ADPS 2022L — Laboratorium 1 (rozwiązania)

Adam Pruszyński

Zadanie 1 (1 pkt)

Treść zadania

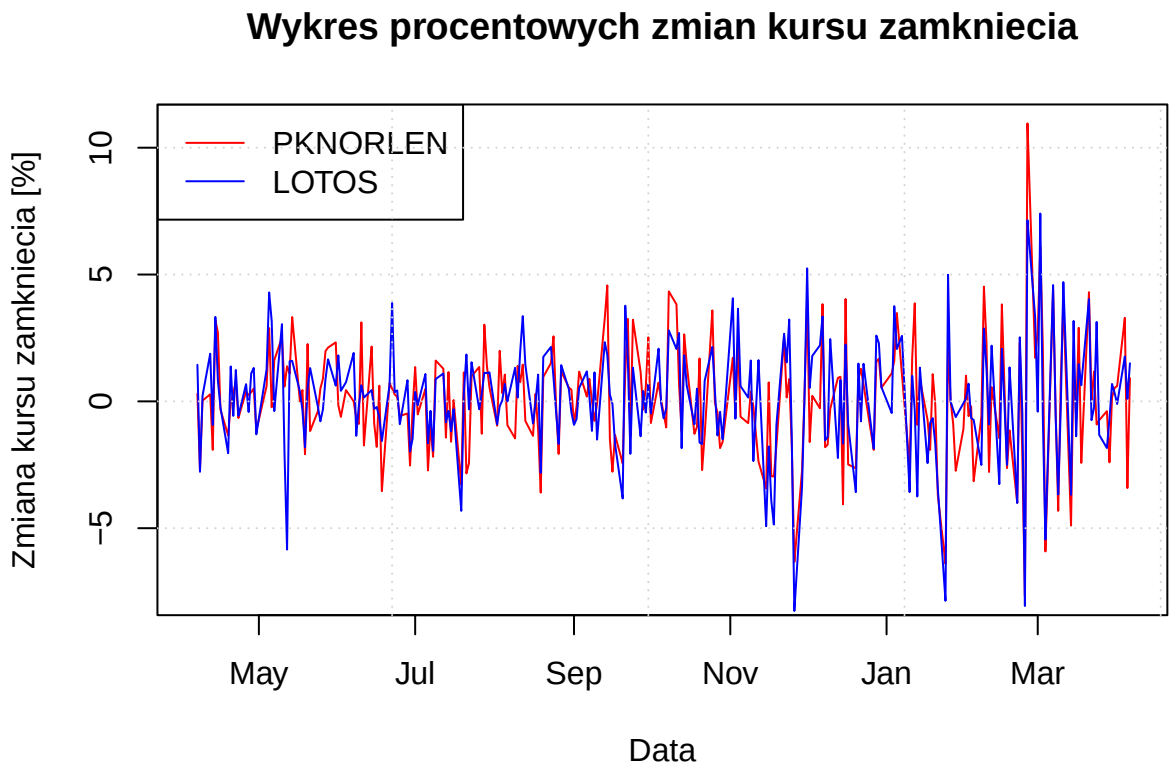
Dla danych z ostatnich 12 miesięcy dotyczacych wybranych dwóch spółek giełdowych:

- 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.

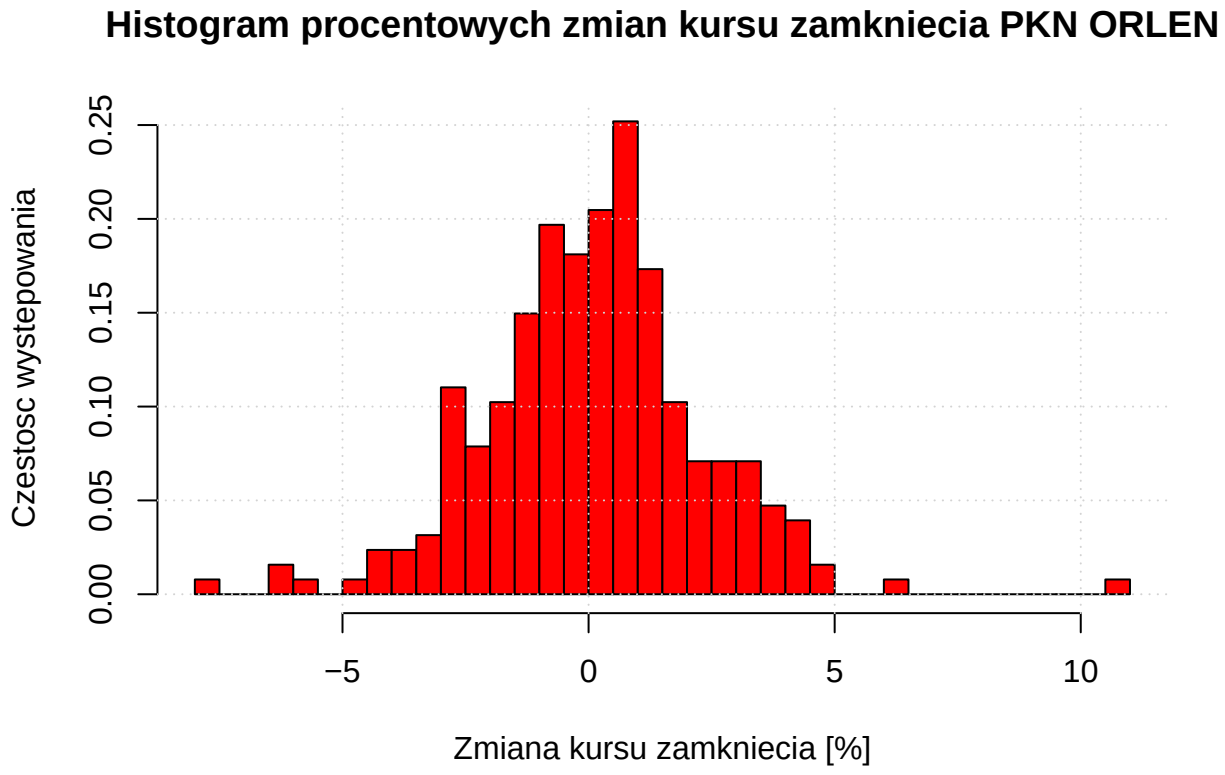
Rozwiązanie

```
if(!file.exists('mstall.zip')) {
download.file('https://info.bossa.pl/pub/metastock/mstock/mstall.zip','mstall.zip')
}

unzip('mstall.zip', files = c('PKNORLEN.mst', 'LOTOS.mst'))

df_PKNORLEN = read.csv('PKNORLEN.mst')
df_LOTOS = read.csv('LOTOS.mst')

col_names = c('ticker', 'date', 'open', 'high', 'low', 'close','vol')
names(df_PKNORLEN) = col_names
names(df_LOTOS) = col_names

df_PKNORLEN$date = as.Date.character(df_PKNORLEN$date, format ='%Y%m%d')
df_LOTOS$date = as.Date.character(df_LOTOS$date, format ='%Y%m%d')

df_PKNORLEN = df_PKNORLEN[which(df_PKNORLEN$date >= '2021-04-06' & df_PKNORLEN$date <= '2022-04-06'),]
df_LOTOS = df_LOTOS[which(df_LOTOS$date >= '2021-04-06' & df_LOTOS$date <= '2022-04-06'),]

df_PKNORLEN$close_ch= with(df_PKNORLEN, c(NA, 100*diff(close)/close[-length(close)]))
df_LOTOS$close_ch= with(df_LOTOS, c(NA, 100*diff(close)/close[-length(close)]))

plot(close_ch ~ date, df_PKNORLEN, type = 'l', col = 'red', xlab = 'Data',
     ylab = 'Zmiana kursu zamknięcia [%]', main = 'Wykres procentowych zmian kursu zamknięcia')

lines(close_ch ~ date, df_LOTOS, type = 'l', col = 'blue',
      xlab = 'Data', ylab = 'Zmiana kursu zamknięcia [%]')
```

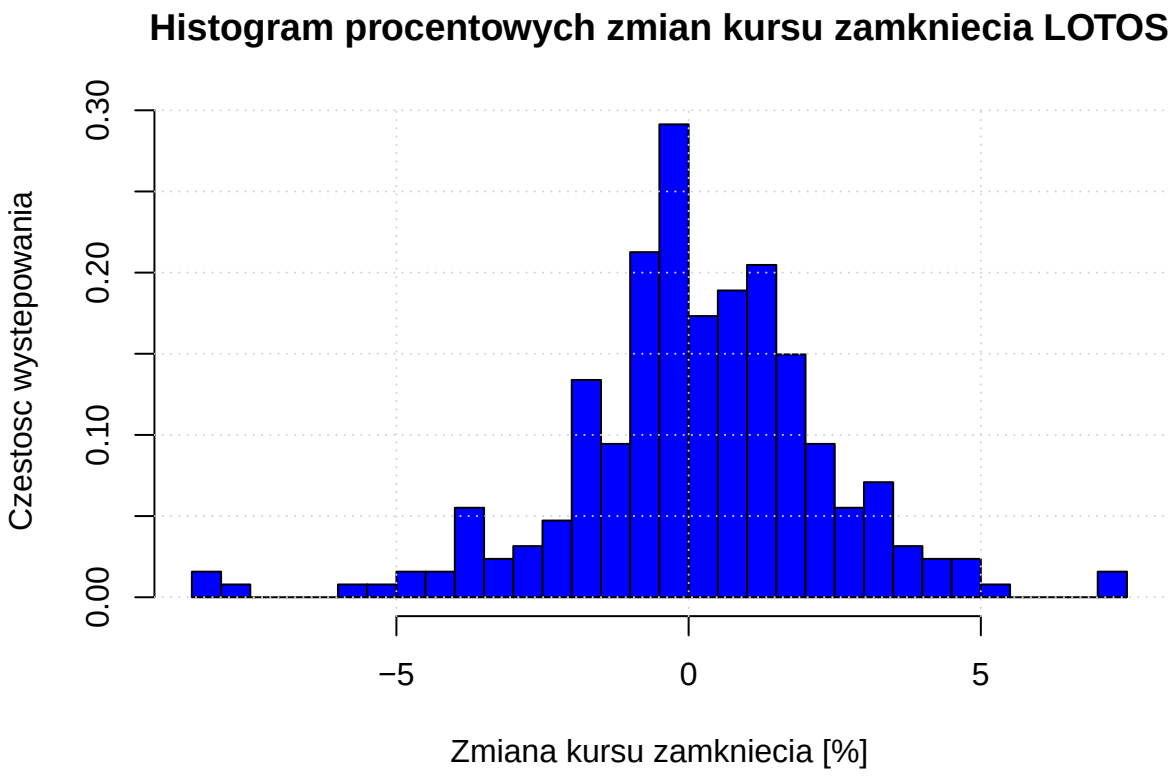
```
legend('topleft', c('PKNORLEN', 'LOTOS'),
    col = c('red', 'blue'), lwd = 1)

grid()
```




```
hist(df_PKNORLEN$close_ch, breaks = 50, prob = T, col = 'red', xlab = 'Zmiana kursu zamknięcia [%]',
ylab = 'Częstość występowania', main = 'Histogram procentowych zmian kursu zamknięcia PKN ORLEN')
grid()
```

