Projekt zaliczeniowy

Mikołaj Leonhardt & Kacper Słoniec

- Porównywane alogrytmy:
  - Metoda wielokrotnego startu (MS)
  - Poszukiwanie przypadkowe (PRS)
- Przeszukiwane funkcje:
  - funkcja Ackleya - $f(x) = -20 \cdot \exp(-0.2 \cdot \sqrt{(\frac{1}{n}\sum_{i=1}^{n} x_i)}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi \cdot x_i))$
  - funkcja Rastrigina - $f(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$

```
library(smoof)
```

Poszukiwanie przypadkowe (Pure Random Search, PRS)

Alogrytm przyjmuje liczbę punktów (sample_size), wymiar (dim), dolną granicę dziedziny (d_lower), górną granicę dziedziny (d_upper) oraz funkcję dla której przeprowadza porównanie (f). Zwraca minimalną uzyskaną wartość wylosowanego punktu.

```
pure_random_search <- function(sample_size, dim, d_lower, d_upper, f) {
    # creates a list of random vectors
    xs <- replicate(
        sample_size,
        runif(dim, min = d_lower, max = d_upper), simplify = FALSE)
    # applies function to every vector in a list
    ys <- lapply(xs, f)
    min(unlist(ys))
}
```

Metoda wielokrotnego startu (multi-start, MS)

Alogrytm przyjmuje liczbę punktów (sample_size), wymiar (dim), dolną granicę dziedziny (d_lower), górną granicę dziedziny (d_upper) oraz funkcję dla której przeprowadza porównanie (fn). Algorytm zwraca parę (min_value, avg_counts) gdzie min_value jest najmniejszą znalezioną przez funkcję optim wartością przeszukiwanej funkcji, a avg_counts jest średnią liczbą wywołań przeszukiwanej funkcje przez funkcję optim.

```
multi_start <- function(sample_size, dim, d_lower, d_upper, fn) {
    xs <- replicate(
        sample_size,
        runif(dim, min = d_lower, max = d_upper), simplify = FALSE)
    ms <- lapply(
        xs,
        function(x){
        optim_result <- optim(x, fn, method = "L-BFGS-B", lower = d_lower, upper = d_upper)
        c(value = optim_result$value, counts = optim_result$counts[[1]])
        })
    values <- sapply(ms, function(x) x["value"])
    counts <- sapply(ms, function(x) x["counts"])
    counts <- na.omit(counts)
```

```
    # returns a minimum of found local minimums
    list(min_value = min(unlist(values)), avg_counts = mean(unlist(counts)))
}
```

Porównywanie algorytmów wybraną funkcją

Algorytm przyjmuje liczbę punktów (sample_size), nazwę funkcji (fn_name) - "ackley" lub "rastrigin" oraz wymiar (dimensions). Algorytm zwraca listę z wylosowanymi wartościami minimalnymi (osobny wektor prs i ms). Algorytm respektuje dziedziny wywoływanych funkcji, które określone są w atrybucie par.set.
Dla funkcji Ackleya: $x_i \in [-32.768, 32.768]$.
Dla funkcji Rastrigina: $x_i \in [-5.12, 5.12]$.

```
compare_fs <- function(sample_size, fn_name, dimensions) {
    fn <- switch(
        fn_name,
        "ackley" = makeAckleyFunction(dimensions),
        "rastrigin" = makeRastriginFunction(dimensions)
        )

    atts <- attributes(fn)
    domain_lower <- rep(atts$par.set[[1]][[1]]$lower, dimensions)
    domain_upper <- rep(atts$par.set[[1]][[1]]$upper, dimensions)
    ms_points_n <- 100

    result <- list(ms = rep(0, sample_size), prs = rep(0, sample_size))

    avg_prs_result <- 0
    avg_ms_result <- 0
    for (i in 1:sample_size){

        ms_result_list <- multi_start(
        ms_points_n, dimensions, domain_lower, domain_upper, fn)

        result$ms[i] <- ms_result_list$min_value

        prs_points_n <- ceiling(ms_result_list$avg_counts) * ms_points_n
        prs_result <- pure_random_search(prs_points_n, dimensions, domain_lower, domain_upper, fn)

        result$prs[i] <- prs_result
    }

    result
}
```

Wykonanie porównania

```
sample_size <- 50
ackley2d <- compare_fs(sample_size, "ackley", 2)
ackley10d <- compare_fs(sample_size, "ackley", 10)
ackley20d <- compare_fs(sample_size, "ackley", 20)

rastrigin2d <- compare_fs(sample_size, "rastrigin", 2)
rastrigin10d <- compare_fs(sample_size, "rastrigin", 10)
rastrigin20d <- compare_fs(sample_size, "rastrigin", 20)
```

Rysowanie diagramów

Algorytm przyjmuje wektor na podstawie którego rysuje wykresy

```
draw_diagrams <- function(result) {

    layout(matrix(c(1, 2), 1, 2, byrow = TRUE))

    hist(result$ms, prob = TRUE, breaks=20, col=rgb(1,0,0,0.5),
    main="Metoda wielokrotnego startu", xlab="Wartość minimum", ylab="Gęstość")
    x <- seq(min(result$ms), max(result$ms), length = 40)
    f <- dnorm(x, mean = mean(result$ms), sd = sd(result$ms))
    lines(x, f, col = "black", lwd = 2)

    boxplot(result$ms, col=rgb(1, 0, 0, 0.5), ylab="Częstotliwość")

    print(paste("Średni wynik: ", mean(result$ms)))

    layout(matrix(c(1, 2), 1, 2, byrow = TRUE))

    hist(result$prs, prob = TRUE, breaks=20, col=rgb(0,0,1,0.5),
    main="Poszukiwanie przypadkowe", xlab="Wartość minimum", ylab="Gęstość")
    x <- seq(min(result$prs), max(result$prs), length = 40)
    f <- dnorm(x, mean = mean(result$prs), sd = sd(result$prs))
    lines(x, f, col = "black", lwd = 2)

    boxplot(result$prs, col=rgb(0,0,1,0.5), ylab="Częstotliwość")

    print(paste("Średni wynik: ", mean(result$prs)))
}
```

Opracowanie wyników

Statystyka różnic między średnimi: $T = \frac{(\overline{PRS} - \overline{MS}) - (m_1 - m_2)}{Sp \cdot \sqrt{\frac{1}{n} + \frac{1}{n}}} \sim t2n - 2$

Z racji tego, że nie znamy rzeczywistego odchylenia standardowego rozkładu średnich minimów oraz korzystając z Centralnego Twierdzenia Granicznego możemy do wykonania analizy istotności statystycznej T można użyć testu Studenta implemenotwanego przez funkcję t.test.

