LICENCIATURA EN ESTADÍSTICA

LA GUÍA DEL APOSTADOR

"El problema del Bandido Multibrazo"

Autores: Franco Santini - Nicolas Gamboa - Andrés Roncaglia

Docentes: Ignacio Evangelista - Tomás Capretto

2024

Tabla de contenidos

Introducción

En los casinos uno de los juegos más llamativos son las máquinas tragamonedas o tragaperras. Estas son conocidas en el mundo de las apuestas por ser fuertemente adictivas y dónde las personas pierden más dinero. Con el fin de que estas personas pierdan lo menos posible se estudiará la eficacia de distintas estrategias de juego. A este problema se lo conoce como "Bandido Multibrazo" (*Multi-armed bandit*).

La principal incógnita se encuentra en descubrir hasta que punto conviene "explotar" por sobre "explorar" o viceversa. Donde explotar refiere a dar prioridad a elegir lo que uno sabe o cree que sabe que funciona, mientras que explorar refiere a ser propenso a salir de la "zona de confort" y buscar nuevas alternativas.

Metodología

Se plantea que se tienen 3 máquinas tragamonedas con distintas probabilidades de ganar una unidad monetaria, las cuales son:

- $\theta_A = 0.30$
- $\theta_B = 0.55$
- $\theta_C = 0.45$

Cada día se hará una sola tirada en una máquina que será elegida según la estrategia, durante los 366 días del año 2024, con el objetivo de ganar la mayor cantidad de dinero posible, teniendo en cuenta que no cuesta dinero jugar, es decir, solo podemos ganar dinero.

Se asume además que la creencia inicial de la probabilidad de ganar para cada máquina sigue una distribución $Beta(2, 2)$, la cual se puede observar a continuación.

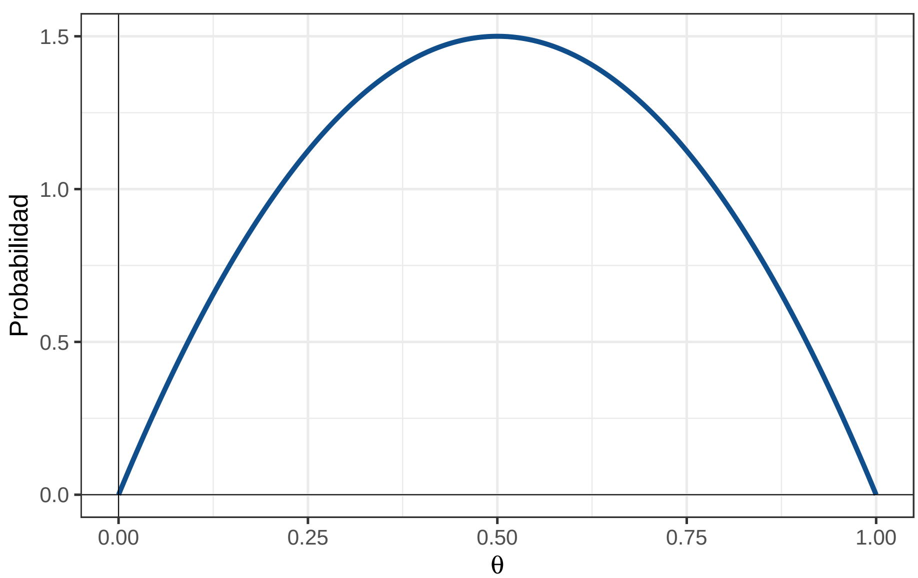

Figura 1: Distribución a priori de la probabilidad de éxito inicial para todas las máquinas.

Resultados

Para obtener una noción de cuanto se ganaría en promedio en un año de juego con la mejor máquina (máquina B), se simulan 1000 secuencias de 366 juegos.

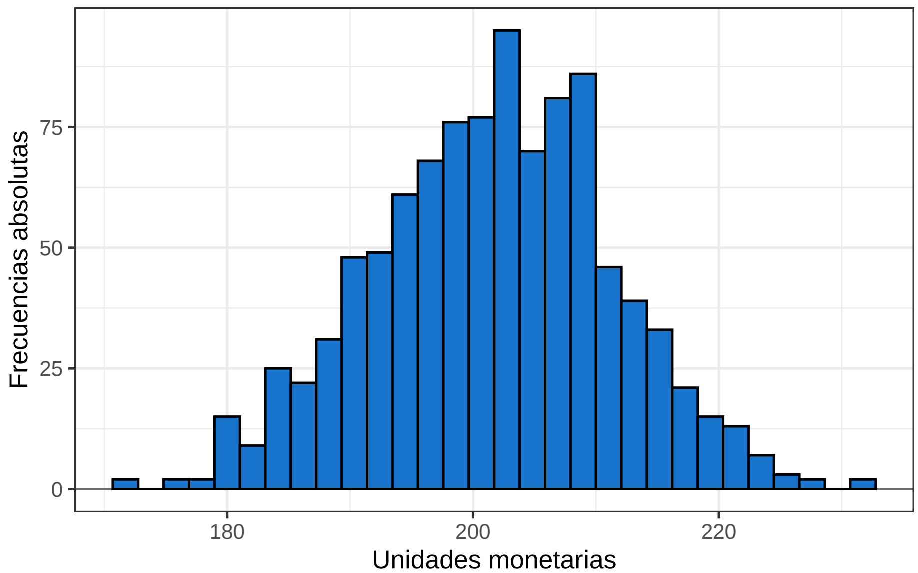

Figura 2: Distribución de las unidades monetarias ganadas en un año con la máquina B.

Resultando 201.58 una estimación de la ganancia esperada durante un año de juego con la máquina B.

Estrategias

Sin embargo las personas no tienen información acerca de las probabilidades de éxito de las máquinas, por lo que se diseñaron un conjunto de estrategias para descubrir con cúal se obtienen mejores resultados.

Para esto se hacen n simulaciones de los 366 días del año mediante la siguiente función:

Función simulación

```
simulacion <- function(metodo, n = 1, param = 0.2) {

  if(!(metodo %in% names(estrategias))){
    stop("El método introducido no está especificado")}
  if(n<1 || (n != round(n))){
    stop("La cantidad de simulaciones debe ser un número natural")}
  if(metodo == "softmax" && param <= 0){
    stop("Para el método softmax el parámetro de temperatura debe ser mayor que 0")}
  if(metodo == "e_greedy" && (param < 0 || param > 1)){
    stop("Para el método e-greedy el parámetro epsilon debe estar entre 0 y 1")}
```

```r
# Se crean los vectores donde se guardaran las ganancias y maquinas de los 366
# dias para cada repeticion. Se crea tambien una futura lista llamada sim que
# va a guardar las ganancias y las maquinas usadas en las repeticiones

ganancias <- NULL
maquinas <- NULL
sim <- NULL

# Hacemos repeticiones de n años
for (rep in 1:n) {

  # Se crean los vectores que guardaran las ganancias y las maquinas usadas en
  # cada dia
  ganancia <- numeric(length = 366)
  maquina <- NULL
  for (dia in 1:366) {

    # Cada dia la maquina sera elegida segun el criterio de alguna de las
    # estrategias diseñadas
    argumentos <- switch (metodo,
                          "al_azar" = list(),
                          "gcto" = list(maquina, ganancia),
                          "gcpp" = list(maquina, ganancia),
                          "e_greedy" = list(maquina, ganancia, param),
                          "softmax" = list(maquina, ganancia, param),
                          "thompson" = list(maquina, ganancia),
                          "upper_bound" = list(maquina, ganancia, param)
                          )

    maquina_dia <- do.call(estrategias[[metodo]], argumentos)

    # Una vez elegida la maquina se simula una bernoulli con la
    # probabilidad de la maquina y se guarda el resultado
    ganancia[dia] <- rbinom(1, size = 1, prob = prob_reales[maquina_dia])

    # Se guarda ademas la maquina utilizada
    maquina[dia] <- maquina_dia
  }
```

```
    # Se guardan en matrices las ganancias y las maquinas de los 366 dias
    # de las diferentes repeticiones
    ganancias <- cbind(ganancias, ganancia)
    maquinas <- cbind(maquinas, maquina)

  }

  # Se transforman las matrices en tibbles, y se les asignan nombres a las
  # columnas segun la repeticion (año)
  ganancias <- as_tibble(ganancias)
  maquinas <- as_tibble(maquinas)

  colnames(ganancias) <- paste0("Rep_",1:1000)
  colnames(maquinas) <- paste0("Rep_",1:1000)

  # Se añaden los tibbles a la lista
  sim[["Ganancias"]] <- ganancias
  sim[["Maquinas"]] <- maquinas

  # Devuelve la lista con los éxitos o fracasos para cada año en cada tirada
  # y la máquina utilizada
  sim
}
```

Completamente al azar

Consiste en elegir cada día una máquina al azar sin importar los resultados que se hayan obtenido en el día anterior. Esta estrategia se concentra únicamente en "explorar" pero nunca considera "explotar" como una posibilidad.

