

R 语言编程：基于 tidyverse

第 02 讲 搭建 R 环境与基本使用

张敬信

2022 年 12 月 7 日

哈尔滨商业大学

一. 搭建 R 环境

1. 安装 R

- 到 R 国内镜像站，我最近常用的是北京外国语大学的：

<https://mirrors.bfsu.edu.cn/CRAN/>

根据自己的操作系统，下载相应的最新版 R-4.2.2 安装即可。

切记：若 Windows 系统用户名为中文，先改成英文！

- 建议安装在 D 盘，不要有中文路径，且路径不要有空格。
- Windows 系统安装时可根据系统只选择 32 位或 64 位版本
- 建议取消勾选 “Message translations”（英文报错更容易搜索解决办法）。

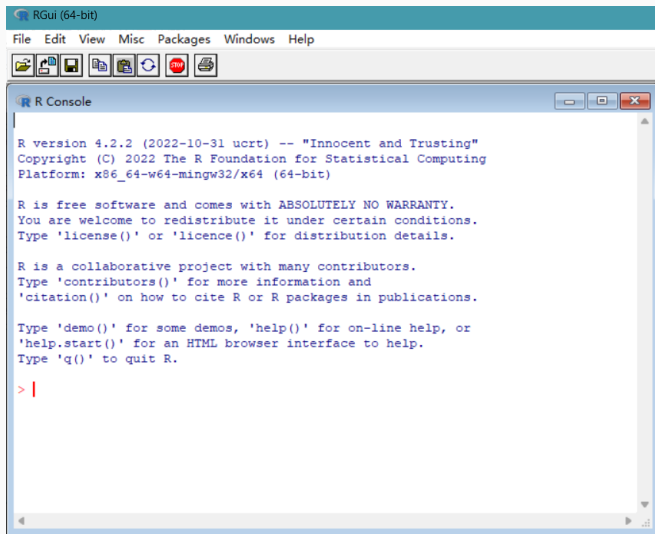


图 1: R 4.2.2 软件界面

2. 安装 RStudio

- 不要直接使用 R, 而是使用更好用的 R 语言集成开发环境 Rstudio, 官网下载地址:

<https://www.rstudio.com/products/rstudio/download>

下载安装 (或直接下载 zip 版解压), 将自动关联已安装的 R。

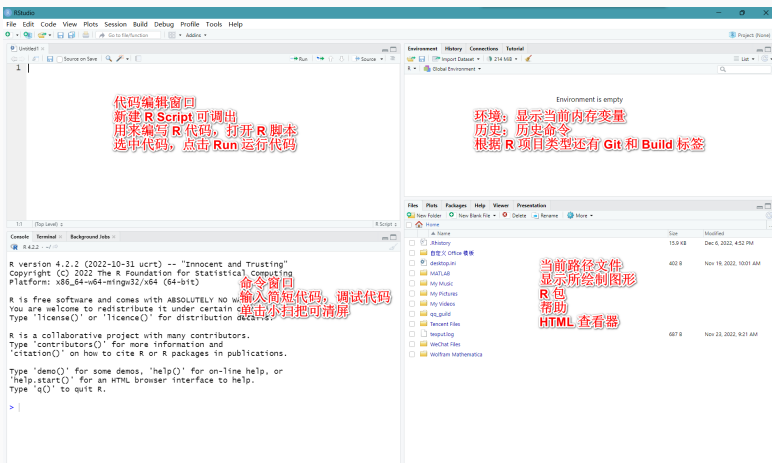


图 2: RStudio 操作界面

- 切换安装包的国内镜像源 (任选其一), 以保证正常安装 R 包:

Tools -> Global Options... -> Options -> Packages, 点击 Change 修改为, 比如 Beijing Foreign Studies University

- 设置保存文件的默认编码方式为 UTF-8, 已保证保存出来的各种 R 文件, 都将采用 UTF-8 编码:

Tools -> Global Options... -> code -> Saving, 在 Default text encoding 框, 点 change, 修改为 UTF-8

