第五周——日期、字符与特殊值处理

题目目的

（一）掌握日期型数据的处理与操作。

（二）掌握正则表达式与字符串检索。

（三）掌握特殊值处理方法。

题目

题目一：日期和时间操作。创建脚本文件 test0501.R，在脚本文件中完成下面操作。

用 Sys.Date() 和 Sys.Time() 函数分别获取当前的日期和时间，分别赋值给 dt 和 tm，然后用 print 函数显示它们的值，用 class 显示它们的对象类型。验证 dt 和 tm 分别减去 30 的结果，并用注释语句解释结果的意义。

用 date() 函获取当前系统的日期赋值给 dts，用适当的函数显示其对象类型，验证它减去 30 的结果，解释出现错误的原因。

定义向量 x 为 c(‘2025-3-31’,‘2025-3-24’)，用 as.Date 把 x 转换成日期型数据，并赋值给 x.date，计算两个日期相差的天数。

定义向量 y 为 c(‘2025-3-24;9:20:45’,‘2025-3-24;9:10:45’)，试用 as.Date 将 y 转换日期型数据，然后计算它们的差；试用 strptime 转换为时间型数据，然后计算它的差，并用注释语句解释 as.Date 和 strptime 两个函数的有什么不同。

dt1 = “2025-3-24 8:30:15”，dt2 = “2025-3-24 10:30:15”, 直接用 difftime 计算两者相差多少小时。

用 strptime 转换 dt1 和 dt2 后，再用 difftime 计算它们相差多少小时；并验证使用 as.Date 函数转换后，再用 difftime 计算它们相差多少小时，可否能得到正确的结果？

题目二：paste 和 paste0 函数、转义字符。新脚本文件 test0502.R，完成下面任务。

请选择适当函数创建如下图所示的字符串向量 x：

```
 [1] "a-1"  "b-2"  "c-3"  "d-4"  "e-5"  "f-6"  "g-7"  "h-8"  "i-9"  "j-10"
[11] "k-11" "l-12" "m-13" "n-14" "o-15" "p-16" "q-17" "r-18" "s-19" "t-20"
[21] "u-21" "v-22" "w-23" "x-24" "y-25" "z-26" "a-27" "b-28" "c-29" "d-30"
[31] "e-31" "f-32" "g-33" "h-34" "i-35" "j-36" "k-37" "l-38" "m-39" "n-40"
[41] "o-41" "p-42" "q-43" "r-44" "s-45" "t-46" "u-47" "v-48" "w-49" "x-50"
[51] "y-51" "z-52"
```

请选择适当函数创建如下图所示的字符串向量 y：

```
 [1] "a-1"  "a-2"  "b-3"  "b-4"  "c-5"  "c-6"  "d-7"  "d-8"  "e-9"  "e-10"
[11] "f-11" "f-12" "g-13" "g-14" "h-15" "h-16" "i-17" "i-18" "j-19" "j-20"
[21] "k-21" "k-22" "l-23" "l-24" "m-25" "m-26" "n-27" "n-28" "o-29" "o-30"
[31] "p-31" "p-32" "q-33" "q-34" "r-35" "r-36" "s-37" "s-38" "t-39" "t-40"
[41] "u-41" "u-42" "v-43" "v-44" "w-45" "w-46" "x-47" "x-48" "y-49" "y-50"
[51] "z-51" "z-52"
```

用一次 cat 函数在命令窗口中输出如下形状的内容：

```
远上寒山石径斜，白天生处有人家。
停车坐爱枫林晚，霜叶红于二月花。
```

题目三：使用正则表达式。打开脚本文件 test0503.R，完成以下操作。

找出含有 es 或 se 的单词。

找出结尾是 es 或 se 的单词。

找出以大写英文字母开头的单词。

找出开头是小写英文字母，而其他位置含有大写字母的单词。

找出含有非英文字母的单词。

找出包含有数字的单词。

找出有连续重叠字母的单词，不区分大小写。

```
x <- scan('Solomon.txt',
          what = '',
          quote = "",
          fileEncoding = 'gb2312')
```

