Assessed Worksheet 3

Omoegbemwen Aigbe 2301397

2023-11-03

# Assessed Worksheet 3

## Element 010-3

## Maths for Biosciences MOD005667

### The Problem: Manufacturing Biased Coins

You work for a company that manufactures biased coins to sell as novelty gifts. The coins are weighted to favour either heads or tails when flipped. Your task is determining how many coin flips it takes to determine if a coin is fair or biased reliably. You want to create coins with 50%, 55%, 60%, and 65% chance of getting heads. Note the 50% coin is fair.

#### Part 1: Building the coin flip simulator (10 marks)

Begin by writing an R function that simulates coin flips. The function should take as input the number of coin flips you want to simulate and the probability of getting heads. It should return a 2 X 2 matrix containing the number of heads and the number of tails recorded. You can use chatGPT to help you write your function.

simulate\_coin\_flips <- function(num\_flips, prob\_heads) {  
 # Simulate coin flips  
 coin\_flips <- sample(c("Heads", "Tails"), num\_flips, replace = TRUE, prob = c(prob\_heads, 1 - prob\_heads))  
   
 # Count the number of heads and tails  
 num\_heads <- sum(coin\_flips == "Heads")  
 num\_tails <- sum(coin\_flips == "Tails")  
   
 # Create a 2x2 matrix  
 result\_matrix <- matrix(c(num\_heads, num\_tails, num\_tails, num\_heads), nrow = 2, byrow = TRUE,  
 dimnames = list(c("Heads", "Tails"), c("Heads", "Tails")))  
   
 # Return the matrix to the user  
 return(result\_matrix)  
}

simulate\_coin\_flips(num\_flips = 10, prob\_heads = 0.5)

## Heads Tails  
## Heads 6 4  
## Tails 4 6

#### Part 2: Experimental design (30 marks)

Design a series of experiments using your function that will allow you to determine the relationship between the number of coin flips and whether you can detect if each of your coins is biased. Each experiment will consist of a single call to the coin flip simulator you wrote in Part 1 combined with your statistical evaluation of the results of that simulation. The appropriate statistical test to determine if a series of coin flips is fair is the Chi-Squared test.

Consider what you must vary in each case to determine this relationship. Also, think about how many times you will need to repeat an experiment to understand how consistent the results are when you repeat the same experiment multiple times. While designing your experiment series, I suggest you experiment with your coin flip simulator to get a feel for its behavior.

**Describe your experimental approach as clearly and succinctly as possible (no more than 100 words).** You will be marked on the clarity of your explanation.

**Solution**

**Aim: To determine the relationship between the number of coin flips and detecting if the coin is biased.**

**Hypothesis:**

**Null hypothesis: the bias of a coin cannot be detected by an increase in the number of flips.**

**Alternative hypothesis: As biases increase, fewer flips are required to show the bias of a coin.**

**Method.**

**Step 1. Create a function to simulate coin flips. Test across a range of numbers (10, 50, 100, 500, 1000, 5000, 10000)**

**Step 2. Adapt the simulation to be 55%, 60% and 65% bias and test for the same range of numbers.**

**Step 3. Repeat the simulation 100 times to get an average value of the head obtained from the flips. Repeating the simulation 100 times allows for the analysis of variance within the sample. Also, a large sample means a more reliable data set.**

**Step 4. Perform a chi-squared test on the average number of heads obtained.**

**Step 5. Analyse your result to reach a conclusion.**

#### Part 3: Data collection (25 marks)

Perform the experiments you have described in part 2 using R. Your R code should perform your experiments and store the results in a data frame. For each experiment, you should store the experimental parameters (number of flips, probability of heads) in your data frame, and the results of the Chi-squared for the simulation (Chi-squared statistic, p-value). Note that if you have written your coin flip simulator correctly, the output of the simulator will be in the correct matrix form to feed directly into the chisq.test() function in R.

