

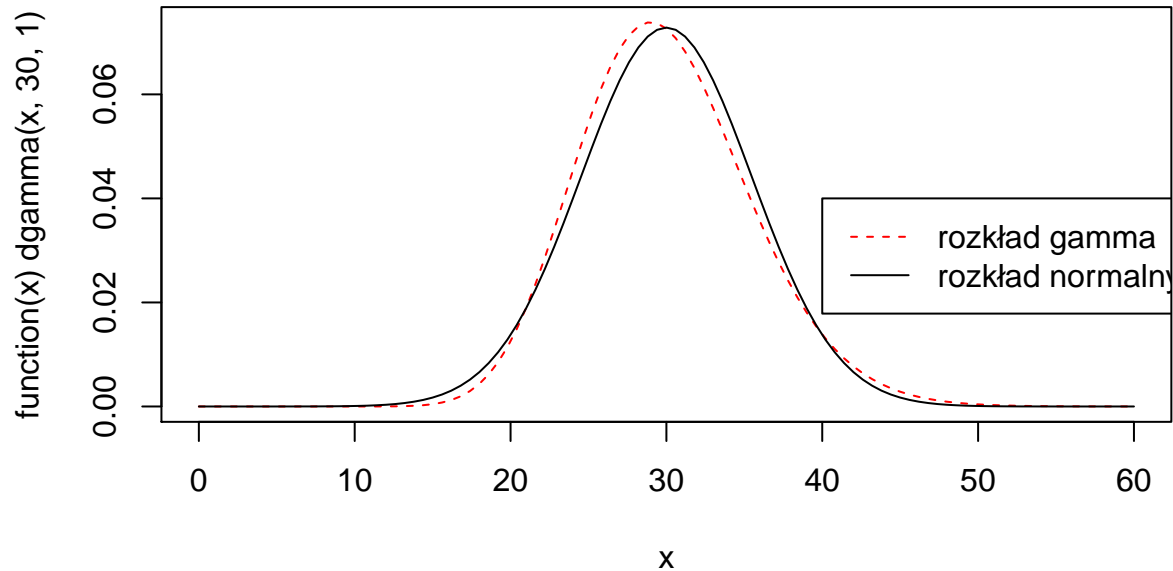
Lista 5

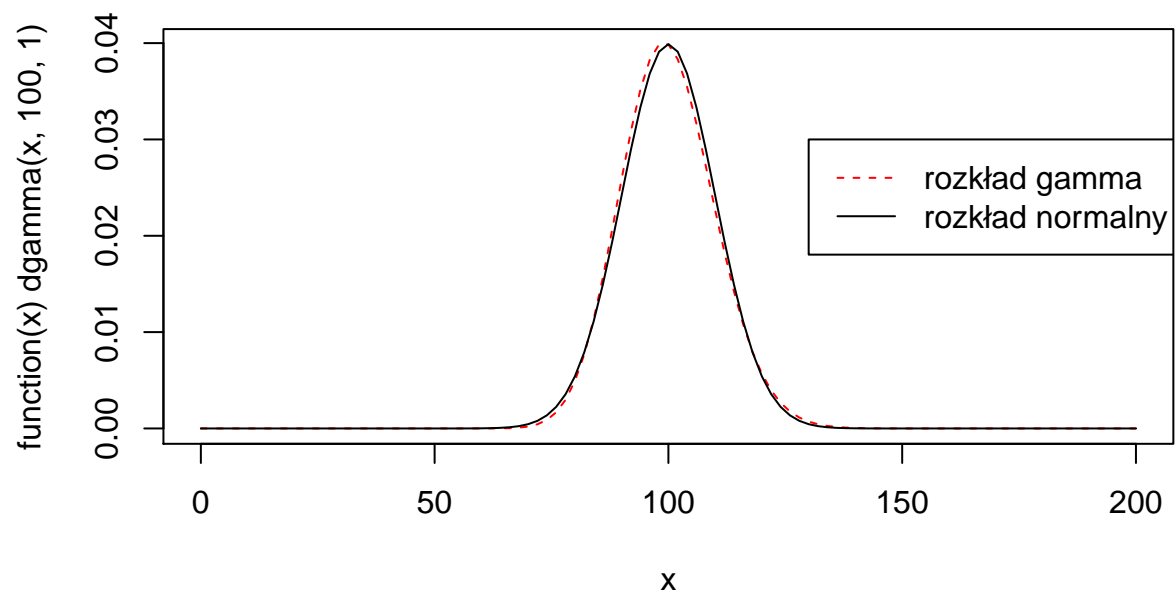
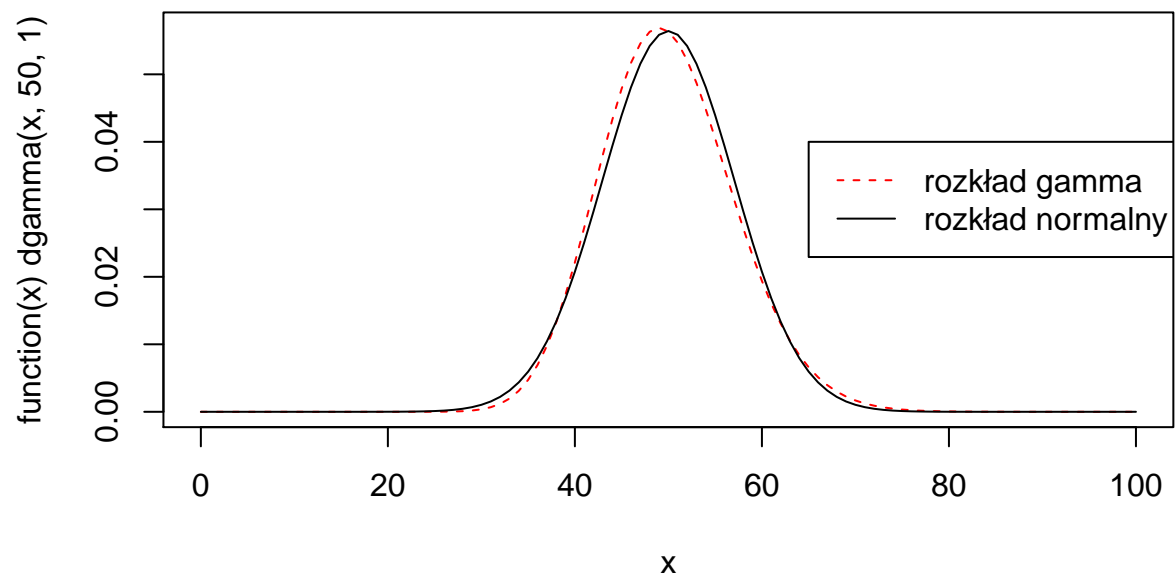
Mikołaj Słupiński

