Assignment 2

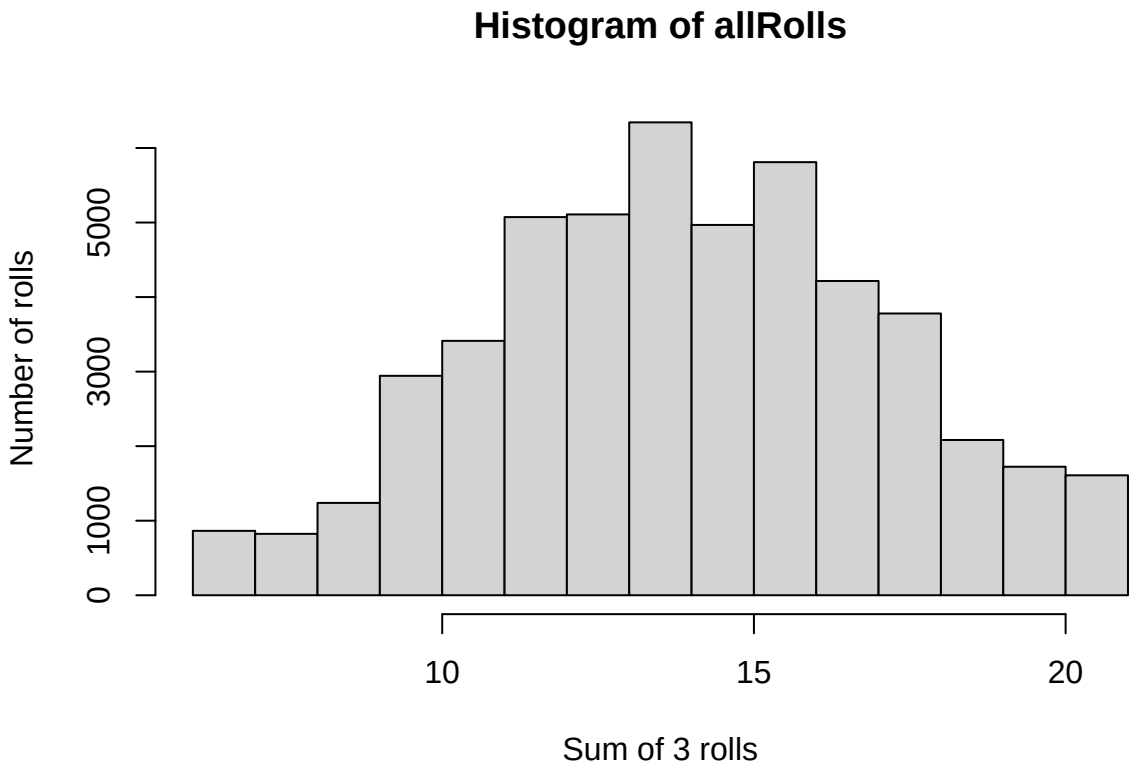
2023-08-02

R Markdown

Question 1

```
roll <- function(max_rolls, sides, num_of_dice, biasVector){
  allRolls = c()
  for(x in 1:max_rolls){
    die <- 1:8
    dice <- sample(die, size =num_of_dice , prob = biasVector)
    allRolls <- append(allRolls,sum(dice))
  }
  hist(allRolls,xlab="Sum of 3 rolls", ylab="Number of rolls")
}

roll(50000, 8, 3, c(1,1,1,1,1,1,3,1)/10)
```



Question 2

```
rescale01 <- function(x){
  rangeOfVector <- range(x, na.rm =TRUE, finite = TRUE)
  x[x == -Inf] = rangeOfVector[1]
  x[x == Inf] = rangeOfVector[2]
  for(i in 1:length(x)){
    if(!is.numeric(x[i])){
      stop("inputs must all be numeric")
    }
    x[i] <- (x[i] - rangeOfVector[1])/(rangeOfVector[2] - rangeOfVector[1])
  }
  return(x)
}
rescale01(c(Inf, 1, 2,3, -Inf))
```

```
## [1] 1.0 0.0 0.5 1.0 0.0
```

```
#to show the error message
rescale01(c('a',2,4,1,Inf))
```

```
## Error in rescale01(c("a", 2, 4, 1, Inf)): inputs must all be numeric
```

Question 3

```
commonNA <- function(x,y){
  if(length(x) != length(y)){
    stop("vectors must be the same length")
  }
  numberOfCommonNA <- 0
  for(i in 1:length(x)){
    if((is.na(x[i])) && (is.na(y[i]))){
      numberOfCommonNA <- numberOfCommonNA + 1
    }
  }
  return(numberOfCommonNA)
}
commonNA(c(NA,1,2,3,NA,5), c(NA,2,NA,4,NA,5))
```

```
## [1] 2
```

```
#To show the error message
commonNA(c(1,2,3,4),c(1,2,3))
```

```
## Error in commonNA(c(1, 2, 3, 4), c(1, 2, 3)): vectors must be the same length
```

Question 4

```
fizzbuzz <- function(x){
  if(x%%5==0 && x%%3==0){
    return("fizzbuzz")
  }
  else if(x%%5 == 0){
    return("buzz")
  }
  else if(x%%3 == 0){
    return("fizz")
  }
  return(x)
}
fizzbuzz(3)
```

```
## [1] "fizz"
```

```
fizzbuzz(5)
```

```
## [1] "buzz"
```

```
fizzbuzz(15)
```

```
## [1] "fizzbuzz"
```

```
fizzbuzz(1)
```

```
## [1] 1
```

Question 5

```
get_temp_desc <- function(temp){
  cut(temp, breaks = c(-Inf,0,10,20,30,Inf), labels= c("freezing","cold","cool","warm","hot"))

}
get_temp_desc(-1)
```

```
## [1] freezing
## Levels: freezing cold cool warm hot
```

```
get_temp_desc(10)
```

```
## [1] cold
## Levels: freezing cold cool warm hot
```

```r
get_temp_desc(12)
```

```
## [1] cool
## Levels: freezing cold cool warm hot
```

```r
get_temp_desc(23)
```

```
## [1] warm
## Levels: freezing cold cool warm hot
```

```r
get_temp_desc(33)
```

```
## [1] hot
## Levels: freezing cold cool warm hot
```