Laboratorium 2

Wstęp do Analizy Danych | Politechnika Krakowska

Jakub Kapała

Numer albumu: 151885

Data: 05.04.2025

Zadanie 1 - rozkład wykładniczy

Treść Odległości między kolejnymi zdarzeniami w procesach losowych o zdarzeniach niezależnych i występujących jednorodnie są opisywane rozkładem wykładniczym. Dobrym przykładem są odstępy czasu pomiędzy kolejnymi uderzeniami kropel deszczu o szybę (przy założeniu, że intensywność deszczu nie zmienia się w czasie). Zasymulować opisane zjawisko korzystając z funkcji `runif()` i porównać wynik ze znanym rozkładem prawdopodobieństwa:

$$f(x) = \lambda e^{-\lambda x},$$

gdzie λ oznacza tempo procesu.

- Rozważyć czasy od $t = 0$ do $t = 10\,000\,\mathrm{s}$. Przyjąć, że w ciągu sekundy o szybę uderza średnio $\lambda = 5$ kropel. Co $\Delta t = 0.01\,\mathrm{s}$ sprawdzić, czy w minionym odstępie czasu Δt o szybę uderzyła kropla, tzn. wylosować zmienną losową z prawdopodobieństwem $\lambda \cdot 0.01 = 0.05$.
- Przekształcić powyższe dane do postaci czasów t, kiedy krople padały na szybę i ostatecznie do postaci odstępów między tymi zdarzeniami. Znaleźć średnią i odchylenie standardowe.
- Przedstawić dane na histogramie i porównać graficznie z wykresem powyższego wzoru.

Rozwiązanie Parametry symulacji:

```
t1 <- 10000 # czas symulacji
lambda <- 5 # srednia ilosc kropel na sekunde
delta_t <- 0.01 # odstep czasowy
probability <- lambda * delta_t # prawdopodobienstwo zmiennej losowej
set.seed(151885) # ustawienie ziarna dla powtarzalności wyników
```

Symulacja uderzeń kropli:

```
n <- t1 / delta_t # liczba kroków czasowych
hits <- runif(n) < probability # losowanie zmiennej losowej
hits[1:10]
```

```
##  [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Wartości `TRUE` oznaczają, ze kropla spadła w danym kroku, a `FALSE` ze nie spadła. Jak widać kropla najpierw uderza w 1 kroku czasowym, a potem przez 9 kroków nie pada.

Przekształcenie do czasów t, kiedy padały krople na szybę

```
times <- (1:n)[hits] * delta_t  # przekształcenie do czasow t
times[1:10]
```

```
##  [1] 0.01 0.27 0.49 0.64 0.78 1.09 1.12 1.49 1.59 1.61
```

Odstępy między zdarzeniami:

```
intervals <- diff(times) # obliczenie odstepow
intervals[1:10]
```

```
##  [1] 0.26 0.22 0.15 0.14 0.31 0.03 0.37 0.10 0.02 0.20
```

Obliczenie średniej i odchylenia standardowego:

```
mean_intervals <- mean(intervals) # srednia
sd_intervals <- sd(intervals) # odchylenie standardowe
mean_intervals
```

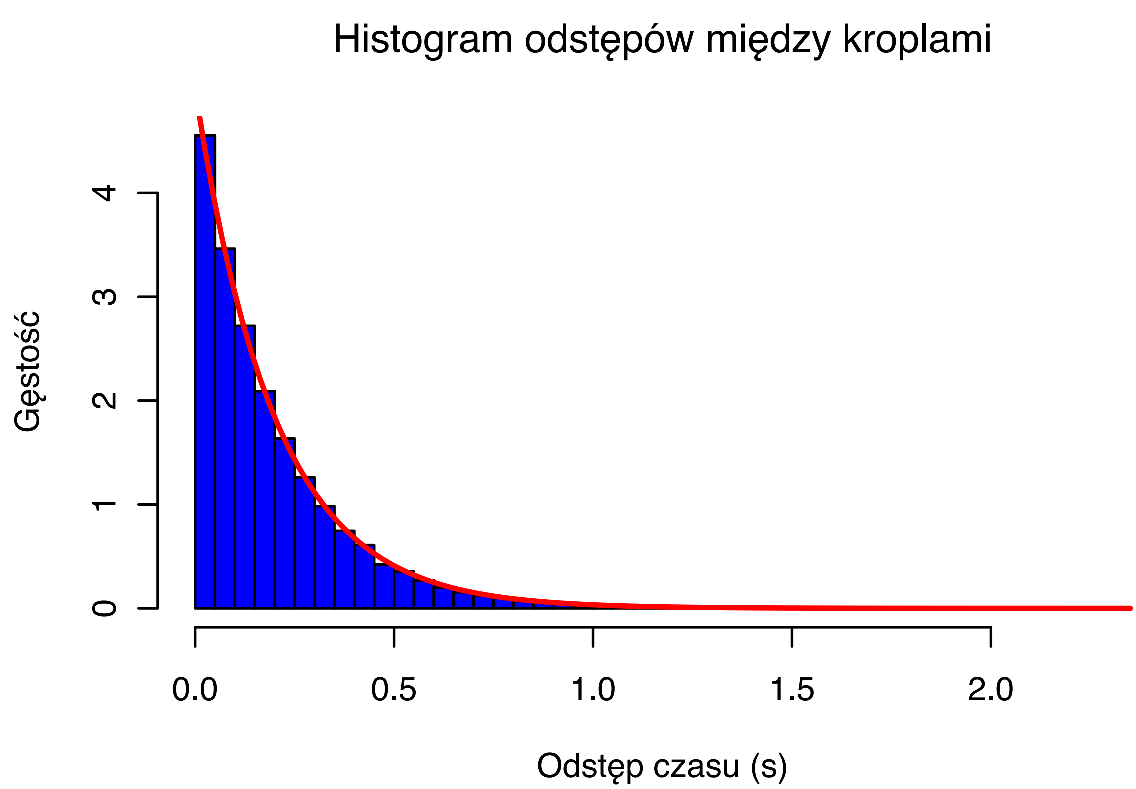
```
## [1] 0.1989388
```