23 kwietnia 2018

Zadanie 1

```
plot(function(x) dgamma(x, 30, 1), 0, 60, col = "red", lty = 2)
plot(function(x) dnorm(x, 30/1, sqrt(30/1^2)), 0, 60, add=TRUE)
legend(40,y=0.04, legend = c("rozkład gamma", "rozkład normalny"),
      col = c("red", "black"), lty = c(2,1))
plot(function(x) dgamma(x, 50, 1), 0, 100, col = "red", lty = 2)
plot(function(x) dnorm(x, 50/1, sqrt(50/1^2)), 0, 100, add=TRUE)
legend(65,y=0.04, legend = c("rozkład gamma", "rozkład normalny"),
      col = c("red", "black"), lty = c(2,1))
plot(function(x) dgamma(x, 100, 1), 0, 200, col = "red", lty = 2)
plot(function(x) dnorm(x, 100/1, sqrt(100/1^2)), 0, 200, add=TRUE)
legend(130,y=0.03, legend = c("rozkład gamma", "rozkład normalny"),
      col = c("red", "black"), lty = c(2,1))
```



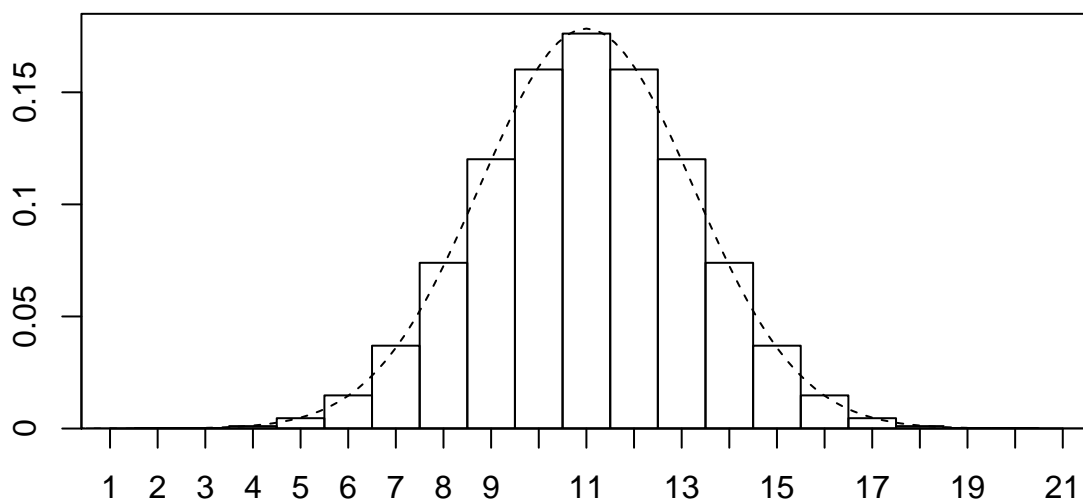


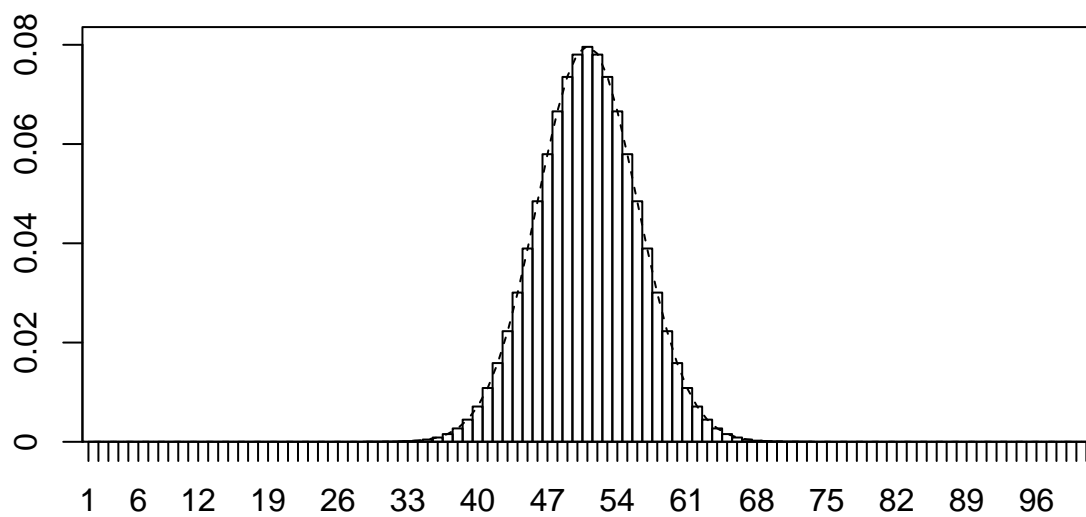
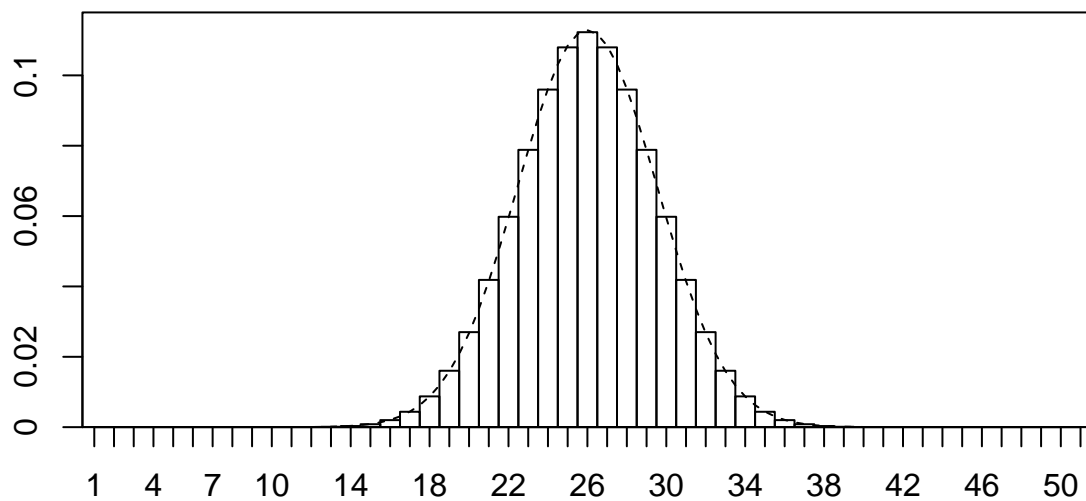
Widzimy, że wraz ze wzrostem liczby niezależnych zmiennych losowych o rozkładzie wykładniczym ich suma zbiega do rozkładu normalnego o odpowiednich parametrach.

Zadanie 2

a)

```
p = 0.5
n = 20
k = 0:n
library(plotrix)
barp(dbinom(k, n, p),0.5)
plot(function(x) dnorm(x-1.0, p*n, sqrt(p*(1-p)*n)), 0, n, add=TRUE, lty = 2)
n = 50
k = 0:n
barp(dbinom(k, n, p),0.5)
plot(function(x) dnorm(x-1.0, p*n, sqrt(p*(1-p)*n)), 0, n, add=TRUE, lty = 2)
n = 100
k = 0:n
barp(dbinom(k, n, p),0.5)
plot(function(x) dnorm(x-1.0, p*n, sqrt(p*(1-p)*n)), 0, n, add=TRUE, lty = 2)
```

