

ADPS 2025Z — Laboratorium 1 (rozwiązania)

Krzysztof Symoniuk

Zadanie 1 (1 pkt)

Treść zadania

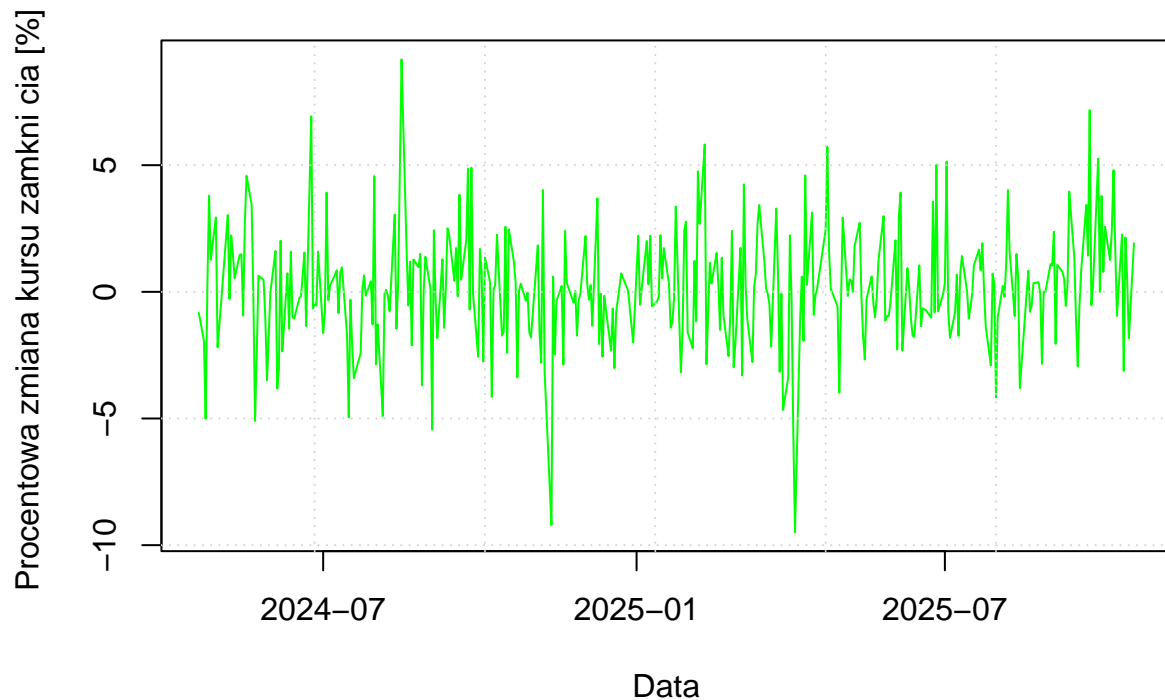
Dla danych z ostatnich 18 miesięcy dotyczących wybranych dwóch spółek giełdowych:

- 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.

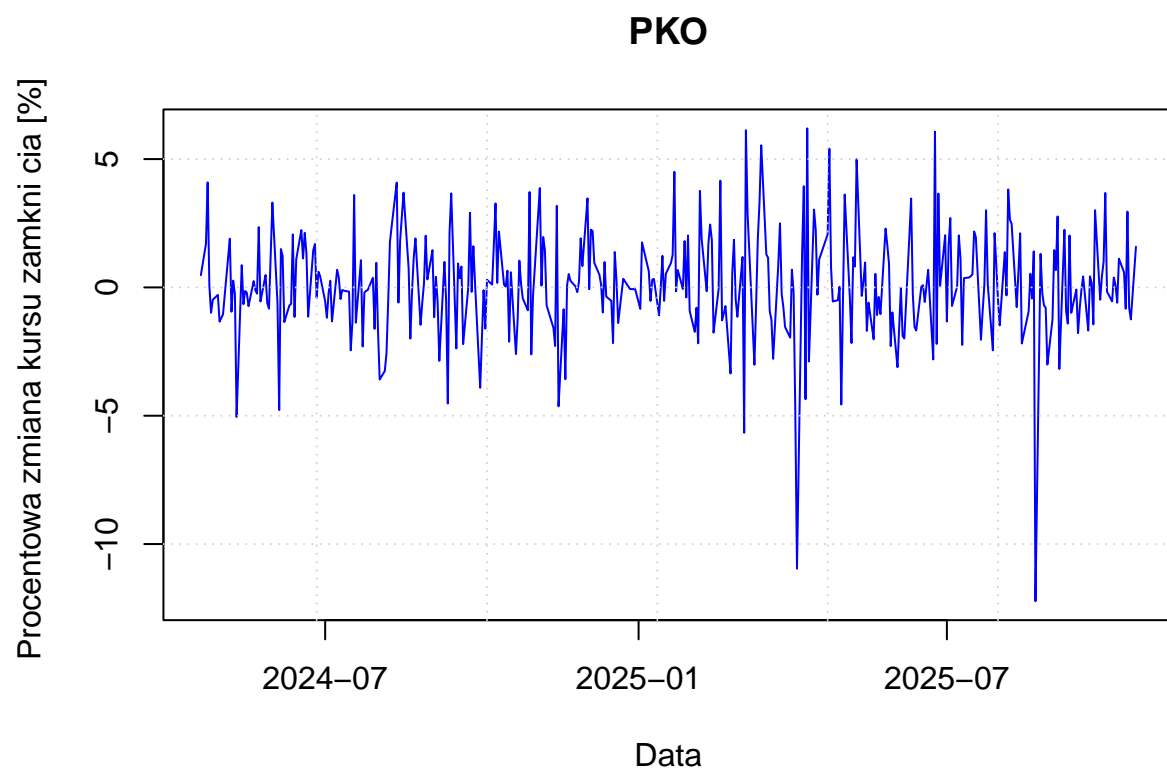
Rozwiązanie

```
# https://stooq.pl/q/d/l/?s=kgh&i=d
Ticket_KGH = 'KGH'
webLink = paste0('https://stooq.pl/q/d/l/?s=', Ticket_KGH, '&i=d')
fileName = paste0(Ticket_KGH, '.csv')
if(!file.exists(fileName)) {
  download.file(webLink, fileName)
}
df_KGH = read.csv('KGH.csv')
df_KGH$Data = as.Date(df_KGH$Data)
df_KGH$Zamkniecie_zm = with(df_KGH, c(NA, 100*diff(Zamkniecie)/Zamkniecie[-length(Zamkniecie)]))
df_KGH_18m = df_KGH[which(df_KGH$Data >= '2024-04-19' & df_KGH$Data <= Sys.Date()),]
plot(Zamkniecie_zm ~ Data, df_KGH_18m, type = 'l', col = 'green', xlab = 'Data',
ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'KGHM')
grid()
```