题目四：sub、gsub 函数。打开脚本文件 test0504.R，完成下面操作。

使用 grep 函数把 x 中的含有单引号、双引号、问号、句号、逗号、分号、冒号和感叹号的单词找出，并显示单词本身。

删除字符串中的” 与标点符号（.,?;:!），并把结果赋值给 y。

```
x<-scan('Solomon2.txt',
        what = '',
        quote = "",
        fileEncoding = 'gb2312')
```

题目五：grepl、regexpr、gregexpr 函数。打开脚本文件 test0505.R，完成下面操作。

用 grepl 找出向量 x 中包含 an 字符且出现 2 次及 2 次以上的字符串

用 regexpr 找出向量 x 中包含 an 字符的字符串。

用 gregexpr 找出向量 x 中包含 an 字符的字符串

请注释语句回答 grep、grepl、regexpr、gregexpr 这四个函数的返回结果有什么不同？

```
x = c('these are bananas and oranges',
      'these are apples and ...',
      'these are peaches')
```

题目六：substr 函数和 substring 函数。创建脚本文件 test0506.R，完成下面操作。

用 substr 函数和 substring 函数”R is a programming language for statistical computing and graphics” 中第 8 个字符开始长度为 7 的子字符串。

把字符串”R is a programming language for statistical computing and graphics” 中第 8 至第 18 个字符修改为” 程序设计”，观察修改结果，可得出什么结论？

题目七：strsplit 函数。打开脚本文件 test0507.R，完成下面操作。

把向量 x 合并为一个字符串，赋值给 y。

对字符串 y 进行分词操作，把操作结果赋值给 z。

删除 z 中的空字符串。

统计 w 中各个单词出现的频数。

```
x = readLines("Solomon2.txt")
```

题目八：特殊数据处理。创建脚本文件 test0508.R，完成下面操作。

创建向量 x，其元素为 NaN、1、NA、3、Inf、5、NULL，用关系表达式判断向量 x 的长度是否等于 7。

分别对 NaN、NA、Inf 和 NULL 进行处理，实现以下目标：

- 删除向量中的 NaN 和 NA 值。
- 删除向量中 Inf 值。

答案及解析

题目一：

```
dt <- Sys.Date()
tm <- Sys.time()
print(dt)
```

```
[1] "2025-08-30"
```

```
print(tm)
```

```
[1] "2025-08-30 19:44:11 CST"
```

```
class(dt)
```

```
[1] "Date"
```

```
class(tm)
```

```
[1] "POSIXct" "POSIXt"
```

```
dt-30 # 日期减 30
```

```
[1] "2025-07-31"
```

```
tm-30 # 秒数减 30
```

```
[1] "2025-08-30 19:43:41 CST"
```

```
dts <- date()
class(dts)
```

```
[1] "character"
```

```
#dts - 30
# 错误于 dts - 30: 二进列运算符中有非数值参数

x <- c('2025-3-31','2025-3-24')
x.date <- as.Date(x)
diff(x.date)
```

```
Time difference of -7 days
```

```
y <- c('2025-3-24;9:20:45','2025-3-24;9:10:45')
as.Date(y)
```

```
[1] "2025-03-24" "2025-03-24"
```

```
y.time <- strptime(y,"%Y-%m-%d;%H:%M:%S")
diff(y.time)
```

```
Time difference of -10 mins
```

```
dt1 = "2025-3-24 8:30:15"
dt2 = "2025-3-24 10:30:15"
difftime(strptime(dt2, format = "%Y-%m-%d %H:%M:%S"),
         strptime(dt1, format = "%Y-%m-%d %H:%M:%S"), units = "hours")
```

```
Time difference of 2 hours
```

```
dt1.time <- strptime(dt1, format = "%Y-%m-%d %H:%M:%S")
dt2.time <- strptime(dt2, format = "%Y-%m-%d %H:%M:%S")
difftime(dt2.time, dt1.time, units = "hours")
```

```
Time difference of 2 hours
```

```
dt1.date <- as.Date(dt1)
dt2.date <- as.Date(dt2)
difftime(dt2.date, dt1.date, units = "hours")
```

```
Time difference of 0 hours
```

