装订线 答题时不要超过此线

中 国 科 学 技 术 大 学

2019-2020 学年第一学期考试试卷

考试时间 14:00-16:00

考试科目: 实用统计软件　　得分:_____

学生所在系:_____　姓名:_____　学号:_____

一、（24 分，每小题 2 分）填空题。请写出以下 R 程序的输出。

(1) 8 %/% 5

(2) seq(from = 5, to = 11, by = 5)

(3) (11:20)[5:6]

(4) (1:10)[-5:4 >0]

(5) !(1 > 2) | 3 == 4 & 5 >= 6

(6) x <- numeric(5); x[3:5] <- 7:8; x

(7) cat("aa = ", round(3.1416, 2), "\n", sep="**")

(8) paste(c("stat", "prob"), 1:3, collapse = "; ")

(9) x <- 10; f <- function(x, y = x * 2) { x + y }; f(2)

(10) 0.3 - 0.2 == 0.1

(11) nchar("实用统计软件\t")

(l2) x <- -2:1; ifelse(x > 0, x, -x)

二、（24 分，每小题 3 分）选择题。

(1) 有以下的 R 程序：

```
m1 <- matrix(1:4, nrow = 2)
m2 <- matrix(1, nrow = 2, ncol = 3)
m2 <- as.data.frame(m2)
m1%*%m2
```

程序执行后的输出结果是（　）

(A)

	[,1]	[,2]	[,3]
[1,]	4	4	4
[2,]	6	6	6

(B)

	[,1]	[,2]	[,3]
[1,]	3	3	3
[2,]	7	7	7

(C) Error in m1 %*% m2 : non-conformable arguments

(D) Error in m1 %*% m2 : requires numeric/complex matrix/vector arguments

(2) 有以下的 R 程序：

```
all.cases <- c()
case.1 <- c(1610,4)
rbind(all.cases,case.1)
case.2 <- c(1877,2)
rbind(all.cases,case.2)
```

```
all.cases
```

程序执行后的输出结果是（　　）

(A)
```
     [,1]  [,2]
[1,] 1610  1877
[2,]    4     2
```
(B) [1] 1877 2　(C) NULL　(D) [1] 1610 4

(3) 有以下的 R 程序：

```
x <- 7
y <- c("A")
adder <- function(y) { x<- x+y; return(x) }
adder(1)
y
```

程序执行后的输出结果是（　　）

(A) 8　8　　(B) 8　"A"　　(C) "A"　"A"

(D) 由于字符不能用在数值计算中，因此结果出错。

(4) 假设 m 和 n 是相同大小的方阵，考虑以下的程序：

```
r <- matrix(rep(0,times=ncol(m)*nrow(m)),nrow=ncol(m),ncol=ncol(m))
for (i in 1:ncol(m)) {
    for (j in 1:ncol(m)) {
        for (k in 1:ncol(n)) {
            r[i,j] <- r[i,j] + m[i,k]*n[k,j]
        }
    }
}
```

那么以下命令与上述程序等价的是（　　）

(A) r <- m %*% n　(B) r <- colSums(m*n)　(C) r <- n*m　(D) 以上都不是

(5) 在 while（!x）语句中的!x 与下面条件表达式等价的是（　　）

(A) x!=0　　(B) x==1　　(C) x!=1　　(D) x==0

(6) 请问以下两个程序等价吗？

甲：
```
s <- 2
curve(function(x) { return(log((0.5/pi)*exp(-0.5*x^2/s^2))) }, from=-1,to=1)
```
乙：
```
s <- 2
curve(log((0.5/pi)*exp(-0.5*x^2/s^2)), from=-1,to=1)
```

请从以下选项中选择正确的结果（　　）

(A) 是的，它们都画出一条抛物线；　(B) 是的，它们都画出一条钟形曲线；

(C) 不是，他们两个都产生错误；　(D) 不是，它们其中一个产生错误

(7) 有以下函数 foo(x,y)的定义

```
foo <- function(x, y) {
#####     空  格     #####
    for(i in 1:length(x)) {
        for(j in 1:length(y)) {
            result[i, j] <- g(x[i], y[j])
        }
    }
return(result)
```

```
}
```

请把空格补充完整（　　）

(A) result <- matrix()

(B) result <- matrix(nrow = length(x), ncol = length(y))

(C) result <- matrix(nrow = length(y), ncol = length(x))

(D) 无需补充

(8) 以下的 R 程序用来展示用置换检验来计算 X 和 Y 的 t 检验的 P 值：

```
B <- 200
m <- length(X)
n <- length(Y)
Z <- c(X,Y)
t0 <- t.test(X,Y)$statistic
reps <- numeric(B)
for(i in 1:B){
    #####      空  格    #####
    x1 <- Z[k]
    y1 <- Z[-k]
    reps[i] <- t.test(x1,y1)$statistic
}
p <- mean(c(t0, reps)>=t0)   # P 值
```

