Coin tossing simulation investigation

Eamonn O'Brien

Saturday, January 03, 2015

This note from David Spiegelhalter inspired me to look at coin tossing problems : 'What are the chances of 4 or more heads or 4 or more tails in *n* coin tosses (with a fair coin)?...about 47% chance from 10 flips...with 11 ~ 51%...'

1 What's the probability of x consecutive results with a limited number of tosses?

1.1 Create a function

```
tmpfun <- function(x, y) {{

  foo <- rbinom(x, 1, 1/2)  # binomial distribution of fair coin tosses
  rx <- rle(foo)            # function to examine runs in a sequence

   a <- any( rx$lengths  >= y )                         # any run of 0s or 1s in sequence?

   b <- any( rx$lengths[ rx$values %in% c(0) ] >= y ) # only run of 0s in sequence?

   c <- any( rx$lengths[ rx$values %in% c(1) ] >= y ) # only run of 1s in sequence?

   d <- any(rx$lengths[ rx$values %in% c(0)]>= y) &    # both 0s and 1s in a sequence?
   any(rx$lengths[ rx$values %in% c(1)]>= y)

   f <- (all(rx$lengths[ rx$values %in% c(0)] < y) & any(rx$lengths[ rx$values %in% c(1)] >= y))

   g <- any(rx$lengths[ rx$values %in% c(0)] >= y)& all(rx$lengths[ rx$values %in% c(1)] < y )
}
ret = list()  # think of a venn diagram
ret$a  = a # any run of desired length , either heads or tails or both
ret$b  = b # any run of desired length , tails only
ret$c  = c # any run of desired length , heads only
ret$d  = d # any run of desired length , of both heads and tails
ret$f  = f # any run of desired length , with a (single) run of heads and so no run of tails
ret$g  = g # any run of desired length , with a (single) run of tails and so no run of heads

return(ret)

}
```

1.2 Execute function and manage the output

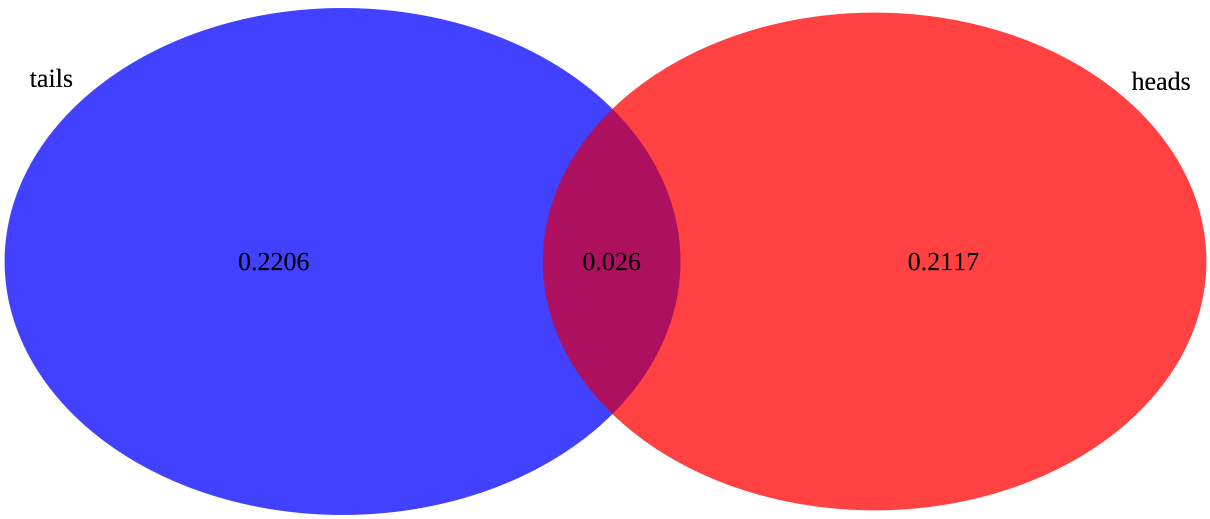
```r
set.seed(123)   # reproducible result
run <- 4        # number of consecutive tosses
tosses <- 10    # total number of tosses

z <- replicate(1e04 , tmpfun(tosses, run))  # execute function large number of times
new_mat  <- array(as.numeric(z), dim(z))    # manage the output
foo <- as.data.frame(as.matrix(t(new_mat))) # create a data frame
head(foo)                                   # look at first 5 rows of data
```

