第 一 章　基础语法

1.1 搭建 R 环境及常用操作

1.1.1 搭建 R 环境

R 语言原生官网速度慢，建议直接到到 R 镜像站，目前国内有 9 个镜像站，我常用的两个是清华大学和同济大学的：

https://mirrors.tuna.tsinghua.edu.cn/CRAN

https://mirrors.tongji.edu.cn/CRAN

根据自己的操作系统，下载相应的最新版 R-4.0.2 安装即可，由于免费软件都是简单的下一步，不再赘述。Windows 系统安装时可根据系统只选择 32 位或 64 位版本。

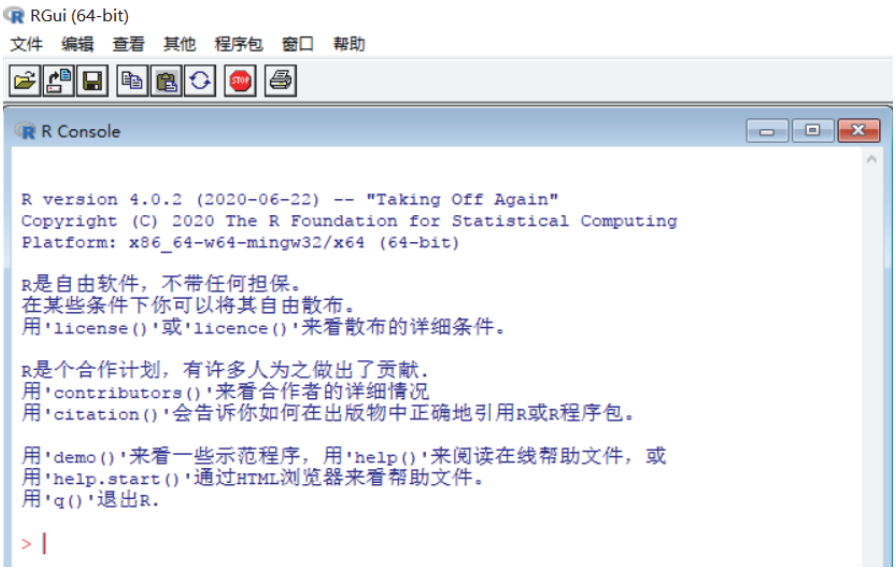


图 1.1: R 4.0.2 运行界面

建议安装在 D 盘，不要有中文路径，且路径不要有空格。

切记：若 Windows 系统用户名为中文，先改成英文！

注意，最好保证电脑里有且只有一个版本的 R，否则 RStudio 自动关联 R 的时候会出现一些麻烦。

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。

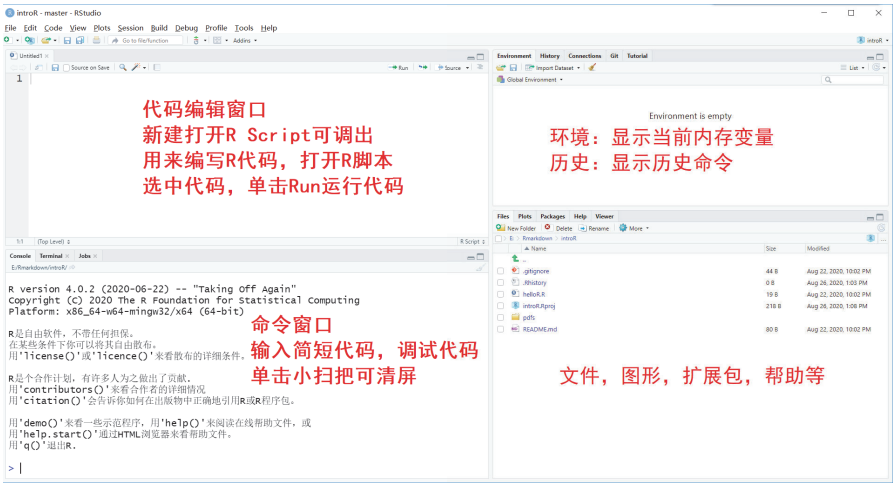


