

# 可重复性数据分析及其工业实践

第 13 届中国 R 语言会议

黄湘云

2021 年 01 月 03 日



Literate programming is a programming paradigm introduced by Donald Knuth in which a computer program is given an explanation of its logic in a natural language, such as English, interspersed with snippets of macros and traditional source code, from which compilable source code can be generated.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Literate\\_programming](https://en.wikipedia.org/wiki/Literate_programming)

下面以 `utils` 包提供的测试文档为例

```
testfile <- system.file("Sweave", "Sweave-test-1.Rnw",  
                        package = "utils")  
  
Sweave(testfile)  
tools::texi2pdf("Sweave-test-1.tex")
```



注意

若已安装 LaTeX 发行版 TinyTeX<sup>2</sup>，需要再装三个 LaTeX 宏包 `a4wide`、`ntgclass` 和 `ae`，以供文档编译

```
tinytex::tlmgr_install(c('a4wide', 'ntgclass', 'ae'))
```

---

<sup>2</sup><https://github.com/yihui/tinytex>

```

1 % -*- mode: noweb; noweb-default-code-mode: R-mode; -*-
2 \documentclass[a4paper]{article}
3
4 \title{A Test File}
5 \author{Friedrich Leisch}
6
7 \SweaveOpts{echo=FALSE}
8 \usepackage[a4wide]{}
9
10 \begin{document}
11
12 \maketitle
13
14 A simple example: the integers from 1 to 10 are
15- <<print=TRUE>>=
16 1:10
17- <<results=hide>>=
18 print(1:20)
19 @ % the above is just to ensure that 2 code chunks can follow each other
20
21 We can also emulate a simple calculator:
22- <<echo=TRUE,print=TRUE>>=
23 1 + 1
24 1 + pi
25 sin(pi/2)
26 @
27
28 Now we look at Gaussian data:
29
30- <<>>=
31 library(stats)
32 x <- rnorm(20)
33 print(x)
34 print(t1 <- t.test(x))
35 @
36 Note that we can easily integrate some numbers into standard text: The
37 third element of vector \texttt{x} is \Sexpr{x[3]}, the

```

A Test File

Friedrich Leisch

November 30, 2020

A simple example: the integers from 1 to 10 are

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

We can also emulate a simple calculator:

```
> 1 + 1
[1] 2
> 1 + pi
[1] 4.141593
> sin(pi/2)
[1] 1
```

Now we look at Gaussian data:

```
[1] -0.6150304 -1.0487100 -0.3163085 -1.2971305 -1.0181418  0.6065824  1.6053482 -0.89656
[11] 0.3220526  1.3049659 -1.1024186  1.8570753  0.3846729  1.6355841  0.5198832  0.68001
```

One Sample t-test

```
data: x
t = 0.47549, df = 19, p-value = 0.6399
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.4189988  0.6603372
sample estimates:
mean of x
0.1231692
```

Note that we can easily integrate some numbers into standard text: The third element of vector `x` is -0.316308456179337, the p-value of the test is 0.63986.

Now we look at a summary of the famous iris data set, and we want to see the commands in the code chunks:

```
> data(iris)
> summary(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min.: 4.300	Min.: 2.000	Min.: 1.000	Min.: 0.100	setosa :50
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300	versicol.:50
Median :5.800	Median :3.000	Median :4.350	Median :1.300	virginica :50

图 1: 形式上是 LaTeX 和 R 代码的混合物

# A minimal R Markdown document<sup>3</sup>

```
---  
title: "A Simple Regression" #----  
author: "Yihui Xie"          #      |  
date: "2019-01-02"           #      |  
output:                        #      |--> metadata  
  html_document:              #      |  
    toc: true                  #----
```

```
---  
We built a linear regression model. <!-- narrative -->
```

```
```${r}  
fit <- lm(dist ~ speed, data = cars)  
b    <- coef(fit)  
plot(fit)  
```
```

The slope of the regression is ``r b[1]``. <!-- narrative w/ code -->

<sup>3</sup><https://slides.yihui.org/2020-covid-rmarkdown.html#7>

