monomolecular.R

davidharvey

Thu May 9 11:02:04 2019

```
monomolecular = function(a = 140, b = 0, x = 0, cycles = 300, coins = 0){

  # total particles in mixture
  marbles = a + b + x

  # vector of times
  time = seq(0,cycles)

  # vectors with id numbers for marbles and for coin flips
  id_marble = seq(1:marbles)
  id_flip = seq(1:2^coins)

  # create vectors to store counts of a, b, and x over time
  a_time = rep(a, cycles + 1)
  b_time = rep(b, cycles + 1)
  x_time = rep(x, cycles + 1)

  # vector giving composition of mixture in jar
  jar = c(rep("A", a), rep("B", b), rep("X", x))

  # create first row in archive of jar's composition as f(time)
  archive = jar

  # loop to play game: draw marble; flip coin; evalute result
  # either change A to B or leave as is; build matrix of results
  # keep track of numbers of A, B, and X at each time
  for (i in 2:(cycles + 1)){
    draw_marble = sample(id_marble, 1)
    flip_coin = sample(id_flip, 1)
    if(jar[draw_marble[1]] == "A" & flip_coin == 1){
      jar[draw_marble[1]] = "B"
    }

    archive = rbind(archive,jar)

    a_time[i] = length(which(jar == "A"))
    b_time[i] = length(which(jar == "B"))
    x_time[i] = length(which(jar == "X"))
  }

  marbles_time = data.frame(a_time, b_time, x_time)

  # output results of simulation; initial counts can be deleted
  out = list("mechanism" = "monomolecular",
             "coins" = coins,
             "marbles" = marbles,
             "time" = time,
```

```
              "marbles_time" = marbles_time,
              "a_initial" = a,
              "b_initial" = b,
              "x_initial" = x,
              "cycles" = cycles,
              "a_time" = a_time,
              "b_time" = b_time,
              "x_time" = x_time,
              "archive"  = archive)

}
```