ADPS 2021Z — Laboratorium 1 (rozwiązania)

Ola Jaglinska

Zadanie 1 (1 pkt)

Treść zadania

Dla danych z ostatnich 12 miesięcy dotyczacych wybranych dwóch spółek giełdowych:

- sporządź wykresy procentowych zmian kursów zamknięcia w zależności od daty,
- wykreśl i porównaj histogramy procentowych zmian kursów zamknięcia,
- wykonaj jeden wspólny rysunek z wykresami pudełkowymi zmian kursów zamknięcia.

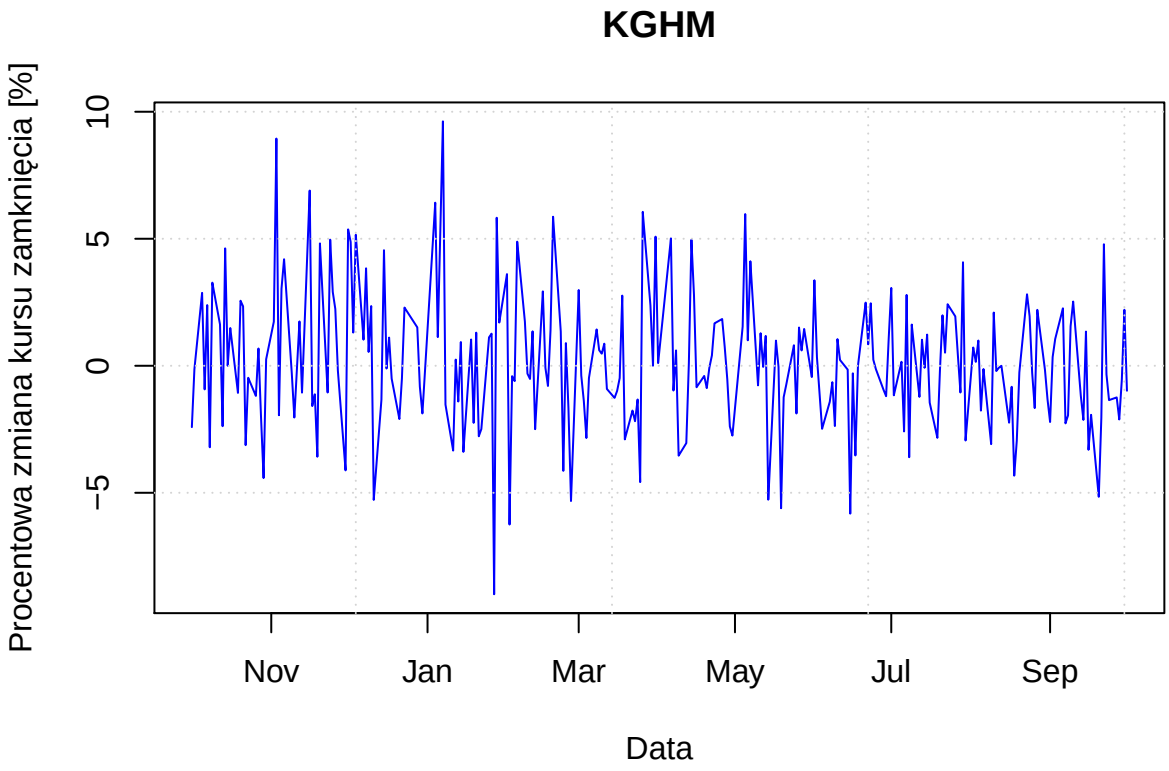
Rozwiązanie

```
unzip('mstall.zip', 'KGHM.mst')
unzip('mstall.zip', 'DATAWALK.mst')
df_KGHM = read.csv('KGHM.mst')
df_DATAWALK = read.csv('DATAWALK.mst')
names(df_KGHM) = c('ticker', 'date', 'open', 'high', 'low', 'close','vol')
names(df_DATAWALK) = c('ticker', 'date', 'open', 'high', 'low', 'close','vol')
df_KGHM$date = as.Date.character(df_KGHM$date, format ='%Y%m%d')
df_DATAWALK$date = as.Date.character(df_DATAWALK$date, format ='%Y%m%d')
```

Wykresy procentowych zmian kursu zamknięcia

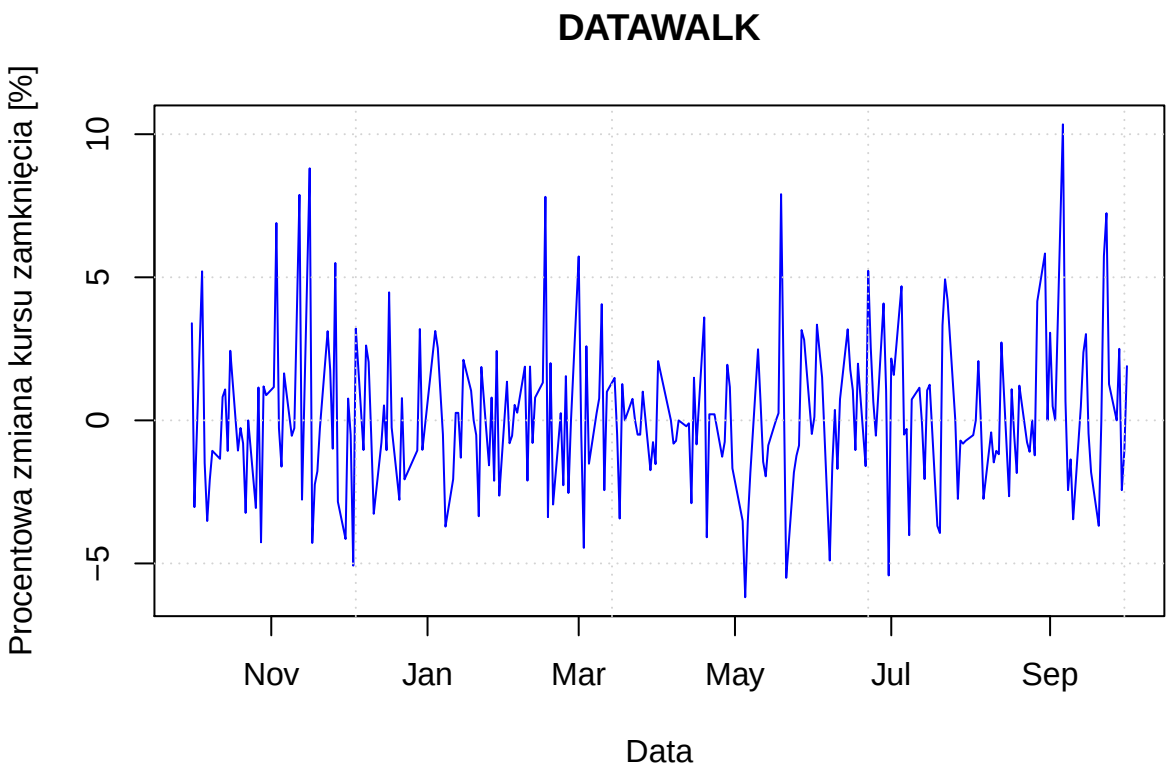
- KGHM:

```
df_KGHM$close_ch = with(df_KGHM, c(NA, 100*diff(close)/close[-length(close)]))
df_KGHM = df_KGHM[which(df_KGHM$date >= '2020-10-01' & df_KGHM$date <= '2021-10-01'),]
plot(close_ch ~ date, df_KGHM, type = 'l', col = 'blue', xlab = 'Data',
     ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'KGHM')
grid()
```



- DATAWALK:

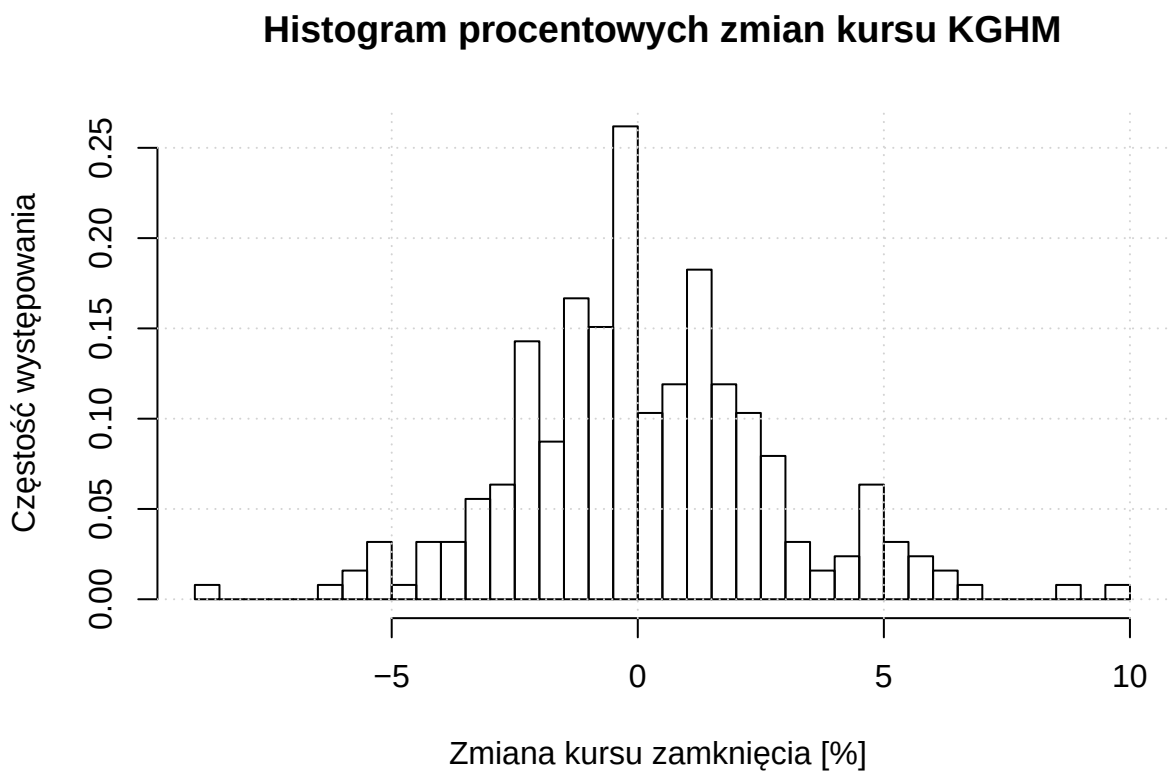
```
df_DATAWALK$close_ch = with(df_DATAWALK, c(NA, 100*diff(close)/close[-length(close)]))
df_DATAWALK = df_DATAWALK[which(df_DATAWALK$date >= '2020-10-01' & df_DATAWALK$date <= '2021-10-01'),]
plot(close_ch ~ date, df_DATAWALK, type = 'l', col = 'blue', xlab = 'Data',
     ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'DATAWALK')
grid()
```



Histogramy procentowych zmian kursu zamknięcia:

