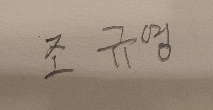
Mid takehome

12151487 조규영

본 과제는 작성자 본인이 직접 작성한 내용으로 통계소프트웨어 실습을 수강하는 어느 누구와도 상의하지 않고 스스로 작성한 내용임을 서약합니다. 이를 위반 시 학점 불인정(F학점)에 대하여 동의합니다.

 서약자 : 조 규 영

1. 20개 코드에 대한 서술

assign('x',1:10) : x라는 변수에 1부터 10까지의 정수값을 넣는 함수이다. x = 1:10으로 쓸 수 있기 때문에 잘 사용하진 않지만 반복문을 이용해 x1, x2 … x100 등의 이름이 비슷한 변수에 값을 넣을 때 유용하다.

v = c('a,b,c,b,e','a,b','d,e,f'); lapply(v,nchar) : v라는 변수에 'a,b,c,b,e','a,b','d,e,f' 의 3개의 값을 넣은 벡터를 저장한다. lapply를 통해 v에서 각 원소가 nchar라는 문자열의 길이를 반환하는 함수가 적용되어 각 결과가 리스트 형태로 나타나게 된다. 결과는 다음과 같다.

[[1]]

[1] 9

[[2]]

[1] 3

[[3]]

[1] 5

s = 0; for(i in 1:50) {s = s+i}; s : s라는 변수에 0을 저장하고 for loop에서는 1부터 50번까지 반복문이 실행된다. 각 loop마다 i에는 1부터 50까지의 정수값이 들어가게 되고 s=s+i에서 i값이 s에 계속 더해지게 된다. 즉 1부터 50까지 합하는 코드이고 s를 출력하면 1275라는 값이 나온다.

paste(c('A','B'),collapse='/') : paste는 인수들을 하나로 합치는 문자열이고 sep=” “가 기본값으로 되어있다. sep는 각 인자를 구분해 이어주는 것인데 여기서는 하나의 벡터 인자만 들어간 상태이므로 의미가 없다고 할 수 있다. A와 B는 하나의 벡터 안에 있고 collapse는 이 벡터를 붕괴시켜 하나로 만들게 한다. sep구분자와 마찬가지로 벡터 내 원소 사이에 “/”문자가 들어가게 되어 원소 결과는 “A/B”로 나오게 된다.

rep(c(1,2),times=3) : rep는 원소를 반복해서 생성하는 함수인데 times=3을 설정하면 원소의 순서대로(여기서는 1 2 가) 3번 반복되어 출력된다. 1 2 1 2 1 2 의 형태로 나타나게 된다.

A <- matrix(1:12,4,3);A[-(2:3),-(1:2)] : A라는 변수에 1부터 12까지 정수값, 총 12개를 4\*3의 행렬로저장한다. byrow=F가 기본값으로 설정되어 있어 1열을 먼저 채우고 2열을 채우고 3열을 채워 다음과 같은 형태가 나타난다.

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

[1,] 1 5 9

[2,] 2 6 10

[3,] 3 7 11

[4,] 4 8 12

행렬의 indexing은 A[행, 열]의 형태로 이루어지고 행, 열자리에는 찾는 행, 열의 숫자가 들어간다.그런데 -를 붙이면 해당 행이나 열을 뺀다는 뜻이다. 즉 코드에서는 2, 3행을 빼고 1,2열을 뺀다는 뜻이므로 남는 원소는 A[1,3]와 A[4,3]뿐이고 같은 열에 있는 원소라 벡터 형태로 9 12 가 출력된다.

getwd(); setwd('d:/') : getwd로 현재 작업 폴더 주소가 문자열 형태로 반환된다.(“C:/Users/Gyu”) setwd로 컴퓨터 내 “d:/”의 폴더를 작업 폴더로 지정하여 문서를 읽고 쓸 때 D드라이브 폴더 안에서 파일이 읽고 쓰인다.

