

# 111208094\_hw2

2024-10-30

1.籃球投籃命中率模擬 設計一個名為 `simple_shooting_simulation()` 的 R function，模擬一位球員進行多次投籃，並根據投籃命中率計算實際命中次數和命中率。規則：+ 投籃次數：模擬中，球員會進行指定次數的投籃。

命中率：玩家可設定球員的投籃命中率（例如0.75），系統會隨機生成投籃結果（命中或未中）。結果分析：模擬完成後，函數應返回實際命中次數和命中率。

輸入參數：`num_shots`：投籃的總次數（例如，100 次）。`hit_rate`：投籃命中率（例如，0.75 表示 75% 的命中率）。

輸出該函數應返回一個列表，包含以下內容：`total_hits`：命中的總次數。`actual_hit_rate`：實際命中率（命中次數 / 投籃次數）。

```
#1.
#寫function!!
simple_shooting_simulation <- function(num_shots, hit_rate){
  total_hits <- rbinom(1, size = num_shots, prob = hit_rate)
  actual_hit_rate <- total_hits/num_shots
  hitted <- list(total_hits=total_hits,actual_hit_rate=actual_hit_rate)
  return(hitted)
}
simple_shooting_simulation(987654321,0.66)
```

```
## $total_hits
## [1] 651829063
##
## $actual_hit_rate
## [1] 0.6599769
```

```
#檢查
set.seed(123)
results <- simple_shooting_simulation(10000, 0.75)
cat(paste0("命中的總次數:",results[1],"\n","實際命中率:",results[2]))
```

```
## 命中的總次數:7537
## 實際命中率:0.7537
```

2.模擬籃球系列賽 撰寫一個名為 `basketball_series()` 的函數，模擬兩隊進行 7 場系列賽。每場比賽得分隨機產生（每隊的得分在 80 到 120 之間），系列賽中首先贏得 4 場的隊伍為勝者。函數應顯示每場的得分及系列賽最終結果。

輸入參數：`team1_name`：隊伍 1 的名稱（預設為“Team 1”）`team2_name`：隊伍 2 的名稱（預設為“Team 2”）

輸出：每場比賽的得分及最終結果

題目拆解、可能需要的工具:D 1. 撰寫一個名為 `basketball_series()` 的函數：自訂函數 `function` 2. 模擬兩隊進行 7 場系列賽：迴圈，我習慣用 `for` 3. 每場比賽得分隨機產生（每隊的得分在 80 到 120 之間）：`sample` 函數 4. 系列賽中首先贏得 4 場的隊伍為勝者：`if` 函數做判斷、`break` 掉迴圈 5. 函數應顯示每場的得分及系列賽最終結果：`return` 函數 6. 最終結果圖示（助教提示）：可知需要的變數有第一隊比賽得分、第二隊比賽得分、第一隊贏的次數、第二隊贏的次數，跟一個 `dataframe` 拿來放七次的比賽得分

```
#2.
#寫function!!
basketball_series <- function(team1_name = "Team 1", team2_name = "Team 2") {

  team1_wins <- 0
  team2_wins <- 0

  scores <- data.frame(Game = 1:7,
                        Team1_Score = c(0, 0, 0, 0, 0, 0, 0),
                        Team2_Score = c(0, 0, 0, 0, 0, 0, 0))
  #迴圈開始
  for (i in 1:7) {
    team1_score <- sample(80:120, 1)
    team2_score <- sample(80:120, 1)

    scores[i, "Team1_Score"] <- team1_score
    scores[i, "Team2_Score"] <- team2_score

    if (team1_score > team2_score) {
      team1_wins <- team1_wins + 1
    } else if (team2_score > team1_score) {
      team2_wins <- team2_wins + 1
    }

    if (team1_wins == 4 || team2_wins == 4) {break}

  } #迴圈結束

  if (team1_wins > team2_wins) {
    cat("最終結果:Team1獲勝")
  } else if (team2_wins > team1_wins) {
    cat("最終結果:Team2獲勝")
  } else {
    cat("平手")
  }
  return(scores)
}
```

```
# 測試
set.seed(123)
basketball_series("Team 1", "Team 2")
```

```
## 最終結果:Team1獲勝
```

```
##   Game Team1_Score Team2_Score
## 1    1         110         94
## 2    2          93         82
## 3    3         116         93
## 4    4         104        105
## 5    5         106         84
## 6    6           0           0
## 7    7           0           0
```