KGHM

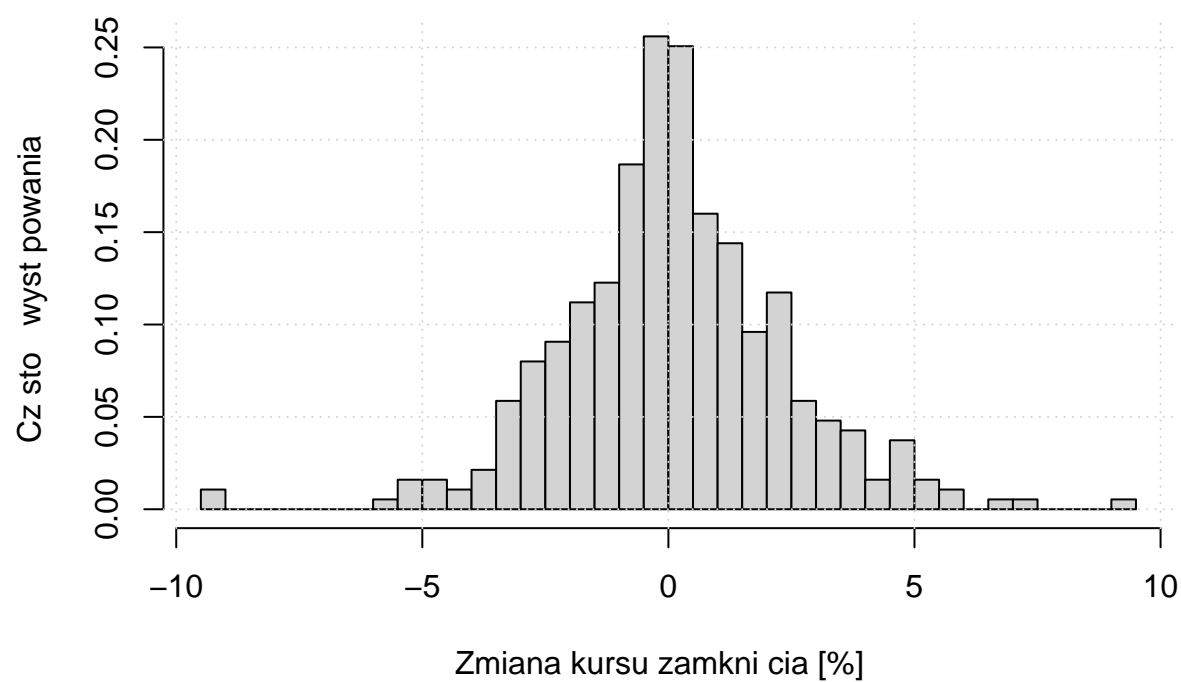


```
# https://stoq.pl/q/d/l/?s=kgh&i=d
Ticket_PK0 = 'PK0'
webLink = paste0('https://stoq.pl/q/d/l/?s=', Ticket_PK0, '&i=d')
fileName = paste0(Ticket_PK0, '.csv')
if(!file.exists(fileName)) {
  download.file(webLink, fileName)
}
df_PK0 = read.csv('PK0.csv')
df_PK0$Data = as.Date(df_PK0$Data)
df_PK0$Zamkniecie_zm = with(df_PK0, c(NA, 100*diff(Zamkniecie)/Zamkniecie[-length(Zamkniecie)]))
df_PK0_18m = df_PK0[which(df_PK0$Data >= '2024-04-19' & df_PK0$Data <= Sys.Date()),]
plot(Zamkniecie_zm ~ Data, df_PK0_18m, type = 'l', col = 'blue', xlab = 'Data',
ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'PK0')
grid()
```



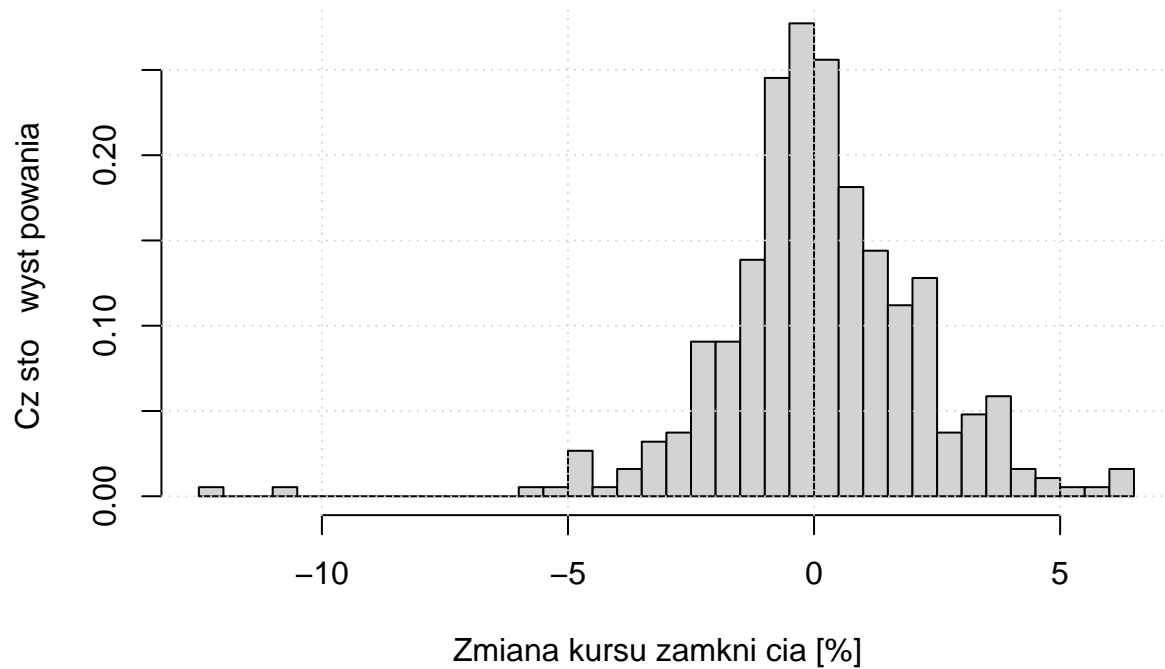
```
hist(df_KGH_18m$Zamknienie_zm, breaks = 50, prob = T,  
xlab = 'Zmiana kursu zamknięcia [%] ',  
ylab = 'Częstość występowania',  
main = paste('Histogram procentowych zmian kursu', 'KGHM') )  
grid()
```

Histogram procentowych zmian kursu KGHM

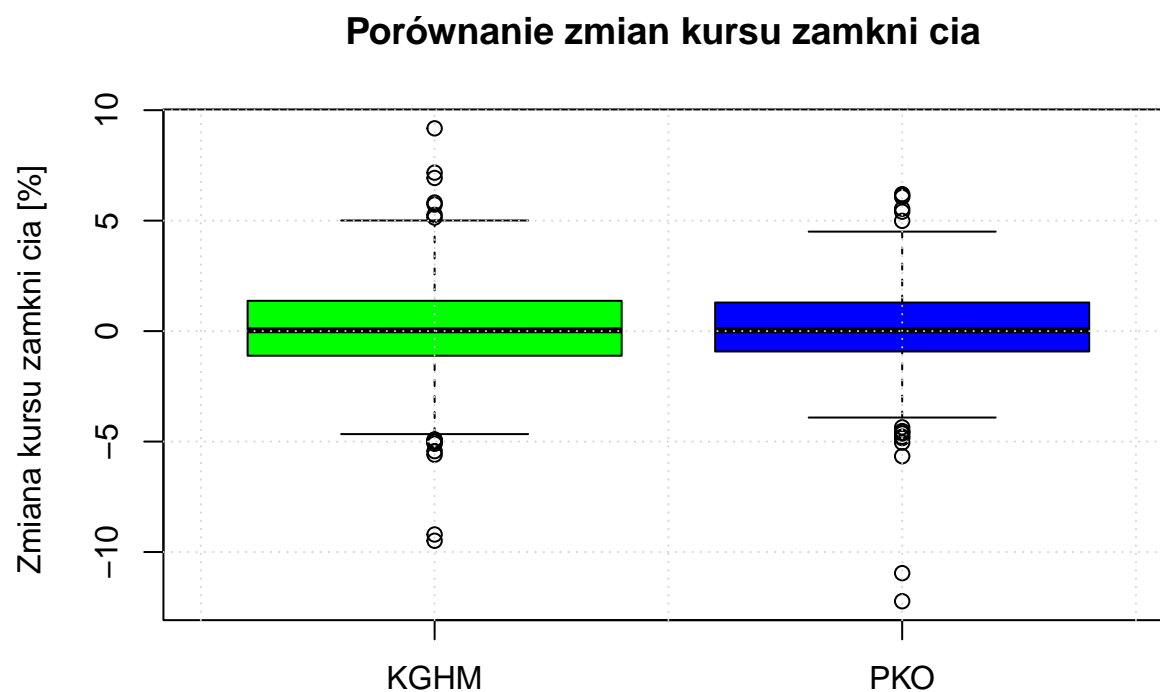


```
hist(df_PKO_18m$Zamkniecie_zm, breaks = 50, prob = T,  
xlab = 'Zmiana kursu zamknięcia [%] ',  
ylab = 'Częstość występowania',  
main = paste('Histogram procentowych zmian kursu', 'PKO') )  
grid()
```

Histogram procentowych zmian kursu PKO



```
boxplot(df_KGH_18m$Zamkniecie_zm,  
        df_PKO_18m$Zamkniecie_zm,  
        names = c("KGHM", "PKO"),  
        col = c("green", "blue"),  
        ylab = "Zmiana kursu zamknięcia [%]",  
        main = "Porównanie zmian kursu zamknięcia")  
grid()
```