x <- matrix(c(NA, 1, 3, NA, NA, 2),3,2); apply(x,2,mean,na.rm=T) : NA, 1, 3, NA, NA, 2의 값들이3\*2행렬 형태로 x라는 변수에 저장되고 byrow=F가 default여서 1열을 먼저 채우고 2열을 채운다. apply는 행렬을 입력받고 행, 열 방향으로 특정 함수를 적용하는 함수이다. apply함수에서 x라는 행렬을 받고 margin방향(1이면 row, 2이면 colunm방향)으로 함수 mean(평균값)이 계산된다. margin이 2이므로 2개 열에서 평균값이 계산되고 na.rm=T로 인해서 NA값은 생각하지 않고 평균값이 계산된다. 결과값은 2 2 가 나온다.

x = c(10,3,6,9); y = c(1,5,4,12); ifelse(x>y,x,y) : x라는 변수에 10,3,6,9가 저장되고 y라는 변수에 1,5,4,12가 저장된다. ifelse문에서 x와 y의 순서대로 10과1, 3과5, 6과4, 9와 12를 비교하여 x가 y보다 크면 x가 출력되고 그렇지 않으면 y값이 출력된다. 결과는 10 5 6 12 이다.

rname <- paste0('n',1:4);cname <- paste0('x',1:3);B <- matrix(1:12,4,3,dimnames=list(rname,cname)); nrow(B)

: paste0는 sep=””로 설정된 함수라 paste와 다르게 공백으로 구분되지 않는다. 1부터 4까지의 정수값이 ‘n’과 각각 합쳐져서 rname 변수에는 “n1” “n2” “n3” “n4” 가 저장된다. 마찬가지로 cname 변수에는 “x1” “x2” “x3”가 저장된다. B에는 1부터 12까지의 정수가 4\*3행렬로 저장되고 byrow=F로 기본값이 설정되어 1열을 채우고 2열을 채우고 3열을 채운다. dimnames는 행과 열의 이름을 설정하는 것이고 리스트 형태로 입력 받아 list(행 이름 목록, 열 이름 목록)처럼 입력받는다. 행은 4개고 열은 3개이기 때문에 개수에 맞게 입력되었다. B는 다음과 같이 출력된다.

x1 x2 x3

n1 1 5 9

n2 2 6 10

n3 3 7 11

n4 4 8 12

nrow는 행의 개수를 출력하는 함수이므로 4가 출력된다.

paste(c('A','B'),1:2,collapse='/') : 4번째 코드에서 말한 sep과 collapse의 기능대로 설명하면, sep으로 각 인자에서 하나씩 결합해 “A 1”, “B 2”형태로 결합되고 이것도 벡터 형태를 띄기 때문에 collapse로 인해 하나로 결합되어 “A 1/B 2”로 출력된다.

3 + (10 \* ( 2 + 3 ) / 5 )%% (d-1) : 맨 안쪽 괄호 안의 2+3부터 계산이 된 후 10\*5/5를 순서 상관없이 계산한다. 그러면 3+10%%(d-1)의 형태인데 나머지 연산자 %%가 우선순위가 더 높으므로 10%%(d-1)가 먼저 계산되어 나머지 값이 나온 후 3과 더해진다.

x = c(rep(5,100),'a',30:50); x : x라는 변수에 rep로 5를 100번 생성한 벡터와 ‘a’값과 30부터 50까지의 정수가 x라는 변수에 벡터 형태로 저장된다. 이 때 ‘a’가 있어서 문자형이 우선 순위가 제일 높으므로 다른 모든 값이 문자형이 되어 저장된다. X의 출력은 다음과 같다.

[1] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[11] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[21] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[31] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[41] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[51] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[61] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[71] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[81] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[91] "5" "5" "5" "5" "5" "5" "5" "5" "5" "5"

[101] "a" "30" "31" "32" "33" "34" "35" "36" "37" "38"

[111] "39" "40" "41" "42" "43" "44" "45" "46" "47" "48"

[121] "49" "50"

dir.create('example2');dir() : 현재 지정된 작업 폴더에서 example2라는 폴더를 생성한다. dir()함수를 통해 현재 작업 폴더에 있는 폴더나 파일들을 원소가 문자형인 벡터 형태로 볼 수 있고 example2 폴더가 추가되어 있는 것을 알 수 있다.

x <- matrix(c(NA, 1, 3, NA, NA, 2),3,2); is.na(x) : NA, 1, 3, NA, NA, 2의 값들이3\*2행렬 형태로 x라는 변수에 저장되고 byrow=F이므로 1열부터 채워지고 2열이 채워진다. is.na는 값이 NA이면 TRUE, 아니면 FALSE를 반환하고 각 원소마다 출력되기 때문에 행렬 형태를 유지한 상태로 다음과 같이 출력된다.

