R 中的统计模拟

wang

2019-07-27

献给……

呃，爱谁谁吧

目录

表格

插图

前言

关于 R 的基础语法, 可以在网上或者书籍中学习.

Github[1]

W3Cschool[2]

这里只是总结一些统计模拟中遇到的问题, 以及实用的技巧.

致谢

这个页面的建立基于 **knitr** (Xie, 2015) 和 **bookdown** (Xie, 2019)。以下是我的 R 进程信息:

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/lib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/lib
##
## locale:
```

[1] https://github.com/yanping/r-spring-camp/blob/master/1-introduction.md

[2] https://www.w3cschool.cn/r/r_overview.html

```
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets
## [6] methods   base
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.1          plyr_1.8.4
##  [3] pillar_1.4.1        compiler_3.6.0
##  [5] RColorBrewer_1.1-2 influenceR_0.1.0
##  [7] viridis_0.5.1       tools_3.6.0
##  [9] zeallot_0.1.0       digest_0.6.19
## [11] jsonlite_1.6        viridisLite_0.3.0
## [13] gtable_0.3.0        evaluate_0.14
## [15] tibble_2.1.3        rgexf_0.15.3
## [17] pkgconfig_2.0.2     rlang_0.4.0
## [19] igraph_1.2.4.1      rstudioapi_0.10
## [21] yaml_2.2.0          xfun_0.7
## [23] gridExtra_2.3       downloader_0.4
## [25] DiagrammeR_1.0.1    dplyr_0.8.1
## [27] stringr_1.4.0       knitr_1.23
## [29] htmlwidgets_1.3     vctrs_0.2.0
## [31] hms_0.5.0           grid_3.6.0
## [33] tidyselect_0.2.5    glue_1.3.1
## [35] R6_2.4.0            Rook_1.1-1
## [37] XML_3.98-1.20       rmarkdown_1.13
## [39] bookdown_0.11       ggplot2_3.1.1
## [41] tidyr_0.8.3         purrr_0.3.2
## [43] readr_1.3.1         magrittr_1.5
## [45] scales_1.0.0        backports_1.1.4
## [47] htmltools_0.3.6     assertthat_0.2.1
## [49] colorspace_1.4-1    brew_1.0-6
## [51] stringi_1.4.3       visNetwork_2.0.7
## [53] lazyeval_0.2.2      munsell_0.5.0
## [55] crayon_1.3.4
```

作者简介

统计学学生.

主要用 R 和 tex.

第一章　常用函数及常见错误

1.1　产生数据

在进行模拟时, 我们经常会需要生成数据. 这里以正态分布为例, 说明如何产生数据. 在 R 中, 每种分布都会有以下 4 个函数:

- 概率密度函数: dnorm(x, mean = 0, sd = 1, log = FALSE)
- 累计分布函数: pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
- 分位数函数: qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
- 随机数产生: rnorm(n, mean = 0, sd = 1):

其中前 3 个函数都支持向量输入, 即计算一组取值的概率密度、累计分布、分位数. 最后一个函数常用来生成数据,n 即产生数据的个数. 如果需要生成多维正态分布, 需要调用 MASS 包中的 *mvrnorm(n = 1, mu, Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE), 其中 Sigma 是指定的协方差矩阵.

1.2　定义运算符

空间模型 (SAR) 中, 会出现对角块矩阵

$$\mathbf{W} = \begin{pmatrix} \mathbf{M} & & & \\ & \mathbf{M} & & \\ & & \ddots & \\ & & & \mathbf{M} \end{pmatrix}$$

M 作为权重矩阵, 这种形式的矩阵可以利用克罗内克积 (Kronecker product, 符号为 $\otimes$) 在 R 中很方便的产生, 命令是%x%. 可以写成

$$\mathbf{W} = \mathbf{I} \otimes \mathbf{M}.$$

```
diag(3)%x%matrix(1:6,2,3)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    3    5    0    0    0    0    0    0
## [2,]    2    4    6    0    0    0    0    0    0
## [3,]    0    0    0    1    3    5    0    0    0
## [4,]    0    0    0    2    4    6    0    0    0
## [5,]    0    0    0    0    0    0    1    3    5
## [6,]    0    0    0    0    0    0    2    4    6
```

这是借助自定义运算符实现的. 自定义运算符是一种特殊的函数, 当参数只有两个变量时, 可以进行定义. 用法如下:

```
'%myop%'<-function(a,b){a^b+b^a}
2%myop%3
```

```
## [1] 17
```

利用自定义运算符, 可以实现很方便的功能.R 中矩阵乘法 (%*%)、Kronecker 乘积 (%x%) 都是这样实现的. 另外还有整除 (%/%) 和取余 (%%)

```
9%/%4
```

```
## [1] 2
```

```
13%%3
```

```
## [1] 1
```

1.3 自动纠错

当我们输入的命令不规范时,R 会自动纠正, 以保证程序正常运行.

比如看下面的例子:

```
1:4 - 1:2
```

```
## [1] 0 0 2 2
```

```
1:5-1:2
```

```
## Warning in 1:5 - 1:2: longer object length is not a
## multiple of shorter object length
```

```
## [1] 0 0 2 2 4
```

当运算的向量长度不一致时,R 会自动重复短的向量, 使之长度与另外的向量长度相同进行运算. 但是当长度是整数倍当时候, 不会有任何提示.

再看下面当例子:

```
matrix(1:9,3,3)*(1:3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    4   10   16
## [3,]    9   18   27
```

本来想计算矩阵与向量的乘积, 但是错误使用了 *, 未使用矩阵乘法%*%,R 也可以计算返回一个矩阵, 不会有 warning.

当条件为向量是, 只会判断第一个值:

```
if( 1 <= 1:3) print("真")  else print("假")
```

```
## Warning in if (1 <= 1:3) print("真") else print("假"):
```

```
## the condition has length > 1 and only the first element
## will be used
```

```
## [1] "真"
```

如果条件为向量, 应该使用 all 或者 any 函数:

```
all(1 <= 1:3)
```

```
## [1] TRUE
```

```
all(2 <= 1:3)
```

```
## [1] FALSE
```

所有都真的时候返回 TRUE.

