R语言在CGED-Q JSL中的运用

陈俊
香港科技大学社会科学部
2025年5月(V2.1.1)

One & Two

初识Rstudio和JSL数据库

01

Why use CGED-Q JSL ?

372
版本数

729,410
武官总记录数

3,899,606
文官总记录数

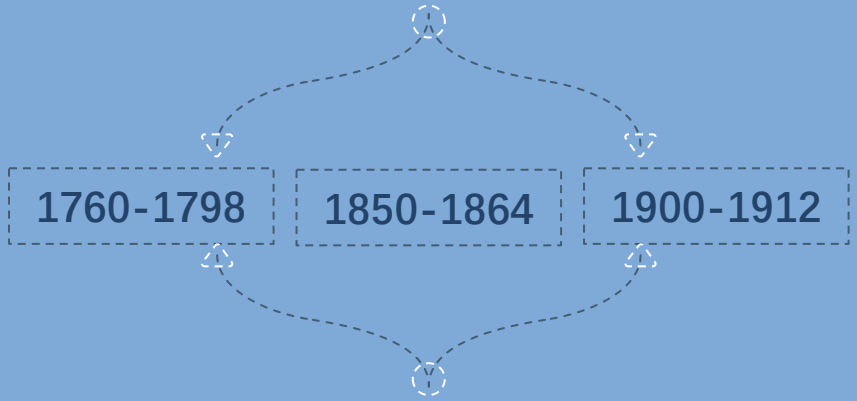
57,251
武官总人数

242,499
文官总人数

4,629,016

CGED-Q JSL records（1760-1912）

Public release

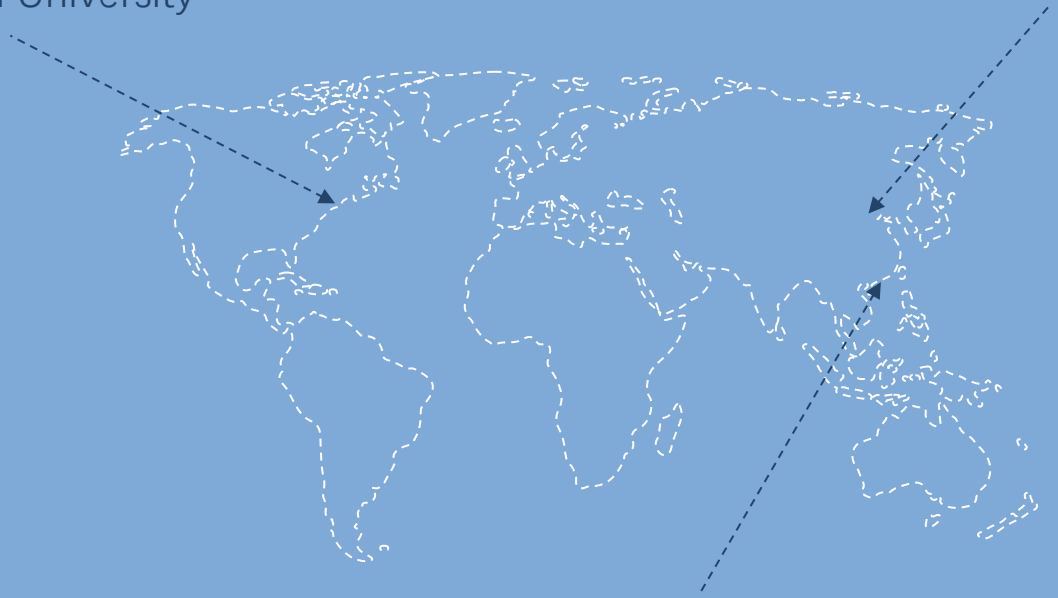


1,232,355

records in 89 Editions

Dataverse
Harvard University

清史数据共享平台
Renmin University of China



DataSpace
The Hong Kong University of
Science and Technology

02

Why use RStudio ?

2.1.1 为什么选择R语言和RStudio?

- R语言的优势

  - 免费开源，经济成本低
  - 功能强大，外扩功能包众多
  - 不断更新，多方进行

- 需要注意的问题

  - 学习和运用R语言需要一定的数理基础
  - 学习和运用R语言需要付出相当的时间成本

2.1.2 如何下载和安装RStudio?

