  
## $ maths : num 69.9 74 70.4 65.4 58.1 ...  
## $ english: num 73.2 62.3 88 59.2 98.3 ...

#数据框摘要  
summary(df1)

## id gender maths english   
## Min. : 1.00 Length:40 Min. :55.90 Min. :59.18   
## 1st Qu.:10.75 Class :character 1st Qu.:66.47 1st Qu.:72.98   
## Median :20.50 Mode :character Median :70.52 Median :79.36   
## Mean :20.50 Mean :69.41 Mean :78.36   
## 3rd Qu.:30.25 3rd Qu.:71.74 3rd Qu.:84.77   
## Max. :40.00 Max. :77.35 Max. :98.30

#数据框行数  
nrow(df1)

## [1] 40

#数据框列数  
ncol(df1)

## [1] 4

#数据框属性  
attributes(df1)

## $names  
## [1] "id" "gender" "maths" "english"  
##   
## $class  
## [1] "data.frame"  
##   
## $row.names  
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

### 如何修改数据框列的名称？

### 修改数据筐聂的名称的第一种方法是在合并向量建立数据框的时候，就对列进行重命名。

### df2 = data.frame (id1 = id, gender1 = gender, maths1 = maths, english1 = english)

### 给数据框增加一列 df2 = cbind(df2, pe)

### 第二种方法是通过一个函数colnames（）来统一设定、更改数据框列名称。

colnames(df2) = c("ID", "SEX","MATHS","ENGLISH","PE")  
  
head(df2)

## ID SEX MATHS ENGLISH PE  
## 1 1 male 69.87996 73.16911 88.78605  
## 2 2 female 74.00595 62.30693 89.18296  
## 3 3 male 70.40945 88.00013 63.05852  
## 4 4 female 65.37252 59.17657 88.22937  
## 5 5 male 58.08811 98.29787 84.73826  
## 6 6 female 75.50516 92.26880 88.69226

### 筛选、读取数据框的某一个部分

公式：数据框名称[行, 列]

# 完整数据框的前六行  
  
head(df2)

## ID SEX MATHS ENGLISH PE  
## 1 1 male 69.87996 73.16911 88.78605  
## 2 2 female 74.00595 62.30693 89.18296  
## 3 3 male 70.40945 88.00013 63.05852  
## 4 4 female 65.37252 59.17657 88.22937  
## 5 5 male 58.08811 98.29787 84.73826  
## 6 6 female 75.50516 92.26880 88.69226

# 提取数据框的2-5行

df2[2:5, ]

## ID SEX MATHS ENGLISH PE  
## 2 2 female 74.00595 62.30693 89.18296  
## 3 3 male 70.40945 88.00013 63.05852  
## 4 4 female 65.37252 59.17657 88.22937  
## 5 5 male 58.08811 98.29787 84.73826

# 选择数据狂的1-3行以及1-2列

df2[1:3 , 1:2]

## ID SEX  
## 1 1 male  
## 2 2 female  
## 3 3 male

# 根据列的名称筛选数据

df2[1:3 , c("ENGLISH", "PE")]

## ENGLISH PE  
## 1 73.16911 88.78605  
## 2 62.30693 89.18296  
## 3 88.00013 63.05852

#前3列

df2[c(1,2,3), ]

## ID SEX MATHS ENGLISH PE  
## 1 1 male 69.87996 73.16911 88.78605  
## 2 2 female 74.00595 62.30693 89.18296  
## 3 3 male 70.40945 88.00013 63.05852

#从1-40行每隔四行选一次

df2[seq(1, 40, 4), ]

## ID SEX MATHS ENGLISH PE  
## 1 1 male 69.87996 73.16911 88.78605  
## 5 5 male 58.08811 98.29787 84.73826  
## 9 9 male 66.72537 92.15171 97.10975  
## 13 13 male 71.72234 73.48177 102.52399  
## 17 17 male 70.41510 70.09197 79.54164  
## 21 21 male 71.59562 77.15748 69.08213  
## 25 25 male 70.86781 83.97190 84.31275  
## 29 29 male 70.63032 79.43726 75.88278  
## 33 33 male 73.24947 86.65819 89.77531  
## 37 37 male 72.16663 96.16427 102.37157

# 数据框的简单计算

下面是一些R自带的函数，可以用来计算：最大值、最小值、取值范围、求和、求平均值、中位数、4分位点、标准差、求近似值。

# Now that you have a vector, you can do some Maths.  
  
max(a)

## [1] 17

min(a)

## [1] 8

range(a)

## [1] 8 17

sum(a)

## [1] 125

mean(a)

## [1] 12.5

median(a)

## [1] 12.5

quantile(a)

## 0% 25% 50% 75% 100%   
## 8.00 10.25 12.50 14.75 17.00

sd(a)

## [1] 3.02765

round(sd(a), 2)

## [1] 3.03

# 我们也可以对于一个数据框的某一列，也就是某一个变量，进行一些简单的统计描述性统计值计算。比如这里，我们对第一个数据框当中的数学成绩，进行相应的描述性统计计算。

#  
max(df1$maths)

## [1] 77.35393

min(df1$maths)

## [1] 55.90001

range(df1$maths)

## [1] 55.90001 77.35393

sum(df1$maths)

## [1] 2776.541

mean(df1$maths)

## [1] 69.41352

median(df1$maths)

## [1] 70.52271

quantile(df1$maths)

## 0% 25% 50% 75% 100%   
## 55.90001 66.47005 70.52271 71.73687 77.35393

sd(df1$maths)

## [1] 4.66338

round(sd(df1$maths), 2)

## [1] 4.66

R语言资源

The R Manuals：<https://cran.r-project.org/manuals.html>

Books：<https://www.r-project.org/doc/bib/R-books.html>