实用统计软件 Homework 1

邵浩然 PB21151801

目录

1 数据结构 **1**
1.1 数值型数据 1
1.2 字符型数据 2
1.3 日期时间类型数据 3
1.4 因子类型数据 4

2 向量, 矩阵和数组 **4**
2.1 向量 4
2.2 矩阵和数组 6

1 数据结构

1.1 数值型数据

1. 判断 `1L` 的数据类型，并解释结果。

```
typeof(1L)
```

```
## [1] "integer"
```

`1L` 是整数类型，因为 `L` 表示整数。

2. 计算 $\log_2 \sin(\pi/3)$, 向下取整，判断数据类型，并将最终结果转换为字符类型。

```
floor(log2(sin(pi/3)))
```

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

为整数类型；

```
as.character(floor(log2(sin(pi/3))))
```

```
## [1] "-1"
```

为转化后的字符类型。

3. 分别使用 ==, all.equal() 和 identical() 判断 $\sqrt{3}, tan(\frac{\pi}{3})$ 两者是否相等，并解释结果。

```
sqrt(3) == tan(pi/3)
```

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

```
all.equal(sqrt(3), tan(pi/3))
```

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

```
identical(sqrt(3), tan(pi/3))
```

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

== 和 identical() 用于判断两个数是否完全相等，all.equal() 则允许两个数之间存在一定误差. 由于计算机的精度问题，$\sqrt{3}$ 和 $tan(\frac{\pi}{3})$ 并不完全相等，因此 == 和 identical() 返回 FALSE，而 all.equal() 返回 TRUE。

4. 找出 1:100 中既能被 2 整除，也能被 3 整除的数。

```
x <- 1:100
x[x %% 2 == 0 & x %% 3 == 0]
```

```
##  [1]  6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96
```

1.2 字符型数据

1. 请把字符串 University of Science and Technology of China 中的'a' 到'f' 的所有字母删除。

```
x <- "University of Science and Technology of China"
gsub("[a-f]", "", x)
```

```
## [1] "Univrsity o Sin n Thnology o Chin"
```

2. 计算字符串中国科学技术大学中的实用统计软件课程 appstat2024 的长度，并提取字串实用统计软件。

```
y <- " 中国科学技术大学中的实用统计软件课程 appstat2024"
nchar(y)
```

```
## [1] 29
```

```
substr(y, 10, 18)
```

```
## [1] "的实用统计软件课程"
```

3. 运行以下命令，并解释出现的结果。如果出错，请修正。

```
c(1, 1, 2, 3, "1") + 2000
```

c(1, 1, 2, 3, "1") 是一个字符型向量，而 2000 是数值型，无法进行运算。修正如下:

```
c(1, 1, 2, 3, as.numeric("1")) + 2000
```

```
## [1] 2001 2001 2002 2003 2001
```

4. 有字符串 x 定义如下：

```
x <- '1A3c5d'
```

请把 x 按照字母进行拆分，并将结果转化成数值型变量。

```
as.numeric(strsplit(x, "[A-Za-z]")[[1]])
```

```
## [1] 1 3 5
```

1.3 日期时间类型数据

定义出生日期变量 date_bir 和发病日期变量 date_dis 如下：

```
date_bir <- "1961/3/1"
date_dis <- "2022 年 1 月 1 日"
```

1. 把上述变量分别转换成 Date 和 POSIXct 日期型。

```
as.Date(date_bir, format = "%Y/%m/%d")
```

```
## [1] "1961-03-01"
```

```
as.POSIXct(date_dis, format = "%Y 年%m 月%d 日")
```

```
## [1] "2022-01-01 CST"
```

2. 输出出生年和月份。

```
format(as.Date(date_bir, format = "%Y/%m/%d"), "%Y")
```

```
## [1] "1961"
```

```
format(as.Date(date_bir, format = "%Y/%m/%d"), "%m")
```

```
## [1] "03"
```

3. 计算发病时的年龄，以周岁论，即过了生日才算；并通过 sink() 函数将发病年龄保存在 test.txt 文件中。

```
age <- as.numeric(format(as.Date(date_dis, format = "%Y 年%m 月%d 日"), "%Y")) - as.numeric(format(as
if (format(as.Date(date_dis, format = "%Y 年%m 月%d 日"), "%m%d") < format(as.Date(date_bir, format =
  age <- age - 1
}
sink("test.txt")
cat(age)
```

```
## 60
```

```
sink()
```

4. 把 date_dis 中发病年月转换成 monyy 格式，这里 mon 是指如 SEP 的英文三字母缩写，yy 是两数字的年份。

```
format(as.POSIXct(date_dis, format = "%Y 年%m 月%d 日"), "%b%y")
```

```
## [1] "1月22"
```