- R语言下载网址：推荐清华大学镜像网站
  - https://mirrors.tuna.tsinghua.edu.cn/CRAN/

- Rstudio下载网址：
  - RStudio Desktop - Posit

2.1.3 R语言的自学资源

- 约翰霍普金斯大学和杜克大学的R语言公开课程、北大李东风老师的《R语言教程》

- 《R语言实战》以及可视化教材《R语言数据可视化实战》等

- tidyverse项目
  - `ggplot2`数据可视化
  - `dplyr`数据处理
  - `tidyr`数据清理
  - `readr`数据提取
  - `purrr`函数化编程
  - `tibble`数据类型定义

- 数据编程共享网站
  - Quick-R
  - Rweekly
  - Github
  - CSDN

2.1.4 CGED-Q JSL数据库公开版下载地址

- 缙绅录数据库的部分内容已经向社会公开，下载网址包括
  - 香港科技大学数据空间
  - 哈佛大学数据储存库（Harvard Dataverse Lee-Campbell Group Dataverse）
  - 中国人民大学清史数据共享平台
- 以香港科技大学数据空间为例，下载方式为
  - 进入香港科技大学数据空间，根据下载目录选择最新版本的缙绅录数据库
  - 已公开的缙绅录数据库包含三个年代（1760-1798；1850-1864；1900-1912）
  - 我们推荐使用者在港科大数据空间下载最新版本dta版本的缙绅录数据库
  - 我们会不定期更新数据库的版本，请使用者留意

2.1.5 下载步骤

- 下载页面中包括
  - 数据库（xlsx格式、dta格式、tab格式）
  - 用户指南、使用指南
  - 重要声明
  - 数据库处理便捷代码集等资源

- 我们建议使用者选择
  - `CGED-Q Public Release 1760-1798 1 July 2024 STATA.tab`（右边下载按钮下拉后选择`Stata 14 Binary Original File Format`）
  - `CGED-Q Public Release 1850-1864 19 Apr 2022`
  - `CGED-Q Public Release 1900-1912 22 May 2022.dta`

- 这三个分别是缙绅录数据库乾隆后期部分、太平天国部分和清末部分，使用dta格式的目的是方便在RStudio中导入和分析

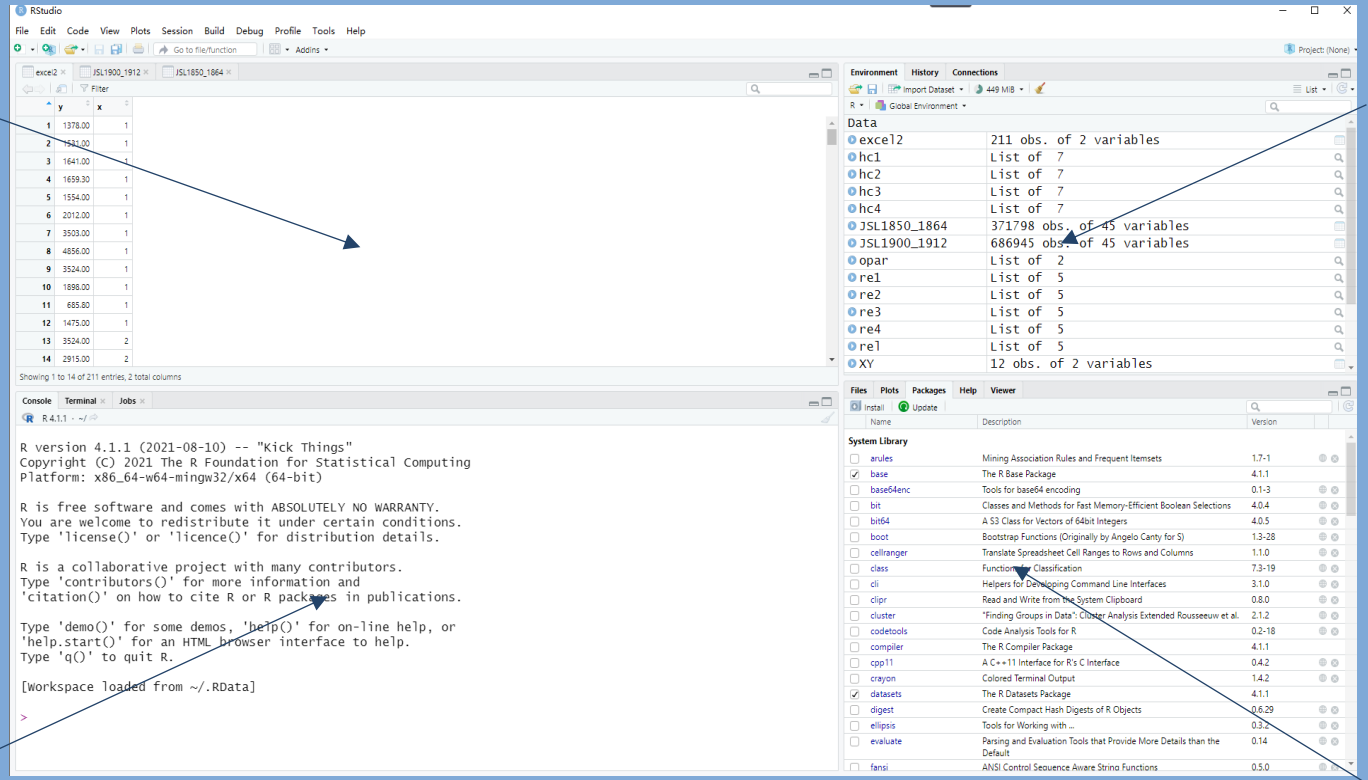
- 另外，数据库整理的便捷代码集我们也建议下载保存，方便后续数据库的分析，同时，我们强烈建议使用者在使用数据库之前阅读用户指南和重要声明

