

R 语言编程：基于 tidyverse

第 24 讲 R Markdown

张敬信

2022 年 12 月 6 日

哈尔滨商业大学

R 用在研究的优势体现在“文档沟通”，这也是 tidyverse 核心 workflow 之一：

- R Markdown¹: 可重复研究
 - 最大限度地降低你的项目/论文等涉及到的重复性工作
 - 提供统一的框架编写代码、结果和文字叙述，支持几十种输出格式，包括 html、pdf、word、幻灯片等。
- Git: 版本控制，协作和跟踪代码和分析的“改动”
- Shiny: 开发 Web app, 网页交互
- 开发 R 包: 分享

¹Jupyter Notebook 是 Python 用户常用的可协作框架，也支持 R，缺点是存储为 JSON 文件，不便于用 Git 跟踪“改动”。

一. Markdown 简介

Markdown 是一种可以使用普通文本编辑器编写的轻量化标记语言，通过简单的标记语法，它可以使普通文本内容具有一定的格式。

Typora 是一款好用的 Markdown 编辑器，Markdown 文件扩展名为.md，可导出为 html，word，pdf，latex 等文件。

■ 标题

```
# 一级标题
## 二级标题
### 三级标题
#### 四级标题
##### 五级标题
##### 六级标题
```

一级标题

二级标题

三级标题

四级标题

五级标题

六级标题

■ 无序列表

- 有序列表
- 下一级
 - 下下一级
- 有序列表
- 有序列表
- [] 其它列表
- [x] 其它列表

- 有序列表
 - 下一级
 - 下下一级
- 有序列表
- 有序列表
- ☐ 其它列表
- ☒ 其它列表

■ 有序列表

1. 有序列表
1. 有序列表
3. 有序列表

1. 有序列表
2. 有序列表
3. 有序列表

■ 引用与文字

> 这是一段引用

>> 二级引用

****加粗文本****

斜体文本

~~删除文本~~

这是一段引用

二级引用

加粗文本

斜体文本

删除文本

- 下划线文本、高亮文本、下标、上标：

<code><u>下划线文本</u></code>	<u>下划线文本</u>
<code>==高亮文本==</code>	高亮文本
<code>H-2-O</code>	H ₂ O
<code>X^2^</code>	X ²

- 设置字体字号颜色，可以用 HTML 语法：

```
<font color = red size = 5> 红色放大文字 </font>
```

用空行分段，用 ` ` 缩进 1 个汉字，输出控制符号需要加转义符\，用`---`生成分割线。

■ 数学公式，支持 Latex 语法

一元二次方程 $ax^2 + bx + c = 0$, $a \neq 0$

的求根公式为：

\$\$

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

\$\$

一元二次方程 $ax^2 + bx + c = 0$, $a \neq 0$ 的求根公式为：

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

■ 代码块，高亮语法显示代码，支持各种程序语言

```
```R
```

```
x = c(2,3,6)
```

```
x ^ 2
```

```
```
```

行内代码，`plot()` 是绘图函数

```
1. x = c(2,3,6)
```

```
2. x ^ 2
```

行内代码，`plot()` 是绘图函数

- 插入图片，提供本地图片的相对或完整路径，或网络图片的网址：

```
![图片描述](xxx.png){width = 80%}
```

- 或者用 HTML 代码：

```

```

- 居中可以套一个 <center>...</center>; 加图标题可以在图片下面增加一行：

```
<center><b>图 1</b> 标题文字</center>
```

- 插入超链接

```
[超链接描述](超链接网址)
```


- 表格，各列分别左对齐、右对齐、居中

7. 绘制表格

项目	价格	数量	
:-----	-----:	:-----:	
计算机	\ \$1600	5	
手机	\ \$12	12	
管线	\ \$1	234	

7. 绘制表格

项目	价格	数量
计算机	\$1600	5
手机	\$12	12
管线	\$1	234

■ 交叉引用

3.1.1 节 交叉引用 {#311}

.....

见 [3.1.1节](#311)

3.1.1 节 交叉引用

.....