Zadanie 2 (1,5 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

```
Sys.setlocale("LC_TIME", "pl_PL.UTF-8")
```

```
## [1] "pl_PL.UTF-8"
```

```

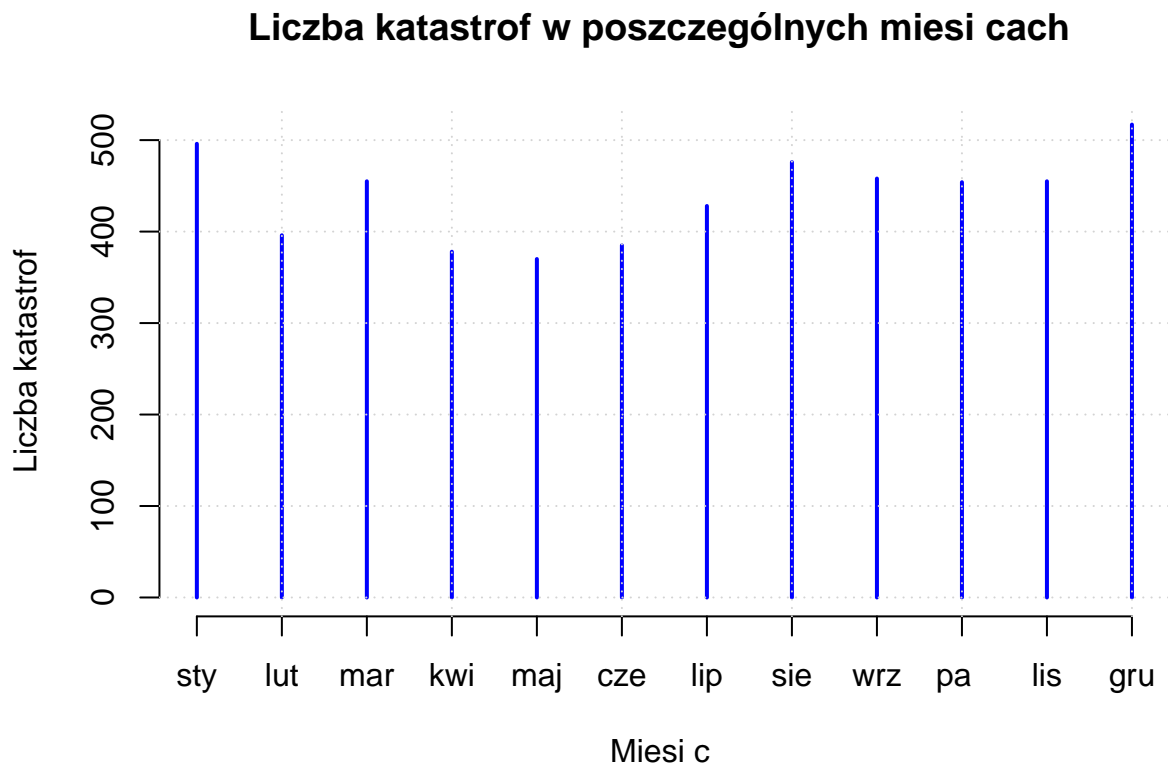
kat = read.csv('crashes.csv')
# Dodanie do danych kolumny z miesiącem:
kat$Month = strptime(as.Date(kat$Date, '%m/%d/%Y'), '%b')

miesiace_pl <- c("sty", "lut", "mar", "kwi", "maj", "cze",
                 "lip", "sie", "wrz", "paź", "lis", "gru")

kat$Month <- factor(kat$Month, levels = miesiace_pl, ordered = TRUE)

# Wykres liczby wypadków w poszczególnych miesiącach:
plot(table(kat$Month), type = 'h', col = 'blue', xlab = 'Miesiąc',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych miesiącach' )
grid()

```

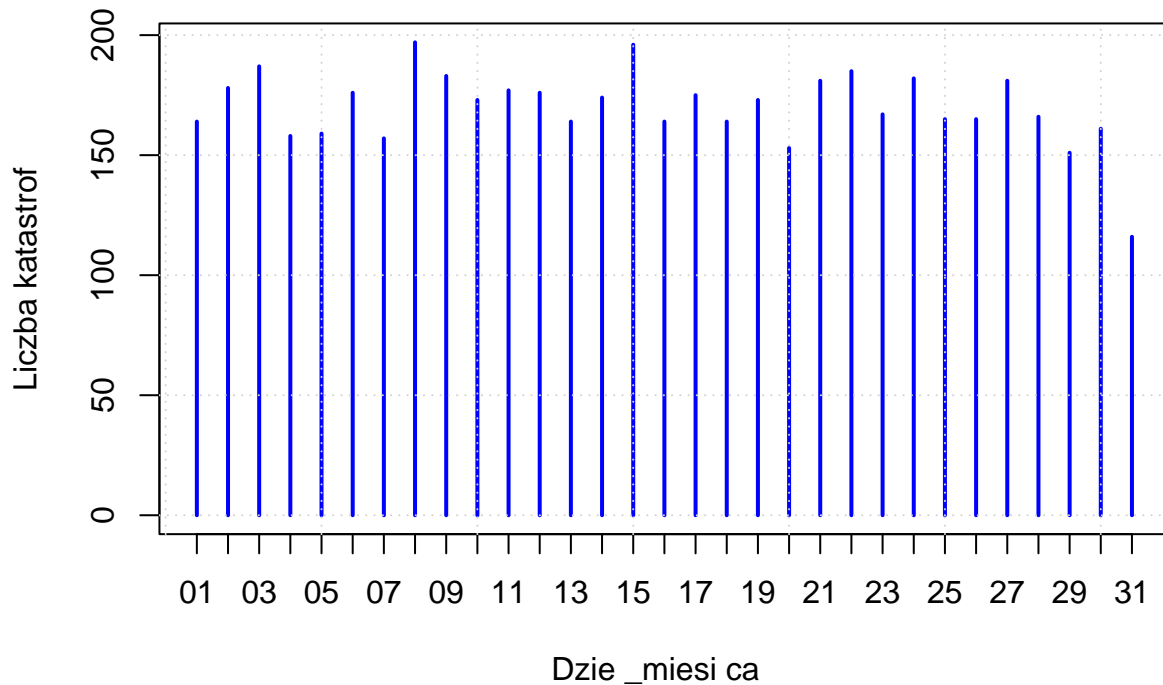


```

# Dodanie do danych kolumny z dniem:
kat$Day = strptime(as.Date(kat$Date, '%m/%d/%Y'), '%d')
# Wykres liczby wypadków w danym dniu miesiąca:
plot(table(kat$Day), type = 'h', col = 'blue', xlab = 'Dzień_miesiąca',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w poszczególnych dniach miesiąca' )
grid()

```

Liczba katastrof w poszczególnych dniach miesiąca



```
try(Sys.setlocale("LC_TIME", "pl_PL.UTF-8"), silent = TRUE)
```