```
sd_intervals
```

```
## [1] 0.1921916
```

Przedstawienie danych do histogramie i porównanie z wykresem $f(x) = \lambda e^{-\lambda x}$:

```
hist(
  intervals,
  main = "Histogram odstępów między kroplami",
  xlab = "Odstęp czasu (s)", col = "blue",
  ylab = "Gęstość",
  breaks = 40,
  probability = TRUE
)
curve(lambda * exp(-lambda * x), col = "red", lwd = 2, add = TRUE)
```



Zadanie 2

Treść Przedstaw na wykresach następujące rozkłady:

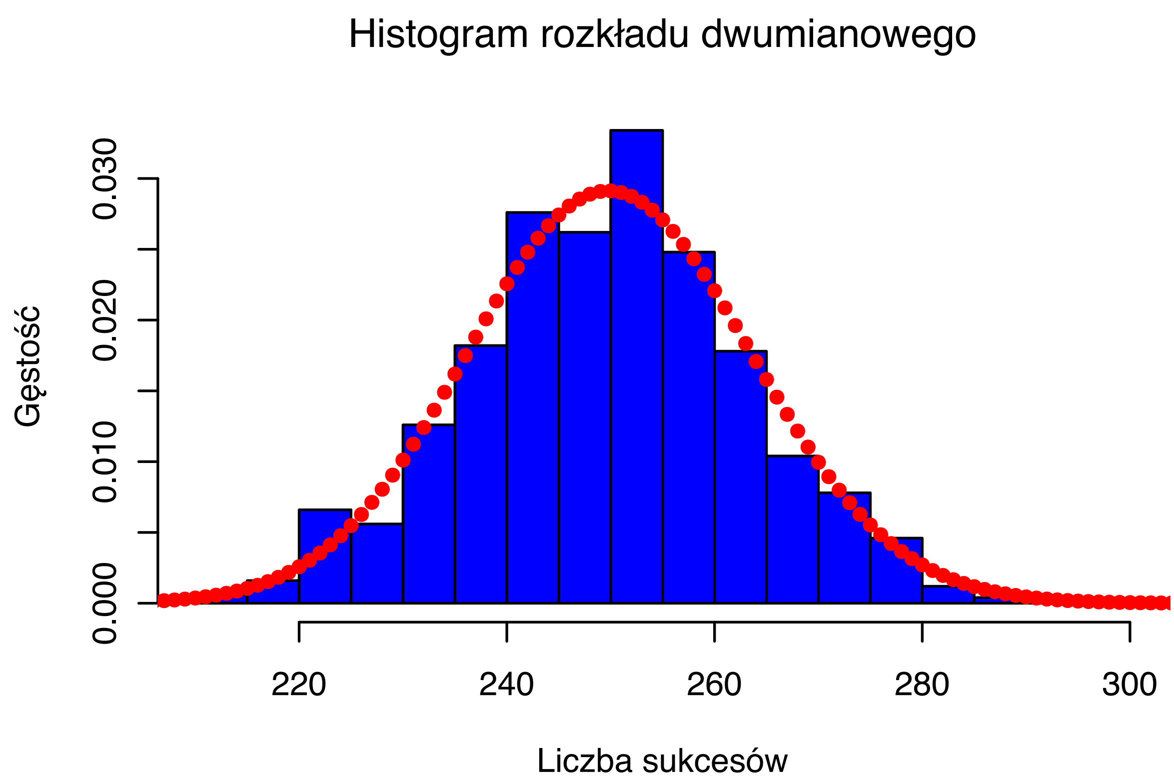
1. Dwumianowy
2. Hipergeometryczny
3. Chi-kwadrat
4. Wykładniczy
5. Weibull

W zadaniach 2.1 i 2.2 ponadto porównaj histogram dla losowej próbki danych o danym rozkładzie prawdopodobieństwa (`rbinom`, `rhyper`) z wykresami rozkładów (`dbinom`, `dhyper`).

Rozwiązanie

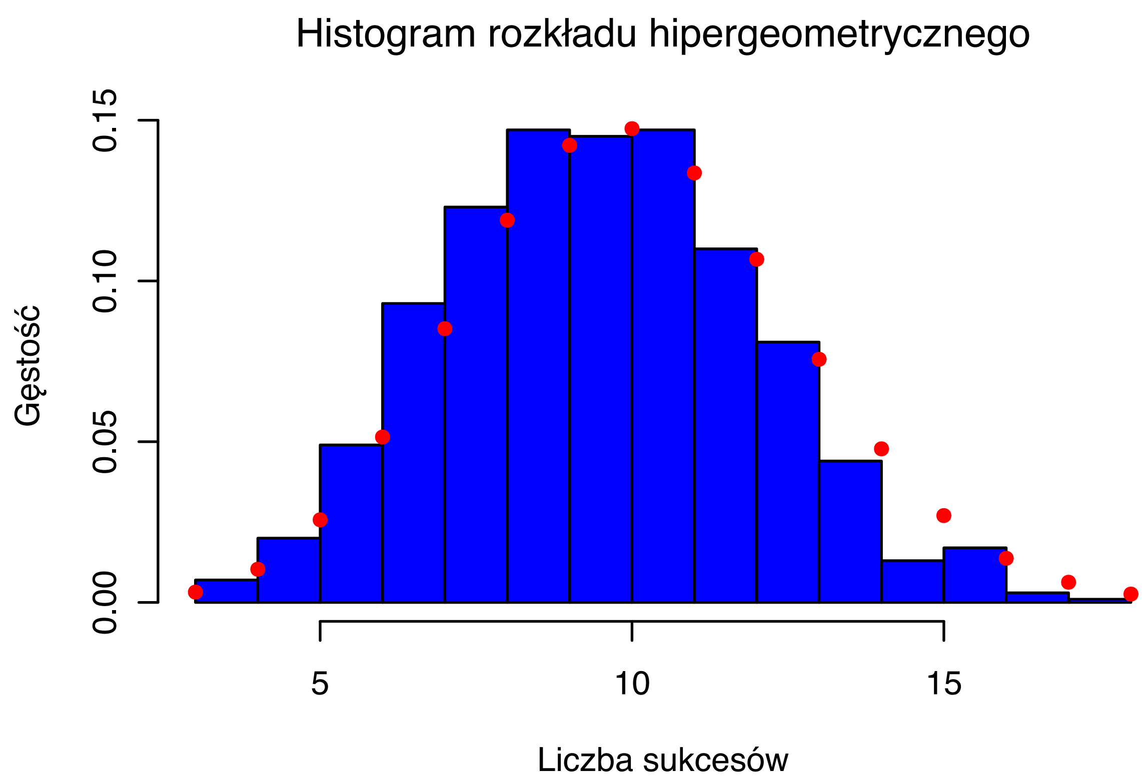
1. Rozkład dwumianowy:

```
n <- 1000 # liczba prob
p <- 0.25 # prawdopodobienstwo sukcesu
hist(rbinom(n, n, p), breaks = 20, probability = TRUE,
     main = "Histogram rozkładu dwumianowego",
     xlab = "Liczba sukcesów",
     ylab = "Gęstość",
     col = "blue")
x <- 0:n
points(x, dbinom(x, n, p), col = "red", pch = 20)
```



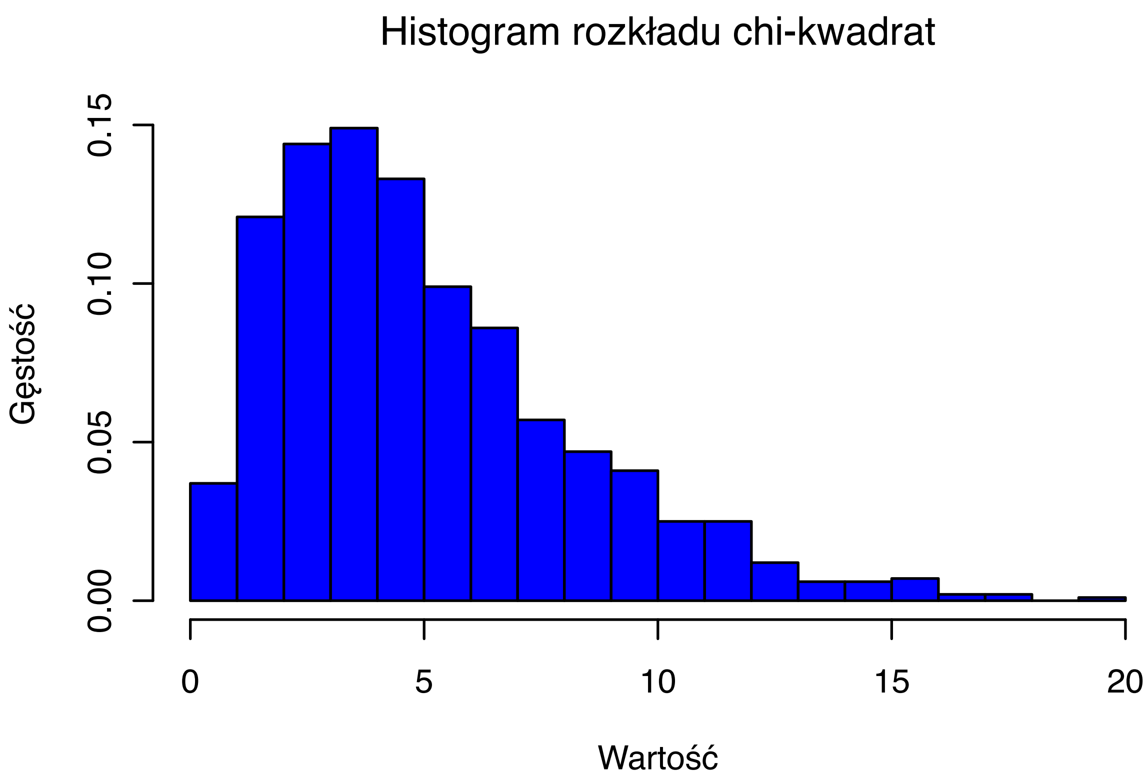
2. Rozkład hipergeometryczny:

```
k <- 50 # liczba sukcesow w populacji
N <- 500 # liczba elementow w populacji
n <- 100 # liczba losowanych elementow
n_samples <- 1000 # liczba probek do wygenerowania
hist(rhyper(n_samples, k, N - k, n), breaks = 20, probability = TRUE,
     main = "Histogram rozkładu hipergeometrycznego",
     xlab = "Liczba sukcesów",
     ylab = "Gęstość",
     col = "blue")
x <- 0:min(k, n)
points(x, dhyper(x, k, N - k, n), col = "red", pch = 20)
```



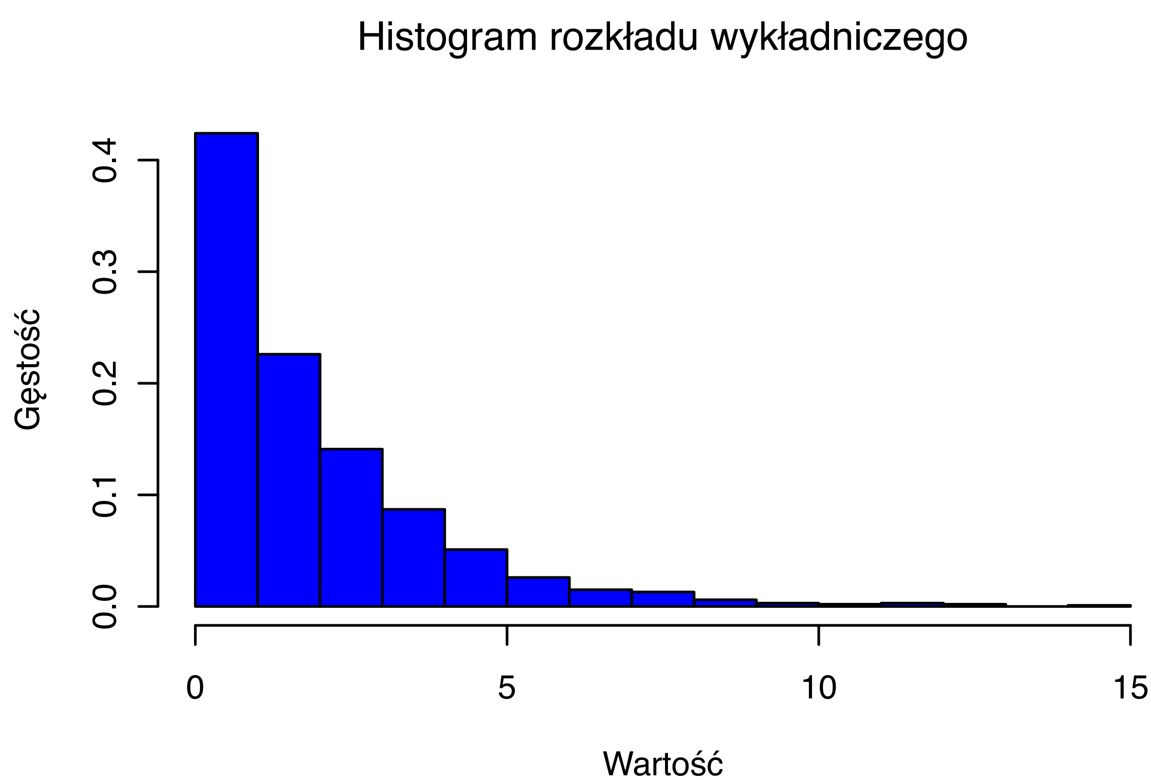
3. Rozkład chi-kwadrat:

```
n <- 1000 # liczba probek
df <- 5 # liczba stopni swobody
hist(rchisq(n, df), breaks = 20, probability = TRUE,
     main = "Histogram rozkładu chi-kwadrat",
     xlab = "Wartość",
     ylab = "Gęstość",
     col = "blue")
```



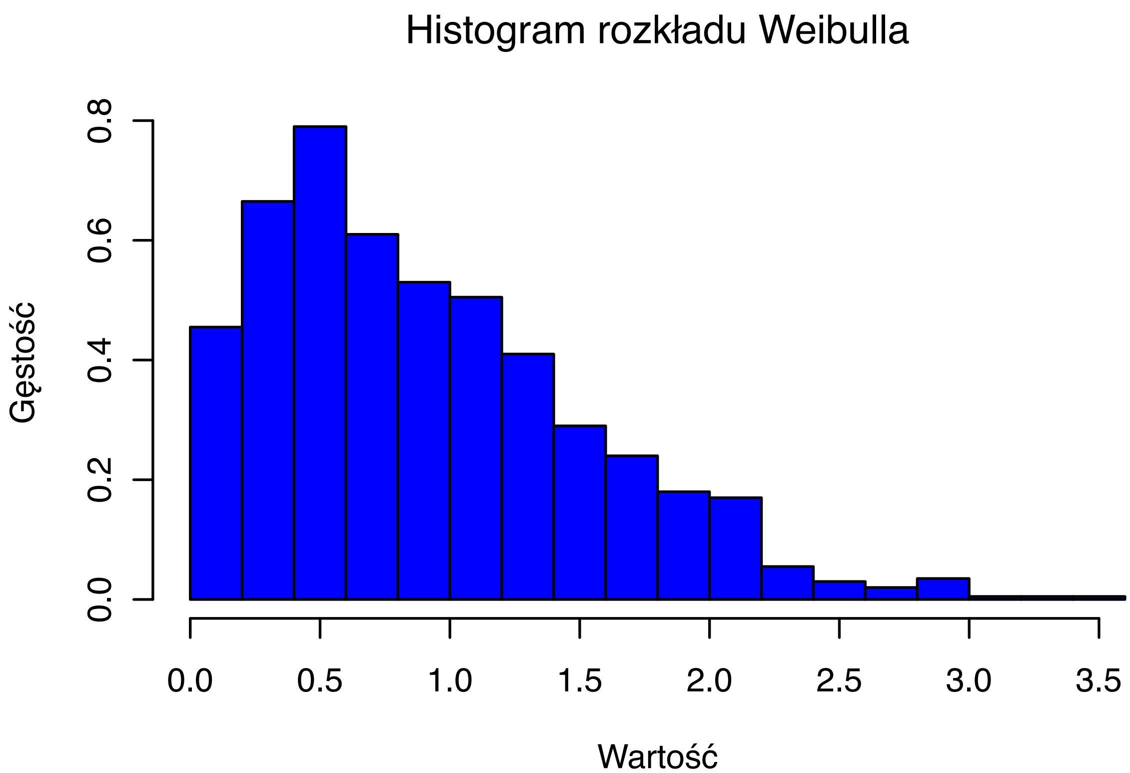
4. Rozkład wykładniczy:

```
n <- 1000 # liczba probek
lambda <- 0.5 # odwrotnosc parametru skali rozkładu
hist(rexp(n, lambda), breaks = 20, probability = TRUE,
     main = "Histogram rozkładu wykładniczego",
     xlab = "Wartość",
     ylab = "Gęstość",
     col = "blue")
```



5. Rozkład Weibulla:

```
n <- 1000 # liczba probek
k <- 1.5 # parametr ksztaltu
lambda <- 1 # parametr skali
hist(rweibull(n, k, lambda), breaks = 20, probability = TRUE,
     main = "Histogram rozkładu Weibulla",
     xlab = "Wartość",
     ylab = "Gęstość",
     col = "blue")
```



Zadanie 3 – matrix(), for(), apply()

Treść Za pomocą polecenia `matrix` utwórz macierz 4×5. Następnie w pętli `for` wypełnij ją liczbami tak, aby stanowiła tabliczkę mnożenia, tzn. aby w i-tym wierszu, w j-tej kolumnie znajdowała się liczba ij. Korzystając z polecenia `apply` znajdź:

- wektor średnich wartości w poszczególnych wierszach,
- wektor sum kolejnych kolumn.

Rozwiązanie Tworzenie macierzy 4×5:

```
N <- 4
M <- 5
mat <- matrix(0, nrow = N, ncol = M)
mat
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
```

Wypełnianie jej liczbami ij:

```
for (i in 1:N) {
  for (j in 1:M) {
    mat[i, j] <- i * j
  }
}
mat
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    2    4    6    8   10
## [3,]    3    6    9   12   15
## [4,]    4    8   12   16   20
```

Znalezienie wektora średnich wartości w poszczególnych wierszach:

```
row_means <- apply(mat, 1, mean)
row_means
```

```
## [1]  3  6  9 12
```

Znalezienie wektora sum kolejnych kolumn:

```
col_sum <- apply(mat, 2, sum)
col_sum
```

```
## [1] 10 20 30 40 50
```

Zadanie 4 – skośność i kurtoza

Treść Za pomocą polecenia `function` zdefiniuj funkcje obliczające dla podanego wektora danych ich skośność oraz kurtozę. Definicje odpowiednio skośności oraz kurtozy nadwyżkowej są następujące:

$$\frac{m_3}{\sigma^3} = \left\langle \left(\frac{x - \langle x \rangle}{\sigma} \right)^3 \right\rangle \approx \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \langle x \rangle)^3}{\left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \langle x \rangle)^2\right]^{3/2}}$$

$$\frac{m_4}{\sigma^4} - 3 = \left\langle \left(\frac{x - \langle x \rangle}{\sigma} \right)^4 \right\rangle - 3 \approx \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \langle x \rangle)^4}{\left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \langle x \rangle)^2\right]^{2}} - 3$$

Rozwiązanie Definicja funkcji do obliczania skośności:

```r
skew <- function(x) {
  n <- length(x)
  mean_x <- mean(x)
  sd_x <- sd(x)
  m3 <- sum((x - mean_x)^3) / n
  skew <- m3 / (sd_x^3)
  skew
}
```

Definicja funkcji do obliczania kurtozy:

```r
kurtosis <- function(x) {
  n <- length(x)
  mean_x <- mean(x)
  sd_x <- sd(x)
  m4 <- sum((x - mean_x)^4) / n
  kurtosis <- (m4 / (sd_x^4)) - 3
  kurtosis
}
```

Przykład użycia funkcji:

```r
x <- rnorm(1000) # losowe dane z rozkładu normalnego
skewness <- skew(x) # obliczenie skosnosci
skewness
```

```
## [1] 0.07055832
```

```r
kurtosis_value <- kurtosis(x) # obliczenie kurtozy
kurtosis_value
```

```
## [1] -0.02055072
```

Zadanie 5 – centralne twierdzenie graniczne

Treść Badamy rozkład próbkowy średniej z n zmiennych losowych o rozkładach z zadania 1 dla N replikacji.

a. Zająć się najpierw rozkładem dwumianowym z $k = 100$ i $p = 0.25$. Niech $N = 100000$ i $n = 2$. Umieścić liczby losowe o rozkładzie dwumianowym w macierzy $n \times N$, a następnie znaleźć średnie kolumn. Przedstawić histogram danych. Znaleźć ich średnią, odchylenie standardowe, skośność i kurtozę.
b. Rozważyć przypadek rozkładu wykładniczego z $\lambda = 1$ i przestawić histogramy dla $n = 1, 2, 10, 50$ na wspólnym wykresie. Liczba replikacji $N = 100000$. Wyciągnąć wnioski.
c. Dla powyższego rozkładu wykładniczego wyznaczyć skośność i kurtozę dla wszystkich $n \in [1, 100]$. Przedstawić je na wykresie.
d. Wykonać zadania b)-c) dla pozostałych rozkładów z Zadania 2.

Rozwiązanie

a) rozkład dwumianowy z $k = 100$ i $p = 0.25$:

Deklaracja parametrów:

```
k <- 100
p <- 0.25
N <- 100000
n <- 2
```

Generowanie macierzy $n \times N$:

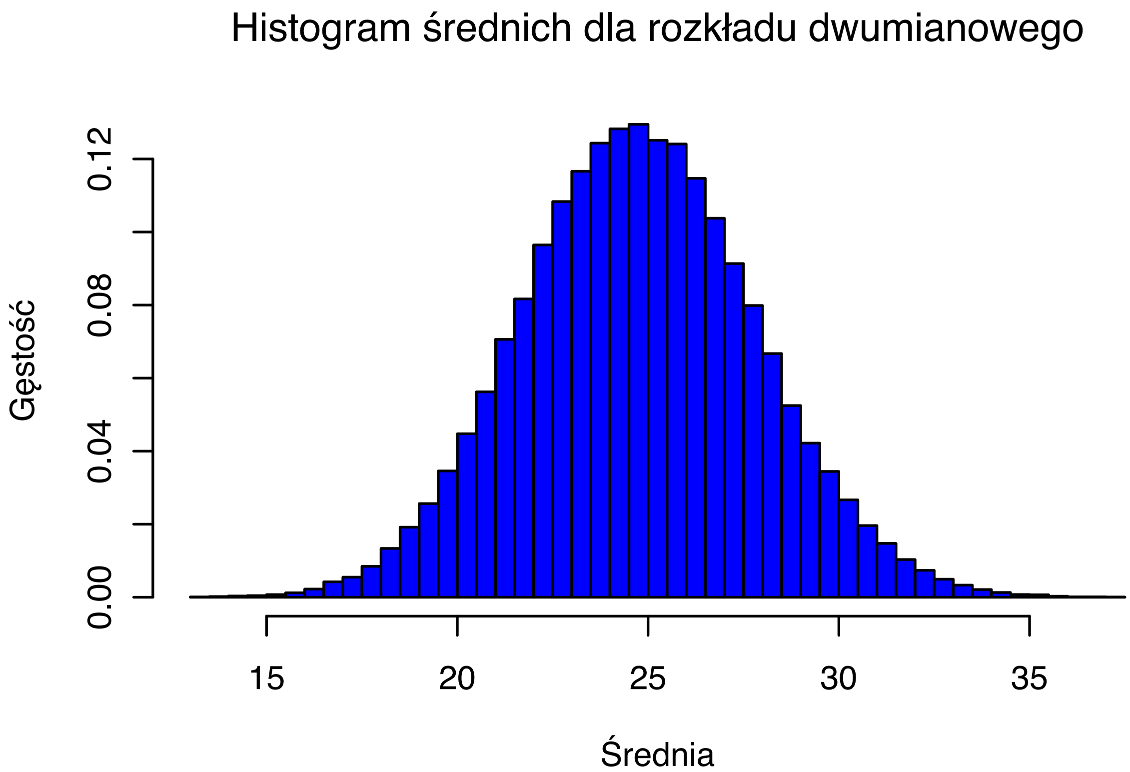
```
data_binom <- matrix(rbinom(n * N, k, p), nrow = n, ncol = N)
```

Znalezienie średnich kolumn:

```
means_binom <- apply(data_binom, 2, mean)
```

Przedstawienie histogramu:

```
hist(means_binom, breaks = 50, probability = TRUE,
     main = "Histogram średnich dla rozkładu dwumianowego",
     xlab = "Średnia",
     ylab = "Gęstość",
     col = "blue")
```



Obliczenie średniej, odchylenia standardowego, skośności i kurtozy:

```
mean_binom <- mean(means_binom)
mean_binom
```

```
## [1] 25.00474
```

```
sd_binom <- sd(means_binom)
sd_binom
```

```
## [1] 3.065309
```

```
skew_binom <- skew(means_binom)
skew_binom
```

```
## [1] 0.07375547
```

```
kurtosis_binom <- kurtosis(means_binom)
kurtosis_binom
```

```
## [1] -0.01110279
```

b) rozkład wykładniczy z $\lambda = 1$:

Deklaracja parametrów:

```
lambda = 1
N <- 100000
n_values <- c(1, 2, 10, 50)
```

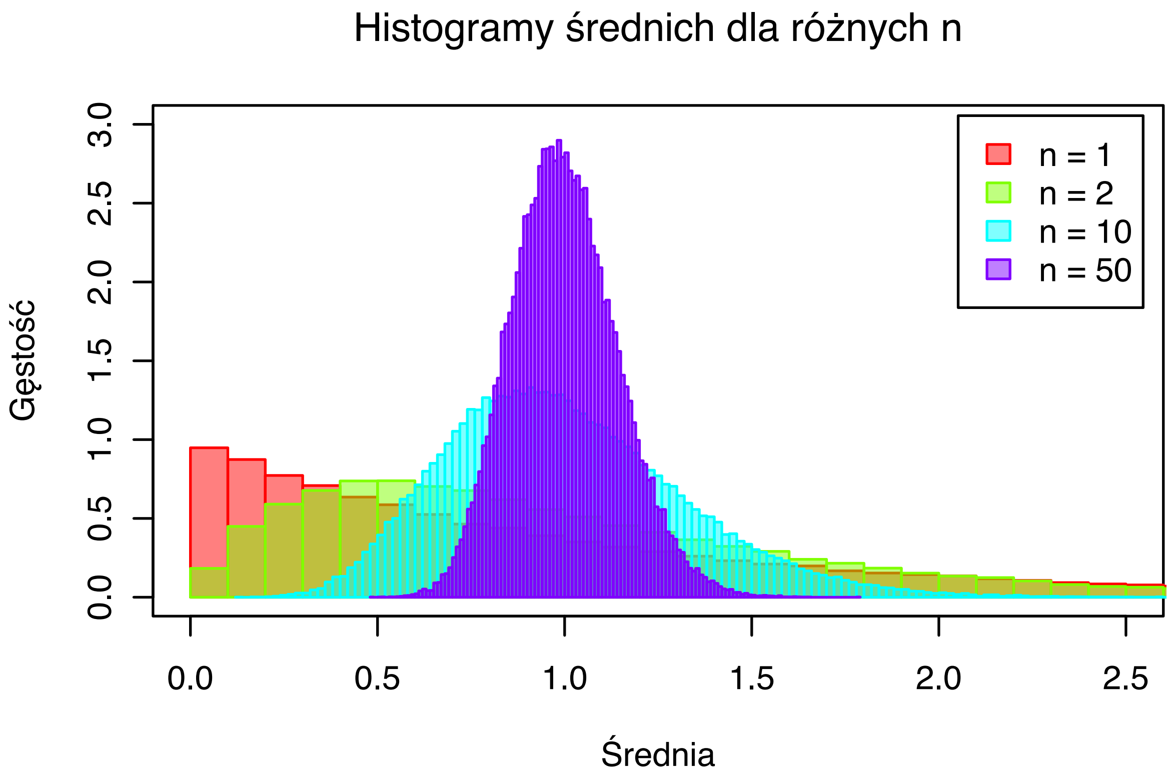
Generowanie histogramów dla różnych wartości n:

```
colors <- rainbow(length(n_values))

# Pusty wykres
plot(
  NA, xlim = c(0, 2.5), ylim = c(0, 3),
  main = "Histogramy średnich dla różnych n",
  xlab = "Średnia", ylab = "Gęstość"
)

# Histogram dla n = 1, 2, 10, 50
for (i in seq_along(n_values)) {
  n <- n_values[i]
  data_exp <- matrix(rexp(n * N, lambda), nrow = n, ncol = N)
  means_exp <- apply(data_exp, 2, mean)
  hist(
    means_exp, breaks = 100, probability = TRUE,
    col = adjustcolor(colors[i], alpha.f = 0.5),
    border = colors[i], add = TRUE
  )
}