见 3.1.1节

■ 脚注

HTML是标记性语言^[1],

[^a]: 将文本以及文本相关的其他信息结合起来, 展现出关于文档结构和数据处理细节的电脑文字编码。

HTML是标记性语言^[1],

[1] 将文本以及文本相关的其他信息结合起来, 展现出关于文档结构和数据处理细节的电脑文字编码。↩

二. R markdown 基础

R markdown 的主要开创和发扬光大者是谢益辉，是 Markdown 的 R 语言扩展，除了具备一般 Markdown 语法功能之外，最关键的是可以在 R markdown 中插入代码块，并能运行代码，将代码运行结果显示出来。

使用 R markdown 能够让您只需关注内容创作，借助现成模板 + 少量自己定制，就可以一劳永逸地、自动化地制作：

- 数据分析报告和文档 (rmarkdown 包、officedown 包)
- 期刊论文 (rticles 包)
- 书籍 (bookdown 包)
- 个人简历 (pagedown 包)
- 个人博客网站 (blogdown 包)
- 幻灯片 (xaringan 包)
- 交互报表 (flexdashboard 包)

使用 R markdown 能够解决的痛点 (Xie et al., 2021):

- 在用 R 或者其他数据分析工具时, 经常需要在 Word 里写结论, 在脚本里敲代码, 在图表区生成图, 将它们复制粘贴到一起后, 还要操心格式问题, 有没有什么自动化的方法?
- 工作需要日常性的数据产出, 如何创作一篇参数化、可复用的文档模板, 从此可以在更新数据的同时同步结论和图表?
- 如何确保分析过程和结论是可重复的, 别人是否能用同样的数据得到我的结论?
- 不了解网页开发, 如何在报告中插入可交互的图表和网页元素?

R Markdown 文件是后缀名为 .Rmd 的纯文本文件，其编译过程如下：

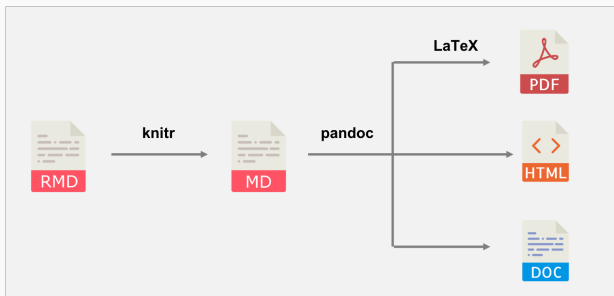


图 1: Rmd 文件编译过程

.Rmd 文件提交给 knitr 后，将执行代码块创建一个新的包含代码和运行结果的 Markdown 文件 (.md)，再被 Pandoc 处理生成最终的输出文件。

1. R markdown 简单示例

需要先安装 rmarkdown 包。在 Rstudio, 依次点 New File--R Markdown..., 进入.Rmd 文件创建向导:

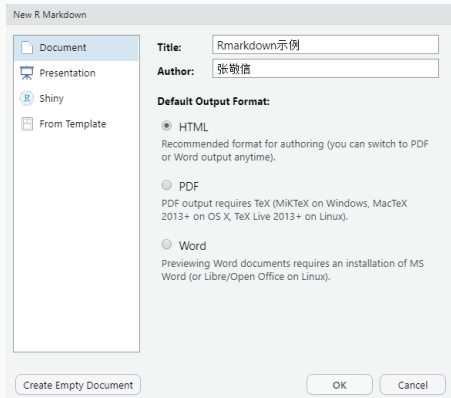


图 2: 新建 R markdown

填写标题和作者，默认输出格式为 html，可选 pdf（需要 Latex 环境）、word（需要 Office 环境）。

新建的.Rmd 文件，提供了 R markdown 最小示例模板，可修改为自己的内容，里面大部分语法都遵循 Markdown 语法，少部分语法是 R markdown 专用语法。

```

1- ---
2- title: "Rmarkdown 示例"
3- author: "张敬信"
4- date: "2021/7/26"
5- output: html_document
6- ---
7-
8- {r setup, include=FALSE}
9- knitr::opts_chunk$set(echo = TRUE)
10-
11-
12- ## R Markdown
13-
14- This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF,
15- and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
16-
17- When you click the Knit button a document will be generated that includes both content as
18- well as the output of any embedded R code chunks within the document. You can embed an R code
19- chunk like this:
20-
21- {r cars}
22- summary(cars)
23-
24- ## Including Plots
25-
26- You can also embed plots, for example:
27-
28- {r pressure, echo=FALSE}
29- plot(pressure)
30-
31- Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R
32- code that generated the plot.

