Noname manuscript No.
(will be inserted by the editor)

Usage of Monte Carlo simulations for solving veridical type paradoxes

Martin Pavlik

Received: date / Accepted: date

Abstract We used both Monte Carlo simulation and analytical method to solve veridical type paradoxes. Following analytical methods were used: conditional probability, Bayes' rule, and Bayes' rule with multiple conditions. We successfully solved all the veridical type paradoxes and discovered why such a big discussion of Monty Hall problem appeared in 90's. We varied Monty Hall problem, using different doors count and prizes count.

Keywords Monte Carlo simulation, Bertrand's box paradox, Three prisoners dilemma, Monty Hall problem, Bayes' rule, Bayes' rule with multiple conditions, conditional probability, veridical type paradox

Mathematics Subject Classification (2000) 60-04, 65C05

1 Introduction

A paradox is a statement that contradicts itself and yet might be true (or wrong at the same time). A veridical paradox produces a result that appears absurd, but is demonstrated to be true nevertheless. We will focus on veridical type paradoxes using Monte Carlo simulations, but we also will provide analytical solutions for most of the cases. The key part of our research was Monty Hall problem, and we provide Monte Carlo simulations for all the calculations in Monty Hall problem case, but analytical solution would be mentioned for most of the cases, too. We will describe and solve following veridical type paradoxes: Bertrand's box paradox, Three prisoners dilemma, Monty Hall problem. The key part of our research was Monty Hall problem, and we will explain, why so many PhD researchers insisted on probability equal 0.5 back in 90's [7].

Martin Pavlik
www.quantitativeresearch.eu

2 Monte Carlo simulation

Monte Carlo simulation belongs to advanced statistical tools. A simulation is an experiment, usually conducted on a computer, involving the use of random numbers. A random number stream is a sequence of statistically independent random variables uniformly distributed usually in the interval [0,1).

Simulations are used where it is difficult to use purely analytical methods to either model the real-life situation or to solve the underlying mathematical problems. They rely on repeated random sampling to obtain numerical results. We used random sampling to obtain approximation of probabilities.

3 Veridical type paradoxes

We solved following veridical type paradoxes: Bertrand's box, Three prisoners dilemma and Monty Hall problem. We used both analytical and numerical approach. Some calculations were conducted just numerically.

3.1 Bertrand's box

Bertrand's box paradox is a paradox of elementary probability theory. There are three boxes:

1. A box containing two gold coins
2. A box containing two silver coins
3. A box containing one gold coin and a silver coin

The question is: What is the probability of choosing gold coin, having known the first coin is also gold? Player is choosing box at random and does not swap boxes after first toss. It is an elementary example of conditional probability:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \tag{1}$$

- P(A) probability of choosing gold coin in second toss
- P(B) probability of choosing gold coin in first toss

$$P(B) = P(gold \mid GG) + P(gold \mid SS) + P(gold \mid SG) \tag{2}$$

$$P(B) = \frac{1 + 0 + \frac{1}{2}}{3} = \frac{1}{2} \tag{3}$$

$$P(A \cap B) = \frac{1}{3} \tag{4}$$

$$P(A \mid B) = \frac{\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3} \tag{5}$$

Result appears to be $\frac{1}{2}$ using common sense, but it is $\frac{2}{3}$ in fact. The correct solution of the problem is well known for a long time. The aim of our research was to build a Monte Carlo simulation of the problem. We coded following code in R:

```
#Bertrand's paradox
set.seed(100)
samplesize<-1000000
a<-sample(0:2,samplesize,replace=T)
# 3 boxes: 0- gold, gold; 1 - silver, silver; 2 - gold, silver
b<-sample(0:1,samplesize,replace=T) # 2 balls in each box
data<-data.frame(a,b)
data2<-subset(data,(a==0) | (a==2 & b==0),select=a)
round(sum(a==0)/nrow(data2),4) # final probability
```

Monte Carlo simulation confirms the probability equal 0.6667, which is $\frac{2}{3}$.

