Monte Carlo simulations

Group 9

M5 ICA1

Introduction

> Monte Carlo simulations are a statistical technique used to model probabilistic (or "stochastic") processes and establish the odds for a variety of outcomes. The concept was first popularized right after World War II. To study nuclear fission, mathematician Stanislaw Ulam coined the term in reference to an uncle who loved playing the odds at the Monte Carlo casino (then a world symbol of gambling, like Las Vegas today).

To get started, load package `tidyverse` and set the random number generator in your setup R chunk. In the following problems you will make heavy use of functions

- `sample(x, size, replace = FALSE, prob = NULL)`,
- `replicate(n, expr, simplify = "array")`.

```
library(tidyverse)
```

Antoine Gombaud's question

Conduct a simulation to answer the question below that was initially posed by Antoine Gombaud (a famous gambler in the 17th century).

Which is more likely:

a. getting at least one 6 when rolling a single fair six-sided die 4 times,
b. getting at least one pair of sixes when two fair six-sided dice are thrown 24 times.

The number of simulation replications you choose is at your discretion, but if you choose a number too small the results will not be accurate.

```
dice<- replicate(n = 100000,
                 sample(c(1),1))
result <- replicate(n = 100000, sample(c(4),1))
tab <- table(dice, result)
prop.table(tab)
```

```
    result
dice       1       2       3       4
   1 0.25021 0.25061 0.24875 0.25043
```

```
dice<- replicate(n = 100000,
                 sample(c(2),1))
result <- replicate(n = 100000, sample(c(24),1))
tab <- table(dice, result)
prop.table(tab)
```

```
    result
dice       1       2       3       4       5       6       7       8       9
   1 0.02071 0.02036 0.01987 0.02082 0.02105 0.02101 0.02015 0.02146 0.02120
   2 0.02113 0.02113 0.02041 0.01997 0.02085 0.02209 0.02096 0.02105 0.02065
    result
dice      10      11      12      13      14      15      16      17      18
   1 0.02067 0.02155 0.02123 0.02160 0.02057 0.02126 0.02100 0.02015 0.02047
   2 0.02133 0.02064 0.02091 0.02125 0.02122 0.02029 0.02121 0.02047 0.02074
    result
dice      19      20      21      22      23      24
   1 0.02086 0.02215 0.01978 0.02068 0.02140 0.02071
   2 0.02043 0.02009 0.02051 0.02073 0.02090 0.02033
```

It is more likely to roll a 6 with one rol 4 times then rolling a pair of sixes thrown 24 times.

Enough?

How many Monte Carlo experiments are enough?

a. Perform Monte Carlo simulation to evaluate the probability of getting "Heads" in a fair coin toss.
b. Use `ggplot()` to plot the probability estimate on the y-axis and the iteration (number of Monte Carlo) experiments on the x-axis. As the number of iterations gets large, you should see this value stabilize at around 0.50.