Para elegir la máquina con la que se va a jugar en el día se hizo uso de la siguiente función:

Función al_azar

```r
al_azar <- function() {
  # Elige una maquina al azar
  sample(names(prob_reales), 1)
}
```

Simulando los 366 días de juego:

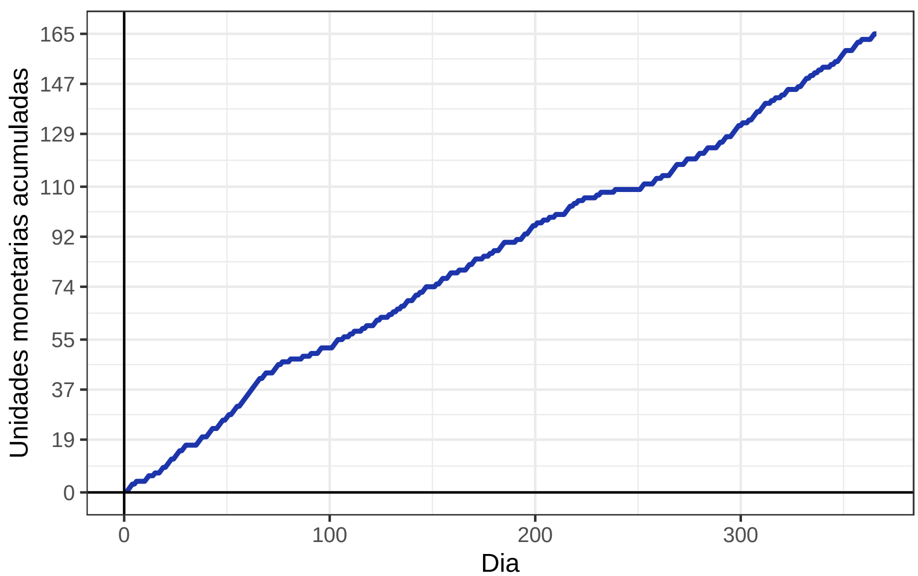

Figura 3: Unidades monetarias acumuladas durante un año con la estrategia al azar.

Se observa que se ganaron 165 unidades monetarias en el año.

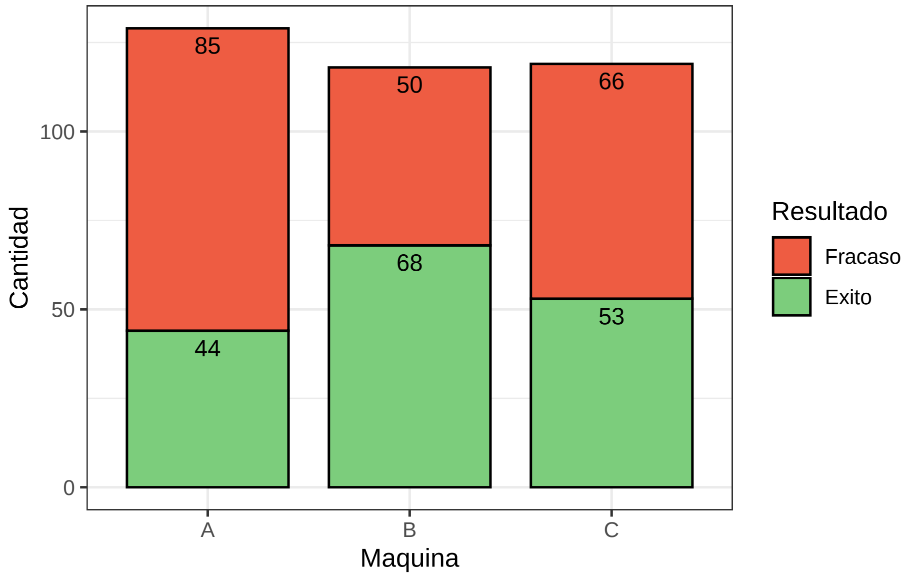

Figura 4: Frecuencia, éxito y fracaso para cada máquina durante un año con la estrategia al azar.

La máquina más jugada fue la A

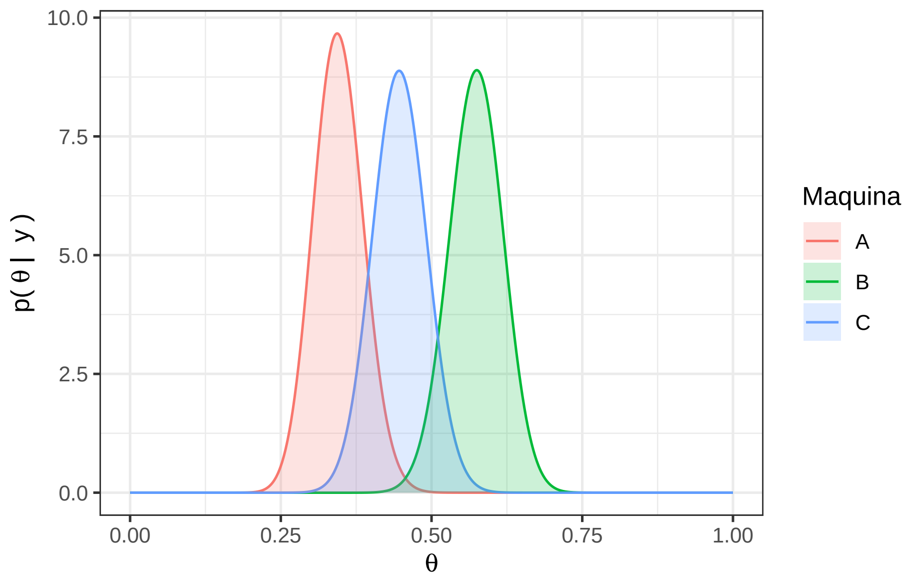

Figura 5: Probabilidades de éxito para cada máquina a posteriori luego de un año con la estrategia al azar.

Y así resultan las distribuciones de la probabilidad de éxito en cada máquina a posteriori.

Ventajas de la estrategia:

- No se corre el riesgo de concentrarse más en una máquina que produciría menos ganancias que las demás.

Desventajas de la estrategia:

- No tiene en cuenta la información que se adquiere a través de los días.

Greedy con tasa observada

La máquina con la que se juega será la que tenga la mayor tasa de éxito hasta el día en que se juega. Esta estrategia se concentra demasiado en "explotar" y casi nada en "explorar".

Para elegir la máquina con la que se va a jugar en el día se hizo uso de la siguiente función:

ℹ Función gcto

```r
gcto <- function(maquina, ganancia) {

  # Las tasas son 0 en la primera iteracion
  tasa <- numeric(3)


  # Cuando una maquina no se usa ni una vez, la tasa va a seguir siendo 0
  # Cuando se juegue al menos una vez su tasa será el numero de veces que gano
  # en la maquina dividido la cantidad de veces que jugo con la maquina
  if (sum(maquina == "A") > 0) {
    tasa[1] <- sum(ganancia[maquina == "A"])/sum(maquina == "A")
  }
  if (sum(maquina == "B") > 0) {
    tasa[2] <- sum(ganancia[maquina == "B"])/sum(maquina == "B")
  }
  if (sum(maquina == "C") > 0) {
    tasa[3] <- sum(ganancia[maquina == "C"])/sum(maquina == "C")
  }

  # Se asigna el nombre de la maquina para cada tasa
  names(tasa) <- c("A","B","C")

  # Verifica cual es la tasa mas alta y la elige, si hay una sola elige la
  # maquina a la cual le pertenezca esa tasa, si hay varias maquinas con la misma
  # tasa elige una al azar entre las que tengan la mayor tasa

  if (sum(tasa == max(tasa)) == 1) {
    maq <- names(tasa[which.max(tasa)])
  } else {
    maq <- sample(names(tasa[tasa == max(tasa)]), size = 1)
  }

  return(maq)
}
```

Simulando los 366 días de juego:

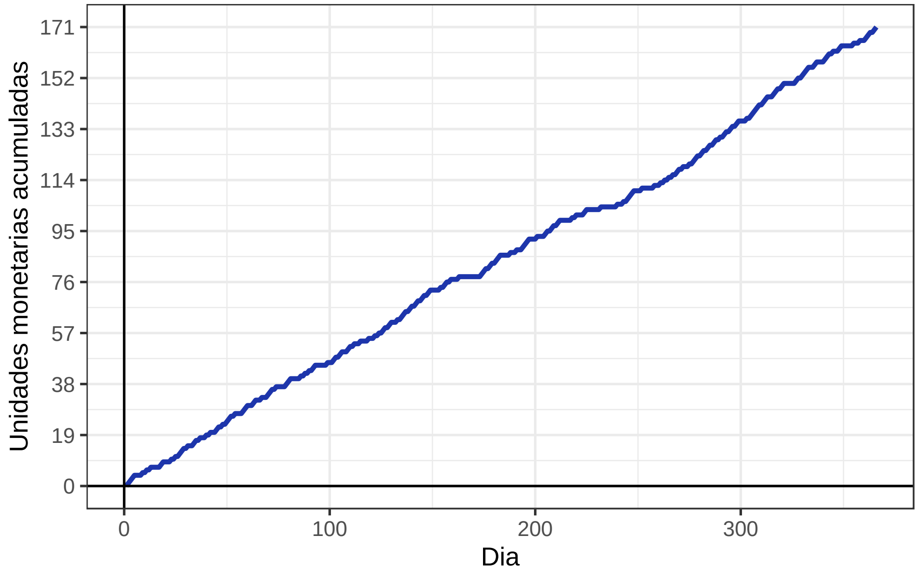

Figura 6: Unidades monetarias acumuladas durante un año con la estrategia greedy con tasa observada.

Se observa que se ganaron 171 unidades monetarias en el año.

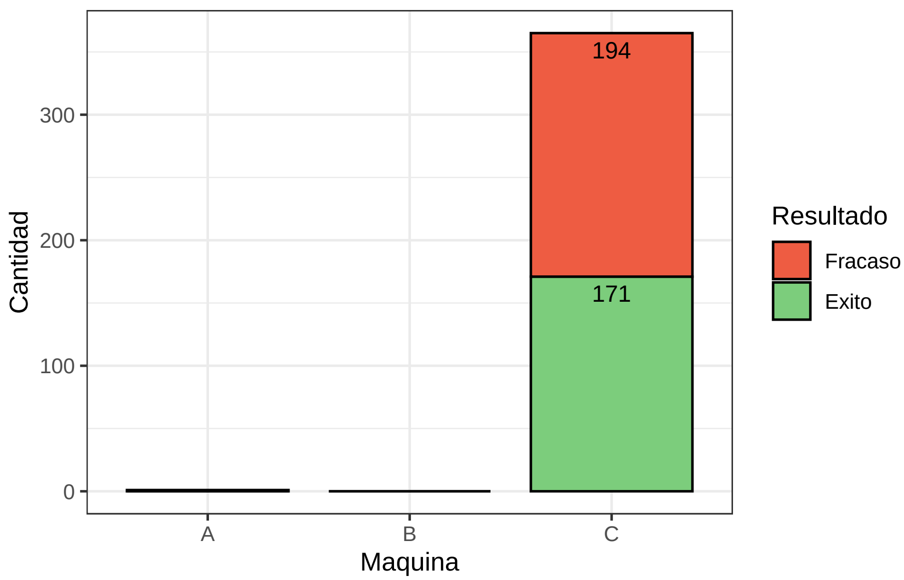

Figura 7: Frecuencia, éxito y fracaso para cada máquina durante un año con la estrategia greedy con tasa observada.

La máquina más jugada fue la C

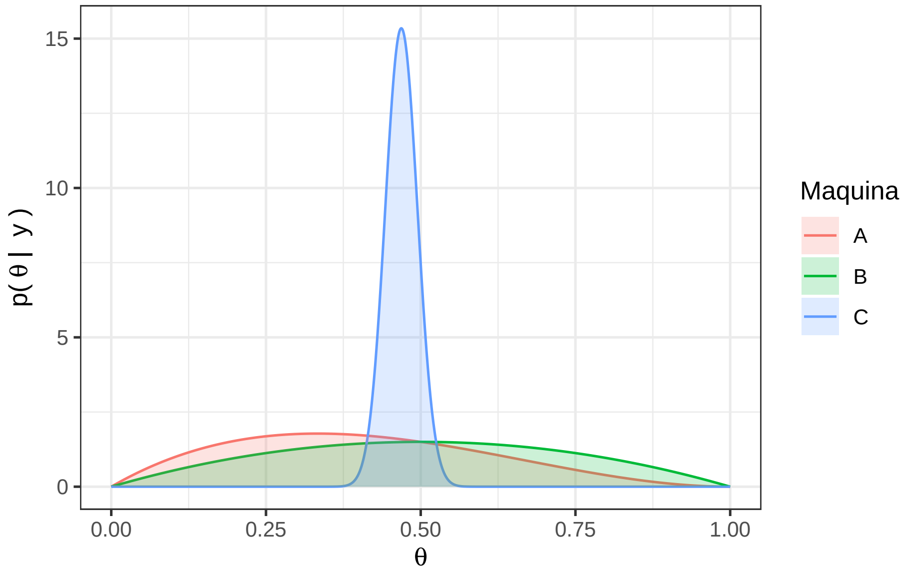

Figura 8: Probabilidades de éxito para cada máquina a posteriori luego de un año con la estrategia greedy con tasa observada.

Y así resultan las distribuciones de la probabilidad de éxito en cada máquina a posteriori.

Ventajas de la estrategia:

- En algunos escenarios es posible elegir siempre la mejor máquina.

Desventajas de la estrategia:

- Una vez que se gane con una máquina, se elegirá siempre esa misma.
- Es posible elegir explotar la peor máquina.

Greedy con probabilidad a posteriori

Partiendo de las creencias iniciales, luego de cada tirada y considerando las realizadas hasta el momento, se redefinen las probabilidades de éxito esperadas de cada máquina y se selecciona la mejor.

Para elegir la máquina con la que se va a jugar en el día se hizo uso de la siguiente función:

Función gcpp

```
gcpp <- function(maquina, ganancia) {

  # Argumentos de las distribuciones a priori de los parámetros
  a1 <- 2; b1 <- 2; c1 <- 2; a2 <- 2; b2 <- 2; c2 <- 2

  # Si hay al menos una tirada con la máquina, modifica los parámetros de la
  # distribución a posteriori según corresponda
  if (sum(maquina == "A") > 0) {
    a1 <- 2 + sum(ganancia[maquina == "A"])
    a2 <- 2 + abs(sum(ganancia[maquina == "A"] - 1))
  }
  if (sum(maquina == "B") > 0) {
    b1 <- 2 + sum(ganancia[maquina == "B"])
    b2 <- 2 + abs(sum(ganancia[maquina == "B"] - 1))
  }
  if (sum(maquina == "C") > 0) {
    c1 <- 2 + sum(ganancia[maquina == "C"])
    c2 <- 2 + abs(sum(ganancia[maquina == "C"] - 1))
  }

  # Se redefinen las probabilidades de éxito esperadas de cada máquina
  prob <- c(a1/(a1+a2), b1/(b1+b2), c1/(c1+c2))
  names(prob) <- c("A","B","C")

  # Selecciona la mejor, o una al azar entre las mejores si tienen el mismo valor
  if (sum(prob == max(prob)) == 1) {
    maq <- names(prob[which.max(prob)])
  } else {
    maq <- sample(names(prob[prob == max(prob)]), size = 1)
  }

  return(maq)

}
```

Simulando los 366 días de juego:

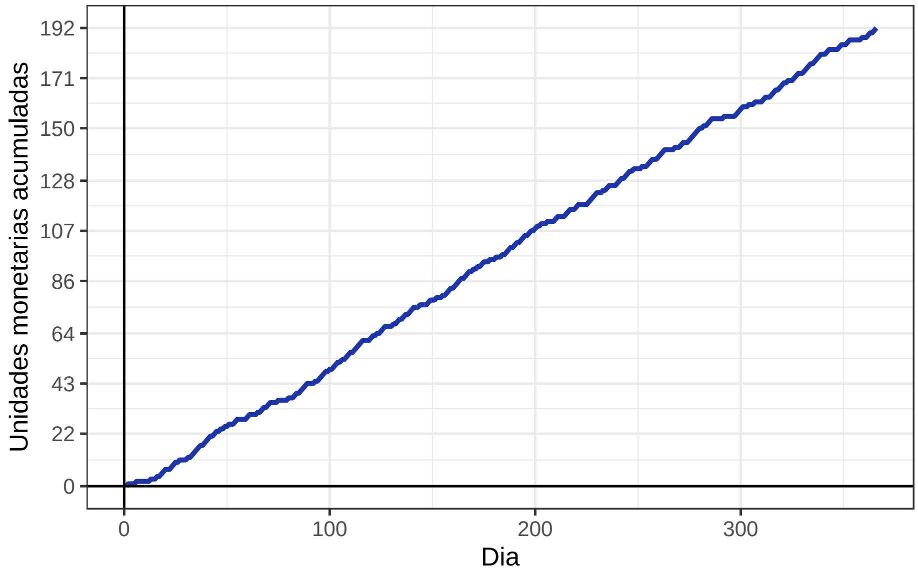

Figura 9: Unidades monetarias acumuladas durante un año con la estrategia greedy con probabilidad a posteriori.

