Statystyka Matematyczna - Sprawozdanie 3

Bartosz Michalak

26 stycznia 2026

Wstęp

Poniższe sprawozdanie zawiera rozwiązania zadań dotyczących czynnika Bayesa, algorytmu Metropolisa-Hastingsa oraz estymacji przedziałów wiarogodnych najwyższej gęstości (HPD).

1 Czynnik Bayesa w modelu dwumianowym

Rozważamy zmienną losową $X \sim \mathcal{B}(n, \vartheta)$ oraz problem testowania hipotez:

$$H_0 : \vartheta \leq \vartheta_0 \quad \text{vs} \quad H_1 : \vartheta > \vartheta_0$$

z rozkładem a priori $\vartheta \sim \mathcal{B}e(\alpha, \beta)$.

```
# Funkcja obliczająca B10 i interpretację
analyze_bayes_factor <- function(x, n, theta0, alpha, beta) {
  p_h0_prior <- pbeta(theta0, shape1 = alpha, shape2 = beta)
  p_h1_prior <- 1 - p_h0_prior

  alpha_post <- alpha + x
  beta_post <- beta + n - x
  p_h0_post <- pbeta(theta0, shape1 = alpha_post, shape2 = beta_post)
  p_h1_post <- 1 - p_h0_post

  prior_odds <- p_h1_prior / p_h0_prior
  post_odds <- p_h1_post / p_h0_post

  # Zabezpieczenie przed dzieleniem przez zero
  if(prior_odds == 0) return(list(B10 = 0, Evidence = "Not worth mention"))

  B10 <- post_odds / prior_odds

  evidence <- ifelse(B10 > 150, "Very strong",
              ifelse(B10 > 20, "Strong",
              ifelse(B10 > 3, "Positive",
              "Not worth mention")))

  return(list(B10 = B10, Evidence = evidence))
}
```

Poniżej przedstawiono wyniki symulacji dla $n = 100$, $\vartheta_0 = 1/2$ i $M = 1000$ powtórzeń.

```r
set.seed(123)
n <- 100
theta0 <- 0.5
M <- 1000
thetas <- c(0.25, 0.50, 0.75)

# Lista parametrów a priori (Alpha, Beta)
priors <- list(
  c(1,1),
  c(0.5, 0.5),
  c(2, 5),
  c(5, 2)
)

prior_names <- c("Uniform (1, 1)", "Jeffreys", "Beta(2, 5)", "Beta(5, 2)")

# Pętla po rozkładach a priori
for (i in 1:length(priors)) {
  a <- priors[[i]][1]
  b <- priors[[i]][2]
  p_name <- prior_names[i]

  results_matrix <- matrix(0, nrow=3, ncol=3)
  colnames(results_matrix) <- c("Theta = 1/4", "Theta = 1/2", "Theta = 3/4")
  rownames(results_matrix) <- c("Positive", "Strong", "Very Strong")

  # Pętla po prawdziwych wartościach theta
  for (j in 1:length(thetas)) {
    th <- thetas[j]
    X_sim <- rbinom(M, size = n, prob = th)

    # Obliczamy B10 dla całej próby
    res <- sapply(
      X_sim,
      function(x) analyze_bayes_factor(x, n, theta0, a, b)$B10
    )

    results_matrix[1, j] <- mean(res > 3 & res <= 20)   # Positive
    results_matrix[2, j] <- mean(res > 20 & res <= 150) # Strong
    results_matrix[3, j] <- mean(res > 150)             # Very Strong
  }

  cat("\\begin{table}[H]\n")
  cat("\\centering\n")

  cat(paste0(
    "\\caption{Oszacowane prawdopodobieństwa dla rozkładu a priori: ",
    p_name,
    "}\n")
  )
```

```r
  print(knitr::kable(
    results_matrix,
    digits = 3,
    format = "latex",
    booktabs = TRUE)
  )

  cat("\\end{table}\n")
  cat("\n")
}
```

Tabela 1: Oszacowane prawdopodobieństwa dla rozkładu a priori: Uniform (1, 1)

	Theta = 1/4	Theta = 1/2	Theta = 3/4
Positive	0	0.209	0.001
Strong	0	0.037	0.002
Very Strong	0	0.003	0.997

Tabela 2: Oszacowane prawdopodobieństwa dla rozkładu a priori: Jeffreys

	Theta = 1/4	Theta = 1/2	Theta = 3/4
Positive	0	0.191	0.000
Strong	0	0.033	0.005
Very Strong	0	0.004	0.995

Tabela 3: Oszacowane prawdopodobieństwa dla rozkładu a priori: Beta(2, 5)

