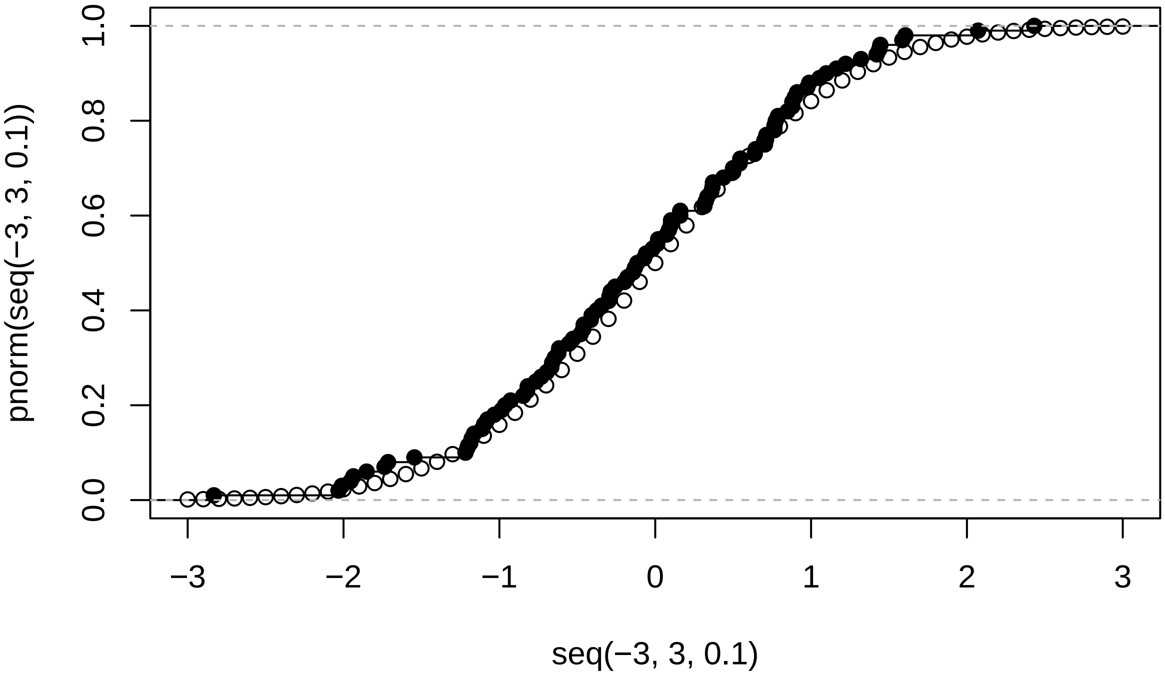
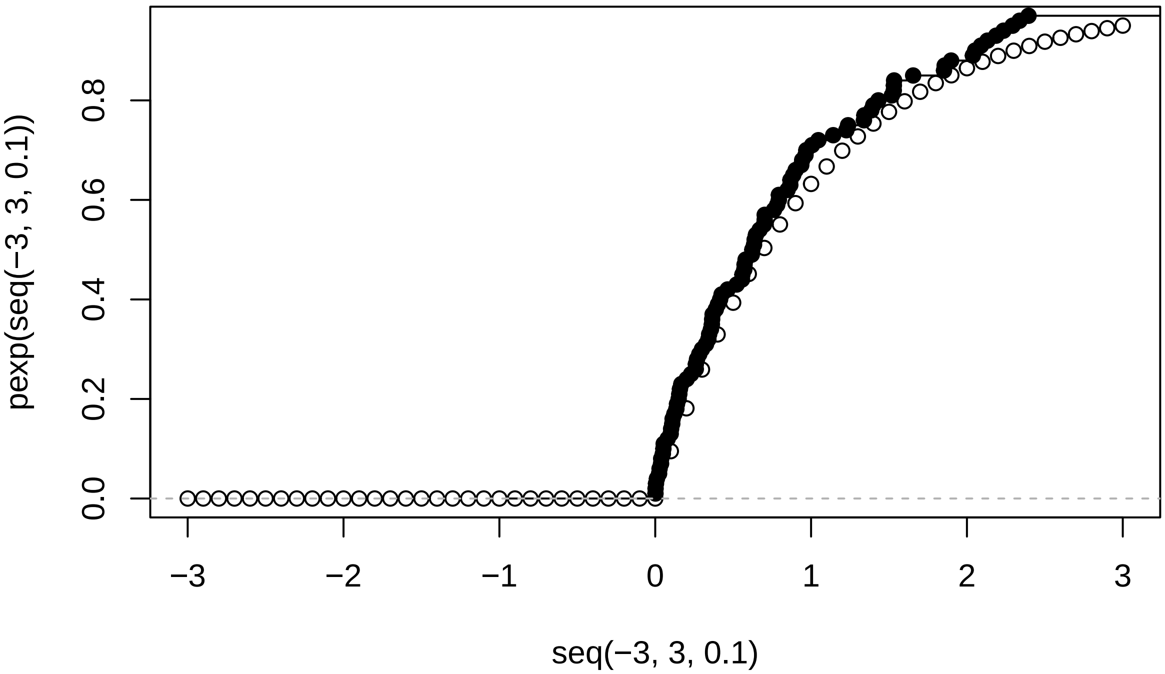
```
set.seed(131131)
```

```
# Zad 1

datan <- rnorm(100)
plot(seq(-3, 3, 0.1), pnorm(seq(-3, 3, 0.1)))
lines(ecdf(datan))
```



```
datan <- rexp(100)
plot(seq(-3, 3, 0.1), pexp(seq(-3, 3, 0.1)))
lines(ecdf(datan))
```



```
# Zad 2
M = 1000
n = 100
alpha <- 0.05
epsilon <- sqrt(log(2 / alpha) / (2 * n))
err <- 0
for (j in 1:M) {
    datae <- rexp(n)
    sam <- ecdf(datae)
    temp <- abs(sam(seq(0, 5, 0.05)) - pexp(seq(0, 5, 0.05)))
    for (i in 1:n) {
        if (temp[i] > epsilon) {
            err <- err + 1
