# Untitled

Phillip Hughes

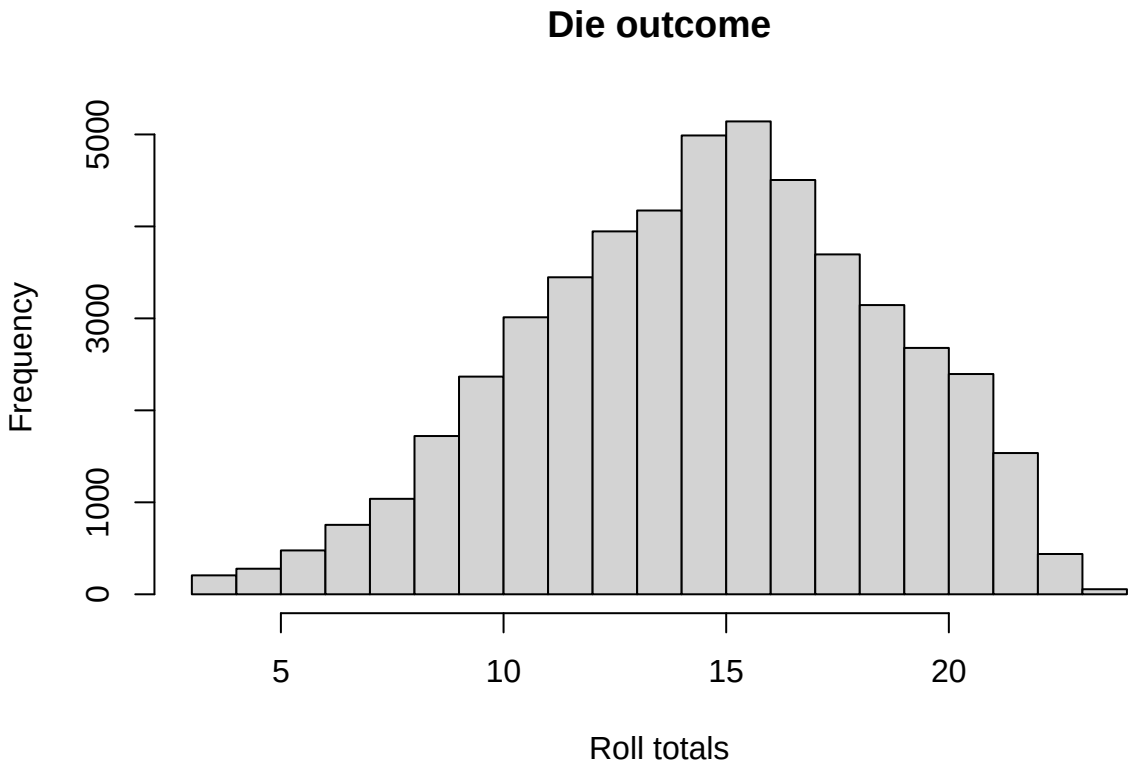
2023-04-26

## functions worksheet

### loaded dice function

```
die_graph <- function(max_rolls = 50000, sides = 8, num_of_dice = 3) {
  rolls <- matrix(sample(c(1:6, 7, 8, 7, 7), max_rolls * num_of_dice, replace = TRUE), nrow = max_rolls

  rolls_sum <- apply(rolls, 1, sum) # calculate the sum of the rolls for each trial
  hist(rolls_sum, breaks = seq(num_of_dice, num_of_dice * sides, by = 1), main = "Die outcome", xlab = 
}
die_graph()
```



### function for rescale01()

```
rescale01 <- function(x) {
  if (!is.numeric(x)) {
    stop("inputs must all be numeric")
  }
  rng <- range(x, na.rm = TRUE, finite = TRUE)
  rng <- (x - rng[1]) / (rng[2] - rng[1])
  rng[rng== Inf]<-1
  rng[rng== -Inf]<-0
  rng
}
rescale01(C('a','b'))
```

```
## Error in C("a", "b"): object not interpretable as a factor
```

```
rescale01(c(Inf, 0, 2, 10))
```

```
## [1] 1.0 0.0 0.2 1.0
```

```
rescale01(c(-Inf, 4, 2, 7))
```

```
## [1] 0.0 0.4 0.0 1.0
```

```
rescale01(c(-Inf, 0, 2, 10, Inf))
```

```
## [1] 0.0 0.0 0.2 1.0 1.0
```

### both vectors of same length where positions NA

```
samesies<- function(x,y){
  if (length(x) != length(y)){
    stop("both vectors must be the same length")
  }
  total<- sum(is.na(x) & is.na(y))
  total
}
x <- c(NA, NA, 3, NA)
y <- c(NA, 2, NA, 4)
samesies(x, y)
```

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

```
x <- c(NA, NA, 3, NA,5)
y <- c(NA, 2, NA, 4)
samesies(x, y)
```

```
## Error in samesies(x, y): both vectors must be the same length
```

### Fizzbuzz function

```
fizzbuzz<-function(x){
  if(x%%3 ==0 & x%%5==0){
    print("fizzbuzz")
  } else if(x%%3==0){
    print('fizz')
  } else if(x%%5==0){
    print("buzz")
  }else {
    print(x)
  }
}
fizzbuzz(15)
```

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

```
fizzbuzz(3)
```

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

```
fizzbuzz(5)
```

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

```
fizzbuzz(2)
```

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

##rewriting function with cut()

```
GetTempDesc <- function(Temp) {
  TempLabels <- c('Freezing', 'Cold', 'Cool', 'Warm', 'Hot')
  TempBreaks <- c(-Inf, 0, 10, 20, 30, Inf)
  TempRange <- cut(Temp, breaks = TempBreaks, labels = TempLabels)
  return(TempRange)
}
GetTempDesc(41)
```

```
## [1] Hot
## Levels: Freezing Cold Cool Warm Hot
```

```
GetTempDesc(29)
```

```
## [1] Warm
## Levels: Freezing Cold Cool Warm Hot
```

```
GetTempDesc(19)
```

```
## [1] Cool
## Levels: Freezing Cold Cool Warm Hot
```

```
GetTempDesc(09)
```

```
## [1] Cold
## Levels: Freezing Cold Cool Warm Hot
```

```
GetTempDesc(-2)
```

```
## [1] Freezing
## Levels: Freezing Cold Cool Warm Hot
```