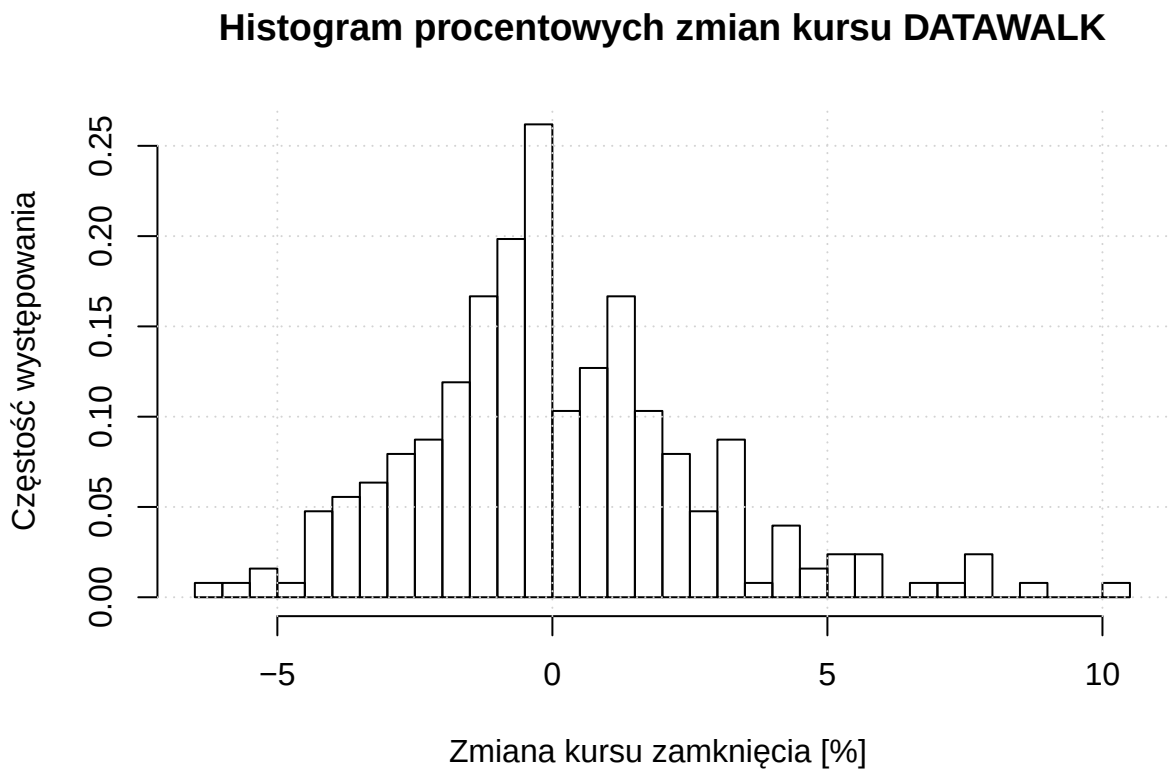
- KGHM

```
hist(df_KGHM$close_ch, breaks = 50, prob = T,
xlab = 'Zmiana kursu zamknięcia [%] ',
ylab = 'Częstość występowania',
main = paste('Histogram procentowych zmian kursu', 'KGHM')  )
grid()
```



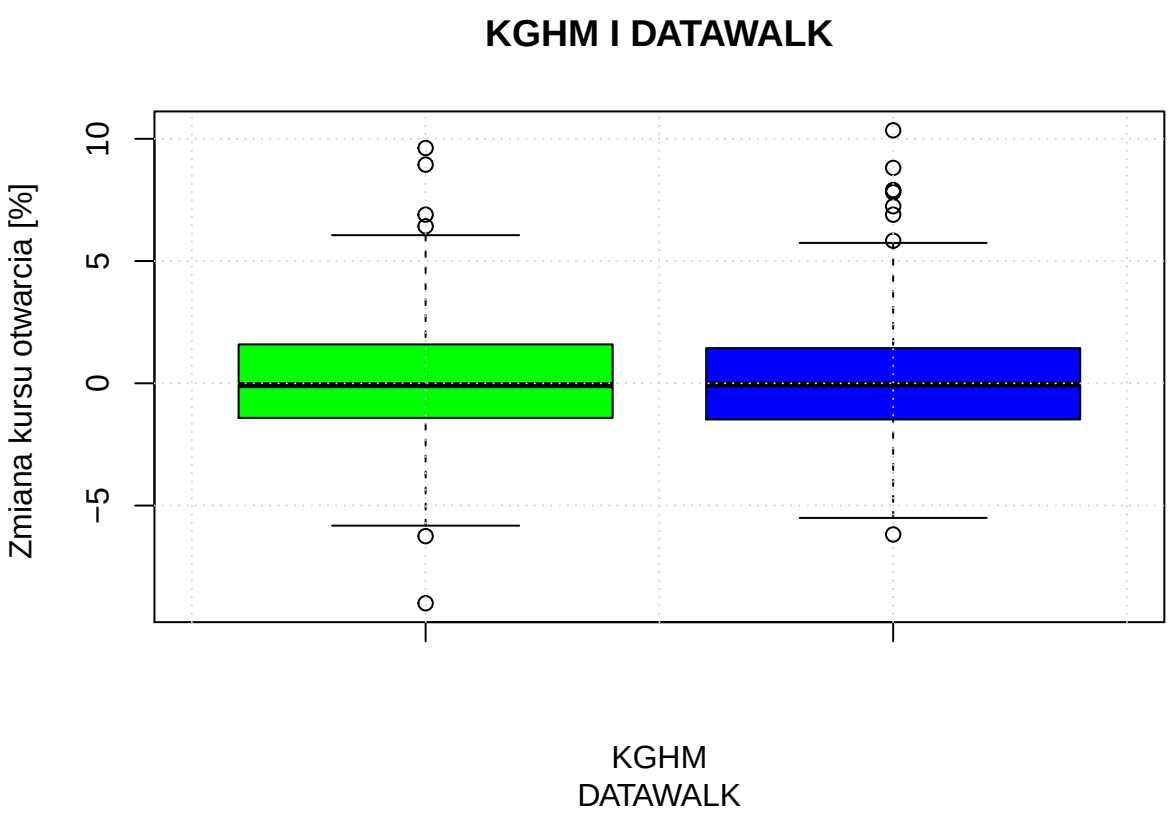
- DATAWALK

```
hist(df_DATAWALK$close_ch, breaks = 50, prob = T,
xlab = 'Zmiana kursu zamknięcia [%] ',
ylab = 'Częstość występowania',
main = paste('Histogram procentowych zmian kursu', 'DATAWALK')  )
grid()
```



Porównanie zmian kursu zamknięcia dla spółek KGHM i DataWalk

```
boxplot(df_KGHM$close_ch,df_DATAWALK$close_ch, col = c('green', 'blue'),
xlab = c('KGHM', 'DATAWALK'), ylab = 'Zmiana kursu otwarcia [%] ',
main = 'KGHM I DATAWALK')
grid()
```



Zadanie 2 (1 pkt)

Treść zadania

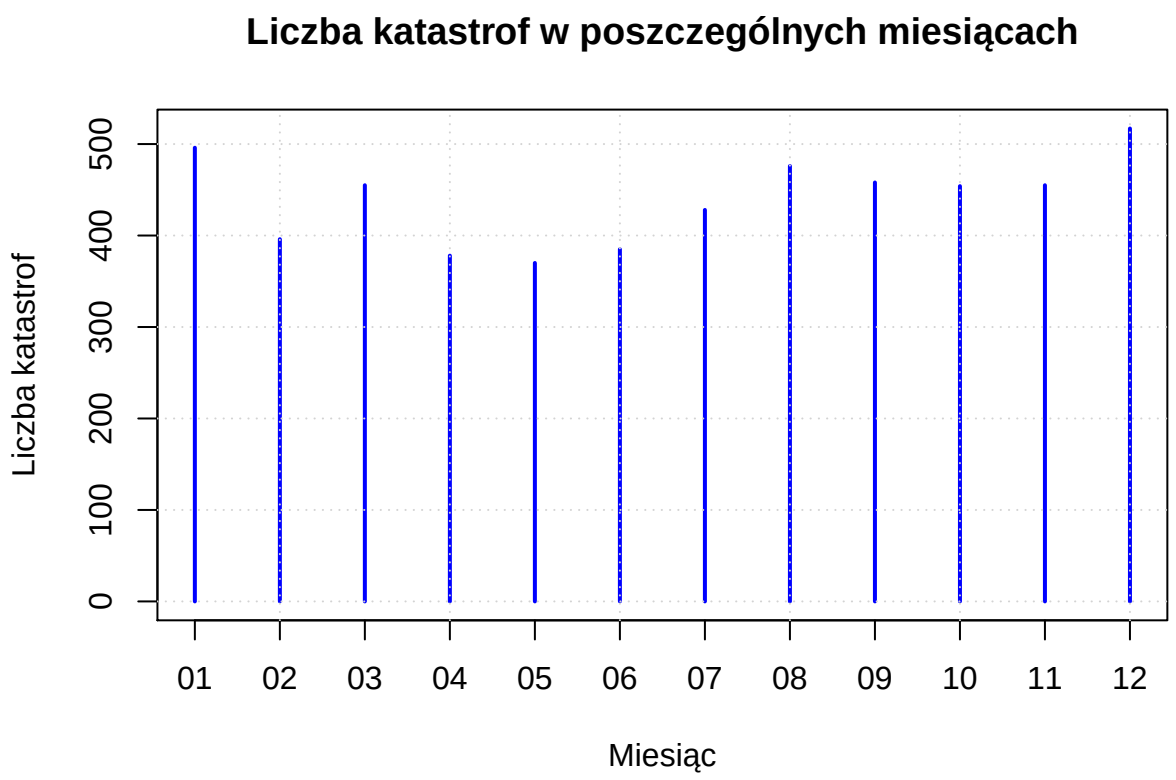
1. Sporządź wykres liczby katastrof lotniczych w poszczególnych:
- miesiącach,
- dniach,
- dniach tygodnia (weekdays()).
2. Narysuj jak w kolejnych latach zmieniały się:
- liczba osób, które przeżyły katastrofy,
- odsetek osób (w procentach), które przeżyły katastrofy.

Rozwiązanie

```
kat = read.csv('crashes.csv')
```

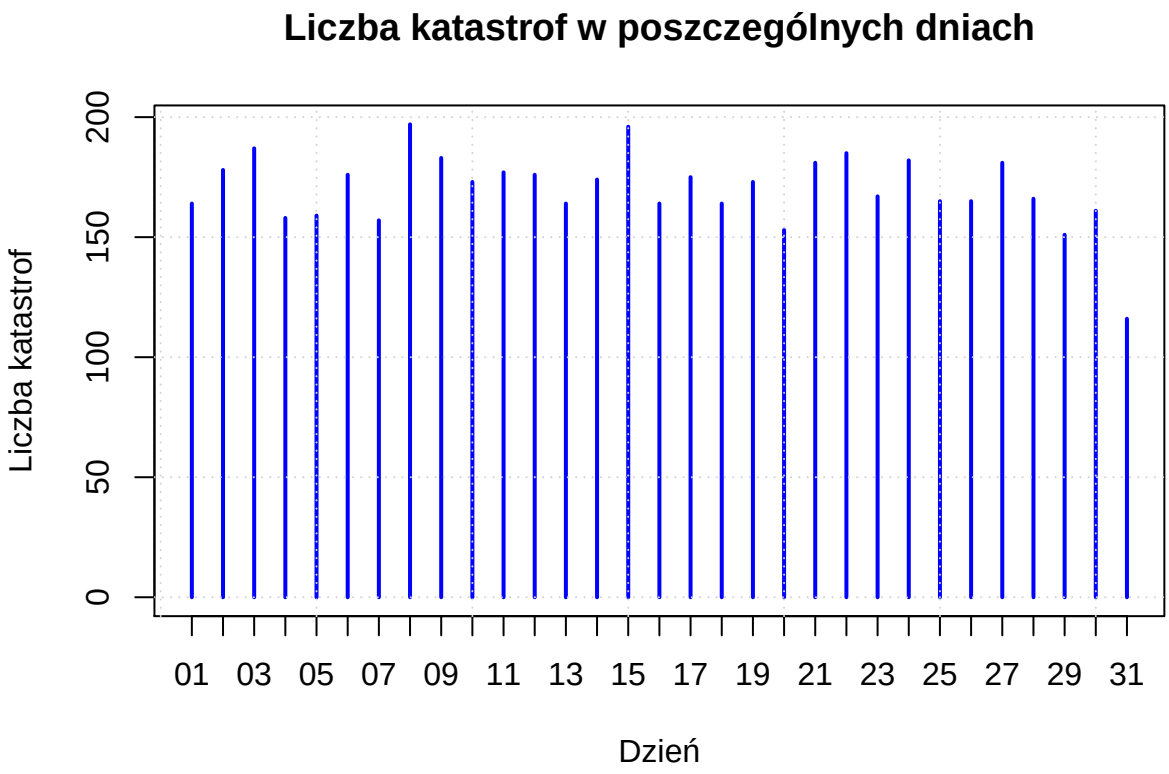
- Wykres liczby katastrof lotniczych w miesiącach:

```
kat$Month = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%m')
plot(table(kat$Month), type = 'h', col = 'blue', xlab = 'Miesiąc',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych miesiącach' )
grid()
```