请把空格补充完整（　　）

(A) k <- sample(1:(m+n), size = m, replace = TRUE)

(B) k <- sample(1:(m+n), size = m, replace = FALSE)

(C) k <- sample(1:(m+n), size = B, replace = TRUE)

(D) k <- sample(1:(m+n), size = B, replace = FALSE

三、（12 分）问答题。

(1) （3 分）请列出一个 R 包在它的生命周期中的五个阶段。

(2) （3 分）请说明 require() 和 library()的差别。

(3) （3 分）请说明在 DESCRIPTION 文档中 Imports 和 Suggests 的差别。

(4) （3 分）请描述如何实现 5 折交叉验证。

四、（6 分）编写一个 R 函数 locmax()，实现在一组数中找到所有的局部最大值，其中局部最大值的定义为大于或等于其左、右两个数值的数。而且，第一个和最后一个数不能作为局部最大值，即使它们比邻近的数更大。比如说有数组 x 如下

```
x <- c(68, 55, 88, 41, 39, 30, 52, 43, 59, 77, 77, 57, 42, 31, 19)
```

你所编写的函数应该输出如下结果：

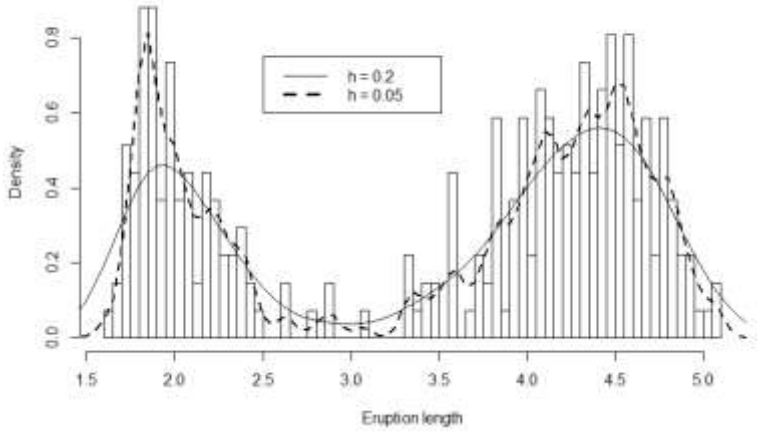
```
> locmax(x)
[1] 88 52 77 77
```

函数的要求：输入参数为对任意长度大于 2 的数值向量，尽量避免用循环。

五、（14 分）对于随机样本$\mathrm{Z} = (\mathrm{Z}_1, \ldots, \mathrm{Z}_\mathrm{n})^\mathsf{T}$，其核密度估计由以下式子给出

$$\mathrm{f}(\mathrm{x}; \mathrm{Z}, \mathrm{h}) = \frac{1}{n}\sum_{i=1}^{n} \phi(x; Z_i, h)$$

其中$\sum_{i=1}^{n}\phi(x;Z_i,h)$表示均值为$Z_i$，标准差为$h$的正态分布密度函数。下图展示了数据集 faithful 中的变量 eruptions 的直方图和两个核密度估计。



(1) （7 分）编写一个 R 函数 kde(x, Z, h)实现核密度估计，其中向量 Z 为一组已知样本，h 为窗宽，x 为待估核密度估计的取值点。尽可能地少用循环。

(2) （7 分）请在 R 函数 kde(x, Z, h)的基础上编写 R 程序重现上图，注意相应的标记和标签要尽可能一样。

六、（20 分）假设我们有一个函数 gamma.mle，输入为数据向量，功能为对该数据向量在 gamma 分布的假设下进行极大似然估计，返回包括参数的极大似然估计和取得的极大似然估计值。基于 gamma.mle，我们有如下函数的定义：

```
gamma_ml_dist <- function(x, B = 100){                                  # 1
    mle.fit <- gama.mle(x)                                              # 2
    mle.ll <- mle.fit$loglik                                            # 3
    params <- mle.fit$params                                            # 4
    boot.logliks <- vector(length = B)                                  # 5
    for(b in 1:B){                                                      # 6
        random.draws <- vector(length = length(x))                      # 7
        for(i in 1:length(x)){                                          # 8
            random.draws[i] <- rgamma(1, shape = params[1], scale = params[2])   # 9
        }                                                               #10
        boot.logliks[b] <- gamma.mle(random.draws)$params               #11
    }                                                                   #12
    P <- mean(boot.logliks <= mle.ll)                                   #13
    return(p)                                                           #14
}
```

(1) （4 分）请解释上述程序是用来做什么的？

(2) （6 分）上述程序有一个 bug，请标记是哪一行或哪几行引起的错误，并说明为什么是错误的，最后修正使其能正确运行。

(3) （4 分）请将上述程序中的内循环用一行代码替代，即向量化运算。

(4) （6 分）请将上述程序中的两重循环用最多两行代码替代。