**2.10 R 语言应用**

**# 陈文贤 着 《大话统计学》 清华大学出版社 2022年**

# 以下 R 代码的 程辑包 是用R版本4.0.5 来建造的

# R版本4.5.0 可能 无法打开

# R例2.1

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

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

x = read.csv("C:/大话统计学 网络资源/StatData/Chap2\_1.csv",header=F) ; str(x)

# 读入 Chap2\_1.csv

x1 <- x[,1] ; str(x1) # x1 是数据框 x 的第1列数据 (x1是向量数值格式)

summary(x) ; summary(x1) # 摘要统计

n <- length(x1) ; n # 数据数目

sum(x1) # 总和

mean(x1) # 平均数

mean(x1, trim=0.1) # 截尾平均

median(x1) # 中位数

prod(x1)^(1/length(x1)) # 几何平均

x2 <- 1/x1 ; length(x2)/sum(x2) # 调和平均

1 / mean(x2) # 调和平均

x3<- c(80, 65, 90, 75) ; wt <- c(3, 4, 5, 2) # 学生成绩与学分数

xm <- weighted.mean(x3, wt) ; xm # 学生加权平均成绩 GPA

xmd <- weightedMedian(x3, wt) ; xmd

weightedMean(x3, wt)

weightedVar(x3, wt)

###

x4<- c(25, 35, 45, 55, 65, 75, 85, 95) ; wt <- c(1, 5, 8, 4, 1, 2, 5, 4) # 2.9.3节 分组数据

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

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

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

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

matrixStats::weightedMean(x4, wt) # 加权平均数

stats::weighted.mean(x4, wt) # 加权平均数

matrixStats::weightedMedian(x4, wt) #加权中位数

spatstat.geom::weighted.median(x4, wt) #加权中位数

matrixStats::weightedVar(x4, wt) # 加权方差

spatstat.geom::weighted.var(x4, wt) # 加权方差

spatstat.geom::weighted.quantile(x4, wt) # 加权百分位数

modi::weighted.quantile(x4, wt, prob=0.5, plot=TRUE) # 加权中位数

modi::weighted.var(x4, wt) # 加权方差

# 累积频数图

# 加权中位数, 方差, 百分位数 在不同的 R 包和函数 , 因公式不同, 结果不同.

pause(10) # 等候 10 秒钟

###

tab <- table(x1) ; names(which.max(tab)) ; names(tab)[max(tab)==tab] # 众数

mode <- function(x) {uniqx <- unique(x)

uniqx[which.max(tabulate(match(x, uniqx)))] } ; mode(x1) # 众数

tab <- table(x1) ; sort(tab, decreasing = TRUE) # 频数表

quantile(x1, probs=c(0.25, 0.75), type=6) # 下四分位数,上四分位数 (计算个数)

quantile(x1, probs=c(0.25, 0.75), type=7) # 下四分位数,上四分位数 (计算间距)

quantile(x1, probs=c(0.05, 0.2, 0.95), type=6) # 5% ,20% ,95%百分位数 (计算个数)

quantile(x1, probs=c(0.05, 0.2, 0.95), type=7) # 5% ,20% ,95%百分位数 (计算间距)

IQR(x1) ; IQR(x1, type=6) # 四分位差或四分位距

percentile.rank <- function(vec, value) {num <- length(sort(vec)[vec < value])

den <- length(vec)

paste(round(num/den,3)\*100, "%") } # 定义百分位排序函数

value <- 60 ; percentile.rank(x1, value)

value <- 50 ; percentile.rank(x1, value)

range(x1) ; diff(range(x1)) #极差或全距(range) *R*

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

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

x = read.csv("C:/大话统计学 网络资源/StatData/Chap2\_1.csv",header=F)

# 读入 Chap2\_1.csv

x1 <- x[,1]

# percent\_rank(x1) # x1 的百分位排序

###

var(x1) # 样本方差

sd(x1) # 样本标准差

var.pop = function(x) var(x)\*(length(x)-1)/length(x) ; var.pop(x1) # 总体方差

sd.pop = function(x) sqrt(var.pop(x)) ; sd.pop(x1) # 总体标准差

scale(x1) # 数据标准化

standard\_error <- function(x) sd(x) / sqrt(length(x)) # 创建标准误差函数

standard\_error(x1) # 标准误差函数

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

std.error(x1) # 标准误差函数

MeanAD(x1, center=mean) #  平均绝对差 mean absolute deviation from mean

MeanAD(x1, center=median) # 平均离中差mean absolute deviation from median

# 中位数绝对差 R::stats::mad() # median of the absolute deviations from the median

mad(x1) # mad(

mad(x1, constant=1) # mad(

if(!require(e1071)){install.packages("e1071")} ; library(e1071)