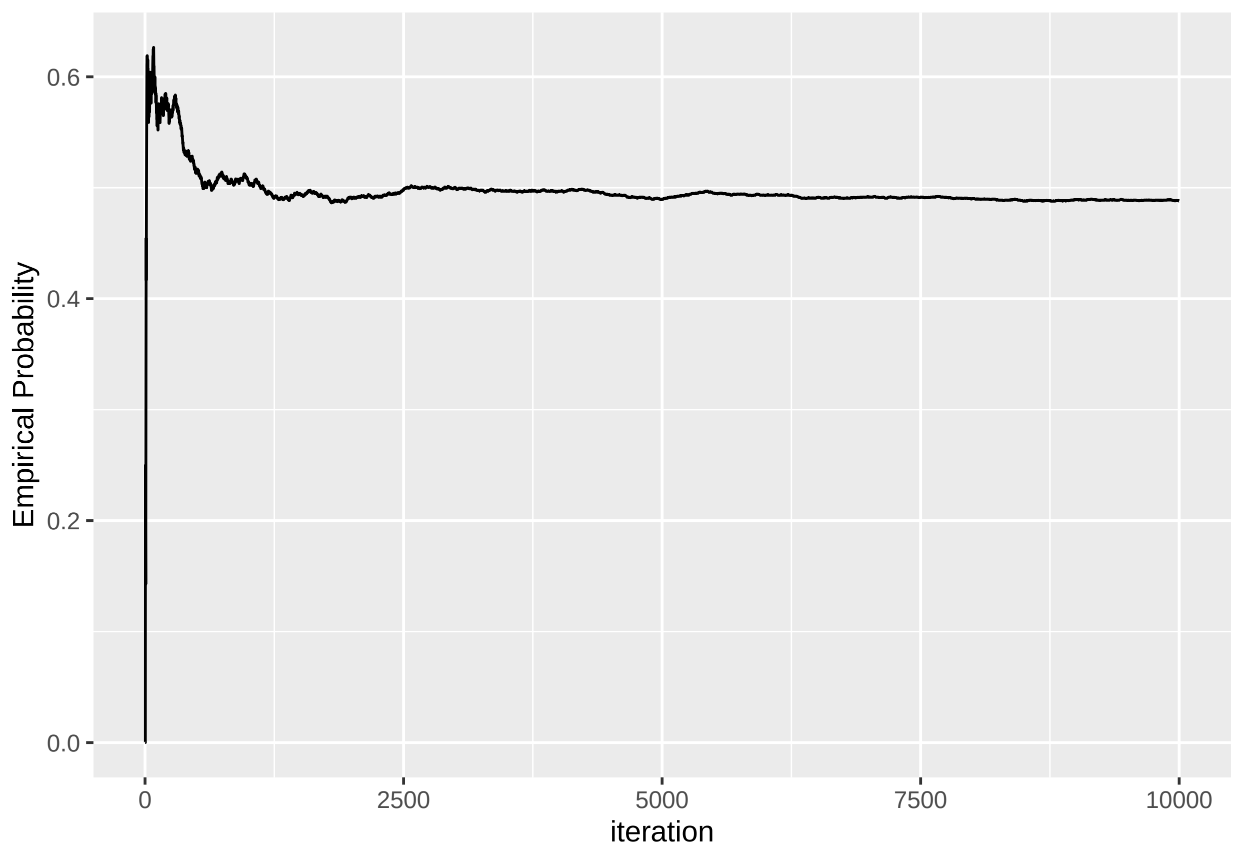
```
#create a function flip() to get the result of one flip
flip<-function(){
  res<-sample(c("H", "T"), size = 1, replace = TRUE)
  return(res=="H")
}

#Use the above to create a dataframe (toss.df) of 10000 replication results.

toss.df<-data.frame(x=replicate(n = 10000, expr = flip()))
mean(toss.df$x)
```

```
[1] 0.4882
```

```
# Use mutate to add two new columns `iterations` and `cum.prob` to the dataframe toss.df. `cum.prob` wi
toss.df %>%
  mutate(cum.prop=cummean(x),iteration=1:10000) %>%
  ggplot(mapping=aes(x=iteration, y=cum.prop))+
  geom_line()+
  labs(x="iteration", y="Empirical Probability")
```



```
# Use ggplot and geom_line with the above dataframe to plot the iteration vs cumulative probabilities g
```

About 5000-10000 experiments are enough are enough in order to have a probability of around .50.

Birthday problem

Conduct a Monte Carlo simulation to answer the following questions related to the birthday problem or birthday paradox.

a. What is the probability of at least two people sharing the same birthday (month and day) from a random sample of 23 individuals?

b. What is the probability of at least two people sharing the same birthday (month and day) from a random sample of 70 individuals?

c. Create a plot using `ggplot` with the number of individuals on the x-axis and the probability of at least one pair on the y-axis. You should simulate the probability of at least two people sharing the same birthday for each number of individuals from 2 to 100.

You may ignore leap year and assume 365 days per year. The number of simulation replications you choose is at your discretion, but if you choose a number too small the results will not be accurate.

```
#create a function bday.match(n) which will check whether there is a birthday match in a year between `
bday.match <- function(n){
k = n
p <- numeric(k)
for (i in 1:k)      {
            q <- 1 - (0:(i - 1))/365
            p[i] <- 1 - prod(q)  }
prob <- p[n]
prob
}



# For (a) check probability of match for 23 people
emp.23 <- mean(replicate(100000, bday.match(23)))
emp.23
```

```
[1] 0.5072972
```

```
emp.70 <- mean(replicate(100000, bday.match(70)))
emp.70
```