03

Rstudio的界面

界面

命令编写区
（区域1）



数据窗口（区域2）

控制台（区域3）

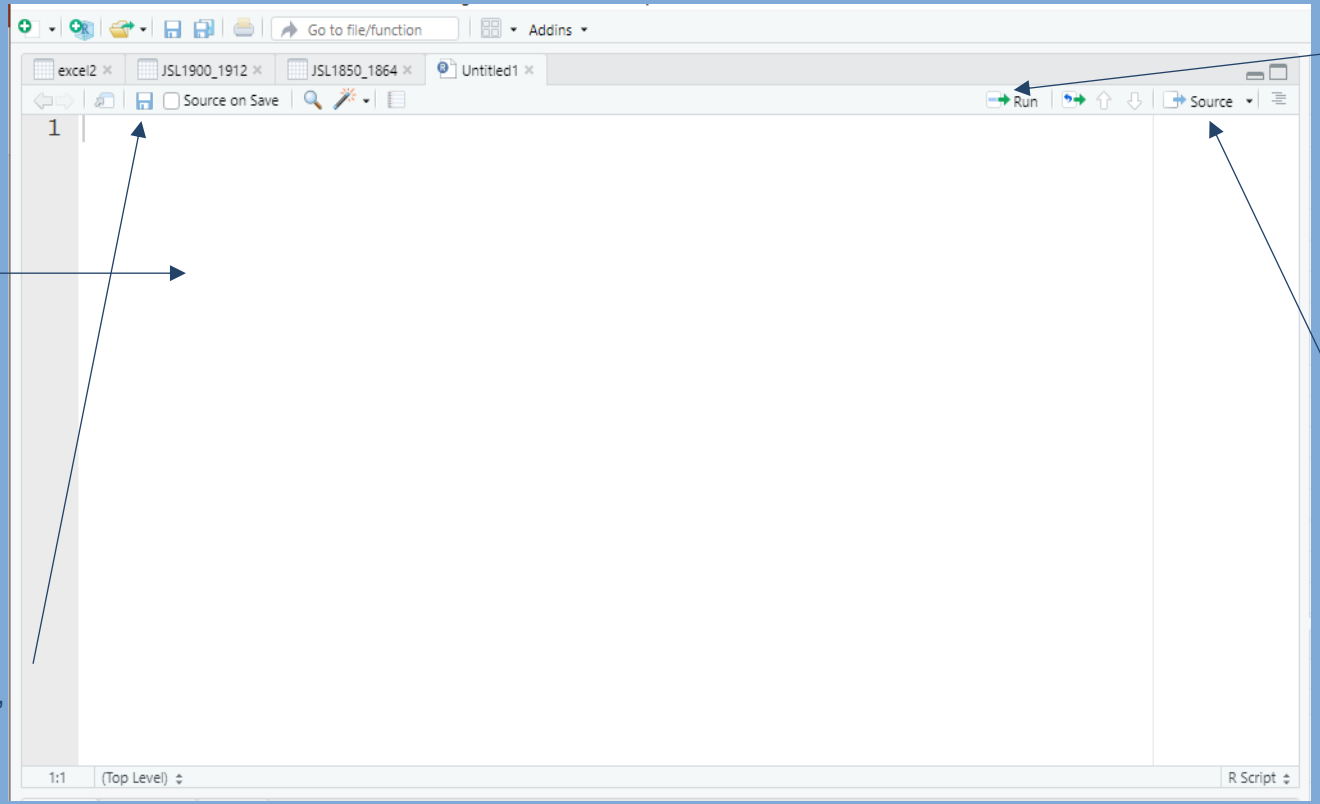
输出窗口（区域4）

区域1：命令编写区

(打开R若没有区域1，请依次点击点击File>New File>R script)