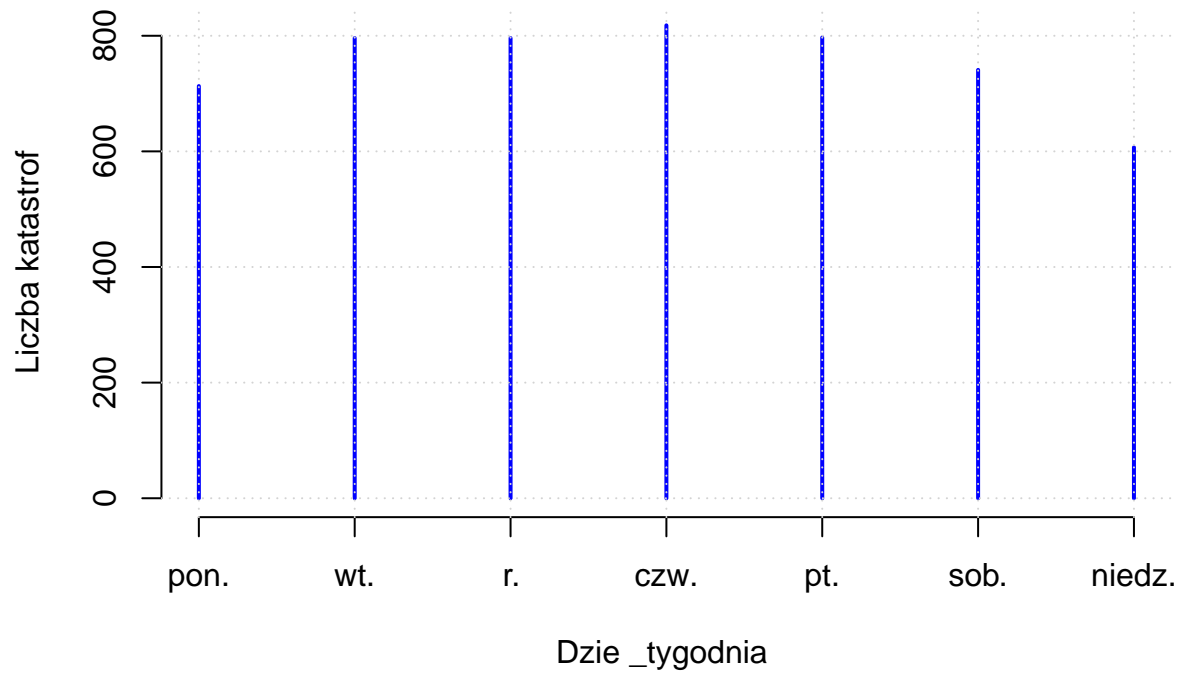
```
## [1] "pl_PL.UTF-8"
```

```
# Dodanie do danych kolumny z dniem tygodnia:
kat$Weekday <- weekdays(as.Date(kat$Date, '%m/%d/%Y'), abbreviate = TRUE)

# Ustawienie właściwej kolejności dni tygodnia (zgodnie z Twoimi nazwami)
kat$Weekday <- factor(kat$Weekday,
                      #levels = c("pon", "wto", "śro", "czw", "pią", "sob", "nie"))
                      levels = c("pon.", "wt.", "śr.", "czw.", "pt.", "sob.", "niedz."))

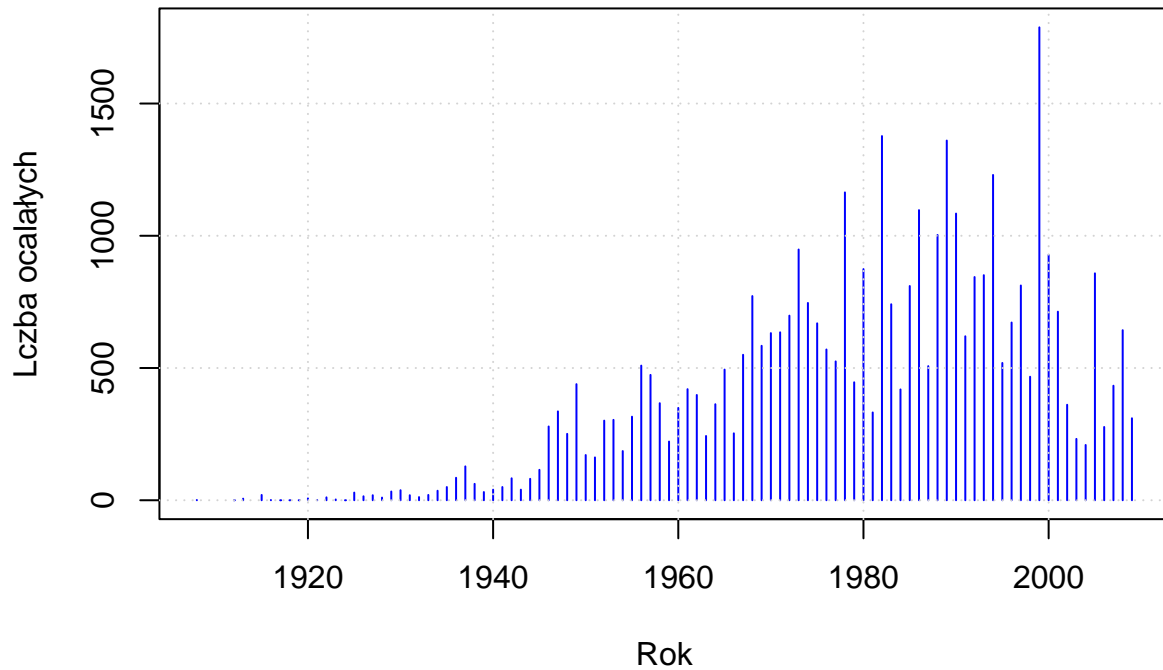
# Wykres liczby wypadków w danym dniu tygodnia:
plot(table(kat$Weekday),
     type = 'h',
     col = 'blue',
     xlab = 'Dzień_tygodnia',
     ylab = 'Liczba katastrof',
     main = 'Liczba katastrof w poszczególnych dniach tygodnia')
grid()
```


Liczba katastrof w poszczególnych dniach tygodnia



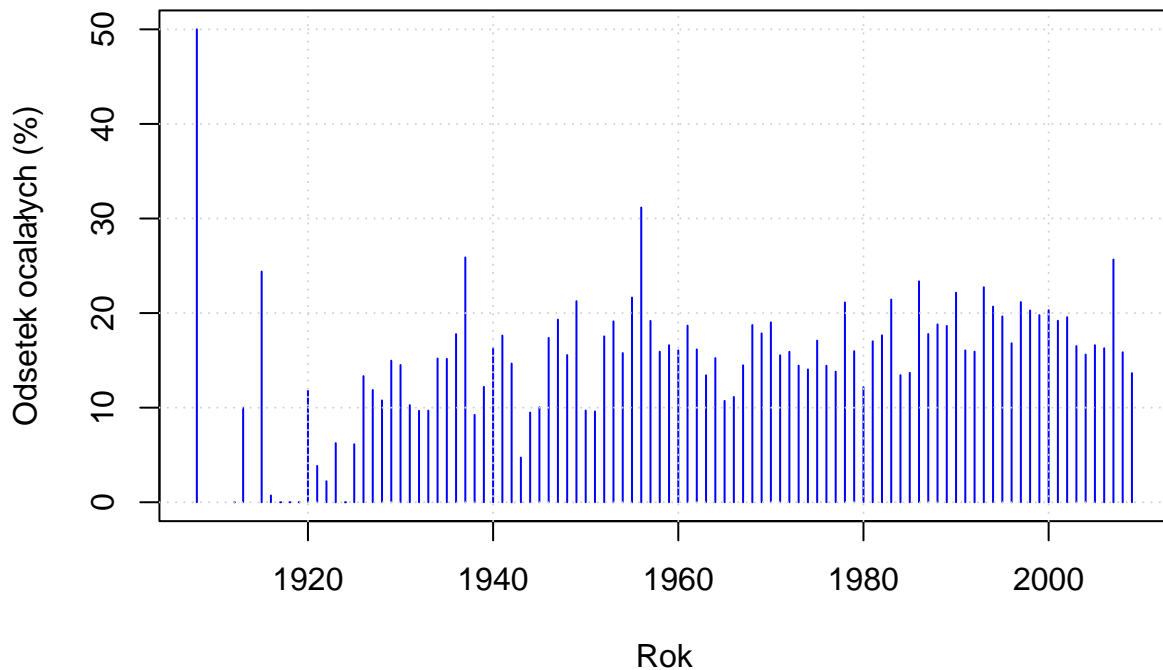
```
kat$Year = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%Y')
# Agregacja danych po latach:
Ocaleni = aggregate((Aboard - Fatalities) ~ Year, kat, FUN = sum)
# Wykres:
plot(Ocaleni, type = 'h', col = 'blue', xlab = 'Rok',
ylab = 'Liczba ocalałych', main = 'Liczba osób, które przeżyły katastrofy' )
grid()
```

Liczba osób, które przeżyły katastrofy



```
Odsetek_ocalonych = aggregate(((Aboard - Fatalities) / Aboard * 100) ~ Year, kat, FUN = mean)
plot(Odsetek_ocalonych, type = 'h', col = 'blue',
     xlab = 'Rok', ylab = 'Odsetek ocalałych (%)',
     main = 'Odsetek osób, które przeżyły katastrofy'
)
grid()
```

Odsetek osób, które przeżyły katastrofy



Zadanie 3 (1 pkt)