[,1] [,2]

[1,] TRUE TRUE

[2,] FALSE TRUE

[3,] FALSE FALSE

l <- c(T,F,'abc'); l : l이라는 변수에 True, False, ‘abc’라는 값을 넣었다. 이 때 TRUE를 T로, FALSE를 F로 쓸 수 있고 이 값들은 출력할 때 TRUE, FALSE로 나온다. 벡터의 원소는 하나의 유형이어야 하므로 문자형의 우선 순위가 높기 때문에 “TRUE” “FALSE”, ‘abc’ 로 모두가 문자형이 되어 출력되게 된다.

y = c('ab/cd/ef~z!yk?c','a/x!b,y,z.t/s');strsplit(y,'[[:punct:]]') : y에 'ab/cd/ef~z!yk?c', 'a/x!b,y,z.t/s'의 두 개의 문자열이 벡터로 저장된다. strsplit은 문자열을 구분자로 각 문자열은 객체가 되어 리스트 형태로 출력한다. [:punct:]는 [][!"#$%&'()\*+,./:;<=>?@\^\_`{|}~-]인데 앞의 []는 대괄호 문자를 뜻하고 뒤의 것들은 대괄호 안의 특수문자 중 하나라는 뜻이다. 바깥에 대괄호를 하나 더해서 이 문자들 중 하나라는 뜻이다. 즉 구두점(특수문자)으로 문자를 구분한다는 뜻이므로 다음과 같이 출력된다.

[[1]]

[1] "ab" "cd" "ef" "z" "yk" "c"

[[2]]

[1] "a" "x" "b" "y" "z" "t" "s"

b = readline('Input two integers with comma (ex: 1,2) :');as.numeric(unlist(strsplit(b,'[,]'))) :

readline은 입력을 받는 함수이다. 실행하면 prompt로 Input two integers with comma (ex: 1,2) : 가 안내 사항처럼 뜨게 되고 다음에 값을 입력할 수 있게 된다. 양식과 같이 1,2를 입력하면 b라는 변수에 “1,2”의 문자열이 저장된다. strsplit는 문자열을 구분자로 나누는데 b를 ‘[,]’로 나눈다는 것이다. []의 의미는 대괄호 문자 중 하나가 있으면 그것으로 나눈다는 뜻이고 따라서 ,로 나눠져 b는 “1” “2”로 나눠진다. 그런데 출력 형태는 리스트기 때문에 unlist를 이용해서 벡터 형태로 만들었다. 그리고 수치형으로 바꾸어 주는 as.numeric을 이용해서 1 2 가 출력되게 된다.

paste('A','B',collapse='/') : 4번째 코드와 다르게 “A”와 “B”가 각각 인자로 들어가 있다. paste로 각 인자들이 연결되고 collapse는 각 인자가 벡터 형태이면 결합하는 parameter이기 때문에 여기서는 적용되지 않고 sep=” “가 defalut이기 때문에 “A B”로 출력된다.

as.numeric('f1.3') : 괄호 안의 값을 수치형으로 바꾸어 주는 함수이다. ‘f1.3’에는 f가 문자형이기 때문에 수치형으로 바꿀 수 없어 NA값이 출력되게 된다.

2.

(1) **코드**

select\_lotto\_number = function(number,num\_game=5){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

matching = function(real\_numbers,select\_numbers){

match\_count = sum(as.numeric(real\_numbers$win %in% select\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(real\_numbers$bonus %in% select\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=5e4

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number\_list) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

onetime\_lotto(number\_list)

**설명**

- onetime\_lotto함수를 통해 문제들의 결과값이 반환된다. onetime\_lotto의 인자로 number\_list, num\_game를 인자를 받는데 number\_list는 사용자가 정한 로또 번호의 list이고 모두 자동으로 했다면 default로 NA값을 받게 한다. num\_game은 게임 횟수를 입력받는데 default로 5000원을 구매했다고 가정하여 5로 설정하였다. 그리고 첫 줄에서 list에 입력 받은 객체 개수만큼 num\_game의 값이 변한다.(모든 코드는 함수 내에서 실행되고 이 값은 함수 내에서만 바뀌기 때문에 문제가 없다.)

