Rozkłady teoretyczne, rozkłady empiryczne, estymacja, symulacje

Bartosz Maćkiewicz

Różne rodzaje rozkładów

- Rozkład empiryczny, a rozkład teoretyczny.
- Próba a populacja.
- Rozkład w próbie (empiryczny), a rozkład w populacji (teoretyczny).

Rozkład teoretyczny

- Opisany przez funkcję matematyczną.
- Określony przez parametry tej funkcji.

Estymator

- Estymator jest funkcją tak skonstruowaną, aby jej wartości "dobrze" szacowały "prawdziwa" wartość parametru z rozkładu.
- Sam estymator też jest zmienną losową zdeterminowaną przez rozkład próby.
- Przyjmujemy, że zmienna losowa w populacji ma jakiś rozkład i za pomocą estymatorów próbujemy oszacować jej parametry.

Estymator

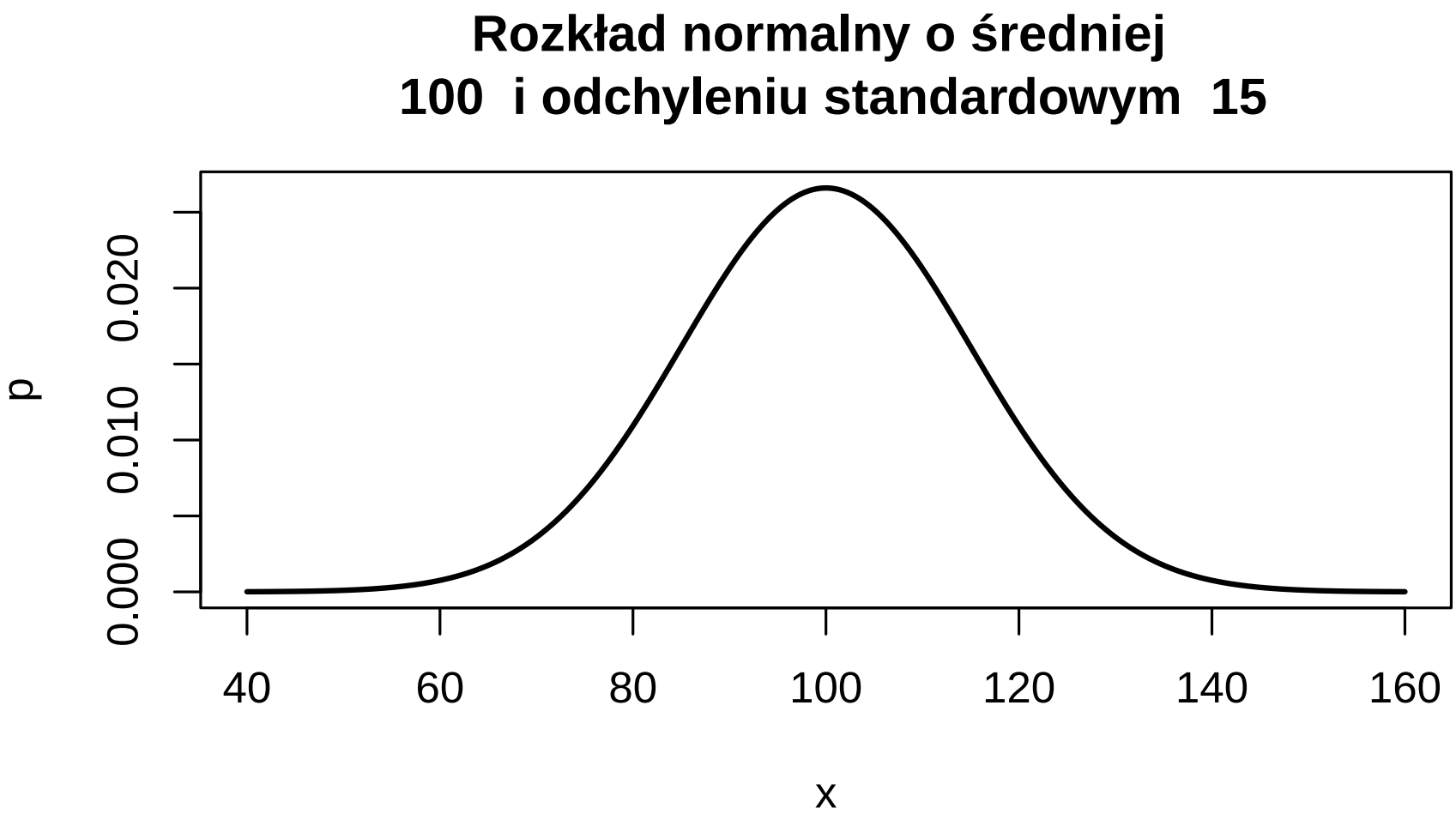
- **Nieobciążoność** - przy wielokrotnym powtarzaniu doświadczenia średnia z wartości estymatora jest równa szacowanemu parametrowi
- **Dostateczność** - estymator zawiera wszystkie możliwe informacje na temat parametru zawarte próbie.
- **Efektywność** - estymator ma najmniejszą możliwą wariancję, czyli jest stabilny
- **Zgodność** - ze zwiększeniem liczebności próby uzyskujemy coraz większe prawdopodobieństwo tego, że estymator będzie przyjmował wartości coraz bliższe szacowanym parametrom

