13-06-2019

Simulación de la martingala

Estudio de diversos factores

ELENA GARRIDO JIMÉNEZ
TÉCNICAS DE SIMULACIÓN

Dibuja la gráfica que relaciona la probabilidad de ganar con los beneficios esperados

En primer lugar, introduzcamos el código realizado para realizar este primer ejercicio. Este código concretamente se utiliza para obtener los beneficios en función de los distintos parámetros introducidos, llamando a la función martingale presentada en el último apartado de este documento.

```
# Study of benefits as a function of the probability of winning the bi
d
benefits <- 0
# Sequence of probabilities
probabilities <- seq(0, 1, by=0.1)
for (i in probabilities){
  benefits <- c(benefits, mean(replicate(1000, martingale(500, 500, 10
0, i*10, "Won money"))))
}
benefits <- benefits[-1]
plot(benefits, type="l", xaxt = "n", main = "Expected benefits as a fu
nction of winning probability")
axis(1, at=c(1:11), labels = probabilities)

dataframe <- data.frame(probabilities)
dataframe[,2] <- benefits

print(dataframe[2])
```

Veamos cómo se relaciona la probabilidad con los beneficios esperados cambiando algunos parámetros de la función para poder realizar una comparación más precisa. En primer lugar, variemos el número de iteraciones; posteriormente, variemos la apuesta inicial y estudiemos el fenómeno que nos pide el enunciado ante distintas condiciones. El parámetro que se mantendrá constante siempre será el dinero disponible, siendo siempre 500. Hay que tener en cuenta que la gráfica representa el valor medio que se obtiene tras mil iteraciones.

A continuación, se presentan las gráficas con los siguientes parámetros constantes: apuesta inicial = 10 y dinero disponible = 500.