1.中间的空白区域可以编写代码可以编写代码，可随时增减、修改代码，添加注释

3.保存为source文件（一个代码集合，保存为后缀为.R的文件），类似于STATA的do文件



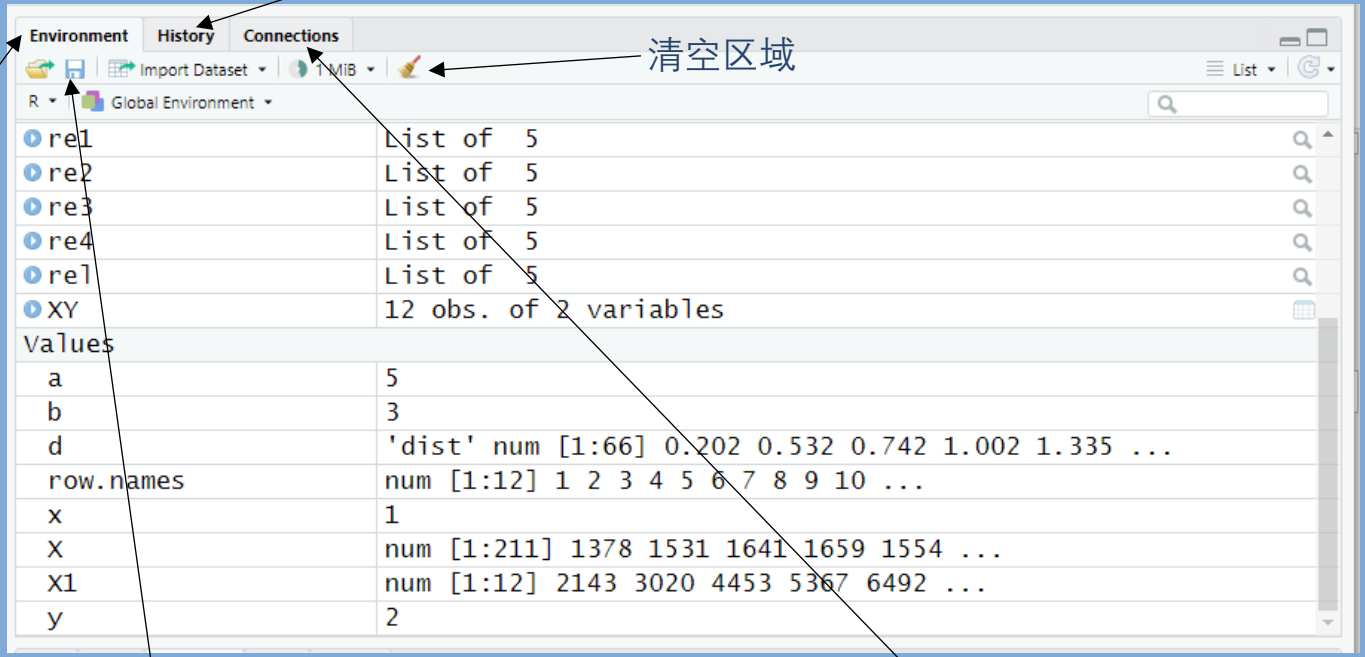
2.代码编写完成后点击Run运行

4.运行source文件

区域2：数据窗口

运行过的所有代码都在历史区

清空区域



环境区。数据集的信息所在区域，展示导入的数据集

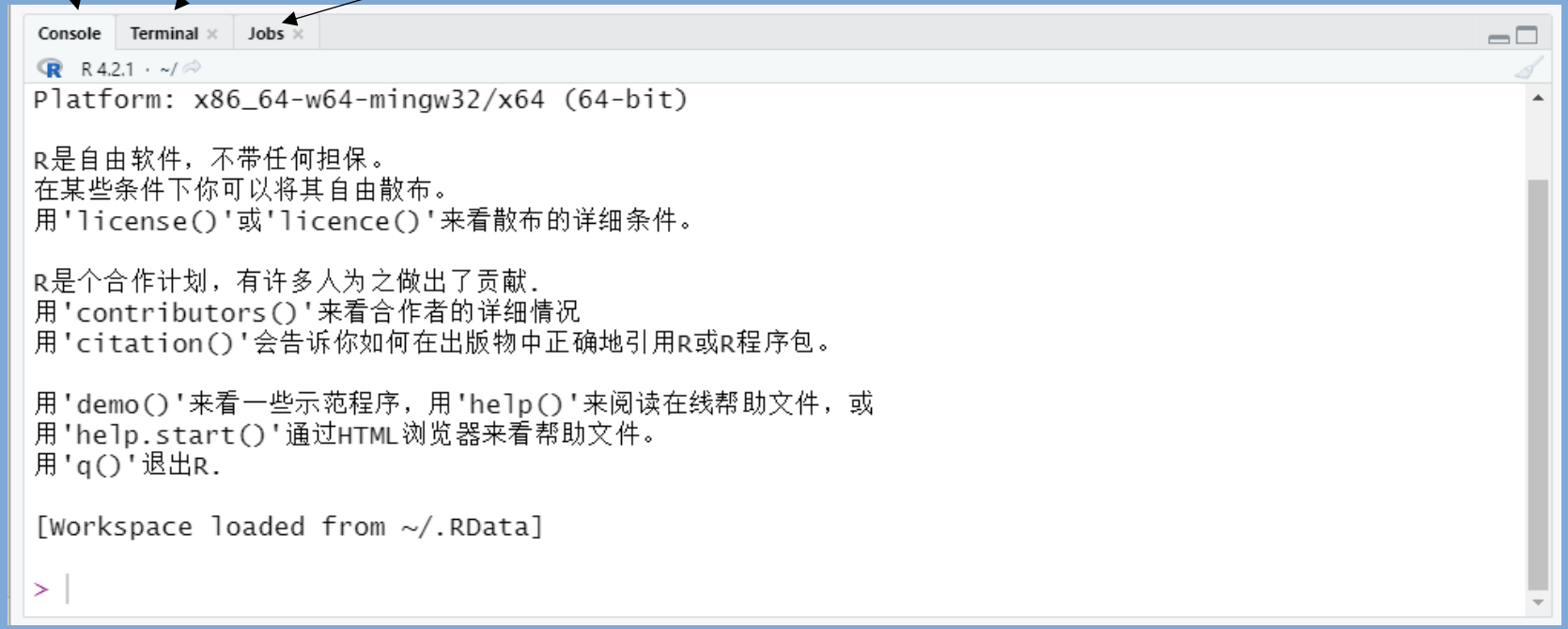
保存数据空间的所有数据，生成一个类型为R Data的文件