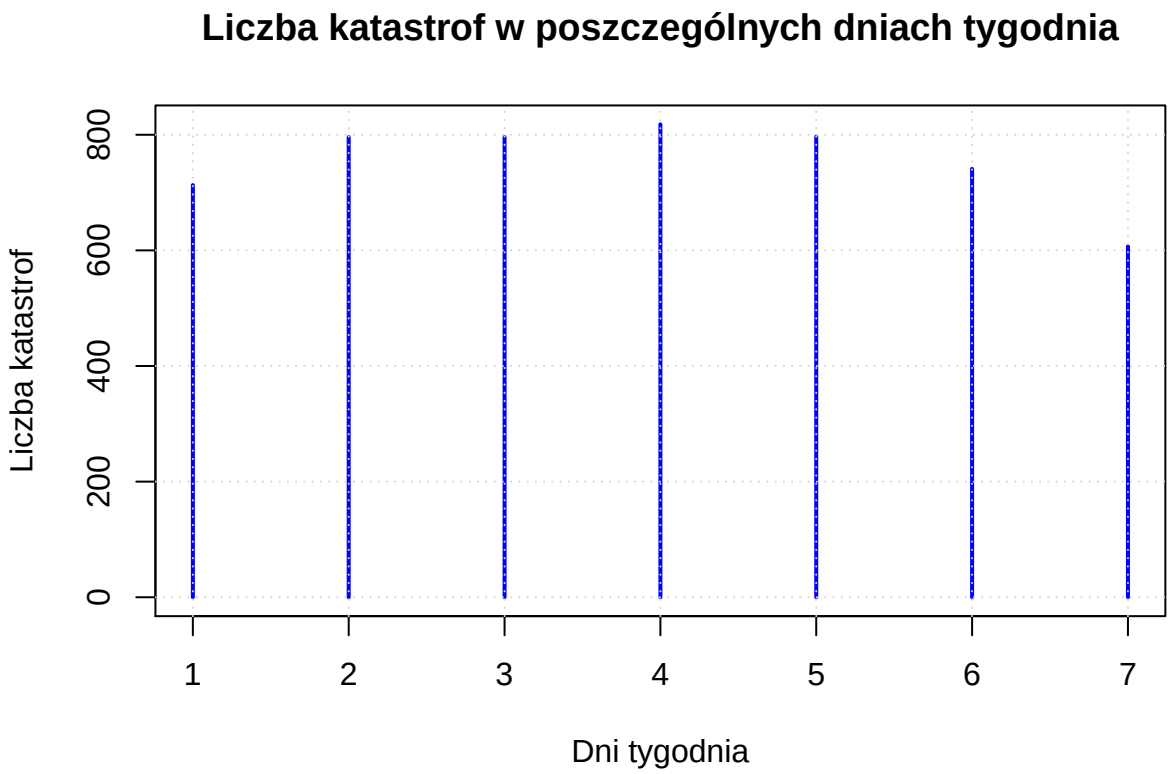
- Wykres liczby katastrof lotniczych w dniach:

```
kat$Day = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%d')
plot(table(kat$Day), type = 'h', col = 'blue', xlab = 'Dzień',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych dniach' )
grid()
```



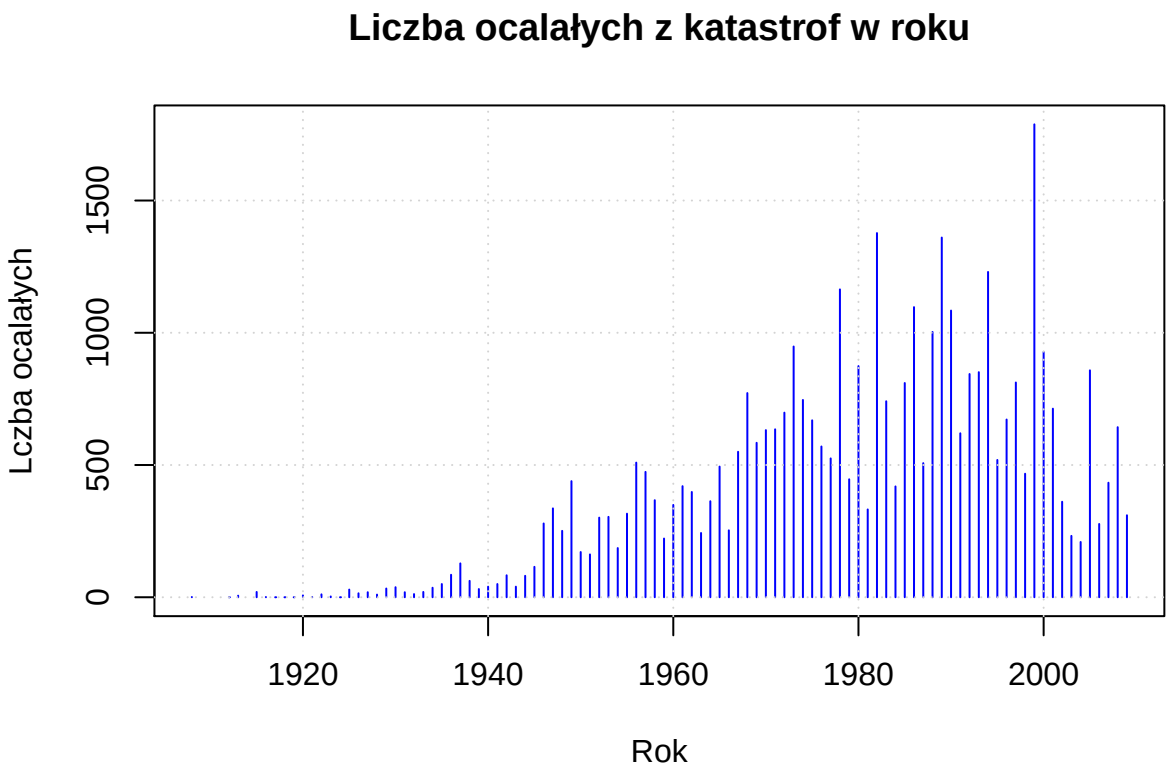
- Wykres liczby katastrof lotniczych w dniach tygodnia:

```
kat$Weekdays = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%u')
plot(table(kat$Weekdays), type = 'h', col = 'blue', xlab = 'Dni tygodnia',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych dniach tygodnia')
grid()
```



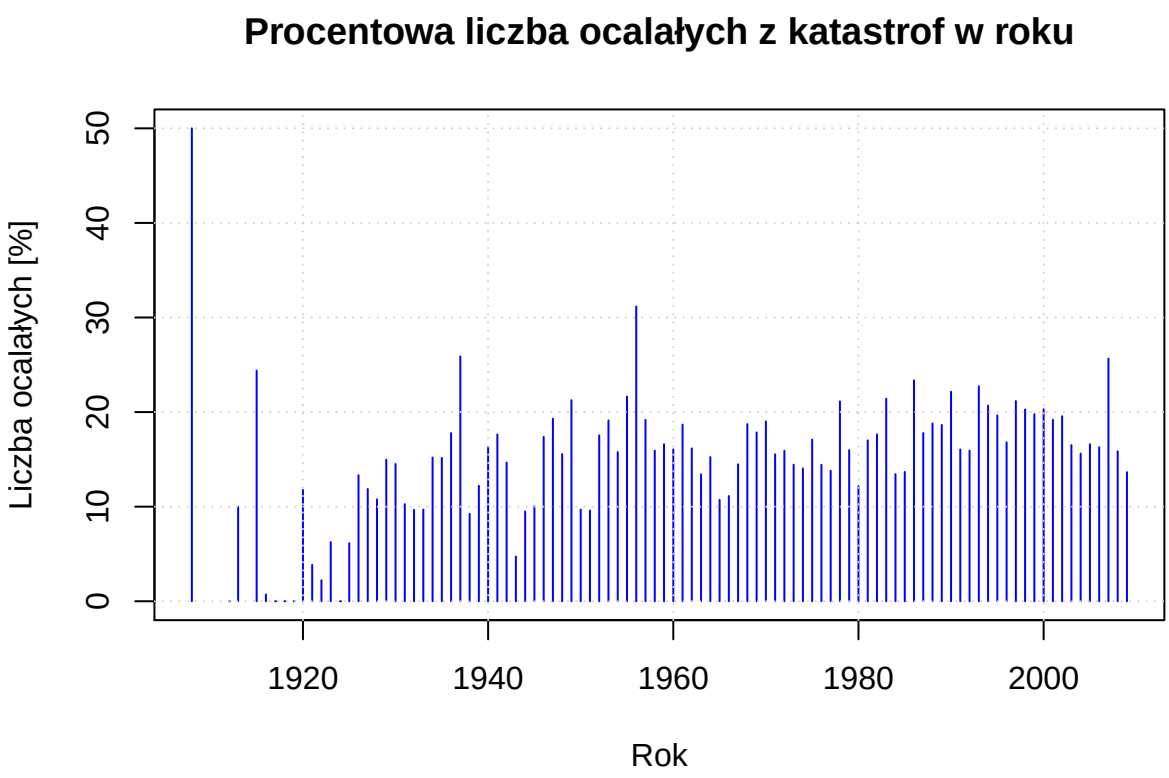
- Liczba osób, które przeżyły katastrofę w poszczególnych latach

```
kat$Year = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%Y')
Ocalali_agr = aggregate((Aboard - Fatalities) ~ Year, kat, FUN = sum)
plot(Ocalali_agr, type = 'h', col = 'blue',xlab = 'Rok',
     ylab = 'Lczba ocalałych', main = 'Liczba ocalałych z katastrof w roku' )
grid()
```



- Odsetek osób, które przeżyły katastrofę

```
Ocalali_agr = aggregate(100*((Aboard - Fatalities) / Aboard) ~ Year, kat, FUN = mean)
plot(Ocalali_agr, type = 'h', col = 'blue',xlab = 'Rok',
     ylab = 'Liczba ocalałych [%]', main = 'Procentowa liczba ocalałych z katastrof w roku' )
grid()
```



Zadanie 3 (1 pkt)

Treść zadania

1. Dla dwóch różnych zestawów parametrów rozkładu dwumianowego (rbinom):

- Binom(20,0.2)
- Binom(20,0.8)

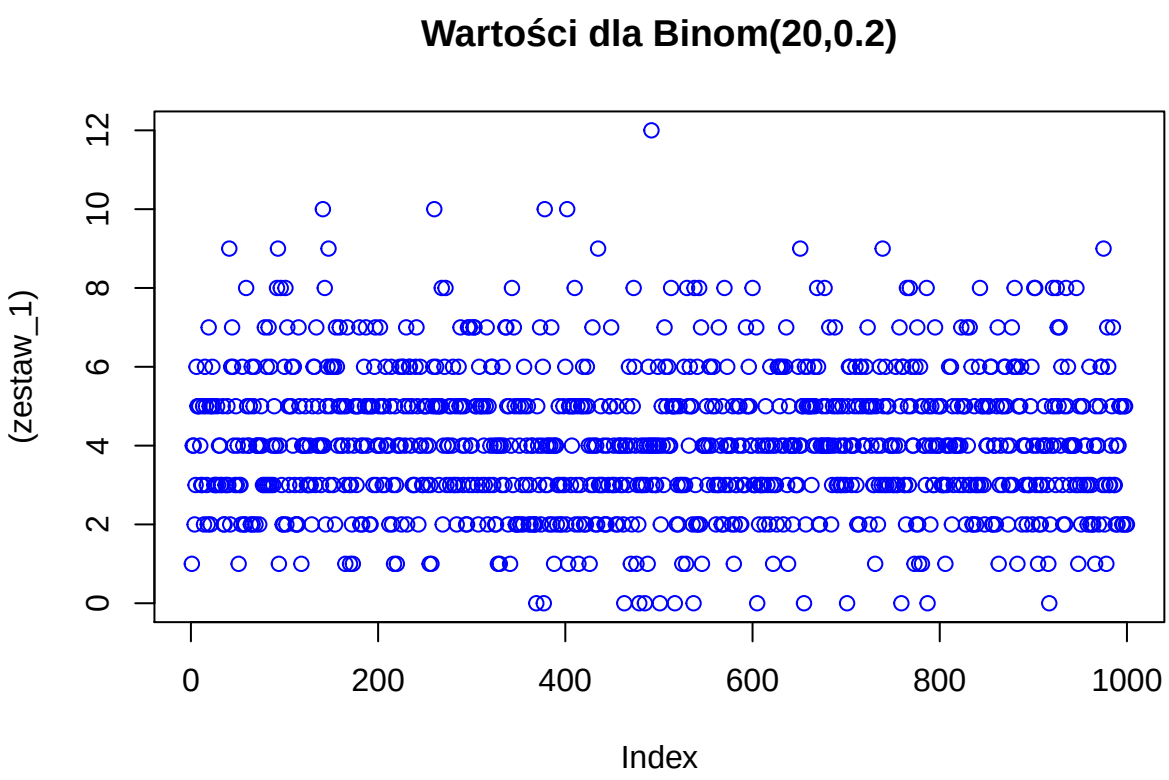
wygeneruj próby losowe składające się z M = 1000 próbek i narysuj wartości wygenerowanych danych.