3.2 Three prisoners dilemma

Three prisoners problem is another veridical type paradox. Three prisoners A,B and C, are in separate cells and sentenced to death. The governor has selected one of them at random to be pardoned. The warden knows which one is pardoned, but is not allowed to tell. Prisoner A begs the warden to let him know the identity of the lucky one. Prisoner A knows that warden can't tell the identity of the one to be pardoned, so he proposed the warden following code : 'If B is pardoned, give me C's name. If C is pardoned give me B's name. If I am pardoned, flip a coin to decide whether to name B or C.' The warden tells A that it will be B. Prisoner A is pleased because he believes that his probability of surviving has gone up from $\frac{1}{3}$ to $\frac{1}{2}$, as it is now between him and C. This is what common sense says. The question is, what the true probabilities are. Analytical solution:

- A,B,C are corresponding prisoners
- P(A),P(B),P(C) are probabilities that governor pardoned corresponding prisoners
- a,b,c are events which warden mentions that corresponding prisoners were pardoned

$$P(A) = P(B) = P(C) = \frac{1}{3}, P(b \mid A) = \frac{1}{2}, P(c \mid A) = \frac{1}{2} \tag{6}$$

$$P(b \mid C) = 1, P(c \mid B) = 1, P(c \mid C) = 0, P(b \mid B) = 0 \tag{7}$$

It is more complicated problem. It can be solved using Bayes' rule.

$$P(A \mid b) = \frac{P(b \mid A)P(A)}{P(b \mid A)P(A) + P(b \mid B)P(B) + P(b \mid C)P(C)} \tag{8}$$

$$P(A \mid b) = \frac{\frac{1}{2} \cdot \frac{1}{3}}{\frac{1}{2} \cdot \frac{1}{3} + 0.\frac{1}{3} + 1.\frac{1}{3}} = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3} \tag{9}$$

$$P(A \mid c) = \frac{P(c \mid A)P(A)}{P(c \mid A)P(A) + P(c \mid B)P(B) + P(c \mid C)P(C)} \tag{10}$$

$$P(A \mid c) = \frac{\frac{1}{2} \cdot \frac{1}{3}}{\frac{1}{2} \cdot \frac{1}{3} + 1.\frac{1}{3} + 0.\frac{1}{3}} = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3} \tag{11}$$

$$P(C \mid b)=\frac{P(b \mid C)P(C)}{P(b \mid A)P(A)+P(b \mid B)P(B)+P(b \mid C)P(C)} \tag{12}$$

$$P(C \mid b)=\frac{1.\frac{1}{3}}{\frac{1}{2}.\frac{1}{3}+0.\frac{1}{3}+1.\frac{1}{3}}=\frac{\frac{1}{3}}{\frac{1}{2}}=\frac{2}{3} \tag{13}$$

Conclusion of three prisoners problem is, that warden's information is not telling anything about prisoner A future. Probability of being pardoned stays at $\frac{1}{3}$. Probability of being pardoned for prisoner C is now $\frac{2}{3}$. Monte Carlo simulation has been coded in R:

```
# Three prisoners problem
set.seed(100)
samplesize<-1000000
governor<-sample(0:2,samplesize,replace=T)
fm<-function(a,j) {
if (a[j]==0) {thisone<-sample(1:2,1,replace=T)}
if (a[j]==1) {thisone<-2}
if (a[j]==2) {thisone<-1}
thisone
}
warden<- sapply(1:samplesize, function(j) fm(governor,j))
r<-data.frame(governor,warden)
# Warden told B, given governor choose A.
sum(r$warden==1 & r$governor==0)/sum(r$warden==1)
# Warden told C, given governor choose A.
sum(r$warden==2 & r$governor==0)/sum(r$warden==2)
# Warden told B, given governor choose C.
sum(r$warden==1 & r$governor==2)/sum(r$warden==1)
# Warden told C, given governor choose B.
sum(r$warden==2 & r$governor==1)/sum(r$warden==2)
```

Monte Carlo simulations provide the same results as analytical solution provides: 0.3335 and 0.6665.