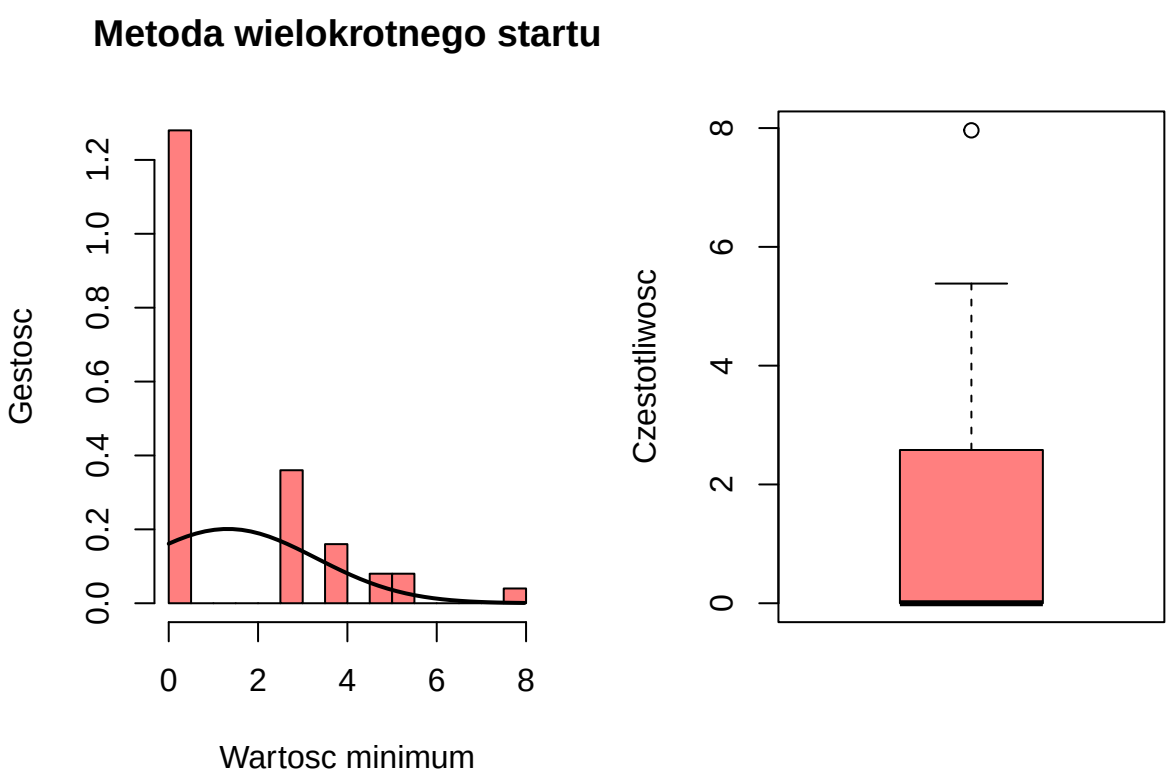
Hipoteza zerowa: różnica między średnimi jest równa 0.
Hipoteza alternatywna: różnica między średimi jest różna od zera.

Funkcja t.test wylicza również 95% przedział ufności dla podanej statystyki.

