在 R Markdown 文档中使用中文

Hongwei Shi

2021-09-08

目录

1 参考资料 **1**
1.1 参考书目 1
1.2 R blogger 主页 2

2 基本介绍 **2**
2.1 R 语言的历史 2
2.2 R 语言，你值得拥有 2
2.3 为什么要可视化? 3
2.4 主要介绍内容 4

3 基础做图 **4**
3.1 直方图 4

1 参考资料

1.1 参考书目

- R for Data Science
- ggplot2: Elegant Graphics for Data Analysis
- Advanced R
- R Packages
- 现代统计图形
- 数据科学中的 R 语言
- R 语言教程

1.2 R blogger 主页

- Hadley Wickham
- 谢益辉
- 黄湘云

2 基本介绍

2.1 R 语言的历史

The History of R (updated for 2020)

- 1992: R development begins as a research project in Auckland, NZ by Robert Gentleman and Ross Ihaka
- 1993: First binary versions of R published at Statlib
- 1995: R first distributed as open-source software, under GPL2 license
- 1997: R core group formed
- 1997: CRAN founded (by Kurt Hornik and Fritz Leisch)
- 1999: The R website, r-project.org, founded
- 1999: First in-person meeting of R Core team, at inaugural Directions in Statistical Computing conference, Vienna
- 2000: R 1.0.0 released (February 29)
- 2000: John Chambers, recipient of the 1998 ACM Software Systems Award for the S language, joins R Core
- 2001: R News founded (later to become the R Journal)
- 2003: R Foundation founded
- 2004: First UseR! conference (in Vienna)
- 2004: R 2.0.0 released
- 2009: First edition of the R Journal
- 2013: R 3.0.0 released
- 2015: R Consortium founded, with R Foundation participation
- 2016: New R logo adopted
- 2017: CRAN exceeds 10,000 published packages
- 2020: R 4.0.0 released

2.2 R 语言，你值得拥有

- R 是一个统计编程语言 (statistical programming)

- R 拥有顶尖水准的制图功能
- R 是**免费的**
- R 应用广泛，拥有丰富的库包，活跃的社区
- 统计学研究者的**重要工具**!

2019 年 8 月,国际统计学年会将考普斯总统奖(The Committee of Presidents of Statistical Societies Awards,简称 COPSS 奖，被誉为统计学的诺贝尔奖）奖颁给 tidyverse 的作者 Hadley Wickham 后，充分说明 R 语言得到了学术界的肯定和认可，未来一片光明!

Hadley Wickham's Homepage，改变了 R 语言的人!

2.3 为什么要可视化？

- 看图片，往往能比表格传达出更多的信息，一图胜千言
- 可视化，“一半是科学、一半是艺术”。要做图，更要做漂亮的图
- 但可视化只是一种手段，根据数据实际情况作展示才是重要的，并不是要追求酷炫，适合自己的才是最好的

2.4 主要介绍内容

- R base：graphic
- Advanced：ggplot2
- 交互图：plotly

3 基础做图

3.1 直方图

直方图（Histogram）是展示连续数据分布最常用的工具，它本质上是对密度函数的一种估计。

```
library(formatR)
usage(hist.default)
```