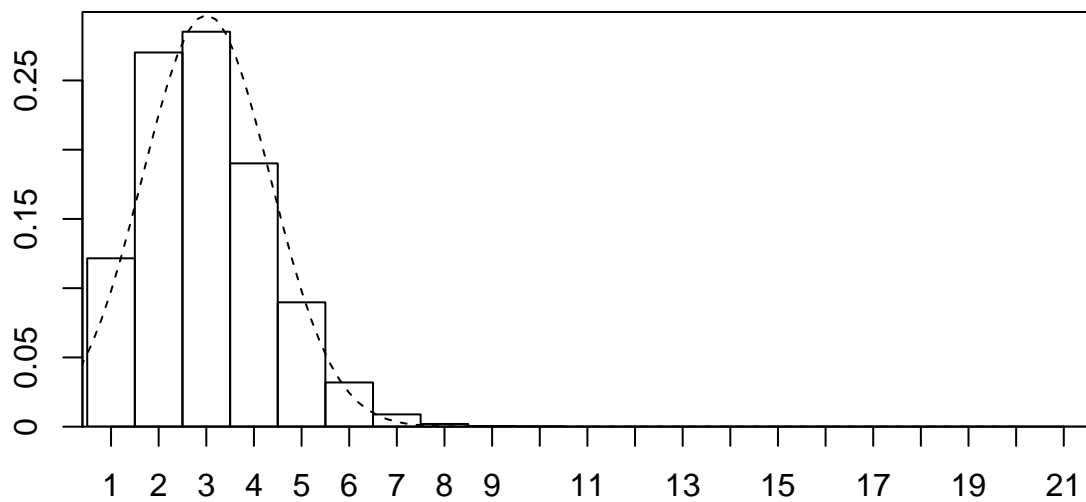


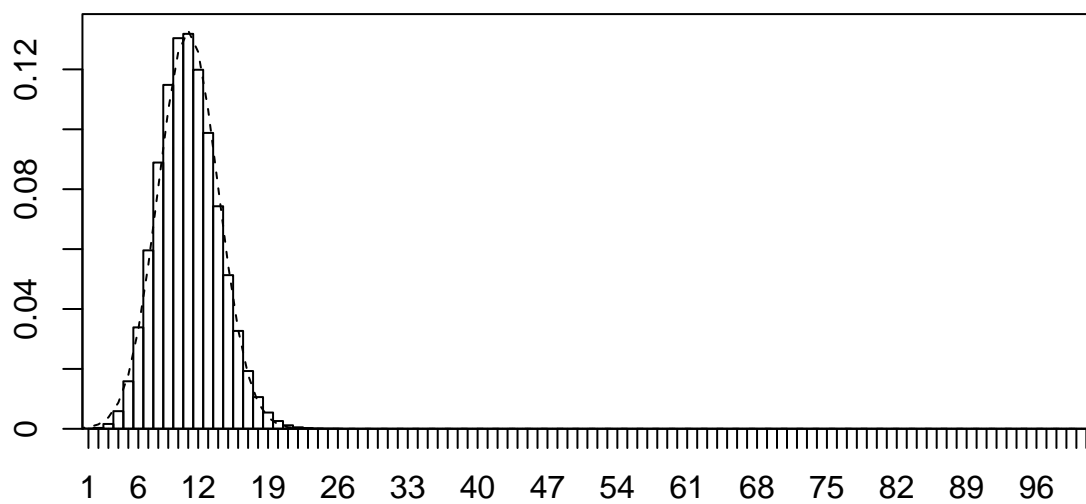
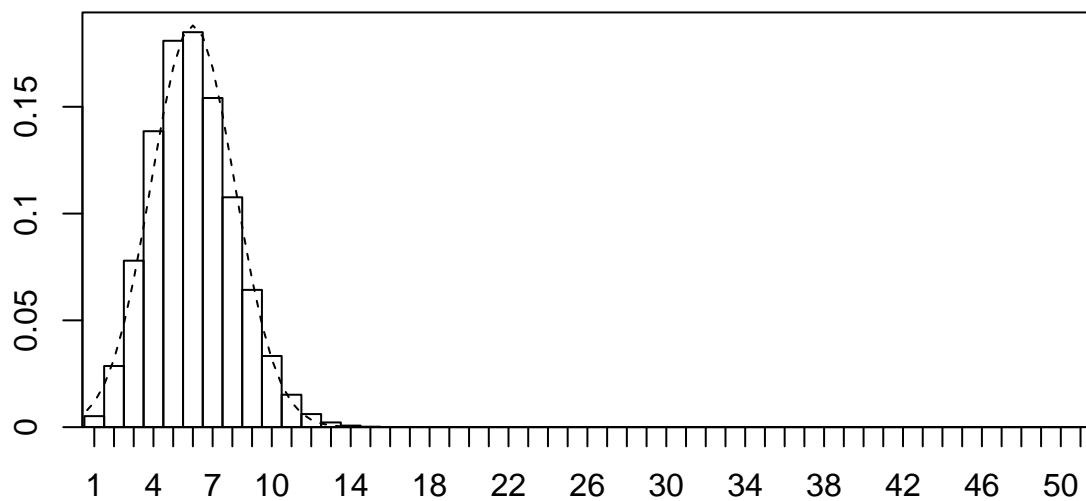


Widzimy, że dla $p = 0.5$ Rozkład normalny niemal idealnie aproksymuje rozkład dwumianowy.

b)

```
p = 0.1
n = 20
k = 0:n
barp(dbinom(k, n, p),.5)
plot(function(x) dnorm(x-1.0, p*n, sqrt(p*(1-p)*n)), 0, n, add=TRUE, lty = 2)
n = 50
k = 0:n
barp(dbinom(k, n, p),.5)
plot(function(x) dnorm(x-1.0, p*n, sqrt(p*(1-p)*n)), 0, n, add=TRUE, lty = 2)
n = 100
k = 0:n
barp(dbinom(k, n, p),.5)
plot(function(x) dnorm(x-1.0, p*n, sqrt(p*(1-p)*n)), 0, n, add=TRUE, lty = 2)
```





Widzimy, że dla $p \neq 0.5$ rozkład normalny jest w pewien sposób przesunięty od środków słupków histogramu, jednakże wraz ze wzrostem n przybliżenie staje się coraz dokładniejsze.