- num\_list와 num\_game인자를 받아 select\_lotto\_number함수에 들어가 select\_number변수에 저장된다. select\_lotto\_number함수를 보면 첫 줄에 if문에 class를 통해 list를 잘 받았다면 if문으로 들어간다. list의 객체 원소가 NA일 경우를 생각해 for loop를 통해 검사하고 NA일 경우 45까지의 수 중 랜덤으로 6개를 설정한 벡터를 원소로 넣고 잘 들어갔다면 다음 객체의 원소를 검사한다. 애초에 number\_list가 default로 NA값을 받았다면 else로 가게 되는데 새 리스트를 만들어 num\_game만큼의 6개 번호를 자동 설정한 벡터들이 들어가게 된다.

- 이렇게 사용자 번호가 지정되고 나서 real\_number 변수 안에get\_lotto\_numbe함수로 6개의 로또 추첨 번호와 보너스 번호를 생성한다. 7개 숫자를 생성한 후 처음 6개는 win이라는 변수에, 마지막 1개는 bonus라는 변수에 넣고 win과 bonus를 list형태로 반환하게 한다.

- cash와 match\_count라는 빈 벡터를 만들고 게임 횟수만큼 for loop를 도는데, loop안에서는 matching함수로 list가 반환되어 cash\_match라는 변수에 저장된다. cash\_match에서 cash 객체의 원소는 cash벡터의 i번째, match\_count의 원소는 match\_count벡터의 i번째에 저장된다.

- matching함수에서는 real\_number와 select\_number를 비교하여 추첨 결과를 알 수 있다. 우선 real\_number에서 결과 번호인 win의 원소가 select\_number에 있는지를 %in%을 통해 비교한다. 그리고 match\_count라는 변수에 저장한다. as.numeric으로 수치형으로 변환 후 sum으로 맞춘 개수를 구하여 match\_count라는 맞춘 개수를 저장하는 변수에 넣는다. 그리고 맞춘 개수를 통해 조건문을 통하여 cash라는 변수에 상금을 부여한다. Match\_count가 5이면 if문을 하나 더 만들어 bonus번호가 select\_number안에 있는지를 통해서 2등인지 3등인지를 구별한다. 1,2,3,4,5등에 해당되지 않으면 cash에는 0가 부여되고 cash와 match\_count를 list를 통해 반환한다.

- 총 상금은 sum(cash)를 통해 total\_reward 객체에, 총 수입은sum(cash) – num\_game\*1000을 통해 total\_income 객체에, 맞춘 개수는 match\_count라는 객체에 저장하여 리스트 형태로 출력되게 한다.

- 실행하면 다음과 같이 나오게 된다.

> number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

> set.seed(12151487)

> x = select\_lotto\_number(number\_list,5)

> y = get\_lotto\_number()

> x

[[1]]

[1] 34 6 24 12 7 8

[[2]]

[1] 5 34 43 19 12 15

[[3]]

[1] 45 22 3 13 16 34

[[4]]

[1] 26 17 13 16 21 8

[[5]]

[1] 3 29 18 5 26 31

- 6개 숫자를 입력했을 때 그대로 나오고 NA일 경우 자동 설정된다.

> y

$win

[1] 24 5 13 25 18 27

$bonus

[1] 12

- 당첨 번호 6개와 보너스 번호 1개로 나뉜다.

> matching(y, x[[1]])

$cash

[1] 0

$match\_count

[1] 1

- x의 리스트 첫 번째 객체 원소를 보고 비교하면 한 개만 맞은 것을 알 수 있다.

