

客戶請你設計一個網路抽獎遊戲的機率規則，要滿足下列要求：

- A. 頭獎整體中獎率大約 5%，貳獎整體中獎率大約 10%
- B. 頭獎和貳獎不能同時中獎
- C. 不要讓參與遊戲的人，在最初的 5 次抽獎都完全沒中獎
- D. 盡量不要連續抽中獎，但也不可以完全沒機會連續抽中獎
- E. 不要讓參與遊戲的人，連續參加 20 次以上卻完全沒有中過獎

綜合 A.B.兩點，可以先設計是否中獎機制，設定平均 15%的中獎機率，若為中獎，再來決定為頭獎(1/3)還是貳獎(2/3)。

再來，因為 C.的緣故，所以如果單看前五次，平均中獎機率將會是 20%，怎麼樣都不會符合 A.B.兩點綜合所述之 15%中獎機率。因此本該微調前五次抽獎之後的中獎機率，來平衡平均中獎機率，但假定玩家會一直玩到無窮時間，前面五次平均中獎機率為 20%則可以忽略；後面平均中獎機率仍以控制在 15%為目標。

設計一個馬可夫鏈，一共有 20 態(state)：中獎、連續一次沒中、連續兩次沒中...、連續十九次沒中。設定初始中獎機率為 x ，若第一次沒中獎，第二次中獎機率提高為 $x+y$ ；若連續兩次沒中獎，第三次中獎機率提高為 $x+2y$...；若連續 18 次沒中獎，第 19 次中獎機率提高為 $x+18y$ ；若連續 19 次沒中獎，第 20 次中獎機率為 100%。若上次中獎，則下次中獎機率為 0.001。

轉移矩陣表示成下圖：

	中	1沒	2沒	...	19沒
中	0.001	0.999	0	...	0
1沒	$x+y$	0	$1-x-y$...	0
2沒	$x+2y$	0	0	$1-x-2y$...
...					
19沒	1	0	0

此馬可夫鏈為 irreducible、沒有週期性，因此是有穩定態的。因為矩陣偏複雜，這邊使用暴力法解出穩定態。配合程式的數值解，就能夠找出可能的 x, y 使得穩定態之中，中獎態的機率為 0.15。

這邊固定 $x = 0.1$ ，解出 $y = 0.01646118$ 。

透過上述數值，我們可以歸納出抽獎設計結構：

0. $x = 0.1$, $y = 0.01646118$ 。

1. 1~5 中抽出一個數字(均勻機率)，用來表示前五次中哪次要中獎。

2. 每次抽獎都記錄：

甲、連續沒中幾次了

乙、這次抽獎有沒有中獎

3. 根據 2.紀錄，來決定這次抽獎中獎機率

甲、連續沒中 i 次，那這次中獎機率就是 $x+i*y$ ；若連續沒中 19 次了，這次就 100%中獎。

乙、若上次有中獎，則這次中獎的機率調低為 0.001

4. 若中獎了，則挑 1/3 的機率為頭獎、2/3 的機率為貳獎。

我也使用 R 語言做了模擬，模擬 1000 個人，每人玩 1000 次的結果如下：

整體平均頭獎率：0.050066；貳獎率：0.099709

當中，每個人前五次都有中獎、沒中獎不會連續超過 20 次、連續中獎發生 220 次，220/總中獎次數約為 0.001。

當然，現實情況不會那麼完美，可能會有一些聰明人瞭解到前五次有一次必中，因而只玩五次，這樣可能造成期望值偏高的現象，或許可以設定前期貳獎機率較高、後期頭獎機率較高等機制，但這不在我這次想討論的範圍裡。

#1. 設計機率

```
win_prob = function(y){  
  x=.1  
  M = c(0.001,0.999,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)  
  for(i in 1:18){  
    temp_array = numeric(20)  
    temp_array[1] = x+i*y  
    temp_array[i+2] = 1-temp_array[1]  
    M = rbind(M,c(temp_array))  
  }  
  M = rbind(M,c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0))  
  for(i in 1:20){  
    M = M**M  
  }  
  return(M[1,1]-.15)  
}
```

```
uniroot(win_prob,c(0,.05),tol=0.001)
```

#2. 跑模擬

```
set.seed(100)
whether_price = function(){
  return(ifelse(runif(1)<(1/3),1,2))
}
Num_lottery = 1000
Num_people = 1000
big_win_list = matrix(0,ncol = Num_lottery,nrow = Num_people)
for(people in 1:Num_people){
  ind_first_five_win = sample(1:5,1)
  num_loss = 0
  whether_win_last_time = F
  win_list = numeric(Num_lottery)
  prob_x = .1
  prob_y = 0.01646118
  for(lottery in 1:Num_lottery){
    if(lottery == ind_first_five_win){
      win_list[lottery] = whether_price();whether_win_last_time = T;num_loss=0
    }else if(num_loss==19){
      win_list[lottery] = whether_price();whether_win_last_time = T;num_loss=0
    }else if((!whether_win_last_time)&&(runif(1)<(prob_x+num_loss*prob_y))){
      win_list[lottery] = whether_price();whether_win_last_time = T;num_loss=0
    }else if((whether_win_last_time)&&(runif(1)<0.001)){
      win_list[lottery] = whether_price();whether_win_last_time = T;num_loss=0
    }else{
      num_loss = num_loss+1
      whether_win_last_time = F
    }
    #print(c(lottery,num_loss))
  }
  #print(mean(win_list))
  big_win_list[people,] = win_list
}
#平均頭獎
```

```
print(length(which(big_win_list==1))/(Num_lottery*Num_people))
#平均貳獎
print(length(which(big_win_list==2))/(Num_lottery*Num_people))
#大家前五次都有中獎嗎
for(people in 1:Num_people){
  if(sum(big_win_list[people,1:5])<=0){
    print('有人前五次沒中')
  }
}
#有沒有連續 20 次沒中獎的
for(people in 1:Num_people){
  temp_ind_win = which(big_win_list[people,]!=0)
  if(length(which(temp_ind_win[-1] - temp_ind_win[-length(temp_ind_win)]>21))!=0){
    print('有人連續 20 次沒中')
    print(people)
  }
}
#連續中獎比例
tt = 0
for(people in 1:Num_people){
  temp_ind_win = which(big_win_list[people,]!=0)
  if(length(which(temp_ind_win[-1] - temp_ind_win[-length(temp_ind_win)]==1))!=0){
    #print('有人連續中獎')
    tt = tt+1
  }
}
print(tt/(length(which(big_win_list!=0))))
```