Untitled

Rongbing Xv

2023-08-11

Table of Contents

[7 Modifying Values 1](#_Toc142743181)

[7.0.1 改变原值 2](#_Toc142743182)

[7.0.2 Logical Subsetting逻辑提取子集 3](#_Toc142743183)

[为hearts游戏制作deck4 4](#_Toc142743184)

[7.0.3布尔运算符 5](#_Toc142743185)

[为"二十一点"游戏制作deck5: 6](#_Toc142743186)

[7.0.4缺失信息 NA (Not Available) 8](#_Toc142743187)

# 7 Modifying Values

你准备好用你的虚拟牌组玩一些游戏了吗？不要那么快！你牌组中的点数系统与许多纸牌游戏不太一致。例如，在战争和扑克中，王牌的得分通常高于国王。它们的点值是14，而不是1。

在这项任务中，你将改变牌组的点数系统三次，以匹配三种不同的游戏：战争、红心大战和21点(war, hearts, and blackjack)。这些游戏中的每一个都会教你一些关于修改数据集内部值的不同内容。首先制作一个可以操作的牌组副本。这将确保您始终拥有一份原始的牌组（如果出现问题）：

deck = data.frame(face = c("king", "queen", "jack", "ten", "nine", "eight", "seven", "six",  
 "five", "four", "three", "two", "ace", "king", "queen", "jack", "ten",   
 "nine", "eight", "seven", "six", "five", "four", "three", "two", "ace",   
 "king", "queen", "jack", "ten", "nine", "eight", "seven", "six", "five",   
 "four", "three", "two", "ace", "king", "queen", "jack", "ten", "nine",   
 "eight", "seven", "six", "five", "four", "three", "two", "ace"),  
 suit = c("spades", "spades", "spades", "spades", "spades", "spades",   
 "spades", "spades", "spades", "spades", "spades", "spades", "spades",   
 "clubs", "clubs", "clubs", "clubs", "clubs", "clubs", "clubs", "clubs",   
 "clubs", "clubs", "clubs", "clubs", "clubs", "diamonds", "diamonds",   
 "diamonds", "diamonds", "diamonds", "diamonds", "diamonds", "diamonds",   
 "diamonds", "diamonds", "diamonds", "diamonds", "diamonds", "hearts",   
 "hearts", "hearts", "hearts", "hearts", "hearts", "hearts", "hearts",   
 "hearts", "hearts", "hearts", "hearts", "hearts"),  
 value = c(13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 13, 12, 11, 10, 9, 8,   
 7, 6, 5, 4, 3, 2, 1, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 13, 12, 11,   
 10, 9, 8, 7, 6, 5, 4, 3, 2, 1))#保留原始数据

deck2 = deck#后续操作使用deck2，以免破坏原始数据

为"战争"游戏制作deck3

### 7.0.1 改变原值

R中可以使用**符号系统(Notions)**和**赋值运算(<-/=)**来修改原值。

在"战争"游戏中，游戏规则规定牌面为**ace**的卡牌所对应的点数值为14（初始为13）。如果牌组没有被打乱过，那么每过十三张我们就会遇到**ace**牌，我们可以手动记住他们对应的位置：13，26，39，52来改变他们的对应点数值。

然而，如果牌被洗过了，那么就需要一些**逻辑判断**来让R帮助我们定位它们的位置，以进行后续的修改：

shuffle = function(deck2){  
 random = sample(1:52,size = 52)  
 deck2 = deck2[random,]  
}

deck3 = shuffle(deck2)#洗牌  
head(deck3)

## face suit value  
## 35 five diamonds 5  
## 52 ace hearts 1  
## 17 ten clubs 10  
## 30 ten diamonds 10  
## 37 three diamonds 3  
## 14 king clubs 13

### 7.0.2 Logical Subsetting逻辑提取子集

R具有七种逻辑运算符：

| **Operator** | **Syntax** | **Tests** |
| --- | --- | --- |
| > | a > b | Is a greater than b? |
| >= | a >= b | Is a greater than or equal to b? |
| < | a < b | Is a less than b? |
| <= | a <= b | Is a less than or equal to b? |
| == | a == b | Is a equal to b? |
| != | a != b | Is a not equal to b? |
| %in% | a %in% c(a, b, c) | Is a in the group c(a, b, c)? |