图 1.2: R Studio 操作界面

一些必要的设置

- 切换安装扩展包的国内镜像源（任选其一）

【Tools】—--【Global Options. . .】，在 Options 窗口点 Packages，点 Change 修改为，比如同济大学镜像源：

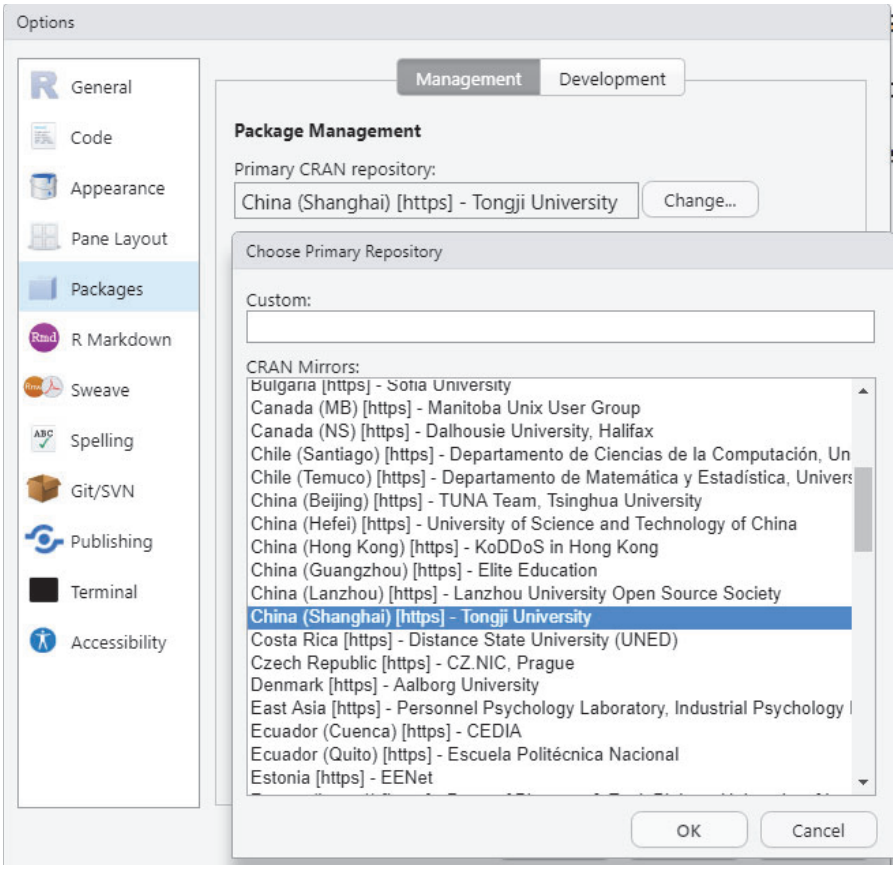


图 1.3: R Studio 设置国内镜像源

- 设置中文编码方式（避免 R 脚本 Rmarkdown 等文件乱码）

【Tools】——-【Global Options. . . 】，点【code】——-【Saving】，在 Default text encoding 框，点【change】，修改为 UTF-8，【OK】

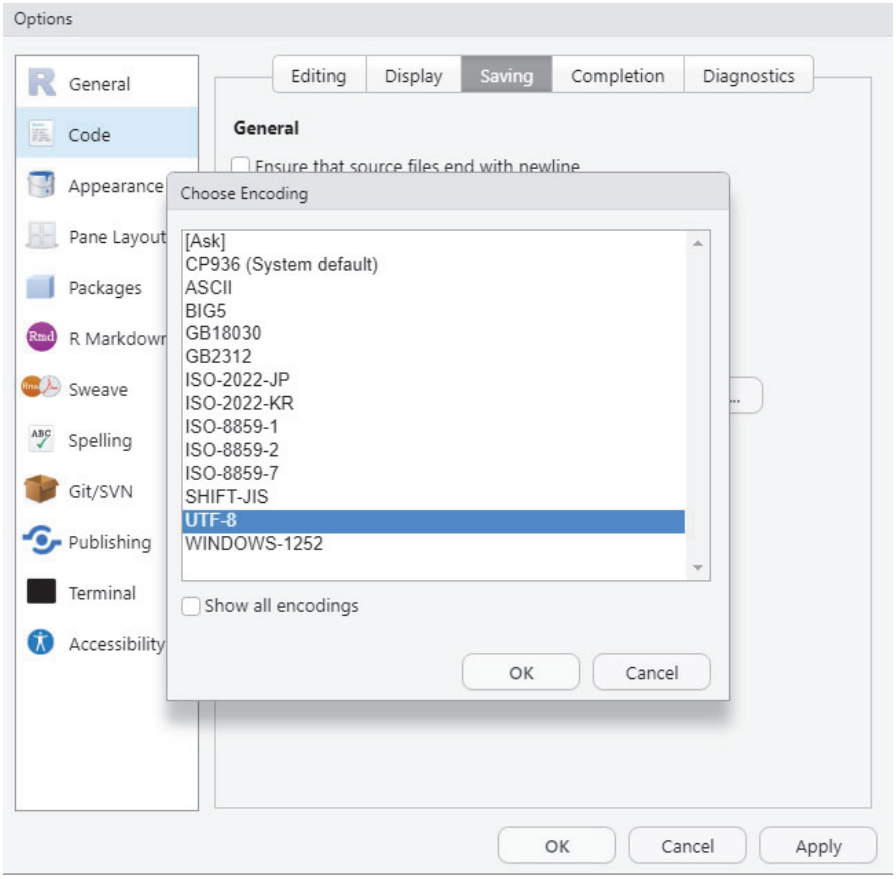


图 1.4: R Studio 设置 code 编码

1.1.2 常用操作

安装包

扩展包（package），简称包。通常 R 包都来自 CRAN，审核比较规范严格，包的质量相对更有保障。建议使用命令安装：

```
install.packages("openxlsx")
```