```
any(4 <= 1:3)
```

```
## [1] FALSE
```

```
any(2 <= 1:3)
```

```
## [1] TRUE
```

只要有一个为真就返回 TRUE.

1.4 predict 函数

当我们拟合好一个模型时, 下一步要做的就是评价模型好坏或者对新数据预测. 这都需要将新的输入值带入模型中计算, 得到预测值. 区别只是有没有真值对比. 对于简单模型, 我们当然可以直接提取系数, 自己计算预测值, 但是当模型复杂时 (比如时间序列模型), 就不太容易操作.

R 借助泛型函数[1], 编写模型都会提供 summary、predict、plot 等函数方便调用. 但是在学习过程中发现经常会不小心错误使用, 特此单独说明一下. 下面以线性模型为例, 先看正确的使用方法:

```
n=100;p=3;beta=c(1,2,3);
X = matrix(rnorm(n*p),n,p)
Y = X%*%beta + rnorm(n)
trainlist = sample(1:n,70)
regData = data.frame(Y,X)
fitmodel = lm(Y~.,data=regData[trainlist,])
pe = predict(fitmodel,newdata=regData[-trainlist,])
sum((pe-regData[-trainlist,1])^2)/length(pe)
```

```
## [1] 0.6623
```

关键点:

所有数据存在一个数据框中 通过下标控制训练集和测试集的数据

错误程序 1:

```
n=100;p=3;beta=c(1,2,3);
X = matrix(rnorm(n*p),n,p)
Y = X%*%beta + rnorm(n)
fitmodel = lm(Y~X)
X2 = matrix(rnorm(n*p),n,p)
Y2 = X2%*%beta + rnorm(n)
#predict(fitmodel,newdata = X2)数据格式错误,不能执行
pe = predict(fitmodel,newdata = data.frame(X2))
sum((pe-Y2)^2)/length(Y2)
```

```
## [1] 26.06
```

这个程序能明显看出问题, 误差不应该这么大, 但是程序没有任何 warning.

[1] 泛型函数简单讲就是根据输入数据类型, 会自动匹配对应的计算方法. 比如 plot 函数, 传入 1 维向量、矩阵, 程序都会正确画图. 泛型函数的声明需要使用 ".", 在自己编写函数时, 最好不要用 ".".

错误程序 2:

```
n=100;p=3;beta=c(1,2,3);
X = matrix(rnorm(n*p),n,p)
Y = X%*%beta + rnorm(n)
fitmodel = lm(Y~X)
X2 = matrix(rnorm(0.2*n*p),0.2*n,p)
Y2 = X2%*%beta + rnorm(0.2*n)
pe = predict(fitmodel,newdata = data.frame(X2))
```

```
## Warning: 'newdata' had 20 rows but variables found have
## 100 rows
```

```
length(pe)
```

```
## [1] 100
```

修改测试数据的条数, 使之与训练集数据量不同, 可以发现 warning. 提示我们数据行数不一样. 并且我们测试集合 X2 是 20 行, 但是预测值 pe 返回的是 100 个值. 问题在于 predict 函数使用不正确.

我们用 all 指令查看:

```
all(predict(fitmodel)==predict(fitmodel,newdata = data.frame(X2)))
```

```
## Warning: 'newdata' had 20 rows but variables found have
## 100 rows
```

```
## [1] TRUE
```

查看函数说明

```
?predict.lm
```

参数	说明
object	Object of class inheriting from “lm”

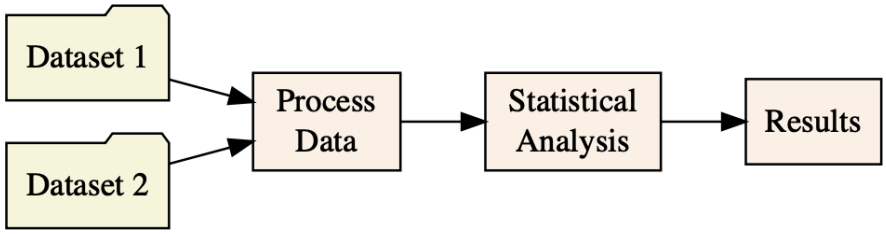
参数	说明
newdata	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.

newdata 不是必要的参数, 当缺失时候会使用拟合模型的数据.

通过以上例子, 想说明当程序的结果和预期不一致时. 当然可能是我们的方法不对, 但是也有可能是调用函数的方式出了问题. 比如线性规划求解的 lp 函数, 默认在正半轴求解[2].

1.5 保存结果

1.6 结果输出



[2]因为任何一个实数可以分解成两个非负数的差, 将搜索范围限定到正半轴, 求解程序更好实现.

第二章　常见错误

自动调整

第三章　参数估计

3.1　M 估计

3.2　Z 估计

第四章　假设检验

第五章　进阶技巧

5.1　apply 函数族

5.2　并行

5.3　Rcpp

附录 A　余音绕梁

呐，到这里朕的书差不多写完了，但还有几句话要交待，所以开个附录，再啰嗦几句，各位客官稍安勿躁、扶稳坐好。

参考文献

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2019). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.11.

索引