Zadanie 3

a)

```
n = 100
k = 1000
q.n = qt(0.975, n-1)/sqrt(n)
m = matrix(rnorm(n*k), k, n)
m1 = apply(m, 1, function(v) {
  c(mean(v) - q.n * sd(v) <= 0 & 0 <= mean(v) + q.n * sd(v), sd(v))
})
paste("Frakcja trafien w przedział:", mean(m1[1,]))
paste("Długość przedziału ufności: ", mean(m1[2,]) * 2 * q.n)

n = 200
k = 2000
q.n = qt(0.975, n-1)/sqrt(n)
m = matrix(rnorm(n*k), k, n)
m1 = apply(m, 1, function(v) {
  c(mean(v) - q.n * sd(v) <= 0 & 0 <= mean(v) + q.n * sd(v), sd(v))
})
paste("Frakcja trafien w przedział:", mean(m1[1,]))
paste("Długość przedziału ufności: ", mean(m1[2,]) * 2 * q.n)

## [1] "Frakcja trafien w przedział: 0.941"
## [1] "Długość przedziału ufności: 0.394635696968379"
## [1] "Frakcja trafien w przedział: 0.942"
## [1] "Długość przedziału ufności: 0.278876140205685"
```

Widzimy, że około 95% obserwacji wpada do przedziału ufności, co zgadza się z teoretycznym poziomem ufności 95%. Można również zaobserwować, że wraz ze wzrostem liczby obserwacji, przedział się zawęża.

b)

```
n = 100
k = 1000
q.n = qt(0.975, n-1)/sqrt(n)
m = matrix(rexp(n*k), k, n)
m1 = apply(m, 1, function(v) {
  c(mean(v) - q.n * sd(v) <= 1 & 1 <= mean(v) + q.n * sd(v), sd(v))
})
paste("Frakcja trafien w przedział:", mean(m1[1,]))
paste("Długość przedziału ufności: ", mean(m1[2,]) * 2 * q.n)

n = 200
k = 2000
q.n = qt(0.975, n-1)/sqrt(n)
m = matrix(rexp(n*k), k, n)
m1 = apply(m, 1, function(v) {
  c(mean(v) - q.n * sd(v) <= 1 & 1 <= mean(v) + q.n * sd(v), sd(v))
})
paste("Frakcja trafien w przedział:", mean(m1[1,]))
paste("Długość przedziału ufności: ", mean(m1[2,]) * 2 * q.n)

## [1] "Frakcja trafien w przedział: 0.942"
```

```
## [1] "Długość przedziału ufności: 0.393603160515626"
## [1] "Frakcja trafien w przedział: 0.94"
## [1] "Długość przedziału ufności: 0.277766685688118"
```

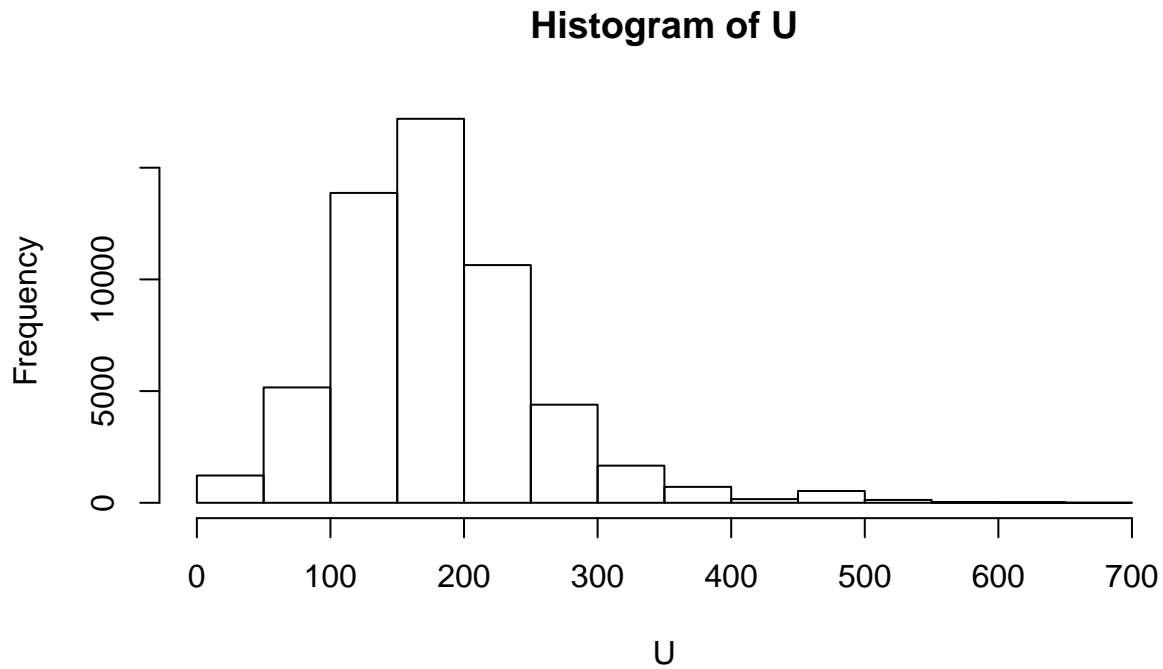
Ponownie można zaobserwować, że około 95% obserwacji wpada do przedziału ufności, co zgadza się z teoretycznym poziomem ufności 95%. Również w tym przypadku, wraz ze wzrostem liczby obserwacji, przedział się zawęża.