来测试一下打乱的牌组中有几张**ace**?

sum(deck3$face == "ace")

## [1] 4

使用逻辑语句来对ace牌的对应数值进行更改：

deck3$value[deck3$face == "ace"] = 14  
deck3

## face suit value  
## 35 five diamonds 5  
## 52 ace hearts 14  
## 17 ten clubs 10  
## 30 ten diamonds 10  
## 37 three diamonds 3  
## 14 king clubs 13  
## 2 queen spades 12  
## 33 seven diamonds 7  
## 44 nine hearts 9  
## 18 nine clubs 9  
## 45 eight hearts 8  
## 5 nine spades 9  
## 1 king spades 13  
## 48 five hearts 5  
## 39 ace diamonds 14  
## 21 six clubs 6  
## 13 ace spades 14  
## 4 ten spades 10  
## 29 jack diamonds 11  
## 9 five spades 5  
## 16 jack clubs 11  
## 28 queen diamonds 12  
## 38 two diamonds 2  
## 46 seven hearts 7  
## 7 seven spades 7  
## 24 three clubs 3  
## 19 eight clubs 8  
## 32 eight diamonds 8  
## 8 six spades 6  
## 41 queen hearts 12  
## 47 six hearts 6  
## 26 ace clubs 14  
## 10 four spades 4  
## 6 eight spades 8  
## 22 five clubs 5  
## 27 king diamonds 13  
## 25 two clubs 2  
## 11 three spades 3  
## 20 seven clubs 7  
## 3 jack spades 11  
## 42 jack hearts 11  
## 51 two hearts 2  
## 49 four hearts 4  
## 23 four clubs 4  
## 15 queen clubs 12  
## 12 two spades 2  
## 50 three hearts 3  
## 31 nine diamonds 9  
## 34 six diamonds 6  
## 36 four diamonds 4  
## 43 ten hearts 10  
## 40 king hearts 13

## 为hearts游戏制作deck4

deck4 = deck2  
deck4$value = 0  
head(deck4) #此时deck4所有牌对应值为0

## face suit value  
## 1 king spades 0  
## 2 queen spades 0  
## 3 jack spades 0  
## 4 ten spades 0  
## 5 nine spades 0  
## 6 eight spades 0

Assign a value of 1 to every card in deck4 that has a suit of **hearts**.

deck4$value[deck4$suit == "hearts"] = 1  
tail(deck4)

## face suit value  
## 47 six hearts 1  
## 48 five hearts 1  
## 49 four hearts 1  
## 50 three hearts 1  
## 51 two hearts 1  
## 52 ace hearts 1

如果我们要把**spades花色**的**queen牌**所对应的值改为13呢？如何能让逻辑判断同时满足两个条件限制？此时需要布尔运算符Boolean Operators来把多个逻辑判断进行组合。

### 7.0.3布尔运算符

| **Operator** | **Syntax** | **Tests** |
| --- | --- | --- |
| & | cond1 & cond2 | Are both cond1 and cond2 true? |
| | | cond1 | cond2 | Is one or more of cond1 and cond2 true? |
| xor | xor(cond1, cond2) | Is exactly one of cond1 and cond2 true? |
| ! | !cond1 | Is cond1 false? (e.g., ! flips the results of a logical test) |
| any | any(cond1, cond2, cond3, ...) | Are any of the conditions true? |
| all | all(cond1, cond2, cond3, ...) | Are all of the conditions true? |

实现刚刚的需求：

deck4$value[deck4$face == "queen" & deck4$suit == "spades"] = 13  
deck4$value[deck4$suit == "spades"]#此时黑桃queen对应的值为13

## [1] 0 13 0 0 0 0 0 0 0 0 0 0 0

测试：

If you think you have the hang of logical tests, try converting these sentences into tests written with R code. To help you out, I've defined some R objects after the sentences that you can use to test your answers:

* Is w positive?
* Is x greater than 10 and less than 20?
* Is object y the word February?
* Is *every* value in z a day of the week?

w <- c(-1, 0, 1)  
x <- c(5, 15)  
y <- "February"  
z <- c("Monday", "Tuesday", "Friday")

solution：

w > 0

## [1] FALSE FALSE TRUE

x > 10 & x < 20

## [1] FALSE TRUE

y == "February"