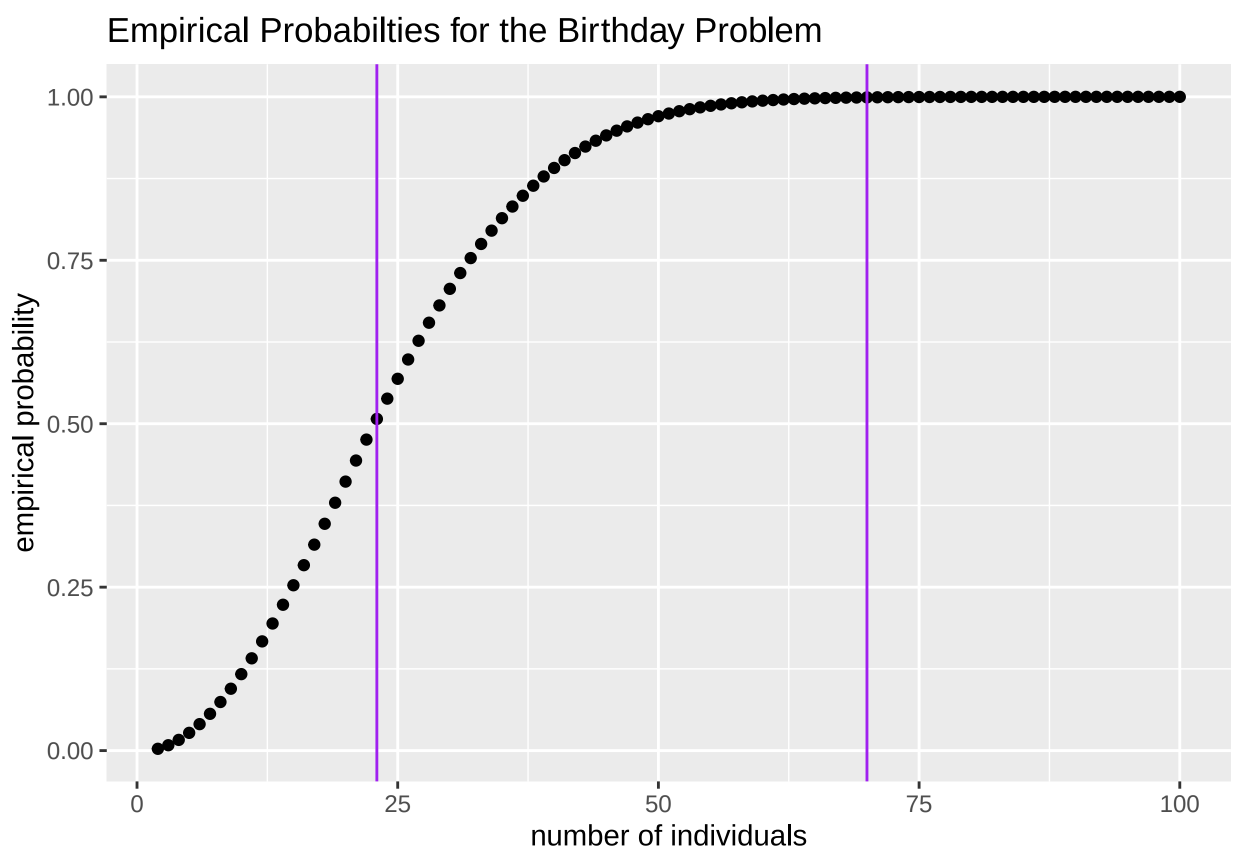
```
[1] 0.9991596
```

```
#(b)

# Here're the steps to create the dataframe needed for part (c)
nmat <- matrix(c(2:100))
event.results <- replicate(10000, apply(nmat, 1, bday.match))
emp.results <- apply(event.results, 1, mean)
emp.results.df <- data.frame(n = nmat, emp.prob = emp.results)
```

Reference: Rpubs https://rpubs.com/StatGirl302/TheBirthdayProblem Accessed on November 3, 2021

```
emp.results.df %>%
  ggplot(mapping=aes(x=nmat, y=emp.results))+
  geom_point()+
  geom_vline(xintercept = 23, color="purple")+
  geom_vline(xintercept = 70, color="purple")+
  labs(x="number of individuals", y="empirical probability", title="Empirical Probabilties for the Birtl
```



References

1. http://news.mit.edu/2010/exp-monte-carlo-0517
2. https://en.wikipedia.org/wiki/Antoine_Gombaud