```
src/p8.r
                Mon Mar 11 12:00:32 2024
                                                 1
    1: df \leftarrow data.frame(a = c(3, 10, 9),
    2:
                         b = c(29, 36, 41)
    3: library(ggplot2)
    4:
       num samples <- 10000000
    5:
       monte_carlo_p8 <- function(alpha, beta) {</pre>
    6:
           theta samples <- rbeta(num samples, alpha, beta)
    7:
           W samples <- rbinom(num samples, 52, theta samples)
    8:
           probability <- mean(W_samples >= 8)
    9:
           cat("Estimated probability:", probability,"\n")
           cat("Num samples:", num_samples, "\n")
   10:
   11: }
   12:
   13: for (i in 1:nrow(df)) {
   14:
         pair <- df[i,]</pre>
   15:
         monte_carlo_p8(pair$a, pair$b)
   16: }
   17:
   18: alpha <- 3
   19: beta <- 29
   20: theta samples <- rbeta(num samples, alpha, beta)
   21: W_samples <- rbinom(num_samples, 52, theta_samples)
   22: probability <- mean(W samples >= 8)
   23: cat("Estimated probability:", probability,"\n")
   24: cat("Num samples:", num_samples, "\n")
   25: df <- data.frame(W=W samples)
   26:
   27: qqplot(df, aes(x = W)) +
   28:
         geom_bar(aes(y = ..count.. / sum(..count..)),
   29:
                  fill = "skyblue", color = "black") +
   30:
         labs(title = "PMF of W_samples",
   31:
              x = "Number of Wins (W)",
   32:
              v = "Probabilitv") +
   33:
         theme minimal()
   34:
   35: ggsave(argv[1])
```

```
src/beta-entropies.r
                       Tue Mar 05 10:17:45 2024
    1: library(qsl)
    2: library(ggplot2)
    3:
    4: beta_entropy <- function(params) {
    5:
        a \leftarrow params[1]
    6:
        b <- params[2]
    7:
         if (a <= 0 | b <= 0) {
    8:
    9:
           stop("Parameters 'a' and 'b' must be greater than zero.")
   10:
   11:
   12:
         psi sum <- psi(a) + psi(b)
   13:
         entropy < - \log(beta(a, b)) - (a - 1) * psi(a) - (b - 1) * psi(b) + (a + b - 2) * psi_sum
   14:
   15:
         cat (a, b, a+b, entropy, "\n")
   16:
         return (entropy)
   17: }
   18:
   19: pairs <- expand.grid(a= 1:10, b=1:10)
   20: pairs$entropy <- apply(pairs, 1, beta_entropy)
   21:
   22: qqplot(pairs, aes(x = a + b, y = entropy)) +
   23:
         geom point() +
   24:
       labs (x = "a + b", y = "Entropy") +
   25:
         gqtitle("Entropy of Beta Distribution vs. a + b")
   26:
   27: ggsave(argv[1])
```

```
src/betas.r
                  Mon Mar 11 12:20:28 2024
                                                    1
    1: library(latex2exp) # for TeX
    2: library(ggplot2)
    3: library(qsl) # for psi
    4:
    5: generate_beta_density <- function(a, b) {
        p < - seq(0, 1, length.out = 1000)
    6:
    7:
         density <- dbeta(p, a, b)
    8:
         data.frame(p = p, density = density)
    9: }
   10:
   11: beta_entropy <- function(p) {</pre>
   12:
        a < - p[1]
   13:
       b <- p[2]
   14:
       psi_sum <- psi(a) + psi(b)
   15:
       entropy \langle - \log(beta(a, b)) - (a - 1) * psi(a) - (b - 1) * psi(b) + (a + b - 2) * psi_sum
   16:
        return(entropy)
   17: }
   18:
   19: hypers \leftarrow rbind(c(1, 1), c(8, 8), c(7, 13), c(3, 29), c(10, 36), c(9, 41))
   20: beta_entropies <- apply(hypers, 1, beta_entropy)</pre>
   21:
   22: df all <- NULL
   23: for (i in 1:nrow(hypers)) {
        df <- generate_beta_density(hypers[i, 1], hypers[i, 2])</pre>
   24:
   25:
       df$group <- paste("a=", hypers[i, 1], ", b=", hypers[i, 2], "", sep="")</pre>
   26:
        df all <- rbind(df all, df)</pre>
  27: }
   28:
   29: qqplot(df_all, aes(x = p, y = density, color = qroup)) +
   30:
         geom_line() +
   31:
        labs(x = TeX("\$\theta\$"), y = TeX("\$p(\theta; a, b)\$"), color = "Parameters") +
   32:
       theme minimal() +
   33:
         theme(legend.position = "top")
   34:
   35: ggsave(argv[1])
```

```
src/beta-summary.r
                         Tue Mar 05 15:36:58 2024
                                                          1
   1: library (MASS)
   2: library(gsl)