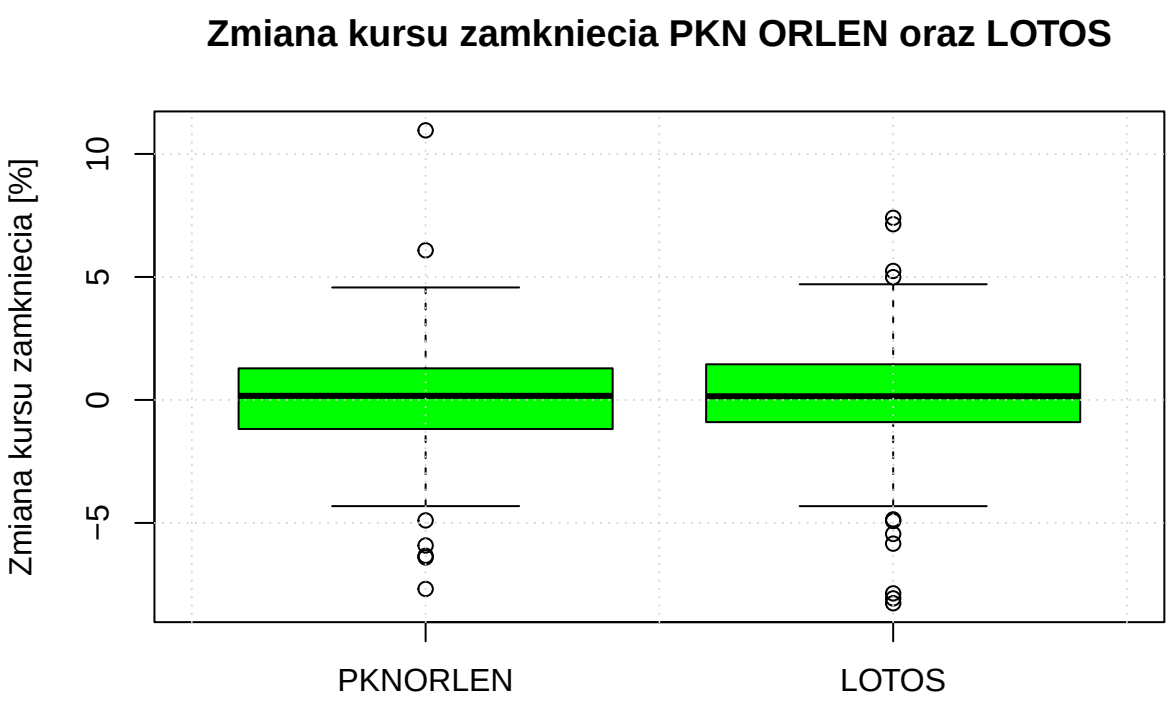



```
hist(df_LOTOS$close_ch, breaks = 50, prob = T, col = 'blue', xlab = 'Zmiana kursu zamknięcia [%]',
ylab = 'Częstość występowania', main = 'Histogram procentowych zmian kursu zamknięcia LOTOS')
grid()
```



Spoglądając na oba histogramy możemy stwierdzić, iż najczęściej procentowa zamiana kursu zamknięcia, zarówno dla PKN ORLEN oraz LOTOS oscylowała przy wartości 0. Świadczyć to może o stabilej sytuacji obu spółek.

```
data = data.frame(PKNORLEN = df_PKNORLEN$close_ch, LOTOS = df_LOTOS$close_ch)

boxplot(data, col = 'green', ylab = 'Zmiana kursu zamknięcia [%]', main = 'Zmiana kursu zamknięcia PKN
grid()
```



***

Zadanie 2 (1 pkt)

Treść zadania

1. Sporządź wykres liczby katastrof lotniczych w poszczególnych:

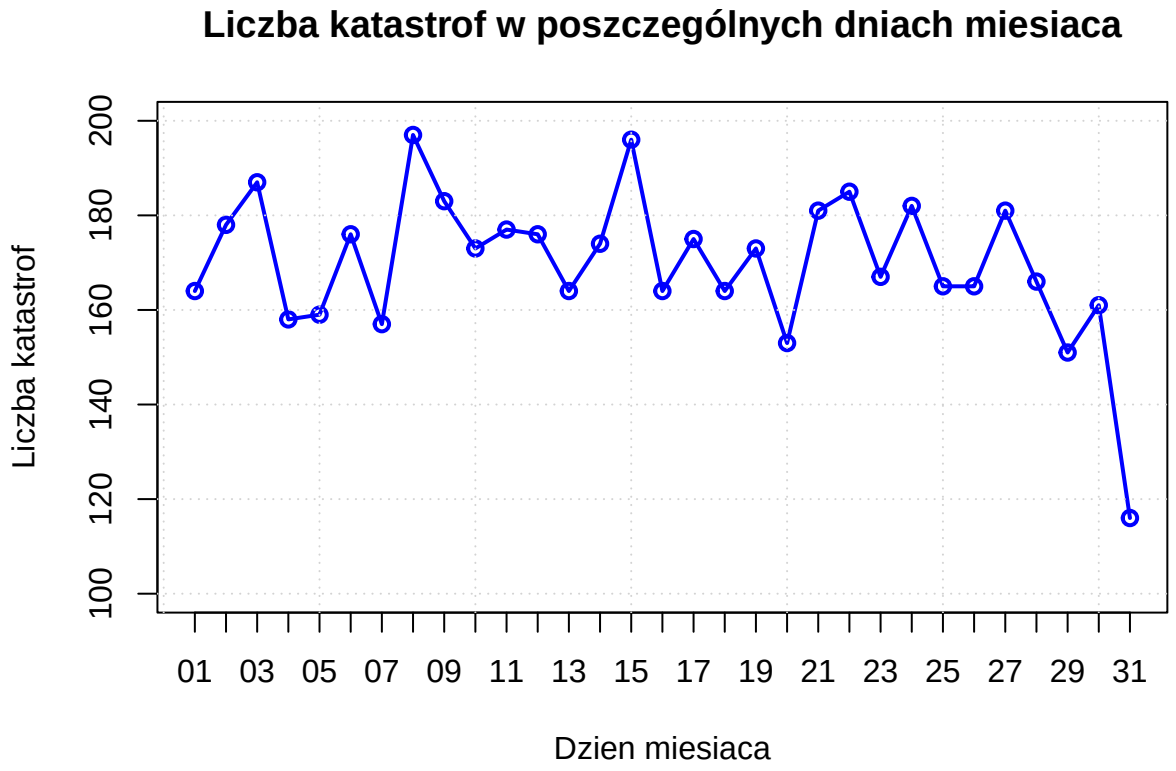
- miesiącach roku (styczeń - grudzień),
- dniach miesiąca (1-31),
- dniach tygodnia (weekdays()).

2. Narysuj jak w kolejnych latach zmieniały się:

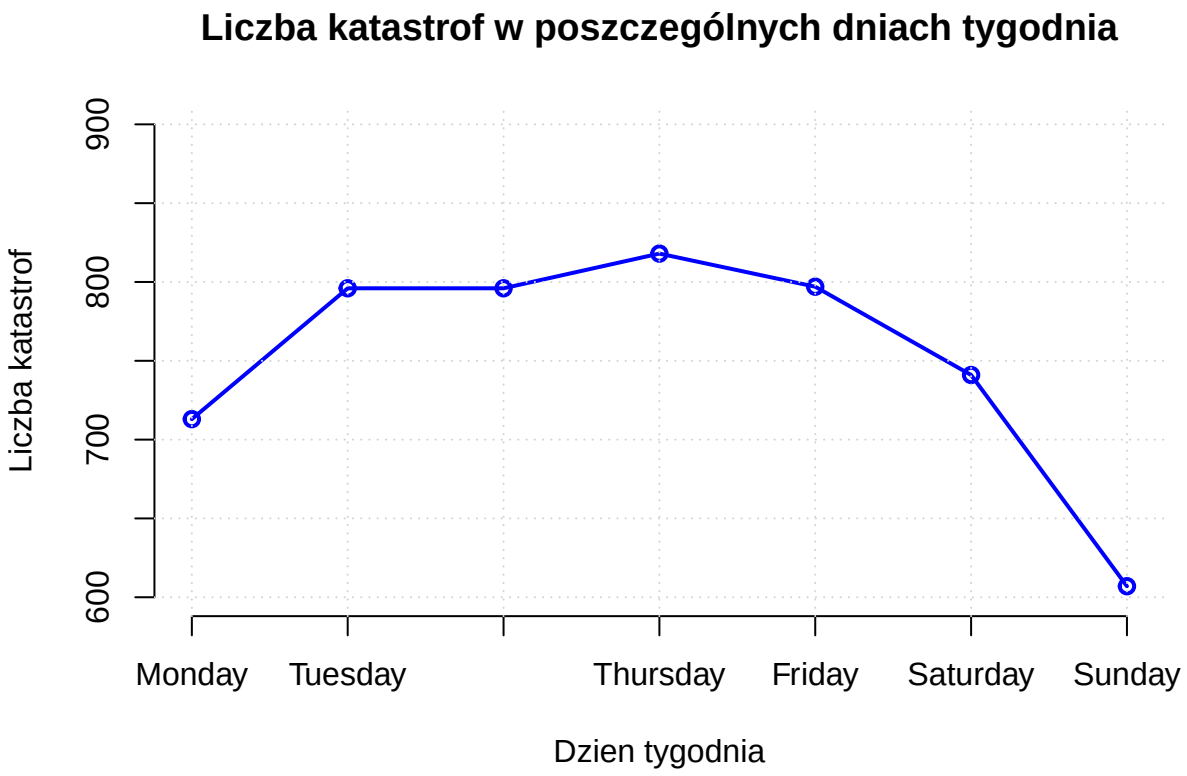
- liczba osób, które przeżyły katastrofy,
- odsetek osób (w procentach), które przeżyły katastrofy.