Rozkłady teoretyczne w R

- W R wbudowane są funkcje związane ze wszystkimi najważniejszymi w statystyce rozkładami teoretycznymi.
- Pełną listę dostępnych rozkładów można zobaczyć, wpisując `?Distributions`
- Dla każdego rozkładu istnieją cztery funkcje:
  - `d...` jak *density* np. `dnorm(x, mu, sigma)`
    - funkcja gęstości rozkładu
  - `q...` jak *quantile* np. `qnorm(p, mu, sigma)`
    - funkcja kwantylowa (odwrotna dystrybuanta)
  - `p...` jak *probability* np. `pnorm(q, mu, sigma)`
    - dystrybuanta (funkcja wyznaczająca rozkład prawdopodobieństwa)
  - `r...` jak *random* np. `rnorm(n, mu, sigma)`
    - generator liczb losowych

Na przykład, żeby narysować rozkład normalny, możemy użyć poniższego kodu:

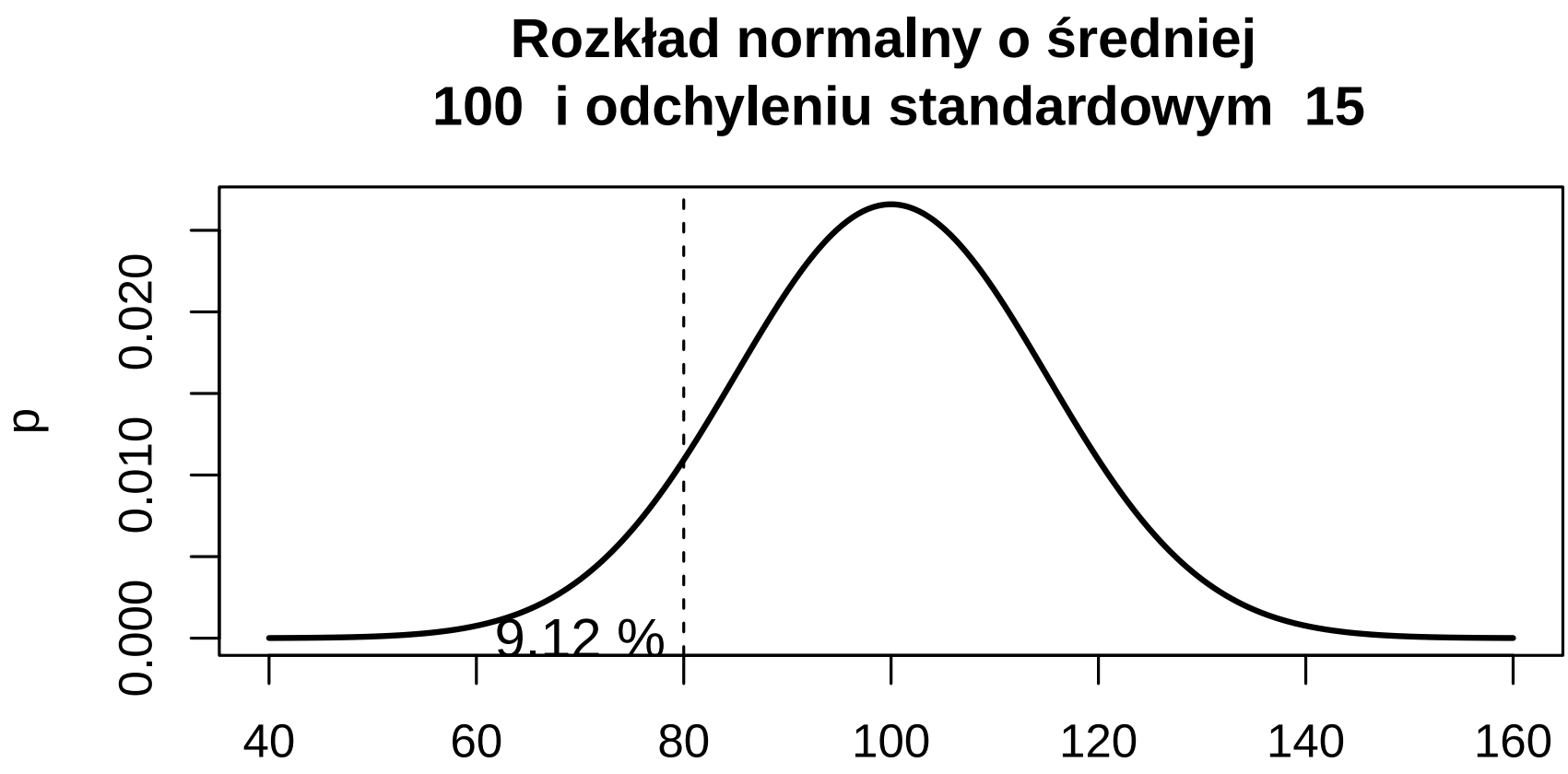
```
sr = 100; odch = 15
x = seq(-4, 4, length=1000) * odch + sr
p = dnorm(x, sr, odch)
plot(x, p, type="l", lwd=2,
     main=paste("Rozkład normalny o średniej \n", sr,
        " i odchyleniu standardowym ", odch))
```



```r
pnorm(80, 100, 15)
```

```
## [1] 0.09121122
```

