SimpleUrnSolution

Paul M

12/9/2020

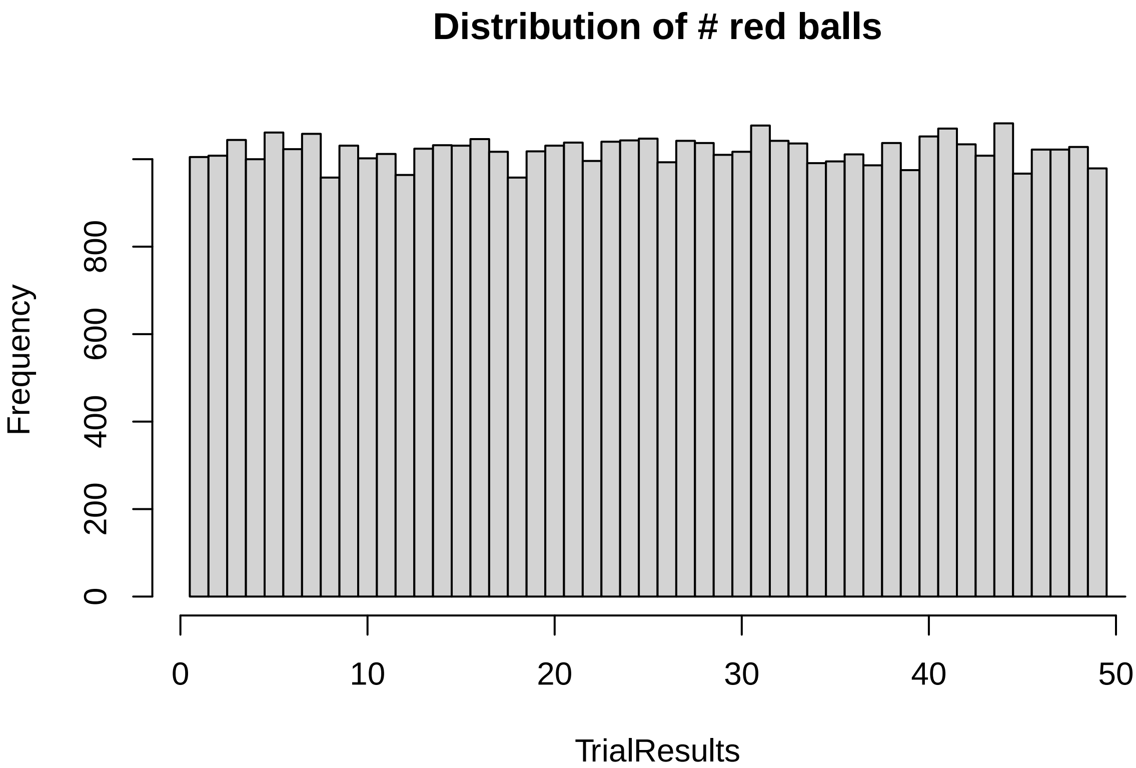
This shows some working code for the version of the Urn in which there is no "black" ball.

```
# set the random number seed
set.seed(16)

# define your variables
# How many balls do we start with
InitialNumberOfBalls<-2
# How many balls do we need eventually
TargetNumberOfBalls<-50

UrnSim <- function(InitialNumberOfBalls, TargetNumberOfBalls){
  # set up the initial state of the urn
  Urn<-rep("NoBall",TargetNumberOfBalls)
  # we will start with two balls of different colors: "red" and "blue"
  Urn[1] <- "blue"
  for(i in 2:InitialNumberOfBalls){
    Urn[i] <- "red"
  }
  # set up a counter (NumberOfBalls) to keep track of how many balls we have
  NumberOfBalls<-sum(Urn=="red")+sum(Urn=="blue")

  # set-up a loop that pulls a ball from the urn and takes the appropriate action
  while (NumberOfBalls<TargetNumberOfBalls){
    # draw a ball (WhichBall)
    ranball <- sample(1:NumberOfBalls,1)
    # return the ball and add another one like it
    if(Urn[ranball] == "red"){
      Urn[NumberOfBalls + 1] <- "red"
    }else{
      Urn[NumberOfBalls + 1] <- "blue"
    }
    # increase the counter of how many balls we have in the urn
    NumberOfBalls<-sum(Urn=="red")+sum(Urn=="blue")
  }
  return(sum(Urn == "red"))
}
```