> openxlsx 为包名，必须要加引号（R 中单双引号通用）。

有些包不能自动安装，可以手动从网上搜索到下载.zip 或.tar.gz 文件到本地，再手动安装（不建议）：

【Tools】——-【Install Packages】，修改 Install from, 然后浏览安装

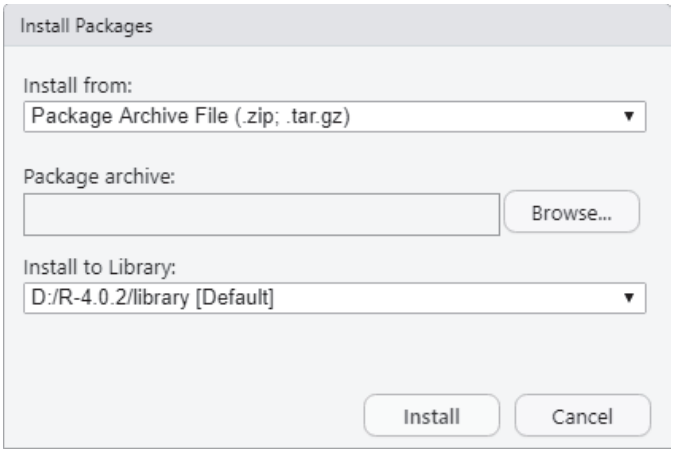


图 1.5: 手动安装扩展包

手动安装包，经常容易安装失败，通常是因为没有先安装该包的依赖包，故需要去包的网页查询其依赖包，确定若未安装，需要先安装它们。这往往又涉及到依赖包的依赖包，所以最好不要手动安装包。另外，尽量用最新版本的 R 会减少很多安装包失败。

Github 也是 R 扩展包的较大的来源，有些作者自己开发的 R 包只放在 Github，也有很多 CRAN R 包的最新开发版都位于 Github。可以先安装 devtools 或 remotes 包，再用其 `install_github()` 安装 Github 来源的包：

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

`::` 前面是包名，这是不单独加载包，而使用包中函数的写法。

`hadlley` 为 Gighub 用户名，`dplyr` 为该用户的名为 dplyr 的 repository（仓库），也是包名。

若网络等原因，导致直接从 Github 安装包失败，也可以将整个包文件夹从网页下载下来，解压缩到当前路径（或提供完整路径），再从本地安装它：

```
install.packages(" 解压文件夹名", repos=NULL, type="source")
```

另外，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")
```

实用场景

R 包默认都安装在 .../R-4.0.x/library 路径下。

你在自己电脑上搭建好 R 语言环境，并安装好了很多常用包，然后你想到一台没有 R 环境、没有联网的电脑上复现你的代码。

方法非常简单：你只需要在那台电脑安装相同版本的 R 软件，安装到相同路径下，将新的 library 文件夹完全替换为你电脑里的 library 文件夹即可[a]，这样运行起 R 代码与在你电脑没有任何区别。

[a]可以用添加压缩包再解压的方式，速度能快一些。

加载包

```
library(openxlsx)
```

更新包

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

删除包

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

获取或设置当前路径

```
getwd()
```

```
## [1] "E:/Rmarkdown/R_Programing_book"
```

```
setwd("D:/R-4.0.2/tests")
getwd()
```

```
## [1] "D:/R-4.0.2/tests"
```

特别注意：路径中的 \ 必须用 / 或 \\ 代替。

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
```

> R 也允许用 = 赋值，建议用更现代和简洁的 = 赋值。

R 是一种基于对象的向量化编程语言，即在定义类的基础上，创建与操作对象；数值向量、函数、图形等都是对象。

基本运算

- 数学运算：
  - `+ - * /`、`^`（求幂）、`%%`（求模[1]）、`%/%`（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)` —— 判断两个浮点数是否近似相等（误差 1.5e8）

```
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）

[1]可以关于小数求模，例如 `5.4 %% 2.3` 为 0.8

`&&` 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。

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[2] 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

保存和载入数据

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

关于相对路径与绝对路径

编程中的文件路径，可以用绝对路径也可以用相对路径。

绝对路径，是从盘符开始的完整路径：比如 `E:/R 语言/datas/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，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。比如，当前文件夹 `E:/R 语言`中有 `datas` 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 `datas/a123.csv`。

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。

若要清除当前变量，用：

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

[2] R 中单双引号通用。

注：单击 Environment 窗口的小刷子也是清除所有当前变量。

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")
```