Rozwiązanie

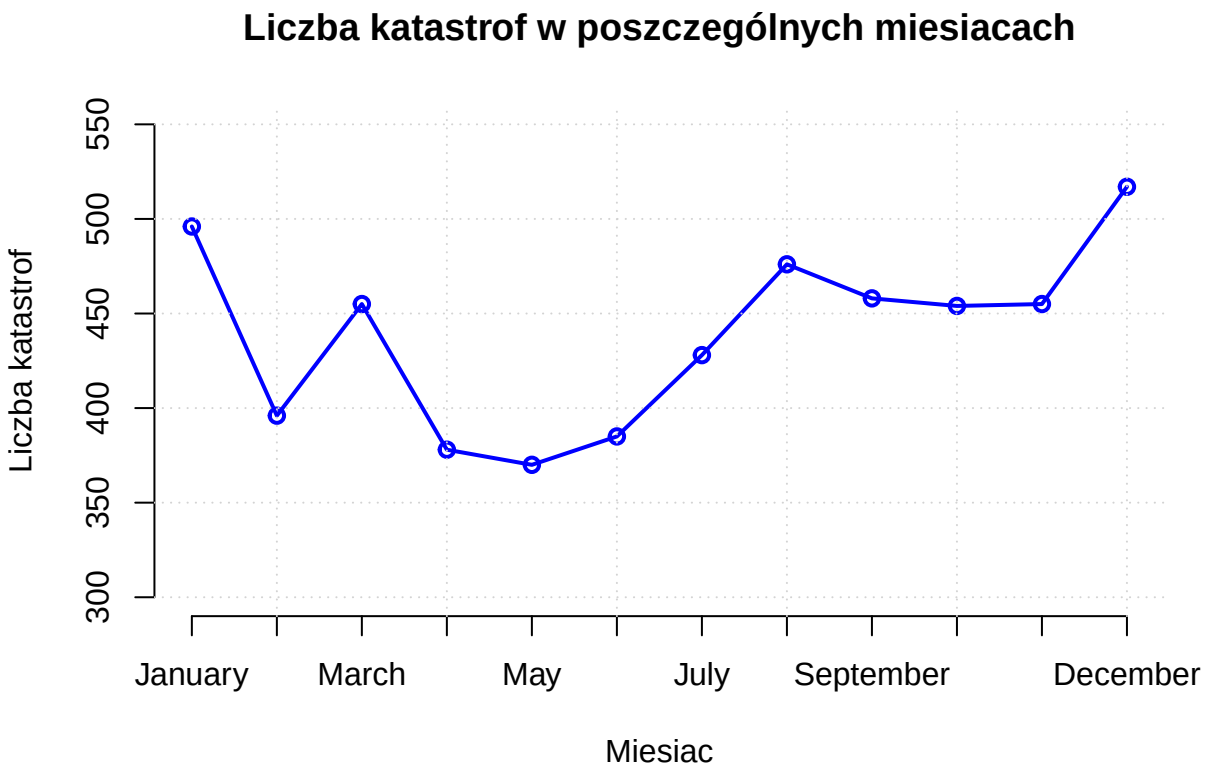
```
kat = read.csv('crashes.csv')

kat$Days = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%d')

kat$Weekdays = weekdays(as.Date(kat$Date, '%m/%d/%Y'))
kat$Weekdays = ordered(kat$Weekdays, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",

kat$Months = months(as.Date(kat$Date, '%m/%d/%Y'))
kat$Months = ordered(kat$Months, levels = c("January", "February", "March", "April", "May", "June", "Ju

plot(table(kat$Days), type = 'o', col = 'blue',  xlab = 'Dzień miesiąca',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych dniach miesiąca', ylim = c(100, 20
grid()
```




```
plot(table(kat$Weekdays), type = 'o', col = 'blue', xlab = 'Dzień tygodnia', ylim = c(600, 900),
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych dniach tygodnia' )
grid()
```




```
plot(table(kat$Months), type = 'o', col = 'blue', xlab = 'Miesiąc', ylim = c(300, 550),
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych miesiącach' )
grid()
```




```
kat$Year = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%Y')
kat$Ocaleni = kat$Aboard - kat$Fatalities

Ocaleni_agr = aggregate(Ocaleni ~ Year, kat, FUN = sum)

plot(Ocaleni_agr, type = 'l', col = 'blue',xlab = 'Rok',
ylab = 'Lczba ocalałych', main = 'Liczba ocalałych w katastrofach w poszczególnych latach' )
grid()
```


```
kat$Ocaleni_ch= with(kat, round(100*(kat$Ocaleni / kat$Aboard), digits = 2))
ocaleni_ch_agr = aggregate(Ocaleni_ch ~ Year, kat, FUN = mean)

plot(ocaleni_ch_agr, type = 'l', col = 'blue', xlab = 'Rok', ylab = 'Lczba ocalałych',
     main = 'Procent ocalałych osób w katastrofach w poszczególnych latach' )
grid()
```

Zadanie 3 (1 pkt)

Treść zadania

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

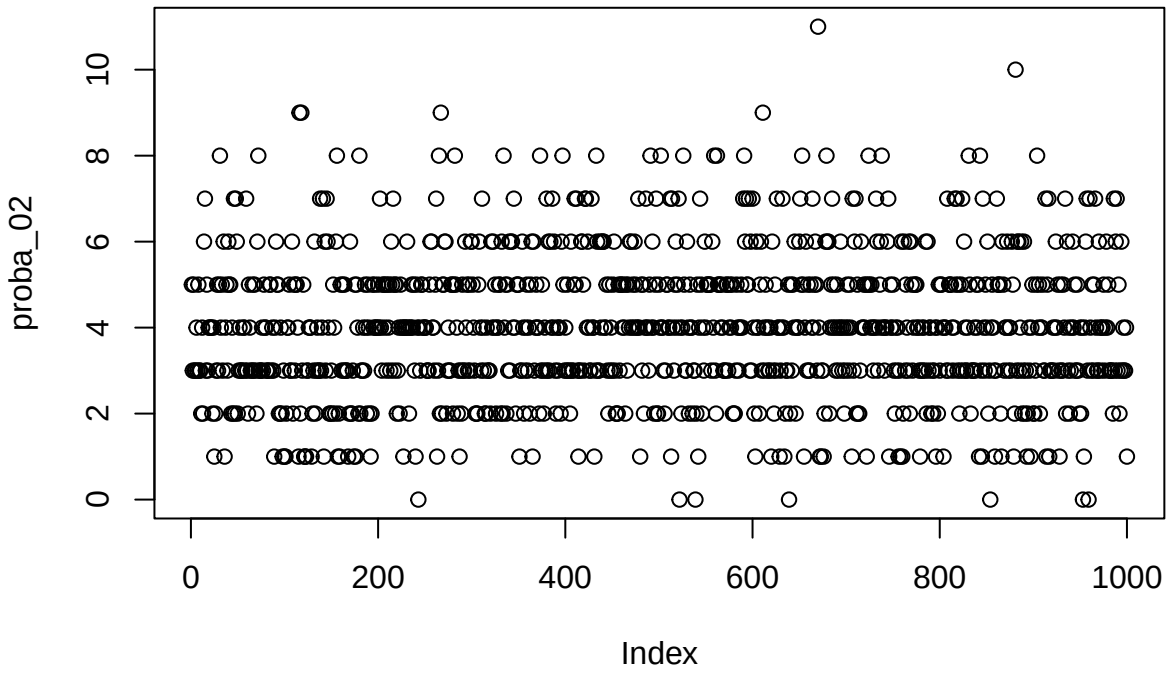
- Binom(20,0.2)
- Binom(20,0.8)

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

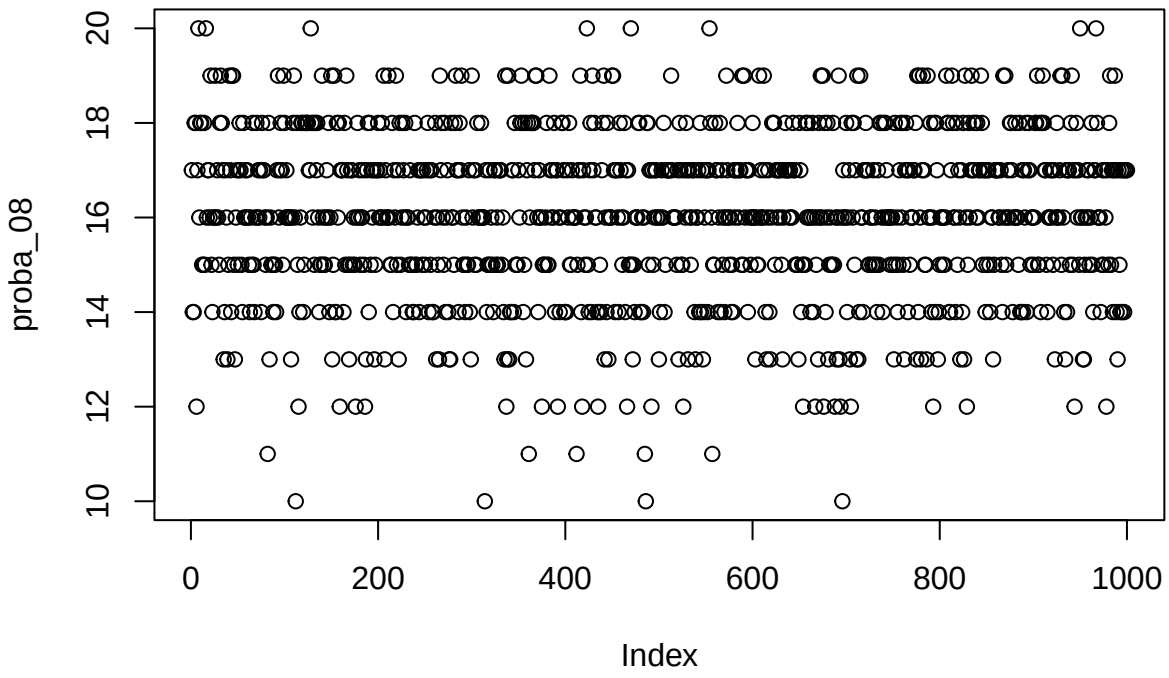
2. Dla obu rozkładów narysuj na jednym rysunku empiryczne i teoretyczne (użyj funkcji dbinom) funkcje prawdopodobieństwa, a na drugim rysunku empiryczne i teoretyczne (użyj funkcji pbinom) dystrybuanty. W obu przypadkach wyskaluj oś odciętych od 0 do 20.