Let's look at the distribution of the number of red balls at the end when we draw until there are 50 balls

```
NumTrials <- 50000
TrialResults <- rep(0,NumTrials)
for (i in 1:NumTrials){
  TrialResults[i] <- UrnSim(2,50)
```

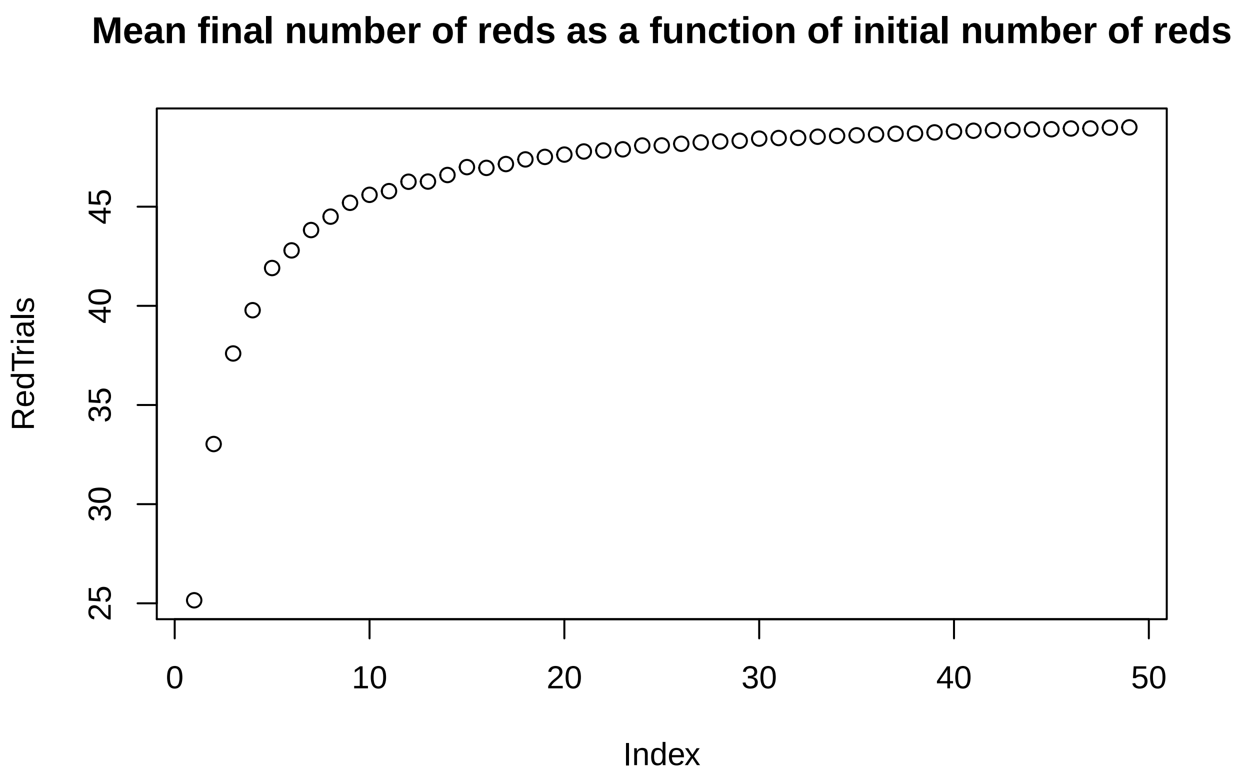
```
}
hist(TrialResults,main="Distribution of # red balls",breaks=seq(0.5,TargetNumberOfBalls+0.5,1))
```



In fact, the distribution of the number of balls of an given color (red or blue here) is Uniform, which is an amazing result!