```
---  
title: "A Simple Regression" #----  
author: "Yihui Xie"          #    |  
date: "2019-01-02"           #    |  
output:                       #    |--> metadata  
  html_document:             #    |  
    toc: true                 #----
```

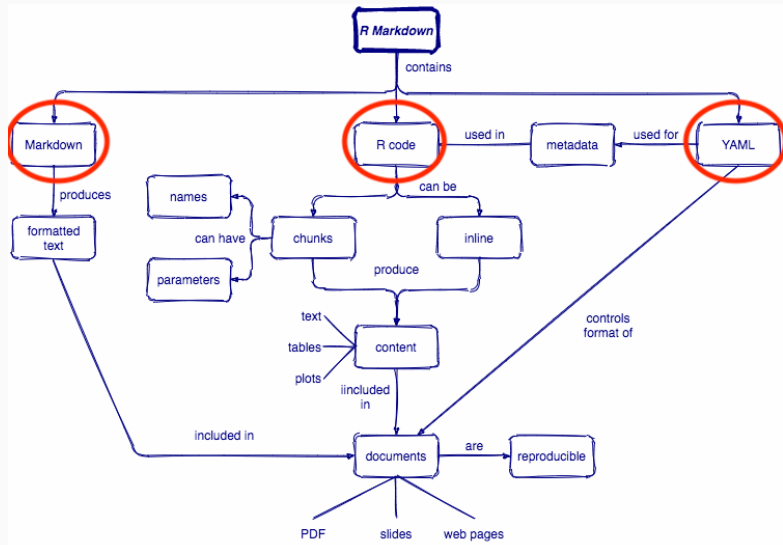
```
---  
We built a linear regression model. <!-- narrative -->
```

```
```r  
fit <- lm(dist ~ speed, data = cars)  
b   <- coef(fit)  
plot(fit)  
```
```

```
![a plot](input_files/figure-html/unnamed-chunk-1.png)
```

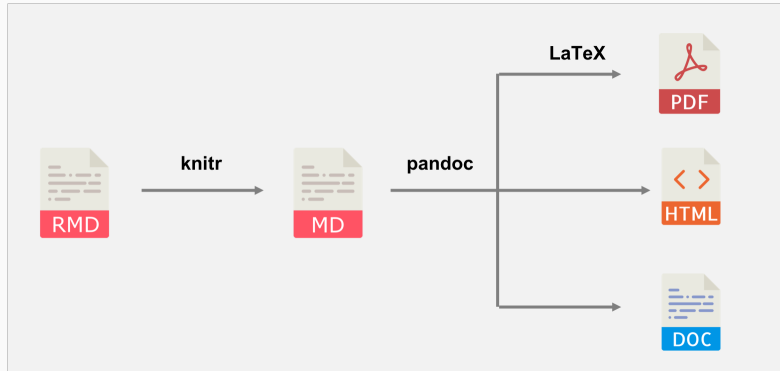
```
The slope of the regression is -17.58. <!-- narrative w/ code -->
```

## R Markdown 思维导图<sup>4</sup>



<sup>4</sup><https://github.com/rstudio/concept-maps#r-markdown>

## 编译 R Markdown 文档的过程<sup>5</sup>



R Markdown = knitr (Literate Programming)

+ Pandoc

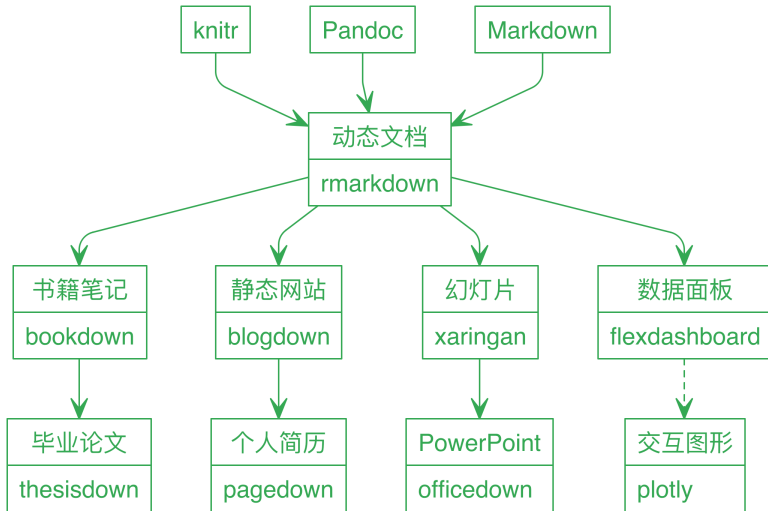
(+ LaTeX for PDF output)

<sup>5</sup><https://slides.yihui.org/2020-covid-rmarkdown.html#7>



# 从 Sweave 到 R Markdown

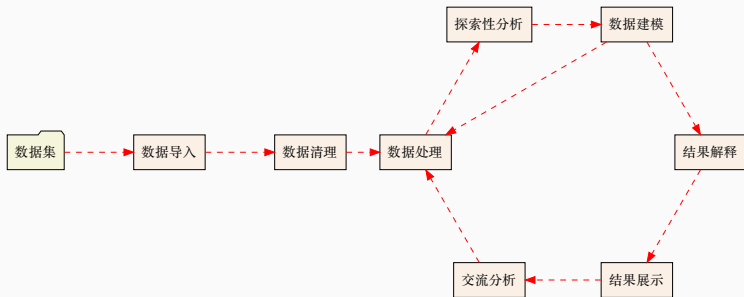




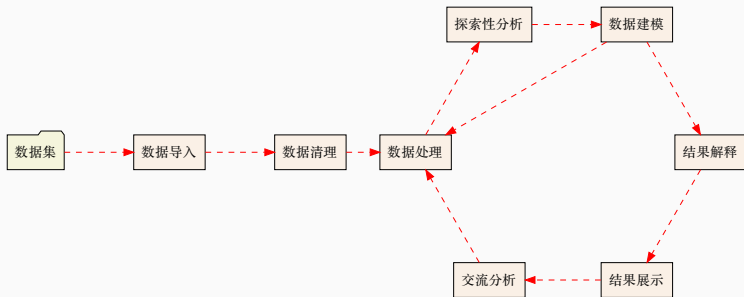
最大的不同在于稳健性和生态圈，基本覆盖 Word / LaTeX 支持的文档处理功能，满足大部分数据分析的场景，比如毕业论文、期刊论文、课程作业、幻灯片、分析报告等。高级的特性依然需要相当的技术实力支撑，包括 Pandoc、Lua、LaTeX、CSS、HTML、JavaScript、Git 和 R，在数据分析师这个行业，它们形成了一簇相当陡峭的学习曲线。

然鹅，有时候真正干活的是搜索引擎和 Ctrl + C/V!

# 数据分析工作流



# 数据分析工作流



面对大量的数据时，探索性分析是一件极其困难的事情！