```

图 3: 自带 R markdown 模板

从.Rmd 到输出目标文档的过程称为 knit, 有两种方式:

- 点击 Knit 按钮, 或从其下拉菜单选择 Knit to HTML, Knit to PDF, Knit to Word
- 在 Console 窗口执行命令:

```
rmarkdown::render("rmddemo.Rmd", "html_document")
```

2. YAML

第 1-6 行位于一组“—”之间的部分，称为 YAML，由若干“键: 值”对组成，控制 R markdown 如何编译.Rmd 文件。

- 日期的值，可以改用行内 R 代码返回今天的日期：

```
date: "`r Sys.Date()`"
```

- output: html_document 设置输出格式为 html 文档，进一步设置深度为 2 的目录：

```
output:  
  html_document:  
    toc: true  
    toc_depth: 2
```

R markdown 的主要优势之一就是可以从单个文件源生成多种输出格式，常用的有：

- `html_document/html_vignette`: 网页文档 (html)
- `md_document/github_document`: markdown 文档 (md)
- `pdf_document`: pdf 文档 (pdf)
- `word_document`: word 文档 (docx)
- `powerpoint_presentation`: PPT 幻灯片 (ppt)
- `beamer_presentation`: Latex beamer 幻灯片 (pdf)
- `ioslides_presentation`: ioslides 幻灯片 (html)
- `slidy_presentation`: slidy 幻灯片 (html)

输出格式的其他选项

- `toc_float: true`: 浮动目录
- `number_sections: true`: 添加标题编号
- `code_folding: hide`: 代码折叠
- `fig_width: 7, fig_height: 6`: 固定图片宽高
- `fig_caption: true`: 添加图片标题
- `df_print: kable`: 数据框输出表格样式
- `highlight: tango`: 设置语法高亮²
- `theme: united`: 设置主题³
- `keep_md: true`: 保存中间.md 文档
- `citation_package: natbib`: Latex 参考文献格式用 natbib 宏包
- `reference_docx: "template.docx"`: 自定义 word 模板

²可选语法高亮有: “default”, “tango”, “pygments”, “kate”, “monochrome”, “espresso”, “zenburn”, “haddock”, “textmate”.

³可选主题有: “default”, “cerulean”, “journal”, “flatly”, “readable”, “spacelab”, “united”, “cosmo”, “lumen”, “paper”, “sandstone”, “simplex”, “yeti”.

以使用自定义 word 模板为例

- 先创建一个输出到 word 文档的.Rmd，在原始模板增加新的格式（内容随意），比如三级标题、参考文献等，然后 Knit to Word 生成 Word 文档，
- 打开继续在 Word 中修改：增加样式、修改格式等（内容随意），比如保存为 template.docx。
- 在.Rmd 的 yaml 中设置输出格式：

output:

word_document:

reference_docx: "template.docx"

则可以使用该参考模板，并在编译 Word 文档时，读取模板中的样式并将其应用到新文档中。

目前, R 与 Office 深度交互已有officeverse 系列包:

- officer 包: 用 R 生成 Word/PowerPoint
- officedown 包: R markdown 与 Word/PowerPoint 沟通
- flextable 包: 定制精美表格
- mschart 包: 根据数据绘制 Office 风格图形
- rvg 包: 生成可修改的矢量图

另外, yaml 的设置还可以借助 ymlthis 包, 有图形交互界面设置 yaml。

3. 代码块

- 第 8-10、18-20、26-28 行是**代码块**，位于````{r}` 和 `````之间，表示 R 语言代码块⁴
- 借助 `reticulate` 包也支持 Python 代码，还支持 Shell，SQL，Stan，C/Fortran，C++，Julia，JavaScript/CSS 以及 SAS，Stata 等代码，需要配置相应开发环境。
- 语言名后面跟着的词语是该代码块的命名，好处是：可在导航栏按名字浏览代码块，让代码生成的图形有意义地命名，避免在缓存中重复计算。
- 其后逗号隔开的是块选项，用来控制代码和运行结果的输出方式。

⁴快捷键：[Win]Ctrl + Alt + I, [Mac]Cmd + Option + I.