- 勾选 Code -> Display 下的 Rainbow parentheses 框, 这样代码中的配对括号将用不同彩虹色匹配;
- 设置 General -> Workspace 取消勾选 Restore .RData into workspace at startup, 并将其下的 save workspace to .RData on exit: 改为 Never. 避免每次打开 RStudio 都加载以前数据。

二. R 包

1. 安装包

(1) 来自官方 CRAN 的 R 包, 直接使用命令安装, 将自动安装依赖包

```
install.packages("openxlsx")    # 包名加引号
```

- 若某依赖包已被 CRAN 移除, 则会安装失败, 需要手动从网上搜索到该包, 下载.zip 或.tar.gz 文件到本地, 再手动安装:

Tools -> Install Packages, 修改 Install from, 然后浏览找到文件安装

- 最常见的安装包的失败有两种情形:
 - 无法自动下载到某个依赖包 (手动搜索下载本地安装)
 - R 环境搭建在 C 盘, 没有写入临时文件的权限

(2) Github 也是 R 包的较大的来源, 自动从 Github 安装¹

```
devtools::install_github("tidyverse/dplyr") # 或者  
remotes::install_github("tidyverse/dplyr")
```

注: tidyverse 为 Github 用户名, dplyr 为该用户的名为 dplyr 的 repository (仓库), 也是包名。注意, 不是所有的仓库都是 R 包 (含有 DESCRIPTION 文件是 R 包的标志)。

¹:: 前面是包名, 这是不单独加载包, 而使用包中函数的写法.

注意，国内往往不能正常访问 Github，所以自动安装通常是以失败告终！

- 建议方法：先从 Github 将整个包文件夹下载下来，解压缩到当前路径（或提供完整路径），再从本地手动安装：

```
install.packages("dplyr-master", repos=NULL, type="source")
```

(3) 生信领域在 R 中自成一派，有专门的包的大本营：

<https://www.bioconductor.org>

先安装 BiocManager 包，再用 `install()` 函数安装 bioconductor 来源的包：

```
BiocManager::install("openxlsx")
```

注：R 包默认都安装在 `.../R-4.x.x/library` 路径下。对于不能联网电脑，可以整体替换 `library` 文件夹的方式，代替安装所有已安装包。

其它包操作

- 要使用 R 包中的函数，首先要加载包：

```
library(openxlsx)           # 加载包
```

- 更新包

```
update.packages("openxlsx")  
update.packages()           # 更新所有包
```

- 删除包

```
remove.packages("openxlsx")
```

三. 绝对路径与相对路径

- **绝对路径**: 从盘符开始的完整路径, 比如 `E:/R 语言/data/a123.csv`
- **相对路径**: 是指相对于当前路径的路径, 因为通常操作的文件都是在当前路径下, 那么“从盘符到当前路径”这部分是大家所共有的, 所以可以省略不写, 只写从当前路径再往下的路径即可, 这就是相对路径。比如, 当前文件夹 `E:/R 语言` 中有 `data` 文件夹, 里面有数据文件 `a123.csv`, 要写能访问到它的路径, 只需写 `data/a123.csv`

- 获取或设置当前路径

```
getwd()
```

```
setwd("D:/R-4.1.1/tests")
```

特别注意：R 路径中的 \ 必须用 / 或 \\ 代替。

四. R Project(项目)

- R 脚本是单个可执行的 R 代码文件，后缀名为.R，单击 New File 按钮，选择 R Script，或使用快捷键 Ctrl + Shift + N，则新建 R 脚本；
- R 脚本中都是可执行的 R 代码 + 注释，选中部分代码，点击 Run 运行选中的代码。
- R 项目是完成某个项目或任务的一系列文件的合集（文件夹），包括数据文件、若干 R 脚本及其他附件，其中包含一个 *.Rproj 文件；
- 强烈建议使用 R 项目 + 相对路径，这样能方便系统地管理服务于共同目的的一系列的文件，可以方便移动位置甚至是移到其他电脑，而不需要做任何有关路径的代码修改就能成功运行。