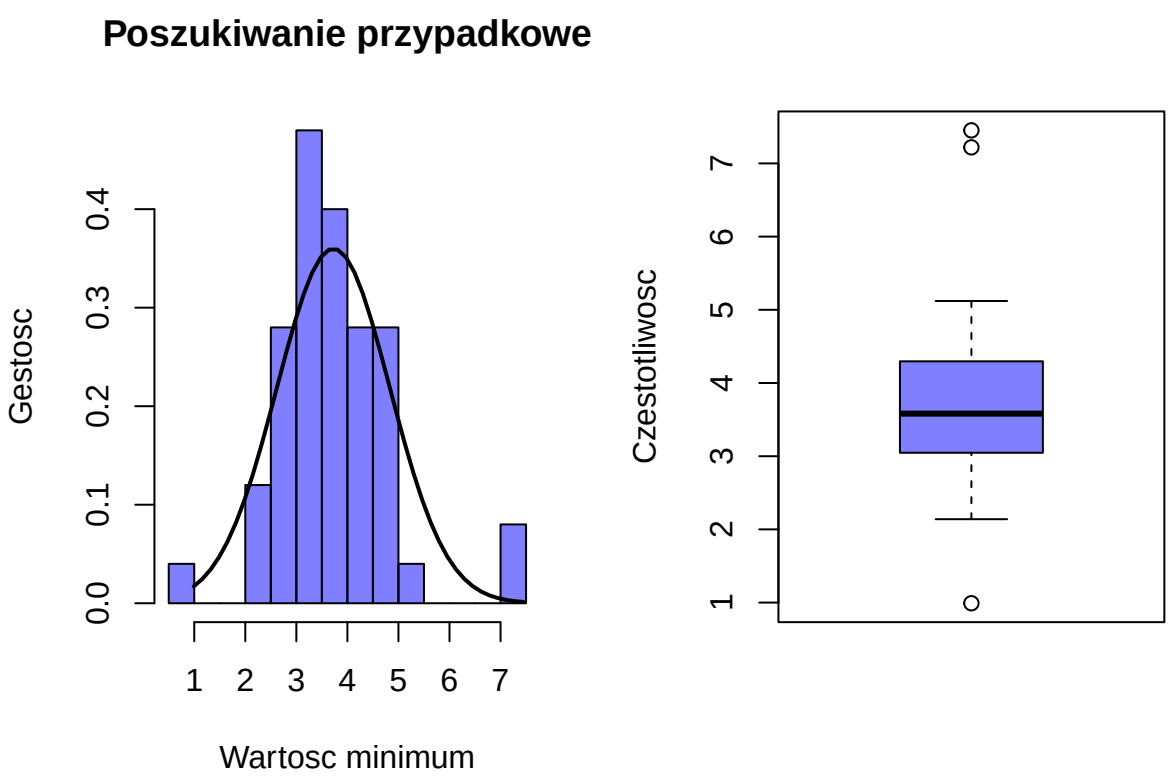
Funkcja Ackley'a

2 wymiary

```
draw_diagrams(ackley2d)
```




```
## [1] "Średni wynik:  1.32021451376942"
```




```
## [1] "Średni wynik:  3.72499919369011"
```

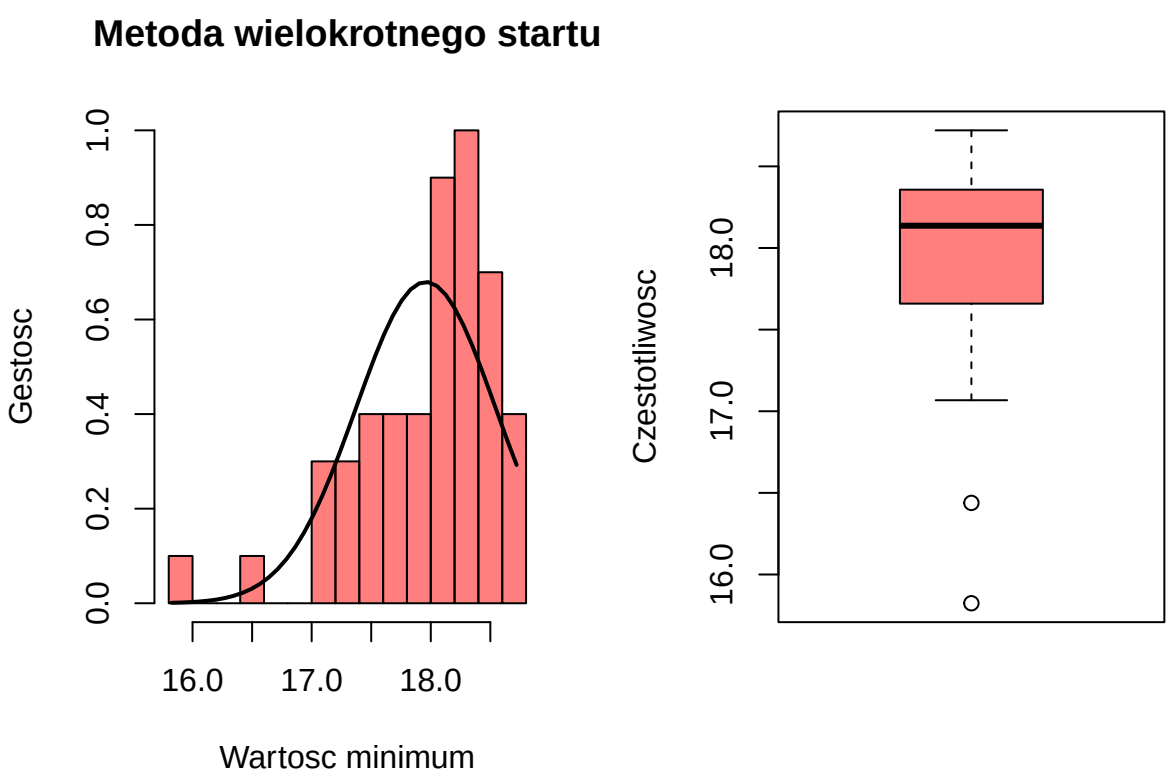
```
ttest <- t.test(ackley2d$ms, ackley2d$prs)
print(ttest)
```

Test Studenta dla różnicy pomiędzy średnimi minimami

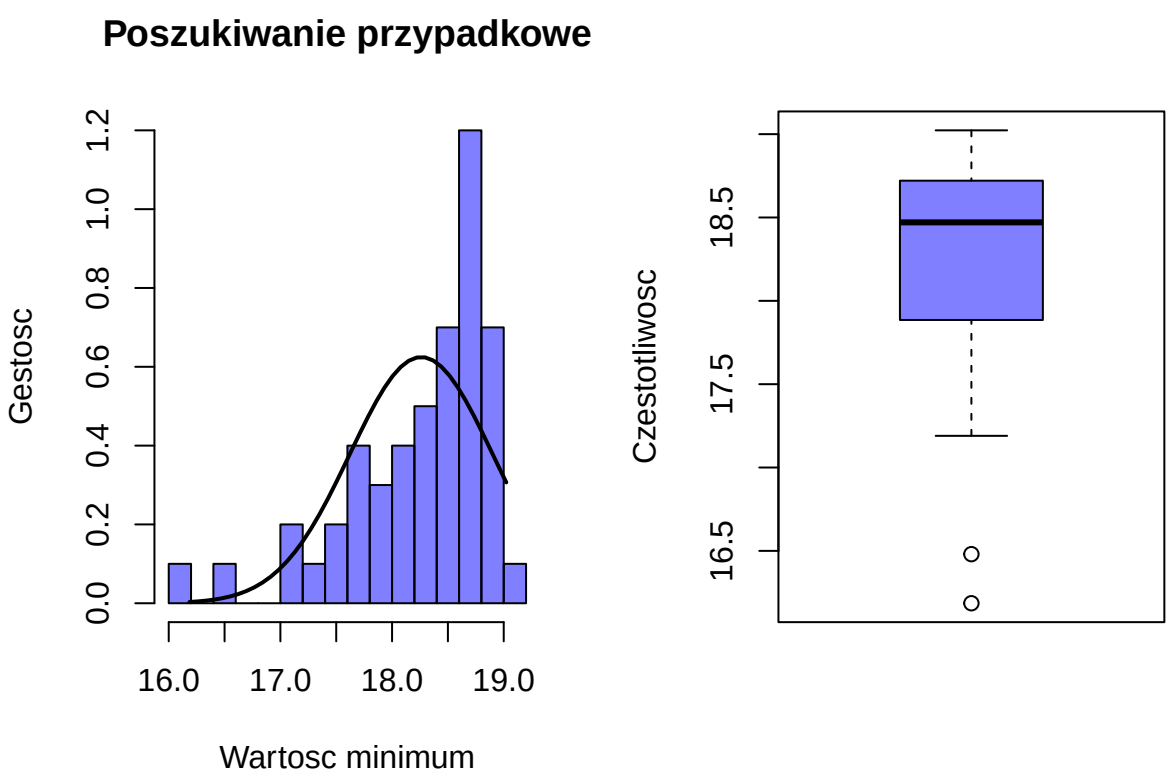
```
##
##  Welch Two Sample t-test
##
## data:  ackley2d$ms and ackley2d$prs
## t = -7.4823, df = 76.865, p-value = 1.012e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.044782 -1.764787
## sample estimates:
## mean of x mean of y
##  1.320215  3.724999
```