> matching(y, c(5,18,27,25,24,12))

$cash

[1] 6e+07

$match\_count

[1] 5

> matching(y, c(5,18,27,25,24,13))

$cash

[1] 2e+09

$match\_count

[1] 6

> matching(y, c(5,18,27,25,24,1))

$cash

[1] 1500000

$match\_count

[1] 5

- 다 맞았을 때, 5개를 맞추고 보너스 번호를 맞췄을 때, 5개 번호만 맞췄을 때 상금과 맞힌 개수가 제대로 나온 것을 알 수 있다.

> onetime\_lotto(num\_game = 5)

$total\_reward

[1] 5000

$total\_income

[1] 0

$match\_count

[1] 0 1 0 3 1

- 게임 횟수만 설정했을 때 숫자는 자동으로 나오고 위의 실행된 함수들이 수행되면서 총상금, 총수입, 맞힌 개수를 return하게 된다. 위의 경우 5000원을 땄지만 로또를 하는데 5000원을 사용했으므로 총 수입은 0이다.

(2)

set.seed(12151487)

total\_income = 0

each\_income = vector()

for (i in 1:as.integer(365\*20/7)){ # 20년 동안 일주일에 한 번씩 로또를 함

onetime = onetime\_lotto(num\_game = 5) # 5000원씩 로또를 함

total\_income = total\_income + onetime$total\_income # 매주 수입을 더함

each\_income[i] = onetime$total\_income # 매주 수입을 저장

}

cat("Total Income : ", total\_income)

each\_income

출력 결과

Total Income : -4025000

[1] -5000 -5000 45000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 50000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 0 -5000 -5000 -5000

[26] -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0

[51] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 0 -5000 -5000 -5000 -5000 -5000 0

[76] -5000 -5000 -5000 -5000 0 0 -5000 -5000 -5000 -5000 0 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000

[101] -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 0 -5000

[126] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 45000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 5000 -5000 0 -5000

[151] -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 0 -5000 -5000 0 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000

[176] -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 0 0 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000

[201] -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000

[226] -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000

[251] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0

[276] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 5000 45000 0 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000

[301] -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[326] -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[351] -5000 -5000 -5000 -5000 -5000 -5000 45000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[376] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 0 -5000

[401] -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 45000 -5000 -5000 -5000 0

[426] 0 -5000 0 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 0 -5000

[451] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[476] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 0 -5000 -5000 -5000 -5000

[501] -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 45000 -5000

[526] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 0 -5000 -5000 -5000 -5000

[551] -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000

[576] -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 0 -5000 -5000

[601] 0 0 0 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[626] -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000

[651] -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[676] -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0

[701] -5000 50000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[726] -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[751] -5000 5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 45000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000

[776] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 0 -5000

[801] -5000 -5000 -5000 10000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 45000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[826] 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0

[851] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[876] -5000 -5000 -5000 -5000 -5000 5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 45000 -5000

[901] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000

[926] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[951] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000

[976] -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 -5000 0 0 -5000 -5000 -5000 0 -5000 -5000 -5000 -5000 -5000 0 -5000

(History)

select\_lotto\_number = function(number = NA){

if(number == NA) number = sample(1:45,6)

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

x,y = get\_lotto\_number()

x = get\_lotto\_number()

x

x = get\_lotto\_number()[1]

x

x = get\_lotto\_number()$win

x

y = select\_lotto\_number(c(4,7,2,9,35,4))

select\_lotto\_number = function(number = NA){

if(number == NA) number = sample(1:45,6)

return(number)

}

y = select\_lotto\_number(c(4,7,2,9,35,4))

y = select\_lotto\_number(number = c(4,7,2,9,35,4))

c(4,7,2,9,35,4) == NA

all(number == NA)

select\_lotto\_number = function(number = NA){

if(all(number == NA)) number = sample(1:45,6)

return(number)

}

y = select\_lotto\_number(number = c(4,7,2,9,35,4))

select\_lotto\_number = function(number = NA){

if(any(number == NA)) number = sample(1:45,6)

return(number)

}

y = select\_lotto\_number(number = c(4,7,2,9,35,4))

c(4,7,2,9,35,4)==NA

any(c(4,7,2,9,35,4)==NA)

all(c(4,7,2,9,35,4)==NA)

is.na(c(4,7,2,9,35,4))

any(is.na(c(4,7,2,9,35,4)))

all(is.na(c(4,7,2,9,35,4)))

select\_lotto\_number = function(number = NA){

if(all(is.na(number))) number = sample(1:45,6)

return(number)