Rozwiązanie

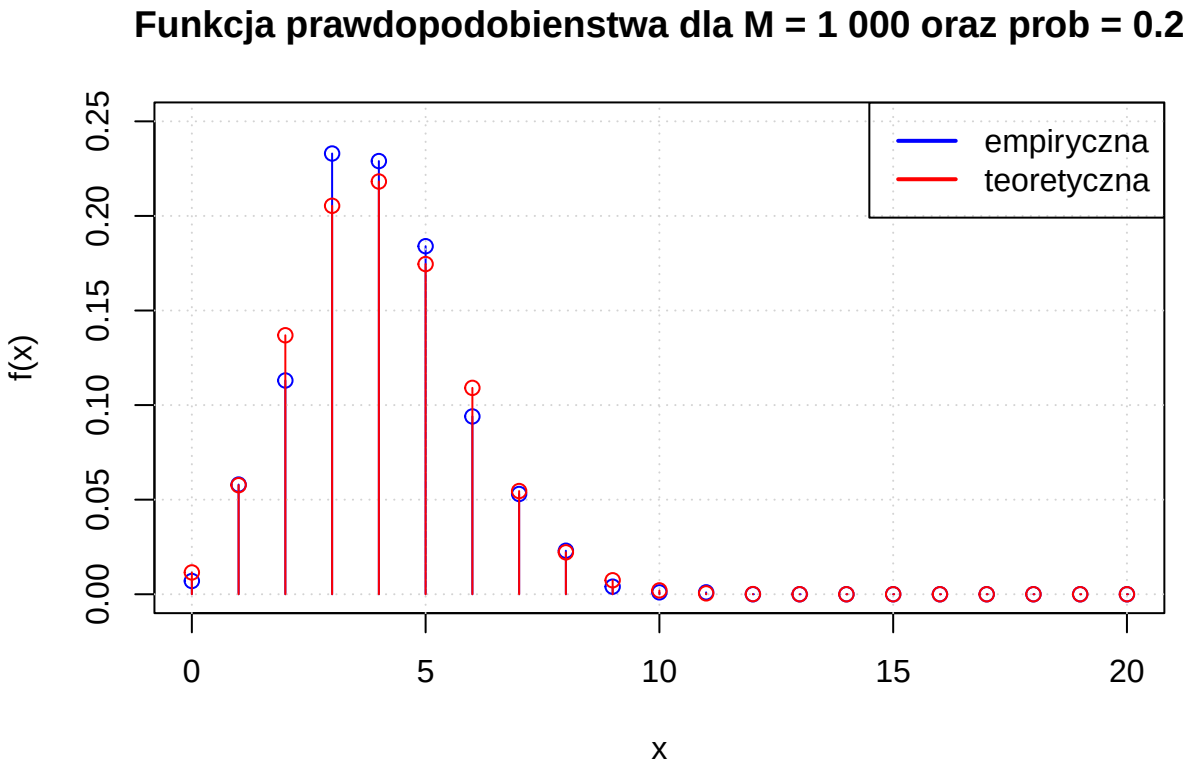
```r
proba_02 = rbinom(1000, size = 20, prob = 0.2)
proba_08 = rbinom(1000, size = 20, prob = 0.8)

plot(proba_02)
```



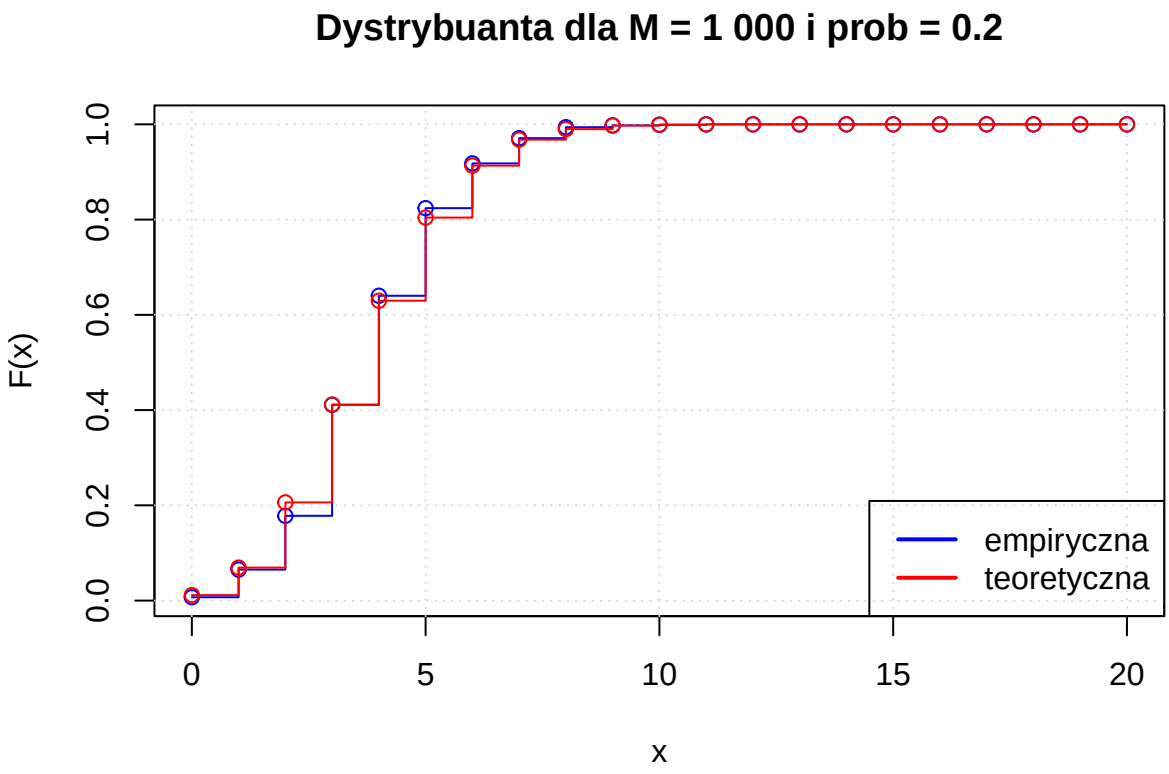

```r
plot(proba_08)
```



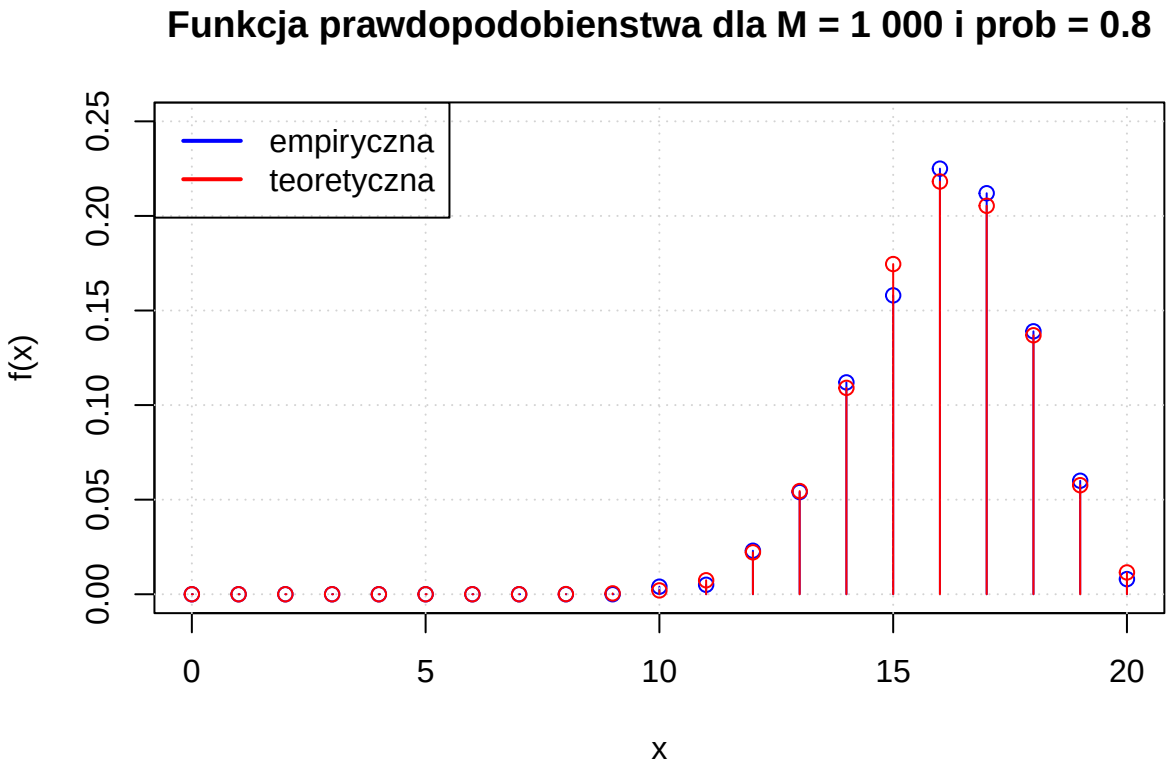

```
Arg = 0:20
Freq = as.numeric(table(factor(proba_02, levels = Arg))) / 1000
plot(Freq ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = 'Funkcja prawdopodobieństwa dla M = 1 000 oraz prob = 0.2', ylim = c(0, 0.25))
grid()
points(Freq ~ Arg, col = 'blue')

theoretical_02 = dbinom(Arg, size = 20, prob = 0.2)
lines(theoretical_02 ~ Arg, type = 'h', col = 'red', xlab = 'x', ylab = 'f(x)', xlim = c(0,20))
points(theoretical_02 ~ Arg, col = 'red')

legend('topright', c('empiryczna', 'teoretyczna'), col = c('blue', 'red'), lwd = 2)
```