**Print your data frame to your final document. If your data frame contains more than 50 rows, print the first 50 rows.**

the result of a fair coin

n\_flips <- c(rep(10, 100), rep(50, 100), rep(100, 100), rep(500, 100), rep(1000, 100), rep(5000, 100), rep(10000, 100))  
  
prob\_heads <- rep(0.5, 700)  
  
df\_p0.5<- data.frame(cbind(n\_flips, prob\_heads))  
  
get\_sim\_chi <- function(n\_flips, p\_heads) {  
 my\_sim <- simulate\_coin\_flips(n\_flips, 0.5)   
 my\_chi <- chisq.test(my\_sim, p = c(0.5, 0.5))  
 my\_p <- my\_chi$p.value  
 return(my\_p)  
}  
df\_p0.5$p\_value <- mapply(FUN = get\_sim\_chi, n\_flips, p\_heads = 0.5)  
head(df\_p0.5)

## n\_flips prob\_heads p\_value  
## 1 10 0.5 0.65472085  
## 2 10 0.5 0.17971249  
## 3 10 0.5 0.65472085  
## 4 10 0.5 0.65472085  
## 5 10 0.5 0.02534732  
## 6 10 0.5 1.00000000

the result of a 55% bias coin

n\_flips <- c(rep(10, 100), rep(50, 100), rep(100, 100), rep(500, 100), rep(1000, 100), rep(5000, 100), rep(10000, 100))  
  
prob\_heads <- rep(0.55, 700)  
  
df\_p0.55<- data.frame(cbind(n\_flips, prob\_heads))  
  
get\_sim\_chi <- function(n\_flips, p\_heads) {  
 my\_sim <- simulate\_coin\_flips(n\_flips, 0.55)   
 my\_chi <- chisq.test(my\_sim, p = c(0.5, 0.5))  
 my\_p <- my\_chi$p.value  
 return(my\_p)  
}  
df\_p0.55$p\_value <- mapply(FUN = get\_sim\_chi, n\_flips, p\_heads = 0.55)  
head(df\_p0.55)

## n\_flips prob\_heads p\_value  
## 1 10 0.55 0.02534732  
## 2 10 0.55 0.02534732  
## 3 10 0.55 0.65472085  
## 4 10 0.55 1.00000000  
## 5 10 0.55 0.65472085  
## 6 10 0.55 0.17971249

Displaying the result of a 60% bias coin

n\_flips <- c(rep(10, 100), rep(50, 100), rep(100, 100), rep(500, 100), rep(1000, 100), rep(5000, 100), rep(10000, 100))  
  
prob\_heads <- rep(0.6, 700)  
  
df\_p0.6<- data.frame(cbind(n\_flips, prob\_heads))  
  
get\_sim\_chi <- function(n\_flips, p\_heads) {  
 my\_sim <- simulate\_coin\_flips(n\_flips, 0.6)   
 my\_chi <- chisq.test(my\_sim, p = c(0.5, 0.5))  
 my\_p <- my\_chi$p.value  
 return(my\_p)  
}  
df\_p0.6$p\_value <- mapply(FUN = get\_sim\_chi, n\_flips, p\_heads = 0.6)  
head(df\_p0.6)

## n\_flips prob\_heads p\_value  
## 1 10 0.6 0.65472085  
## 2 10 0.6 0.65472085  
## 3 10 0.6 0.65472085  
## 4 10 0.6 0.65472085  
## 5 10 0.6 1.00000000  
## 6 10 0.6 0.02534732

the result of a 65% bias coin

n\_flips <- c(rep(10, 100), rep(50, 100), rep(100, 100), rep(500, 100), rep(1000, 100), rep(5000, 100), rep(10000, 100))  
  
prob\_heads <- rep(0.65, 700)  
  
df\_p0.65<- data.frame(cbind(n\_flips, prob\_heads))  
  
get\_sim\_chi <- function(n\_flips, p\_heads) {  
 my\_sim <- simulate\_coin\_flips(n\_flips, 0.65)   
 my\_chi <- chisq.test(my\_sim, p = c(0.5, 0.5))  
 my\_p <- my\_chi$p.value  
 return(my\_p)  
}  
df\_p0.65$p\_value <- mapply(FUN = get\_sim\_chi, n\_flips, p\_heads = 0.65)  
head(df\_p0.65)