Se observa que se ganaron 192 unidades monetarias en el año.

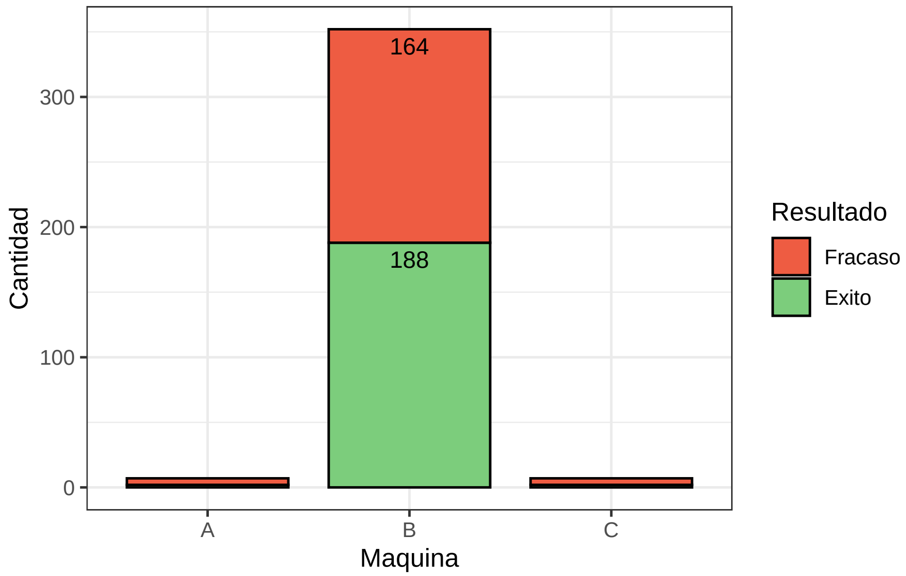

Figura 10: Frecuencia, éxito y fracaso para cada máquina durante un año con la estrategia greedy con probabilidad a posteriori.

La máquina más jugada fue la B

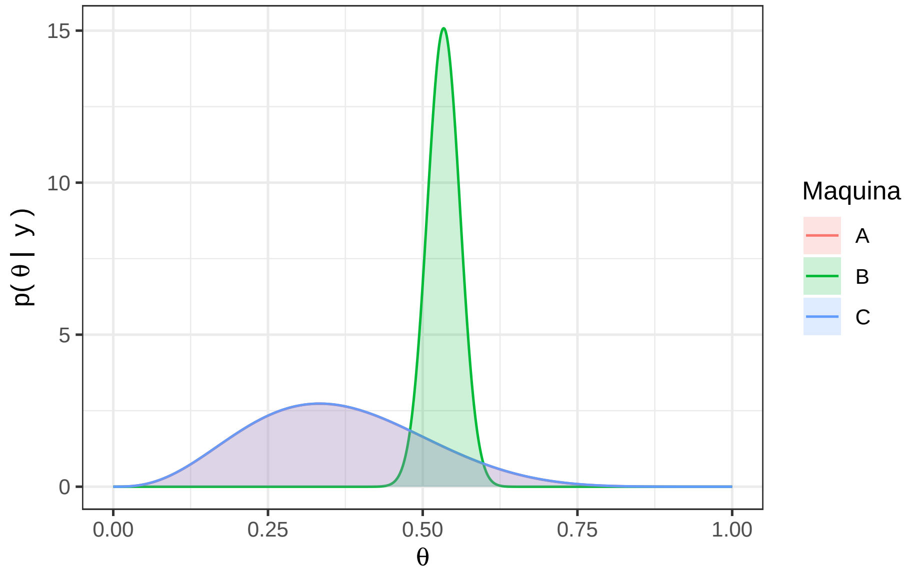

Figura 11: Probabilidades de éxito para cada máquina a posteriori luego de un año con la estrategia greedy con probabilidad a posteriori.

Y así resultan las distribuciones de la probabilidad de éxito en cada máquina a posteriori.

Ventajas de la estrategia:

- Se tiene en cuenta toda la informacíon obtenida hasta el momento.
- A comparación del método anterior, el riesgo de explotar demasiado una máquina incorrecta es mucho menor.

Desventajas de la estrategia:

- Si una buena máquina, en un principio arroja muchos fracasos seguidos, existe el riesgo de que nunca vuelva a ser elegida.
- Aunque de manera menos frecuente que en el caso anterior, existe la posibilidad de explotar la máquina incorrecta.

ϵ-Greedy con tasa observada

Se selecciona con una probabilidad de $1 - \epsilon$ la máquina que hasta el momento tiene la mayor tasa de éxito observada, o con una probabilidad de ϵ, una al azar. ϵ es un parámetro de exploración, cuando es igual a 0 la estrategia es igual a "Greedy con tasa observada", cuando es igual a 1 es igual a la estrategia "Al azar".

Para elegir la máquina con la que se va a jugar en el día se hizo uso de la siguiente función:

Función e_greedy

```
e_greedy <- function(maquina, ganancia, param) {
  # Las tasas son 0 en la primera iteracion
  tasa <- numeric(3)

  # Cuando una maquina no se usa ni una vez, la tasa va a seguir siendo 0
  if (sum(maquina == "A") > 0) {
    tasa[1] <- sum(ganancia[maquina == "A"])/sum(maquina == "A")
  }
  if (sum(maquina == "B") > 0) {
    tasa[2] <- sum(ganancia[maquina == "B"])/sum(maquina == "B")
  }
  if (sum(maquina == "C") > 0) {
    tasa[3] <- sum(ganancia[maquina == "C"])/sum(maquina == "C")
  }

  # Se asigna el nombre de la maquina para cada tasa
  names(tasa) <- c("A","B","C")
  # Se selecciona entre explorar o explotar
  aleatorio <- runif(1)

  # Si el valor aleatorio es menor o igual a epsilon elige una máquina al azar
  # en caso contrario elige a la que tenga mayor tasa observada
  probs <- c(0,0,0)
  if (aleatorio <= param) {
    return(sample(names(tasa),size = 1))
    } else {
      for (i in 1:3) {
        if((tasa == max(tasa))[i]){
          probs[i] <- 1/sum(tasa == max(tasa))}
        }
      return(sample(names(tasa),size = 1,prob = probs))
    }
  }
```

Dado que ϵ puede tomar cualquier valor en el intervalo $[0;1]$, mediante simulaciones buscaremos el valor de este que maximice la cantidad de unidades monetarias ganadas en un año.

Búsqueda del valor óptimo de ϵ.

```
# Se seleccionan unos cuantos epsilons a probar
eps <- seq(0.075,0.25,.025)


final <- NULL

# Para cada epsilon se hacen mil simulaciones de un año de 366 días
for (i in 1:length(eps)) {
  temp <- simulacion("e_greedy",n = 1000,param = eps[i])
  final <- cbind(final, apply(temp$Ganancias,2,sum))
}

# Cada columna tiene las ganancias totales en 1000 años con un epsilon
# Se nombran las columnas según el epsilon con el que se hicieron las simulaciones
colnames(final) <- eps

# Se observan las medias y los desvios
apply(final, 2, mean)
apply(final, 2, sd)

# Se calcula la probabilidad de que las ganancias obtenidas en un año con un epsilon
# sean mayores que con otro y nos quedamos con aquel con el que sea más probable
# obtener mayores ganancias
for (i in 1:length(eps)) {
  for (j in 1:length(eps)) {
    print(mean(final[,i]>=final[,j]))
  }
}
```

Siendo X una variable aleatoria que representa la cantidad de unidades monetarias ganadas en total en un año de juego y estimando las $P(X_\epsilon > X_{\epsilon'})$ para los ϵ planteados, concluímos que el valor de $\epsilon / P(X_\epsilon > X_{\epsilon'}) > P(X_\epsilon < X_{\epsilon'}) \quad \forall \epsilon' \neq \epsilon$ es $\epsilon = 0,125$. Lo que significa que con un ϵ de 0.125 es más probable obtener una mayor cantidad de unidades monetarias totales que con otros valores.