可以连接到各种数据源，并浏览连接内的对象和数据

区域3：控制台

控制窗口。输入代码以运行

安装包的工作进度可以在这里看到

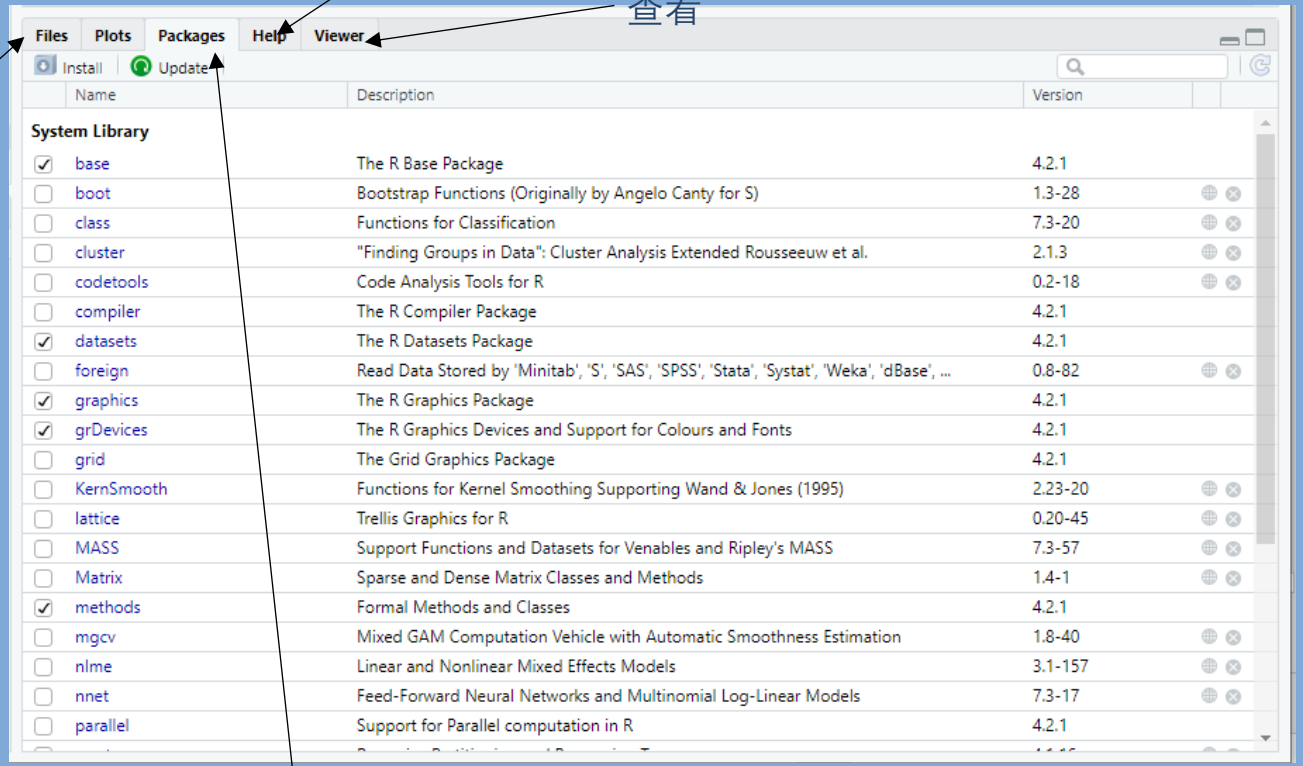


区域4：输出窗口

图片输出

帮助

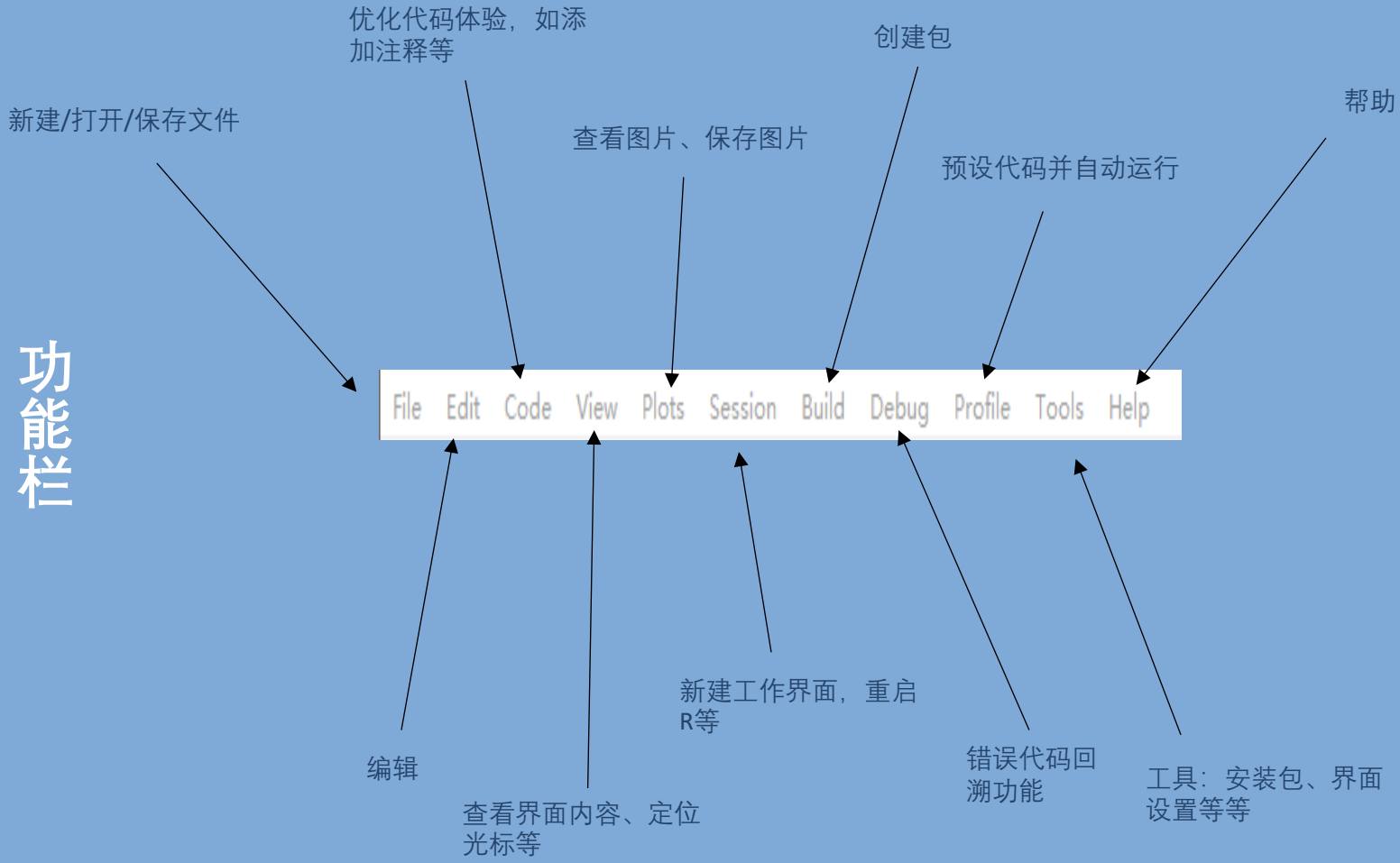
查看



你所保存的文件都在这里

R的扩展包

功能栏


新建/打开/保存文件
优化代码体验，如添加注释等
查看图片、保存图片
创建包
预设代码并自动运行
帮助
File Edit Code View Plots Session Build Debug Profile Tools Help
编辑
查看界面内容、定位光标等
新建工作界面，重启R等
错误代码回溯功能
工具：安装包、界面设置等等

04

导入数据库

2.2 导入数据库

方法一：利用Import功能导入

- 先要保证联网，因为RStudio下载资源包需要转到镜像网站下载，没联网无法开启Import功能
- 之后下拉区域2中`Environment`的`Import dataset`选项，选择需要导入的文件类型
- 然后点击浏览，选择文件，预览无误后，点击Import即可
- 数据库较大，可能需要稍等几分钟
- 或者点击功能栏`File`下的`Import dataset`，效果是一样的
- Import导入快捷方便，推荐导入缙绅录dta版本

方法二：利用函数

```
library(haven)

JSL1900_1912 <- read_dta(
  "/Users/jc/Documents/2-数据库-缙绅录数据库/原始文件/缙绅录公开版（新）/1900-1912 19 Apr 2022.dta")

# 更换你自己的文件储存路径
```

注意事项

- 这种方法的思路和Import导入是一样
- 不同点是Import是自动运行函数，而代码需要自己调整路径
- 请使用者注意根据文件储存位置调整`read_dta()`括号中的内容，同时请记得添加引号