题目二：

```
x <- letters
y <- 1:52
paste(x,y,sep = '-')
```

```
 [1] "a-1"  "b-2"  "c-3"  "d-4"  "e-5"  "f-6"  "g-7"  "h-8"  "i-9"  "j-10"
[11] "k-11" "l-12" "m-13" "n-14" "o-15" "p-16" "q-17" "r-18" "s-19" "t-20"
```

```
[21] "u-21" "v-22" "w-23" "x-24" "y-25" "z-26" "a-27" "b-28" "c-29" "d-30"
[31] "e-31" "f-32" "g-33" "h-34" "i-35" "j-36" "k-37" "l-38" "m-39" "n-40"
[41] "o-41" "p-42" "q-43" "r-44" "s-45" "t-46" "u-47" "v-48" "w-49" "x-50"
[51] "y-51" "z-52"
```

```
a <- rep(letters,each = 2)
paste(a,y,sep = '-')
```

```
 [1] "a-1"  "a-2"  "b-3"  "b-4"  "c-5"  "c-6"  "d-7"  "d-8"  "e-9"  "e-10"
[11] "f-11" "f-12" "g-13" "g-14" "h-15" "h-16" "i-17" "i-18" "j-19" "j-20"
[21] "k-21" "k-22" "l-23" "l-24" "m-25" "m-26" "n-27" "n-28" "o-29" "o-30"
[31] "p-31" "p-32" "q-33" "q-34" "r-35" "r-36" "s-37" "s-38" "t-39" "t-40"
[41] "u-41" "u-42" "v-43" "v-44" "w-45" "w-46" "x-47" "x-48" "y-49" "y-50"
[51] "z-51" "z-52"
```

```
x1 <- '远上寒山石径斜，白天生处有人家。\n停车坐爱枫林晚，霜叶红于二月花。'
cat(x1)
```

```
远上寒山石径斜，白天生处有人家。
停车坐爱枫林晚，霜叶红于二月花。
```

题目三：

```
x <- scan('Solomon.txt',
          what = '',
          quote = "",
          fileEncoding = 'gb2312')

x[grepl('es$',x)]
```

```
[1] "babies"  "riches"  "enemies" "riches"  "slaves"
```

```
x[grepl('se$',x)]
```

```
[1] "house"   "Please"  "chose"   "because" "because" "those"   "those"
```

```
x[grepl('^[A-Z]',x)]
```

```
 [1] "Long"      "Solomon"   "He"        "In"        "They"      "One"
 [7] "The"       "The"       "No"        "The"       "Each"      "So"
[13] "King"      "Solomon"   "Bring"     "King"      "Oh"        "Your"
[19] "MajestTy"  "Give"      "Please"    "Then"      "King"      "Solomon"
[25] "Give"      "She"       "God"       "Solomon"   "This"      "I"
[31] "I"         "Why"       "Now"       "I"         "God"       "When"
[37] "Pharaoh"   "Jerusalem" "Solomon"   "I"         "King"      "SoloMon"
[43] "I"         "Solomon"   "PhaRaoh's" "I"         "I"         "King"
[49] "Solomon"   "God"       "I"         "I"         "At"        "I"
[55] "They"      "It"        "I"
```

```
x[grepl('^[a-z].*[A-X]',x)]
```

```
[1] "moTher"       "younGest"     "converSation"
```

```
x[grepl('[^a-zA-Z]',x)]
```

```
 [1] "women3"    "baby's"    "woman's"   "don't"     "couldn't"  "hadn't"
 [7] "so-called" "PhaRaoh's" "above2"    "don't"
```

```
x[grepl('\\d',x)]
```

```
[1] "women3" "above2"
```

```
x[grepl('([a-z])\\1',x,ignore.case = T)]
```

```
 [1] "took"      "quarrel"   "see"       "kill"      "needs"     "good"
 [7] "called"    "finally"   "alliance"  "so-called" "marrying"  "wedding"
[13] "too"       "soon"      "makiing"   "pulling"   "looks"     "feel"
```

