Take home final 106207338 張家寧

whoWinsGameFn 可以模擬一場 game 雙方的得分情況。代入在 Roger 得分機率(pRogerPoint)為 0.6 的情況下，重複抽選出每一球贏的人，直到雙方或其中一人的得分至少為 winningScore 四分，且贏家須超過對手(winBy)至少兩分以上，才算贏得一場 game，並在結束後判斷出贏家以及列出他們的 runningScore。

```
> whoWinsGameFn(pRogerPoint = 0.6, winBy = 2, winningScore = 4)
$runningScore
  pointHistory a r
1            r 0 1
2            a 1 1
3            r 1 2
4            a 2 2
5            a 3 2
6            a 4 2

$Winner
[1] "Andy"
```

上圖是用 whoWinsGameFn 做出的結果，可以看到 Andy 的比數至少有四分，而且 Andy 贏過 Roger 兩分時，game 才結束，並且顯示贏家為 Andy。

```
> rogerWinsGameFn(pRogerPoint = 0.6)
  AndyPoints RogerPoints RogerWins
1          4           2     FALSE
```

上圖用 rogerWinsGameFn 模擬一次 game 的結果，若贏家為 Roger，RogerWins 會顯示 TRUE，反之則顯示 FALSE。

A. pRogerGame

```
> #A. pRogerGame
> gameResult <- table(replicate(1000, rogerWinsGameFn(pRogerPoint = 0.6)
$RogerWins))
> gameResult

FALSE  TRUE
  288   712
> pRogerGame <- gameResult[2] / 1000
> pRogerGame
 TRUE
0.712
```

重複模擬 1000 次後得出 Roger 平均贏 game 的機率約為 71.2%。

B. pRogerSet

```
> rogerWinsSetFn(pRogerGame = 0.712)
  AndyGames RogerGames RogerWins
1         2          6      TRUE
```

上圖是用 rogerWinsSetFn 模擬一次 set 的結果。代入 A 題算出的 Roger 贏 game 的機率(pRogerGame)為 0.712 的情況下，重複抽選出贏得 game 的人，直到雙方或其中一人至少贏得 6 場 games，且贏家須超過對手至少 2 場 games，才算贏一次 set，最後若贏家為 Roger，RogerWins 會顯示 TRUE，反之則顯示 FALSE。

```
> setResult <- table(replicate(1000, rogerWinsSetFn(pRogerGame = 0.712)$RogerWins))
> pRogerSet <- setResult[2] / 1000
> pRogerSet
 TRUE
0.953
```

重複模擬 1000 次後得出 Roger 平均贏 set 的機率約為 95.3%。

C. pRogerMatch

```
> rogerWinsMatchFn(pRogerSet = 0.953)
  AndySets RogerSets RogerWins
1        0         3      TRUE
```

上圖是用 rogerWinsMatchFn 模擬一次 match 的結果。代入 B 題算出的 Roger 贏 set 的機率(pRogerSet)為 0.953 的情況下，重複抽選出贏得 set 的人，直到其中一人先勝出 3 sets 就贏得這場 match，最後若贏家為 Roger，RogerWins 會顯示 TRUE，反之則顯示 FALSE。

```
> matchResult <- table(replicate(1000, rogerWinsMatchFn(pRogerSet = 0.953)$RogerWins))
> pRogerMatch <- matchResult[2] / 1000
> pRogerMatch
 TRUE
0.998
```