常用的块选项（只写非默认情形）：

- `eval = FALSE`：只显示代码，不运行代码
- `echo = FALSE`：不显示代码，只显示运行结果
- `include = FALSE`：运行代码，不显示代码和运行结果
- `tidy = TRUE`：整洁代码格式
- `message = FALSE`：不输出提示信息，比如包的载入信息
- `warning = FALSE`：不输出警告
- `error = TRUE`：忽略错误，继续编译文档
- `collapse = TRUE`：代码块结果放在一个文本块
- `cache = TRUE`：缓存运行结果，能加速后续再编译

其他选项:

- `results = "hide"`: 隐藏输出结果
- `fig.width`, `fig.height`, `fig.align`, `fig.cap`: 设置输出图形的宽和高 (英寸)、对齐、标题
- `out.width`, `out.height`: 设置输出图形的宽和高 (百分比)

例如, 设置代码块不输出代码本身、消息、警告, 只输出运行结果:

```
```{r echo = FALSE, message = FALSE, warning = FALSE}  
具体代码
```
```

- 有些块选项更建议进行全局设置，比如示例中的 8-10 行：

```
```{r setup, include=FALSE}  
knitr::opts_chunk$set(echo = TRUE)
```
```

就是设置全局所有代码块都” **不显示代码，只显示运行结果**”，有特殊需要的代码块再局部设置。

- R markdown 还支持行内代码，即在文字叙述中间使用 R 代码，基本格式是 ``r ...``

```
mdl = lm(mpg ~ disp, mtcars)
b = mdl$coefficients
```

上面回归方程的斜率为 ``r b[2]``，完整的回归方程为 ``r mpg = b[1] + b[2] * disp``。

- 渲染后，上述回归系数将变成具体数值。

4. 插入图片、表格

- 对于可以用 R 代码绘制的图形，直接在代码块绘制即可。
- 插入图片除了用前文介绍的 Markdown 语法，还可以用 `knitr::include_graphics()` 函数：

```
```${r, echo=FALSE , out.width="50%", fig.cap=" 图标题 ",  
fig.align="center"}
knitr::include_graphics("xxx.png")
```
```

- RStudio 1.4 开始，提供了可视化 Markdown 编辑器，点击编辑窗口右上角按钮可切换过去：

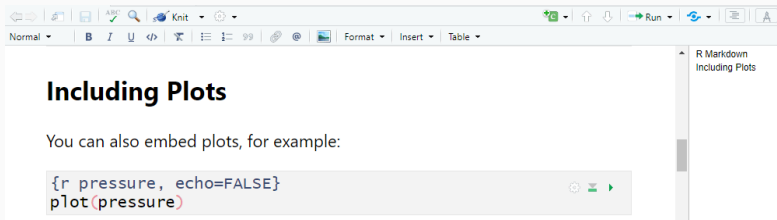


图 4：启用可视化 Markdown 编辑器

- 这就相当于是 Markdown 编辑器，以菜单操作的方式实现各种 Markdown 语法。点击“插入图片”按钮，浏览找到图片即可完成插入，插入图片下方可以调整图片大小。

- 插入简单表格除了用前文介绍的 Markdown 语法，也可以切换到可视化 Markdown 编辑器，点击“插入表格”按钮，类似 word 中的插入表格操作。
- 对于现成的数据框或矩阵，可以用 `knitr::kable()` 生成简单表格，参数 `align` 设置各列对齐方式，`digits` 设置小数位数，`col.names` 设置新列名，`caption` 设置表标题

```
knitr::kable(mtcars[1:3,1:7], align = "lccrr", digits = 2,  
             col.names = str_c("x", 1:7),  
             caption = " 部分 iris 数据")
```

表 1: 部分 iris 数据

| | x1 | x2 | x3 | x4 | x5 | x6 | x7 |
|---------------|------|----|-----|-----|------|------|------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.62 | 16.5 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.88 | 17.0 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.32 | 18.6 |