3.3 Monty Hall problem

Monty Hall problem was the key part of our research. We will try to explain the discussion of the problem back in 90's [7]. Suppose you are on a game show, and you are given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what is behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice? Common sense tells that whether you switch your choice or not, you still have 50 per cent chance to win. Analytical solution of Monty Hall problem for three doors is :

- Events C_1, C_2, C_3 are indicating the car is behind door 1,2 or 3.
- $P(C_1)=P(C_2)=P(C_3)=\frac{1}{3}$
- Event X_1 is indicating player initialy choosing door 1.
- As the position of the car is independent of the players first choice $P(C_i|X_1)=\frac{1}{3}$.

– H_3 is host opening door 3.

Following probabilities are obvious :

$$P(H_3 \mid C_1, X_1) = \frac{1}{2}, P(H_3 \mid C_2, X_1) = 1, P(H_3 \mid C_3, X_1) = 0 \tag{14}$$

Probability that car is behind door No. 2, given player initialy choosing door 1 and host opening door No. 3 is :

$$P(C_2 \mid H_3, X_1) = \frac{P(H_3 \mid C_2, X_1)P(C_2|X_1)}{P(H_3 \mid X_1)} \tag{15}$$

$$P(C_2 \mid H_3, X_1) = \\ = \frac{P(H_3 \mid C_2, X_1)P(C_2 \mid X_1)}{P(H_3 \mid C_1, X_1)P(C_1 \mid X_1) + P(H_3 \mid C_2, X_1)P(C_2 \mid X_1) + P(H_3 \mid C_3, X_1)P(C_3|X_1)} \tag{16}$$

$$P(C_2|H_3, X_1) = \frac{1.\frac{1}{3}}{\frac{1}{2}\frac{1}{3} + 1.\frac{1}{3} + 0.\frac{1}{3}} = \frac{2}{3} \tag{17}$$

Probability that car is behind door No.1, given player initially choosing door No. 1 and host opening door No. 3 is :

$$P(C_1|H_3, X_1) = \frac{(H_3|C_1, X_1)P(C_1|X_1)}{P(H_3|X_1)} \tag{18}$$

$$P(C_1 \mid H_3, X_1) = \\ = \frac{P(H_3 \mid C_1, X_1)P(C_1 \mid X_1)}{P(H_3 \mid C_1, X_1)P(C_1 \mid X_1) + P(H_3 \mid C_2, X_1)P(C_2 \mid X_1) + P(H_3 \mid C_3, X_1)P(C_3 \mid X_1)} \tag{19}$$

$$P(C_1|H_3, X_1) = \frac{\frac{1}{2}\frac{1}{3}}{\frac{1}{2}\frac{1}{3} + 1.\frac{1}{3} + 0.\frac{1}{3}} = \frac{1}{3} \tag{20}$$

'Flip a coin' decision has following probability :

$$P(flip\ a\ coin) = \frac{1}{2}P(C_1|H_3, X_1) + \frac{1}{2}P(C_2|H_3, X_1) \tag{21}$$

$$P(flip\ a\ coin) = \frac{1}{2}\frac{1}{3} + \frac{1}{2}\frac{2}{3} = \frac{1}{2} \tag{22}$$

Analytical solution shows that player should switch her choice, since chance to win is $\frac{2}{3}$. In case player does not switch her choice, chance to win is just $\frac{1}{3}$. We used Bayes' rule with multiple conditions. 'Flip a coin' decision probability is equal to $\frac{1}{2}$. Since Monty Hall problem is key part of our research, we have focused on many different decisions, which were simulated in Monte Carlo simulations. We explored following cases:

1. Not switching decision
2. Switching decision
3. Flip a coin decision
4. Tic-toc decision
5. Opposite tic-toc decision

We coded simulations which vary doors count and also prizes count. If contestant decides to switch door, and there is more than one door available, she will choose another door randomly. If contestant flips a coin, she does it just once. If there is more just one door available, she will choose another door randomly, too. Following code simulates different doors count and also cars count options. Code has been written in R:

```
#Monty Hall problem
samplesize<-100000 # sample size
doors<-9 # doors count -1
door<-vector("numeric",length=doors*(doors-1)/2)
prize<-vector("numeric",length=doors*(doors-1)/2)
changedp<-vector("numeric",length=doors*(doors-1)/2)
nochangep<-vector("numeric",length=doors*(doors-1)/2)
ttp<-vector("numeric",length=doors*(doors-1)/2)
ottp<-vector("numeric",length=doors*(doors-1)/2)
flipp<-vector("numeric",length=doors*(doors-1)/2)
results <- data.frame(door, prize, changedp, ttp,flipp,
ottp,nochangep)
#tic-toc oppposite tic-toc function
ttott<-function(ttott1,win1,win2,trigger){
if (ttott1==0) {if (win2==0) {thisone<-0
} else {thisone<-1 }}
if (ttott1==1) {if (win1==0) {thisone<-1
} else {thisone<-0 }}
if (trigger=="ott") {if (thisone==0) {thisone<-1
} else {thisone<-0 }}
thisone}
for (i in 2:doors) {
for (j in 1:(i-1)) {
set.seed(100)
initial<-replicate(samplesize,list(sample(0:i,1,replace=F)))
priz<-replicate(samplesize,list(sample(0:i,j,replace=F)))
# initial guess & prizes behind doors
flip<-sample(0:1,samplesize, replace=T) # flip a coin
tt<-sample(0,samplesize,replace=T) # tic toc
ott<-sample(0,samplesize,replace=T) # opposite tic toc
choices<-replicate(samplesize,list(c(0:i)))
# WHICH GOAT WILL BE SHOWN
goats<-mapply(setdiff,choices,priz)
goats2<- lapply(1:ncol(goats), function(p) goats[,p])
# goats are opposite prizes
notinitial<-mapply(setdiff,choices,initial)
notinitial2<-lapply(1:ncol(notinitial),
function(p) notinitial[,p])
# group of not initial decisions
goats3<-mapply(intersect,goats2,notinitial2)
goats4<-lapply(goats3, function(p) c(p,p))
goat<-lapply(goats4, function(p) sample(p,1))
# to show just one goat from intersect not initial & goats
remove(goats,goats2,choices,notinitial,goats3,goats4)
chmind<-mapply(setdiff,notinitial2,goat)#to change mind
#to exclude goat which was shown from not initial group
if (is.null(ncol(chmind))==FALSE) {
chmind2<- lapply(1:ncol(chmind), function(p) chmind[,p])
} else {chmind2<-chmind }
chmind3<-lapply(chmind2, function(p) c(p,p))
```

```
newmind<-lapply(chmind3, function(p) sample(p,1))
# to choose 1 new decision from all the available
remove(notinitial2, goat,chmind,chmind2,chmind3)
win1<-mapply(intersect,newmind,priz)#win1 changed mind
win2<-mapply(intersect,initial,priz)#win2 not changed mind
win13<-as.numeric(lapply(win1,function(p) length(p)==0))
win23<-as.numeric(lapply(win2, function(p) length(p)==0))
# intersection - 1 no intersection,0 intersection
for(l in 1:(samplesize-1)) { # TIC-TOC, OPPOSITE TIC TOC
tt[l+1]<-ttott(tt[l],win13[l],win23[l],'tt')
ott[l+1]<-ttott(ott[l],win13[l],win23[l],'ott')}
#PROBABILITIES
win1p<-sum(win13==0)/samplesize #win1p changed mind
win2p<-sum(win23==0)/samplesize #win2p not changed mind
remove(newmind,priz,initial,win1,win2)
flipfr<-data.frame(win13,win23,flip,tt,ott)
flip1<-nrow(flipfr[flipfr$flip==1 & flipfr$win13==0,])
flip2<-nrow(flipfr[flipfr$flip==0 & flipfr$win23==0,])
# flip1 - changed mind,flip2 - initial decision
tt1<-nrow(flipfr[flipfr$tt==1 & flipfr$win13==0,])
tt2<-nrow(flipfr[flipfr$tt==0 & flipfr$win23==0,])
ott1<-nrow(flipfr[flipfr$ott==1 & flipfr$win13==0,])
ott2<-nrow(flipfr[flipfr$ott==0 & flipfr$win23==0,])
flipp<-(flip1+flip2)/samplesize
tictocp<-(tt1+tt2)/samplesize
opptictocp<-(ott1+ott2)/samplesize
results$door[i*(i-1)/2-i+1+j]<-i+1#RESULTS - WRITTING
results$prize[i*(i-1)/2-i+1+j]<-j
results$changedp[i*(i-1)/2-i+1+j]<-round(win1p*100,3)
results$ttp[i*(i-1)/2-i+1+j]<-round(tictocp*100,3)
results$flipp[i*(i-1)/2-i+1+j]<-round(flipp*100,3)
results$ottp[i*(i-1)/2-i+1+j]<-round(opptictocp*100,3)
results$nochangep[i*(i-1)/2-i+1+j]<-round(win2p*100,3)
remove(win13,win23,flipfr,flip,tt,ott) } }
nname<- paste(toString(samplesize),"r.txt",sep="")
write.table(results,nname,append=FALSE)
```