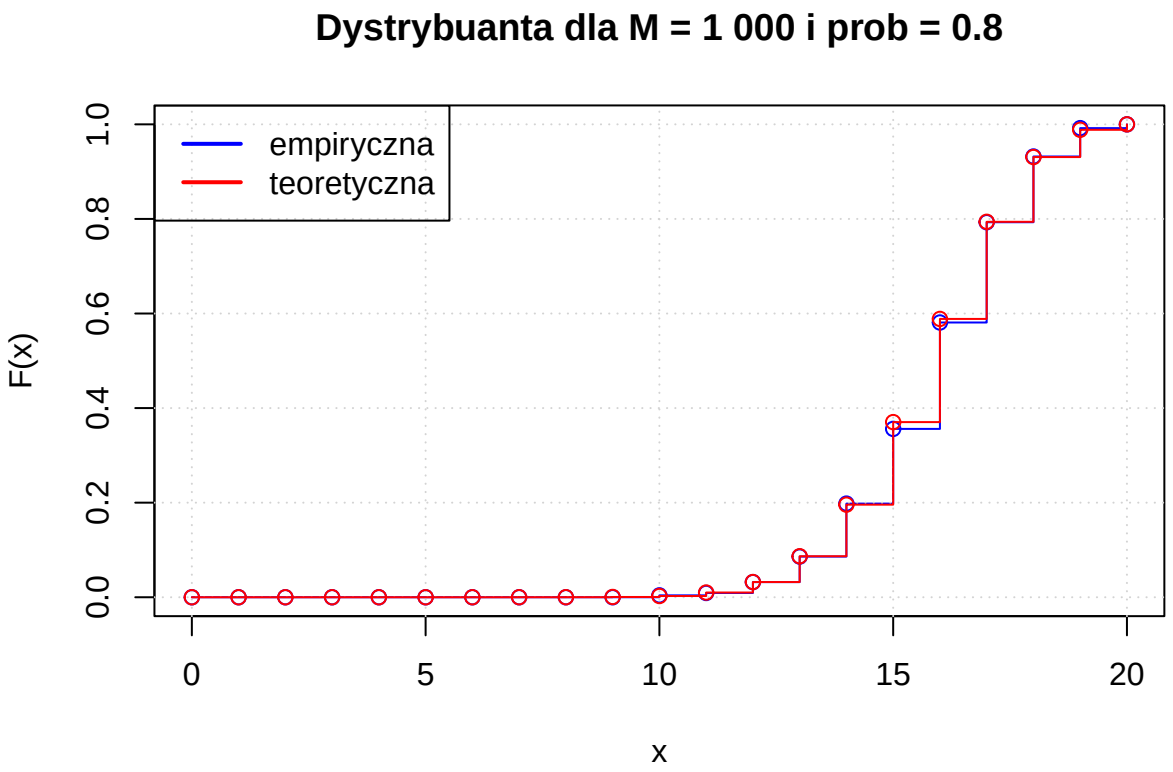

```
plot(cumsum(Freq) ~ Arg, type = 's', col = 'blue',
xlab = 'x', ylab = 'F(x)', main = 'Dystrybuanta dla M = 1 000 i prob = 0.2', xlim = c(0,20))
grid()
points(cumsum(Freq) ~ Arg, col = 'blue')

lines(pbinom(Arg, size = 20, prob = 0.2, lower.tail = TRUE, log.p = FALSE) ~ Arg, type = 's', col = 'red
xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg, size = 20, prob = 0.2, lower.tail = TRUE, log.p = FALSE) ~ Arg, col = 'red')
legend('bottomright', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 2)
```




```
Arg = 0:20
Freq = as.numeric(table(factor(proba_08, levels = Arg))) / 1000
plot(Freq ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = 'Funkcja prawdopodobieństwa dla M = 1 000 i prob = 0.8', xlim = c(0,20), ylim = c(0, 0.25))
grid()
points(Freq ~ Arg, col = 'blue')

theoretical_08 = dbinom(Arg, size = 20, prob = 0.8)
lines(theoretical_08 ~ Arg, type = 'h', col = 'red', xlab = 'x', ylab = 'f(x)', xlim = c(0,20))
points(theoretical_08 ~ Arg, col = 'red')

legend('topleft', c('empiryczna', 'teoretyczna'), col = c('blue', 'red'), lwd = 2)
```




```
plot(cumsum(Freq) ~ Arg, type = 's', col = 'blue',
xlab = 'x', ylab = 'F(x)', main = 'Dystrybuanta dla M = 1 000 i prob = 0.8', xlim = c(0,20))
grid()
points(cumsum(Freq) ~ Arg, col = 'blue')

lines(pbinom(Arg, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg, type = 's', col = 'red
xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg, col = 'red')
legend('topleft', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 2)
```



Zadanie 4 (1 pkt)

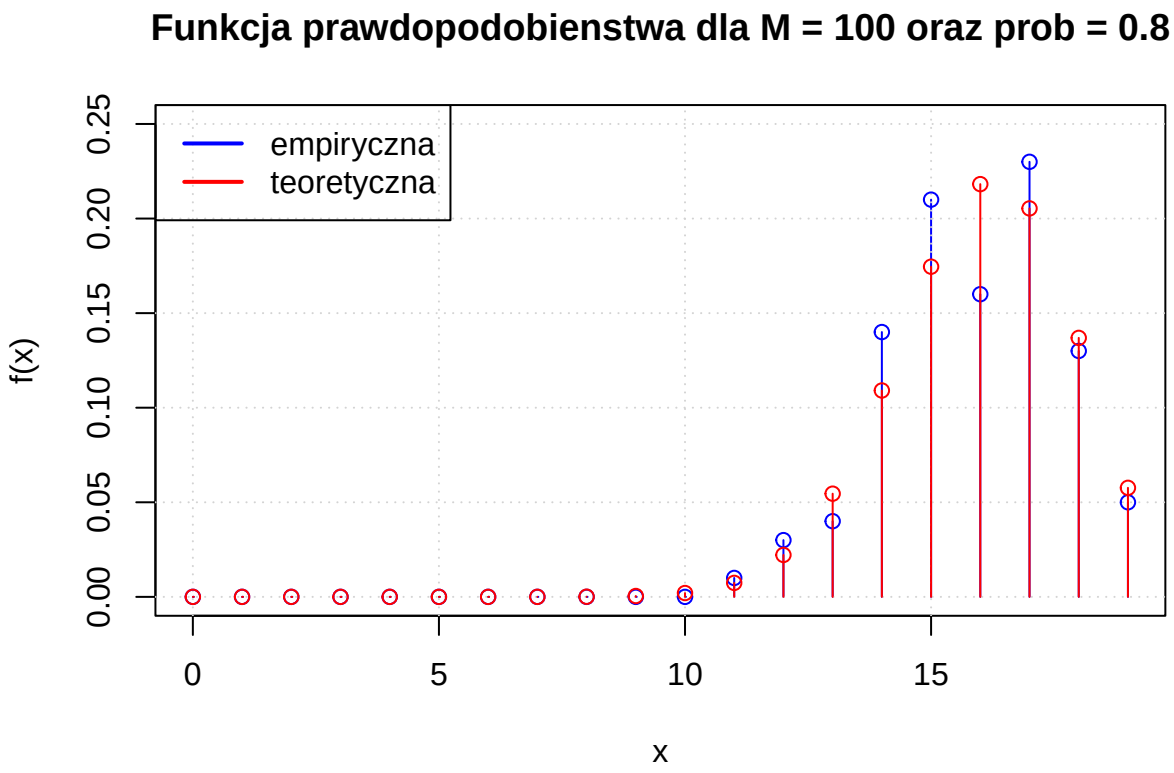
Treść zadania

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).

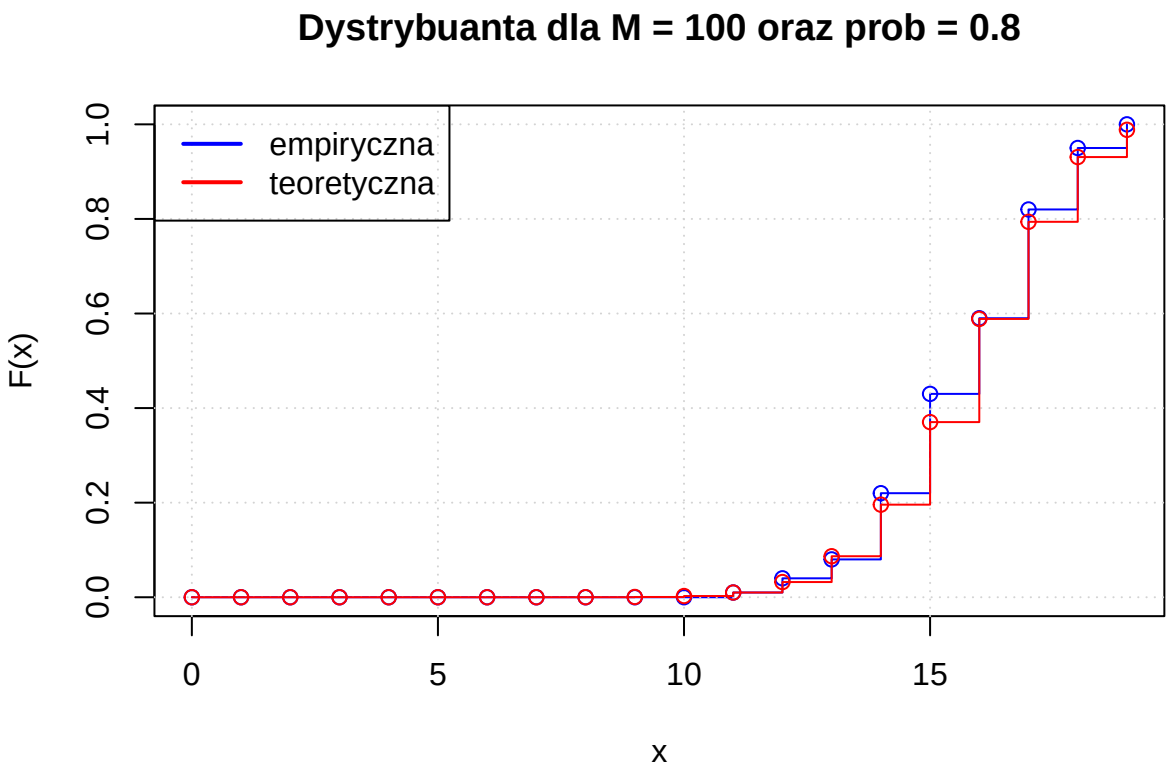
Rozwiązanie

```
proba_100 = rbinom(100, size = 20, prob = 0.8)
proba_1000 = rbinom(1000, size = 20, prob = 0.8)
proba_10000 = rbinom(10000, size = 20, prob = 0.8)

M = 100
```

```r
Arg_100 = 0:max(proba_100)
Freq_100 = as.numeric(table(factor(proba_100, levels = Arg_100))) / M
plot(Freq_100 ~ Arg_100, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = 'Funkcja prawdopodobieństwa dla M = 100 oraz prob = 0.8', ylim = c(0,0.25))
grid()
points(Freq_100 ~ Arg_100, col = 'blue')

theoretical_100 = dbinom(Arg_100, size = 20, prob = 0.8)
lines(theoretical_100 ~ Arg_100, type = 'h', col = 'red', xlab = 'x', ylab = 'f(x)')
points(theoretical_100 ~ Arg_100, col = 'red')

legend('topleft', c('empiryczna', 'teoretyczna'), col = c('blue', 'red'), lwd = 2)
```