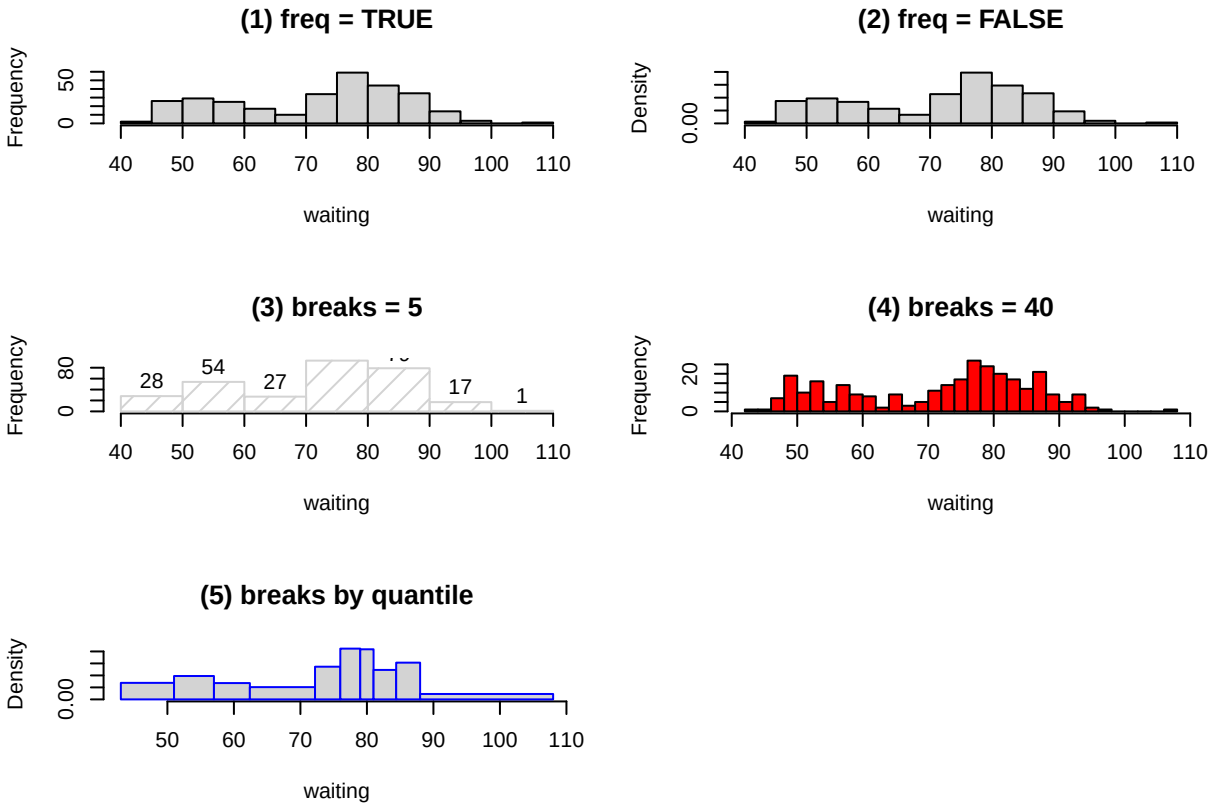
```
## ## Default S3 method:
## hist(x, breaks = "Sturges", freq = NULL, probability = !freq,
##     include.lowest = TRUE, right = TRUE, density = NULL, angle = 45,
##     col = "lightgray", border = NULL, main = paste("Histogram of", xname),
##     xlim = range(breaks), ylim = NULL, xlab = xname, ylab, axes = TRUE,
##     plot = TRUE, labels = FALSE, nclass = NULL, warn.unused = TRUE, ...)
```

- x 为欲估计分布的数值向量
- breaks 决定了计算分段区间的方法，它可以是一个向量（依次给出区间端点），或者一个数字（决定拆分为多少段），或者一个字符串（给出计算划分区间的算法名称），或者一个函数（给出划分区间个数的方法），区间的划分直接决定了直方图的形状，因此这个参数是非常关键的
- freq 和 probability 参数均取逻辑值（二者互斥），前者决定是否以频数作图，后者决定是否以概率密度作图（这种情况下矩形面积为 1）
- labels 为逻辑值，决定是否将频数的数值添加到矩形条的上方