Table 1 and table 2 show output of the code. Probabilities were calculated with sample size equal 1,5 million. Probability theory allows a simple derivation of Monte Carlo simulations error. If contestant does not change her mind, probability of winning is a very simple formula.

$$P(not\ switching\ decision) = \frac{prizes\ count}{doors\ count} \tag{23}$$

We can measure Mean Absolute Error of Monte Carlo simulations probabilities for 'not switching' decision and Bias.

$$MAE = \frac{1}{36}\sum_{i=1}^{36} \mid p_{MC_i} - p_{true_i} \mid = \frac{0.933}{36} = 0.026 \tag{24}$$

$$Bias = \frac{1}{36}\sum_{i=1}^{36}(p_{MC_i} - p_{true_i}) = \frac{-0.191}{36} = -0.005 \tag{25}$$

Table 1 Simulations results

No	Door	Prize	Changep	Ttp	Flipp	Ottp	No changep
1	3	1	66.645	55.585	50.061	44.415	33.355
2	4	1	37.545	31.857	31.278	30.005	24.992
3	4	2	75	66.686	62.54	60.013	49.993
4	5	1	26.704	23.5	23.35	22.854	19.979
5	5	2	53.287	47.463	46.657	45.705	40.047
6	5	3	80.014	73.354	69.997	68.534	59.99
7	6	1	20.86	18.802	18.791	18.505	16.649
8	6	2	41.67	37.76	37.493	37.015	33.303
9	6	3	62.565	57.244	56.253	55.6	50.053
10	6	4	83.273	77.724	75.042	74.074	66.719
11	7	1	17.155	15.756	15.767	15.591	14.286
12	7	2	34.268	31.582	31.477	31.138	28.581
13	7	3	51.457	47.512	47.114	46.796	42.892
14	7	4	68.592	63.727	62.917	62.387	57.163
15	7	5	85.661	80.856	78.553	77.89	71.399
16	8	1	14.564	13.591	13.562	13.408	12.49
17	8	2	29.171	27.15	27.097	26.916	25.009
18	8	3	43.77	40.764	40.609	40.408	37.492
19	8	4	58.368	54.537	54.252	53.886	49.989
20	8	5	72.953	68.614	67.716	67.327	62.504
21	8	6	87.511	83.353	81.251	80.743	74.955