Treść zadania

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

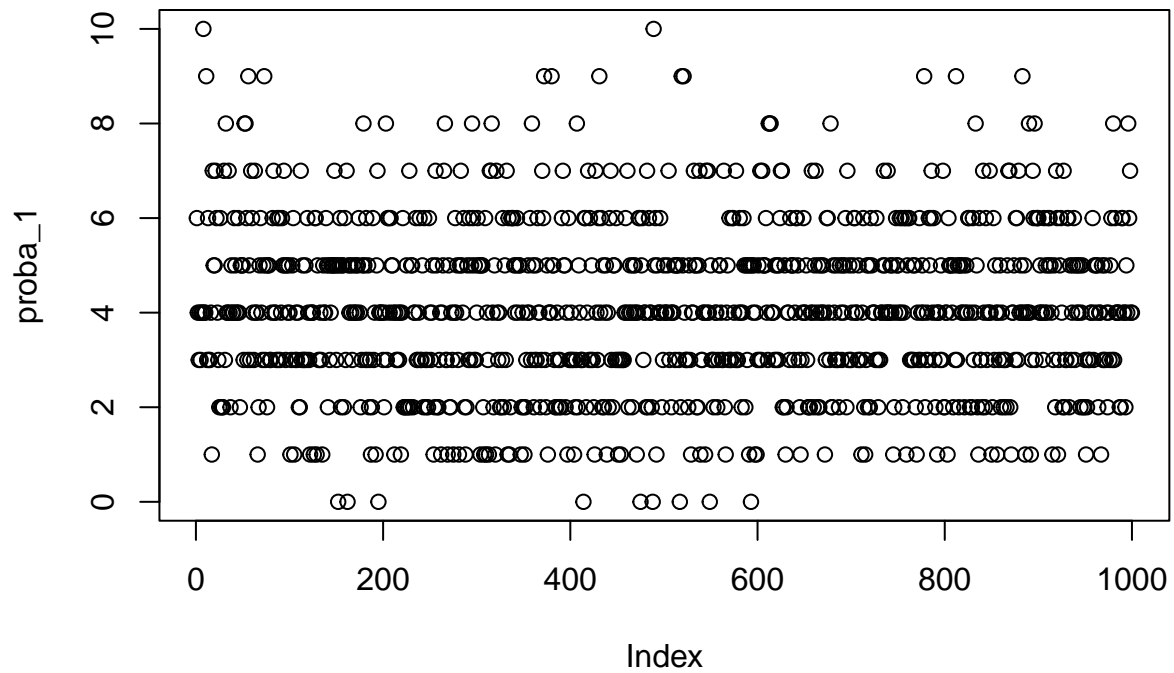
- `Binom(20,0.2)`
- `Binom(20,0.8)`

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

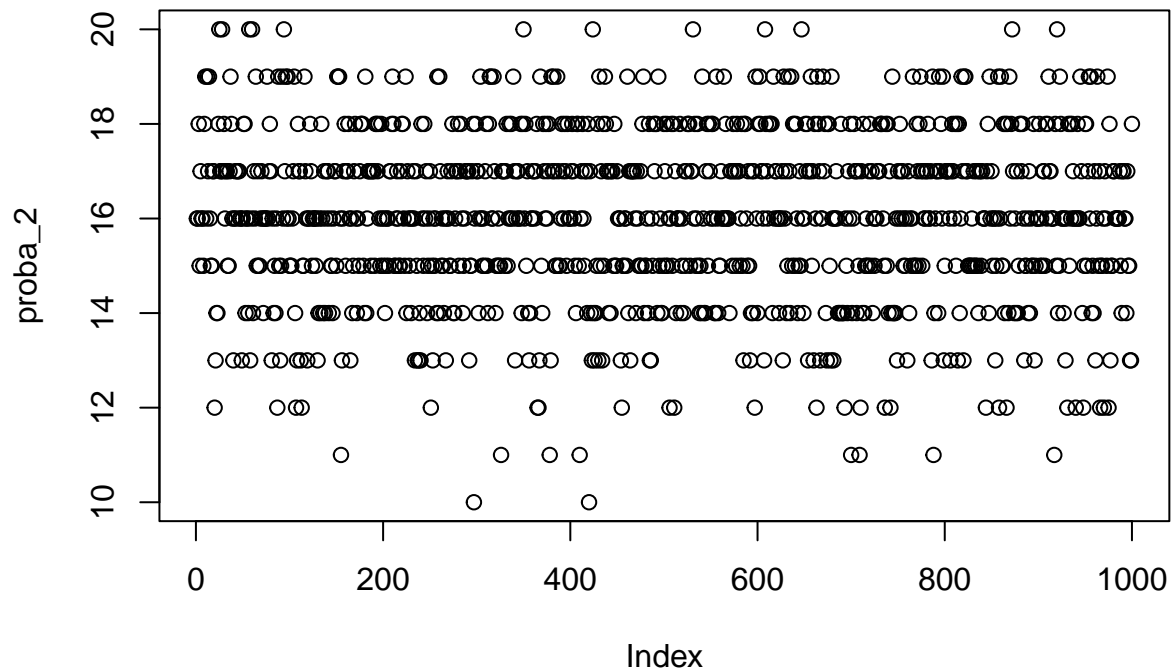
2. Dla każdego z rozkładów narysuj na jednym rysunku empiryczne i teoretyczne (użyj funkcji `dbinom`) funkcje prawdopodobieństwa, a na drugim rysunku empiryczne i teoretyczne (użyj funkcji `pbinom`) dystrybuanty. W obu przypadkach wyskaluj oś odciętych od 0 do 20.

Rozwiązanie

```
proba_1 = rbinom(1000, 20, 0.2)
plot(proba_1)
```



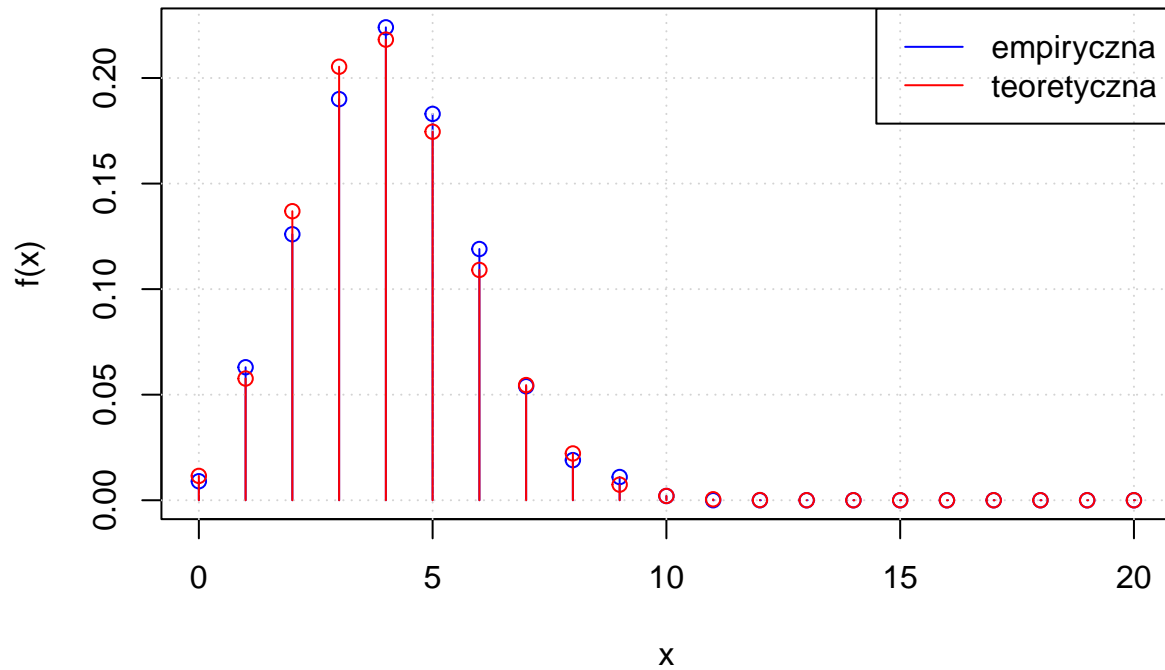
```
proba_2 = rbinom(1000, 20, 0.8)
plot(proba_2)
```



```
M = 1000
n = 20
Arg = 0:n
Freq_1 = as.numeric(table(factor(proba_1, levels = Arg))) / M
teor_1 = dbinom(Arg, n, 0.2)

plot(Freq_1 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = paste0('Funkcja prawdopodobieństwa rozkładu_1 dla M = ', M))
grid()
points(Freq_1 ~ Arg, col = 'blue')
lines(teor_1 ~ Arg, type = 'h', col = 'red',
xlab = 'x', ylab = 'f(x)')
points(teor_1 ~ Arg, col = 'red')
legend('topright', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 1)
```