## [1] TRUE

z %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")

## [1] TRUE TRUE TRUE

## 为"二十一点"游戏制作deck5:

在二十一点中，每张**数字牌**的值等于其面值。每张**花牌（K、Q 或 J）**的值为 10。最后，每张**ACE** 的值为 11 或 1，具体取决于游戏的最终结果。

deck5 = deck  
head(deck5,13)

## face suit value  
## 1 king spades 13  
## 2 queen spades 12  
## 3 jack spades 11  
## 4 ten spades 10  
## 5 nine spades 9  
## 6 eight spades 8  
## 7 seven spades 7  
## 8 six spades 6  
## 9 five spades 5  
## 10 four spades 4  
## 11 three spades 3  
## 12 two spades 2  
## 13 ace spades 1

修改对应的点数值：

deck5$value[deck5$face == "jack"|deck5$face == "queen"|deck5$face == "king"] = 10  
deck5

## face suit value  
## 1 king spades 10  
## 2 queen spades 10  
## 3 jack spades 10  
## 4 ten spades 10  
## 5 nine spades 9  
## 6 eight spades 8  
## 7 seven spades 7  
## 8 six spades 6  
## 9 five spades 5  
## 10 four spades 4  
## 11 three spades 3  
## 12 two spades 2  
## 13 ace spades 1  
## 14 king clubs 10  
## 15 queen clubs 10  
## 16 jack clubs 10  
## 17 ten clubs 10  
## 18 nine clubs 9  
## 19 eight clubs 8  
## 20 seven clubs 7  
## 21 six clubs 6  
## 22 five clubs 5  
## 23 four clubs 4  
## 24 three clubs 3  
## 25 two clubs 2  
## 26 ace clubs 1  
## 27 king diamonds 10  
## 28 queen diamonds 10  
## 29 jack diamonds 10  
## 30 ten diamonds 10  
## 31 nine diamonds 9  
## 32 eight diamonds 8  
## 33 seven diamonds 7  
## 34 six diamonds 6  
## 35 five diamonds 5  
## 36 four diamonds 4  
## 37 three diamonds 3  
## 38 two diamonds 2  
## 39 ace diamonds 1  
## 40 king hearts 10  
## 41 queen hearts 10  
## 42 jack hearts 10  
## 43 ten hearts 10  
## 44 nine hearts 9  
## 45 eight hearts 8  
## 46 seven hearts 7  
## 47 six hearts 6  
## 48 five hearts 5  
## 49 four hearts 4  
## 50 three hearts 3  
## 51 two hearts 2  
## 52 ace hearts 1

#或者：  
# facecard <- deck5$face %in% c("king", "queen", "jack")  
# deck5$value[facecard] <- 10

很难决定 ACE 的对应值是多少，因为它们的确切值会随着牌局的不同而变化。每手牌结束时，如果玩家的牌总点数不超过 21，则 ACE 等于 11。否则，ACE等于 1。ACE 的实际值将取决于玩家手中的其他牌。这是一个信息缺失的情况。目前，您没有足够的信息来为 ACE 卡分配正确的分值。

### 7.0.4缺失信息 NA (Not Available)

#### 7.0.4.1缺失值的传播性：

na = c(NA,1:50)  
na

## [1] NA 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24  
## [26] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49  
## [51] 50

mean(na)#包含NA，返回NA

## [1] NA

当运算过程中一旦包含NA，那么结果就会返回NA，从而使计算过程无法进行。此时，需要参数na.rm 来删除缺失值：

mean(na,na.rm = TRUE)

## [1] 25.5

#### 7.0.4.2 is.na

判断一组数据中是否有缺失值：

is.na(na)

## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49] FALSE FALSE FALSE

现在，我们将所有ACE牌的点数设为NA：

deck5$value[deck5$face == "ace"] = NA  
head(deck5,13)

## face suit value  
## 1 king spades 10  
## 2 queen spades 10  
## 3 jack spades 10  
## 4 ten spades 10  
## 5 nine spades 9  
## 6 eight spades 8  
## 7 seven spades 7  
## 8 six spades 6  
## 9 five spades 5  
## 10 four spades 4  
## 11 three spades 3  
## 12 two spades 2  
## 13 ace spades NA