Table 2 Simulations results

No	Door	Prize	Changep	Ttp	Flipp	Ottp	No changep
22	9	1	12.709	11.932	11.919	11.793	11.063
23	9	2	25.378	23.882	23.85	23.712	22.286
24	9	3	38.05	35.79	35.62	35.541	33.302
25	9	4	50.793	47.765	47.603	47.438	44.429
26	9	5	63.514	59.971	59.574	59.209	55.543
27	9	6	76.224	72.26	71.459	71.043	66.624
28	9	7	88.856	85.161	83.35	82.965	77.803
29	10	1	11.263	10.659	10.663	10.589	9.974
30	10	2	22.496	21.287	21.273	21.272	20.021
31	10	3	33.796	31.901	31.856	31.801	29.957
32	10	4	44.983	42.591	42.497	42.38	40.009
33	10	5	56.281	53.328	53.151	52.905	49.971
34	10	6	67.475	64.136	63.734	63.504	59.945
35	10	7	78.761	75.157	74.413	74.125	69.99
36	10	8	90.008	86.636	84.967	84.669	79.949

Mean Absolute Error is 0.026 per cent, and Bias is -0.005 per cent. These figures are also approximations of Monte Carlo simulations error and Bias for whole table 1 and table 2.

Taking the true probabilities for 'not switching' decision into account, we can also reestimate 'flip a coin' decision probabilities. They are in combined column for that decision.

$$P(flip\ a\ coin) = \frac{1}{2}P(switching_{Monte\ Carlo}) + \frac{1}{2}P(no\ switch_{true}) \tag{26}$$

Mean Absulute Error and Bias for this modification of 'flip a coin' decision are:

$$MAE_{modified} = \frac{1}{36}\sum_{i=1}^{36} \mid p_{MC\,i} - p_{true_i} \mid \tag{27}$$

$$p_{MC_i} = \frac{1}{2}P(switching_{Monte\ Carlo_i}) + \frac{1}{2}P(no\ switch_{true_i}) \tag{28}$$

$$p_{true_i} = \frac{1}{2}P(switching_{true_i}) + \frac{1}{2}P(no\ switch_{true_i}) \tag{29}$$

$$MAE_{modified} = \frac{1}{36}\sum_{i=1}^{36} \mid \frac{1}{2}p_{switching_{MC_i}} - \frac{1}{2}p_{switching_{true_i}} \mid \tag{30}$$

Since (24) and (25) are approximations of MAE and Bias for whole table 1 and table 2:

$$MAE_{modified} \doteq \frac{1}{2}MAE \tag{31}$$

$$Bias_{modified} \doteq \frac{1}{2}Bias \tag{32}$$

Modified probabilities show table 3 and table 4.

Changedp column describes probabilities in case player changed her mind. *No changep* column describes probabilities in case player did not change her mind. *Flipp* column describes probabilities in case player made 'flip a coin' decision. *Ttp* column describes probabilities in case player is applying 'tic-toc' decision. *Ottp* column describes probabilities in case player is applying 'opposite tic-toc' decision.

'Tic-toc' decision is a kind of decision when player makes decision according to last known decision. If last decision was to change mind and was successful, she would also change mind. If last decision was to change mind and was not successful, she would not change mind. 'Opposite tic-toc' decision is a kind of decision when player also makes decision according to last known decision. If last decision is to change mind and was successful, she will not change mind. If the last decision was to change mind and was not successful, she would change mind. She will do the opposite, because she is expecting that situation will change in next turn. Another interesting fact is how probabilities are increasing with increasing prices count.

Tables 1 and 2 show that changing mind is the best decision. 'Flip a coin' decision probabilities always lie between 'change mind' probabilities and 'do not change mind' probabilities. 'Flip a coin' decision for 3 doors and 1 prize is those 50 per cent, which caused a lot of discussion about the Monty Hall problem. Those 50 per cent is those 50 per cent, what common sense tells that the probability should be. 'Tic-toc' decision is better than 'flip a coin' decision.