## n\_flips prob\_heads p\_value  
## 1 10 0.65 0.179712495  
## 2 10 0.65 0.001745119  
## 3 10 0.65 0.654720846  
## 4 10 0.65 0.654720846  
## 5 10 0.65 0.654720846  
## 6 10 0.65 0.654720846

Combining the average result of the coins (50%, 55%, 60%, 65%)

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

sum\_df\_0.5 <- df\_p0.5 |>  
 group\_by(n\_flips, prob\_heads) |>  
 summarize(mean\_p\_value = mean(p\_value))

## `summarise()` has grouped output by 'n\_flips'. You can override using the  
## `.groups` argument.

sum\_df\_0.5

## # A tibble: 7 × 3  
## # Groups: n\_flips [7]  
## n\_flips prob\_heads mean\_p\_value  
## <dbl> <dbl> <dbl>  
## 1 10 0.5 0.555  
## 2 50 0.5 0.486  
## 3 100 0.5 0.426  
## 4 500 0.5 0.386  
## 5 1000 0.5 0.437  
## 6 5000 0.5 0.420  
## 7 10000 0.5 0.385

filtered\_df <- sum\_df\_0.5[sum\_df\_0.5$mean\_p\_value <= 0.05,]  
  
n\_flips <- filtered\_df$n\_flips  
  
cat("The n\_flips value for the row with mean p value less than or equal to 0.05 is", n\_flips, "\n")

## The n\_flips value for the row with mean p value less than or equal to 0.05 is

**The minimum number of words needed could not be determined for a fair coin.**

sum\_df\_0.55 <- df\_p0.55 |>  
 group\_by(n\_flips, prob\_heads) |>  
 summarize(mean\_p\_value = mean(p\_value))

## `summarise()` has grouped output by 'n\_flips'. You can override using the  
## `.groups` argument.

sum\_df\_0.55

## # A tibble: 7 × 3  
## # Groups: n\_flips [7]  
## n\_flips prob\_heads mean\_p\_value  
## <dbl> <dbl> <dbl>  
## 1 10 0.55 4.93e- 1  
## 2 50 0.55 4.28e- 1  
## 3 100 0.55 3.55e- 1  
## 4 500 0.55 7.75e- 2  
## 5 1000 0.55 1.41e- 2  
## 6 5000 0.55 1.45e-14  
## 7 10000 0.55 1.82e-31

filtered\_df <- sum\_df\_0.55[sum\_df\_0.55$mean\_p\_value <= 0.05,]  
  
n\_flips <- filtered\_df$n\_flips  
  
cat("The n\_flips value for the row with mean p-value less than or equal to 0.05 is", n\_flips, "\n")

## The n\_flips value for the row with mean p-value less than or equal to 0.05 is 1000 5000 10000

**The minimum number of flips required to detect a 65% biased coin is 1000.**

sum\_df\_0.6 <- df\_p0.6 |>  
 group\_by(n\_flips, prob\_heads) |>  
 summarize(mean\_p\_value = mean(p\_value))

## `summarise()` has grouped output by 'n\_flips'. You can override using the  
## `.groups` argument.

sum\_df\_0.6

## # A tibble: 7 × 3  
## # Groups: n\_flips [7]  
## n\_flips prob\_heads mean\_p\_value  
## <dbl> <dbl> <dbl>  
## 1 10 0.6 4.97e- 1  
## 2 50 0.6 2.31e- 1  
## 3 100 0.6 9.99e- 2  
## 4 500 0.6 1.14e- 4  
## 5 1000 0.6 7.96e- 6  
## 6 5000 0.6 1.76e- 59  
## 7 10000 0.6 2.38e-131

filtered\_df <- sum\_df\_0.6[sum\_df\_0.6$mean\_p\_value <= 0.05,]  
  
n\_flips <- filtered\_df$n\_flips  
  
cat("The n\_flips value for the row with mean p value less than or equal to 0.05 is", n\_flips, "\n")