2. Dla obu rozkładów narysuj na jednym rysunku empiryczne i teoretyczne (użyj funkcji dbinom) funkcje prawdopodobieństwa, a na drugim rysunku empiryczne i teoretyczne (użyj funkcji pbinom) dystrybuanty. W obu przypadkach wyskaluj oś odciętych od 0 do 20.

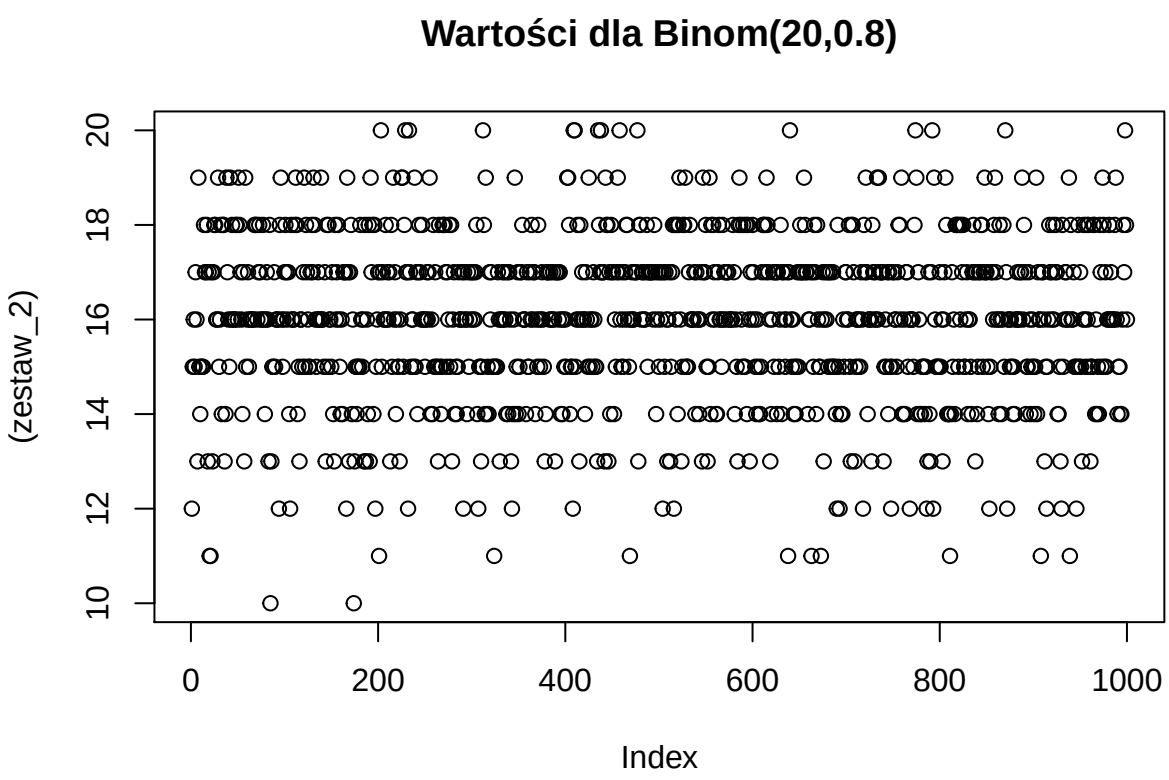
Rozwiązanie

- Próby losowe

```
M = 1000
zestaw_1 = rbinom(M, 20, 0.2)
plot((zestaw_1), col = 'blue', main = 'Wartości dla Binom(20,0.2)')
```




```
zestaw_2 = rbinom(M, 20, 0.8)
plot((zestaw_2), col = 'black', main = 'Wartości dla Binom(20,0.8)')
```



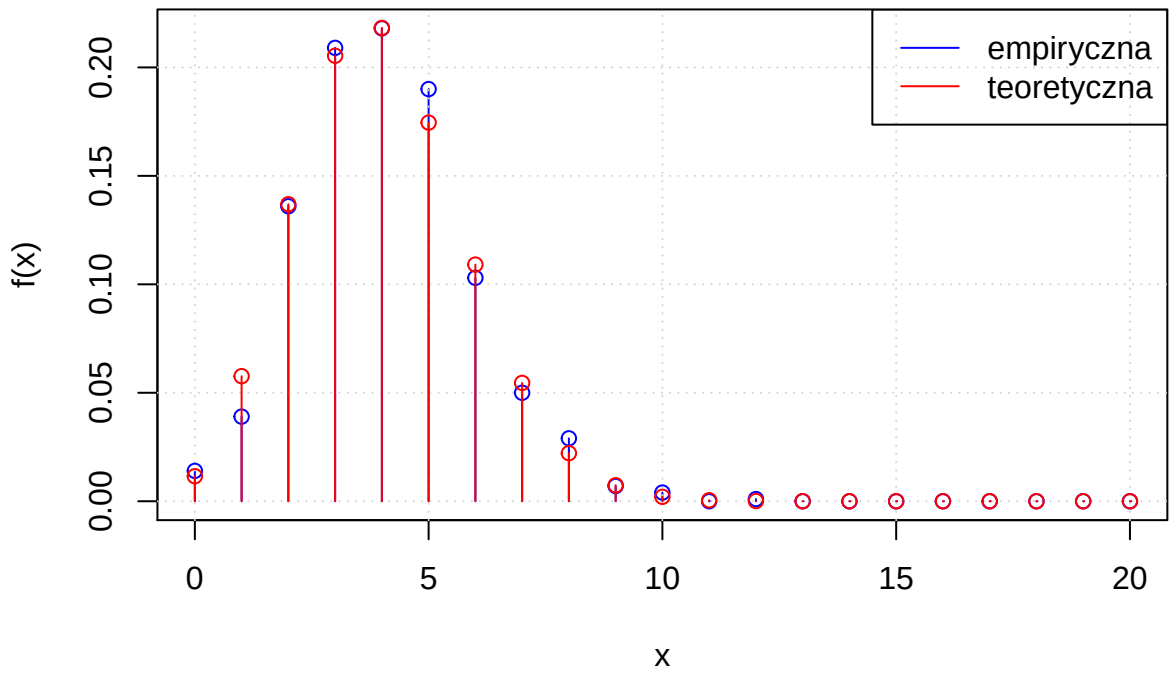
- Wartości parametrów z próby:

```
m = mean(zestaw_1); v = var(zestaw_1)
```

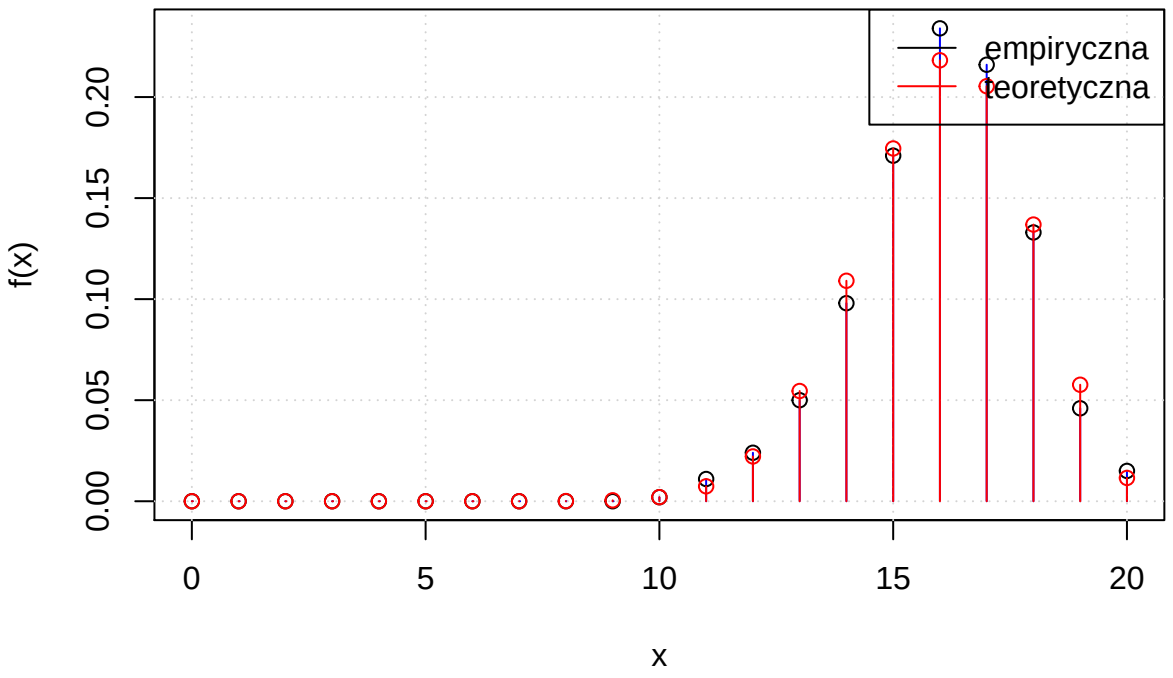
```
m = mean(zestaw_2); v = var(zestaw_2)
```

- empiryczne i teoretyczne funkcje prawdopodobieństwa

```
Arg = 0:20
Freq1 = as.numeric(table(factor(zestaw_1, levels = Arg))) / M
plot(Freq1 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
    main = paste0('Funkcja prawdopodobieństwa dla M = dla ', M, ' dla Binom(20,0.2)'))
grid()
points(Freq1 ~ Arg, col = 'blue')
lines(dbinom(Arg, 20, prob = 0.2) ~ Arg, type = 'h', col = 'red',
      xlab = 'x', ylab = 'f(x)')
points(dbinom(Arg, 20, prob = 0.2) ~ Arg, col = 'red')

legend('topright', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```

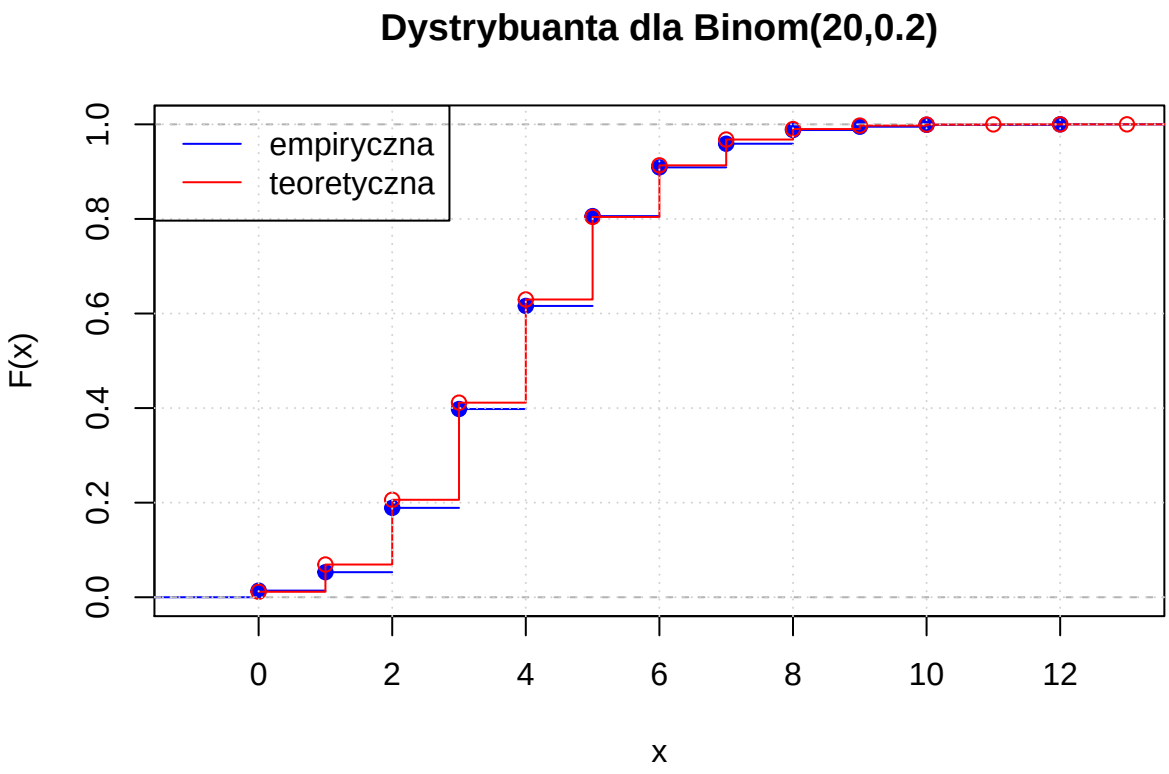