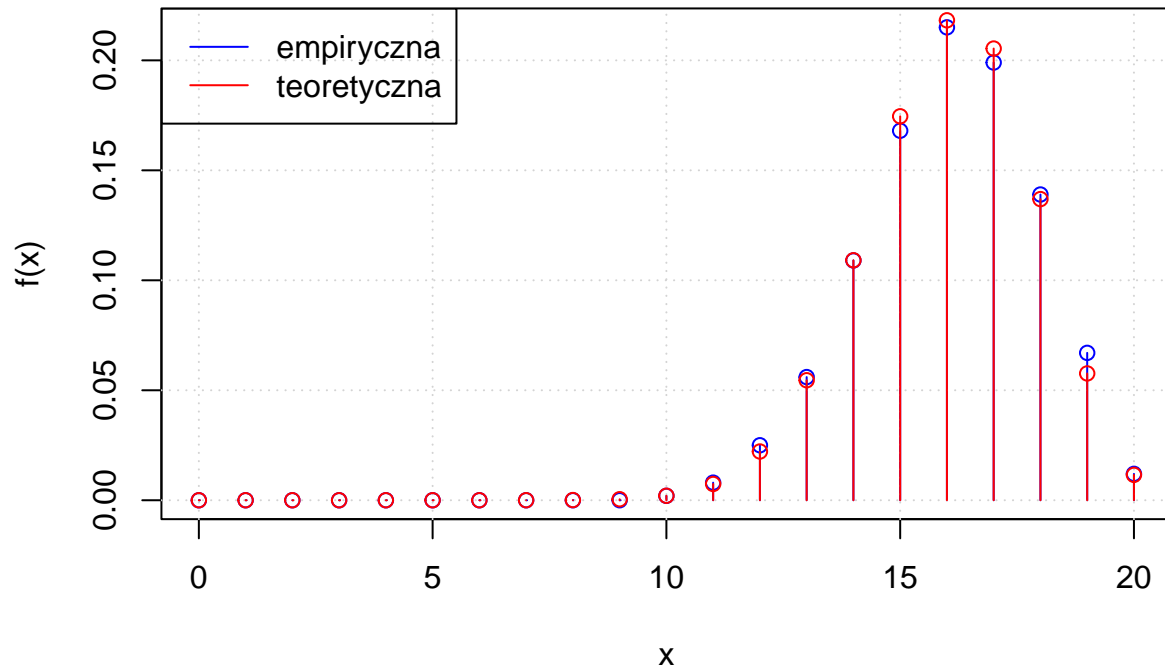
Funkcja prawdopodobieństwa rozkładu_1 dla M = 1000



```
Freq_2 = as.numeric(table(factor(proba_2, levels = Arg))) / M
teor_2 = dbinom(Arg, n, 0.8)

plot(Freq_2 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = paste0('Funkcja prawdopodobieństwa rozkładu_2 dla M = ', M))
grid()
points(Freq_2 ~ Arg, col = 'blue')
lines(teor_2 ~ Arg, type = 'h', col = 'red',
xlab = 'x', ylab = 'f(x)')
points(teor_2 ~ Arg, col = 'red')
legend('topleft', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 1)
```

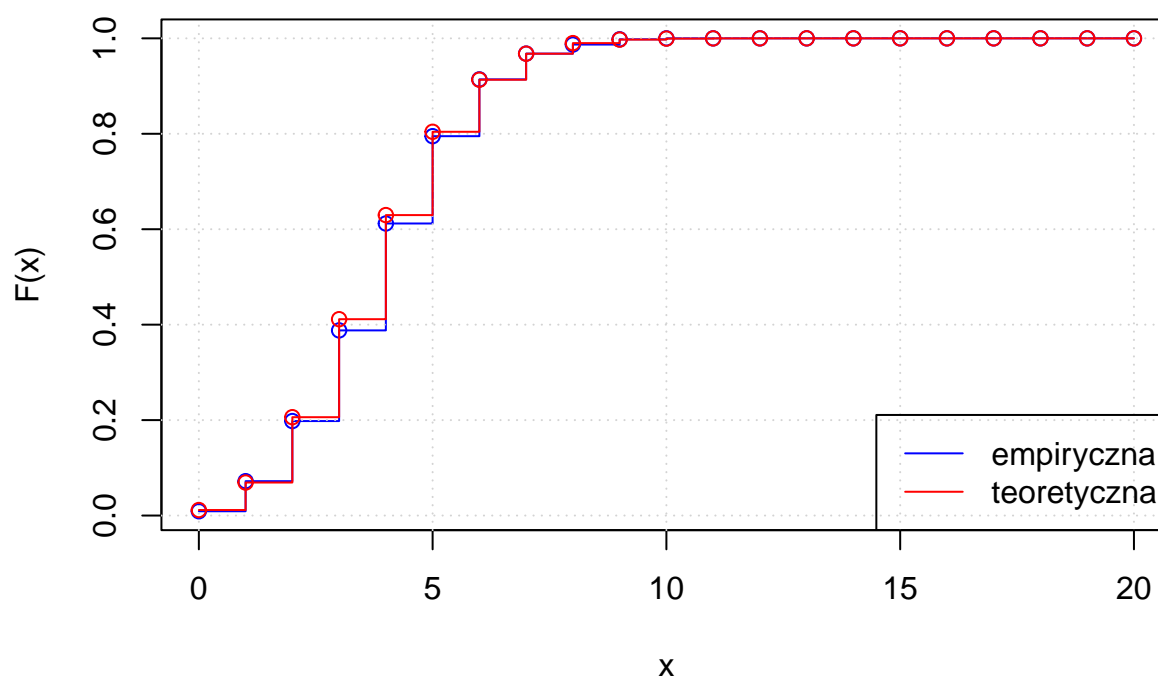
Funkcja prawdopodobieństwa rozkładu₂ dla M = 1000



```

teor_1b = pbinom(Arg, n, 0.2, lower.tail = TRUE, log.p = FALSE)
Freq_1b = cumsum(as.numeric(table(factor(proba_1, levels = Arg)))) / M
plot(Freq_1b ~ Arg, type = 's', col = 'blue',
     xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybucja dla M = ', M))
grid()
points(Freq_1b ~ Arg, col = 'blue')
lines(teor_1b ~ Arg, type = 's', col = 'red',
     xlab = 'x', ylab = 'F(x)')
points(teor_1b ~ Arg, col = 'red')
legend('bottomright', c('empiryczna', 'teoretyczna'),
     col = c('blue', 'red'), lwd = 1)
    
```

Dystrybuanta dla M = 1000

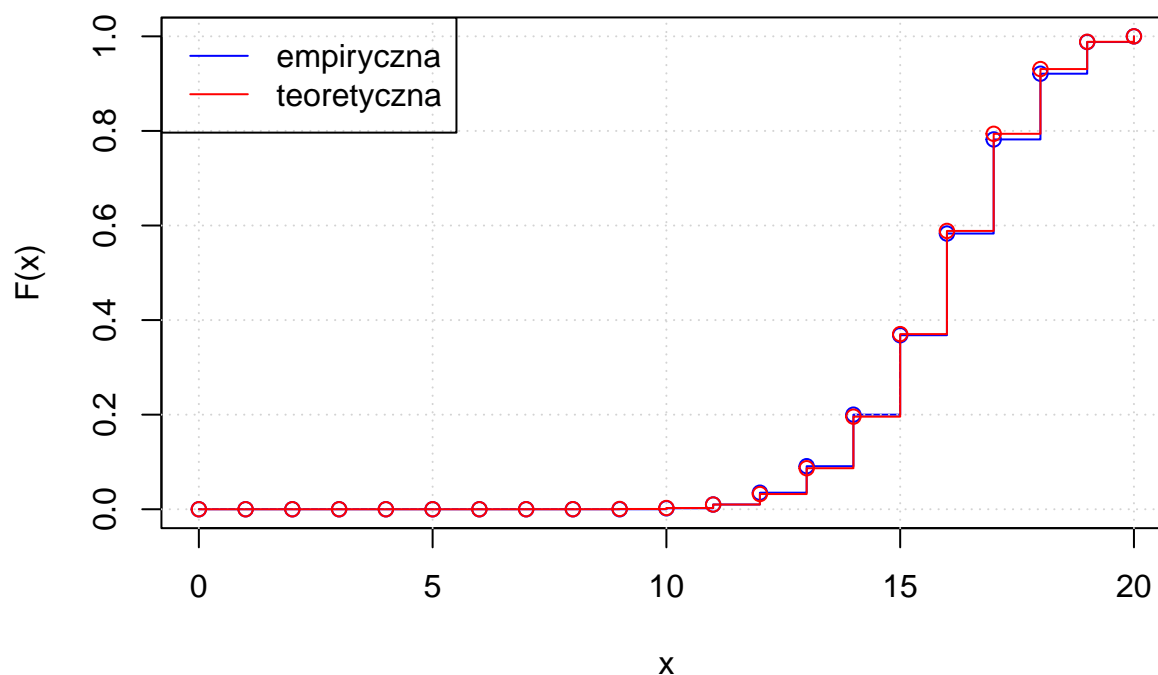


```

teor_2b = pbinom(Arg, n, 0.8, lower.tail = TRUE, log.p = FALSE)
Freq_2b = cumsum(as.numeric(table(factor(proba_2, levels = Arg)))) / M)
plot(Freq_2b ~ Arg, type = 's', col = 'blue',
     xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybuanta dla M = ', M))
grid()
points(Freq_2b ~ Arg, col = 'blue')
lines(teor_2b ~ Arg, type = 's', col = 'red',
     xlab = 'x', ylab = 'F(x)')
points(teor_2b ~ Arg, col = 'red')
legend('topleft', c('empiryczna', 'teoretyczna'),
     col = c('blue', 'red'), lwd = 1)