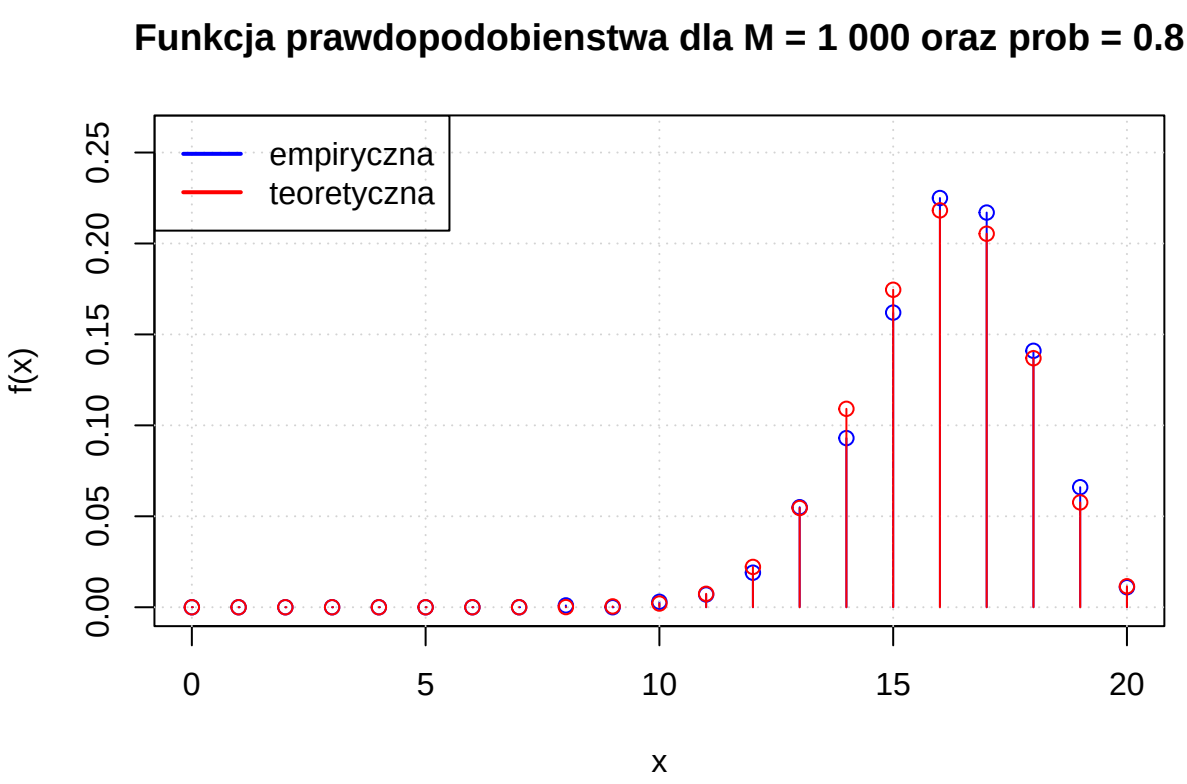

```r
plot(cumsum(Freq_100) ~ Arg_100, type = 's', col = 'blue',
xlab = 'x', ylab = 'F(x)', main = 'Dystrybuanta dla M = 100 oraz prob = 0.8')
grid()
points(cumsum(Freq_100) ~ Arg_100, col = 'blue')

lines(pbinom(Arg_100, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg_100, type = 's', c
xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg_100, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg_100, col = 'red')
legend('topleft', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 2)
```



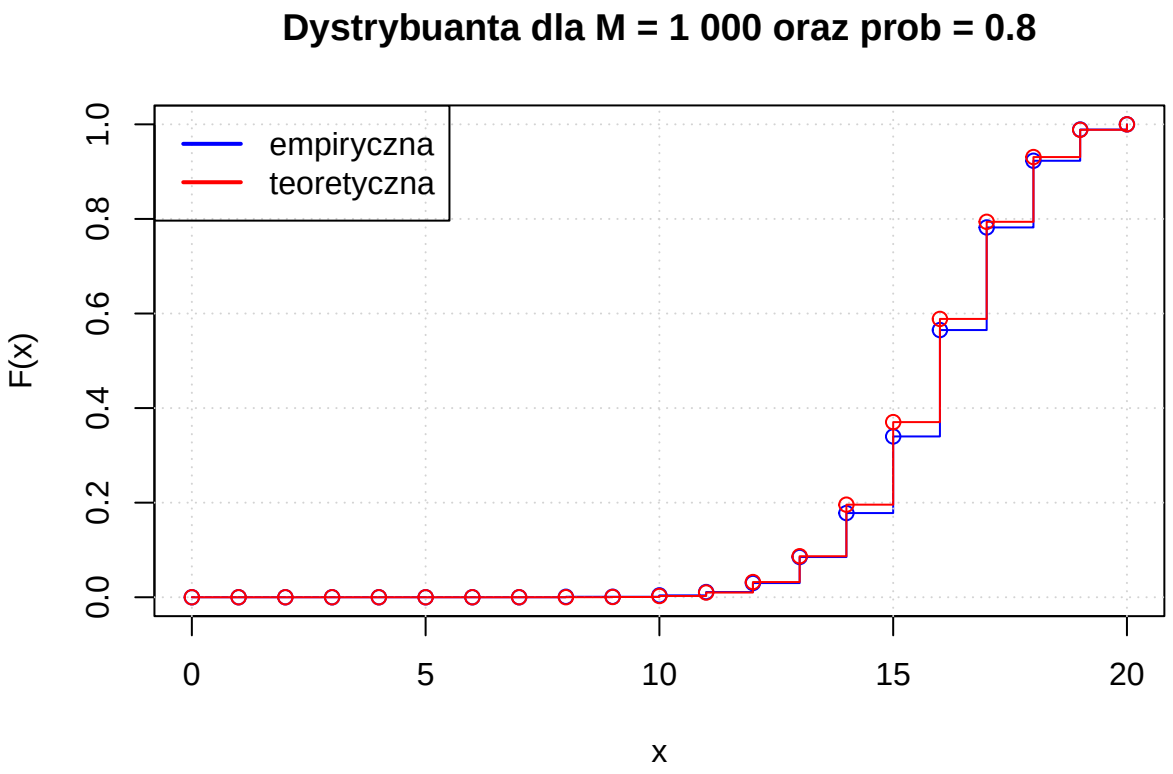

```
empirical_mean_100 = mean(proba_100)
empirical_var_100 = var(proba_100)
```

Empiryczne parametry rozkładu dwumianowego dla podanych wartości wynoszą: średnia 15.86, wariancja 3.0711.

```
M = 1000
Arg_1000 = 0:max(proba_1000)
Freq_1000 = as.numeric(table(factor(proba_1000, levels = Arg_1000))) / M
plot(Freq_1000 ~ Arg_1000, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = 'Funkcja prawdopodobieństwa dla M = 1 000 oraz prob = 0.8', ylim = c(0, 0.26))
grid()
points(Freq_1000 ~ Arg_1000, col = 'blue')

theoretical_1000 = dbinom(Arg_1000, size = 20, prob = 0.8)
lines(theoretical_1000 ~ Arg_1000, type = 'h', col = 'red', xlab = 'x', ylab = 'f(x)')
points(theoretical_1000 ~ Arg_1000, col = 'red')

legend('topleft', c('empiryczna', 'teoretyczna'), col = c('blue', 'red'), lwd = 2)
```



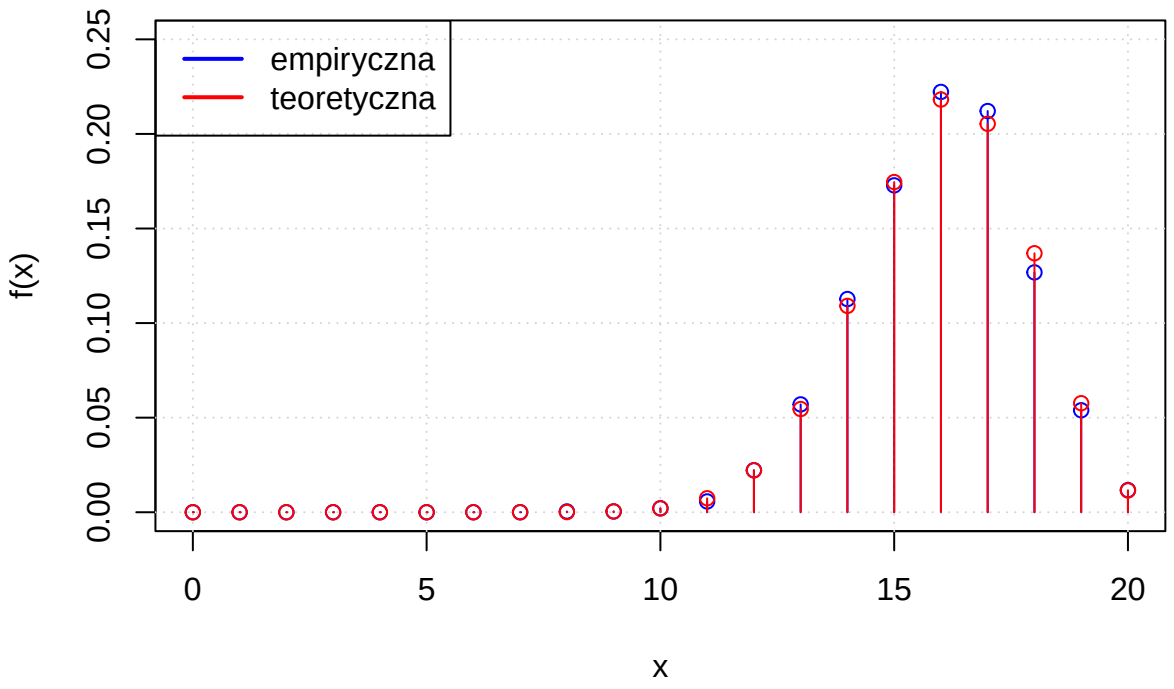

```
plot(cumsum(Freq_1000) ~ Arg_1000, type = 's', col = 'blue',
xlab = 'x', ylab = 'F(x)', main = 'Dystrybuanta dla M = 1 000 oraz prob = 0.8')
grid()
points(cumsum(Freq_1000) ~ Arg_1000, col = 'blue')

lines(pbinom(Arg_1000, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg_1000, type = 's',
xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg_1000, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg_1000, col = 'red
legend('topleft', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 2)
```