	Theta = 1/4	Theta = 1/2	Theta = 3/4
Positive	0	0.434	0
Strong	0	0.161	0
Very Strong	0	0.022	1

Tabela 4: Oszacowane prawdopodobieństwa dla rozkładu a priori: Beta(5, 2)

	Theta = 1/4	Theta = 1/2	Theta = 3/4
Positive	0	0.049	0.001
Strong	0	0.008	0.011
Very Strong	0	0.000	0.988

Wnioski: Tabele przedstawiają prawdopodobieństwo uzyskania określonej siły dowodu przeciwko H_0. Dla $\vartheta = 1/4$ (gdy H_0 jest prawdziwa) wiersze "Strong" i "Very Strong" zawierają wartości bliskie 0, co jest pożądane (brak błędu I rodzaju). Dla $\vartheta = 3/4$ (gdy H_1 jest prawdziwa) dominuje kategoria "Very Strong", co świadczy o wysokiej mocy testu.

2 Algorytm Metropolisa-Hastingsa dla rozkładu Weibulla

W tym zadaniu generujemy próbę $X_1, \ldots, X_n$ z rozkładu Weibulla $\mathcal{W}(\vartheta, 1)$ z gęstością:

$$p_\vartheta(x) = \vartheta x^{\vartheta-1} \exp(-x^\vartheta) \mathbf{1}_{(0,\infty)}(x)$$

Przyjmujemy rozkład a priori $\vartheta \sim \mathcal{U}(0, 1)$.

Algorytm Metropolisa-Hastingsa (Random Walk) posłuży do estymacji $E[\vartheta|x]$ oraz $Median(\vartheta|x)$.