Zadanie 4

a)

```
tab = read.table("http://www.math.uni.wroc.pl/~mbogdan/Podstawy/Dane/individuals.dat")
U = sqrt(tab[[5]][tab[[5]] >= 0])
n = dim(tab)[1]
m.U = mean(U)
m.D = mean(tab[[5]])
p.W = table(tab[[3]])["3"]/n
p.S = table(tab[[3]])["6"]/n
hist(U)
paste("mu_U:", m.U)
paste("mu_D:", m.D)
paste("p_W:", p.W)
paste("p_S:", p.S)
```

```
## [1] "mu_U: 179.408704387844"
## [1] "mu_D: 37864.6094563409"
## [1] "p_W: 0.316714073597023"
## [1] "p_S: 0.100198572425267"
```

Otrzymaliśmy nową zmienną o jednomodalnym rozkładzie i średniej ok. 124.73.

b)

```
set.seed(213)
k = 200
M = sapply(1:k, function(i) sample(n,k))
z = qt(0.975, k-1)
M1 = apply(M, 2, function(v) {
  u = sqrt(tab[v,5][tab[v,5] >= 0])
  k1 = length(u)
  z1 = qt(0.975, k1-1)
  m.d = mean(tab[v,5])
  m.u = mean(u)
  p.w = table(tab[v,3])["3"]/k
  p.s = table(tab[v,3])["6"]/k
  i.u = m.u-sd(u)*z1/sqrt(k1) <= m.U & m.u+sd(u)*z1/sqrt(k1) >= m.U
  w = tab[v,5]
  i.d = m.d-sd(w)*z/sqrt(k) <= m.D & m.d+sd(w)*z/sqrt(k) >= m.D
  w = tab[v,3] == "3"
  i.s = p.s-sd(w)*z/sqrt(k) <= p.S & p.s+sd(w)*z/sqrt(k) >= p.S
  w = tab[v,3] == "6"
  i.w = p.w-sd(w)*z/sqrt(k) <= p.W & p.w+sd(w)*z/sqrt(k) >= p.W
  c(m.d, m.u, p.w, p.s, i.d, i.u, i.w, i.s)
})
paste("Estymator mu_D:", M1[1,1])
paste("Estymator mu_U:", M1[2,1])
paste("Estymator p_W:", M1[3,1])
```

```

paste("Estymator p_S:", M1[4,1])
paste("Czy przedział ufności zawiera mu_D:", M1[5,1])
paste("Czy przedział ufności zawiera mu_U:", M1[6,1])
paste("Czy przedział ufności zawiera p_W:", M1[7,1])
paste("Czy przedział ufności zawiera p_S:", M1[8,1])

```

```

## [1] "Estymator mu_D: 35516.78"
## [1] "Estymator mu_U: 174.946019345648"
## [1] "Estymator p_W: 0.325"
## [1] "Estymator p_S: 0.105"
## [1] "Czy przedział ufności zawiera mu_D: 1"
## [1] "Czy przedział ufności zawiera mu_U: 1"
## [1] "Czy przedział ufności zawiera p_W: 1"
## [1] "Czy przedział ufności zawiera p_S: 1"

```

W tym wypadku każdy przedział ufności zawierał estymowaną wartość.