- 单击 **Create a Project** 按钮创建 R 项目

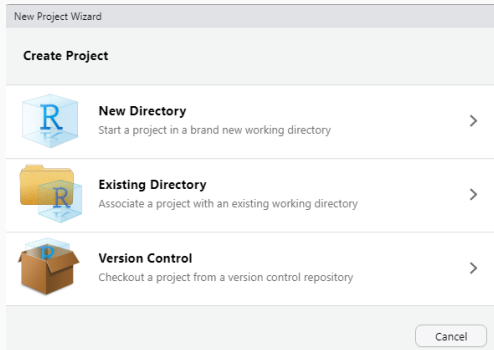


图 3: 创建 R Project

- 创建完成后，在文件夹下出现一个 `*.Rproj` 文件，双击它（关联 RStudio 打开），则进入该 R 项目，做各种具体访问、编辑文件和运行脚本等操作。

五. 一带而过的 R 语法

1. 赋值

- R 标准语法中赋值不是用 `=`, 而是 `<-`或`>`

```
x <- 1:10
```

```
x + 2
```

```
#> [1] 3 4 5 6 7 8 9 10 11 12
```

注: R 也允许用 `=` 赋值, 建议用更现代和简洁的 `=` 赋值。

2. 基本运算

- 数学运算:
 - `+` `-` `*` `/`、`^` (求幂)、`%%` (按模求余²)、`/%` (整除)
- 比较运算
 - `>`、`<`、`>=`、`<=`、`==`、`!=`
 - `identical(x,y)` —— 判断两个对象是否严格相等;
 - `all.equal(x,y)` 或 `dplyr::near(x,y)` ——

²可以关于小数按模求余, 例如 `5.4 %% 2.3` 为 `0.8`

- 判断两个浮点数是否近似相等 (误差 $1.5e-8$)

```
0L == 0
```

```
#> [1] TRUE
```

```
identical(0L, 0)
```

```
#> [1] FALSE
```

```
sqrt(2)^2 == 2
```

```
#> [1] FALSE
```

```
identical(sqrt(2)^2, 2)
```

```
#> [1] FALSE
```

```
all.equal(sqrt(2)^2, 2)
```

```
#> [1] TRUE
```

```
dplyr::near(sqrt(2)^2, 2)
```

```
#> [1] TRUE
```

- 逻辑运算:

- | (或) , & (与) , ! (非) , xor() (异或)

注: && 和 || 是短路运算, 即遇到 TRUE (FALSE) 则返回 TRUE (FALSE) 而不继续往下计算; 而 & 和 | 是向量运算符, 对向量中所有元素分别进行运算。

3. 基本数据类型

- R 中的基本数据类型包括：
 - `numeric`——数值型，又分为 `integer` (整数型) 和 `double` (浮点型)
 - `logical`——逻辑型，只有 `TRUE` 和 `FALSE`，或 `T` 和 `F`
 - `character`——字符型，引号³括起来的若干字符
- R 中用 `NA` 表示缺失值，`NULL` 表示空值，`NaN` 表示非数，`Inf` 表示无穷大
- 对于 R 中大多数函数，`NA` 具有传染性，即 `NA` 参与的运算，结果会变成 `NA`；R 中很多自带函数都提供了 `na.rm` 参数设置在计算时是否忽略缺失值。
- **特别注意：**判断 `x` 是否为 `NA`，不是用 `x == NA`，而是用 `is.na(x)`

³R 中单双引号通用。

- R 中注释一行代码用 #
- 可用函数 `class(x)` / `typeof(x)` / `mode(x)` 查看对象 `x` 的类型
 - 在展现数据的细节上, `mode()` < `class()` < `typeof()`
 - `str(x)` 显示对象 `x` 的结构

4. 保存和载入数据

```
save(x, file = "data/dat.Rda")
```

```
load("data/dat.Rda")
```

5. 清屏和清除内存变量

- Ctrl + L 或单击命令窗口右上角的**小刷子**可对命令窗口清屏。
- 清除当前变量：

```
rm(x)                                # 清除变量 x  
rm(list = ls(all = TRUE))           # 清除所有当前变量
```

