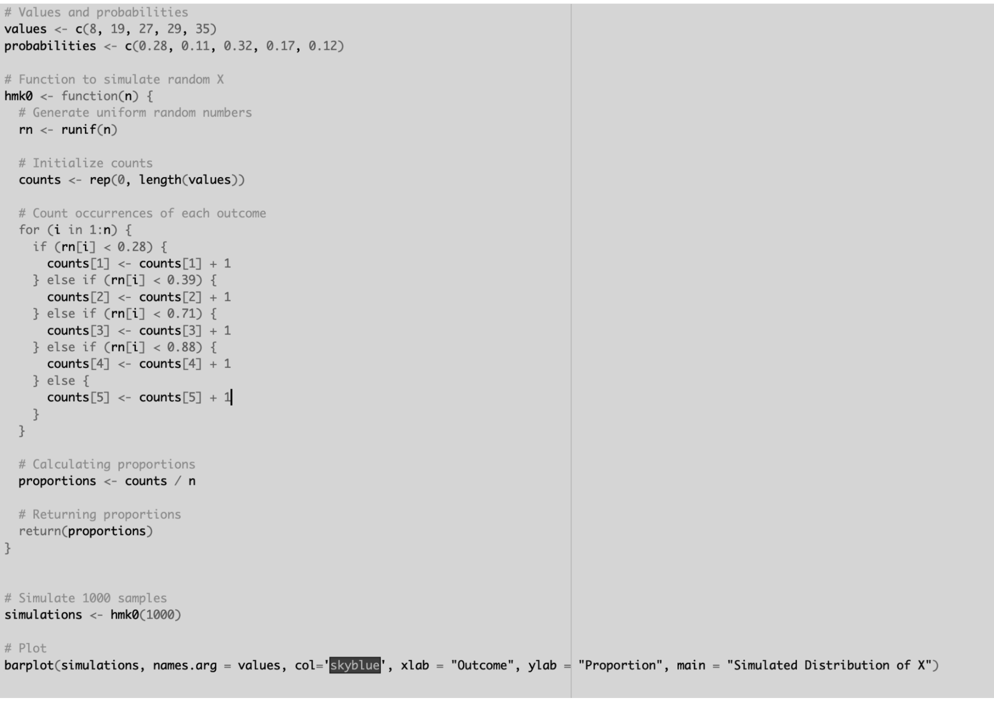
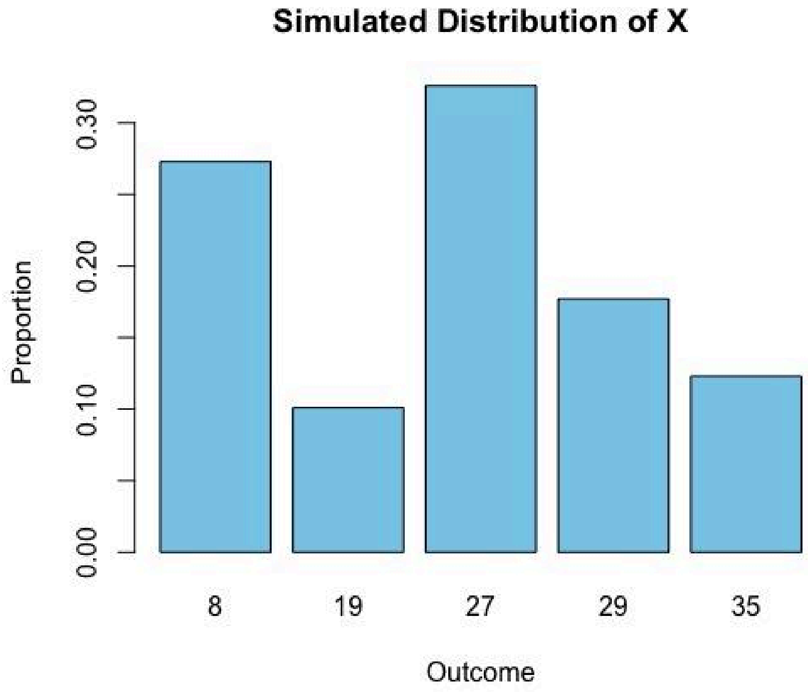
Emiliana Geronimo
Math 666: Simulations of Stochastic Systems
Homework 0:

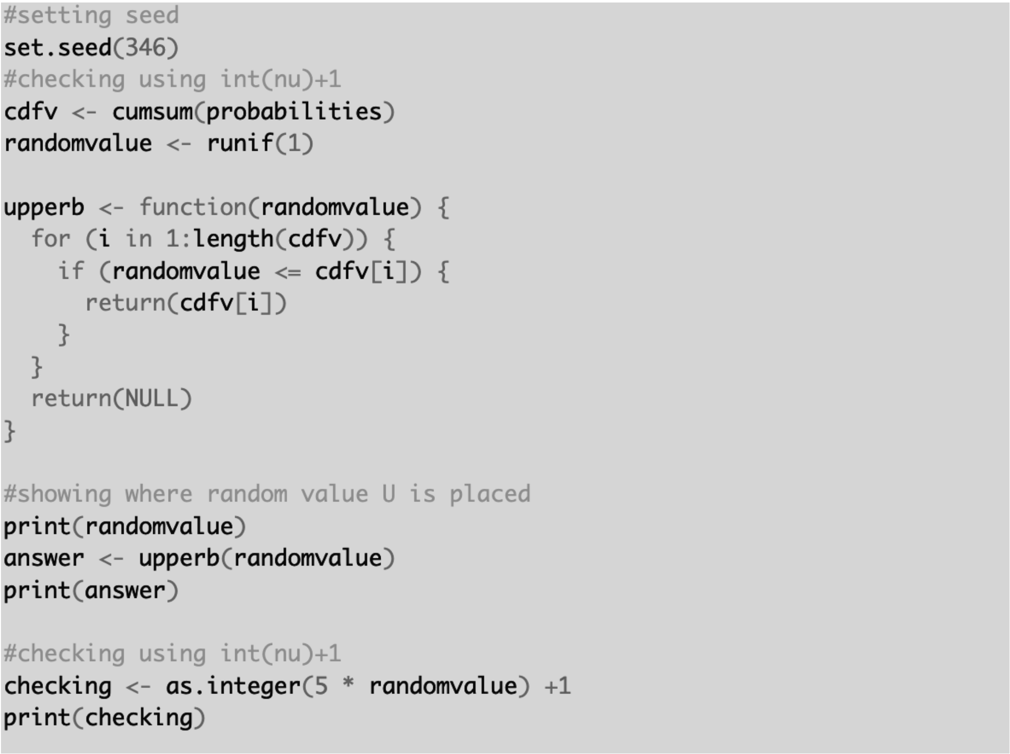
1.



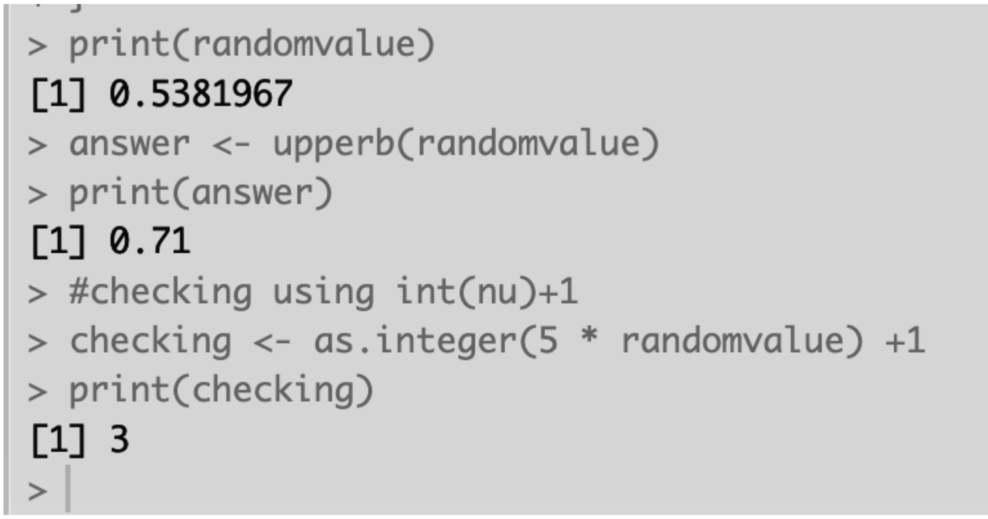
```
# Values and probabilities
values <- c(8, 19, 27, 29, 35)
probabilities <- c(0.28, 0.11, 0.32, 0.17, 0.12)

# Function to simulate random X
hmk0 <- function(n) {
  # Generate uniform random numbers
  rn <- runif(n)

  # Initialize counts
  counts <- rep(0, length(values))

  # Count occurrences of each outcome
  for (i in 1:n) {
    if (rn[i] < 0.28) {
      counts[1] <- counts[1] + 1
    } else if (rn[i] < 0.39) {
      counts[2] <- counts[2] + 1
    } else if (rn[i] < 0.71) {
      counts[3] <- counts[3] + 1
    } else if (rn[i] < 0.88) {
      counts[4] <- counts[4] + 1
    } else {
      counts[5] <- counts[5] + 1
    }
  }

  # Calculating proportions
  proportions <- counts / n

  # Returning proportions
  return(proportions)
}


# Simulate 1000 samples
simulations <- hmk0(1000)

# Plot
barplot(simulations, names.arg = values, col='skyblue', xlab = "Outcome", ylab = "Proportion", main = "Simulated Distribution of X")
```



2.



```r
#setting seed
set.seed(346)
#checking using int(nu)+1
cdfv <- cumsum(probabilities)
randomvalue <- runif(1)

upperb <- function(randomvalue) {
  for (i in 1:length(cdfv)) {
    if (randomvalue <= cdfv[i]) {
      return(cdfv[i])
    }
  }
  return(NULL)
}

#showing where random value U is placed
print(randomvalue)
answer <- upperb(randomvalue)
print(answer)

#checking using int(nu)+1
checking <- as.integer(5 * randomvalue) +1
print(checking)
```



```
> print(randomvalue)
[1] 0.5381967
> answer <- upperb(randomvalue)
> print(answer)
[1] 0.71
> #checking using int(nu)+1
> checking <- as.integer(5 * randomvalue) +1
> print(checking)
[1] 3
> 
```