5. 从发病日开始计算，需要每隔一个星期进行复诊，请计算未来 10 次的复诊时间。

```
seq(as.POSIXct(date_dis, format = "%Y 年%m 月%d 日"), by = "week", length.out = 10)
```

```
##  [1] "2022-01-01 CST" "2022-01-08 CST" "2022-01-15 CST" "2022-01-22 CST"
##  [5] "2022-01-29 CST" "2022-02-05 CST" "2022-02-12 CST" "2022-02-19 CST"
##  [9] "2022-02-26 CST" "2022-03-05 CST"
```

1.4 因子类型数据

```
x <- factor(c(" 男", " 女", " 男", " 女"))
```

1. 变量 x 有几个水平值，分别是什么?

```
levels(x)
```

```
## [1] "男" "女"
```

2. 通过 levels() 函数把 x 中的“男”改成“M”，“女”改成“F”，并将 x 通过 cat() 函数输出到 factor.txt 文档中。

```
levels(x) <- c("M", "F")
sink("factor.txt")
cat(x)
```

```
## 1 2 1 2
```

```
sink()
```

3. 将以下因子类型数据按照年级从低到高进行排序。

```
y <- factor(c(" 大一", " 大四", " 大三", " 大二"))
```

```
y <- factor(y, levels = c(" 大一", " 大二", " 大三", " 大四"))
```

2 向量, 矩阵和数组

2.1 向量

现有 10 个人的期末考试成绩为

100, 65, 80, 79, 88, 95, 93, 35, 56, 68

1. 创建向量 x 来存储上述数据;

```
x <- c(100, 65, 80, 79, 88, 95, 93, 35, 56, 68)
```

2. 将 x 从大到小排序, 并分别找出最大值、最小值、中位数、第三大和第三小的元素;

```
sort(x, decreasing = TRUE)
```

```
##  [1] 100  95  93  88  80  79  68  65  56  35
```

```
max(x)
```

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

```
min(x)
```

```
## [1] 35
```

```
median(x)
```

```
## [1] 79.5
```

```
sort(x, decreasing = TRUE)[3]
```

```
## [1] 93
```

```
sort(x, decreasing = FALSE)[3]
```

```
## [1] 65
```

3. 计算平均值、标准差和方差;

```
mean(x)
```

```
## [1] 75.9
```

```
sd(x)
```

```
## [1] 20.16846
```

```
var(x)
```

```
## [1] 406.7667
```

4. 60 分以上为及格, 计算及格的人数;

```
sum(x >= 60)
```

```
## [1] 8
```

5. 给 x 命名，名字为学号，分别为”PB1”, “PB2”, … “PB10”（提示，可以先创建以名字为元素的字符型向量 y，然后进行赋值）；

```
y <- paste0("PB", 1:10)
names(x) <- y
```

6. 提取奇数位置的元素并显示；

```
x[seq(1, length(x), 2)]
```

```
## PB1 PB3 PB5 PB7 PB9
## 100  80  88  93  56
```

7. 返回偶数（不包含缺失值）。

```
x[seq(2, length(x), 2)]
```

```
##  PB2  PB4  PB6  PB8 PB10
##   65   79   95   35   68
```

2.2 矩阵和数组

1. 创建一个 10×10 的矩阵 m，其对角线元素分别为 1, 2, …, 10；

```
m <- diag(1:10)
```

2. 计算 m 的逆，并求解关于 y 的线性方程组 $my = x$；

```
y <- 1:10
solve(m, y)
```

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

3. 将上述矩阵的第 2 行除对角线外的所有元素赋值为 100；

```
m[2, -2] <- 100
```

4. 提取 m 的第 5 行第 9 列的元素并输出到命令行，然后将其重新赋值为 NA；

```
m[5, 9]
```

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

```
m[5, 9] <- NA
```

5. 给 m 的列命名为”c1”, “c2”,…“c10”；

```
colnames(m) <- paste0("c", 1:10)
```

6. 给 m 增加一行，元素为上面定义的向量 x，存为新的矩阵 m2;

```
m2 <- rbind(m, x)
```

7. 计算 m2 的行数、列数，核对是否正确;

```
nrow(m2)
```

```
## [1] 11
```

```
ncol(m2)
```

```
## [1] 10
```

8. 对 m2 按行计算平均值，输出的结果不能包含 NA;

```
apply(m2, 1, mean, na.rm = TRUE)
```

```
##
##  0.1000000 90.2000000  0.3000000  0.4000000  0.5555556  0.6000000  0.7000000
##                                           x
##  0.8000000  0.9000000  1.0000000 75.9000000
```

9. 构造一个 $10 \times 10 \times 3$ 的数组 arr，其中 arr[,,1] 为 m，arr[,,1] 为 2*m，arr[,,1] 为 3*m。

```
arr <- array(1:300, dim = c(10, 10, 3))
arr[,,1] <- m
arr[,,2] <- 2 * m
arr[,,3] <- 3 * m
```