注：单击 Environment 窗口的小**刷子**也是清除所有当前变量。

6. 获取帮助

编程语言最好的学习资料就是帮助！

- 函数帮助，命令窗口执行：

```
?plot
```

则在 help 窗口打开 `plot()` 函数的帮助：包括函数来自哪个包、函数的描述、参数说明、更多解释、实例等。

- 在线帮助（需联网）

若想根据某算法的名字或关键词，搜索哪个包能实现该算法：

```
RSiteSearch("network")
```


- R 官方镜像站（北外）下的各种资源，建议自己去发掘

<https://mirrors.bfsu.edu.cn/CRAN/>

比如，最常用的是包的帮助文档：在镜像站，点左侧的 Packages，再点 sort by name，则出现所有可用的 CRAN 包列表

- 点击某个包名，则进入该包的介绍页：

Reference manual: [tidyverse.pdf](#)
Vignettes: [The tidy tools manifesto](#)
[Welcome to the tidyverse](#)
Package source: [tidyverse_1.3.0.tar.gz](#)
Windows binaries: r-devel: [tidyverse_1.3.0.zip](#), r-devel-gcc8: [tidyverse_1.3.0.zip](#),
OS X binaries: r-release: [tidyverse_1.3.0.tgz](#), r-oldrel: [tidyverse_1.3.0.tgz](#)
Old sources: [tidyverse archive](#)

Reverse dependencies:

Reverse depends: [CVE](#), [GADMTTools](#), [neuropsychology](#), [optimos.prime](#), [Tushare](#)

图 4: tidyverse 包的官网介绍页

Reference manual 为参考手册，包含该包所有函数和自带数据集的说明，供查阅使用；Vignettes（若有），是包的作者写的使用文档，它是该包的最佳学习资料。

- 建议将英文报错信息⁴，用 bing 国际版搜索更容易找到答案；
- Github 是丰富的程序代码仓库，在 bing 搜索时，加上 github 关键词，可能有意想不到的收获；
- 其他开放的 R 社区：
 - Stack overflow: <https://stackoverflow.com/questions/tagged/r>
 - R-Bloggers: <https://www.r-bloggers.com>
 - Tidyverse: <https://www.tidyverse.org>
 - Rstudio: <https://rstudio.com>
 - 统计之都: <https://d.cosx.org>
 - 最新免费 R 书: <https://www.bookdown.org>

⁴设置 `Sys.setenv(LANGUAGE = "en")`。

六. R markdown

- 后缀名为.Rmd 的交互式文档, 是 Markdown 语法与 R 脚本的结合, 可以将可执行 R 代码和不可执行的文字叙述, 融为一个文件;
- 单击 New File 按钮, 选择 R Markdown 创建 R markdown, 建议优先使用自带和来自网络的现成模板;
- R markdown 适合编写包含 R 语言代码的学习笔记、演示文档、论文、书籍等, 可以生成 docx, pptx, html, pdf 等多种文档格式。

- RStudio 推出的支持多种语言、下一代 R Markdown 版本, 包括几十个新的特性和功能, 同时能够渲染大多数现有的 Rmd 文件而不需要修改。
- Quarto 是一个建立在 Pandoc 上的开源科技出版系统。
- 用 Python、R、Julia 和 Observable 创建动态内容。
- 以纯文本 markdown 或 Jupyter 笔记本的形式编写文档。
- 以 HTML、PDF、MS Word、ePub 等格式发布高质量的文章、报告、简报、网站、博客和书籍。
- 用科学的 markdown 进行创作, 包括公式、文献引用、交叉引用、图形面板、插图编号、高级布局等。

基本使用:

- (1) 到 (<https://quarto.org/>) 官网下载 Quarto CLI, 并安装
- (2) 选择一种开发环境: VS code/RStudio/Jupyter Lab/Text Editor
- (3) RStudio 用法与 Rmd 基本相同, Yaml 设置更简洁。新建 Quarto Document/Presentation 进入创建向导, 文件后缀名为 .qmd
- (4) 官网的帮助文档分门别类非常详细, 是最佳的学习资料。

本篇主要参阅 (张敬信, 2022), 模板感谢 (黄湘云, 2021), (谢益辉, 2021)。

参考文献

张敬信 (2022). *R 语言编程：基于 tidyverse*. 人民邮电出版社, 北京.

谢益辉 (2021). *rmarkdown: Dynamic Documents for R*.

黄湘云 (2021). *Github: R-Markdown-Template*.