10 wymiarów

```
draw_diagrams(ackley10d)
```




```
## [1] "Średni wynik:  17.9582052810938"
```




```
## [1] "Średni wynik:  18.2610614883699"
```

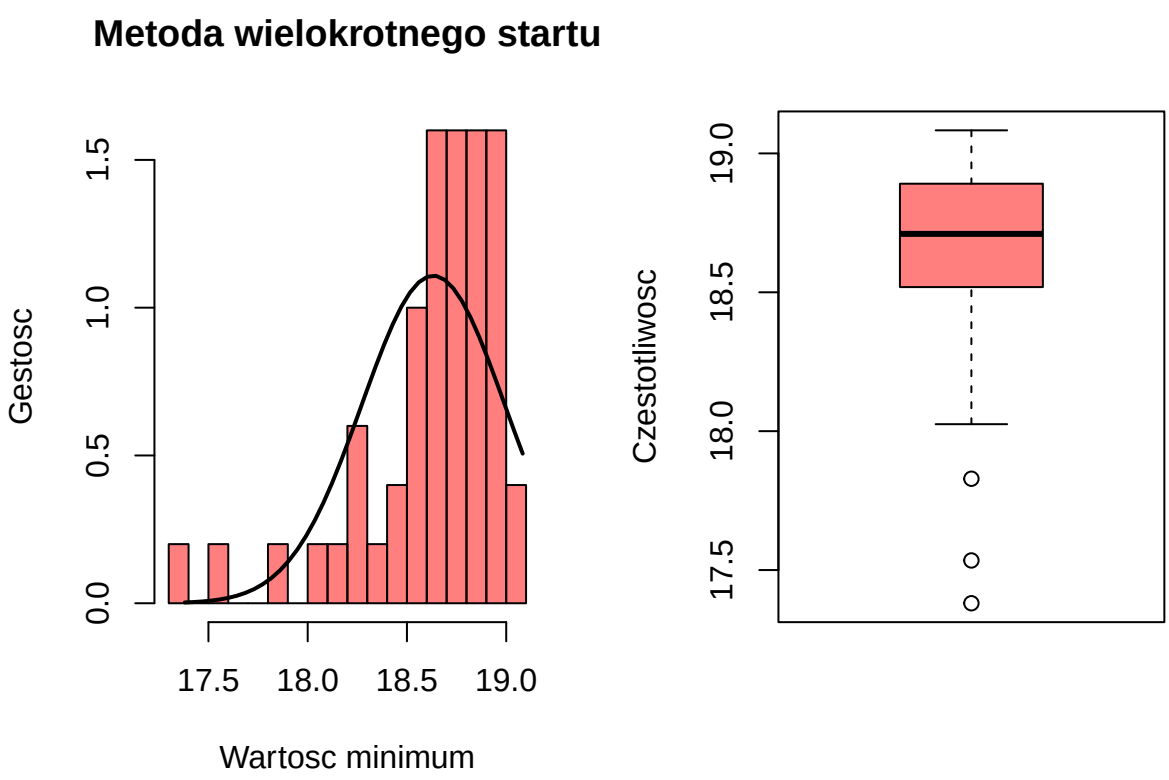
```
ttest <- t.test(ackley10d$ms, ackley10d$prs)
print(ttest)
```