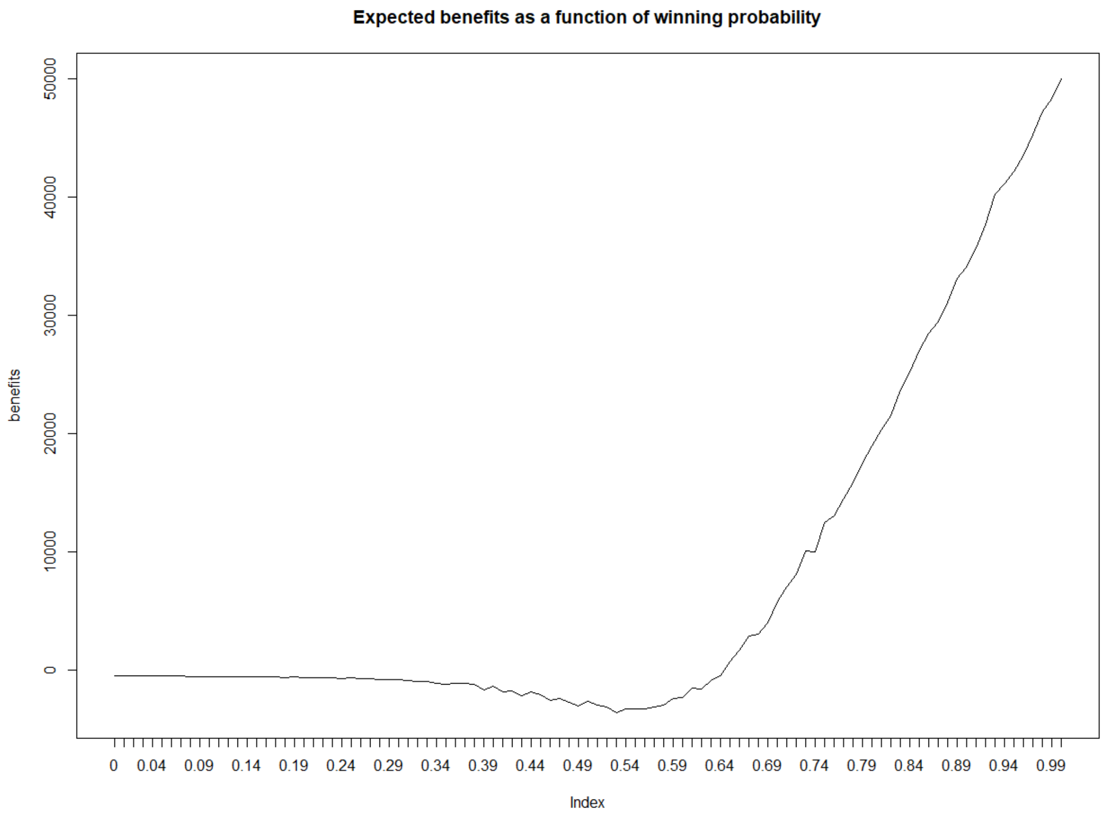

Figura 1. Máximas jugadas realizadas: 100

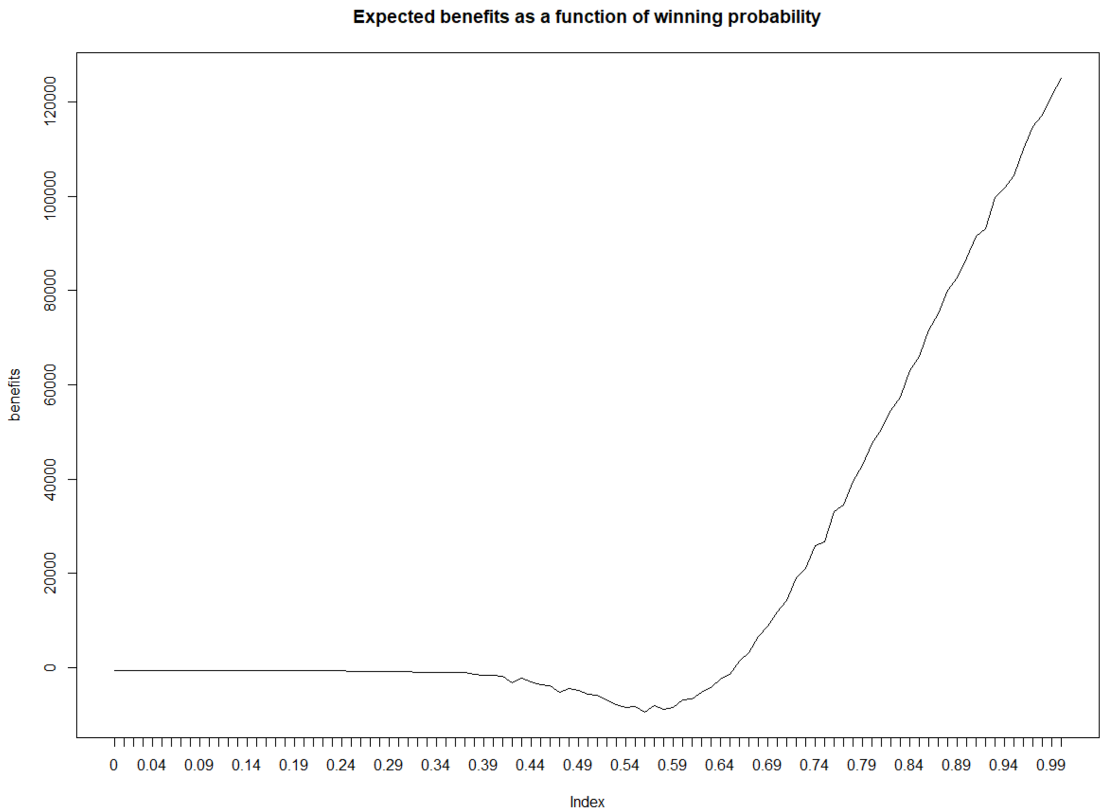

Figura 2. Máximas jugadas realizadas: 250

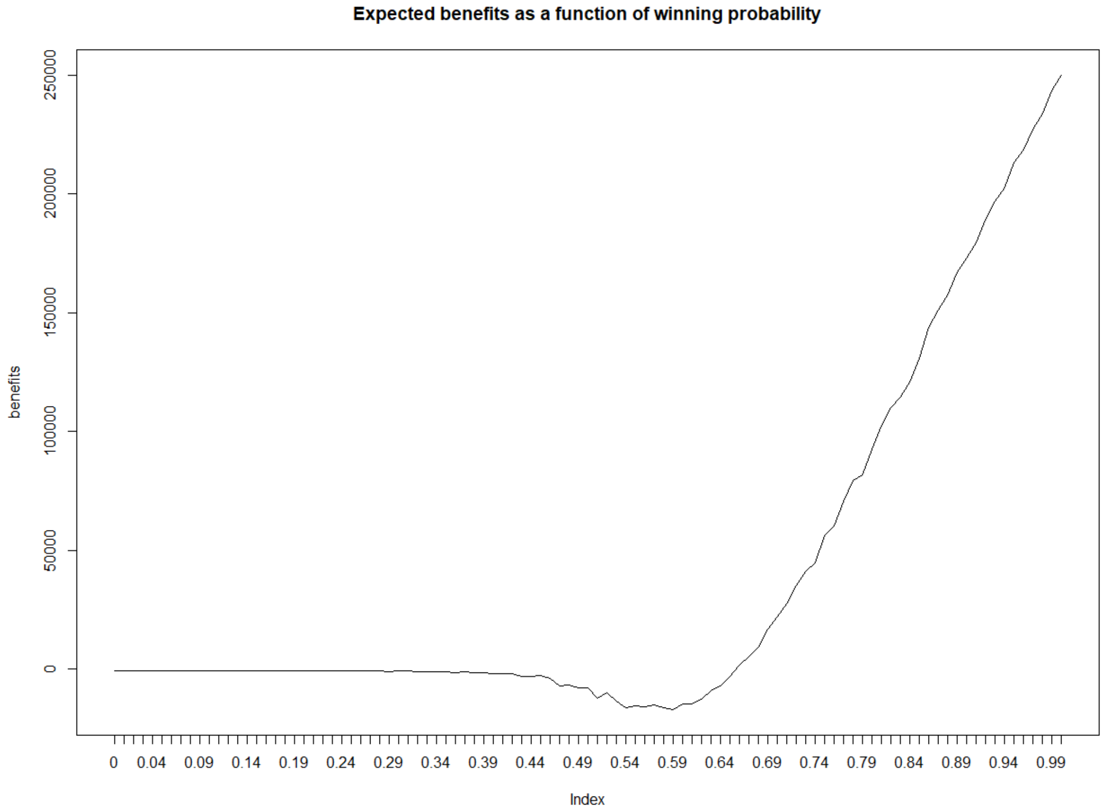

Figura 3. Máximas jugadas realizadas: 500

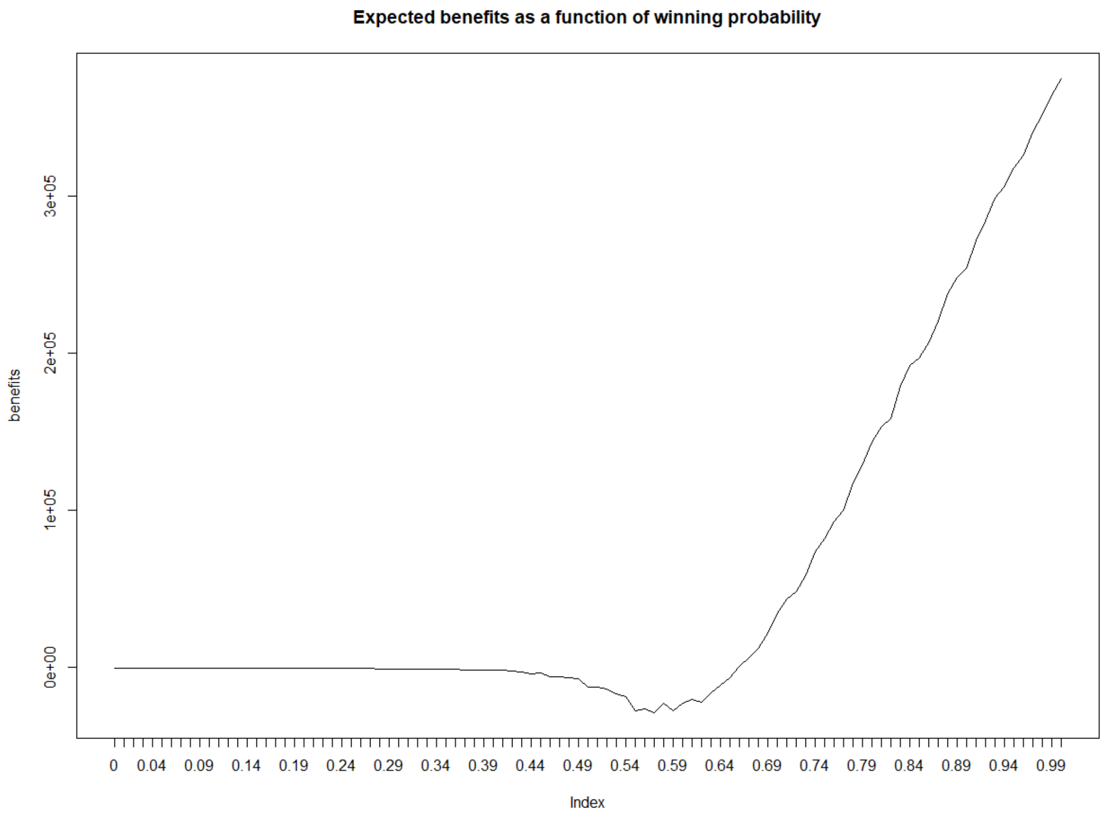