Simulando los 366 días de juego con un ϵ de 0.125:

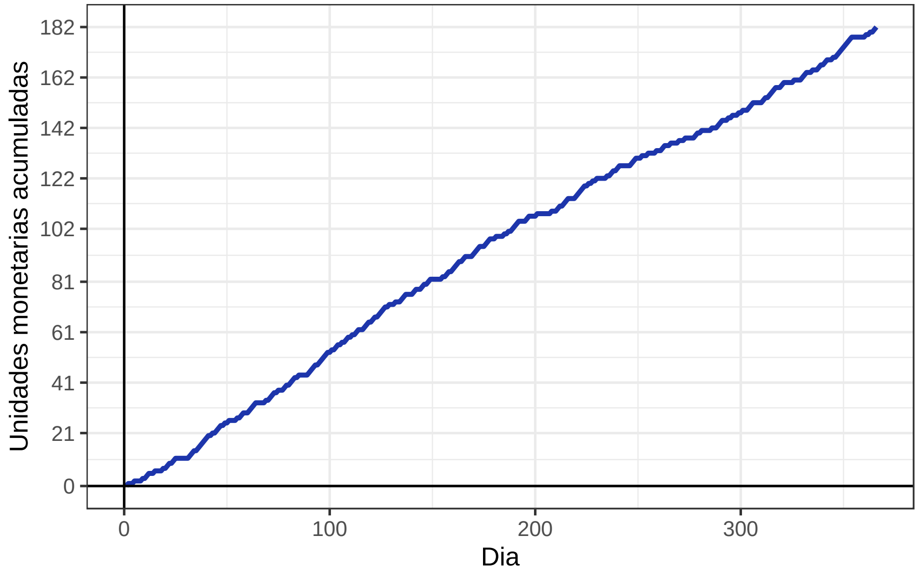

Figura 12: Unidades monetarias acumuladas durante un año con la estrategia e-greedy.

Se observa que se ganaron 182 unidades monetarias en el año.

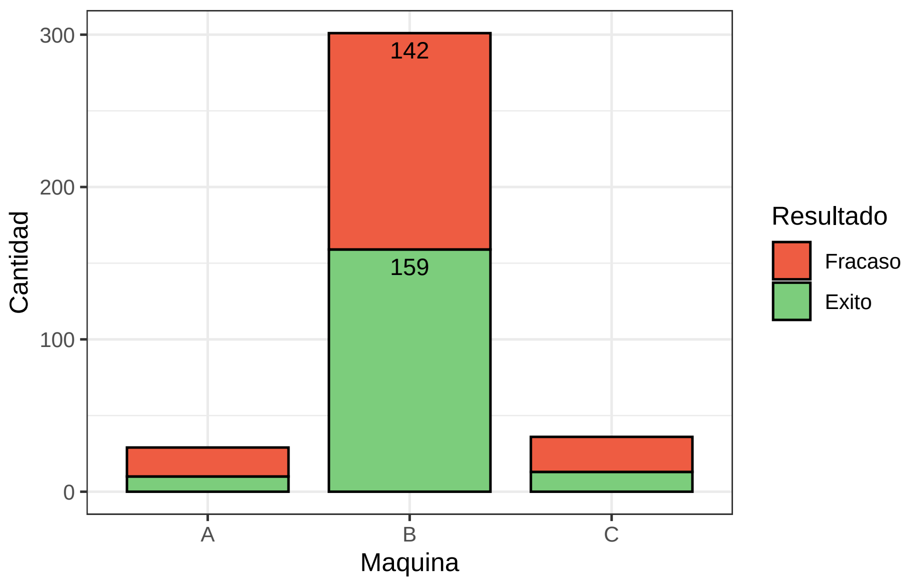

Figura 13: Frecuencia, éxito y fracaso para cada máquina durante un año con la estrategia e-greedy.

La máquina más jugada fue la B

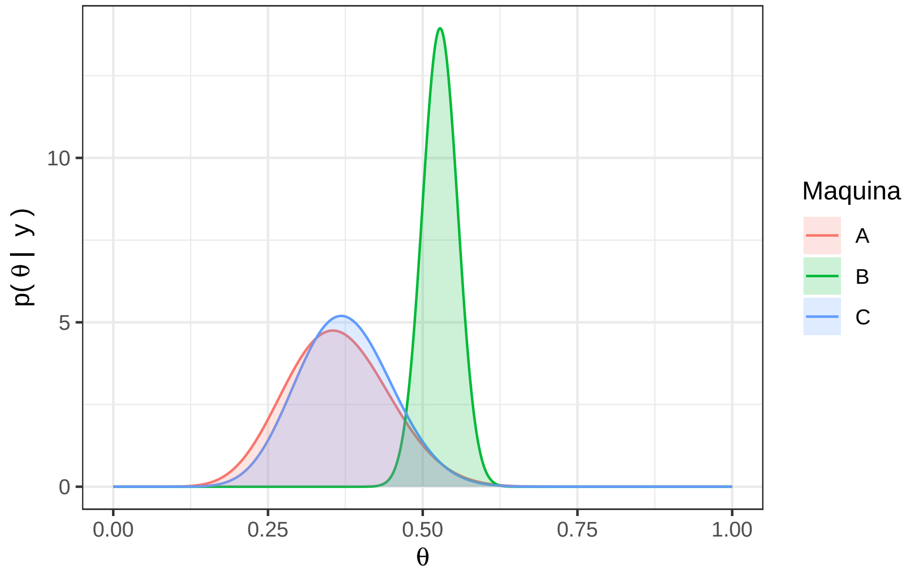

Figura 14: Probabilidades de éxito para cada máquina a posteriori luego de un año con la estrategia e-greedy.

Y así resultan las distribuciones de la probabilidad de éxito en cada máquina a posteriori.

Ventajas de la estrategia:

- La asignación del parámetro ϵ permite regular hasta que punto se prefiere explorar por sobre explotar o viceversa.
- Con buenos valores de ϵ se obtiene un buen balance entre explotación y exploración, llevando en muchos casos a buenas ganancias.
- La estrategia nunca comete el error de explotar una sola máquina, ya que siempre existe una probabilidad $\epsilon > 0$ de elegir una máquina al azar.

Desventajas de la estrategia:

- Valores demasiado altos de ϵ producen un exceso de exploración y no se aprovecha la información obtenida.
- Valores demasiado pequeños de ϵ pueden producir que se sobreexplote una mala máquina.

Softmax

Dada la tasa observada para cada máquina (π_i) y el parámetro de temperatura (τ) definido, se calcula una probabilidad de elegir cada una de estas utilizando la función softmax:

$$Pr(i) = \frac{e^{\pi_i/\tau}}{\sum_{i=1}^{3} e^{\pi_i/\tau}}$$

Donde i representa a las máquinas y τ, a diferencia del método ϵ-greedy, en el cual el parámetro determina la probabilidad de explorar o explotar. τ determina de manera inversa el peso que tendrán las tasas de éxitos observadas hasta el momento de cada máquina al calcular las probabilidades de elegir cada una de estas.

Para elegir la máquina con la que se va a jugar en el día se hizo uso de la siguiente función:

Función softmax

```
softmax <- function(maquina, ganancia, param) {

  # Las tasas son 0 en la primera iteracion
  tasa <- numeric(3)

  # Cuando una maquina no se usa ni una vez, la tasa va a seguir siendo 0
  # Cuando se juegue al menos una vez su tasa será el numero de veces que gano
  # en la maquina dividido la cantidad de veces que jugo con la maquina
  if (sum(maquina == "A") > 0) {
    tasa[1] <- sum(ganancia[maquina == "A"])/sum(maquina == "A")
  }
  if (sum(maquina == "B") > 0) {
    tasa[2] <- sum(ganancia[maquina == "B"])/sum(maquina == "B")
  }
  if (sum(maquina == "C") > 0) {
    tasa[3] <- sum(ganancia[maquina == "C"])/sum(maquina == "C")
  }

  # Se asigna el nombre de la maquina para cada tasa
  names(tasa) <- c("A","B","C")

  # Se calculan las probabilidades en base a las tasas
  probs <- exp(tasa/param)/sum(exp(tasa/param))

  return(sample(names(tasa),size = 1,prob = probs))
}
```

Sin embargo, el valor elegido para $\tau > 0$ puede influenciar la cantidad de unidades monetarias ganadas en un año, es por eso que es de importancia descubrir, a través de simulaciones, para que valor de τ se maximizan.