Test Studenta dla różnicy pomiędzy średnimi minimami

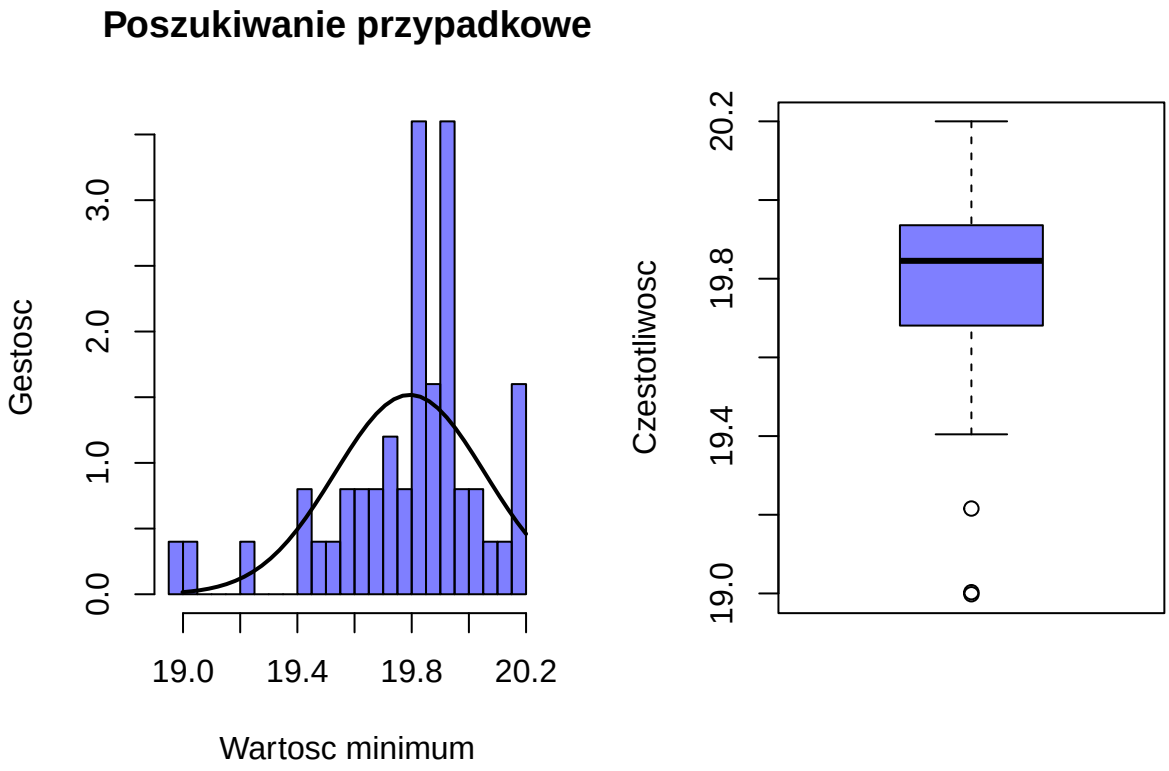
```
##
##  Welch Two Sample t-test
##
## data:  ackley10d$ms and ackley10d$prs
## t = -2.4687, df = 97.318, p-value = 0.01531
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.54633184 -0.05938057
## sample estimates:
## mean of x mean of y
##  17.95821  18.26106
```

20 wymiarów

```
draw_diagrams(ackley20d)
```




```
## [1] "Średni wynik:  18.6323766125116"
```




```
## [1] "Średni wynik:  19.793651033241"
```

```
ttest <- t.test(ackley20d$ms, ackley20d$prs)
print(ttest)
```

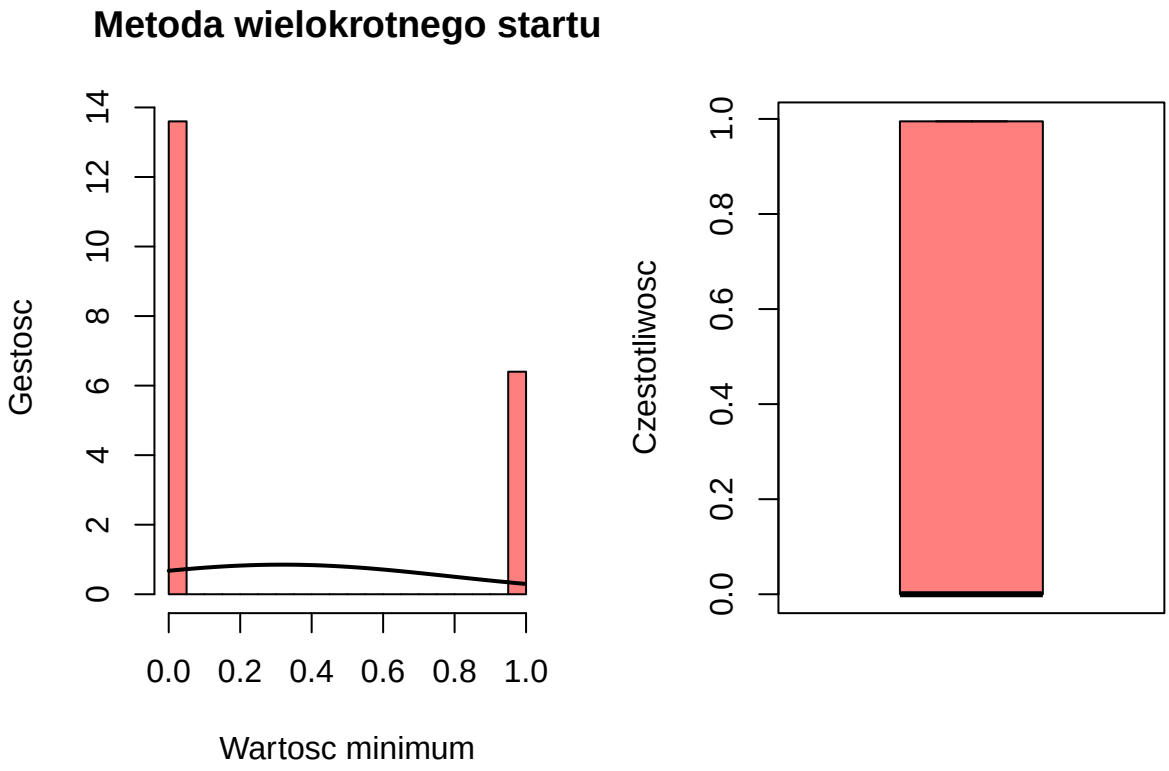
Test Studenta dla różnicy pomiędzy średnimi minimami

```
##
##  Welch Two Sample t-test
##
## data:  ackley20d$ms and ackley20d$prs
## t = -18.427, df = 89.683, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.286480 -1.036069
## sample estimates:
## mean of x mean of y
##  18.63238  19.79365
```