Figura 4. Máximas jugadas realizadas: 750

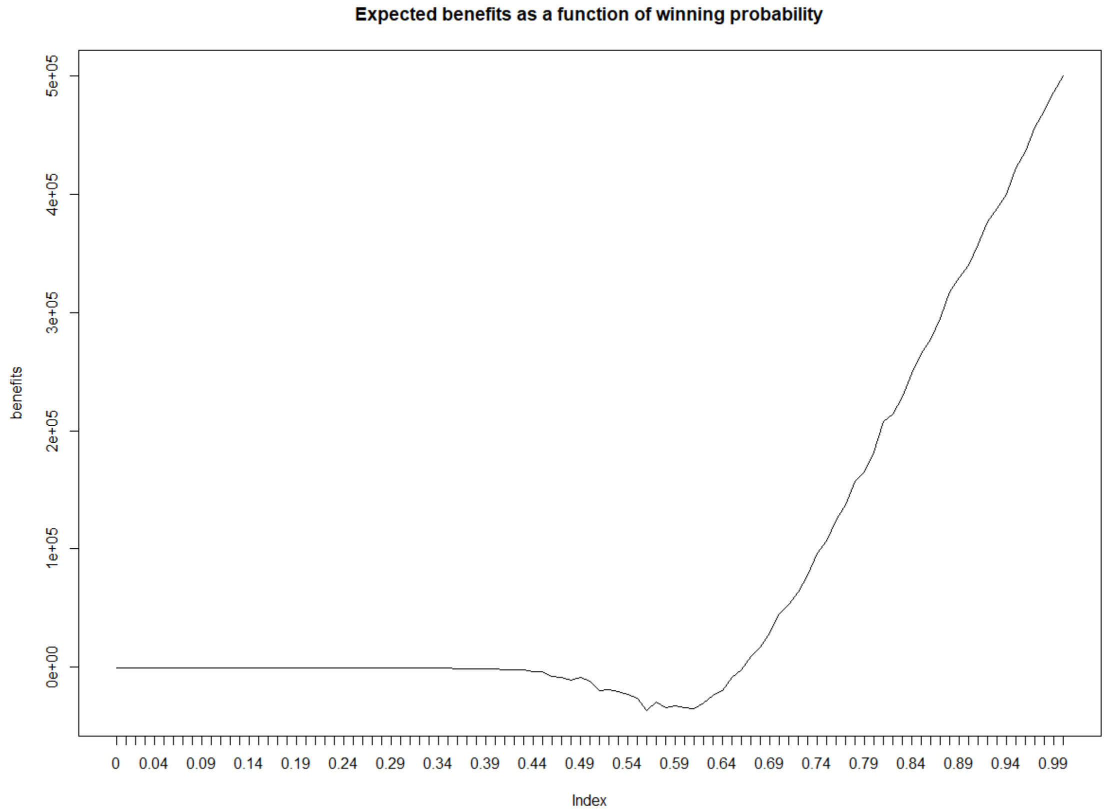

Figura 5. Máximas jugadas realizadas: 1000

A continuación, se presentan las gráficas con los siguientes parámetros constantes: máximas jugadas = 250 y dinero disponible = 500.

