

Gambler's Ruin, A Curious Coin Flipping Game

Eamonn

28 November, 2016

INTRODUCTION

Each gambler has a pot of coins and each gambler simultaneously flips a single coin. If $n-1$ coins show the same side and 1 coin differs, the owner of the differing coin wins. The winner collects a coin from all the losers. In all other situations nobody wins or loses and no exchange of coins occurs. This process continues until one of the gamblers has no coins. What is the average number of games required until one gambler is ruined? In the simulation the probability p of the coin can be varied, however all gamblers' coins have probability p . We can also vary the number of gamblers and the number of coins each gambler starts the game with.

Function in R to simulate Gambler's Ruin game to approximate the answers

```

# if the probability (p) is not the same for each player and all coins,
# the coin may change hands and should be tracked, this is not done here

# n is a vector of players and their pot of coins
# p is a vector of players'/coins' constant probability

ComputeNbTosses <- function(n, p) {

  n <- n
  p <- p
  nbTosses <- 0
  k <- length(n)

  # keep tossing while no player has zero coins
  while (sum(n==0)==0) {

    # array to hold coin flips
    res <- vector("list",k)

    # toss the coins
    for(i in 1:k) { res[[i]] <- rbinom(1, 1, p[i]) }
    res <- unlist(res)

    # work out who won, if there is a winner
    # this code will order a vector, if there is a winner, they are index #1
    if ( sum(res) == k-1 ) {                                # if tails wins all are 1 except for a tail '0'

      (winner <- which( res %in% min(res))) # which position is the '0'
      res1 <- c(winner, (1:k)[-winner])      # put winner in 1st position and all others after winner

    } else if (sum(res) == 1) {                    # if head wins, all are 0 except for a single head

      (winner <- which(!res %in% min(res))) # which position is the '1'
      res1 <- c(winner, (1:k)[-winner])

    } else { res1 <- NULL }                        # stay the same if all tosses agree

    # update players' pot of coins
    if (is.null(res1)) {n=n} else {

      # the winner in vector position 1, so add k-1 coins to her pot
      n[res1[1]] <- n[res1[1]] + k-1
      # the losers are in vector positions not 1, so take 1 coin from each
      n[res1[2:k]] <- n[res1[2:k]]-1
    }

    # print(n) # show this if you want to see the realisations
    nbTosses = nbTosses + 1    # count the tosses
  }
}

```

```
}

ret = list()
ret$nbTosses = nbTosses      # record the count of the tosses
return(ret)
}
```

RESULTS

Execute function a large number of times for the scenarios in the book (P117), n is a vector of players and their pot of coins and p is a vector of all coins success with a constant probability.

p=0.5

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	1.000	1.000	1.333	1.000	7.000

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	1	1	2	2	13

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2.000	3.000	4.000	4.607	6.000	27.000

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.000	3.000	4.000	5.147	6.000	23.000

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.00	8.00	15.00	18.75	25.00	140.00

p=0.4

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	1.000	1.000	1.416	2.000	8.000

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	1.000	2.000	2.064	3.000	12.000

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2.000	3.000	4.000	4.815	6.000	29.000

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.000	3.000	4.000	5.357	6.000	29.000

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.00	9.00	15.00	19.57	26.00	119.00

It would be interesting perhaps to track percentage wins for each gambler?

CONCLUSION

The simulated results agree very well with those in the book (see the table at the foot of page 117)

REFERENCES

[1] Digital Dice Computational Solutions to Practical Probability Problems P42 and P114, Paul Nahin, Second Edition, 2013

COMPUTING ENVIRONMENT

R version 3.2.2 (2015-08-14)

Platform: x86_64-w64-mingw32/x64 (64-bit)

Running under: Windows 8 x64 (build 9200)

locale:

[1] LC_COLLATE=English_United Kingdom.1252

[2] LC_CTYPE=English_United Kingdom.1252

[3] LC_MONETARY=English_United Kingdom.1252

[4] LC_NUMERIC=C

[5] LC_TIME=English_United Kingdom.1252

attached base packages:

[1] stats graphics grDevices utils datasets methods

[7] base

other attached packages:

[1] MKmisc_0.993 knitr_1.15

loaded via a namespace (and not attached):

[1] DEoptimR_1.0-8 magrittr_1.5 assertthat_0.1

[4] tools_3.2.2 htmltools_0.3.5 RColorBrewer_1.1-2

[7] yaml_2.1.14 tibble_1.2 Rcpp_0.12.8

[10] stringi_1.1.2 rmarkdown_1.1 robustbase_0.92-6

[13] stringr_1.1.0 digest_0.6.10 evaluate_0.10

[1] "C:/Users\\User\\Documents\\GIT\\Digital-Dice-Gamblers-Ruin"

This took 10.98 seconds to execute.