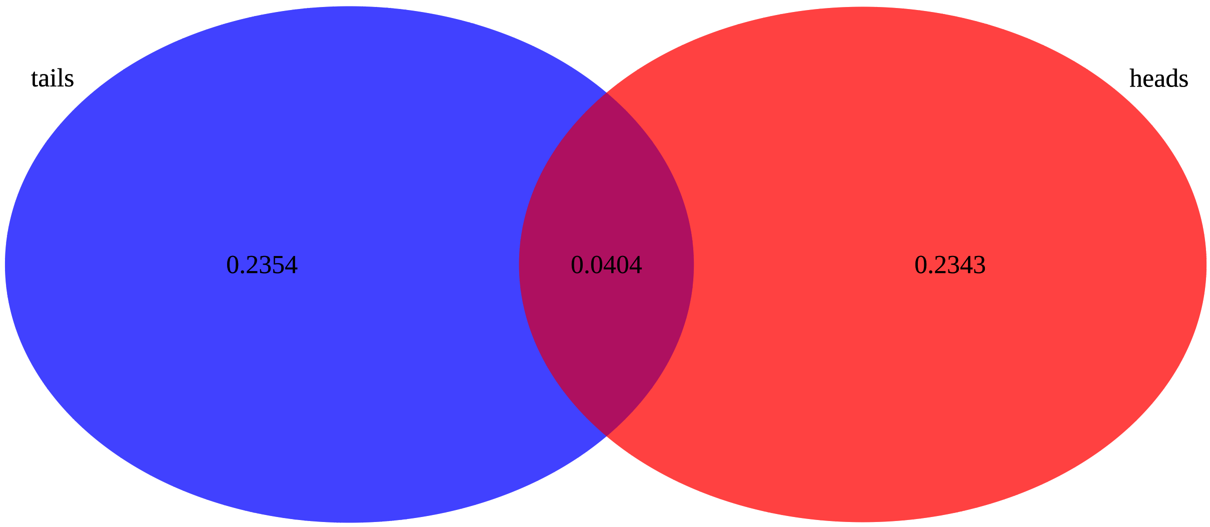
```
  V1 V2 V3 V4 V5 V6
1  0  0  0  0  0  0
2  0  0  0  0  0  0
3  1  0  1  0  1  0
4  1  0  1  0  1  0
5  1  1  0  0  0  1
6  0  0  0  0  0  0
```

```r
res <- apply(foo,2,mean)                    # calculate column averages and examine
```

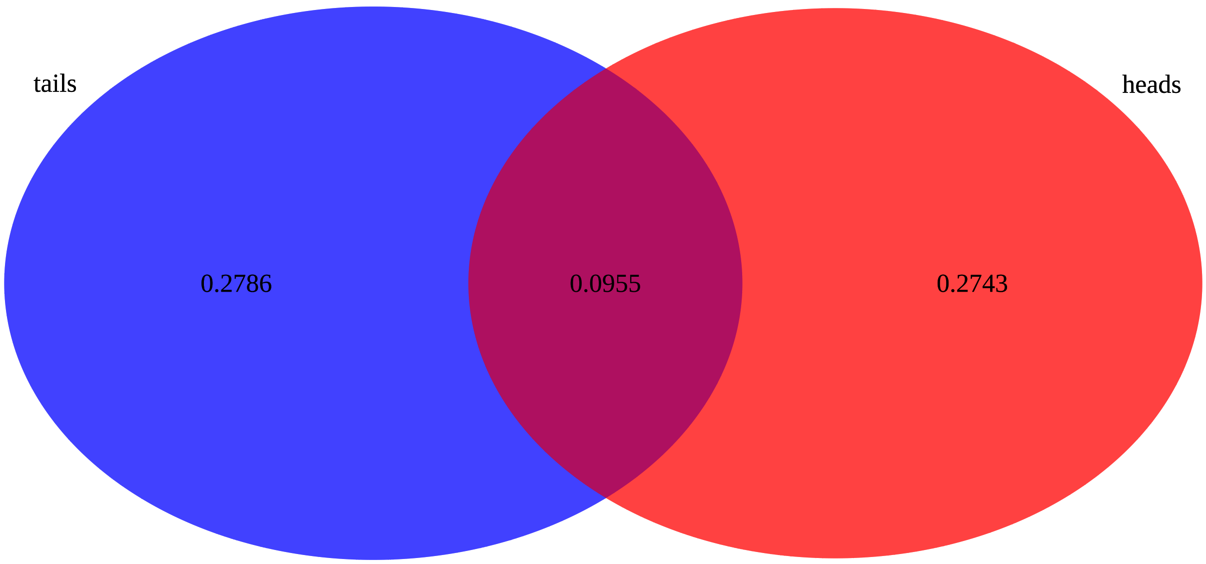
1.3 The probability of 4 runs or more in 10 tosses is 0.4583. So 0.5417 of the time no run of 4 heads and/or tails observed.