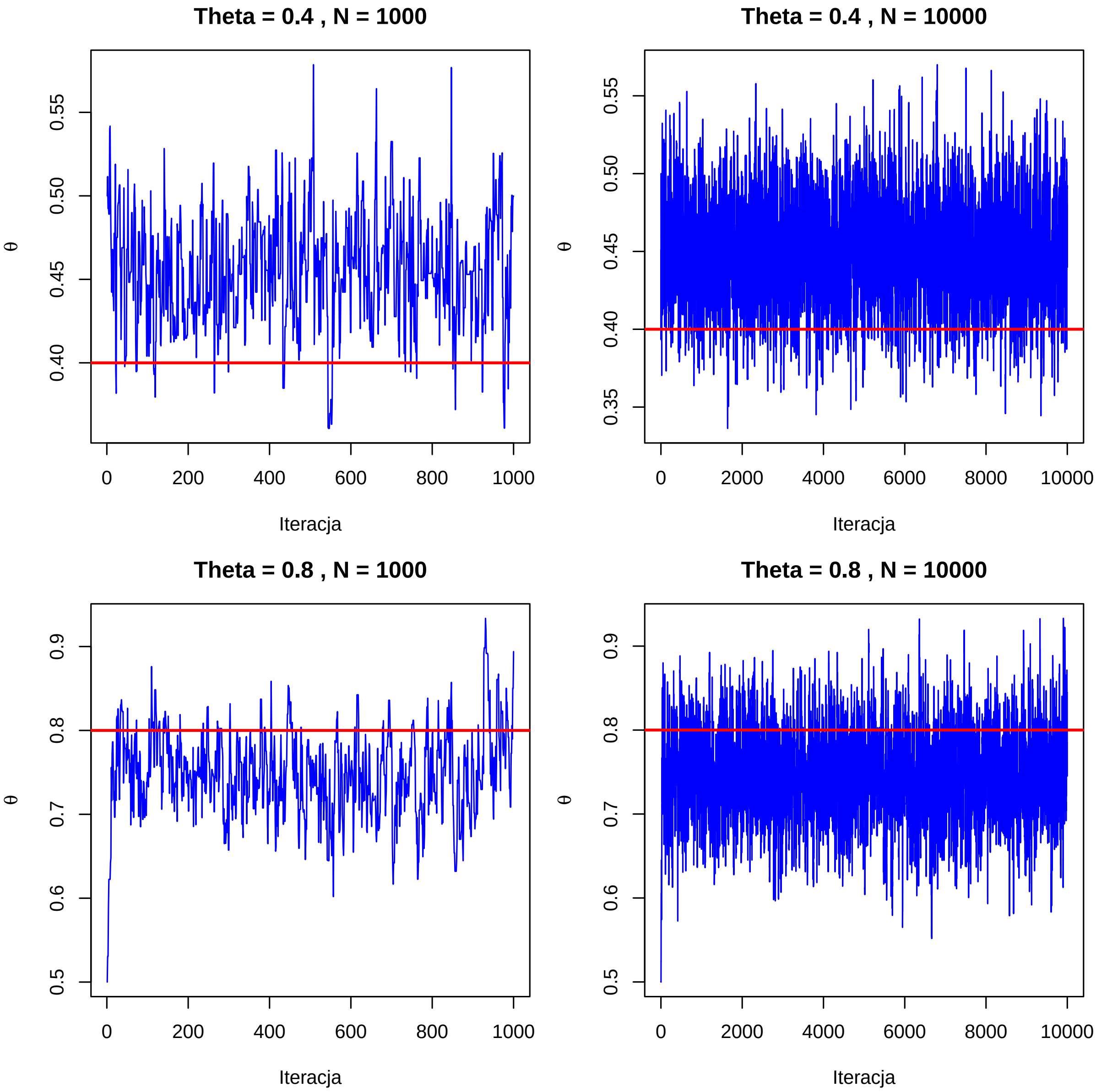
```
set.seed(4321)

# Dwie wartości prawdziwego parametru theta
thetas_to_test <- c(0.4, 0.8)
n <- 100

log_posteriori_weibull <- function(theta, x) {
  if (theta <= 0 || theta >= 1) return(-Inf)
  n <- length(x)
  val <- n * log(theta) + (theta - 1) * sum(log(x)) - sum(x^theta)
  return(val)
}

run_mh <- function(iter, x_data, start_val=0.5, sd_prop=0.05) {
  chain <- numeric(iter)
  chain[1] <- start_val
  for(i in 2:iter) {
    curr <- chain[i-1]
    prop <- rnorm(1, curr, sd_prop)
    log_alpha <- log_posteriori_weibull(prop, x_data) -
                 log_posteriori_weibull(curr, x_data)
    if (log(runif(1)) < log_alpha) chain[i] <- prop else chain[i] <- curr
  }
  return(chain)
}

# Miejsce na wyniki
estimates <- data.frame()
chains_for_plots <- list()
longest_chain_task3 <- NULL # Zmienna dla Zadania 3

# Pętla po dwóch wartościach Theta (0.4 i 0.8)
for (th_true in thetas_to_test) {

  # Generujemy dane dla aktualnego theta
  X_weibull <- rweibull(n, shape = th_true, scale = 1)

  lengths <- c(1000, 10000, 100000)

  for (L in lengths) {
    # Uruchomienie algorytmu
    ch <- run_mh(L, X_weibull)
```

```
    # Zapisujemy łańcuchy do wykresów (tylko 1k i 10k)
    if (L %in% c(1000, 10000)) {
      chains_for_plots[[paste(th_true, L, sep="_")]] <- ch
    }

    # Zapamiętujemy najdłuższy łańcuch dla pierwszego przypadku (theta=0.4)
    # do Zadania 3
    if (L == 100000 && th_true == thetas_to_test[1]) {
      longest_chain_task3 <- ch
    }

    # Statystyki do tabeli
    burn_in <- floor(L * 0.1)
    ch_clean <- ch[-(1:burn_in)]

    estimates <- rbind(estimates, data.frame(
      Prawdziwa_Theta = th_true,
      Dlugosc_Lancucha = L,
      Est_Srednia = mean(ch_clean),
      Est_Mediana = median(ch_clean)
    ))
  }
}
```

Tabela 5: Estymacja parametru ϑ metodą MCMC dla dwóch przypadków

Prawdziwa_Theta	Dlugosc_Lancucha	Est_Srednia	Est_Mediana
0.4	1e+03	0.4546	0.4535
0.4	1e+04	0.4491	0.4487
0.4	1e+05	0.4490	0.4486
0.8	1e+03	0.7489	0.7473
0.8	1e+04	0.7451	0.7439
0.8	1e+05	0.7437	0.7429

Poniżej przedstawiono przebiegi łańcuchów (Trace plots) dla obu badanych parametrów ϑ oraz długości $N = 1000$ i $N = 10000$.



Rysunek 1: Przebieg łańcuchów Markowa (Trace plots) dla $\vartheta = 0.4$ (góra) i $\vartheta = 0.8$ (dół).

Wnioski: Analiza wykresów śladowych (trace plots) wskazuje na poprawną zbieżność algorytmu Metropolisa-Hastingsa. Łańcuchy dla obu wartości (0.4 oraz 0.8) szybko osiągają stacjonarność. Obserwowane przesunięcie średniej wartości łańcucha (niebieski wykres) względem prawdziwej wartości parametru (czerwona linia) jest zjawiskiem naturalnym i wynika z losowości generowania próby danych ($n = 100$).