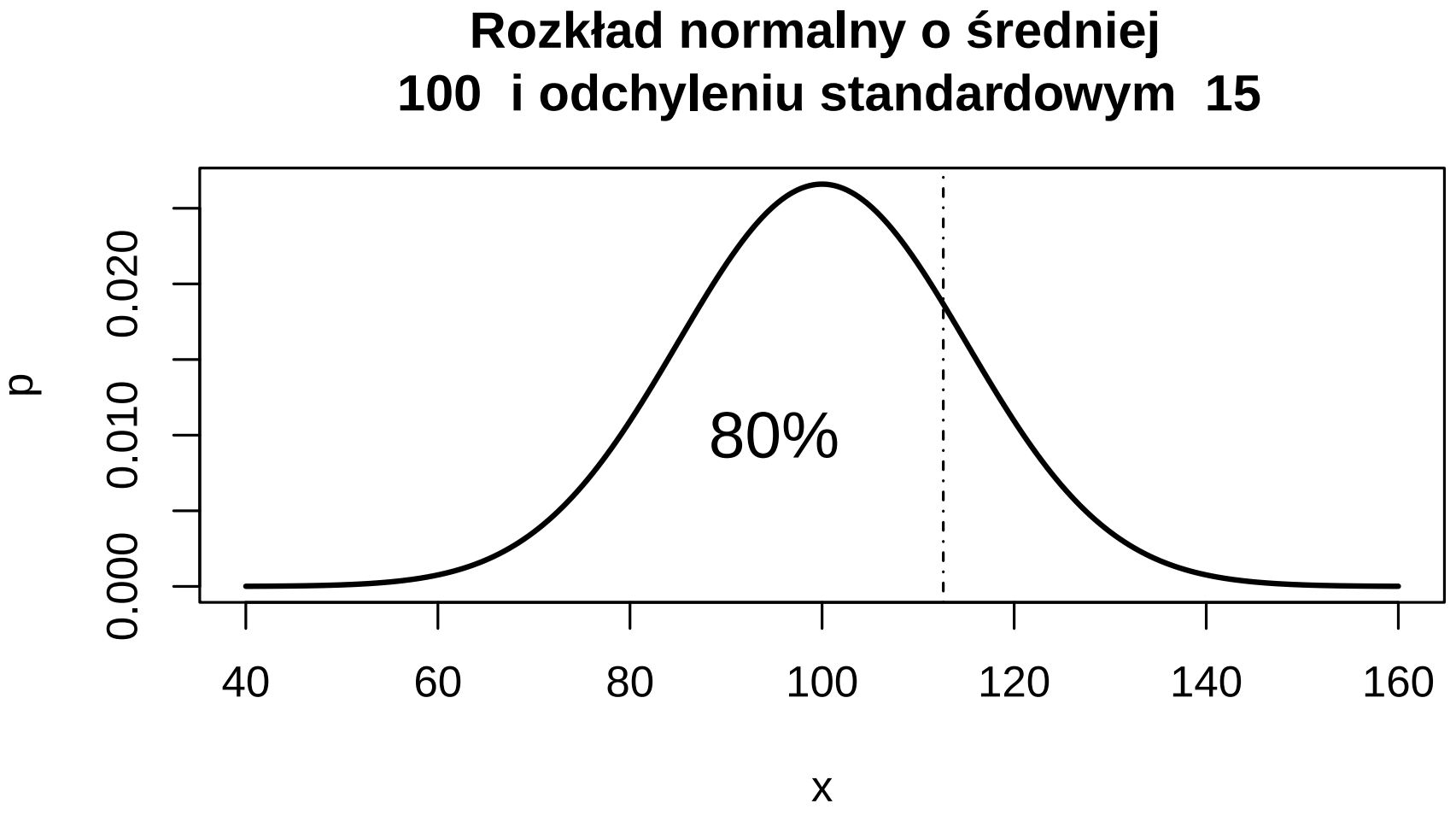
```r
plot(x, p, type="l", lwd=2,
     main=paste("Rozkład normalny o średniej \n", sr,
           " i odchyleniu standardowym ", odch))
abline(v=80, lty=2)
text(x=70, y=0,
     labels=paste(round(pnorm(80, 100, 15)*100, 2),"%"), cex=1.2)
```



```r
qnorm(.8, 100, 15)
```

```
## [1] 112.6243
```

```r
plot(x,p, type="l", lwd=2,
     main=paste("Rozkład normalny o średniej \n", sr,
        " i odchyleniu standardowym ", odch))
abline(v=qnorm(.8, 100, 15), lty=4)
text(x=95, y=.01, labels="80%", cex=1.5)
```