            break
        }
    }
}
print(err)
```

```
## [1] 35
```

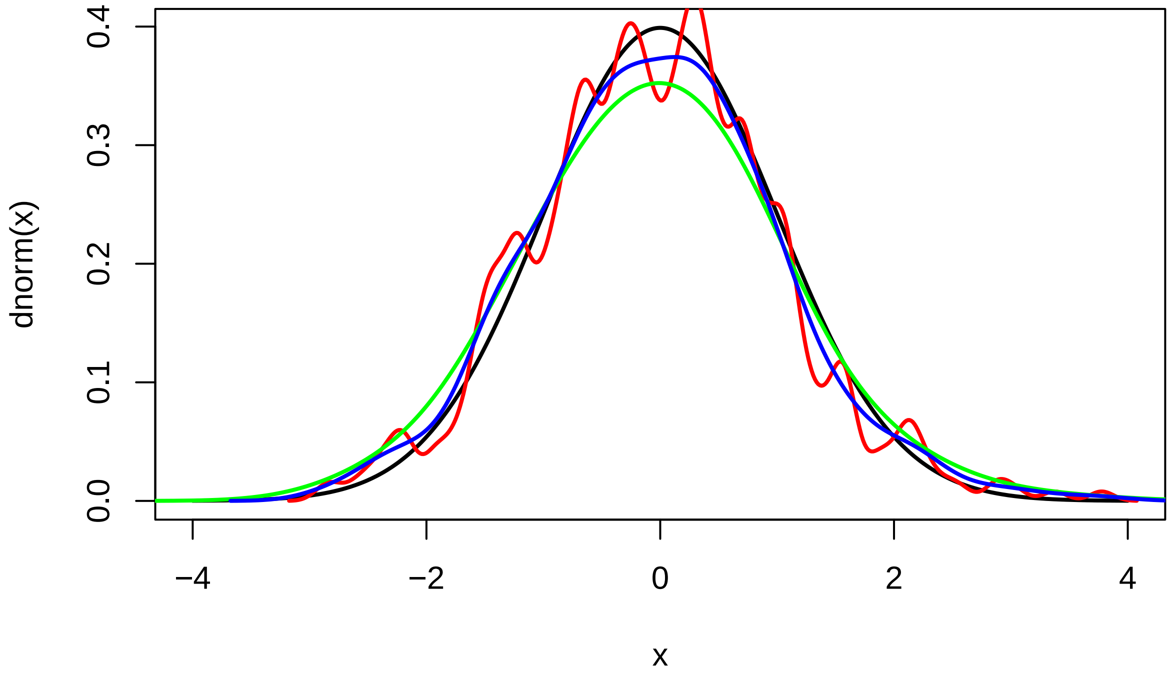
```
# Zad 3
n = 500
datan <- rnorm(n)

s <- sd(datan)
IQR <- IQR(datan)
h_silverman <- 0.9 * min(s, IQR / 1.34) * n ^ (-1 / 5)

bandwidths <- c(0.1, 0.5, h_silverman)

x <- seq(-4, 4, length = 500)
plot(x, dnorm(x), type = "l", lwd = 2)

colors <- rainbow(length(bandwidths))
for (i in 1:length(bandwidths)) {
    lines(density(datan, bw = bandwidths[i], kernel = "gaussian"), col = colors[i], lwd = 2)
}
```

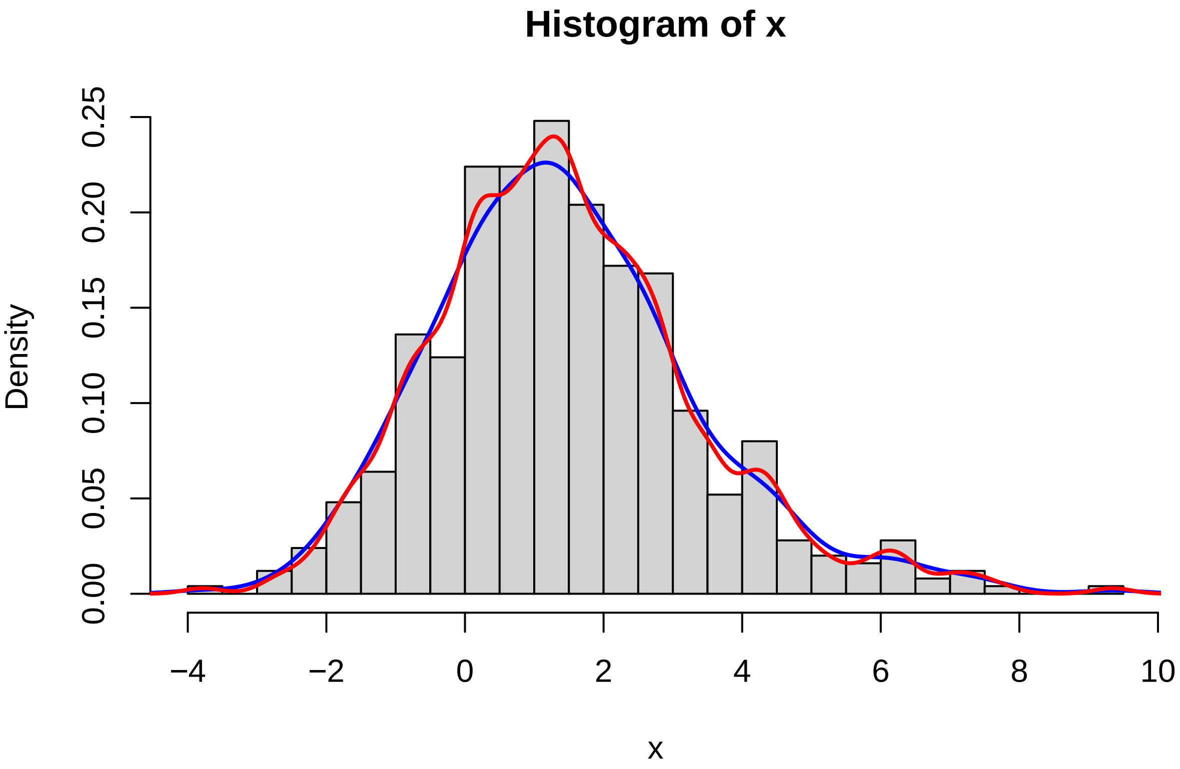


```
# Zad 4
n = 500
U <- runif(n, 0, 1)
x <- c()
for (i in 1:n) {
    if (U[i] <= 4 / 10) {
        x <- c(x, rnorm(1, 0, 1))
    }
    else if (U[i] <= 8 / 10) {
        x <- c(x, rnorm(1, 2, 1))
    }
    else {
        x <- c(x, rnorm(1, 4, 2))
    }
}

IQR <- IQR(x)
h <- 2 * IQR / (n ^ (1 / 3))
bins <- diff(range(x)) / h
r <- ceiling(bins)

hist(x, breaks = r, freq = FALSE)
lines(density(x), lwd = 2, col = "blue")

h_silverman <- 0.9 * min(s, IQR / 1.34) * n ^ (-1 / 5)
lines(density(x, bw = h_silverman), col = "red", lwd = 2)
```



Estymator utworzony za pomocą reguły kciuka Silverman wydaje się być “bliżej” faktycznej gęstości zmiennej losowej