第五周作业

1. 输入a的值与相加数字的个数，求s=a+aa+aaa+aaaa+aa...a的值，其中a是一个数字。例如2+22+222+2222+22222(此时共有5个数相加)
2. 在掷骰子游戏中，玩家（射手）准备掷出两个骰子，每个骰子都有六个面。如果掷出的总数为2、3 或12，则该射手失败。如果总数是7 或11，则对手（他）胜出。如果是其他任何得分，那么此得分将变成新的目标，它将被称为“点”。使用以下辅助函数为掷骰子生成得分：

two\_d6 <- function(n)

{

random\_numbers <- matrix(

sample(6, 2 \* n, replace = TRUE),

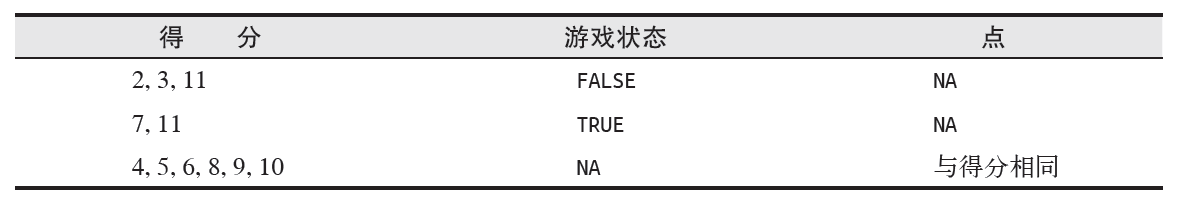
nrow = 2

)

colSums(random\_numbers)

}

编写代码为掷骰子生成得分，并将以下值分配给game\_status 和point 变量：



如果射手没有马上赢或马上输，那么他必须不断地滚动骰子，直到他赢得得分点值取胜，或得分为7 而失败为止。编写代码检查游戏状态是否为NA，如果是则反复生成掷骰子直到碰到点值为止（设置game\_status 为TRUE）或得分为7（设置game\_status 为FALSE）。

1. 这是著名的sea shells 绕口令：

sea\_shells <- c(

"She", "sells", "sea", "shells", "by", "the", "seashore",

"The", "shells", "she", "sells", "are", "surely", "seashells",

"So", "if", "she", "sells", "shells", "on", "the", "seashore",

"I'm", "sure", "she", "sells", "seashore", "shells"

)

使用nchar 函数来计算每个单词的字母数。现在，循环遍历所有可能的单词长度，找出所有与其长度相等的单词。例如，长度为6 的单词应该有shell 和surely，它们都有六个字母

sea\_shells <- c(

"She", "sells", "sea", "shells", "by", "the", "seashore",

"The", "shells", "she", "sells", "are", "surely", "seashells",

"So", "if", "she", "sells", "shells", "on", "the", "seashore",

"I'm", "sure", "she", "sells", "seashore", "shells"

)

a<-nchar(sea\_shells)

a1<-unique(a)

k<-list()

for(i in 1:length(a1)){

k[[i]]<-sea\_shells[which(a==a1[i])]

print(k)

}

unique(k)

1. 判断101-200之间有多少个素数，并输出所有素数

a1<-101:200

a2<-1:100

x<-c()

k<-c()

for(i in 1:length(a1)){

a<-c()

a<-a1[i]/a2

if(length(which((a%%1)==0))==1)

x[i]<-a1[i]

}

na.omit(x)

length(na.omit(x))

1. 一个5位数，判断它是不是回文数。即12321是回文数，个位与万位相同，十位与千位相同。

hw<-function(x){

ts<-floor(x/10000)

s<-floor(x/1000)%%10

h<-floor(x/100)%%10

t<-floor(x/10)%%10

g<-floor(x%%10)

if(ts==g&&s==t){

message("回文数")

}else{

message("不是回文数")

}

}

1. 输入一个整数，判断它能否被3、5、7整除，并输出以下信息之一：

⑴能同时被3、5、7整除；

⑵能被其中两个数（说明哪两个数）整除；

⑶只能被其中一个整除（说明哪个数）；

⑷不能被3、5、7任何一个整除。

zc<-function(x){

x1<-x%%3

x2<-x%%5

x3<-x%%7

k<-c()

if(x1==0&&x2==0&&x3==0){

message("能同时被3、5、7整除")

}

if(x1>0&&x2>0&&x3>0){

message("不能被3、5、7任何一个整除")

}else{

if(x1==0){

message("能被3整除")

}

if(x2==0){

message("能被5整除")

if(x3==0){

message("能被7整除")

}

}

}

}

1. 企业发放的奖金根据利润提成。利润(I)低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万到60万之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%，高于100万元时，超过100万元的部分按1%提成，输入当月利润I，求应发放奖金总数？

f<-function(I){

if((I-10000000)>0){

a<-(I-1000000)\*0.01+(1000000-600000)\*0.015+(600000-400000)\*0.03+(400000-200000)\*0.05+(200000-100000)\*0.075+(100000)\*0.1

}

if(I-600000>0){

a<-((I-600000)>0)\*0.015+(600000-400000)\*0.03+(400000-200000)\*0.05+(200000-100000)\*0.075+(100000)\*0.1

}

if((I-400000)>0){

a<-(I-400000)\*0.03+(400000-200000)\*0.05+(200000-100000)\*0.075+(100000)\*0.1

}

if((I-200000)>0){

a<-(I-200000)\*0.05+(200000-100000)\*0.075+(100000)\*0.1

}

a<-(I-100000)\*0.075+(100000)\*0.1

print(a)

}

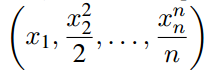
1. 使用循环语句，编写一个函数，用于寻找向量x中的最小值（不能用现成的函数，如min，sort等）

f1<-function(x){

a<-x[order(x)]

print(a[1])

}

1. （a）编写函数tmpFn1 和 函数 tmpFn2。记xVec表示向量，则tmpFn1（xVec）返回，tmpFn2（xVec）返回

fa<-function(xVec){

y<-1:length(xVec)

print(xVec\*y)

}

fb<-function(xVec){

y<-1:length(xVec)

print((xVec\*y)/y)

}

1. 编写函数以实现以下功能：输入一个矩阵，返回同样维度大小的矩阵，将原矩阵中的奇数变为原来的2倍，偶数不变。

|  |
| --- |
| > m<-function(a){  + if(is.matrix(a)){  + k<-c()  + for(i in a){  + k[i]<-i  + }  + k[which(k%%2>0)]<-k[which(k%%2>0)]\*2  + b<-matrix(k,nrow=nrow(a),byrow=TRUE)  + print(b)  + }else{  + message("not a matrix")  + }  + }  > m(a)  [,1] [,2]  [1,] 2 2  [2,] 6 4 |
|  |
| |  | | --- | | > | |