```
# 连接 Spark
library(sparklyr)
sc <- spark_connect(
  master = "local[2]",
  spark_home = Sys.getenv("SPARK_HOME")
)

# 准备数据
tbl <- copy_to(sc, ggplot2::diamonds, "diamonds")

# 数据查询
library(DBI)
diamonds_db <- dbGetQuery(sc, "
  SELECT count(*) as cnt, cut FROM diamonds GROUP BY cut
")
```

## 探索性分析（续）

```
# 关闭连接
spark_disconnect(sc)

# 数据探索
library(ggplot2)
ggplot(diamonds_db, aes(cut, cnt)) +
  geom_col() +
  theme_minimal()

# 数据呈现
library(plotly)
diamonds_db %>%
  plot_ly(x = ~cut, y = ~cnt, type = "bar") %>%
  add_text(
    text = ~ scales::comma(cnt), y = ~cnt,
    textposition = "top middle",
```



```
cliponaxis = FALSE, showlegend = FALSE  
) %>%  
config(displayModeBar = FALSE)
```

然鹅。。。。

## 请从配置环境开始

```
# 安装 openjdk 11
brew install openjdk@11
# 全局设置 JDK 11
sudo ln -sfn /usr/local/opt/openjdk@11/libexec/openjdk.jdk \