5. 表格输出

有十几个包致力于通过 R 语言编程做出更加精美表格，其中最优秀的几个是：

- **kableExtra 包**: `knitr::kable()` 的扩展，支持管道，可生成复杂精美的 html 或 LaTeX 表格
- **huxtable 包**: 支持输出格式最全面，特别是 Latex 输出，丰富的自定义功能
- **flextable 包**: 从 R markdown 创建用于报告或出版的 html, pdf, Word, PowerPoint 表格
- **gt 包**: RStudio 出品，用整洁语法组合不同的表格组件创建表格，暂不支持 Latex, pdf 输出
- **DT 包**: 多与 Shiny 配合，将数据表渲染成 HTML
- **reactable 包**: 基于 React-Table 库的交互表格

它们的操作和功能类似，都是通过相应函数精细控制，比如单元格背景、边框、对齐方式、颜色、数字格式等。

(1) 导出三线表到 word

选用更擅长与 Office 交互的 `flextable` 包实现，这里只演示几个功能：增加标题、带合并列的题头行、文字对齐、颜色、加粗、合并单元格、高亮文字。

```
library(flextable)                # word, ppt
iris[1:5,] %>% flextable() %>%
  set_caption(" 定制表格示例") %>%
  add_header_row(colwidths = c(2, 2, 1),
                 values = c("Sepal", "Petal", "")) %>%
  align(align = "center", part = "all") %>%
  color(color = "red", part = "header") %>%
  bold(bold = TRUE, part = "header") %>%
  merge_v(j = 3:4) %>%
  highlight(i = ~ Sepal.Length < 5, j = 1,
           color = "yellow") %>%
  save_as_docx(path = "output/threelinetable.docx")
```

若只运行到 `save_as_docx()` 之前, 则在 Viewer 窗口显示结果表格。
最终写入 word 的效果如下:

| 定制表格示例 | | | | |
|--------------|-------------|--------------|-------------|---------|
| Sepal | | Petal | | |
| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | | | setosa |
| 4.7 | 3.2 | 1.3 | | setosa |
| 4.6 | 3.1 | 1.5 | | setosa |
| 5.0 | 3.6 | 1.4 | | setosa |

图 5: 自动生成三线表

(2) 将统计模型结果整理成模型结果汇总表

期刊论文经常要求将统计模型结果以规范格式的表格展示，以最常用的回归分析结果表为例。

`modelsummary` 包基于 `broom` 和 `broom.mixed` 整洁模型结果，可与上述最优秀的四个表格包连用，制作精美的可定制统计模型结果表格，支持各种常见输出格式。

模型汇总表通常是需要 `pdf` 或 `latex` 代码，用 `modelsummary()` 函数，可接受多个模型对象的 `list`，选择相应的参数定制想要的表格，这里只演示修改参数名、标记显著性星号、小数位数、表标题、不输出部分统计量等。

若上述定制已能满足要求，可以设置参数 `output = "file.tex"` 等，可直接导出到文件；否则，可以设置输出到其他表格对象，比如 `output = "huxtable"`，则得到 `huxtable` 表格对象，这就相当于转到 `huxtable` 包，继续做相应的美化修改，再导出到文件。

```
df = read_csv("data/Guerry.csv")
models = list(
  "OLS" = lm(Donations ~ Literacy + Clergy, data = df),
  "Poisson" = glm(Donations ~ Literacy + Commerce,
                  family = poisson, data = df))
cm = c("(Intercept)" = "Constant",
       "Literacy" = "Literacy (%)",
       "Clergy" = "Priests/capita")
cap = "Regression Tables with modelsummary"
```

- 先用 `modelsummary()` 定制回归分析结果表，导出为 `huxtable` 对象做美化：增加带合并的表头行、设置第 3 行字体颜色，设置第 5 行背景色；再导出到 pdf（中文有编码问题）

```
library(modelsummary)
library(huxtable)          # pdf
modelsummary(models, output = "huxtable", coef_map = cm,
              stars = TRUE, fmt = "%.2f",
              title = cap, gof_omit = 'IC|Log|Adj') %>%
# 转到 huxtable
set_text_color(row = 4, col = 1:ncol(.),
               value = "red") %>%
set_background_color(row = 6, col = 1:ncol(.),
                     value = "lightblue") %>%
quick_pdf(file = "output/tablepdf.pdf")
```