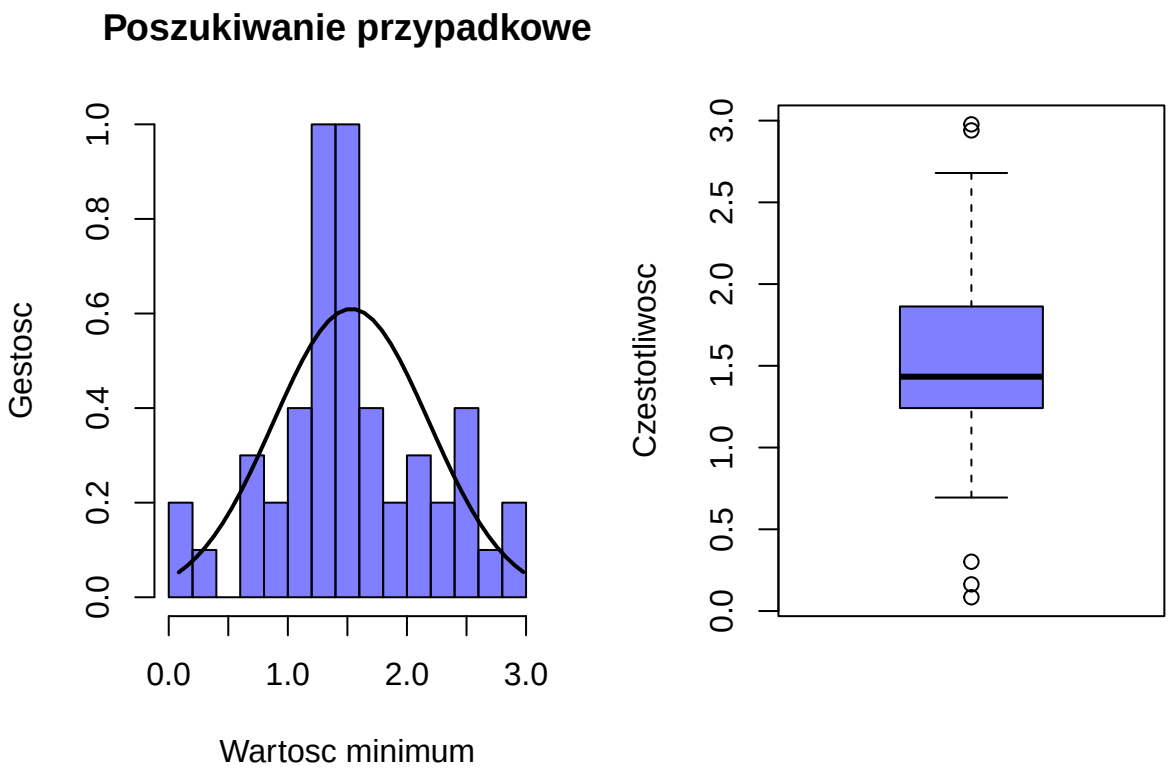
Funkcja Rastrigin'a

2 wymiary

```
draw_diagrams(rastrigin2d)
```




```
## [1] "Średni wynik:  0.318386898269936"
```




```
## [1] "Średni wynik:  1.53224360661248"
```

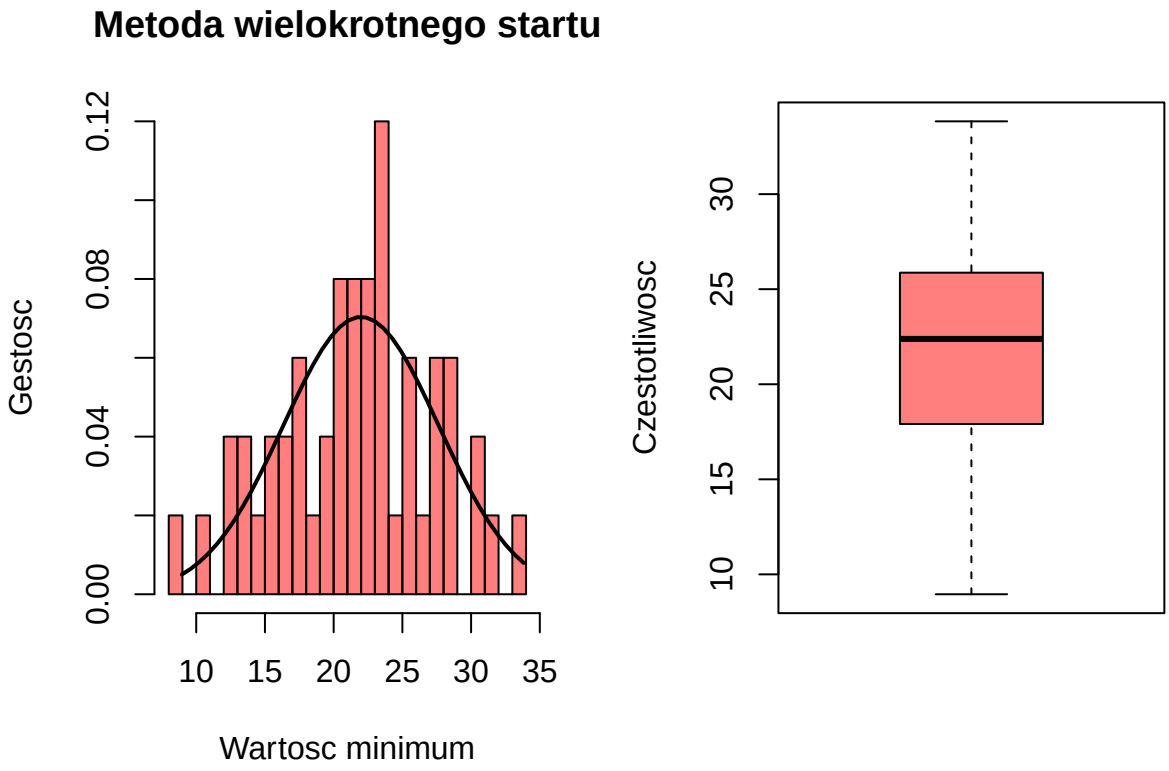
```
ttest <- t.test(rastrigin2d$ms, rastrigin2d$prs)
print(ttest)
```