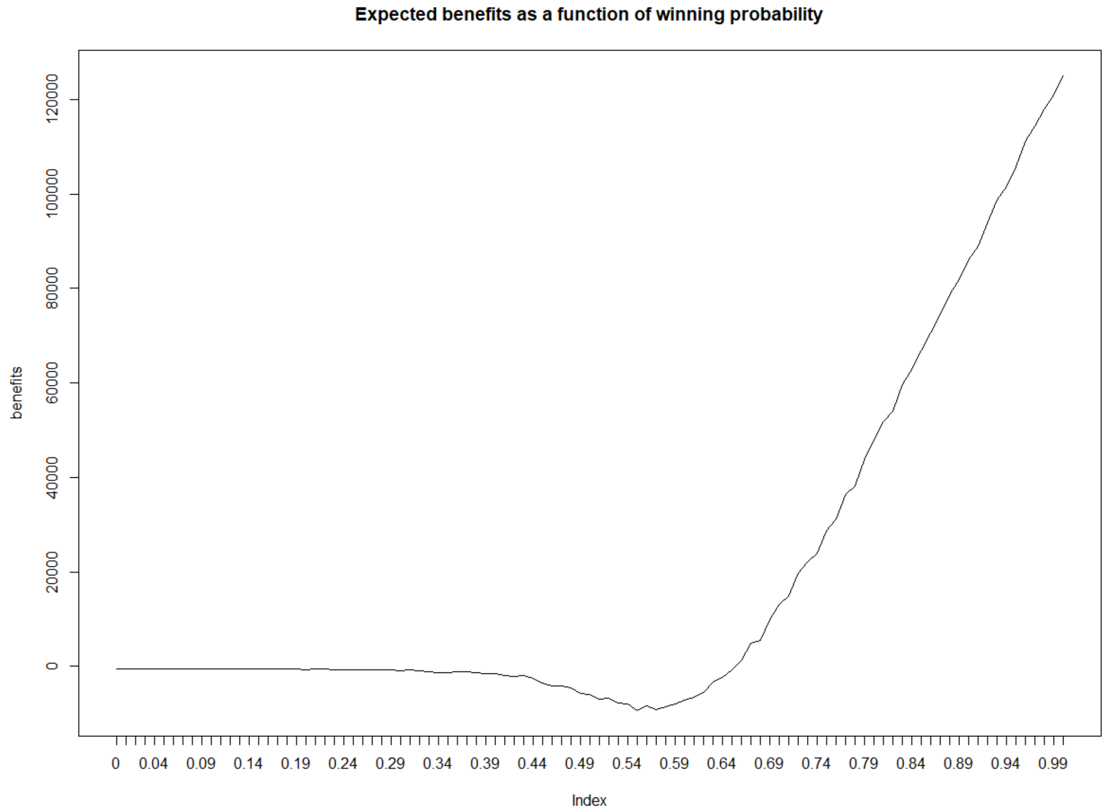

Figura 6. Apuesta inicial simulada: 10

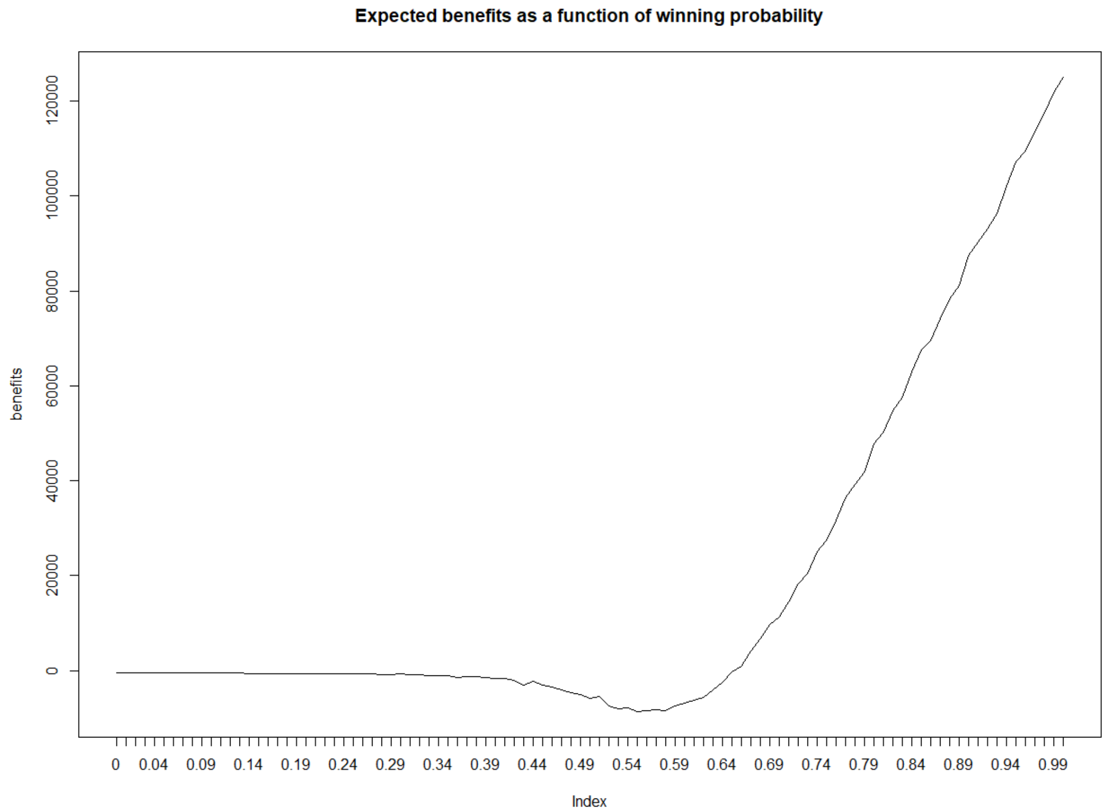

Figura 7. Apuesta inicial simulada: 25

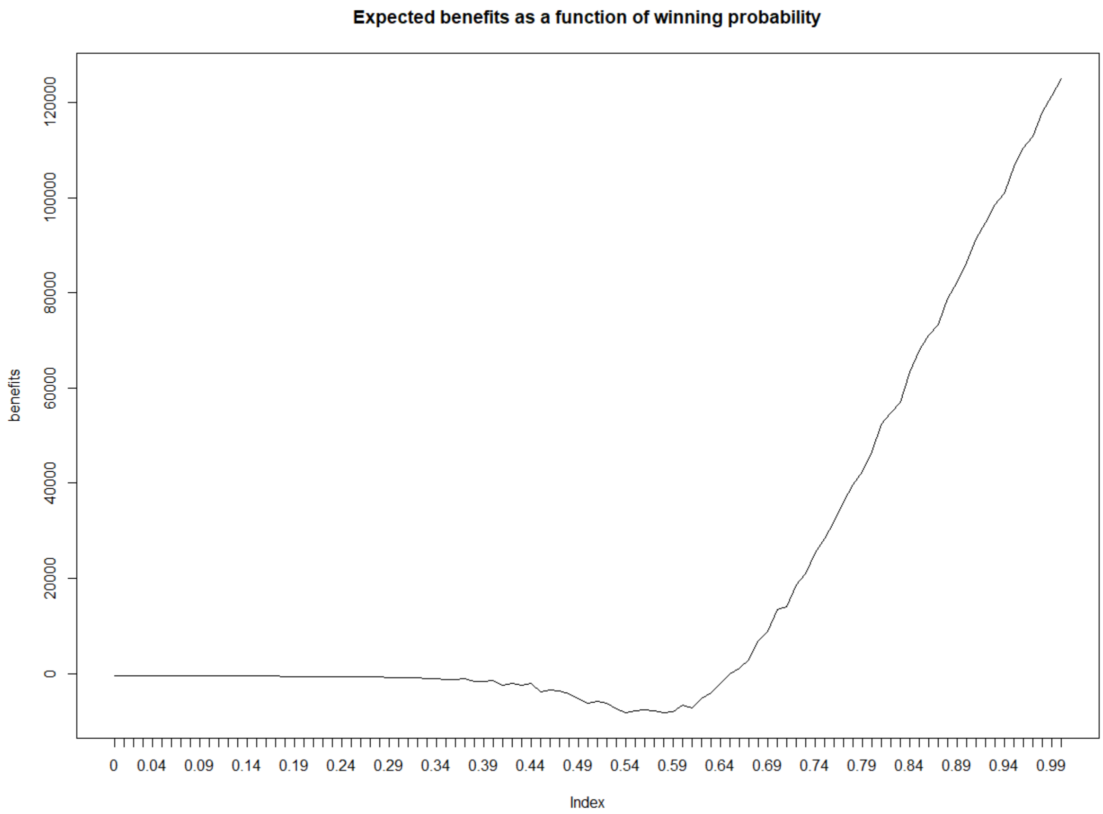

Figura 8. Apuesta inicial simulada: 50

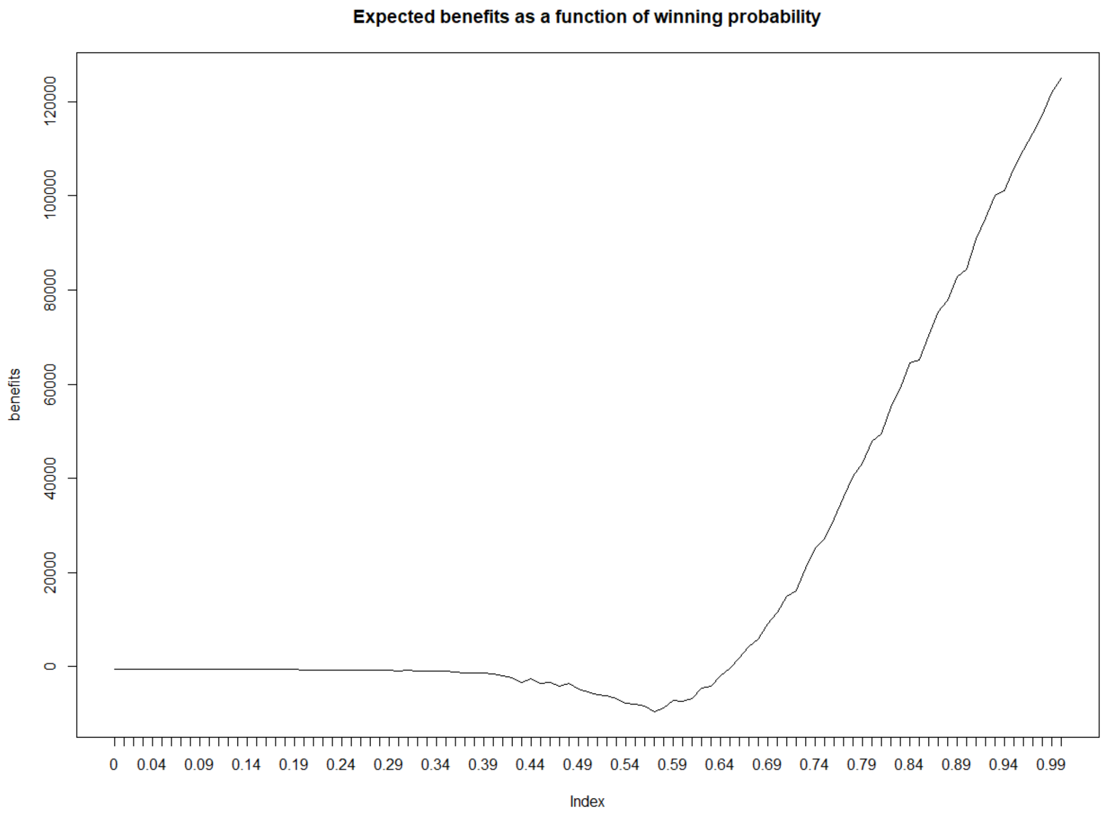

Figura 9. Apuesta inicial simulada: 100

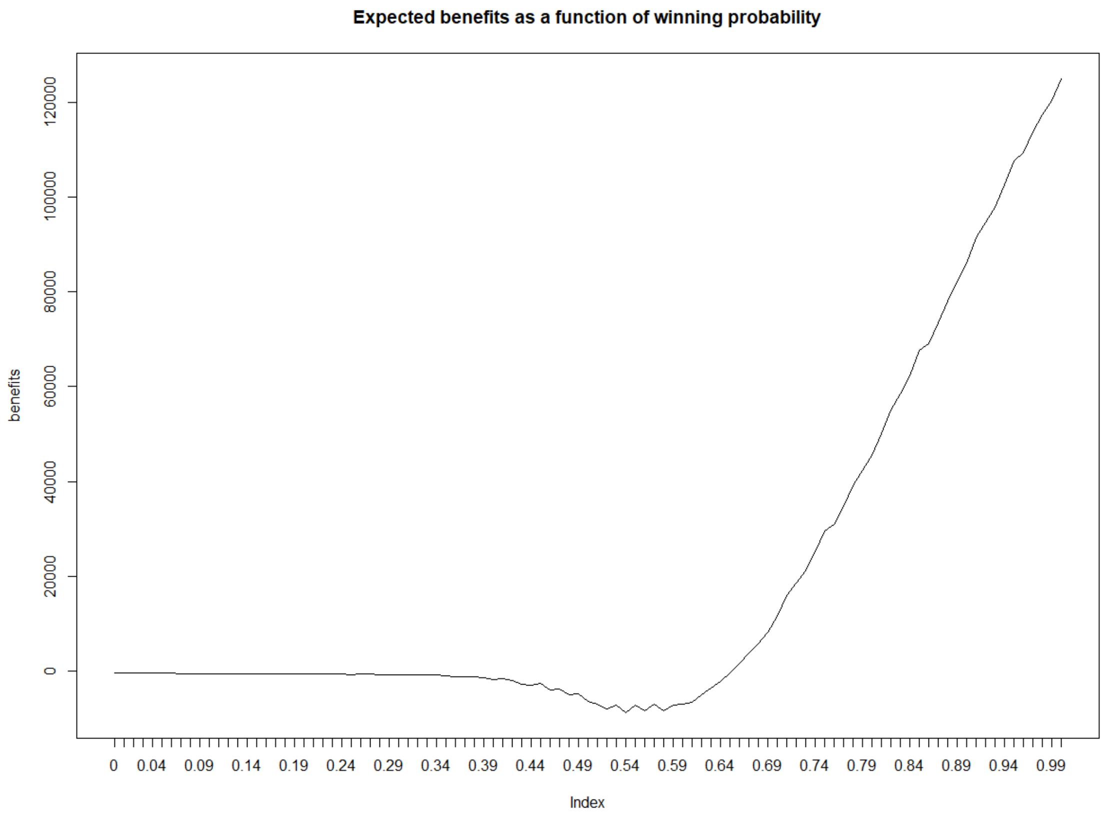

Figura 10. Apuesta inicial simulada: 250

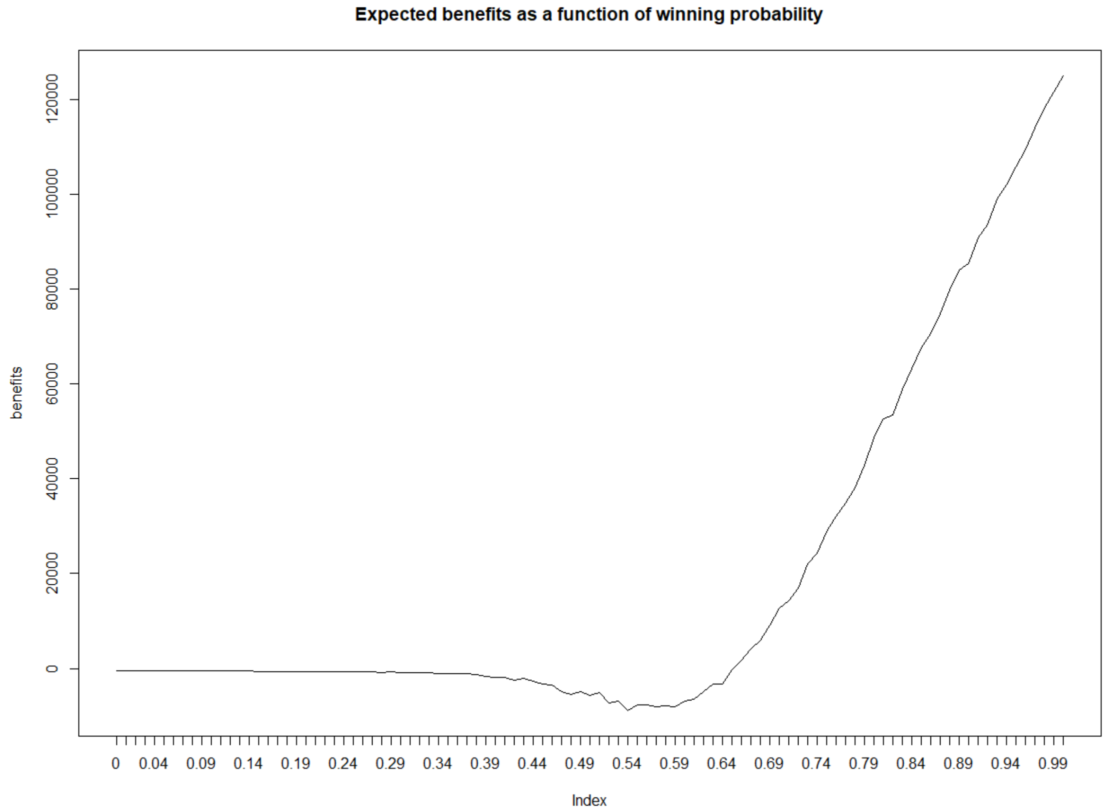

Figura 11. Apuesta inicial simulada: 500

Tal y como se constata en los diferentes casos ilustrados, incluso alterando en función de las distintas variables, encontramos que las pérdidas son frenadas por el propio algoritmo (dado que no se continúa apostando si el dinero disponible es menor a "0"), acumulando menores pérdidas dada a esta condición del código. El máximo número de pérdidas se concentra cuando la probabilidad de ganar ronda el valor de 0.5, y van aumentando a partir de ese punto las ganancias a partir de ese mínimo global, obteniendo beneficios a partir de un valor de probabilidad de ganar de 0.65 aproximadamente.

Analiza cómo afectan el número de iteraciones, la cantidad inicial y la apuesta a los beneficios esperados

Antes de comentar los resultados obtenidos, merece la pena destacar la modificación realizada al código anterior, donde para poder obtener los resultados deseados, añadimos lo siguiente:

```
dataframe <- data.frame(probabilidades)
dataframe[,2] <- beneficios

print(dataframe[2])
```

Tabla 1:

Beneficios obtenidos en función de la probabilidad de ganar y del número máximo de jugadas (iteraciones), tomando como constantes los valores de la apuesta inicial (10) y la cantidad inicial de dinero disponible (500).

Probabilidad de ganar	Iteraciones (número máximo de jugadas)					
	10	25	50	100	250	500
0	-500.0	-500.0	-500.0	-500.0	-500.0	-500.0
0.1	-557.5	-540.5	-554.0	-554.0	-545.0	-547.0
0.2	-632.0	-643.5	-622.0	-614.5	-622.5	-642.0