导出到 pdf 的表格效果:

| Table 1: Regression Tables with moelsummary | | |
|---|-------------------------|-------------------|
| | OLS | Poisson |
| Constant | 7948.67***
(2078.28) | 8.24***
(0.01) |
| Literacy (%) | -39.12
(37.05) | 0.00***
(0.00) |
| Priests/capita | 15.26
(25.74) | |
| Num.Obs. | 86 | 86 |
| R2 | 0.020 | |
| F | 0.866 | |
| + p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001 | | |

图 6: 回归结果表导出到 pdf 效果

- 若要将表格导出到 latex 源代码, 需要设置 `output = "latex"`, 在 `kableExtra` 下美化, 再用 `save_kable()` 保存到.tex 文件:

```
library(kableExtra)          # latex
modelsummary(models, output = "latex", coef_map = cm,
              stars = TRUE, fmt = "%.2f",
              title = cap, gof_omit = 'IC|Log|Adj') %>%
# 转到 kableExtra
add_header_above(c(" " = 1, "Donations" = 2)) %>%
row_spec(3, color = "red") %>%
row_spec(5, background = "lightblue") %>%
save_kable("output/modeltable.tex")
```

导出到.tex 的 latex 代码效果:

```
\begin{table}

\caption{(\#tab:unnamed-chunk-27)Regression Tables with moelsummary}
\centering
\begin{tabular}[t]{lcc}
\toprule
\multicolumn{1}{c}{ } & \multicolumn{2}{c}{Donations} \\
\cmidrule{1}{3pt}r{3pt}{2-3}
& OLS & Poisson\\
\midrule
Constant & 7948.67*** & 8.24***\\
& (2078.28) & (0.01)\\
\textcolor{red}{Literacy (\%)} & \textcolor{red}{-39.12} & \textcolor{red}{0.00***}\\
& (37.05) & (0.00)\\
\cellcolor{lightblue}{Priests/capita} & \cellcolor{lightblue}{15.26} & \cellcolor{lightblue}{}\\
& (25.74) & \\
\midrule
Num.Obs. & 86 & 86\\
R2 & 0.020 & \\
F & 0.866 & \\
\bottomrule
\multicolumn{3}{l}{+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001}
\end{tabular}
\end{table}
```

图 7: 导出到 latex 文件效果

注：从回归模型对象到 latex 代码的结果表，也可以用 `stargazer`、`gtsummary` 包。另外，包寒吴霜开发的 `bruceR` 包，支持很多统计模型建模与输出结果表。

最后，R Markdown 的可重复报告，通常是先建立分析模版，然后再通过自动加载数据的方式，自动化生成分析报告。比如，想要只更换数据集就能生成同样格式的分析结果报告：

- 先准备一份可重复使用的报告模板：可重复报告.Rmd，在其 `yaml` 中设置传递数据集的参数 `name`（将用来传递数据集名字），`name` 可在标题中直接使用，正文中要获取该名字的数据集，可以用 `df = get(params$name)`

```
1 ---
2 title: "数据概览: `r name`"
3 author: "张敬信"
4 date: "`r Sys.Date()`"
5 params:
6 name: "input your data name"
7 output: html_document
8 ---
9 ## 输出数据概要
10 ```{r}
11 df = get(params$name)
12 summary(df)
13 ```
```

图 8: 可重复报告 Rmd

- 接着，只需要让“可重复报告.Rmd”在若干个数据集名字构成的字符向量上，重复应用函数 `render()`（渲染）即可，只批量地生成结果报告不需要返回结果，适合用 `walk()`：

```
library(rmarkdown)
names = c("iris", "mtcars", "CO2")
purrr::walk(names,
~ render("Reproducible.Rmd", params = list(name = .x),
output_file = paste0(.x, " 分析报告.html")))
```

注：也可以直接在 Rmd 中使用当前内存变量。

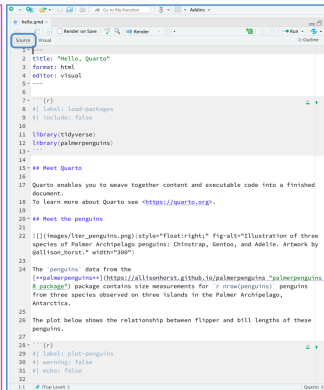
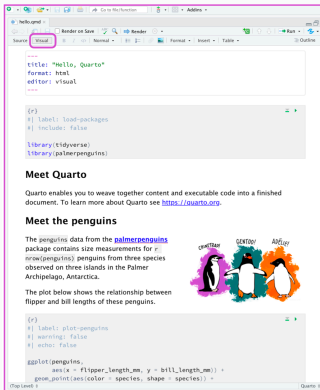
三. 新一代文档沟通: Quarto 简介