Test Studenta dla różnicy pomiędzy średnimi minimami

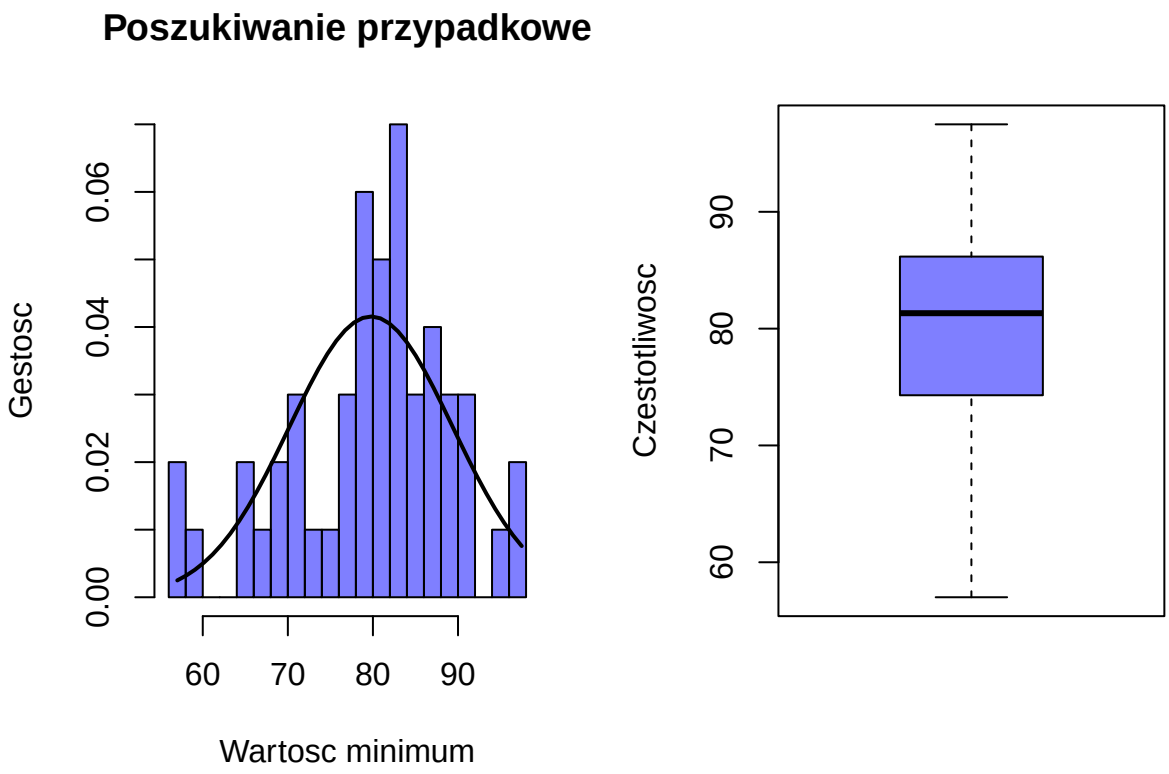
```
##
##  Welch Two Sample t-test
##
## data:  rastrigin2d$ms and rastrigin2d$prs
## t = -10.663, df = 88.821, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.4400476 -0.9876658
## sample estimates:
## mean of x mean of y
## 0.3183869 1.5322436
```

10 wymiarów

```
draw_diagrams(rastrigin10d)
```




```
## [1] "Średni wynik:  21.9885480220251"
```




```
## [1] "Średni wynik:  79.7784831369155"
```

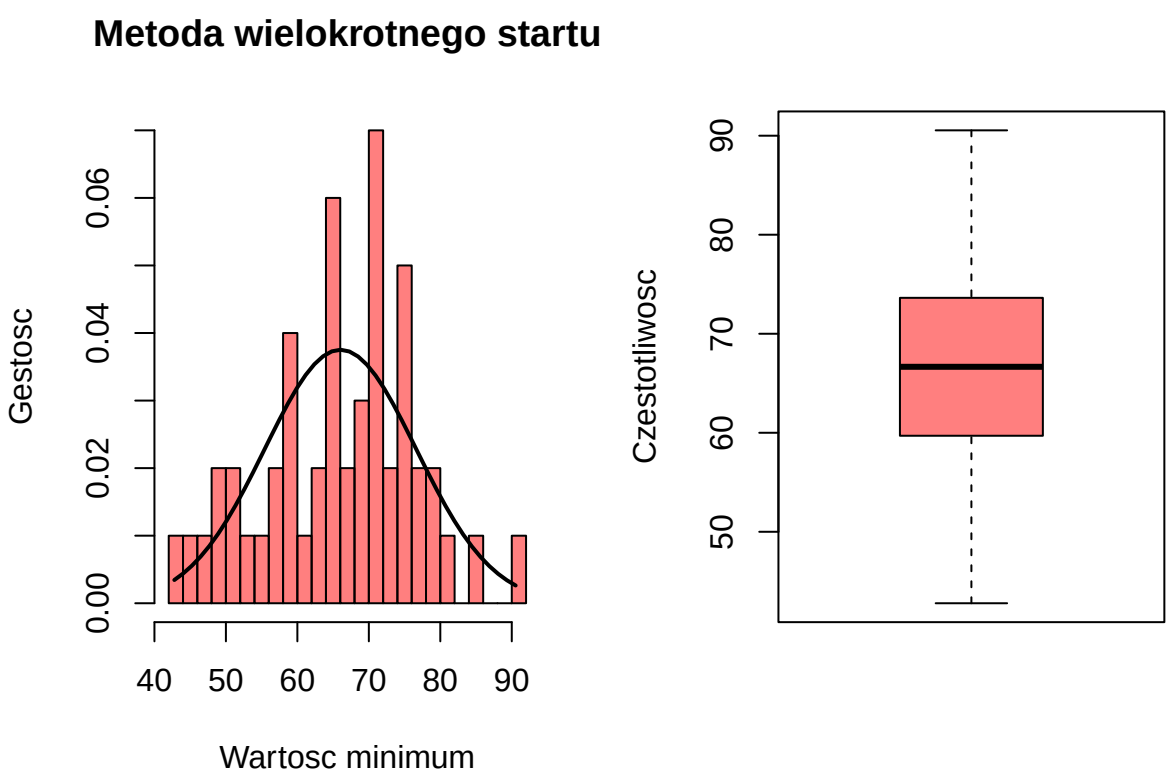
```
ttest <- t.test(rastrigin10d$ms, rastrigin10d$prs)
print(ttest)
```