# Legenda
legend(
  "topright", legend = paste("n =", n_values),
  fill = adjustcolor(colors, alpha.f = 0.5),
  border = colors, inset = 0.02
)
```


Histogramy średnich dla różnych n
Gęstość
Średnia
n = 1
n = 2
n = 10
n = 50
0.0
0.5
1.0
1.5
2.0
2.5
3.0

c) Wyznaczenie skośności i kurtozy dla wszystkich $n \in [1, 100]$ oraz ich przedstawienie na wykresie:

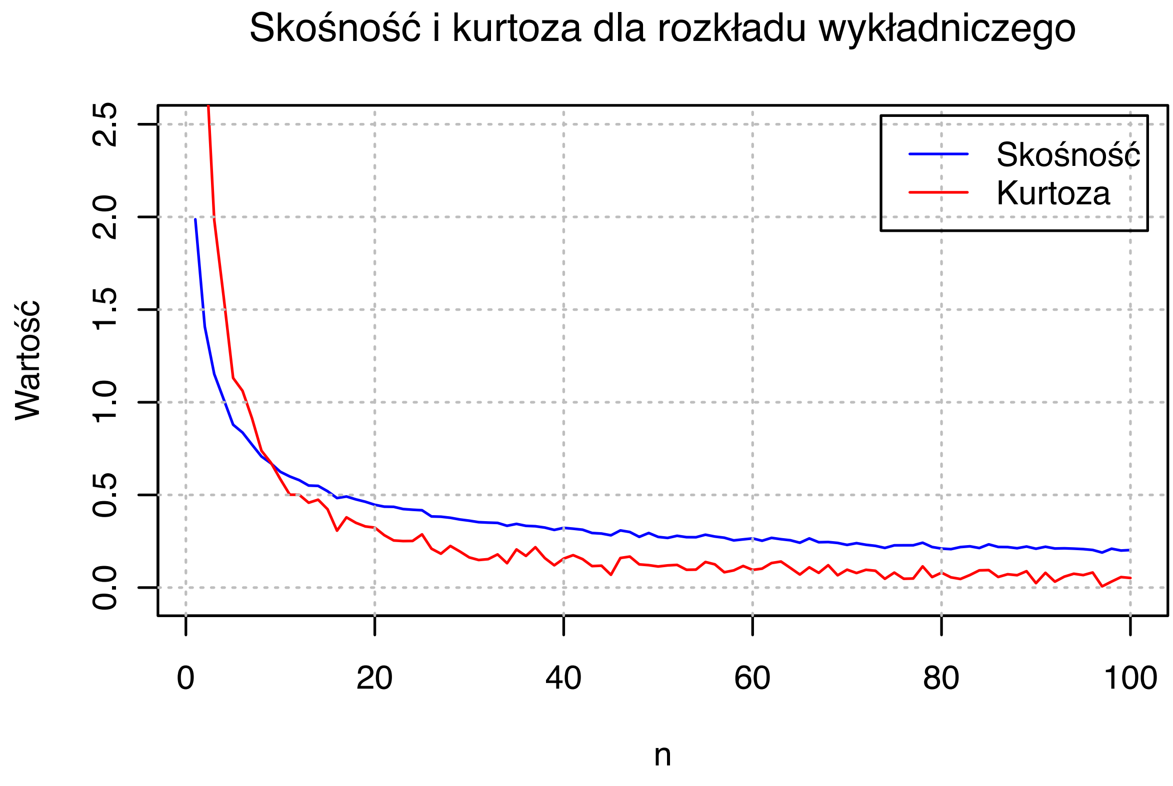
Deklaracja parametrów:

```
n_values <- 1:100
skew_values <- numeric(length(n_values))
kurtosis_values <- numeric(length(n_values))
N = 100000
```

Wyznacznie skośności i kurtozy dla każdego $n \in [1, 100]$:

```
for (n in n_values) {
  data_exp <- matrix(rexp(n * N, lambda), nrow = n, ncol = N)
  means_exp <- apply(data_exp, 2, mean)
  skew_values[n] <- skew(means_exp)
  kurtosis_values[n] <- kurtosis(means_exp)
}

plot(
  n_values, skew_values, type = "l", col = "blue",
  main = "Skośność i kurtoza dla rozkładu wykładniczego",
  xlab = "n", ylab = "Wartość", ylim = c(-0.05, 2.5),
)
lines(n_values, kurtosis_values, type = "l", col = "red")
grid(col = "gray", lty = "dotted")
legend(
  "topright", legend = c("Skośność", "Kurtoza"),
  col = c("blue", "red"), lty = 1, inset = 0.02,
  text.width = max(strwidth(c("Skośność", "Kurtoza"))) * 1.2,
)
```


Skośność i kurtoza dla rozkładu wykładniczego
Skośność
Kurtoza
Wartość
n
0.0
0.5
1.0
1.5
2.0
2.5
0
20
40
60
80
100