- 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。

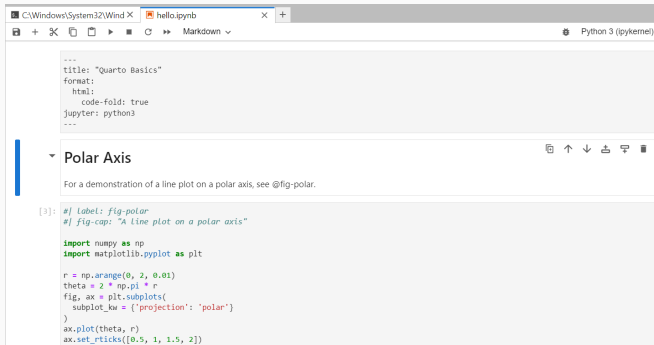
基本使用:

- (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

用官网提供的 hello.qmd 演示



(3') Jupyter Lab 用法, 在 Jupyter Lab 编写.ipynb 文件



The screenshot shows the Jupyter Lab application window. The title bar indicates the file path is C:\Windows\System32\Wind X and the file name is hello.ipynb. The interface includes a toolbar with icons for file operations and a dropdown menu set to 'Markdown'. The main content area is divided into two sections. The top section is a code editor with a light gray background, containing a YAML frontmatter block for a Quarto document. The bottom section is a markdown cell with a light gray background, featuring a heading 'Polar Axis' and a paragraph of text. Below the text is a code input area with a light gray background, containing a Python code snippet that uses NumPy and Matplotlib to create a polar plot.

```
---  
title: "Quarto Basics"  
format:  
  html:  
    code-fold: true  
  jupyter: python3  
---
```

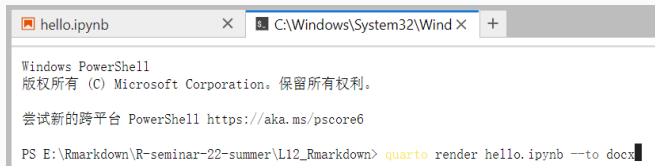
Polar Axis

For a demonstration of a line plot on a polar axis, see @fig-polar.

```
[3]: #!/ Label: fig-polar  
#!/ fig-cap: "A line plot on a polar axis"  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
r = np.arange(0, 2, 0.01)  
theta = 2 * np.pi * r  
fig, ax = plt.subplots(  
    subplot_kw = {'projection': 'polar'}  
)  
ax.plot(theta, r)  
ax.set_rticks([0.5, 1, 1.5, 2])
```


render 到文档: 打开 File--New--Terminal, 输入以下代码回车

```
quarto render hello.ipynb --to html # 或  
quarto render hello.ipynb --to docx
```



The screenshot shows a Windows PowerShell terminal window. The title bar includes a tab labeled 'hello.ipynb' and the window title 'C:\Windows\System32\Wind X'. The terminal content displays the standard PowerShell startup text: 'Windows PowerShell', '版权所有 (C) Microsoft Corporation。保留所有权利。', and '尝试新的跨平台 PowerShell https://aka.ms/pscore6'. The command prompt shows the current directory as 'PS E:\Rmarkdown\R-seminar-22-summer\L12_Rmarkdown>' and the command 'quarto render hello.ipynb --to docx' is being entered, with the cursor at the end of the line.

更多内容 Quarto, 参阅官网 (有非常丰富、分门别类的使用文档)。

本篇主要参阅 (张敬信, 2022), (Xie et al., 2021), (Xie et al., 2018), 以及包文档等, 模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

Xie, Y., Allaire, J., and Grolemund, G. (2018). *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 9781138359338.

Xie, Y., Dervieux, C., and Riederer, E. (2021). *R Markdown Cookbook*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 9780367563837.

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*.