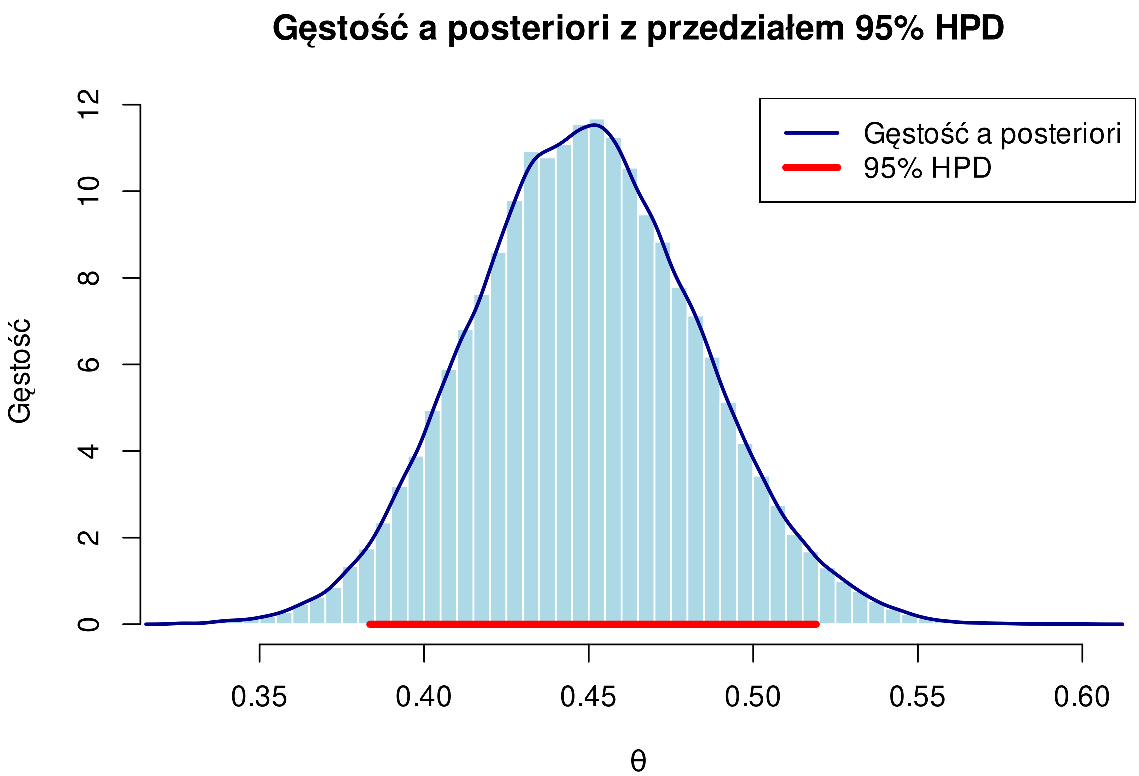
3 Przedział wiarogodny najwyższej gęstości (HPD)

Wyznaczamy 95% przedział HPD dla parametru ϑ z Zadania 2. Do analizy wybrano przypadek, gdzie prawdziwe $\vartheta = 0.4$ (najdłuższy wygenerowany łańcuch).

```
calc_HPD <- function(chain, alpha=0.05) {
  M <- length(chain)
  burn_in <- floor(M/10)
```

```
  chain_clean <- sort(chain[-(1:burn_in)])
  Mu <- length(chain_clean)

  k <- floor((1 - alpha) * Mu)
  idx_start <- 1:(Mu - k)
  idx_end <- idx_start + k

  diffs <- chain_clean[idx_end] - chain_clean[idx_start]
  best_idx <- which.min(diffs)

  lower <- chain_clean[best_idx]
  upper <- chain_clean[best_idx + k]

  return(c(Lower=lower, Upper=upper, Length=diffs[best_idx]))
}

# Używamy zapamiętanego łańcucha z Zadania 2 (dla theta = 0.4)
hpd_res <- calc_HPD(longest_chain_task3, alpha=0.05)

# Dane do wykresu
burn_in_plot <- floor(length(longest_chain_task3) * 0.1)
chain_for_hist <- longest_chain_task3[-(1:burn_in_plot)]
```

Otrzymany 95% przedział HPD: $[0.3835, 0.5191]$.



Rysunek 2: Rozkład a posteriori dla $\vartheta = 0.4$ z zaznaczonym przedziałem HPD.

Wnioski: Można zauważyć, że maksimum funkcji gęstości a posteriori (moda) jest nieco przesunięte względem prawdziwej wartości parametru (0.4). Jest to bezpośrednia konsekwencja zjawiska opisanego w zadaniu drugim, algorytm wyznaczył rozkład a posteriori na podstawie konkretnej, wylosowanej 100-elementowej próby, która mogła losowo faworyzować nieco wyższe wartości parametru. Mimo to, przedział HPD zawiera prawdziwą wartość.