Búsqueda del valor óptimo de τ.

```
# Se prueban varios valores de tau
tau <- c(0.8,seq(1,10,1))
final <- NULL

# Para cada tau se hacen mil simulaciones de un año de 366 días
for (i in 1:length(tau)) {
  temp <- simulacion("softmax",n = 1000,param = tau[i])
  final <- cbind(final, apply(temp$Ganancias,2,sum))
}
colnames(final) <- tau

# Se calculan las medias y desvios
apply(final, 2, mean)
apply(final, 2, sd)

# Se prueban valores en intervalos mas chicos
tau <- c(seq(0,1,0.05))[-1]
final <- NULL

# Para cada tau se hacen mil simulaciones de un año de 366 días
for (i in 1:length(tau)) {
  temp <- simulacion("softmax",n = 1000,param = tau[i])
  final <- cbind(final, apply(temp$Ganancias,2,sum))
}
colnames(final) <- tau

# Se calculan las medias y desvios
apply(final, 2, mean)
apply(final, 2, sd)

# Se calcula la probabilidad de que las ganancias obtenidas en un año con un tau
# sean mayores que con otro y nos quedamos con aquel con el que sea más probable
# obtener mayores ganancias
for (i in 1:length(eps)) {
  for (j in 1:length(eps)) {
    print(mean(final[,i]>=final[,j]))
  }
}
```

Se puede concluir que se tiene una mayor probabildad de obtener más unidades monetarias en un año de juego con un τ de $0,1$.

Simulando los 366 días de juego:

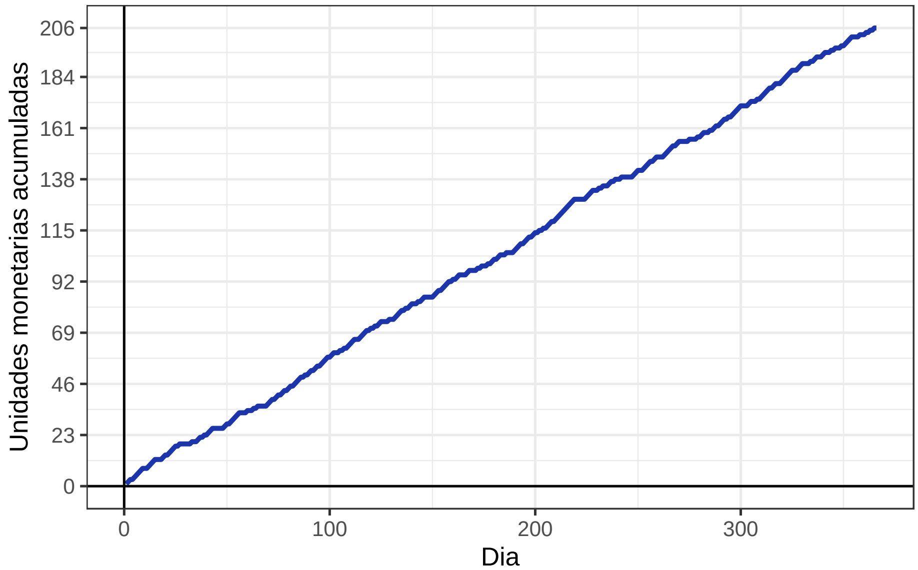

Figura 15: Unidades monetarias acumuladas durante un año con la estrategia softmax.

Se observa que se ganaron 206 unidades monetarias en el año.

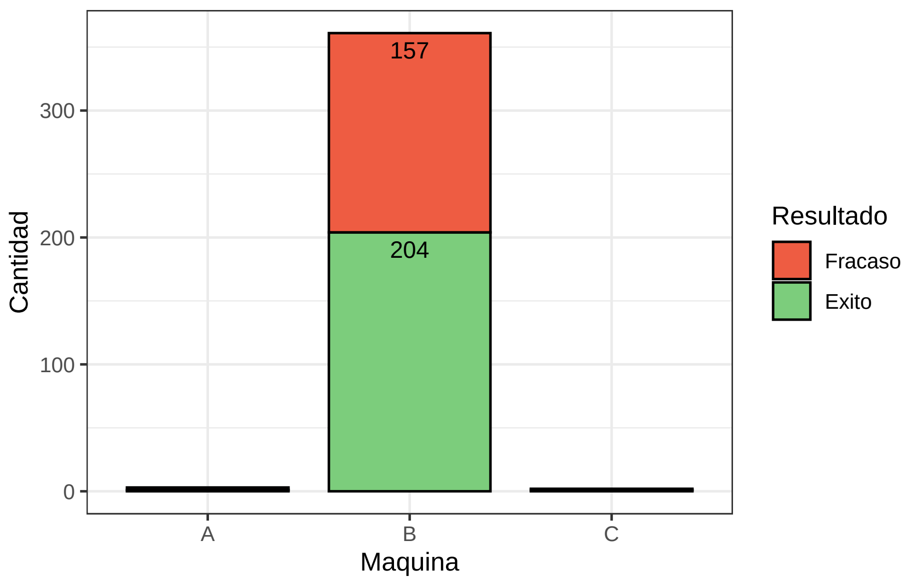

Figura 16: Frecuencia, éxito y fracaso para cada máquina durante un año con la estrategia softmax.

La máquina más jugada fue la B

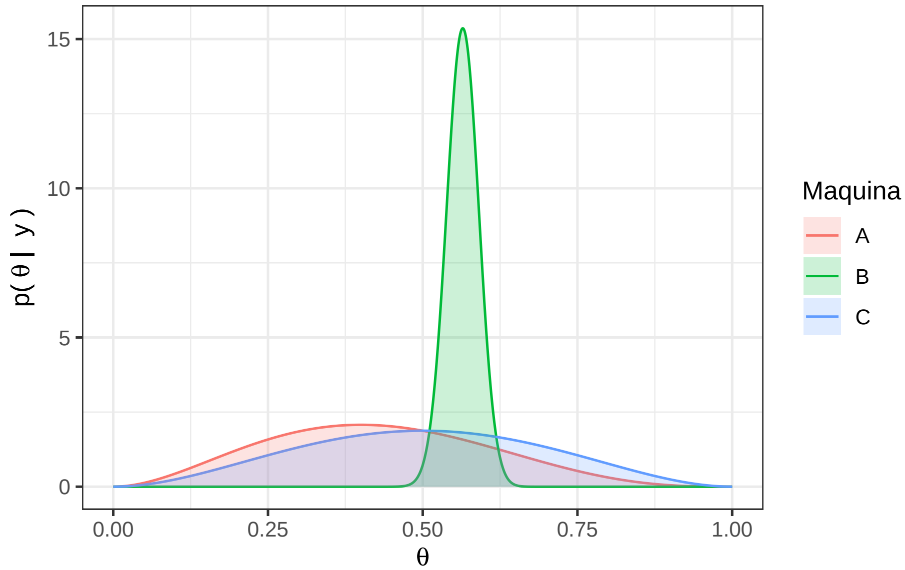

Figura 17: Probabilidades de éxito para cada máquina a posteriori luego de un año con la estrategia softmax.

Y así resultan las distribuciones de la probabilidad de éxito en cada máquina a posteriori.

Ventajas de la estrategia:

- Al igual que la estrategia "ϵ-greedy", se le puede asignar valores al parámetro τ para lograr un equilibrio entre exploración y explotación, y así lograr mayores ganancias.
- Hace uso de la información de las tiradas anteriores.

Desventajas de la estrategia:

- Para valores de τ muy cercanos a cero, se corre el riesgo de sobreexplotar una máquina
- Cuando $\tau \to \infty$, la estrategia tiende a elegir máquinas al azar.

Upper-Bound

Se selecciona la máquina que tenga el mayor límite superior, en un intervalo de credibilidad centrado (construido a partir de la distribución a posteriori de θ), del 95% de probabilidad.