```
Arg = 0:20
Freq2 = as.numeric(table(factor(zestaw_2, levels = Arg))) / M
plot(Freq2 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
    main = paste0('Funkcja prawdopodobieństwa dla M = dla ', M, ' dla Binom(20,0.8)'))
grid()
points(Freq2 ~ Arg, col = 'black')
lines(dbinom(Arg, 20, prob = 0.8) ~ Arg, type = 'h', col = 'red',
      xlab = 'x', ylab = 'f(x)')
points(dbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

legend('topright', c('empiryczna', 'teoretyczna'),
    col = c('black', 'red'), lwd = 1)
```

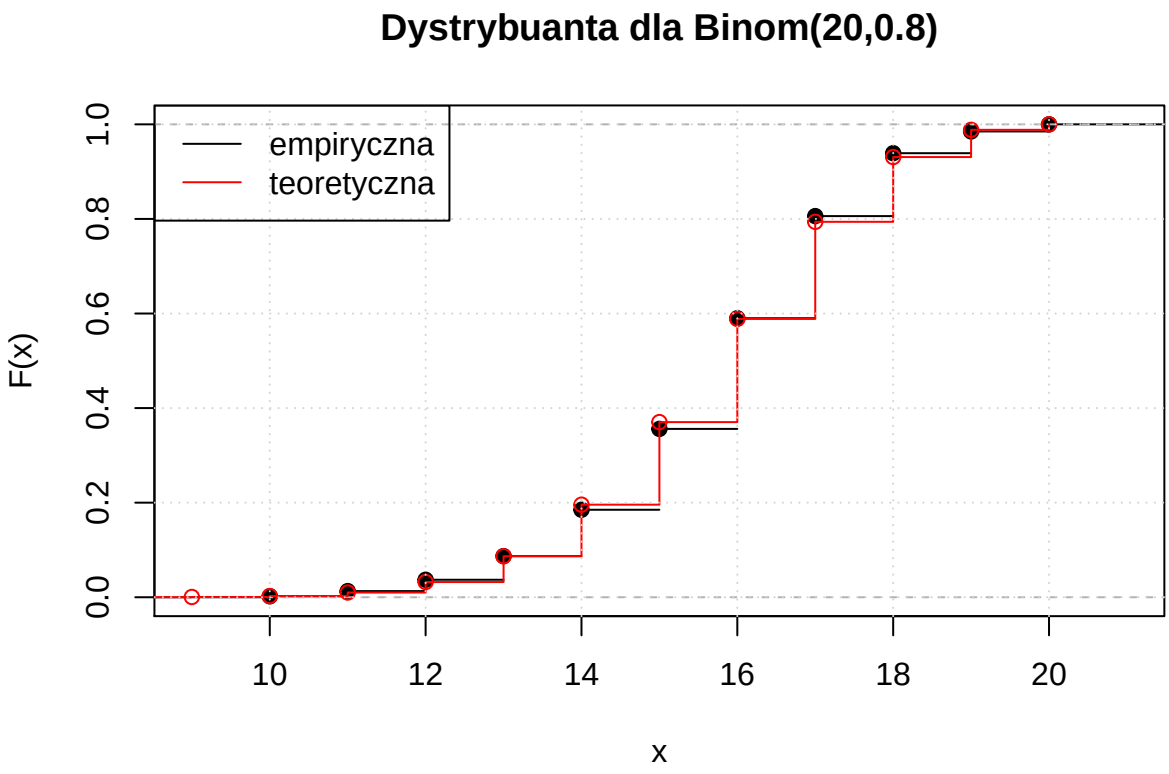


- empiryczne i teoretyczne dystrybuanty

```
Arg = 0:20
plot(ecdf(zestaw_1), col = 'blue', xlab = 'x', ylab = 'F(x)', main = 'Dystrybuanta dla Binom(20,0.2)')
lines(pbinom(Arg, 20, prob = 0.2)~ Arg, type = 's', col = 'red',
      xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg, 20, prob = 0.2) ~ Arg, col = 'red')

grid()
legend('topleft', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```




```
Arg = 0:20
plot(ecdf(zestaw_2), col = 'black', xlab = 'x', ylab = 'F(x)', main = 'Dystrybuanta dla Binom(20,0.8)')
lines(pbinom(Arg, 20, prob = 0.8) ~ Arg, type = 's', col = 'red',
      xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

grid()
legend('topleft', c('empiryczna', 'teoretyczna'),
    col = c('black', 'red'), lwd = 1)
```



Zadanie 4 (1 pkt)

Treść zadania

1. Dla rozkładu dwumianowego Binom(20, 0.8) wygeneruj trzy próby losowe składające się z M = 100, 1000 i 10000 próbek.
2. Dla poszczególnych prób wykreśl empiryczne i teoretyczne funkcje prawdopodobieństwa, a także empiryczne i teoretyczne dystrybuanty.
3. We wszystkich przypadkach oblicz empiryczne wartości średnie i wariancje. Porównaj je ze sobą oraz z wartościami teoretycznymi dla rozkładu Binom(20, 0.8).

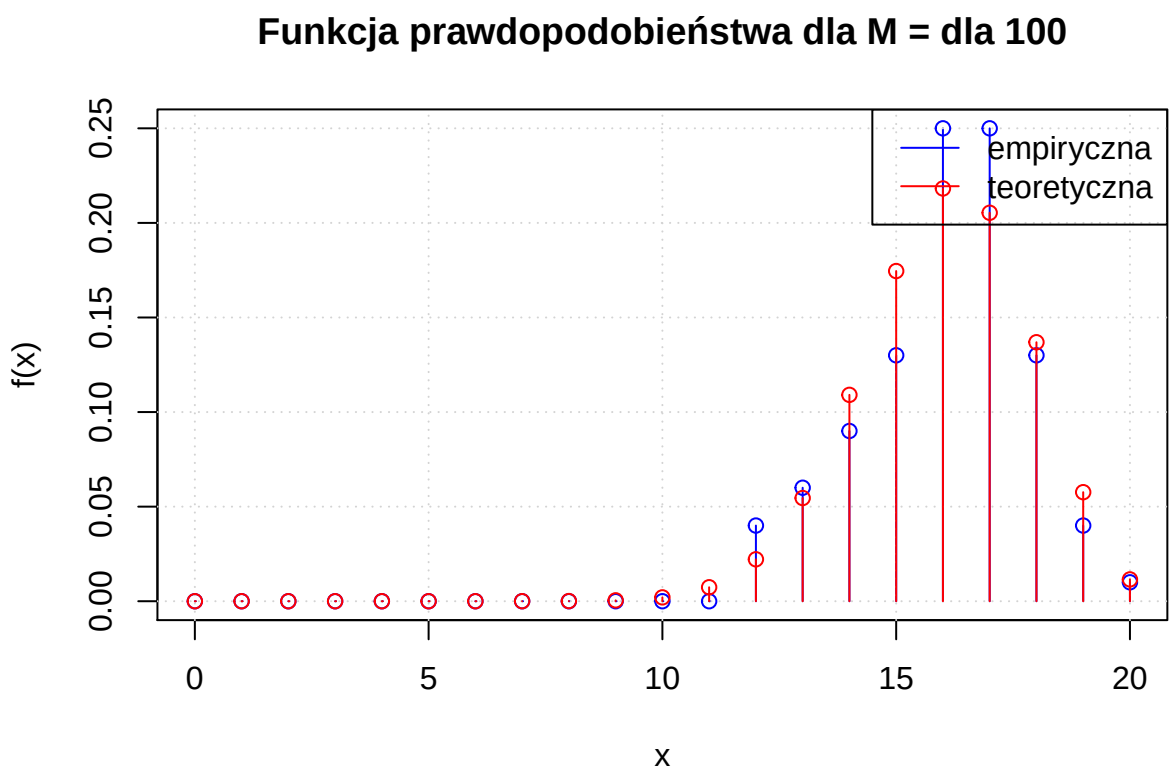
Rozwiązanie

- Trzy próby losowe (M = 100, 1000, 10000) dla rozkładu dwumianowego Binom(20, 0.8)

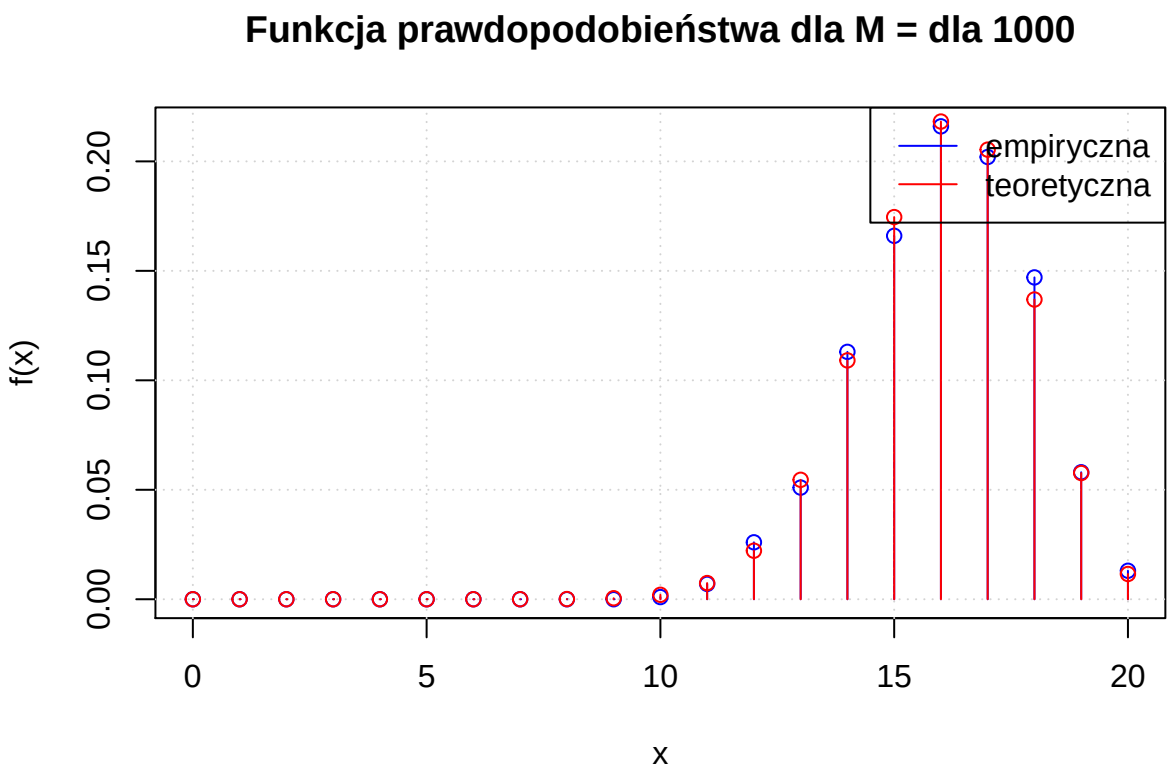
```
M1 = 100
proba1 = rbinom(M1, 20, 0.8)
M2 = 1000
proba2 = rbinom(M2, 20, 0.8)
M3 = 10000
proba3 = rbinom(M3, 20, 0.8)
```