## The n\_flips value for the row with mean p value less than or equal to 0.05 is 500 1000 5000 10000

**The minimum number of flips required to detect a 65% biased coin is 500.**

sum\_df\_0.65 <- df\_p0.65 |>  
 group\_by(n\_flips, prob\_heads) |>  
 summarize(mean\_p\_value = mean(p\_value))

## `summarise()` has grouped output by 'n\_flips'. You can override using the  
## `.groups` argument.

sum\_df\_0.65

## # A tibble: 7 × 3  
## # Groups: n\_flips [7]  
## n\_flips prob\_heads mean\_p\_value  
## <dbl> <dbl> <dbl>  
## 1 10 0.65 4.00e- 1  
## 2 50 0.65 9.98e- 2  
## 3 100 0.65 2.92e- 2  
## 4 500 0.65 1.04e- 11  
## 5 1000 0.65 2.53e- 29  
## 6 5000 0.65 1.53e-166  
## 7 10000 0.65 0

filtered\_df <- sum\_df\_0.65[sum\_df\_0.65$mean\_p\_value <= 0.05,]  
  
n\_flips <- filtered\_df$n\_flips  
  
cat("The n\_flips value for the row with mean p value less than or equal to 0.05 is", n\_flips, "\n")

## The n\_flips value for the row with mean p value less than or equal to 0.05 is 100 500 1000 5000 10000

**The minimum number of flips required to detect a 65% biased coin is 100.**

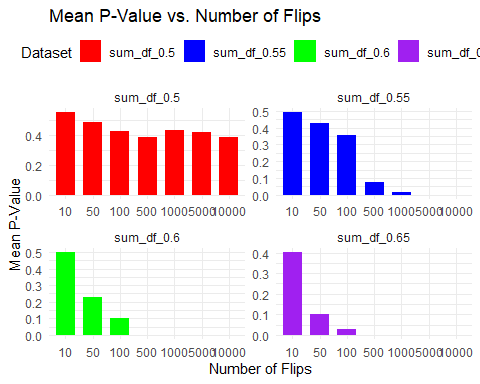
#### Part 4: Results (25 marks)

Display the results of your experiments from Part 3. Is your data best displayed as a table, graph, or both? Decide how you want to display your results and produce your display using R. There are many possible ways to display your data. A high-marking results display will favour clarity and simplicity.

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.2

# Combine the data frames into a single data frame  
all\_data <- rbind(  
 cbind(sum\_df\_0.5, dataset = "sum\_df\_0.5"),  
 cbind(sum\_df\_0.55, dataset = "sum\_df\_0.55"),  
 cbind(sum\_df\_0.6, dataset = "sum\_df\_0.6"),  
 cbind(sum\_df\_0.65, dataset = "sum\_df\_0.65")  
)  
  
# Convert the dataset column to a factor for proper grouping  
all\_data$dataset <- factor(all\_data$dataset)  
  
# Create a multiple bar chart with facets  
ggplot(all\_data, aes(x = factor(n\_flips), y = mean\_p\_value, fill = dataset)) +  
 geom\_bar(stat = "identity", position = position\_dodge(width = 0.8), width = 0.7) +  
 labs(title = "Mean P-Value vs. Number of Flips",  
 x = "Number of Flips",  
 y = "Mean P-Value",  
 fill = "Dataset") +  
 scale\_fill\_manual(values = c("sum\_df\_0.5" = "red", "sum\_df\_0.55" = "blue", "sum\_df\_0.6" = "green", "sum\_df\_0.65" = "purple")) +  
 theme\_minimal() +  
 theme(legend.position = "top") +  
 facet\_wrap(~dataset, scales = "free")



#### **Part 5: Discussion (10 marks)**

What is the relationship between the amount of bias in the coin and the number of flips required to detect that the coin is biased? Why do you think this might be? **Give your answer in 50 words or less.**

**The analysis above shows that as the bias of the coin increases, the number of flips required to detect the bias decreases. The small preferences showed that more flips were needed to see the bias.**