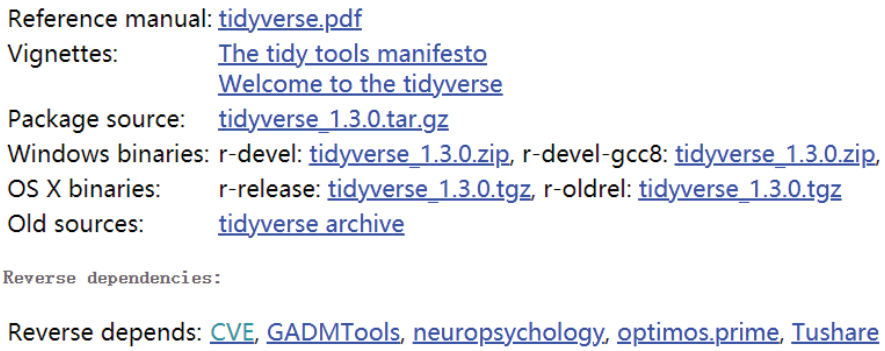
注：很奇怪，现在只能查一个单词，在打开的网页，可以输入多个单词查询

- 其它主要网络资源
  R 官方镜像站

https://mirrors.tongji.edu.cn/CRAN

下的各种资源，建议自己去发掘。

比如，最常用的是包的帮助文档：在镜像站，点左侧的 Packagess，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, GADMTools, neuropsychology, optimos.prime, Tushare

其中，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。

在使用 R 语言过程中遇到各种问题，都建议优先用 Google 搜索相应关键词，更容易找到答案。另外，Github 是丰富的程序代码仓库，在 Google 搜索时，加上 github 关键词，可能有意想不到的收获。

其它开放的 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

R Script 与 R Project

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 工程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 脚本及其它附件，其中包含一个 `*.Rproj` 文件；

强烈建议使用 R 工程，它能方便系统地管理服务于共同目的一系列的文件，可以方便移动位置甚至是移到其它电脑，而不需要做任何路径设置就能成功运行。

创建 R 工程：单击 Create a Project 按钮

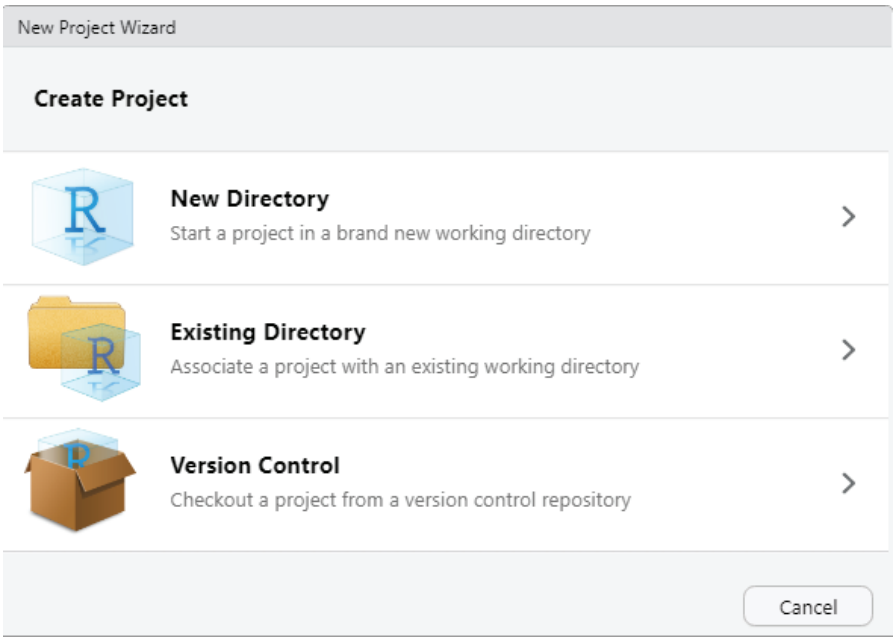


图 1.6: 创建 R Project

若在某个已存在的文件夹下创建工程，则选择 Existing Directory；若需要新建文件

夹创建工程，则选择 New Directory。

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 工程，做各种具体访问、编辑文件和运行脚本等操作。

Rmarkdown

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 Rmarkdown，建议优先使用自带和来自网络的现成模板。

Rmarkdown 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。更多 Rmarkdown 内容将在第五章展开讨论。