- Empiryczne i teoretyczne funkcje prawdopodobieństwa

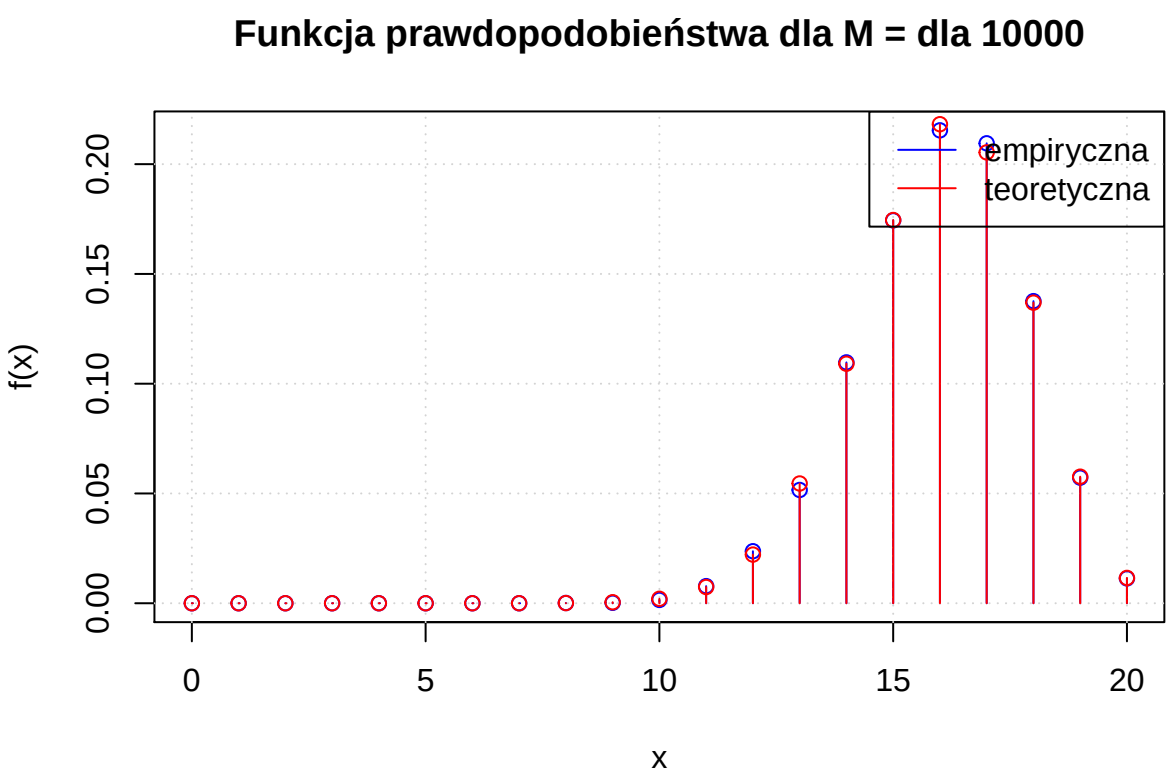
```
Freq100 = as.numeric(table(factor(proba1, levels = Arg))) / M1
plot(Freq100 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
    main = paste0('Funkcja prawdopodobieństwa dla M = dla ', M1))
grid()
points(Freq100 ~ Arg, col = 'blue')
lines(dbinom(Arg, 20, prob = 0.8) ~ Arg, type = 'h', col = 'red',
      xlab = 'x', ylab = 'f(x)')
points(dbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

legend('topright', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```



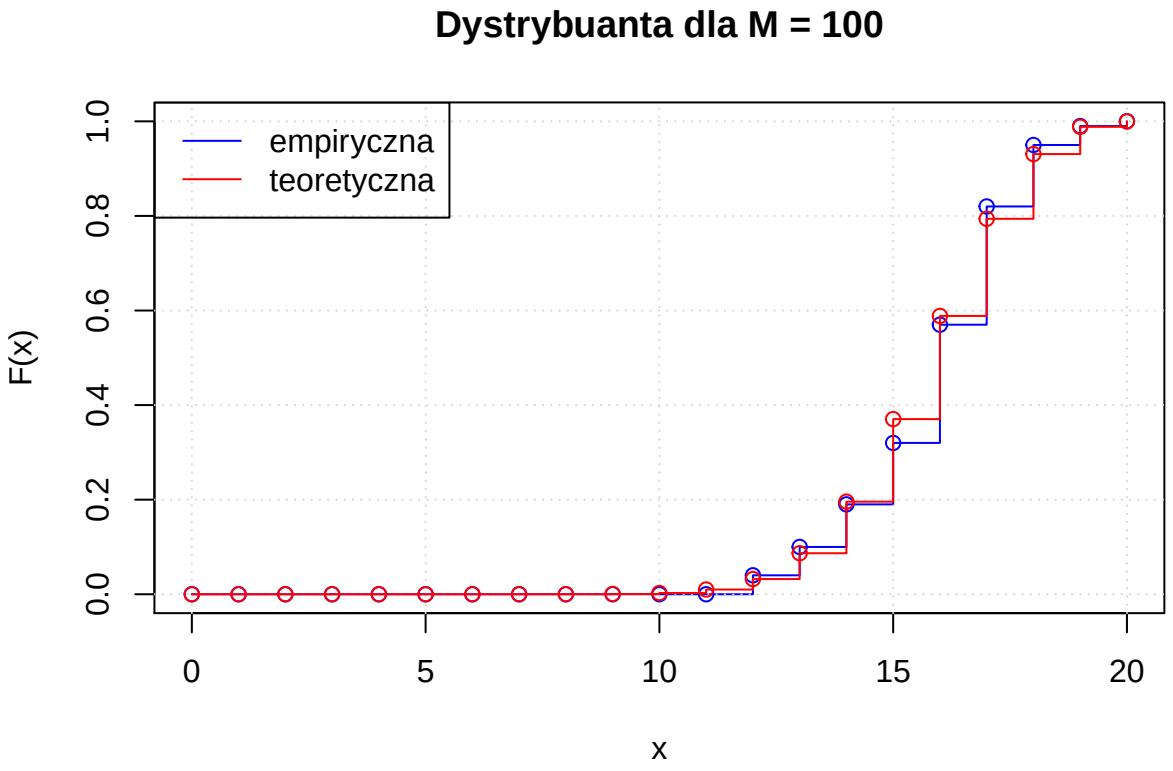

```
Freq1000 = as.numeric(table(factor(proba2, levels = Arg))) / M2
plot(Freq1000 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
    main = paste0('Funkcja prawdopodobieństwa dla M = dla ', M2))
grid()
points(Freq1000 ~ Arg, col = 'blue')
lines(dbinom(Arg, 20, prob = 0.8) ~ Arg, type = 'h', col = 'red',
      xlab = 'x', ylab = 'f(x)')
points(dbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

legend('topright', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```



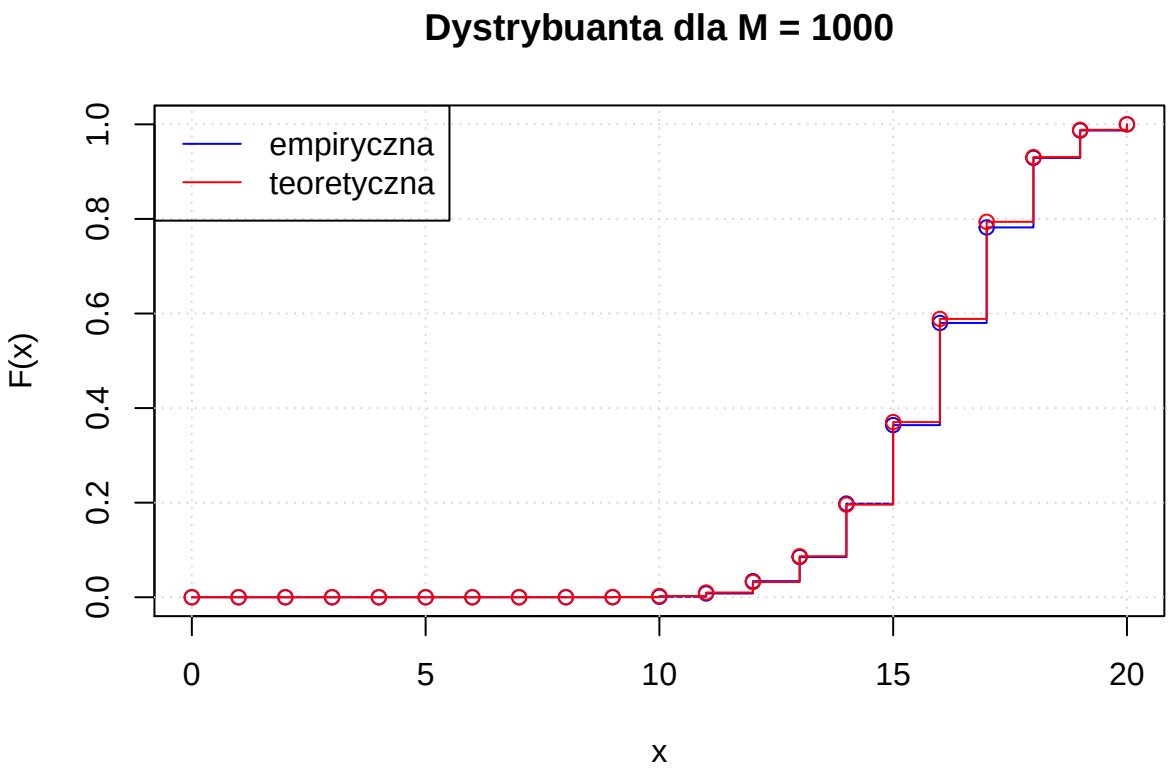

```
Freq10000 = as.numeric(table(factor(proba3, levels = Arg))) / M3
plot(Freq10000 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
    main = paste0('Funkcja prawdopodobieństwa dla M = dla ', M3))
grid()
points(Freq10000 ~ Arg, col = 'blue')
lines(dbinom(Arg, 20, prob = 0.8) ~ Arg, type = 'h', col = 'red',
      xlab = 'x', ylab = 'f(x)')
points(dbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

legend('topright', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```



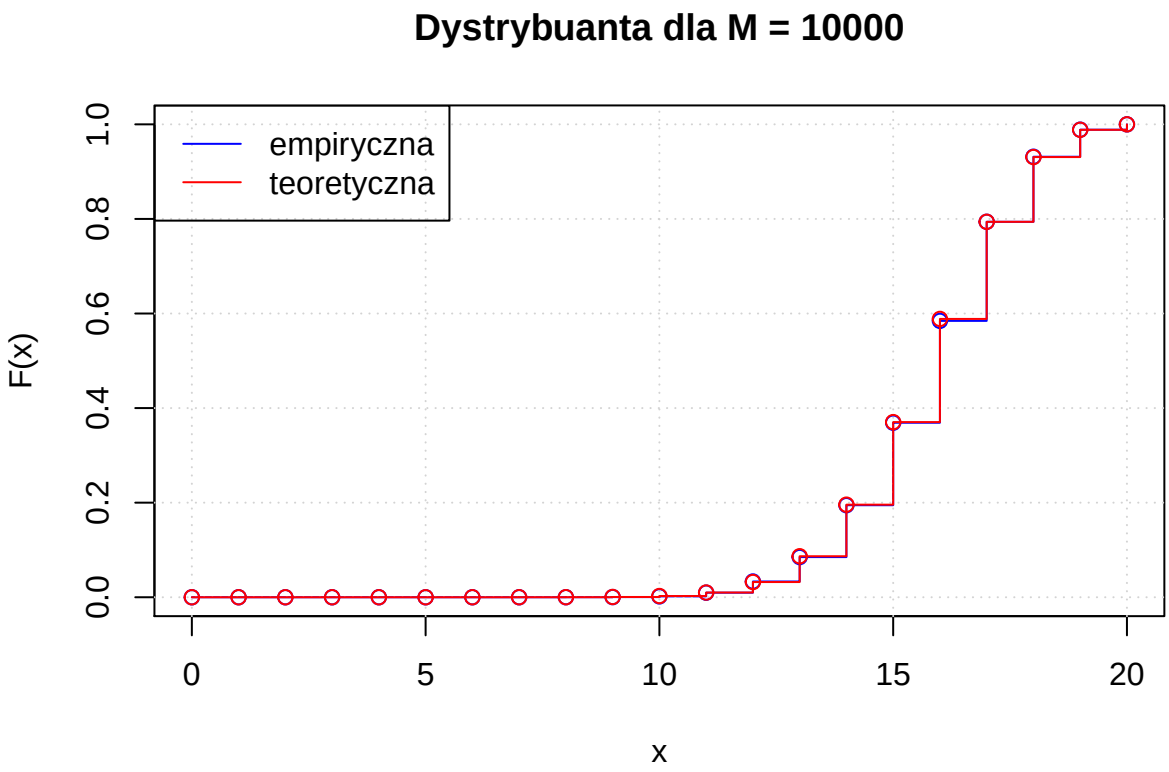