1.4 The probability of 4 runs or more in 11 tosses is 0.5101. So 0.4899 of the time no run of 4 heads and/or tails observed.



1.5 The probability of 4 runs or more in 15 tosses is 0.6484. So 0.3516 of the time no run of 4 heads and/or tails observed.



2 Examine how many tosses are required for x consecutive results in a row? We are not concerned if it is heads or tails… whichever consecutive result comes first.

2.1 Create a function

```r
coin <- c(0,1)

ComputeNbTosses <- function(targetTosses) {
  nbTosses <- 0
  nbHeadsInRow <-  0
  nbTailsInRow <-  0
  allTosses <-  c()

  # keep tossing unless we reach target for either heads or tails
  while  (nbHeadsInRow < targetTosses & nbTailsInRow < targetTosses)  {

    toss = sample(coin,1,T)          # toss an unbiased coin
    allTosses = c(allTosses, toss)   # accumulate the tosses

    # count occurrences of runs of heads and of tails
    if (toss == 1) {nbHeadsInRow = nbHeadsInRow + 1} else {nbHeadsInRow = 0}
    if (toss == 0) {nbTailsInRow = nbTailsInRow + 1} else {nbTailsInRow = 0}

    nbTosses = nbTosses + 1         # count the tosses
  }

  ret = list()
  ret$nbTosses = nbTosses           # record the count of the tosses
  # ret$allTosses = allTosses       # collect this if you want to check simulation works as expected
  return(ret)

}
```

2.2 Execute function for one scenario and manage the output

```
set.seed(4321)                              # reproducible result
n <- 4                                        # number of consecutive tosses
result <- replicate(10000, ComputeNbTosses(n)) # execute function large number of times
```

2.3 Summary of results

```
summary(unlist(result))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    4.0     6.0    11.0    14.9    19.0   140.0 
```

```
quantile( unlist(result),c(.001, .025,.1,.9,.95,.975,.99, .999))
```

```
  0.1%   2.5%    10%    90%    95%  97.5%    99%  99.9% 
 4.000  4.000  4.000 30.000 39.000 48.000 59.000 88.001 
```

Therefore on average (mean) 14.8978 tosses are required to obtain 4 consecutive results. It is unlikely (less than 5% of the time) that 4 runs will require more than 39 tosses.

2.4 Perform the simulation en masse (simulate for a number of scenarios in one function)

```
set.seed(123)   ##for a reproducible result
I <-  c(2:10)     ##consecutive runs of interest
nrep <- 1000    ##simulations to run on each scenario

pwpr <- array(dim=c(length(I),nrep,1),
              dimnames=list(consecutive=I ,
                            simulation=seq(nrep),
                            Estimate=c("mean"))
)
```

2.5 Loop

```
for (i in seq_along(I)) {
      pwpr[i,,]  <- plyr::raply(nrep,
                     unlist(ComputeNbTosses(I[i]))[1][[1]]
                        )

    }
```

2.6 Summary

```
pwpr[,1:10,] # first 10 simulations
```

```
           simulation
consecutive    1    2   3    4    5    6   7    8    9   10
         2     5    3   6    4    3    2   2    2    3    2
         3    18    3   6   13   13    8   5   13    3    3
         4    21   11   6    6   50   16  28    6   25    6
         5    20  116  32    5   28    6 127   13   38   12
         6    18   73  25   10   83   49 348  102   10   28
         7    78   58 373   36   99  494  69   58  259  225
         8    85   84 325   46  160  396 262  646  478  217
         9  1241 2289 391  382 1088   33 409  108   20  548
         10   72  202 878 1290  294 2246 353 3471 1774 1352
```

```
p0 <- function(x) {formatC(x, format="f", digits=0)}
```