```


Dystrybuanta dla $M = 1000$



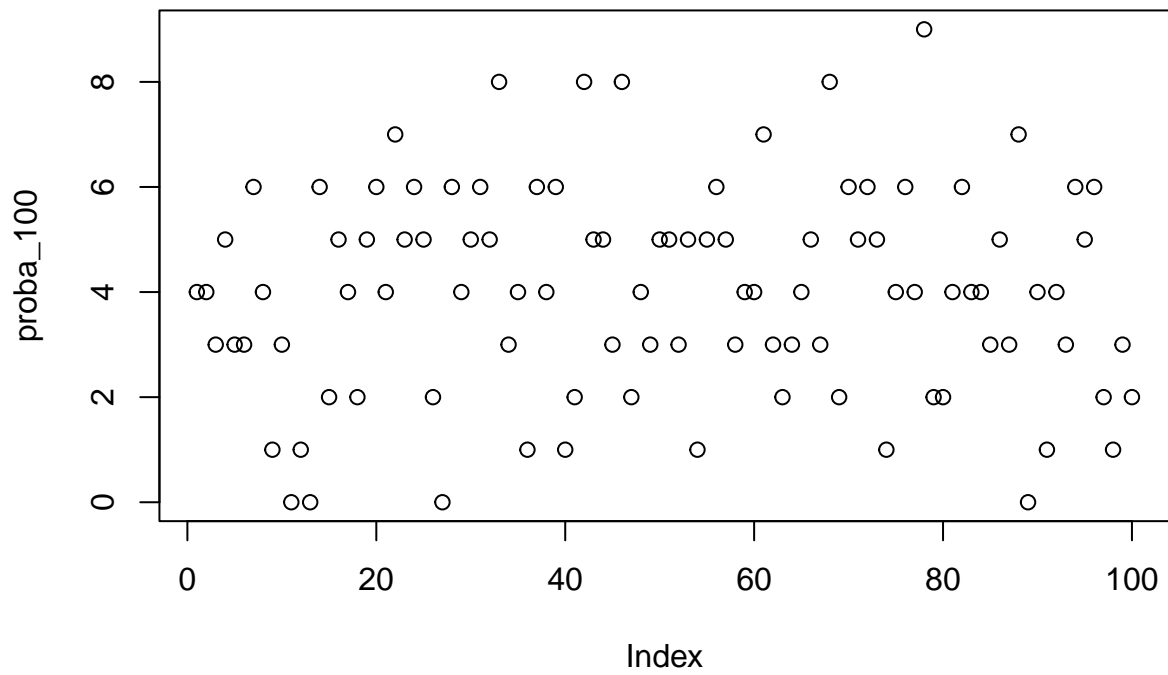
Zadanie 4 (1,5 pkt)

Treść zadania

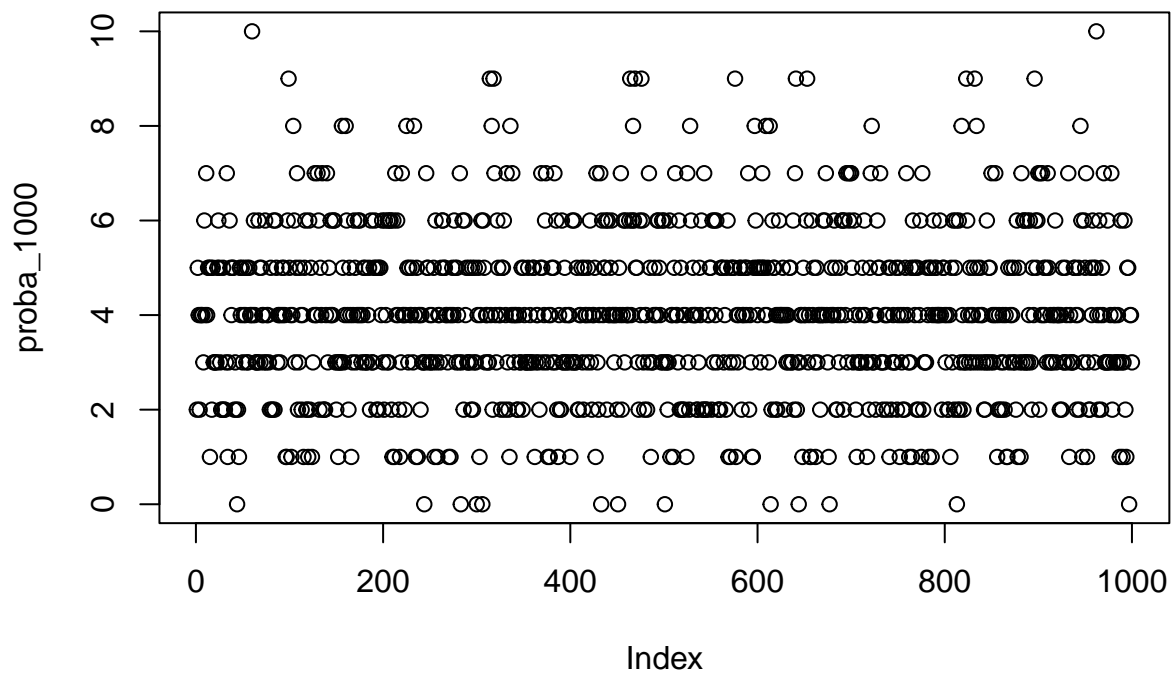
1. Dla rozkładu dwumianowego $\text{Binom}(20, 0.2)$ wygeneruj trzy próby losowe składające się z $M = 100$, 1000 i 10000 próbek.
2. Dla poszczególnych prób wykreśl empiryczne i teoretyczne funkcje prawdopodobieństwa, a także empiryczne i teoretyczne dystrybuanty.
3. We wszystkich przypadkach oblicz empiryczne wartości średnie i wariancje. Porównaj je ze sobą oraz z wartościami teoretycznymi dla rozkładu $\text{Binom}(20, 0.2)$.