Table 3 Combined results

No	Door	Prize	Changep	No changep Monte Carlo	No changep True	Flipp	Flipp combined
1	3	1	66.645	33.355	33.333	50.061	49.989
2	4	1	37.545	24.992	25	31.278	31.273
3	4	2	75	49.993	50	62.54	62.5
4	5	1	26.704	19.979	20	23.35	23.352
5	5	2	53.287	40.047	40	46.657	46.644
6	5	3	80.014	59.99	60	69.997	70.007
7	6	1	20.86	16.649	16.667	18.791	18.764
8	6	2	41.67	33.303	33.333	37.493	37.502
9	6	3	62.565	50.053	50	56.253	56.283
10	6	4	83.273	66.719	66.667	75.042	74.97
11	7	1	17.155	14.286	14.286	15.767	15.721
12	7	2	34.268	28.581	28.571	31.477	31.42
13	7	3	51.457	42.892	42.857	47.114	47.157
14	7	4	68.592	57.163	57.143	62.917	62.868
15	7	5	85.661	71.399	71.429	78.553	78.545
16	8	1	14.564	12.49	12.5	13.562	13.532
17	8	2	29.171	25.009	25	27.097	27.086
18	8	3	43.77	37.492	37.5	40.609	40.635
19	8	4	58.368	49.989	50	54.252	54.184
20	8	5	72.953	62.504	62.5	67.716	67.727
21	8	6	87.511	74.955	75	81.251	81.256

Table 4 Combined results

No	Door	Prize	Changep	No changep Monte Carlo	No changep True	Flipp	Flipp combined
22	9	1	12.709	11.063	11.111	11.919	11.91
23	9	2	25.378	22.286	22.222	23.85	23.8
24	9	3	38.05	33.302	33.333	35.62	35.692
25	9	4	50.793	44.429	44.444	47.603	47.619
26	9	5	63.514	55.543	55.556	59.574	59.535
27	9	6	76.224	66.624	66.667	71.459	71.446
28	9	7	88.856	77.803	77.778	83.35	83.317
29	10	1	11.263	9.974	10	10.663	10.632
30	10	2	22.496	20.021	20	21.273	21.248
31	10	3	33.796	29.957	30	31.856	31.898
32	10	4	44.983	40.009	40	42.497	42.492
33	10	5	56.281	49.971	50	53.151	53.141
34	10	6	67.475	59.945	60	63.734	63.738
35	10	7	78.761	69.99	70	74.413	74.381
36	10	8	90.008	79.949	80	84.967	85.004

'Opposite tic-toc' decision is worse than 'flip a coin' decision, but better than 'do not change mind' decision. If player decides to not change her mind, it is the worst decision ever. 'Flip a coin' decision is the third best decision of five different options, and it confirms the old truth: 'If you do not know what you should do, make a 'flip a coin' decision and you will not decide bad'. Research

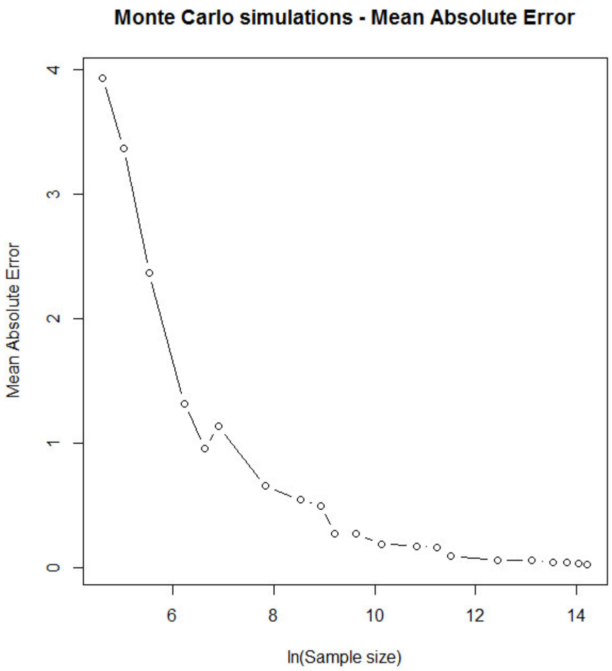

Fig. 1 Dependence between Mean Absolute Error and sample size in Monty Hall problem, calculated from probabilities obtained from Monte Carlo simulations and exact probabilities, in case contestant does not change her mind.