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

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

n <- length(x1) ; n # 数据数目

moment(x1, order=3, center=TRUE) #三阶中心距

moment(x1, order=4, center=TRUE) #四阶中心距

skew.p <- function(x) mean((x-mean(x))^3)/sd.pop(x)^3 ; skew.p(x1) # 总体偏度

skew <- function(x) mean((x-mean(x))^3)/sd(x)^3 \* (n/(n-1)) ; skew(x1) # 样本偏度

kurt.p <- function(x) mean((x-mean(x))^4)/sd.pop(x)^4 ; kurt.p(x1) # 总体峰度

kurt <- function(x) mean((x-mean(x))^4)/sd(x)^4\* (n/(n-1)) ; kurt(x1) # 样本峰度

EnvStats::skewness(x1) ; EnvStats::kurtosis(x1) # kurtosis(x1) = kurt.p(x1) -3

TSA::skewness(x1) ; TSA::kurtosis(x1) # skewness = skew.p ; kurtosis = kurt.p -3

diff(x1) # 差分 , 请见第14.11节

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

describe(x1) # psych 包的描述统计

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

stat.desc(x1, basic=TRUE, desc=TRUE, norm= TRUE, p=0.95) # pastecs 包的描述统计

# R 语言绘图

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

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

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

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

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

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

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

x = read.csv("C:/大话统计学 网络资源/StatData/Chap2\_1.csv",header=F)

# 读入 Chap2\_1.csv

x1 <- x[,1] # x1 是数据框 x 的第1列数据 (x1是向量数值格式)

breaks <- seq(from=min(x1), to=max(x1), length=8) # 分成 8-1 = 7 组

freq <- cut(x1, breaks=breaks, right=TRUE, include.lowest=TRUE)

table(freq) ; hist(x1, breaks=breaks, col='pink') # 频数分布表 与 直方图

pause(10) # 等候 10 秒钟

hist(x1, freq=FALSE, col='light green') ; lines(density(x1), lwd=3, col='blue')

pause(10) # 等候 10 秒钟

brk <- c(20,35,40,50,65,70,80,90,100)

hist(x1,breaks=brk, col='yellow') # 不同组宽 直方图

pause(10) # 等候 10 秒钟

n <- length(x1) ; plot(sort(x1),(1:n)/n,type="s",ylim=c(0,1) , col='purple') # 累积频率图

pause(10) # 等候 10 秒钟

boxplot(x1, col="yellow", main=paste("例题2.1 箱线图")) # 箱线图

pause(10) # 等候 10 秒钟

plot(ecdf(x1), main=paste("例题2.1 累积概率函数"), col.hor='#3971FF',

col.points='#3971FF')

pause(10) # 等候 10 秒钟

bound <- hist(x1, right=TRUE, plot=FALSE )$breaks

plot(bound, ecdf(x1)(bound), type="l", main = "例题2.1 累积频率图", ylab= "频率",

xlab= "分数", col="red", lwd=3) #累积频率图

pause(10) # 等候 10 秒钟

stem.leaf(x1, style="bare") # 茎叶图

class <- hist(x1, right=TRUE, freq=FALSE, col="green")

pause(10) # 等候 10 秒钟

class <- hist(x1, right=F, freq=F, col="yellow", main="例题2.1 直方图", xlab="人数")

pause(10) # 等候 10 秒钟

class <- hist(x1, right=TRUE, freq=F,col="yellow", main="例题2.1 多边形图", xlab="人数")

middles <- class$mid ; mlon <- length(middles) ; densities <- class$density

pause(10) # 等候 10 秒钟

segments(middles[1:mlon-1],densities[1:mlon-1], middles[2:mlon],densities[2:mlon],

col=rgb(0.4196078, 0.4196078, 0.1372549,0.9), lwd=3, main=paste("例题2.1 多边形图"))

pause(10) # 等候 10 秒钟

x2 = read.csv("C:/大话统计学 网络资源/StatData/Chap2\_4\_1.csv",header=TRUE)

# 读入 Chap2\_4\_1.csv

table(x2) ; col2 = c("red", "yellow", "blue")

barplot(table(x2), bes=TRUE, col=col2, legend.text = T, args.legend = list(x = "top",

inset = c(- 0.15, 0)), main=paste("例题2.4 条形图")) # 両个定类变量条形图

col1 = c("red", "yellow", "blue", "sandybrown", "olivedrab", "purple", "green", "orange")

pause(10) # 等候 10 秒钟

x = read.csv("C:/大话统计学 网络资源/StatData/Chap2\_6.csv",header=TRUE)

# 读入 Chap2\_6.csv

barplot(table(x$Rank),col=col1, main=paste("例题2.6 条形图"))

pause(10) # 等候 10 秒钟

pareto.chart(table(x$Rank), main= "例题2.6 帕累托图", ylab = "频率", ylab2 =

"累积百分比", cumperc = seq(0, 100, by = 25), col = col1, plot = TRUE) # 帕累托图

col<-brewer.pal(8,"Pastel2") ;

pause(10) # 等候 10 秒钟

pie(table(x$Rank),col=col1, main=paste("例题2.6 饼图") )

pause(10) # 等候 10 秒钟

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

slices <- c(10, 12,4, 16, 8) ; lbls <- c("US", "UK", "Australia", "Germany", "France")

pie3D(slices, labels=lbls,explode=0.1, main="3D 饼图 Pie Chart ")

qqnorm(x1) # QQ 正态分布图

qqline(x1,lty=2)