    /Library/Java/JavaVirtualMachines/openjdk-11.jdk
# Java 11 JDK 添加到 .zshrc
export CPPFLAGS="-I/usr/local/opt/openjdk@11/include"
export PATH="/usr/local/opt/openjdk@11/bin:$PATH"
# 配置 R 环境
sudo R CMD javareconf
# 安装 rJava 包
Rscript -e 'install.packages(c("rJava", "sparklyr"))'
```

然鹅，请从下载软件开始

手动从官网下载 Spark 软件，配置变量。。。

然鹅，请从下载软件开始

手动从官网下载 Spark 软件，配置变量。。。



警告

Spark 依赖特定版本的 Java、Hadoop，三者之间的版本应该要相融。

然鹅，请从下载软件开始

手动从官网下载 Spark 软件，配置变量。。。



### 警告

Spark 依赖特定版本的 Java、Hadoop，三者之间的版本应该要相融。

回到开头，重新下载软件，配置环境，再来三遍。

一周过去了，终于可以开始干点活了。。。

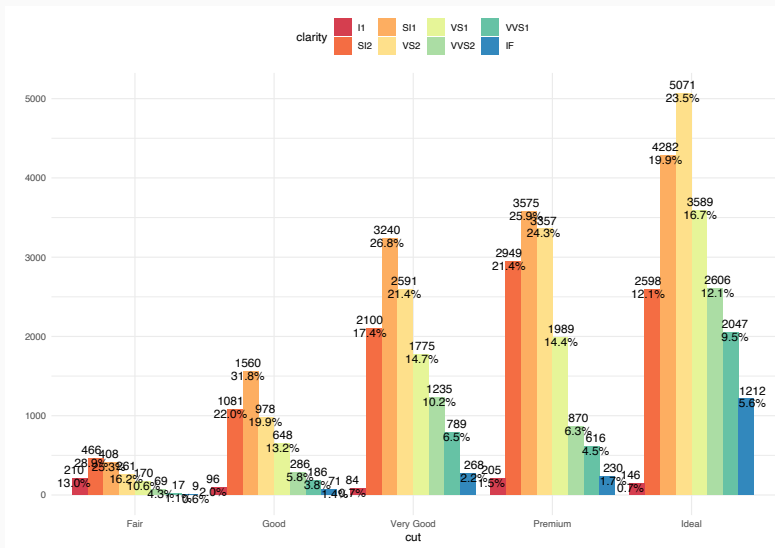


图 2: 柱形图



## 数据可视化（续）

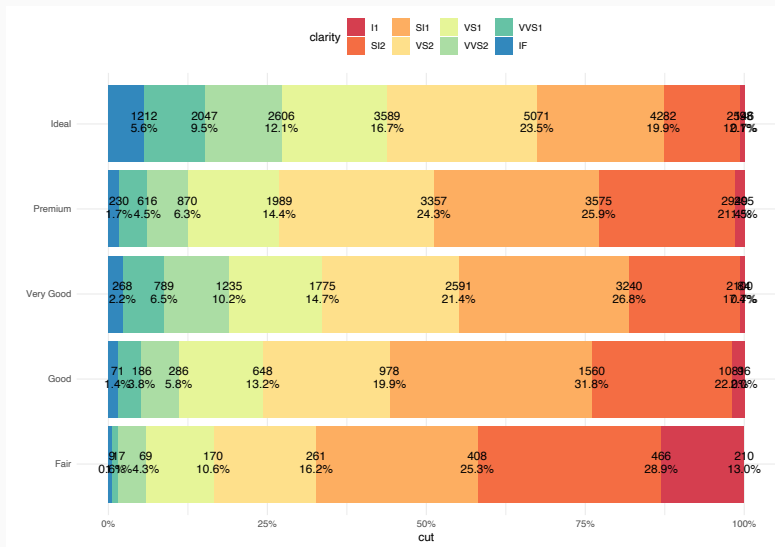


图 3: 条形图

然鹅，要这样。。。

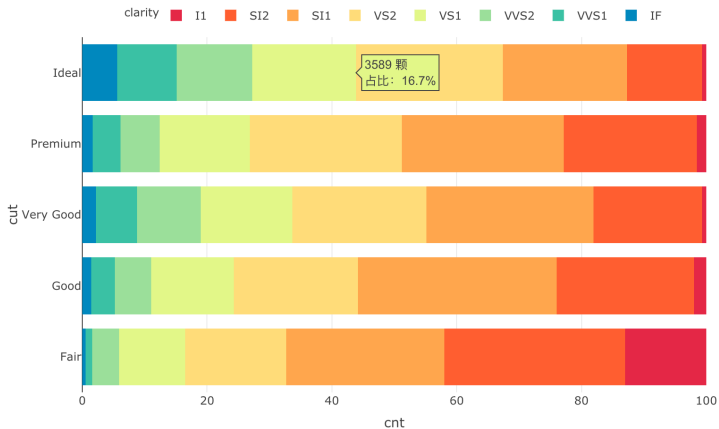
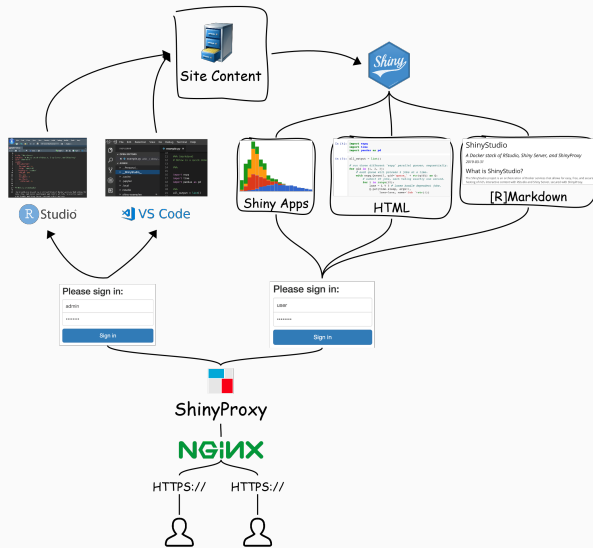


图 4: 交互条形图

## 工业实践





<sup>6</sup><https://github.com/openanalytics>

## 分层结构



Q & A

谢谢