Para elegir la máquina con la que se va a jugar en el día se hizo uso de la siguiente función:

Función UB

```r
UB <- function(maquina, ganancia, param) {
  # Argumentos de las distribuciones de los parámetros
  a1 <- 2; b1 <- 2; c1 <- 2; a2 <- 2; b2 <- 2; c2 <- 2

  # Si hay al menos una tirada con la máquina, modifica los parámetros de la
  # distribución a posteriori según corresponda
  if (sum(maquina == "A") > 0) {
    a1 <- 2 + sum(ganancia[maquina == "A"])
    a2 <- 2 + abs(sum(ganancia[maquina == "A"] -1))
  }
  if (sum(maquina == "B") > 0) {
    b1 <- 2 + sum(ganancia[maquina == "B"])
    b2 <- 2 + abs(sum(ganancia[maquina == "B"] -1))
  }
  if (sum(maquina == "C") > 0) {
    c1 <- 2 + sum(ganancia[maquina == "C"])
    c2 <- 2 + abs(sum(ganancia[maquina == "C"] -1))
  }

  # Limite superior de 1-param% credibilidad
  ls <- c(qbeta(1 - param/2, a1, a2), qbeta(1 - param/2, b1, b2),
          qbeta(1 - param/2, c1, c2))
  names(ls) <- c("A","B","C")

  # Se elige la maquina que tenga el mayor limite de credibilidad o una
  # al azar entre las que compartan el mayor valor
  if (sum(ls == max(ls)) == 1) {
    maq <- names(which.max(ls))

  } else if (sum(ls == max(ls)) == 2) {
    maq <- sample(names(ls)[ls == max(ls)], size = 1)

  } else {
    maq <- sample(names(ls), size = 1)
  }

  return(maq)
}
```

Simulando los 366 días de juego:

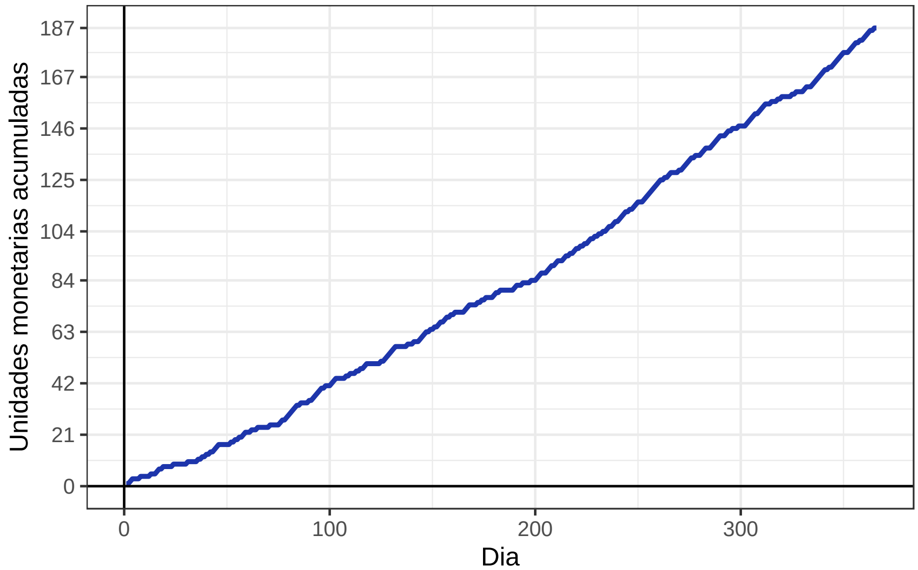

Figura 18: Unidades monetarias acumuladas durante un año con la estrategia upper-bound.

Se observa que se ganaron 187 unidades monetarias en el año.

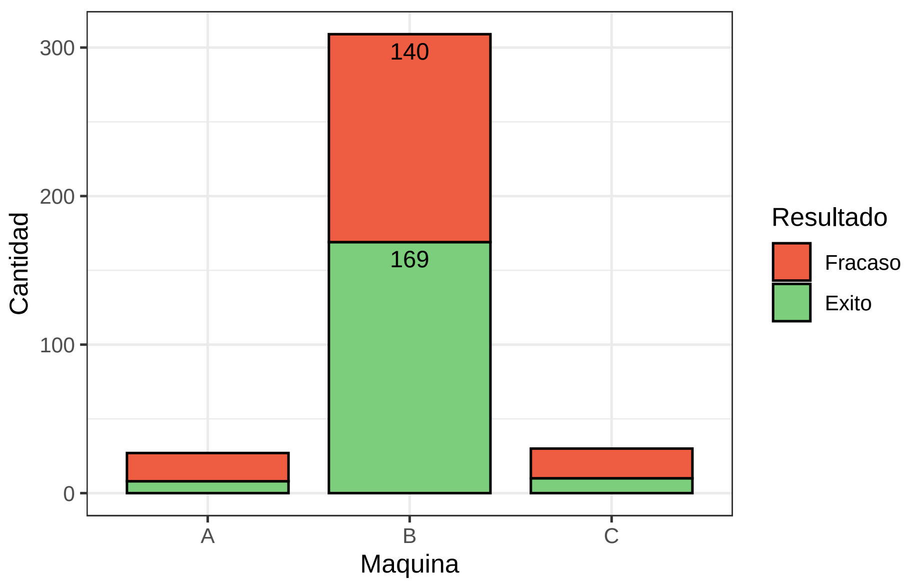

Figura 19: Frecuencia, éxito y fracaso para cada máquina durante un año con la estrategia upper-bound.

La máquina más jugada fue la B

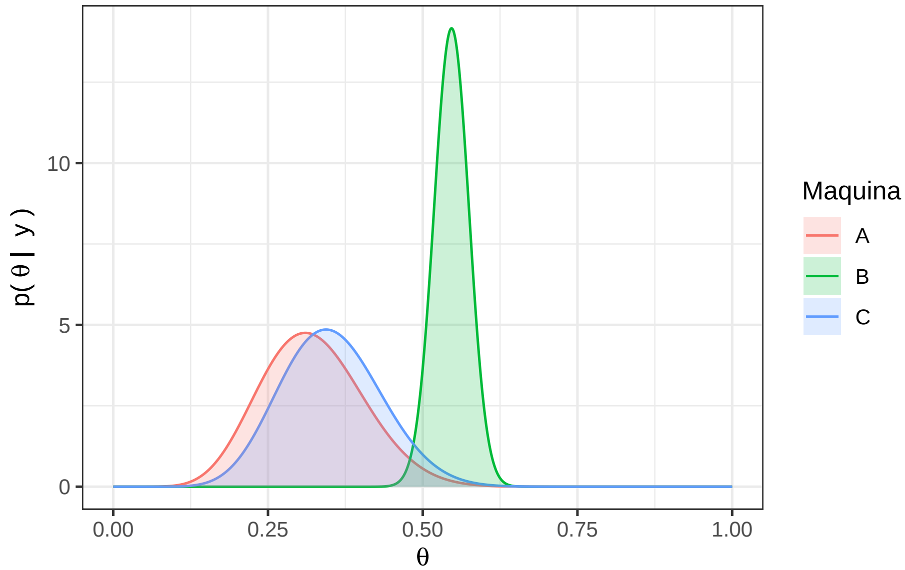

Figura 20: Probabilidades de éxito para cada máquina a posteriori luego de un año con la estrategia upper-bound.

Y así resultan las distribuciones de la probabilidad de éxito en cada máquina a posteriori.

Ventajas de la estrategia:

- Utiliza la información de las tiradas anteriores.

Thompson

Se extrae una muestra de la distribución a posteriori de las probabilidades de éxito de cada máquina y se juega con la que se obtenga la muestra más grande.

Para elegir la máquina con la que se va a jugar en el día se hizo uso de la siguiente función:

Función thompson

```
thompson <- function(maquina, ganancia) {

  # Argumentos de las distribuciones de los parámetros
  a1 <- 2; b1 <- 2; c1 <- 2; a2 <- 2; b2 <- 2; c2 <- 2

  # Si hay al menos una tirada con la máquina, modifica los parámetros de la
  # distribución a posteriori según corresponda
  if (sum(maquina == "A") > 0) {
    a1 <- 2 + sum(ganancia[maquina == "A"])
    a2 <- 2 + abs(sum(ganancia[maquina == "A"] -1))
  }
  if (sum(maquina == "B") > 0) {
    b1 <- 2 + sum(ganancia[maquina == "B"])
    b2 <- 2 + abs(sum(ganancia[maquina == "B"] -1))
  }
  if (sum(maquina == "C") > 0) {
    c1 <- 2 + sum(ganancia[maquina == "C"])
    c2 <- 2 + abs(sum(ganancia[maquina == "C"] -1))
  }


  # Se toma una muestra de la distribución a posteriori de cada máquina
  muestras <- c(rbeta(1, a1, a2), rbeta(1, b1, b2), rbeta(1, c1, c2))

  # Se nombran las valores obtenidos según de que máquina se obtuvieron
  names(muestras) <- c("A","B","C")

  # Se elige la máquina de la cual se obtuvo el mayor valor de su distribución
  # a posteriori
  return(names(muestras[which.max(muestras)]))

}
```

Simulando los 366 días de juego:

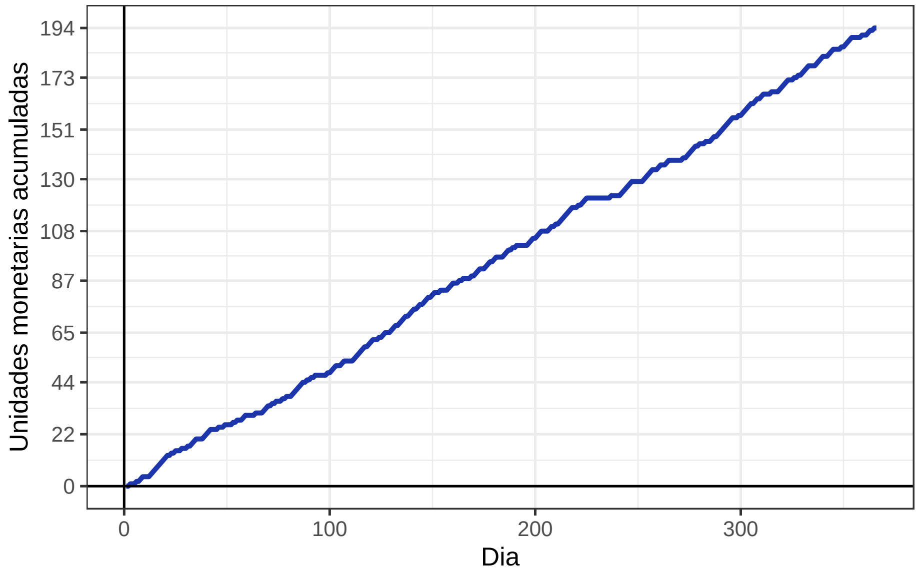

Figura 21: Unidades monetarias acumuladas durante un año con la estrategia Thompson.