重複模擬 1000 次後得出 Roger 平均贏 match 的機率約為 99.8%。

附錄：R 程式碼

```
runningScoreFn = function(pointHistory){
  playerUni = sort(unique(pointHistory))
  cols = sapply(playerUni, function(thisPlayer){
    cumsum(pointHistory == thisPlayer)
  })
  names(cols) = playerUni
  cbind(pointHistory, as.data.frame(cols))
}
```

```
pRogerPoint = 0.6
whoWinsGameFn = function(pRogerPoint, winBy, winningScore){
   pointHistory = c()
   i = 1
   repeat{
      AndyPoints = 0
      RogerPoints = 0
      pointHistory[i] <- sample(c('r', 'a'), 1, replace = T, prob = c(pRogerPoint, 1 -
pRogerPoint))
      i <- i + 1
      for(i in c(1:i)){
         ifelse(pointHistory[i] == 'a', AndyPoints <- AndyPoints + 1, RogerPoints <-
RogerPoints + 1)
      }
      if(AndyPoints >= winningScore & abs(AndyPoints - RogerPoints) >= winBy) break
      if(RogerPoints >= winningScore & abs(AndyPoints - RogerPoints) >= winBy)
break
   }
   winner = ''
   ifelse(AndyPoints > RogerPoints, winner <- 'Andy', winner <- 'Roger')
   list(runningScore = runningScoreFn(pointHistory), Winner = winner)
}
whoWinsGameFn(pRogerPoint = 0.6, winBy = 2, winningScore = 4)

rogerWinsGameFn = function(pRogerPoint){
   pointHistory = c()
   i = 1
   repeat{
      AndyPoints = 0
      RogerPoints = 0
      pointHistory[i] <- sample(c('r', 'a'), 1, replace = T, prob = c(pRogerPoint, 1 -
pRogerPoint))
      i <- i + 1
      for(i in c(1:i)){
         ifelse(pointHistory[i] == 'a', AndyPoints <- AndyPoints + 1, RogerPoints <-
RogerPoints + 1)
      }
      if(AndyPoints >= 4 & abs(AndyPoints - RogerPoints) >= 2) break
```

```
    if(RogerPoints >= 4 & abs(AndyPoints - RogerPoints) >= 2) break
  }
  winner = ''
  ifelse(AndyPoints > RogerPoints, winner <- 'Andy', winner <- 'Roger')
  RogerWins <- winner == 'Roger'
  data.frame(AndyPoints, RogerPoints, RogerWins)
}
rogerWinsGameFn(pRogerPoint = 0.6)

#A. pRogerGame
gameResult <- table(replicate(1000, rogerWinsGameFn(pRogerPoint =
0.6)$RogerWins))
pRogerGame <- gameResult[2] / 1000
pRogerGame

rogerWinsSetFn = function(pRogerGame){
  gameHistory = c()
  i = 1
  repeat{
    AndyGames = 0
    RogerGames = 0
    gameHistory[i] <- sample(c('r', 'a'), 1, replace = T, prob = c(pRogerGame, 1 -
pRogerGame))
    i <- i + 1
    for(i in c(1:i)){
      ifelse(gameHistory[i] == 'a', AndyGames <- AndyGames + 1, RogerGames <-
RogerGames + 1)
    }
    if(AndyGames >= 6 & abs(AndyGames - RogerGames) >= 2) break
    if(RogerGames >= 6 & abs(AndyGames - RogerGames) >= 2) break
  }
  winner = ''
  ifelse(AndyGames > RogerGames, winner <- 'Andy', winner <- 'Roger')
  RogerWins <- winner == 'Roger'
  data.frame(AndyGames, RogerGames, RogerWins)
}
rogerWinsSetFn(pRogerGame = 0.712)
```

```
#B. pRogerSet
setResult <- table(replicate(1000, rogerWinsSetFn(pRogerGame =
0.712)$RogerWins))
pRogerSet <- setResult[2] / 1000
pRogerSet

rogerWinsMatchFn = function(pRogerSet){
   setHistory = c()
   i = 1
   repeat{
      AndySets = 0
      RogerSets = 0
      setHistory[i] <- sample(c('r', 'a'), 1, replace = T, prob = c(pRogerSet, 1 -
pRogerSet))
      i <- i + 1
      for(i in c(1:i)){
         ifelse(setHistory[i] == 'a', AndySets <- AndySets + 1, RogerSets <- RogerSets +
1)
      }
      if(RogerSets == 3 | AndySets == 3) break
   }
   winner = ''
   ifelse(AndySets > RogerSets, winner <- 'Andy', winner <- 'Roger')
   RogerWins <- winner == 'Roger'
   data.frame(AndySets, RogerSets, RogerWins)
}
rogerWinsMatchFn(pRogerSet = 0.953)

#C. pRogerMatch
matchResult <- table(replicate(1000, rogerWinsMatchFn(pRogerSet =
0.953)$RogerWins))
pRogerMatch <- matchResult[2] / 1000
pRogerMatch
```