```r
df <- data.frame(a = c(3 , 10, 9),
                 b = c(29, 36, 41))
library(ggplot2)
num_samples <- 10000000
monte_carlo_p8 <- function(alpha, beta) {
    theta_samples <- rbeta(num_samples, alpha, beta)
    W_samples <- rbinom(num_samples, 52, theta_samples)
    probability <- mean(W_samples >= 8)
    cat("Estimated probability:", probability,"\n")
    cat("Num samples:", num_samples, "\n")
}

for (i in 1:nrow(df)) {
  pair <- df[i,]
  monte_carlo_p8(pair$a, pair$b)
}

alpha <- 3
beta <- 29
theta_samples <- rbeta(num_samples, alpha, beta)
W_samples <- rbinom(num_samples, 52, theta_samples)
probability <- mean(W_samples >= 8)
cat("Estimated probability:", probability,"\n")
cat("Num samples:", num_samples, "\n")
df <- data.frame(W=W_samples)

ggplot(df, aes(x = W)) +
  geom_bar(aes(y = ..count.. / sum(..count..)),
           fill = "skyblue", color = "black") +
  labs(title = "PMF of W_samples",
       x = "Number of Wins (W)",
       y = "Probability") +
  theme_minimal()

ggsave(argv[1])
```

```r
library(gsl)
library(ggplot2)

beta_entropy <- function(params) {
  a <- params[1]
  b <- params[2]

  if (a <= 0 || b <= 0) {
    stop("Parameters ’a’ and ’b’ must be greater than zero.")
  }

  psi_sum <- psi(a) + psi(b)
  entropy <- log(beta(a, b)) - (a - 1) * psi(a) - (b - 1) * psi(b) + (a + b - 2) * psi_sum

  cat(a,b,a+b,entropy,"\n")
  return(entropy)
}

pairs <- expand.grid(a= 1:10, b=1:10)
pairs$entropy <- apply(pairs, 1, beta_entropy)

ggplot(pairs, aes(x = a + b, y = entropy)) +
  geom_point() +
  labs(x = "a + b", y = "Entropy") +
  ggtitle("Entropy of Beta Distribution vs. a + b")

ggsave(argv[1])
```

```r
library(latex2exp) # for TeX
library(ggplot2)
library(gsl) # for psi

generate_beta_density <- function(a, b) {
  p <- seq(0, 1, length.out = 1000)
  density <- dbeta(p, a, b)
  data.frame(p = p, density = density)
}

beta_entropy <- function(p) {
  a <- p[1]
  b <- p[2]
  psi_sum <- psi(a) + psi(b)
  entropy <- log(beta(a, b)) - (a - 1) * psi(a) - (b - 1) * psi(b) + (a + b - 2) * psi_sum
  return(entropy)
}

hypers <- rbind(c(1, 1), c(8, 8), c(7, 13), c(3, 29), c(10, 36), c(9, 41))
beta_entropies <- apply(hypers, 1, beta_entropy)

df_all <- NULL
for (i in 1:nrow(hypers)) {
  df <- generate_beta_density(hypers[i, 1], hypers[i, 2])
  df$group <- paste("a=", hypers[i, 1], ", b=", hypers[i, 2], "",sep="")
  df_all <- rbind(df_all, df)
}

ggplot(df_all, aes(x = p, y = density, color = group)) +
  geom_line() +
  labs(x = TeX("$\\theta$"), y = TeX("$p(\\theta; a, b)$"), color = "Parameters") +
  theme_minimal() +
  theme(legend.position = "top")

ggsave(argv[1])
```

```r
library(MASS)
library(gsl)

a <- as.integer(argv[1])
b <- as.integer(argv[2])
df <- data.frame(a = c(1, 8, 7,  3 ),
                 b = c(1, 8, 13, 29))

beta.mean <- function(a,b) {
  return(a/(a+b))
}
beta.mode <- function(a,b) {
  return((a-1)/(a+b-2))
}
beta.var  <- function(a,b) {
  return((a*b)/((a+b)*(a+b)*(a+b+1)))
}

beta.entropy <- function(a,b) {
  psi_sum <- psi(a) + psi(b)
  entropy <- log(beta(a, b)) - (a - 1) * psi(a) - (b - 1) * psi(b) + (a + b - 2) * psi_sum
  return(entropy)
}

summarise_beta <- function(params){
  a <- params[1]
  b <- params[2]
  cat("a =",a,", b = ", b,"\n")
  conf_interval <- qbeta(c(0.025, 0.975),a,b)

  cat("Mean:",     beta.mean(a,b), "\n")
  cat("Mode:",     beta.mode(a,b), "\n")
  cat("Variance:", beta.var(a,b),  "\n")
  # cat("Entropy:", beta.entropy(a,b),  "\n")
  cat("95% Confidence Interval: |", conf_interval[1], "-", conf_interval[2], "| = ", conf_interval[2] - conf_interval[1], "\n\n")
}

for (i in 1:nrow(df)) {
  pair <- df[i,]
  summarise_beta(c(pair$a, pair$b))
}
```

```r
library(gsl)
library(ggplot2)

beta.var  <- function(pair) {
  a <- pair[1]
  b <- pair[2]
  return((a*b)/((a+b)*(a+b)*(a+b+1)))
}

pairs <- expand.grid(a= 1:10, b=1:10)
pairs$entropy <- apply(pairs, 1, beta.var)

ggplot(pairs, aes(x = a + b, y = entropy)) +
  geom_point() +
  labs(x = "a + b", y = "Variance") +
  ggtitle("Variance of Beta Distribution vs. a + b")

ggsave(argv[1])
```

```r
library(stats)

# Parameters
alpha <- 3
beta <- 29

df <- data.frame(a = c(10, 9,  3 ),
                 b = c(36, 41, 29))
apply(df, 1, function(row) {
  num_samples <- 10000000
  num_games <- 52
  alpha <- row[1]
  beta <- row[2]

  # Simulate Game 1
  theta_samples <- rbeta(num_samples, alpha, beta)
  W_samples <- rbinom(num_samples, num_games, theta_samples)
  game1_return <-  (10 * W_samples) - 100

  # Simulate Game 2
  game2_return <- (10 * W_samples^2) - 1000

  # Calculate expected return for each game
  expected_return_game1 <- mean(game1_return)
  expected_return_game2 <- mean(game2_return)

  cat(paste("a =", row[1], ", b =", row[2]), "\n")
  cat(paste("Expected return for Game 1:", expected_return_game1), "\n")
  cat(paste("Expected return for Game 2:", expected_return_game2),"\n")
  cat("\n")
})
```

```r
library(stats)

# Define the functions for p(W >= 8 | theta) and p(theta)
p_W_given_theta <- function(theta) {
  1 - pbinom(7, 52, theta)
}

p_theta <- function(theta) {
  dbeta(theta, 3, 29)
}

# Compute the joint probability by integrating the product of the two functions
joint_probability <- function(theta) {
  p_W_given_theta(theta) * p_theta(theta)
}

# Integrate the joint probability function numerically
result <- integrate(joint_probability, lower = 0, upper = 1)

# The result$value contains the estimated probability
print(paste("Estimated probability:", result$value))
```