**7.9 R 语言应用**

**# 陈文贤 着 《大话统计学》 清华大学出版社 2022年**

if(!require(DescTools)){install.packages("DescTools")} ; library(DescTools)

if(!require(dplyr)){install.packages("dplyr")} ; library(dplyr)

if(!require(boot)){install.packages("boot")} ; library(boot)

if(!require(profvis)){install.packages("profvis")} ; library(profvis)

x = read.csv("C:/大话统计学 网络资源/StatData/Chap7\_2.csv",header=TRUE)

# 读入 Chap7\_2.csv

x1 <- x[,1] ; n <- length(x1) ; n # 数据数目

t.test(x1, conf.levele=0.95)$conf.int # 区间估计

t.test(x1, conf.levele=0.95) # t 检验

x <- c(3,1.8,2.5,2.1,2.7,1.9,1.5,1.7,2,1.6)

bootmean <- function(x,ind) mean(x[ind])

boot1 <- boot(x, bootmean,R=999,stype="i",sim="ordinary") # 自助法置信区间

boot.ci(boot1, conf = 0.95, type = c("norm", "basic", "perc", "bca"))

Varb <- function(x,ind) var(x[ind])

bootVar <- boot(x, Varb, R=999, stype="i", sim="ordinary") # 自助法方差置信区间

boot.ci(bootVar, conf = 0.95, type = c("norm", "basic", "perc", "bca"))

Md <- function(x,ind) median(x[ind])

bootMd <- boot(x, Md, R=999, stype="i", sim="ordinary") #自助法中位数置信区间

boot.ci(bootMd, conf = 0.95, type = c("norm", "basic", "perc", "bca"))

x <- c(3,1.8,2.5,2.1,2.7,1.9,1.5,1.7,2,1.6)

m1 <- qbinom(0.025, length(x),0.5) ; m2 <- qbinom(1-0.025, length(x),0.5)

median.ci <- c(sort(x[m1]),sort(x[m2+1])) ; median.ci #中位数置信区间

if(!require(epitools)) {install.packages("epitools")} ; library(epitools)

binom.approx(141, 226)

binom.exact(x=141, n=226, conf.level = 0.95)

binom.test(x=141, n=226, conf.level = 0.95)$conf

binom.test(x=141, n=226)$conf # 比例置信区间

x <- c(14.816, 14.863, 14.814, 14.998, 14.965, 14.824, 14.884, 14.838, 14.916,

15.021, 14.874, 14.856, 14.860, 14.772, 14.980, 14.919)

VarCI(x, conf.level=0.9) # 自助法方差置信区间

sqrt(VarCI(x, conf.level=0.9)) # 自助法方差置信区间

# 模拟 均值(平均数) 置信区间

if(!require(gridExtra)){install.packages("gridExtra")} ; library(gridExtra)

if(!require(ggplot2)){install.packages("ggplot2")} ; library(ggplot2)

mu = 65 ; sigma = 7 ; n=50 ; alpha = 0.05

x0= c(60, 70) ; y0=c(0, 105)

plot(x0, y0, type = "n", ylab = " 置信区间 ", xlab = " x ")

abline(v=mu, lty=2)

for (i in 1:100) { x =rnorm(n, mu, sigma)

width <- qt(1-alpha/2, n-1)\*sd(x) / sqrt(n)

c1 <- mean(x) - width

c2 <- mean(x) + width

if (mu >= c1 && mu <= c2) { lty=2 ; lwd =1 }

else { lty=1 ; lwd =2 }

lines (c(c1, c2), c(i, i) , lty=lty , lwd =lwd , col = lwd) }

pause(10) # 等候 10 秒钟

# 模拟 总体比例 置信区间

par(mfrow=c(1,1))

m <- 50; n <- 20 # m 组样本数据(m个置信区间) , n = 每组数据样本量

pi <- .6 ; p = rbinom(m,n,pi)/n # pi = 总体比例π , p = 样本比例(m 组)

alpha <- 0.10 ; za <- qnorm(1-alpha/2) # alpha = 置信水平α , za = z(α/2)

SE = sqrt(p\*(1-p)/n) # 总体比例的标准误差

sum(p - za\*SE < pi & pi < p + za\* SE) # m个置信区间中有总体比例pi(π)的数目

matplot(rbind(p - za\*SE, p + za\*SE), rbind(1:m,1:m), type="l",

lty=1, xlab= "总体比例 pi (π)的置信区间", ylab= "50个置信区间" ) # 如下图

abline(v=pi) # 总体比例 pi=0.6

pause(10) # 等候 10 秒钟

# 模拟 置信区间 包::函数 PASWR:: CIsim

if(!require(PASWR)){install.packages("PASWR")} ; library(PASWR)

# PASWR2:: cisim

CIsim(samples = 100, n = 30, sigma = 10, conf.level = 0.90)

pause(10) # 等候 10 秒钟

CIsim(100, 30, 100, 10)

pause(10) # 等候 10 秒钟

CIsim(100, 50, .5, type="Pi", conf.level=.90)

pause(10) # 等候 10 秒钟

CIsim(100, 30, 100, 10, type="Var")

pause(10) # 等候 10 秒钟

[CIsim](https://rdrr.io/cran/BSDA/man/CIsim.html)(samples=100, n=30, [sigma](https://rdrr.io/r/stats/sigma.html)=1, conf.level=0.95, [type](https://rdrr.io/r/base/typeof.html)="Mean")