Now let's look at the results as a function of the number of red balls at the beginning (assuming there is always 1 blue ball at the beginning)

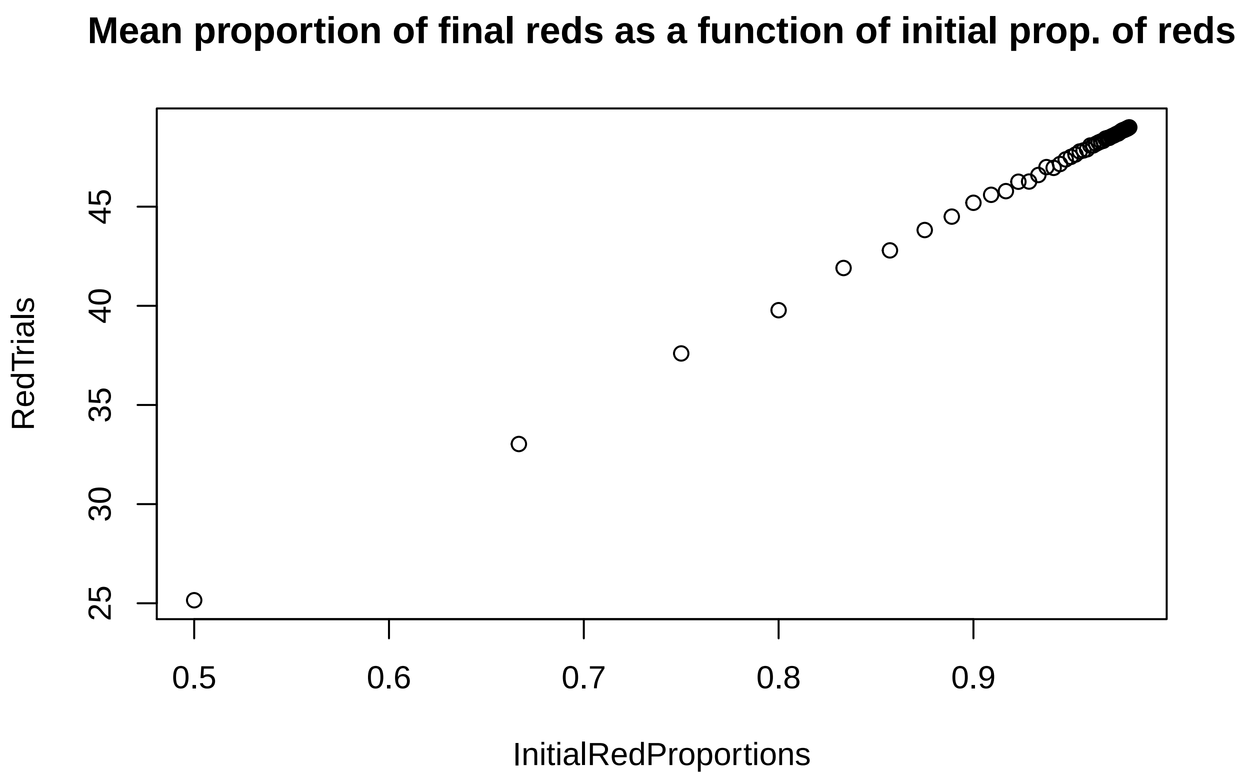
```
NumInitial <- seq(2,50,1)
RedTrials <- rep(0,length(NumInitial))
NumTrials <- 1000
for(r in 1:length(RedTrials)){
  TrialResults <- rep(0,NumTrials)
  for(i in 1:length(TrialResults)){
    TrialResults[i] <- UrnSim(NumInitial[r],50)
  }
  RedTrials[r] <- mean(TrialResults)
}
plot(RedTrials,main="Mean final number of reds as a function of initial number of reds")
```



Is there a simple relationship that explains this curve? (The answer is "yes", but you have to think a bit to work out what it is.)

Plot final proportion of reds versus initial proportion:

```
InitialNumberOfReds <- 1:49
InitialNumberOfBalls <- 2:50
InitialRedProportions <- InitialNumberOfReds/InitialNumberOfBalls
plot(y=RedTrials,x=InitialRedProportions, main="Mean proportion of final reds as a function of initial p
```


Mean proportion of final reds as a function of initial prop. of reds
RedTrials
25
30
35
40
45
0.5
0.6
0.7
0.8
0.9
InitialRedProportions