also showed that people should change their mind, because individuals who do not change their mind have the lowest probability for success. Second worst chance for success have individuals who are speculating too much - opposite tic-toc decision. The best chance for success have individuals who change their minds.

Finally, we can plot dependence between Monte Carlo simulations sample size and Mean Absolute Error as $MAE = f(sample\ size)$, and also dependence between Monte Carlo simulations sample size and Bias as $Bias = f(sample\ size)$. Dependence show Figure 1 and Figure 2.

4 Conclusion

We explored veridical type paradoxes using conditional probabilities, Bayes' rule and mainly with Monte Carlo simulations. We discovered why a lot of discussion of Monty Hall problem took place in 90's, and why so many researchers thought that the probability would be 50 per cent [7]. Those 50 per cents represent 'flip a coin' decision. We discovered that contestants should change their mind, and it is the best decision among all the explored decisions.

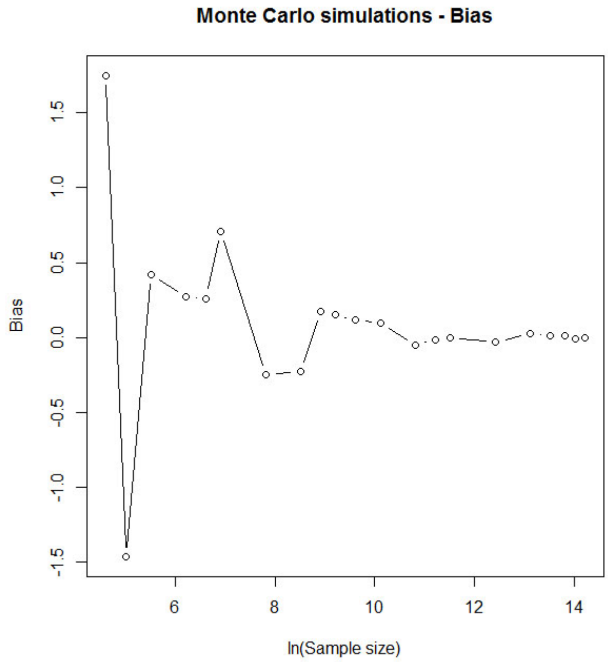

Fig. 2 Dependence between Bias and sample size in Monty Hall problem, calculated from probabilities obtained from Monte Carlo simulations and exact probabilities, in case contestant does not change her mind.

If contestants do not change their minds, it is the worst decision ever. Second worst decision is, if contestants are speculating too much - opposite tic-toc decision. On the other hand, appropriate speculating is possitive - tic-toc decision, being second best decision. Research showed a lot about Monty Hall problem and also about life. The aim of the research has been carried out.

References

1. T. Carsey, J. Harden: Monte Carlo Simulation and Resampling Methods for Social Science. SAGE Publications, London 2014.
2. J. S. Dagpunar: Simulation and Monte Carlo with application in finance and MCMC. Jogn Wiley & Sons, Chichester 2007.
3. D. Koller, N. Friedman: Probabilistic Graphical Models Principles and Techniques. The MIT Press, London 2009.
4. S. Page: Diversity and Complexity. Princeton University Press, Woodstock 2010.
5. https://en.wikipedia.org/wiki/Bertrand's_box_paradox
6. https://en.wikipedia.org/wiki/Three_Prisoners_problem
7. https://en.wikipedia.org/wiki/Monty_Hall_problem