题目四：

```
x <- scan('Solomon.txt',
          what = '',
          quote = "",
          fileEncoding = 'gb2312')

x[grepl('es$',x)]
```

```
[1] "babies"  "riches"  "enemies" "riches"  "slaves"
```

```
x[grepl('se$',x)]
```

```
[1] "house"   "Please"  "chose"   "because" "because" "those"   "those"
```

```
x[grepl('^[A-Z]',x)]
```

```
 [1] "Long"      "Solomon"   "He"        "In"        "They"      "One"
 [7] "The"       "The"       "No"        "The"       "Each"      "So"
[13] "King"      "Solomon"   "Bring"     "King"      "Oh"        "Your"
[19] "MajestTy"  "Give"      "Please"    "Then"      "King"      "Solomon"
[25] "Give"      "She"       "God"       "Solomon"   "This"      "I"
[31] "I"         "Why"       "Now"       "I"         "God"       "When"
[37] "Pharaoh"   "Jerusalem" "Solomon"   "I"         "King"      "SoloMon"
[43] "I"         "Solomon"   "PhaRaoh's" "I"         "I"         "King"
[49] "Solomon"   "God"       "I"         "I"         "At"        "I"
[55] "They"      "It"        "I"
```

```
x[grepl('^[a-z].*[A-X]',x)]
```

```
[1] "moTher"       "younGest"     "converSation"
```

```
x[grepl('[^a-zA-Z]',x)]
```

```
 [1] "women3"    "baby's"    "woman's"   "don't"     "couldn't"  "hadn't"
 [7] "so-called" "PhaRaoh's" "above2"    "don't"
```

```
x[grepl('\\d',x)]
```

```
[1] "women3" "above2"
```

```
x[grepl('([a-z])\\1',x,ignore.case = T)]
```

```
 [1] "took"      "quarrel"   "see"       "kill"      "needs"     "good"
 [7] "called"    "finally"   "alliance"  "so-called" "marrying"  "wedding"
[13] "too"       "soon"      "makiing"   "pulling"   "looks"     "feel"
```

题目五：

```
x = c('these are bananas and oranges',
      'these are apples and ...',
      'these are peaches')
x[grepl("(an).*\\1", x)]
```

```
[1] "these are bananas and oranges"
```

```
regexpr("an", x)
```

```
[1] 12 18 -1
attr(,"match.length")
[1]  2  2 -1
attr(,"index.type")
[1] "chars"
attr(,"useBytes")
[1] TRUE
```

```r
gregexpr("an", x)
```

```
[[1]]
[1] 12 14 19 25
attr(,"match.length")
[1] 2 2 2 2
attr(,"index.type")
[1] "chars"
attr(,"useBytes")
[1] TRUE

[[2]]
[1] 18
attr(,"match.length")
[1] 2
attr(,"index.type")
[1] "chars"
attr(,"useBytes")
[1] TRUE

[[3]]
[1] -1
attr(,"match.length")
[1] -1
attr(,"index.type")
[1] "chars"
attr(,"useBytes")
[1] TRUE
```

```r
# - grep: 返回匹配的元素的索引。
# - grepl: 返回逻辑向量，表示每个元素是否匹配。
# - regexpr: 返回第一个匹配项的位置和长度，结果是一个数值向量。
# - gregexpr: 返回所有匹配项的位置和长度，结果是一个列表，每个元素对应一个字符串的所有匹配项。
```

题目六：

```
original_string <- "R is a programming language for statistical computing and graphics"
substr_result <- substr(original_string, start = 8, stop = 8 + 7 - 1)
print(substr_result)
```

```
[1] "program"
```

```
substring_result <- substring(original_string, first = 8, last = 8 + 7 - 1)
print(substring_result)
```

```
[1] "program"
```

```
part1 <- substr(original_string, 1, 7)
part2 <- substr(original_string, 19, nchar(original_string))
modified_string <- paste0(part1, " 程序设计", part2)
print(modified_string)
```

```
[1] "R is a 程序设计 language for statistical computing and graphics"
```

题目七：

```
x = readLines("Solomon2.txt")

y <- as.character(x)

z <- strsplit(y,split = ' ')[[1]]
z[z ==''] <- NA
word_freq <- table(z)
```

题目八：

```
x = readLines("Solomon2.txt")

y <- as.character(x)

z <- strsplit(y,split = ' ')[[1]]
z[z ==''] <- NA
word_freq <- table(z)
```