0.3	-630.0	-791.5	-835.0	-873.0	-857.5	-944.5
0.4	-631.0	-986.0	-1280.0	-1492.0	-1660.5	-1842.5
0.5	-353.5	-924.0	-1733.0	-3423.5	-5876.5	-7879.5
0.6	134.5	-54.5	-826.0	-2701.5	-7032.5	-16818.5
0.7	1005.5	2022.5	3504.0	6291.0	11765.0	22830.0
0.8	2279.5	5117.5	9955.5	19132.5	45951.5	94603.0
0.9	3417.5	8899.0	17530.0	34310.0	87563.5	172620.0
1	5000.0	12500.0	25000.0	50000.0	125000.0	250000.0

Examinemos los resultados obtenidos en la Tabla 1. De ella se desprenden las siguientes conclusiones: en primer lugar, podemos observar que las pérdidas iniciales son siempre las mismas, dado que el algoritmo está programado para que no se continúe apostando en caso de no quedar dinero disponible.

Continuando con los resultados observados, podemos ver que las pérdidas se acentúan para los mismos valores de la probabilidad de ganancia a medida que aumenta el número máximo de jugadas permitidas en nuestra función, y lo mismo podemos decir de las ganancias: éstas se acentúan para los mismos valores de probabilidad a medida que aumentan el número de iteraciones. También podemos observar que el máximo de pérdidas varía en función del número de iteraciones: con tan solo 10 iteraciones, el mayor número de pérdidas se sitúa en una probabilidad de ganancia de "0", mientras que al realizar 500 iteraciones, la mayor pérdida se localiza en una probabilidad de ganancia de 0.5 (esto se debe a la programación comentada en el párrafo anterior). Finalmente, podemos observar dos fenómenos: primero, que la tendencia a obtener beneficios se observa en la probabilidad 0.7, y segundo, que los beneficios y pérdidas siguen una tendencia lineal tan solo cuando nos hallamos en una probabilidad de ganancia de 1 (en otro caso, el ratio entre dos beneficios esperados ante la misma probabilidad y distintas iteraciones no es constante a medida que aumentamos o disminuimos la probabilidad de ganar).

Tabla 2:

Beneficios obtenidos en función de la probabilidad de ganar y cantidad inicial de dinero disponible, tomando como constantes los valores de la apuesta inicial (10) y el número de iteraciones (100).

Probabilidad de ganar	Cantidad inicial					
	50	100	250	500	1000	2500
0	-50.00	-100.0	-250.00	-500.0	-1000	-2500.0
0.1	-55.50	-110.6	-275.25	-550.0	-1098	-2725.0
0.2	-63.25	-128.1	-305.50	-641.5	-1248	-3142.5
0.3	-85.35	-174.7	-405.00	-863.5	-1550	-4372.5
0.4	-145.00	-263.4	-715.00	-1400.5	-3517	-8170.0
0.5	-318.00	-688.6	-1479.00	-2966.5	-6624	-15800.0
0.6	-236.75	-428.4	-1174.75	-2436.5	-5008	-9917.5
0.7	523.90	1085.8	2689.50	5461.5	9950	27837.5
0.8	1852.80	3802.0	10114.00	18807.5	37803	96980.0
0.9	3521.00	6955.6	17295.50	34939.0	69628	174460.0
1	5000.00	10000.0	25000.00	50000.0	100000	250000.0

Los resultados obtenidos en la Tabla 2 comparten con la Tabla 1 la característica de obtener beneficios (valores positivos) a partir de una probabilidad de ganancia de 0.7, al igual que limitar las pérdidas por las propiedades del algoritmo anteriormente comentadas, mas encontramos algunos resultados propios.

En este caso, los beneficios no aumentan de manera sustancial a medida que aumenta la cantidad inicial, pero sí lo hace las pérdidas, que aumentan sustancialmente a medida que lo hace la cantidad inicial apostada. Como podemos comprobar igualmente, la cantidad inicial apostada no influye en los máximos beneficios esperados a medida que la probabilidad de ganancia se acerca a 1, siendo constante en este valor los beneficios obtenidos. Finalmente, el valor máximo de pérdidas se desplaza a medida que aumenta la cantidad inicial de dinero disponible, de valores máximos de pérdidas situados en 0.5 con una cantidad inicial apostada de 50, hasta valores máximos de pérdidas situados en una probabilidad de ganancias de 0.3. Aun así, como hemos comentado anteriormente, la recuperación (beneficios esperados que toman valores positivos) se produce en 0.7, como hemos comentado anteriormente.