Test Studenta dla różnicy pomiędzy średnimi minimami

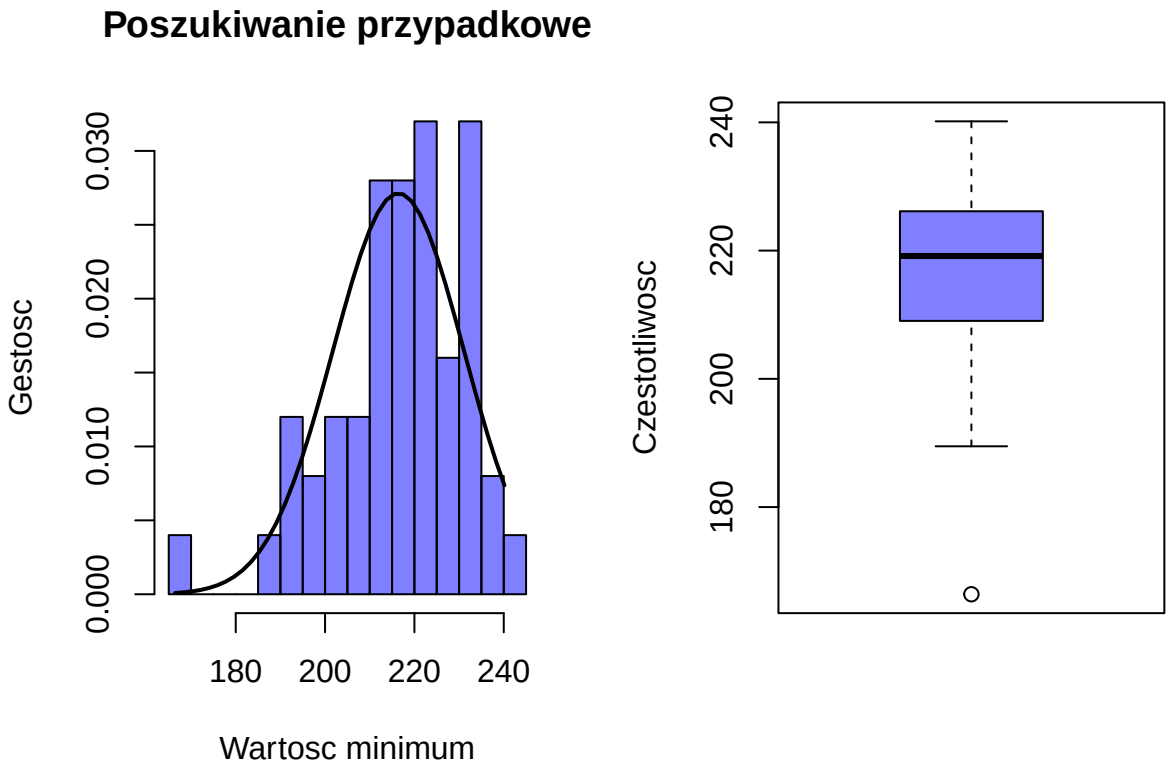
```
##
##  Welch Two Sample t-test
##
## data:  rastrigin10d$ms and rastrigin10d$prs
## t = -36.653, df = 79.464, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -60.92795 -54.65192
## sample estimates:
## mean of x mean of y
##  21.98855  79.77848
```

20 wymiarów

```
draw_diagrams(rastrigin20d)
```




```
## [1] "Średni wynik:  66.0252546614804"
```




```
## [1] "Średni wynik:  216.425531830993"
```

```
ttest <- t.test(rastrigin20d$ms, rastrigin20d$prs)
print(ttest)
```

Test Studenta dla różnicy pomiędzy średnimi minimami

```
##
##  Welch Two Sample t-test
##
## data:  rastrigin20d$ms and rastrigin20d$prs
## t = -58.613, df = 89.243, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -155.4986 -145.3019
## sample estimates:
## mean of x mean of y
##  66.02525 216.42553
```

Analiza zawartości diagramów

W większości przypadków dane są skupione wokół średniej oraz mediany (których wartości są do siebie zbliżone) i rozproszone wokół nich. Występuje wąski przedział między Q_1 a Q_3, co oznacza dużą koncentrację danych wokół środka rozkładu. Rozkład danych we wszystkich przypadkach poza dwoma przypomina rozkład naturalny. Dla 2. wymiarów MS (Metoda wielokrotnego startu) daje wyniki rozbieżne, a dane od siebie odstają.

Porównując średnie, w każdym przypadku mniejszy średni wynik zwraca MS, co jest szczególnie widoczne w 2. wymiarach i przy funkcji Rastrigin'a. Różnice średnich dwóch algorytmów są największe przy 10. i 20. wymiarach funkcji Rastrigin'a.

Wykres zbiorczy

```
layout(matrix(c(1, 2), 1, 2, byrow = TRUE))


ackley_mean_diff <-
    c(mean(ackley2d$ms), mean(ackley10d$ms), mean(ackley20d$ms)) -
    c(mean(ackley2d$prs), mean(ackley10d$prs), mean(ackley20d$prs))

plot(c(2,10,20), ackley_mean_diff, main = "Funkcja Ackleya", xlab = "wymiar",
ylab = "różnica średnich minimów")

rastrigin_mean_diff <-
    c(mean(rastrigin2d$ms), mean(rastrigin10d$ms), mean(rastrigin20d$ms)) -
    c(mean(rastrigin2d$prs), mean(rastrigin10d$prs), mean(rastrigin20d$prs))

plot(c(2,10,20), rastrigin_mean_diff, main = "Funkcja Rastrigina", xlab = "wymiar",
ylab = "różnica średnich minimów")
```