我们以黄石国家公园喷泉数据 geyser 为例，展示了喷泉喷发间隔时间的分布情况。

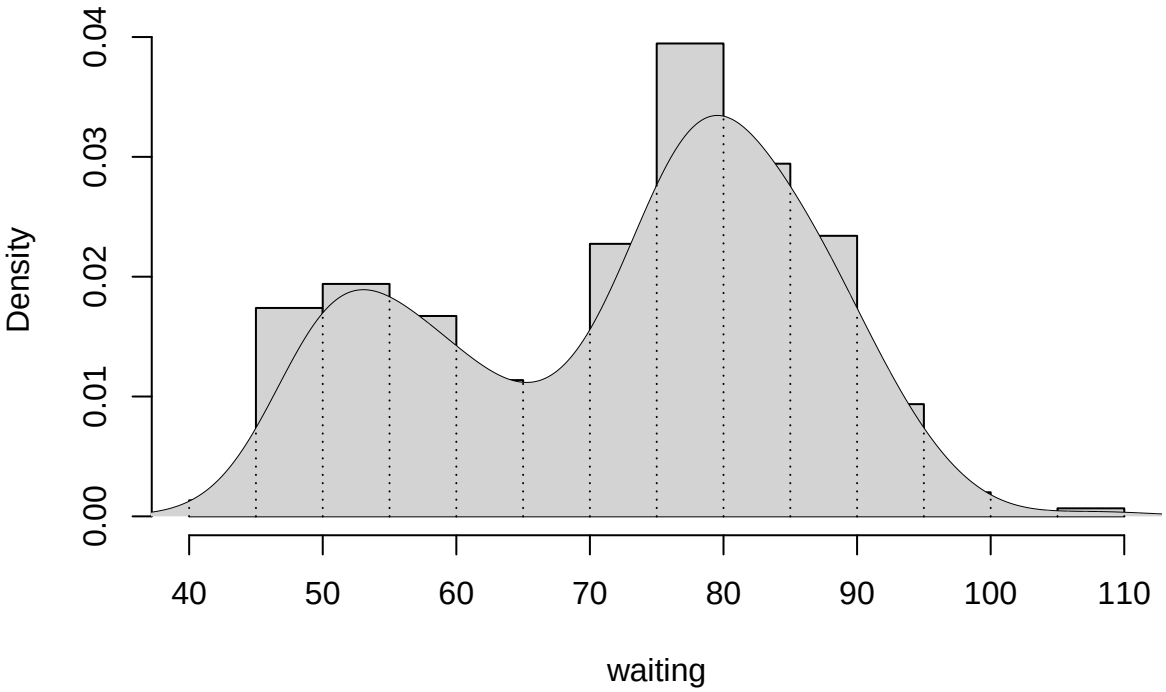
```
par(mfrow = c(3, 2))
data(geyser, package = "MASS")
hist(geyser$waiting, main = "(1) freq = TRUE", xlab = "waiting")
hist(geyser$waiting, freq = FALSE, xlab = "waiting", main = "(2) freq = FALSE")
hist(geyser$waiting, breaks = 5, density = 10, labels = TRUE, xlab = "waiting", main = "(3) breaks
hist(geyser$waiting, breaks = 40, col = "red", xlab = "waiting", main = "(4) breaks = 40")
hist(geyser$waiting,
     breaks = quantile(geyser$waiting, probs = seq(0, 1, 0.1)),
```

```
    border = "blue", xlab = "waiting", main = "(5) breaks by quantile")
```



直方图与密度曲线的结合：借助函数 density() 可以计算出数据的核密度估计

```
hist(geyser$waiting, probability = TRUE, main = "", xlab = "waiting")
d <- density(geyser$waiting)
lines(d) # 添加密度曲线
polygon(c(min(d$x), d$x, max(d$x)), c(0, d$y, 0), col = "lightgray", border = NA) # 填充颜色
yend <- c(); brk <- seq(40, 110, 5)
for (i in brk) {yend <- c(yend, d$y[which.min(abs(d$x - i))])}
segments(brk, 0, brk, yend, lty = 3) # 在点对之间画线段
```


Density
0.00
0.01
0.02
0.03
0.04
40
50
60
70
80
90
100
110
waiting