}

y = select\_lotto\_number(number = c(4,7,2,9,35,4))

y = select\_lotto\_number()

y

x %in y

x %in% y

x = get\_lotto\_number()$win

x

y = select\_lotto\_number()

y

x %in% y

x = get\_lotto\_number()$win

y = select\_lotto\_number()

x %in% y

x

y

as.numeric(x %in% y)

x %in% y

365\*20/7

as.integer(365\*20/7)

x

y

matching = function(x,y){

match\_count = sum(as.numeric(x$win %in% y))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(x$bonus %in% y) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=2e9

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

matching(x,y)

x = get\_lotto\_number()

y = select\_lotto\_number()

x

y

matching = function(x,y){

match\_count = sum(as.numeric(x$win %in% y))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(x$bonus %in% y) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=2e9

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

matching(x,y)

matching(c(34,6,27,4,60,3),c(34,27,6,4,60,3))

matching(list(win = c(34,6,27,4,60,3), bonus=5,c(34,27,6,4,60,3)))

matching(list(win = c(34,6,27,4,60,3), bonus=5),c(34,27,6,4,60,3))

matching(list(win = c(34,15,27,4,60,3), bonus=6),c(34,27,6,4,60,3))

matching(list(win = c(34,15,27,4,60,3), bonus=7),c(34,27,6,4,60,3))

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

len(number\_list)

length(number\_list)

num\_game = length(number\_list)

number\_list

select\_lotto\_number = function(number = NA,num\_game){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

return(number)

}

select\_lotto\_number(number\_list, num\_game)

select\_number = select\_lotto\_number(number\_list, num\_game)

y = select\_lotto\_number()

y = select\_lotto\_number(5)

y = select\_lotto\_number(num\_game = 5)

number = NA;class(number)

class(number)

class(number\_list)

number[[i]] = sample(1:45,6)

select\_lotto\_number = function(number = NA,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

number[[i]] = sample(1:45,6)

}

return(number)

}

y = select\_lotto\_number(num\_game = 5)

select\_lotto\_number = function(number = NA,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

y = select\_lotto\_number(num\_game = 5)

y

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

select\_number = select\_lotto\_number(number\_list, num\_game)

select\_number

onetime\_lotto = function(number\_list, num\_game = 5){

if(class(number) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(select\_number[[i]],real\_numbers)

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(select\_number[[i]],real\_numbers)

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

matching = function(select\_numbers,real\_numbers){

match\_count = sum(as.numeric(select\_numbers$win %in% real\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(select\_numbers$bonus %in% real\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=2e9

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(select\_number[[i]],real\_numbers)

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

onetime\_lotto(number\_list)

rm(list = ls())

select\_lotto\_number = function(number = NA,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

matching = function(select\_numbers,real\_numbers){

match\_count = sum(as.numeric(select\_numbers$win %in% real\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(select\_numbers$bonus %in% real\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=2e9

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

select\_lotto\_number = function(number,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

onetime\_lotto(number\_list)

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number\_list) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

onetime\_lotto(number\_list)

rm(list = ls())

select\_lotto\_number = function(number,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

matching = function(select\_numbers,real\_numbers){

match\_count = sum(as.numeric(select\_numbers$win %in% real\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(select\_numbers$bonus %in% real\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=2e9

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number\_list) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

onetime\_lotto(number\_list)

set.seed(12151487)

total\_income = 0

each\_income = vector()

for (i in 1:as.integer(365\*20/7)){

onetime = onetime\_lotto(num\_game = 5)

total\_income = total\_income + onetime$total\_income

each\_income[i] = onetime$total\_income

}

cat("Total Income : ", total\_income)

each\_income

rm(list = ls())

select\_lotto\_number = function(number,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

matching = function(select\_numbers,real\_numbers){

match\_count = sum(as.numeric(select\_numbers$win %in% real\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(select\_numbers$bonus %in% real\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=5e4