Tabla 3:

Beneficios obtenidos en función de la probabilidad de ganar y la apuesta inicial, tomando como constantes los valores de la cantidad inicial disponible (500) y el número de iteraciones (100).

Probabilidad de ganar	Apuesta inicial					
	10	25	50	100	250	500
0	-500.0	-500.0	-500.0	-500.0	-500.0	-500.0
0.1	-549.5	-547.5	-557.0	-548.0	-563.0	-566.0
0.2	-625.0	-622.5	-629.5	-624.5	-617.0	-612.0
0.3	-842.5	-836.5	-851.5	-808.0	-894.5	-829.0
0.4	-1449.5	-1498.5	-1508.5	-1511.5	-1567.5	-1600.5
0.5	-2923.0	-2885.0	-2989.0	-3169.0	-3115.0	-3062.5
0.6	-2152.5	-2199.5	-2057.0	-2487.0	-2124.5	-2151.0
0.7	5651.5	5694.5	5632.0	5188.0	5889.5	4964.5
0.8	19155.5	18598.0	19468.0	19156.5	18827.5	18652.5
0.9	35322.0	34309.0	34620.5	34295.5	35035.0	34624.0
1	50000.0	50000.0	50000.0	50000.0	50000.0	50000.0

Por último, comentemos los resultados de la simulación obtenido en la Tabla 3. De nuevo, existen varias características en común respecto a las otras dos simulaciones comentadas anteriormente: la primera de ellas es la obtención de beneficios positivos a partir de una probabilidad de ganar de 0.7, y la segunda es el crecimiento únicamente lineal de los beneficios en un valor de la probabilidad de ganar de 1.0, siendo no lineal en el resto de valores.

En este caso concreto, merece la pena destacar que los valores iniciales de “beneficios negativos” (pérdidas) no se mantienen constantes como en la Tabla 1, sino que sucede algo similar a lo ocurrido en la Tabla 2. Finalmente, el máximo de pérdidas alcanzadas va cambiando en cuanto a su correspondiente valor de probabilidad de ganancia a medida que nos desplazamos por la tabla de valores. Así, el valor máximo de pérdidas lo encontramos en una apuesta inicial de 10 y en una probabilidad de ganar de 0.4, mientras que el máximo en una apuesta inicial de 500 se encuentra en una probabilidad de 0.5.

De todo lo anterior, se desprenden las siguientes conclusiones:

- La localización del máximo de pérdidas varía no solo en función de la probabilidad de ganancias, sino también del número máximo de jugadas realizadas, la cantidad inicial apostada y la cantidad inicial de dinero disponible (siendo directamente proporcional a las mismas para todas las variables contempladas)
- La localización del punto de recuperación (momento en el que se obtienen beneficios en lugar de pérdidas) en función de la probabilidad no se ve afectado por ninguna de las tres variables estudiadas. Para corroborar esta propiedad, sería necesaria mayor resolución (un estudio más detallado) en los valores de las probabilidades estudiadas y en los valores de las tres variables contempladas en este estudio de simulación

- La tendencia lineal de variación de los beneficios esperados en función de cada una de las variables contempladas se sitúa ante una probabilidad de ganancia de 1, variando para casos anteriores: esto significa que, para una probabilidad de ganancia de 1, si invertimos 10 veces más cantidad de dinero, podemos esperar obtener 10 veces más beneficios. Sin embargo, si invertimos 10 veces más cantidad de dinero y tenemos una probabilidad de ganancia de 0.4, las pérdidas no serán 10 veces mayores, todo ello gracias a la propiedad del algoritmo de dejar de apostar cuando no se tiene dinero disponible

Con qué probabilidad el juego se hace más largo (más iteraciones)

Para estudiar este apartado, comenzaremos modificando la función para que nos dé el máximo de iteraciones posibles alcanzadas, de manera que la nueva función resultante sería la siguiente:

```
# Study of the maximum number of plays that can be played as a functio
n of the winning probability
num_plays <- 0
probabilities <- seq(0,1, by= 0.01)
for (i in probabilities){
  num_plays <- c(num_plays, mean(replicate(1000, martingale(10, 500, 1
000, i*10, "Number of plays"))))
}
num_plays <- num_plays[-1]
dataframe_num_plays <- data.frame(probabilities)
dataframe_num_plays[,2] <- num_plays

print(dataframe_num_plays[2])
plot(num_plays, xaxt="n", type="l", main = "Number of maximum plays as
a function of winning probability")
axis(1, at=c(1:101), labels=probabilities)
```

Acto seguido, estudiemos cómo varía el número de máximas jugadas permitidas por el juego en función de la probabilidad. Como constantes, la apuesta inicial será de 10 unidades y la cantidad inicial de divisas disponibles será de 500 unidades. Para no simular un bucle infinito, el número máximo de jugadas permitidas serán diez mil (10^4), de modo que podemos suponer que si se alcanza esta cifra, el juego tendería a prolongarse en el tiempo sin cesar. Tal y como puede contemplarse en la imagen, la función resultante sigue la tendencia de una regresión logística, incrementando sustancialmente la cantidad de jugadas permitidas a partir de una probabilidad de ganancia de 0.5. De este modo, podemos observar que, a mayor probabilidad de ganancia, se produce un incremento desigual: desde una probabilidad de ganancia de 0 a 0.4, el incremento en el número de jugadas apenas es apreciable, incrementándose notoriamente desde una probabilidad de ganar de 0.5 hasta una probabilidad de ganar de 0.7, para de nuevo desacelerar su crecimiento hasta una probabilidad de ganar de 1.

Así, podemos confirmar que los resultados son similares a partir de una probabilidad de ganancia de 0.86, hallándose el máximo teórico en una probabilidad de 1.