Pewien test diagnostyczny wskazuje na ryzyko problemów badanego dziecka, jeśli jego wynik nie przekracza dziesiątego percentyla. Zakładając, że wyniki testu mają rozkład normalny ze średnią 150 i odchyleniem standardowym 30, określ górną granicę wyników, które wskazują na możliwe problemy dziecka.

```
qnorm(0.1, 150,30)
```

```
## [1] 111.5535
```

set.seed

1. Komputer generuje liczby pseudolosowe
2. Żeby ktoś łatwo mógł zreplikować naszą symulację, używamy `set.seed`.

```
set.seed(12345)
```

3. To jest seed z którym robimy wszystkie zadania.

Symulacje - rzut kostką

Wykonajmy symulację 6000 rzutów kostką. Policzmy, ile było poszczególnych wyników.

```
kostka = 1:6
symulowane_rzuty = sample(kostka, size = 6000,
                          replace = TRUE)
table(symulowane_rzuty)
```

```
## symulowane_rzuty
##    1    2    3    4    5    6
## 1019  981  995 1007 1002  996
```

Symulacje - rzut kostką dalej

Jak często otrzymujemy więcej niż 1030 jedynek?

```
jedynki = replicate(1000, {
table(sample(kostka, size = 6000, replace = TRUE))[1]
})
length(jedynki[jedynki > 1030])/length(jedynki)
```

```
## [1] 0.144
```

Symulacje - rzut monetą

Rozważmy dziesięciokrotny rzut symetryczną monetą. Jakie jest prawdopodobieństwo wyrzucenia:

- dokładnie 4 orłów,
- co najmniej 4 orłów.

Symulacje - rzut monetą

Dokładnie 4 orły.

```
dbinom(x=4, size=10, prob=0.5)
```

```
## [1] 0.2050781
```

```
s = rbinom(size=10, n=10000, prob = 0.5)
length(s[s==4])/length(s)
```

```
## [1] 0.2042
```

Symulacje - rzut monetą

Co najmniej 4 orły.

```
sum(dbinom(x=4:10, size=10, prob=0.5))
```

```
## [1] 0.828125
```

```
s = rbinom(size=10, n=10000, prob = 0.5)
length(s[s>=4])/length(s)
```

```
## [1] 0.8234
```

Symulacje - delegacja

Jako szef kampusu masz za zadanie skompletować delegację składającą się z 7 członków organizacji. Organizacja składa się z 18 kobiet oraz 15 mężczyzn. Jakie jest prawdopodobieństwo, że w losowo wybranej delegacji będzie więcej niż 4 mężczyzn?

```
sum(dhyper(5:(15+18), 15, 18, 7))
```

```
## [1] 0.1301446
```