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number\_list) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

onetime\_lotto(number\_list)

set.seed(12151487)

total\_income = 0

each\_income = vector()

for (i in 1:as.integer(365\*20/7)){

onetime = onetime\_lotto(num\_game = 5)

total\_income = total\_income + onetime$total\_income

each\_income[i] = onetime$total\_income

}

cat("Total Income : ", total\_income)

each\_income

set.seed(12151487)

total\_income = 0

each\_income = vector()

for (i in 1:as.integer(365\*20/7)){

onetime = onetime\_lotto(num\_game = 5)

total\_income = total\_income + onetime$total\_income

each\_income[i] = onetime$total\_income

}

cat("Total Income : ", total\_income)

select\_lotto\_number = function(number,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

matching = function(real\_numbers,select\_numbers){

match\_count = sum(as.numeric(real\_numbers$win %in% select\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(real\_numbers$bonus %in% select\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=5e4

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number\_list) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

onetime\_lotto(num\_game = 5)

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA)

onetime\_lotto(number\_list)

rm(list = ls())

select\_lotto\_number = function(number,num\_game){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

matching = function(real\_numbers,select\_numbers){

match\_count = sum(as.numeric(real\_numbers$win %in% select\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(real\_numbers$bonus %in% select\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=5e4

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number\_list) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

set.seed(12151487)

x = select\_lotto\_number(number\_list,5)

y = get\_lotto\_number()

matching(y, x)

x

y

set.seed(12151487)

x = select\_lotto\_number(number\_list,5)

y = get\_lotto\_number()

x

y

matching(y, x)

matching(y, x[[1]])

matching(y, c(5,18,27,25,24,12))

matching(y, c(5,18,27,25,24,13))

matching(y, c(5,18,27,25,24,1))

set.seed(12151487)

onetime\_lotto(num\_game = 5)

x = select\_lotto\_number(number\_list,5)

y = get\_lotto\_number()

matching(y, c(5,18,27,25,24,1))

onetime\_lotto(num\_game = 5)

rm(list = ls())

select\_lotto\_number = function(number,num\_game=5){

if(class(number) == "list"){

for(i in 1:num\_game){

if(all(is.na(number[[i]]))) number[[i]] = sample(1:45,6)

}

}else{

number = list()

for(i in 1:num\_game){

number[[i]] = sample(1:45,6)

}

}

return(number)

}

get\_lotto\_number = function(){

number = sample(1:45,7)

win = number[1:6]

bonus = number[7]

return(list(win=win, bonus = bonus))

}

matching = function(real\_numbers,select\_numbers){

match\_count = sum(as.numeric(real\_numbers$win %in% select\_numbers))

cash = 0

if(match\_count==6) cash=2e9

else if(match\_count==5){

if(real\_numbers$bonus %in% select\_numbers) cash=6e7 else cash=1.5e6

}

else if(match\_count==4) cash=5e4

else if(match\_count==3) cash=5e3

else cash=0

return(list(cash=cash, match\_count = match\_count))

}

onetime\_lotto = function(number\_list = NA, num\_game = 5){

if(class(number\_list) == "list") num\_game = length(number\_list)

select\_number = select\_lotto\_number(number\_list, num\_game)

real\_numbers = get\_lotto\_number()

cash = vector()

match\_count = vector()

for(i in 1:num\_game){

cash\_match = matching(real\_numbers,select\_number[[i]])

cash[i] = cash\_match$cash

match\_count[i] = cash\_match$match\_count

}

return(list(total\_reward = sum(cash),total\_income=sum(cash) - num\_game\*1000, match\_count = match\_count))

}

onetime\_lotto(num\_game = 5)

number\_list = list(c(34,6,24,12,7,8),NA,c(45,22,3,13,16,34),NA,NA)

onetime\_lotto(number\_list)

set.seed(12151487)

x = select\_lotto\_number(number\_list)

y = get\_lotto\_number()

matching(y, X[[1]])

matching(y, x[[1]])

matching(y, c(5,18,27,25,24,1))

matching(y, c(5,18,27,25,24,13))

matching(y, c(5,18,27,25,24,12))

onetime\_lotto(num\_game = 5)