Se observa que se ganaron 194 unidades monetarias en el año.

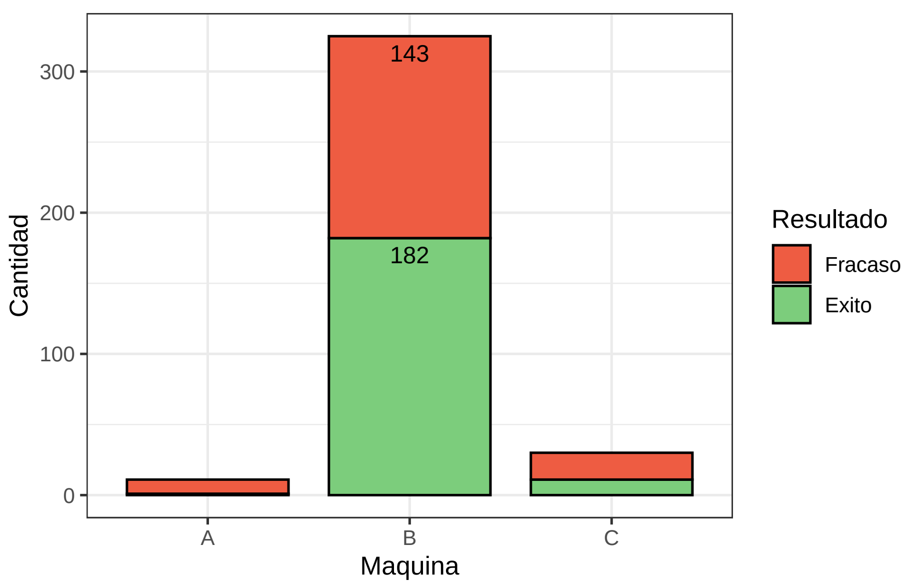

Figura 22: Frecuencia, éxito y fracaso para cada máquina durante un año con la estrategia Thompson.

La máquina más jugada fue la B

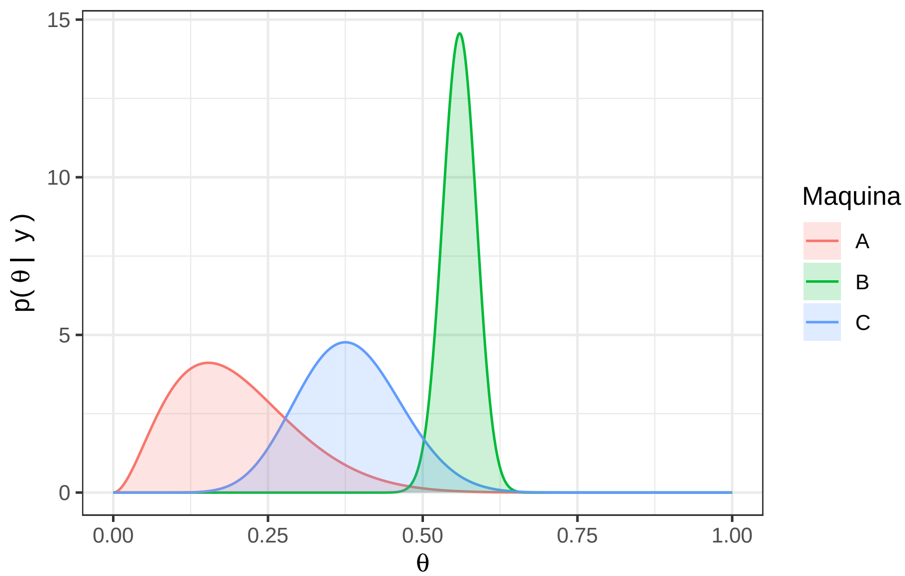

Figura 23: Probabilidades de éxito para cada máquina a posteriori luego de un año con la estrategia Thompson.

Y así resultan las distribuciones de la probabilidad de éxito en cada máquina a posteriori.

Ventajas de la estrategia:

- Utiliza la información de las tiradas anteriores.
- A pesar de que la tasa de éxito de una máquina sea mucho mayor que las demás, al estar generando valores aleatorios siempre existe la posibilidad de seguir explotando, aunque sea más probable explotar

Desventajas de la estrategia:

- Si las distribuciones a priori de las probabilidades de éxito de cada máquina no se definen adecuadamente, la estrategia puede llevar a conclusiones erróneas en períodos de tiempo que no sean suficientemente largos.

Comparaciones

Luego de presentar todas las estrategias es lógico hacerse la pregunta ¿pero cúal método es el mejor? Para responder a esto se comparan las ganancias anuales de 1000 simulaciones en un año de juego.

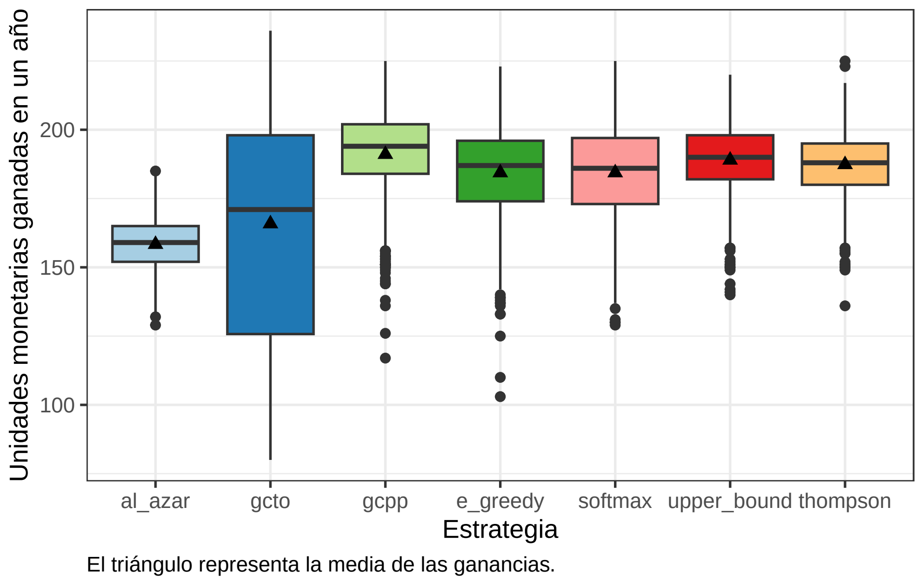

Figura 24: Boxplots comparativos de las unidades monetarias ganadas (en un año) según estrategia.

Se puede ver que las estrategias con las que se obtienen mejores resultados son **Greedy con probabilidad a posteriori** y **Upper-Bound**, siendo más probable obtener mayores ganancias con la primera. Mientras que las que peor desempeño tuvieron fueron las estrategias **al azar** y **Greedy con tasa observada**.

Se puede destacar que las estrategias que usan métodos bayesianos para la elección de máquinas son:

- Greedy con probabilidad a posteriori
- Upper-Bound
- Thompson

Porque son las que se basan en las distribuciones a posteriori para elegir la máquina a utilizar.

Conclusión

Si bien el presente estudio no busca generar la falsa ilusión de que es posible generar grandes ganancias con máquinas tragamonedas (tragaperras) es interesante desde el punto de vista de la estadistica bayesiana estudiar distintas tecnicas y ver cual es la que resuelve de mejor manera la problematica del "bandido multibrazo". Se llegó a la conclusión de que la estrategia a seguir para maximizar la probabilidad de obtener mayores ganancias, es **Greedy con probabilidad a posteriori**.

Anexo

Se pueden replicar los resultados y comprobar los códigos utilizados consultando el repositorio del trabajo.