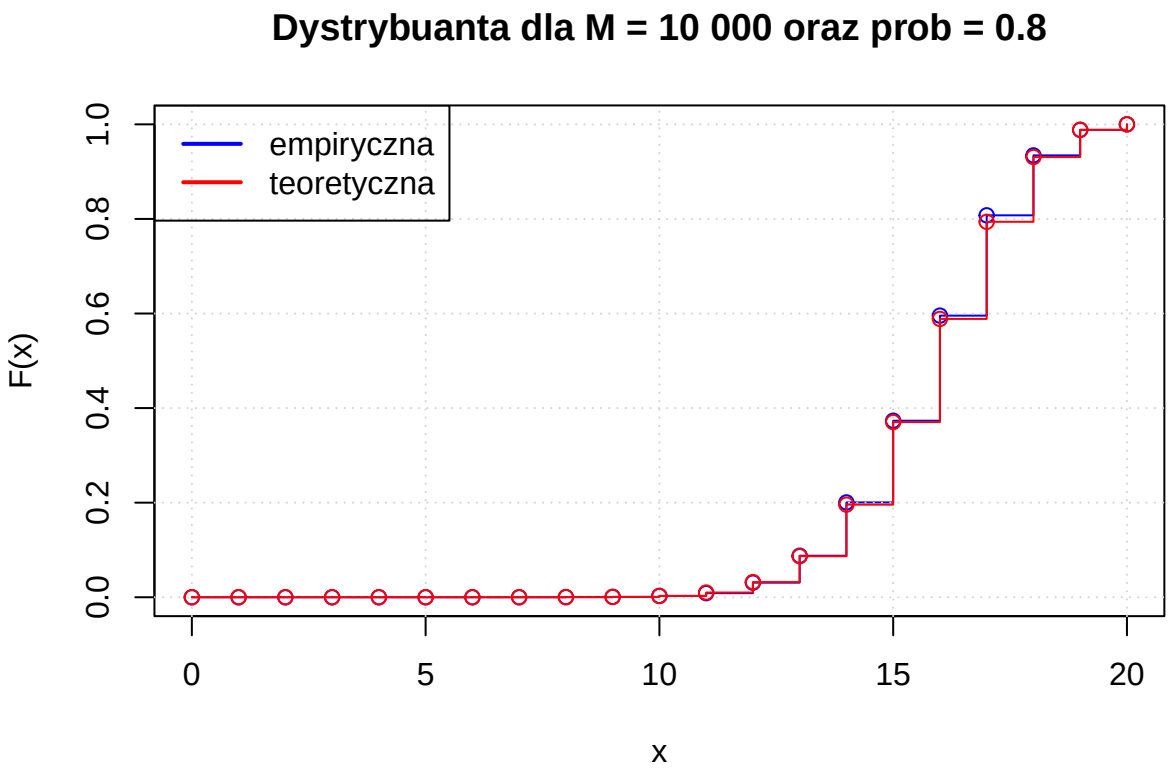

```
empirical_mean_1000 = mean(proba_1000)
empirical_var_1000 = var(proba_1000)
```

Empiryczne parametry rozkładu dwumianowego dla podanych wartości wynoszą: średnia 16.091, wariancja 3.2259.

```
M = 10000
Arg_10000 = 0:max(proba_10000)
Freq_10000 = as.numeric(table(factor(proba_10000, levels = Arg_10000))) / M
plot(Freq_10000 ~ Arg_10000, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = 'Funkcja prawdopodobieństwa dla M = 10 000 oraz prob = 0.8', ylim = c(0, 0.25))
grid()
points(Freq_10000 ~ Arg_10000, col = 'blue')

theoretical_10000 = dbinom(Arg_10000, size = 20, prob = 0.8)
lines(theoretical_10000 ~ Arg_10000, type = 'h', col = 'red', xlab = 'x', ylab = 'f(x)')
points(theoretical_10000 ~ Arg_10000, col = 'red')

legend('topleft', c('empiryczna', 'teoretyczna'), col = c('blue', 'red'), lwd = 2)
```




```
plot(cumsum(Freq_10000) ~ Arg_10000, type = 's', col = 'blue',
xlab = 'x', ylab = 'F(x)', main = 'Dystrybuanta dla M = 10 000 oraz prob = 0.8')
grid()
points(cumsum(Freq_10000) ~ Arg_10000, col = 'blue')

lines(pbinom(Arg_10000, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg_10000, type = 's
xlab = 'x', ylab = 'F(x)')
points(pbinom(Arg_10000, size = 20, prob = 0.8, lower.tail = TRUE, log.p = FALSE) ~ Arg_10000, col = 'r
legend('topleft', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 2)
```




```
empirical_mean_10000 = mean(proba_10000)
empirical_var_10000 = var(proba_10000)
```

Empiryczne parametry rozkładu dwumianowego dla podanych wartości wynoszą: średnia 15.9691, wariancja 3.1461.

```
theoretical_mean = 20 * 0.8
theoretical_var = 20 * 0.8 * (1 - 0.8)
```

Zwiększenie liczby próbek powoduje, że otrzymane wyniki eksperymentalne są bardziej zbliżone do wartości teoretycznych, które wynoszą: średnia 16, wariancja 3.2.

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).

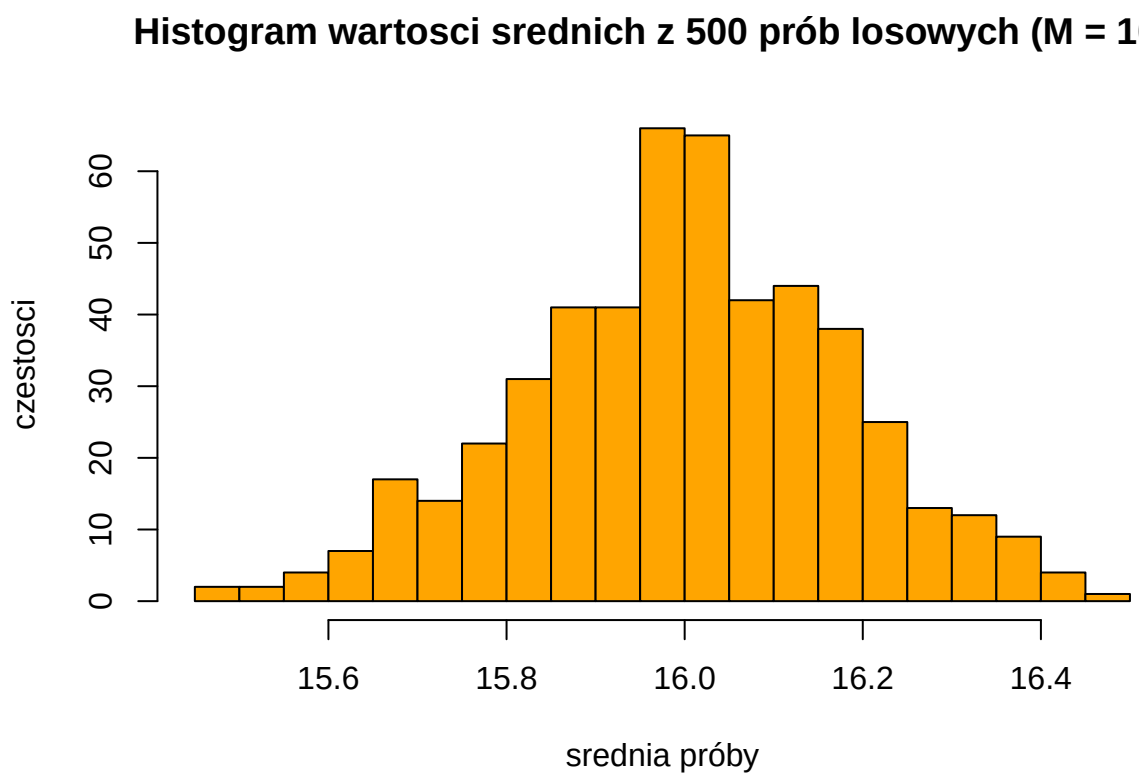
3. Powtórz eksperymenty dla M = 1000 i M = 10000. Wyjaśnij dlaczego zmieniają się histogramy wraz ze zmianą liczby próbek?