方法三：复制粘贴

- 将需要的内容复制到剪切板，然后输入代码：

```
data <- read.table("clipboard",head=T)
```

- 即可完成剪切板内容的导入
- 但复制粘贴这种方法只适合于较小的数据，对于大型的数据库来说，Import和代码导入是最实用高效的导入数据的方式

05

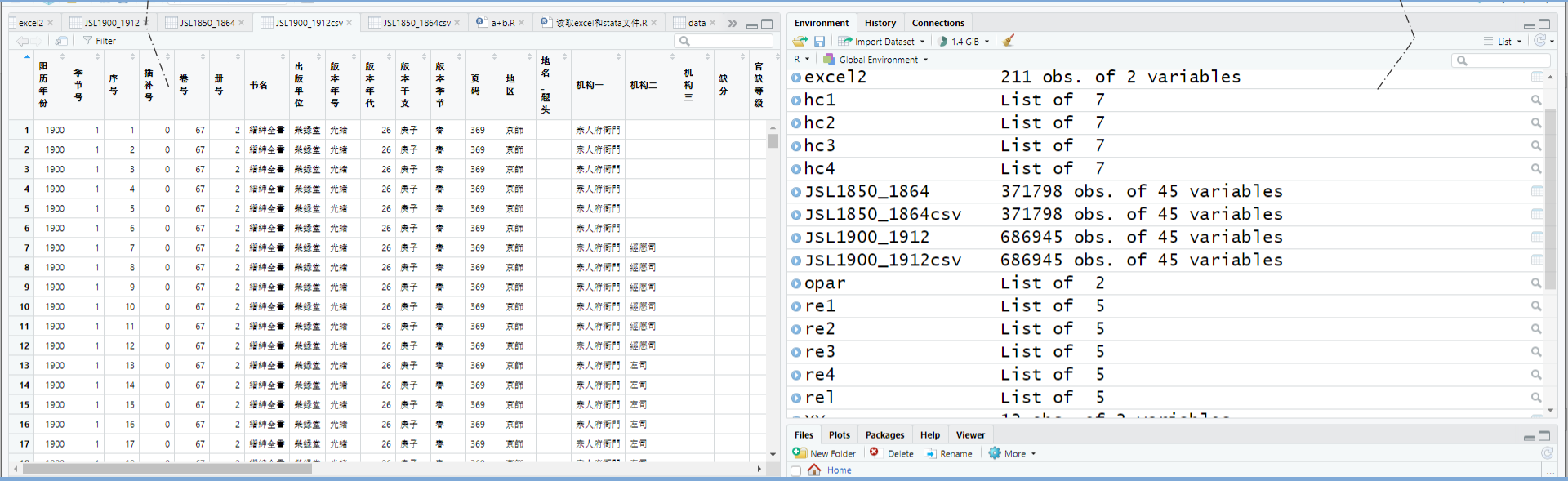
查看、保存数据

2.3.1 查看数据

在区域1看到数据库的内容，在区域2可以看到数据库的信息

数据库的所有记录和变量

数据的基本信息，总记录数和变量数



2.3.2 保存数据

Rstudio保存的主要文件类型

- RStudio的保存系统主要可以保存以下几个核心类型：
  - 以`.Rproj`为后缀的文件，这是R语言的主项目文件，保存了运行R语言的元数据信息
  - 以`.R`为后缀的文件，这是R的脚本文件`R Script`，可编写并运行代码
  - 以`.Rhistory`为后缀的文件，这是根据区域二的`history`保存的文件，有R语言的会话历史
  - 以`.Rmd`为后缀的文件，这是`R Markdown`文件，混合了Markdown文本和R代码块，可用于创建可重复生成的研究报告、论文和演示文稿
  - 以`.RData`为后缀的文件，这是区域二中`Enviroment`的数据文件，可以单独保存，亦可以导入到RStudio中

保存数据的步骤

- 结合以上内容，保存数据工作的重点就较为清晰了：
- 第一步，点击功能栏`File`下拉项`Save All`保存所有脚本文件（`.R`、`.Rmd`等）
  - 我们强烈建议使用者对载有重要代码信息的`.R`、`.Rmd`文件进行备份，以免重要代码文件丢失
  - 但请注意`Save All`并不会保存`Enviroment`中的对象（比如数据框、变量等）

保存数据的步骤

- 第二步，要保存工作空间（即区域二中`Enviroment`的数据文件）
  - 可以使用`save.image()`函数指定路径进行保存
  - 或者通过RStudio的菜单`Session -> Save Workspace As...`保存为`.RData`文件
  - 同时，还可以在`Project Options > General`中可配置退出时是否自动保存工作空间`.RData`和命令历史`.Rhistory`
  - 我们强烈建议使用者对`.RData`文件进行备份，以免重要数据文件丢失

保存数据工作的意义

- 只要我们将脚本文件和数据空间进行及时保存和备份，即便遇到R语言出错或者电脑宕机等情况也能从容应对
- 只需要导入相应脚本文件和数据空间文件便能再次回到原来的进度中
- 保存数据工作异常重要，请确保在 RStudio 中的工作得到系统化保存，避免数据丢失，并保持项目的可重复性和可维护性

Thanks!