c)

```

hist(M1[1,], main = expression('Histogram estymatora '*mu["D"]))
hist(M1[2,], main = expression('Histogram estymatora '*mu["U"]))
hist(M1[3,], main = expression('Histogram estymatora p'["W"]))
hist(M1[4,], main = expression('Histogram estymatora p'["S"]))
paste("Średnia trafień w przedział estymatora mu_D:", mean(M1[5,]))
paste("Średnia trafień w przedział estymatora mu_U:", mean(M1[6,]))
paste("Średnia trafień w przedział estymatora p_W:", mean(M1[7,]))
paste("Średnia trafień w przedział estymatora p_S:", mean(M1[8,]))

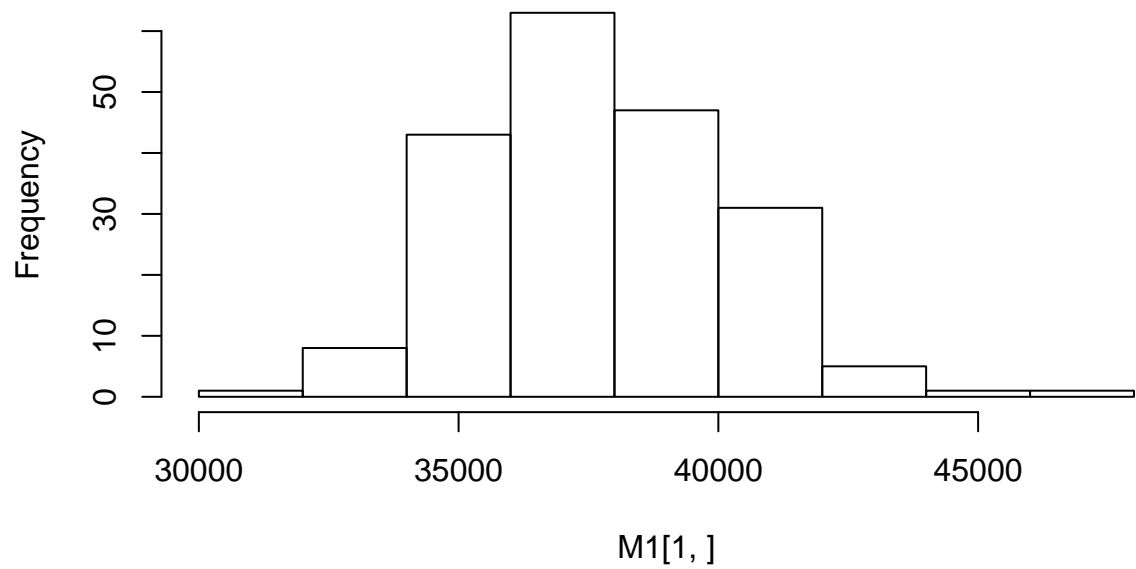
```

```

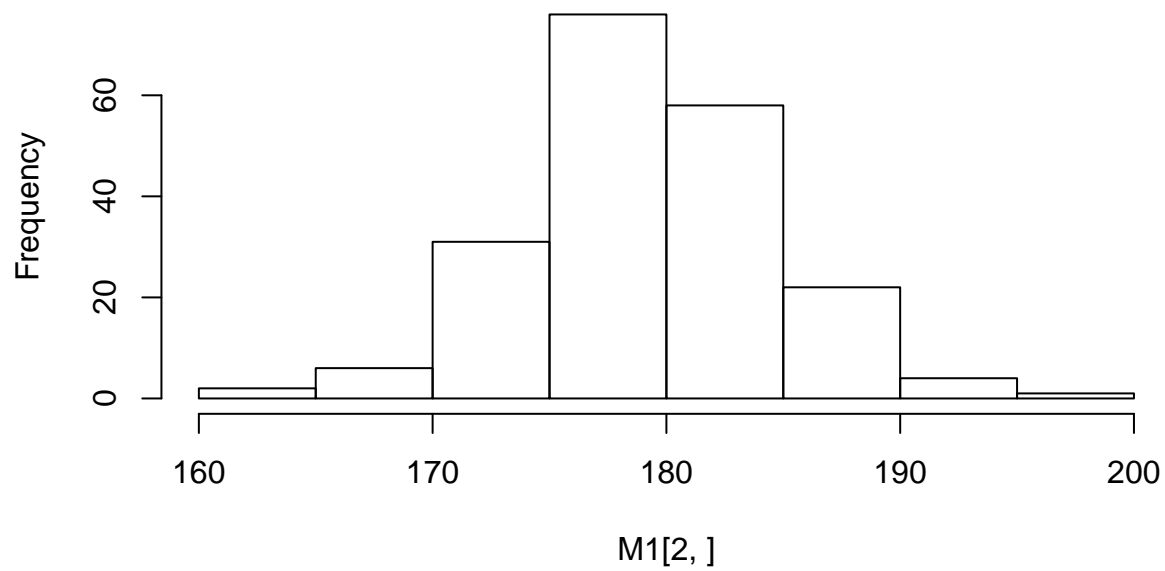
## [1] "Średnia trafień w przedział estymatora mu_D: 0.95"
## [1] "Średnia trafień w przedział estymatora mu_U: 0.945"
## [1] "Średnia trafień w przedział estymatora p_W: 0.8"
## [1] "Średnia trafień w przedział estymatora p_S: 1"

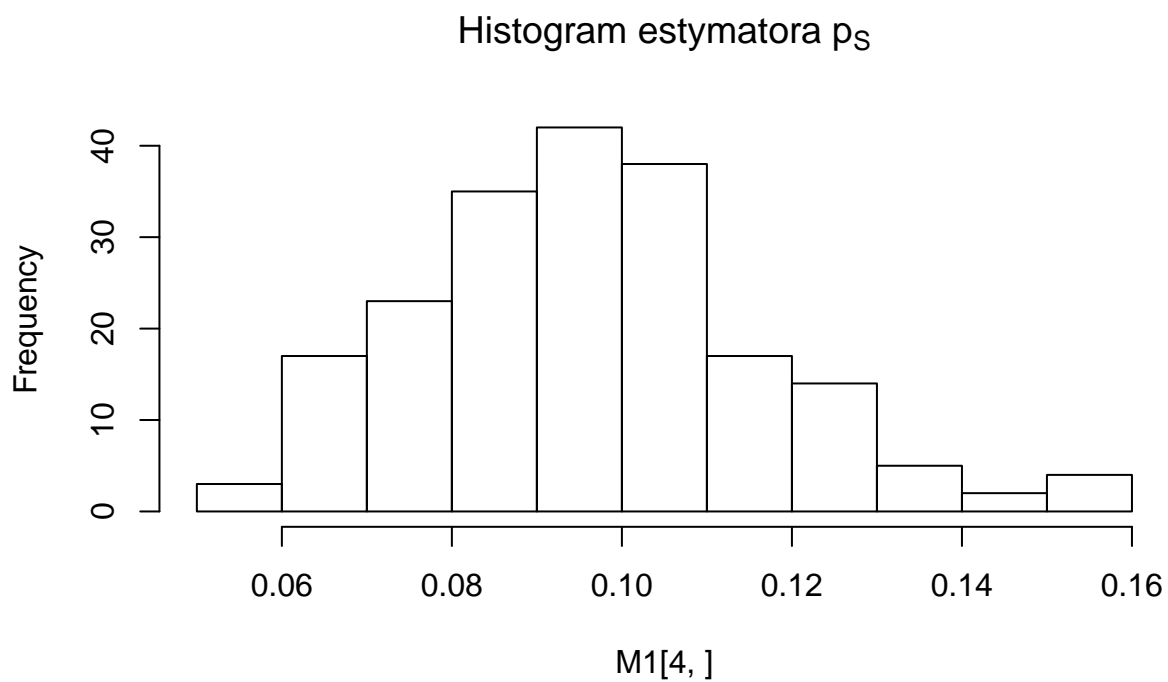