2.7 Means, rownames are the desired run of consecutive tosses and below the average number of tosses

```
(resmeans  <-  (apply(pwpr,c(1),mean,na.rm=TRUE) ))
```

```
      2       3       4       5       6       7       8       9      10
  3.059   7.071  15.252  30.338  64.312 133.022 258.113 495.839 969.412
```

2.8 Percentiles

```
(resci <-  (apply(pwpr,c(1 ),
                 quantile,c(.001, .025,.1,.25,.5,.75,.9,.95,.975,.99, .999), na.rm=TRUE)))
```

```
       consecutive
            2      3      4       5       6       7        8        9       10
  0.1%  2.000  3.000  4.000   5.000   6.000   7.000    8.000    9.000   10.000
  2.5%  2.000  3.000  4.000   5.000   6.000   8.000   13.000   18.000   31.975
  10%   2.000  3.000  4.000   7.000  11.000  20.000   30.900   52.000  110.700
  25%   2.000  4.000  6.000  11.000  21.000  42.000   87.750  139.000  282.750
  50%   3.000  6.000 11.000  23.000  44.000  93.000  182.000  356.500  677.500
  75%   4.000  9.000 20.000  39.000  90.000 178.250  363.500  694.750 1371.750
  90%   5.000 13.000 31.000  64.000 141.000 302.200  571.300 1126.400 2156.000
  95%   6.000 16.050 40.000  78.100 180.050 397.100  697.450 1451.800 2757.950
  97.5% 7.000 19.000 47.000  99.025 226.025 459.200  889.200 1721.425 3566.650
  99%   8.000 23.010 58.000 123.040 290.220 594.090 1115.500 2242.470 4375.420
  99.9% 15.001 38.005 81.005 207.015 394.024 767.067 1466.228 3508.373 5937.867
```

Interpret one example: The median number of tosses required for 3 consecutive runs of a fair coin are 6. It is unlikely (less than 5% of the time) that to observe 3 consecutive runs will require more than 16.05 tosses.

3 References

http://stackoverflow.com/questions/21392665/homework-simulating-coin-tosses-until-consecutive-heads-using-r
https://twitter.com/d_spiegel/status/544422589670916096
http://stats.stackexchange.com/questions/21825/probability-over-multiple-blocks-of-events
http://r.789695.n4.nabble.com/Plot-does-not-show-in-R-td4693637.html http://math.stackexchange.com/questions/364038/expected-number-of-coin-tosses-to-get-five-consecutive-heads .5^2 http://www.cs.cornell.edu/~ginsparg/physics/INFO295/mh.pdf
error on bottom of page 5, n=5 not reported

(2^(n+1))-2

2*(2^n-1)

http://stats.stackexchange.com/questions/91518/waiting-time-for-successive-occurrences-of-a-result-when-rolling-a-die?rq=1
p<-.5^n 2*(1-p)/p

http://stats.stackexchange.com/questions/12174/time-taken-to-hit-a-pattern-of-heads-and-tails-in-a-series-of-coin-tosses
http://stats.stackexchange.com/questions/91518/waiting-time-for-successive-occurrences-of-a-result-when-rolling-a-die?rq=1
http://stats.stackexchange.com/questions/126884/how-many-times-do-i-have-to-roll-a-die-to-get-six-six-times-in-a-row
http://math.stackexchange.com/questions/192177/how-many-times-to-roll-a-die-before-getting-two-consecutive-sixes

4 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] grid      stats     graphics  grDevices utils     datasets
[7] methods   base

other attached packages:
[1] VennDiagram_1.6.17  futile.logger_1.4.3 knitr_1.14

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.6          codetools_0.2-14     digest_0.6.10
 [4] plyr_1.8.4           futile.options_1.0.0 formatR_1.4
 [7] magrittr_1.5         evaluate_0.9         stringi_1.1.1
[10] rmarkdown_1.0        lambda.r_1.1.9       tools_3.2.2
[13] stringr_1.1.0        yaml_2.1.13          htmltools_0.3.5

[1] "C:/Users/User/Documents/GIT/coin-tossing-simulation"
```

This took 28.71 seconds to execute.