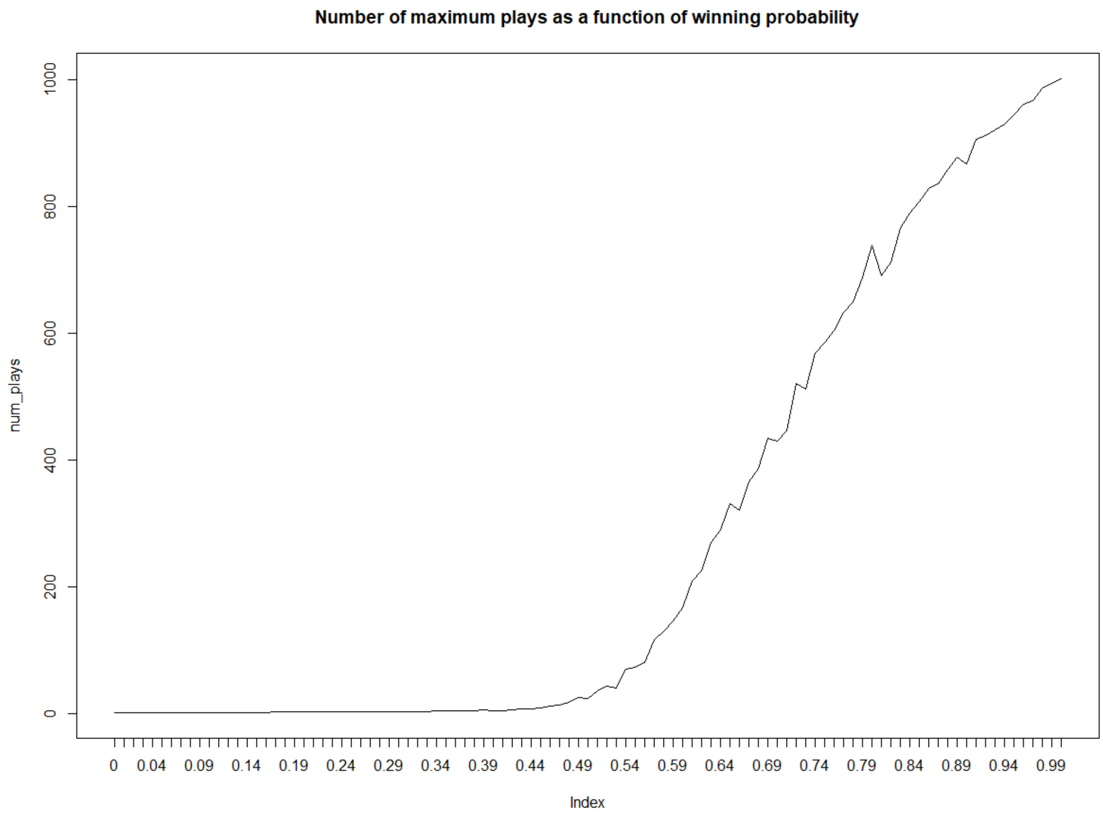

Figura 12. Número de jugadas posibles realizadas en función de la probabilidad de ganancia.

Para corroborar lo anterior, a continuación se presentan los valores de probabilidad y del número de jugadas tabulados:

Tabla 4:

Probabilidad de éxito	Número de jugadas realizadas
0	2.000
0.01	2.020
0.02	2.049
0.03	2.065
0.04	2.086
0.05	2.123
0.06	2.133
0.07	2.180
0.08	2.211
0.09	2.240
0.1	2.208
0.11	2.233
0.12	2.369
0.13	2.368
0.14	2.377
0.15	2.420
0.16	2.348
0.17	2.476
0.18	2.610

0.19	2.577
0.2	2.607
0.21	2.674
0.22	2.760
0.23	2.835
0.24	2.922
0.25	2.952
0.26	2.853
0.27	3.146
0.28	3.147
0.29	3.236
0.3	3.343
0.31	3.686
0.32	3.413
0.33	3.643
0.34	4.029
0.35	4.020
0.36	4.575
0.37	4.484
0.38	4.670
0.39	5.668
0.4	5.327
0.41	4.883
0.42	6.679
0.43	7.426
0.44	7.656
0.45	8.664
0.46	11.972
0.47	13.109
0.48	17.653
0.49	25.500
0.5	24.382
0.51	35.773
0.52	43.819
0.53	41.195
0.54	70.292
0.55	72.672
0.56	81.332
0.57	117.232
0.58	129.562
0.59	146.355
0.6	166.902
0.61	208.701
0.62	225.324
0.63	270.163
0.64	288.974
0.65	331.418
0.66	320.913

0.67	365.465
0.68	386.803
0.69	433.882
0.7	429.333
0.71	445.471
0.72	520.821
0.73	511.142
0.74	568.547
0.75	585.327
0.76	603.751
0.77	632.544
0.78	648.779
0.79	687.168
0.8	738.838
0.81	691.319
0.82	711.808
0.83	764.680
0.84	788.556
0.85	806.392
0.86	827.353
0.87	835.570
0.88	858.245
0.89	877.350
0.9	867.179
0.91	905.159
0.92	911.128
0.93	921.137
0.94	930.099
0.95	945.080
0.96	961.052
0.97	966.043
0.98	986.015
0.99	993.010
1	1001.000

Para terminar, convierte el simulador en algo más realista y parecido al juego de verdad.

A continuación, se presenta el código de una versión mucho más interactiva del juego:

```
# Interactive version of the program

# We ask for the necessary data to the user
available_money <- NA
while(is.na(available_money) || available_money <= 0) {
  available_money <- readline(prompt = "Indicate the available money: ")
  available_money <- as.integer(available_money)
}
```

```r
initial_money <- available_money

initial_bid <- NA
while(is.na(initial_bid) || initial_bid > available_money) {
  initial_bid <- readline(prompt = "Insert and initial amount for bidd
ing: ")
  initial_bid <- as.integer(initial_bid)
}

max_plays <- 0
while(is.na(max_plays) || max_plays <= 0) {
  max_plays <- readline(prompt = "Indicate the maximum number of plays
desired: ")
  max_plays <- as.integer(max_plays)
}

prob_win <- sample(0:10, size=1)  # For the interactive version of the
code, the probability of winning is random

# The martingale function is defined. It receives as argument the init
ial bid, the initial amout of total money available,
# and the maximum number of plays

martingale <- function(available_money, initial_bid, max_plays, prob_w
in, output) {
  num_plays <- 1
  bid <- initial_bid
  won_money <- 0

  # While the total amount of available money is more than 0, and the
number of plays id less or equal to the maximum
  # number of plays
  while (available_money > 0 & num_plays <= max_plays) {
    #   You win or lose randomly
    win_lose <- sample(c("win", "lose"), prob=c(prob_win,10-prob_win),
size = 1)
    #   If you win
    if (win_lose=="win") {
      # The total amount of money adds the initial quantity
      available_money <- bid + available_money
      # The bid is still the initial bid
      bid <- initial_bid
      # The won money adds the bid
      won_money <- won_money + bid
    }
    # If you lose
    else {
      # You lose the bid from the available money you've got
      available_money <- available_money - bid
      # The bid is substracted from the won money
      won_money <- won_money - bid
      # Bid is doubled
      bid <- bid * 2
    }
```

```r
    ##  num_plays increases by 1
    num_plays <- num_plays + 1
  }
  if (output == "Won money"){
    return(won_money)
  } else {
    return(num_plays)
  }
}

# The function is called and the result is provided
cat(sprintf("The won money is: %5.3f", martingale(available_money, ini
tial_bid, max_plays, prob_win, "Won money")))
```