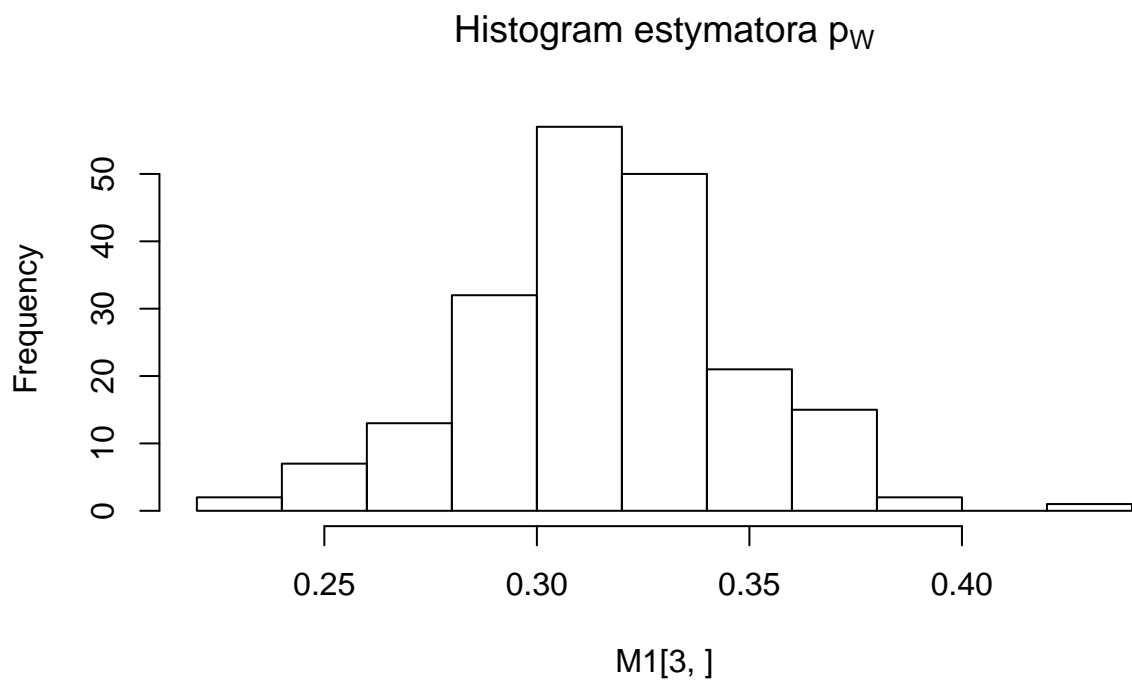
```

Histogram estymatora μ_D



Histogram estymatora μ_U





Widzimy, że wartość eksperymentalna estymacji średniej z zarobków oraz pierwiastka tej średniej pokrywa się z wartością teoretyczną. Około 95% uzyskanych przedziałów zawierało estymowaną wartość średniej. W przypadku p_S i p_W uzyskano inne wartości. Może to wynikać z tego, że rozkład tych zmiennych jest dwupunktowy.

Zadanie 5

```
tab = read.table("http://www.math.uni.wroc.pl/~mbogdan/Podstawy/Dane/table1_6.TXT")
#tab[[3]] iloraz inteligencji
#tab[[5]] samoocena

n = length(tab[[5]])
q.n = qt(0.975, n-1)/sqrt(n)
v = tab[[3]]
c(mean(v) - q.n * sd(v), mean(v) + q.n * sd(v))
v = tab[[5]]
c(mean(v) - q.n * sd(v), mean(v) + q.n * sd(v))
```

```
## [1] 105.9535 111.8927
```

```
## [1] 54.16301 59.76006
```

Przedział ufności dla ilorazu inteligencji to [105.9535, 111.8927], natomiast dla samooceny to [54.16301, 59.76006].