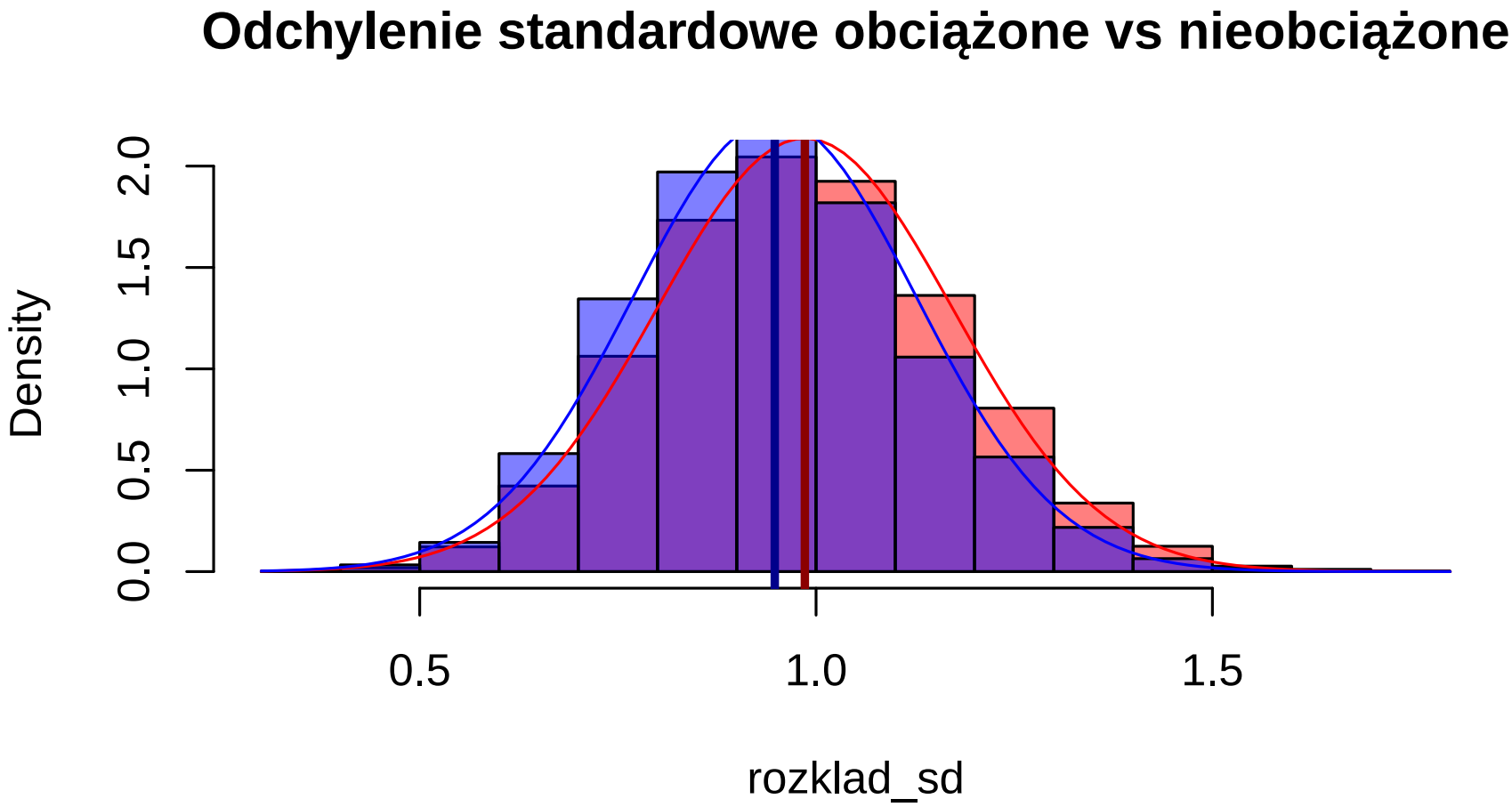
```
org = c(rep('M', 15), rep('K', 18))
s = replicate(1000, table(sample(org,7))['M'])
length(which(s>4))/length(s)
```

```
## [1] 0.127
```

Estymatory

```
t_red = adjustcolor('red', alpha.f = 0.5)
t_blue = adjustcolor('blue', alpha.f = 0.5)

biased_sd = function(x){
    sqrt((sum((x-mean(x))**2))/length(x))
}

rozklad_sd = replicate(10000, sd(rnorm(15)))

rozklad_biased_sd = replicate(10000, biased_sd(rnorm(15)))

hist(rozklad_sd, col = t_red, freq = FALSE,
     main = "Odchylenie standardowe obciążone vs nieobciążone")

hist(rozklad_biased_sd, add = TRUE, col = t_blue, freq = FALSE)

curve(dnorm(x, mean = mean(rozklad_sd), sd = sd(rozklad_sd)),
      col = 'red', add = TRUE)

curve(dnorm(x, mean = mean(rozklad_biased_sd),
        sd = sd(rozklad_biased_sd)), col = 'blue', add = TRUE)

abline(v=mean(rozklad_sd), col='darkred')
abline(v=mean(rozklad_biased_sd), col='darkblue')
```

Estymatory


Odchylenie standardowe obciążone vs nieobciążone
Density
0.0
0.5
1.0
1.5
2.0
0.5
1.0
1.5
rozklad_sd