Wskazówka:

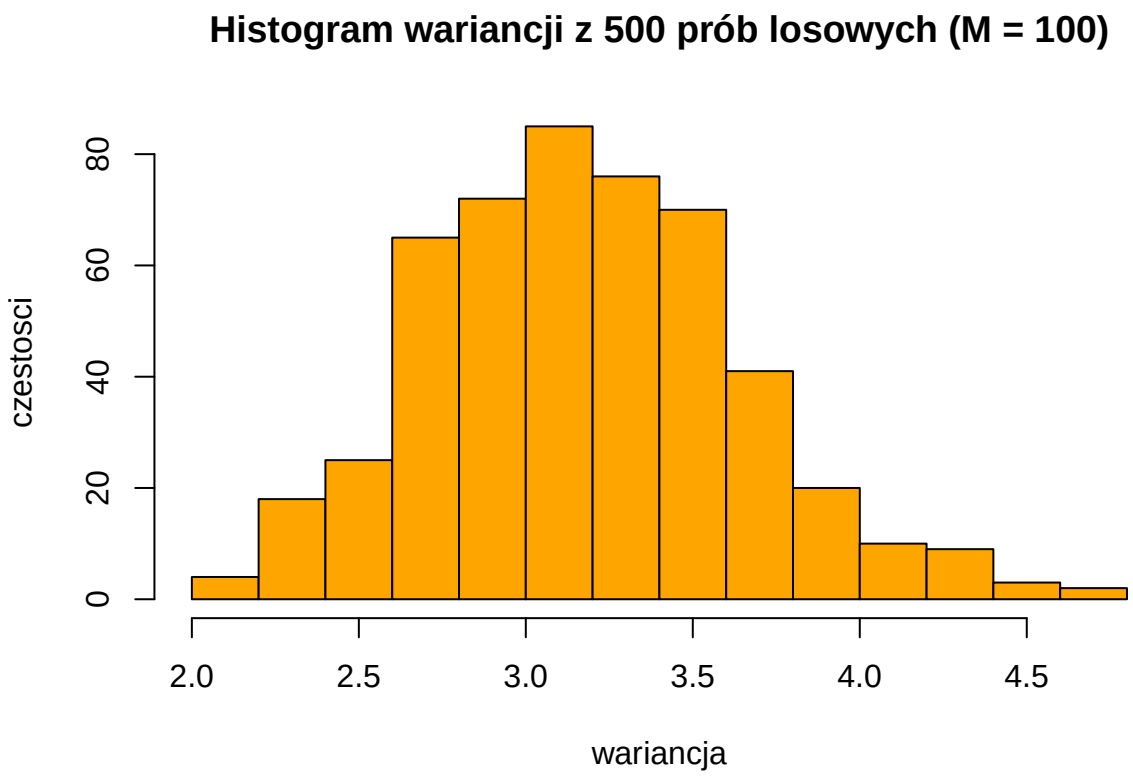
mm = replicate(500, mean(rbinom(M, 20, 0.8)))

Rozwiązanie

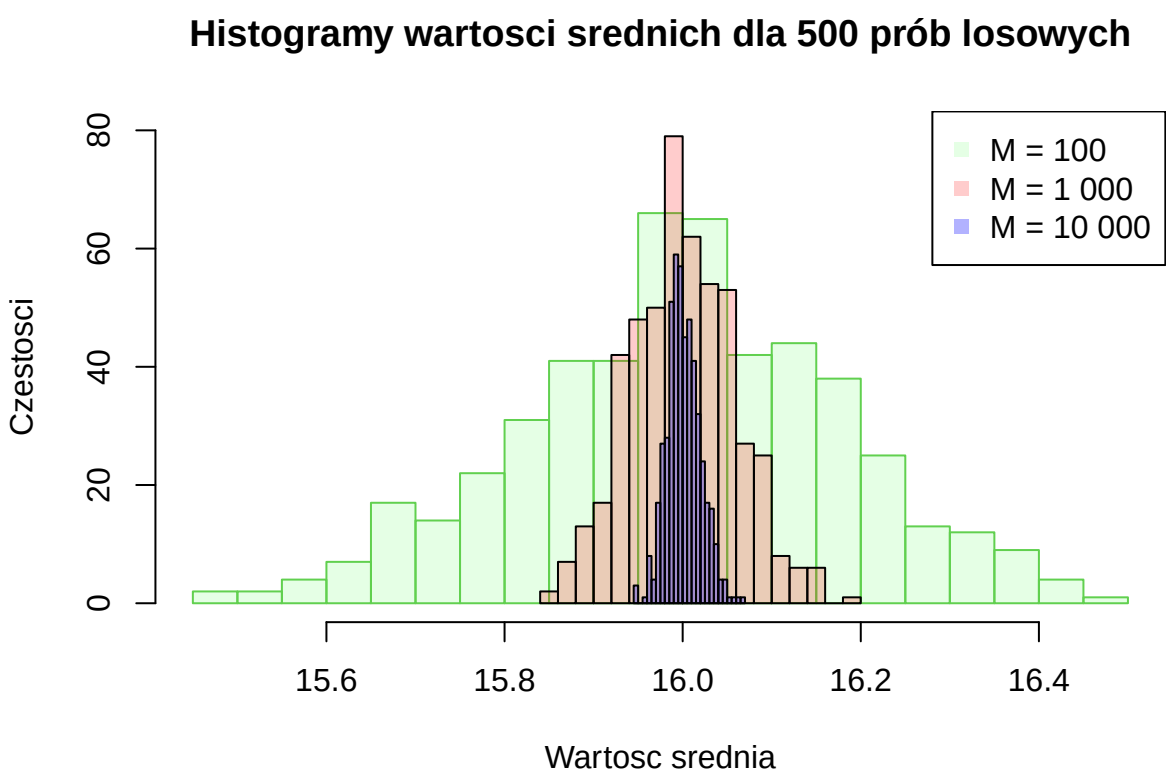
```
mm100 = replicate(500, rbinom(100, 20, 0.8))

average100 = apply(mm100, 2, mean)
variance100 = apply(mm100, 2, var)

hist(average100, col = 'orange', xlab = 'średnia próby', ylab = 'częstości', breaks = 20,
     main = 'Histogram wartości średnich z 500 prób losowych (M = 100)')
```



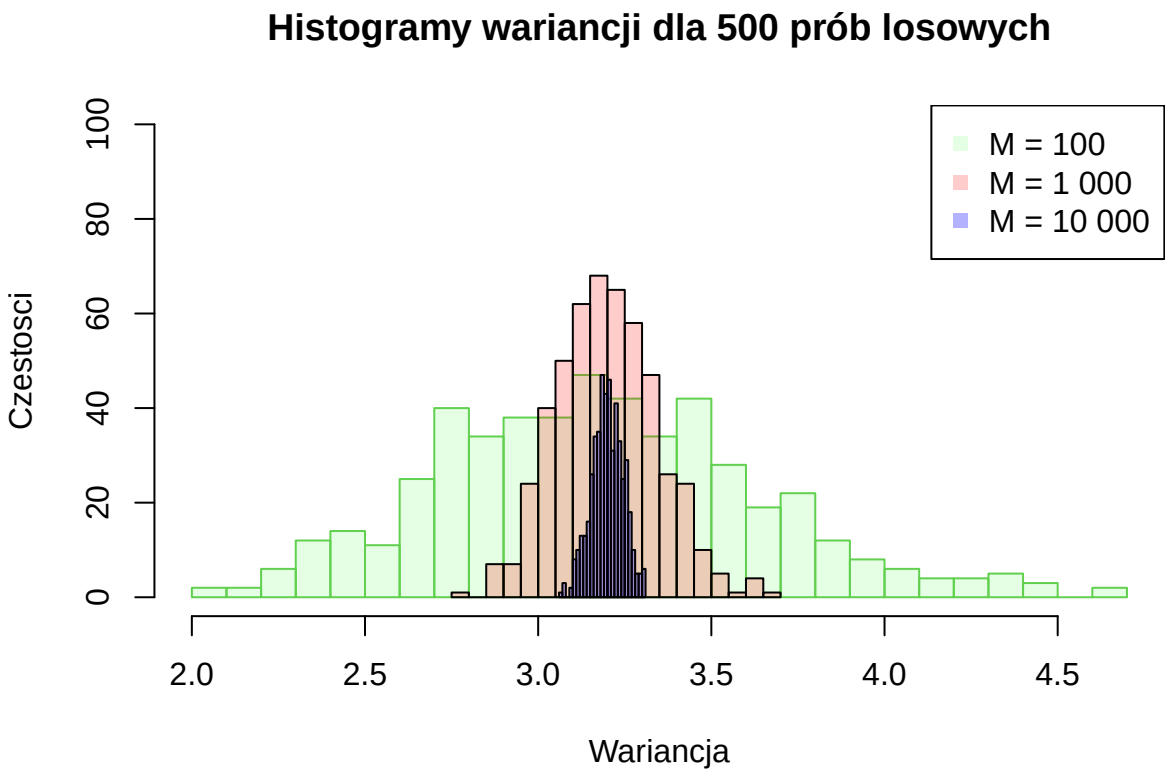

```
hist(variance100, col = 'orange', xlab = 'wariancja', ylab = 'częstości',
     main = 'Histogram wariancji z 500 prób losowych (M = 100)')
```




```
mm1000 = replicate(500, rbinom(1000, 20, 0.8))
mm10000 = replicate(500, rbinom(10000, 20, 0.8))

average1000 = apply(mm1000, 2, mean)
variance1000 = apply(mm1000, 2, var)

average10000 = apply(mm10000, 2, mean)
variance10000 = apply(mm10000, 2, var)

col_100 = rgb(green = 1, red = 0, blue = 0,  alpha = 0.1)
col_1000 = rgb(green = 0, red = 1, blue = 0,  alpha = 0.2)
col_10000 = rgb(green = 0, red = 0, blue = 1,  alpha = 0.3)

hist(average100, breaks = 20, col = col_100, border = 3, main = 'Histogramy wartości średnich dla 500 p

hist(average1000, breaks = 20, add = TRUE, col = col_1000)

hist(average10000, breaks = 20, add = TRUE, col = col_10000)

legend('topright', c('M = 100', 'M = 1 000', 'M = 10 000'), pch = 15, col = c(col_100, col_1000, col_10
```




```
hist(variance100, breaks = 20, col = col_100, border = 3, main = 'Histogramy wariancji dla 500 prób los

hist(variance1000, breaks = 20, add = TRUE, col = col_1000)

hist(variance10000, breaks = 20, add = TRUE, col = col_10000)

legend('topright', c('M = 100', 'M = 1 000', 'M = 10 000'), pch = 15, col = c(col_100, col_1000, col_10
```



Otrzymane wyniki są bardziej rozproszone wzduż osi X w przypadku mniejszej liczby próbkek. Zwiększenie liczby próbek powoduje, że praktyznie wszystkie otrzymane wartości są skoncentrowane wokoł wartości teoretycznych, obliczonych w poprzednim zadaniu.