```
plot(cumsum(Freq100) ~ Arg, type = 's', col = 'blue',
    xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybuanta dla M = ', M1))
grid()
points(cumsum(Freq100) ~ Arg, col = 'blue')

lines(pbinom(Arg, 20, prob = 0.8) ~ Arg, type = 's', col = 'red',
      xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

legend('topleft', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```




```
plot(cumsum(Freq1000) ~ Arg, type = 's', col = 'blue',
    xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybuanta dla M = ', M2))
grid()
points(cumsum(Freq1000) ~ Arg, col = 'blue')

lines(pbinom(Arg, 20, prob = 0.8) ~ Arg, type = 's', col = 'red',
      xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

legend('topleft', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```




```
plot(cumsum(Freq10000) ~ Arg, type = 's', col = 'blue',
    xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybuanta dla M = ', M3))
grid()
points(cumsum(Freq10000) ~ Arg, col = 'blue')

lines(pbinom(Arg, 20, prob = 0.8) ~ Arg, type = 's', col = 'red',
      xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg, 20, prob = 0.8) ~ Arg, col = 'red')

legend('topleft', c('empiryczna', 'teoretyczna'),
    col = c('blue', 'red'), lwd = 1)
```



- wartości średnie i wariancje

```
m1 = mean(proba1); v1 = var(proba1)
m2 = mean(proba2); v2 = var(proba2)
m3 = mean(proba3); v3 = var(proba3)
```

Wyestymowane parametry rozkładu M1 = 100 wynoszą: wartość średnia 16.02, wariancja 2.9895. Wyestymowane parametry rozkładu M2 = 1000 wynoszą: wartość średnia 16.032, wariancja 3.2262. Wyestymowane parametry rozkładu M3v= 10000 wynoszą: wartość średnia 16.0079, wariancja 3.1778.

Zadanie 5 (1 pkt)

Treść zadania

1. Wygeneruj K = 500 realizacji (powtórzeń) prób losowych składających się z M = 100 próbek pochodzących z rozkładu Binom(20, 0.8).
2. Dla wszystkich realizacji oblicz wartości średnie i wariancje. Następnie narysuj histogramy wartości średnich i histogramy wariancji (przyjmij breaks = 20).
3. Powtórz eksperymenty dla M = 1000 i M = 10000. Wyjaśnij dlaczego zmieniają się histogramy wraz ze zmianą liczby próbek?

Wskazówka:

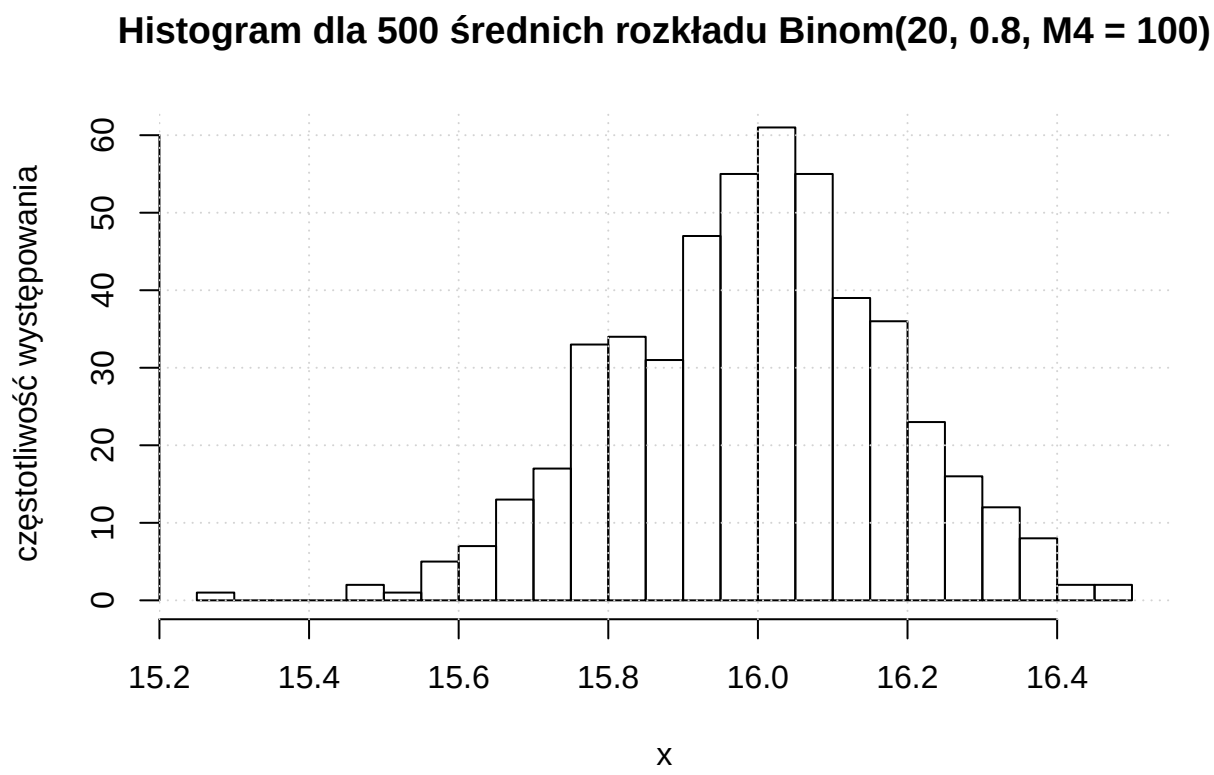
mm = replicate(500, mean(rbinom(M, 20, 0.8)))

Rozwiązanie

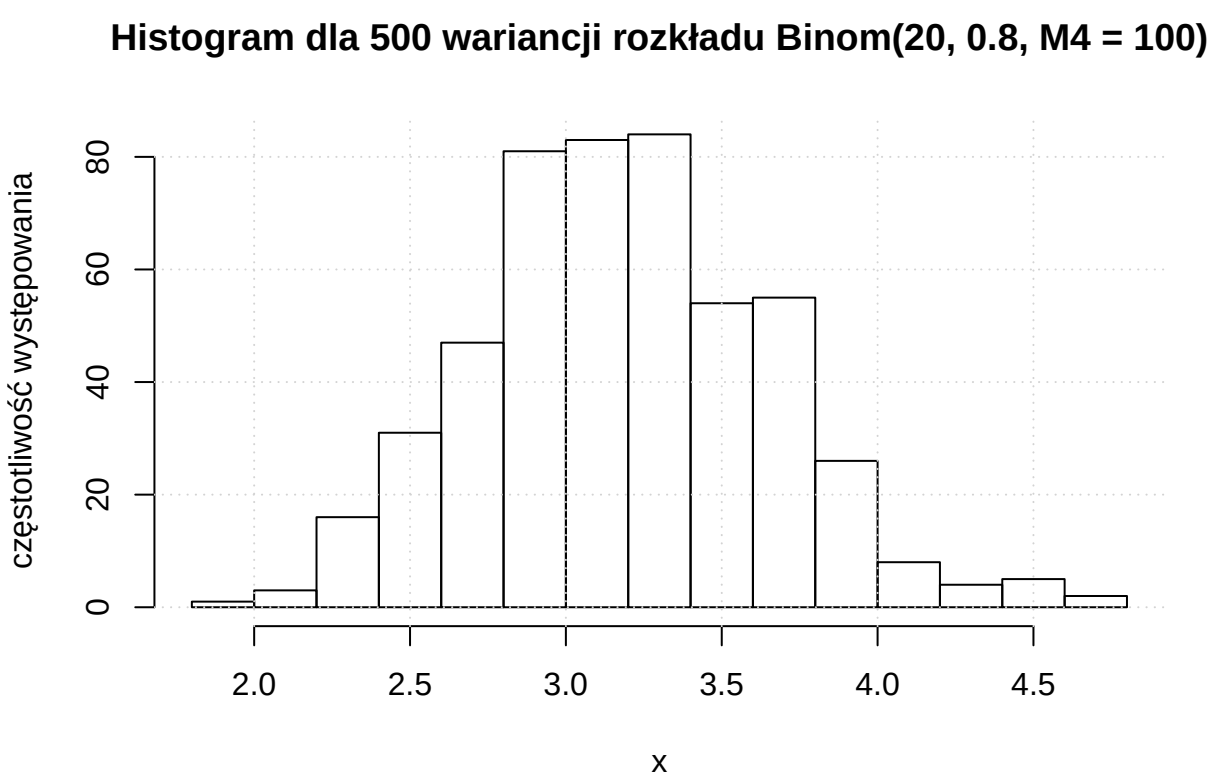
- Dla M = 100

```
M4 = 100
mm4 = replicate(500, mean(rbinom(M4, 20, 0.8)))
vv4 = replicate(500, var(rbinom(M4, 20, 0.8)))
```

```
hist(mm4, breaks = 20, xlab = 'x', ylab = 'częstotliwość występowania',
     main = 'Histogram dla 500 średnich rozkładu Binom(20, 0.8, M4 = 100)')
grid()
```



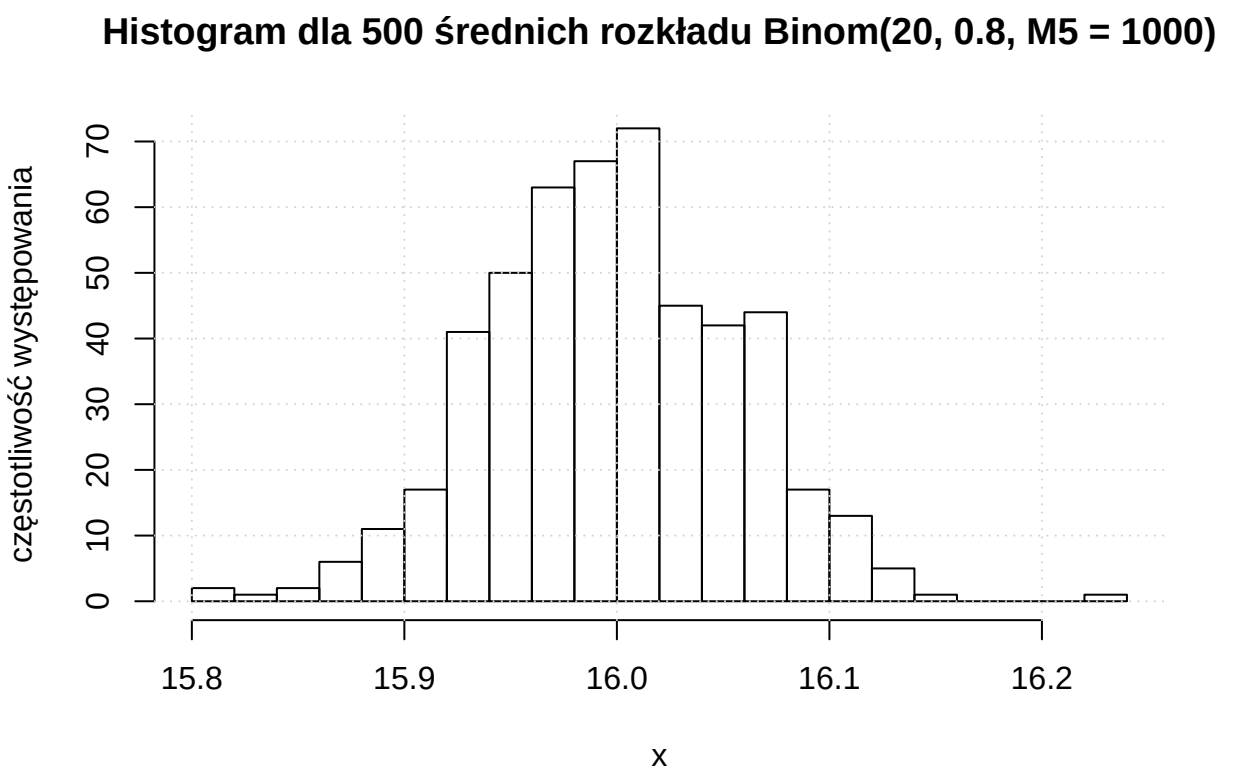