   3:
   4: a <- as.integer(argv[1])
   5: b <- as.integer(argv[2])
   6: df \leftarrow data.frame(a = c(1, 8, 7, 3),
                        b = c(1, 8, 13, 29)
   7:
   8:
   9: beta.mean <- function(a,b) {
        return(a/(a+b))
  10:
  11: }
  12: beta.mode <- function(a,b) {
  13: return((a-1)/(a+b-2))
  14: }
  15: beta.var <- function(a,b) {
        return ((a*b) / ((a+b)*(a+b)*(a+b+1)))
  17: }
  18:
  19: beta.entropy <- function(a,b) {
      psi_sum <- psi(a) + psi(b)</pre>
  20:
        entropy < - \log(beta(a, b)) - (a - 1) * psi(a) - (b - 1) * psi(b) + (a + b - 2) * psi_sum
  21:
  22:
        return (entropy)
  23: }
  24:
  25: summarise_beta <- function(params) {
  26: a <- params[1]
  27:
        b <- params[2]
  28:
        cat ("a =",a,", b = ", b,"\n")
  29:
        conf_interval \leftarrow qbeta(c(0.025, 0.975), a, b)
  30:
  31:
        cat("Mean:",
                       beta.mean(a,b), "\n")
                        beta.mode(a,b), "\n")
  32:
        cat("Mode:",
        cat("Variance:", beta.var(a,b), "\n")
  33:
        # cat("Entropy:", beta.entropy(a,b), "\n")
  34:
        cat("95% Confidence Interval: | ", conf_interval[1], "-", conf_interval[2], " | = ", conf_interval[2] - conf_interval[1], "\n
  35:
\n")
  36: }
  37:
  38: for (i in 1:nrow(df)) {
  39:
      pair <- df[i,]
         summarise_beta(c(pair$a, pair$b))
  40:
  41: }
```

```
src/beta-variances.r
                           Tue Mar 05 12:17:49 2024
    1: library(qsl)
    2: library(ggplot2)
    3:
    4: beta.var <- function(pair) {
   5: a <- pair[1]
   6: b <- pair[2]
   7: return ((a*b) / ((a+b)*(a+b)*(a+b+1)))
   8: }
    9:
   10: pairs <- expand.grid(a= 1:10, b=1:10)
   11: pairs$entropy <- apply(pairs, 1, beta.var)
   12:
   13:
      qqplot(pairs, aes(x = a + b, y = entropy)) +
   14: geom point() +
  15: labs (x = "a + b", y = "Variance") +
   16:
       qqtitle("Variance of Beta Distribution vs. a + b")
   17:
   18: ggsave(argv[1])
```

```
src/games.r Fri Mar 08 12:28:28 2024
                                                     1
    1: library(stats)
    2:
    3: # Parameters
    4: alpha <- 3
    5: beta <- 29
    6:
    7: df < -data.frame(a = c(10, 9, 3),
    8:
                         b = c(36, 41, 29)
    9: apply(df, 1, function(row) {
   10:
         num_samples <- 10000000</pre>
   11: num_games <- 52
   12: alpha <- row[1]
   13:
        beta <- row[2]</pre>
   14:
   15:
         # Simulate Game 1
   16:
         theta_samples <- rbeta(num_samples, alpha, beta)</pre>
   17:
         W_samples <- rbinom(num_samples, num_games, theta_samples)</pre>
   18:
         game1 return <- (10 * W samples) - 100</pre>
   19:
   20:
         # Simulate Game 2
   21:
         game2 return \leftarrow (10 * W samples^2) - 1000
   22:
   23:
         # Calculate expected return for each game
   24:
         expected return game1 <- mean(game1 return)</pre>
   25:
         expected return game2 <- mean(game2 return)</pre>
   26:
   27:
         cat (paste ("a =", row[1], ", b =", row[2]), "\n")
   28:
         cat(paste("Expected return for Game 1:", expected_return_game1), "\n")
   29:
         cat(paste("Expected return for Game 2:", expected_return_game2),"\n")
   30:
         cat("\n")
   31: })
```

```
src/integrate-p8.r
                         Tue Mar 05 16:02:58 2024
   1: library(stats)
    2:
   3: # Define the functions for p(W \ge 8 \mid theta) and p(theta)
    4: p_W_given_theta <- function(theta) {
    5: 1 - pbinom(7, 52, theta)
    6: }
   7:
   8: p theta <- function(theta) {
    9:
         dbeta(theta, 3, 29)
   10: }
   11:
   12: # Compute the joint probability by integrating the product of the two functions
   13: joint_probability <- function(theta) {</pre>
   14: p_W_given_theta(theta) * p_theta(theta)
   15: }
   16:
   17: # Integrate the joint probability function numerically
   18: result <- integrate (joint probability, lower = 0, upper = 1)
   19:
   20: # The result$value contains the estimated probability
   21: print(paste("Estimated probability:", result$value))
```