```
hist(vv4, breaks = 20, xlab = 'x', ylab = 'częstotliwość występowania',
     main = 'Histogram dla 500 wariancji rozkładu Binom(20, 0.8, M4 = 100)')
grid()
```



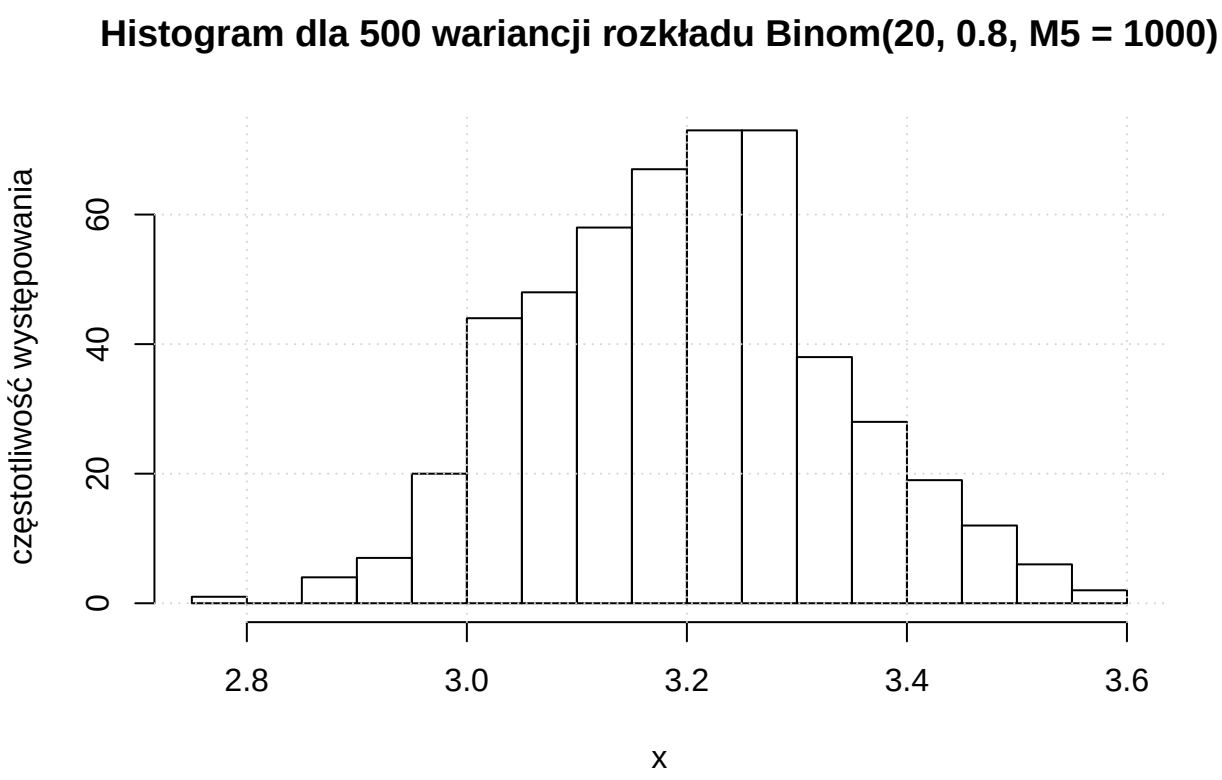
- Dla M = 1000

```
M5 = 1000
mm5 = replicate(500, mean(rbinom(M5, 20, 0.8)))
vv5 = replicate(500, var(rbinom(M5, 20, 0.8)))
```

```
hist(mm5, breaks = 20, xlab = 'x', ylab = 'częstotliwość występowania',
     main = 'Histogram dla 500 średnich rozkładu Binom(20, 0.8, M5 = 1000)')
grid()
```



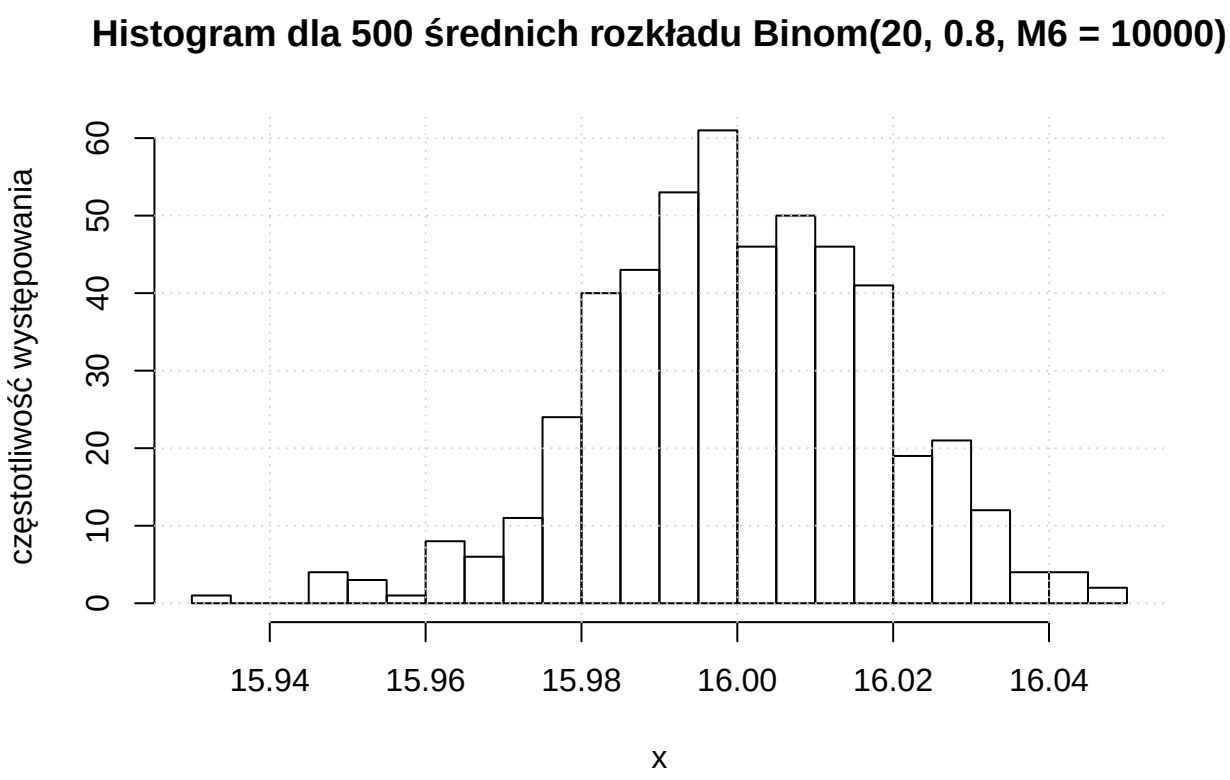

```
hist(vv5, breaks = 20, xlab = 'x', ylab = 'częstotliwość występowania',
     main = 'Histogram dla 500 wariancji rozkładu Binom(20, 0.8, M5 = 1000)')
grid()
```



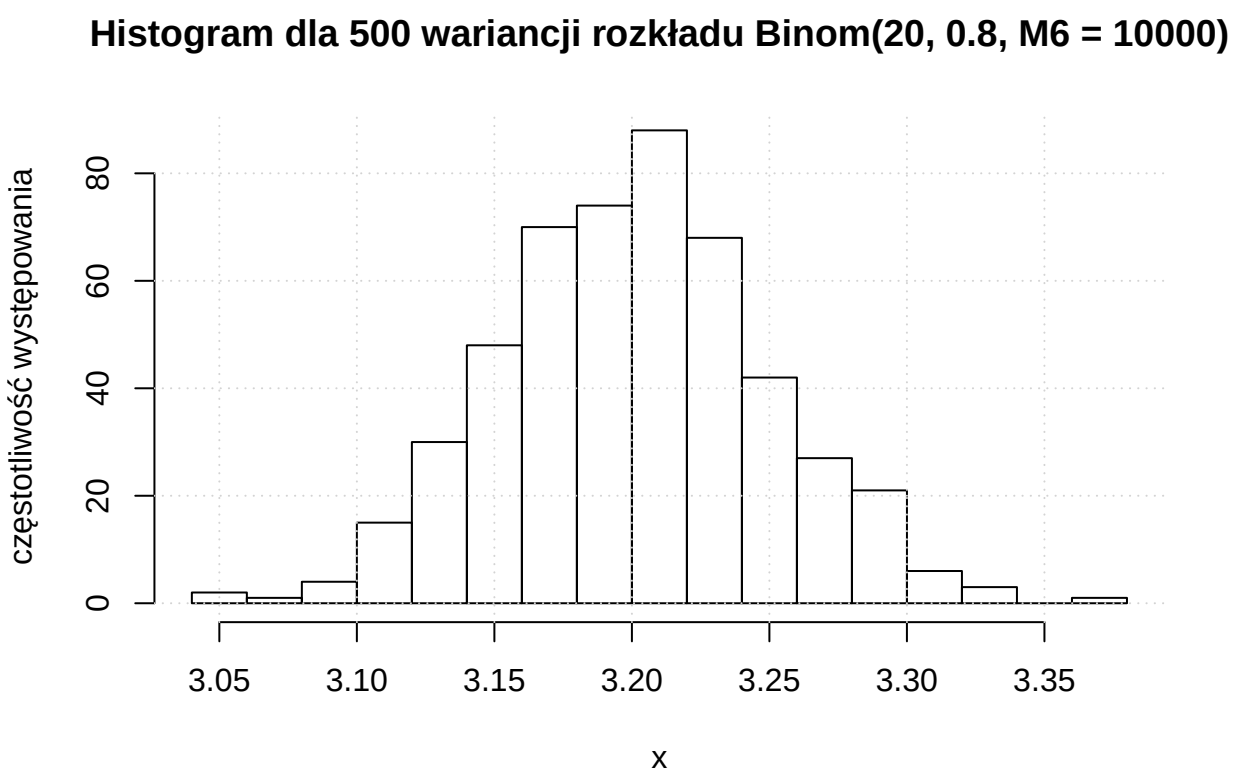
* Dla M = 10000

```
M6 = 10000
mm6 = replicate(500, mean(rbinom(M6, 20, 0.8)))
vv6 = replicate(500, var(rbinom(M6, 20, 0.8)))
```

```
hist(mm6, breaks = 20, xlab = 'x', ylab = 'częstotliwość występowania',
     main = 'Histogram dla 500 średnich rozkładu Binom(20, 0.8, M6 = 10000)')
grid()
```




```
hist(vv6, breaks = 20, xlab = 'x', ylab = 'częstotliwość występowania',
     main = 'Histogram dla 500 wariancji rozkładu Binom(20, 0.8, M6 = 10000)')
grid()
```


Histogram dla 500 wariancji rozkładu Binom(20, 0.8, M6 = 10000)
częstotliwość występowania
0
20
40
60
80
3.05
3.10
3.15
3.20
3.25
3.30
3.35
x