Rozwiązanie

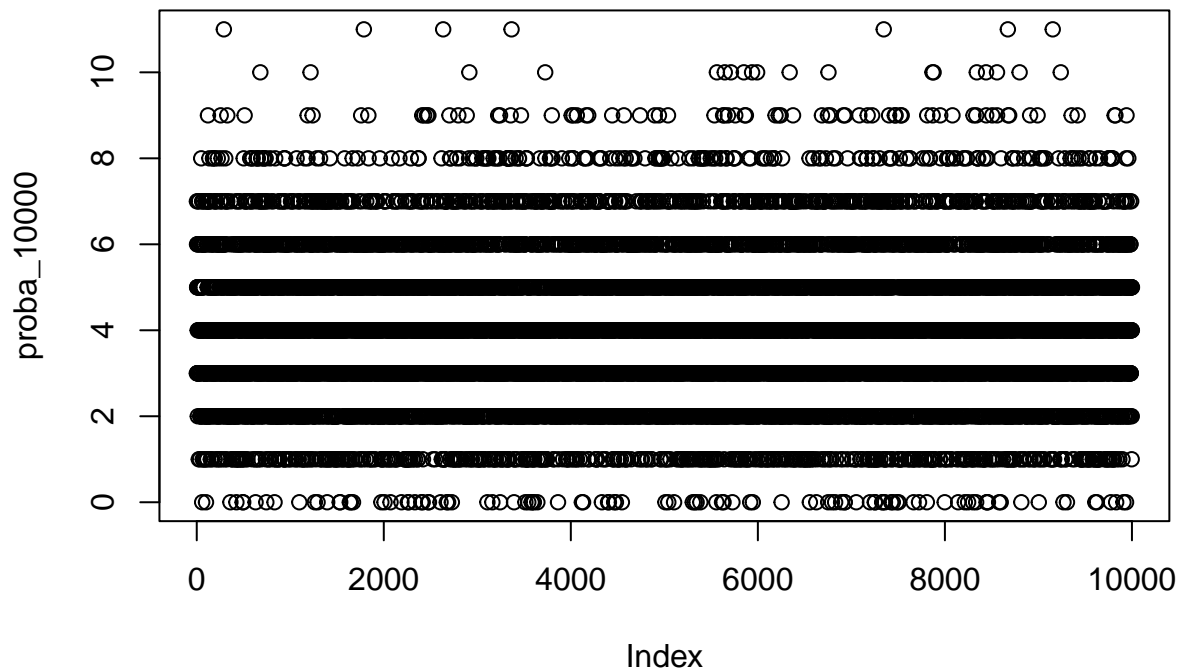
```
proba_100 = rbinom(100, 20, 0.2)
plot(proba_100)
```



```
proba_1000 = rbinom(1000, 20, 0.2)
plot(proba_1000)
```



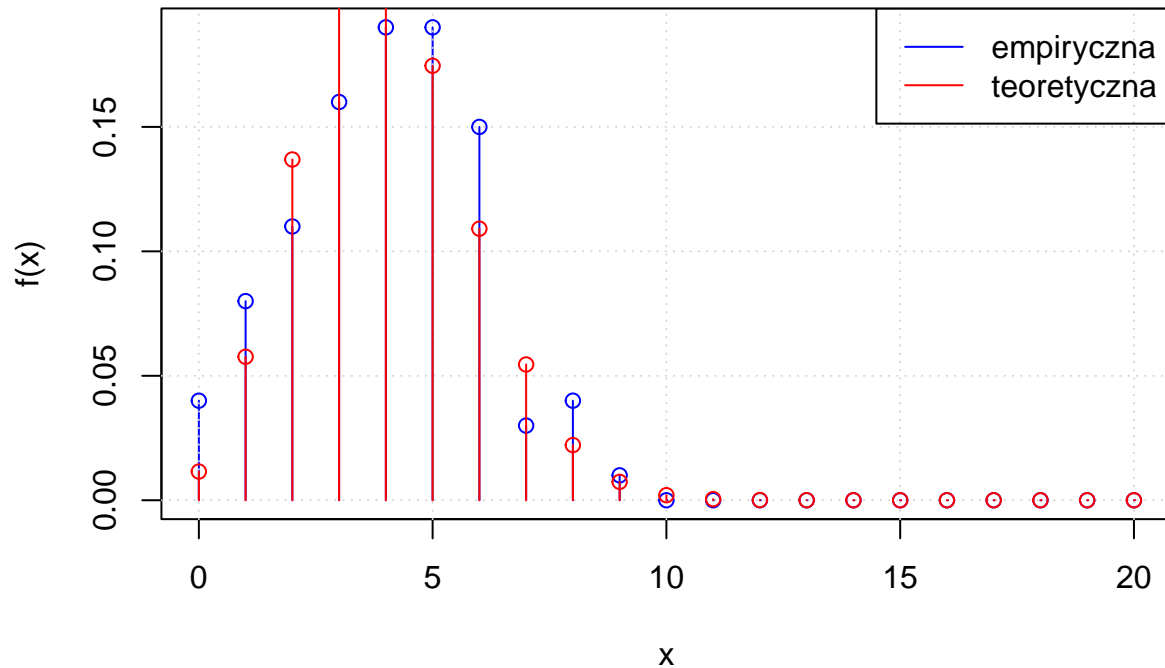
```
proba_10000 = rbinom(10000, 20, 0.2)
plot(proba_10000)
```



```
M_100 = 100
n = 20
Arg = 0:n
Freq_100 = as.numeric(table(factor(proba_100, levels = Arg))) / M_100
teor_100 = dbinom(Arg, n, 0.2)

plot(Freq_100 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = paste0('Funkcja prawdopodobieństwa rozkładu_1 dla M = ', M_100))
grid()
points(Freq_100 ~ Arg, col = 'blue')
lines(teor_100 ~ Arg, type = 'h', col = 'red',
xlab = 'x', ylab = 'f(x)')
points(teor_100 ~ Arg, col = 'red')
legend('topright', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 1)
```

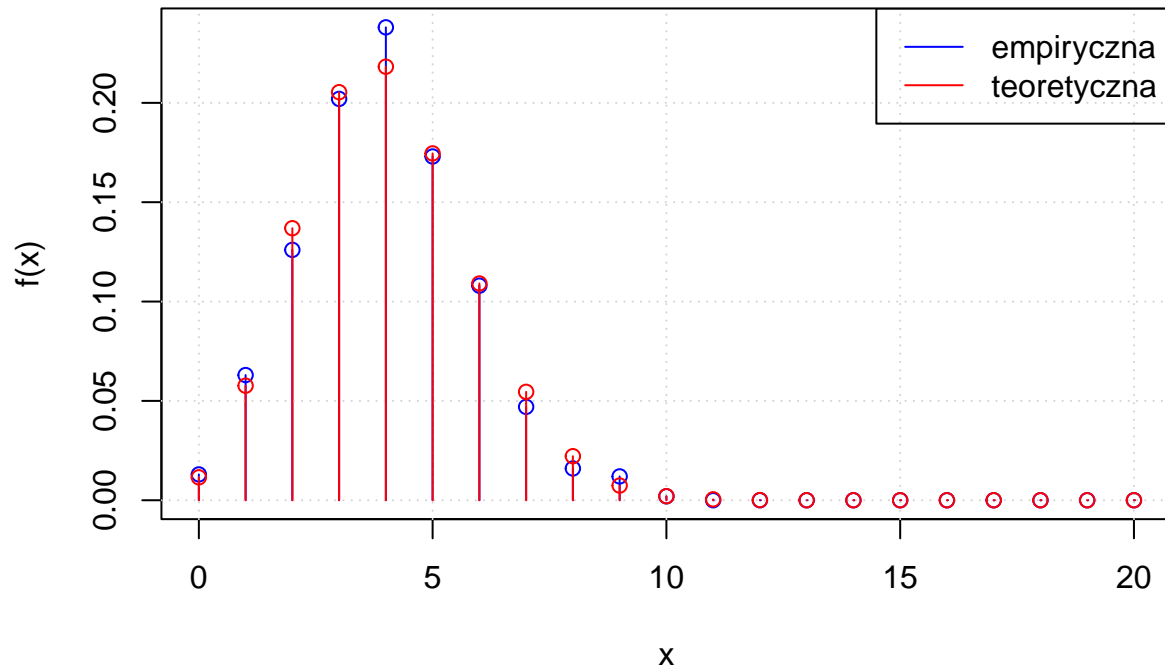
Funkcja prawdopodobieństwa rozkładu_1 dla M = 100



```
M_1000 = 1000
n = 20
Arg = 0:n
Freq_1000 = as.numeric(table(factor(proba_1000, levels = Arg))) / M_1000
teor_1000 = dbinom(Arg, n, 0.2)

plot(Freq_1000 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = paste0('Funkcja prawdopodobieństwa rozkładu_1 dla M = ', M_1000))
grid()
points(Freq_1000 ~ Arg, col = 'blue')
lines(teor_1000 ~ Arg, type = 'h', col = 'red',
xlab = 'x', ylab = 'f(x)')
points(teor_1000 ~ Arg, col = 'red')
legend('topright', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 1)
```

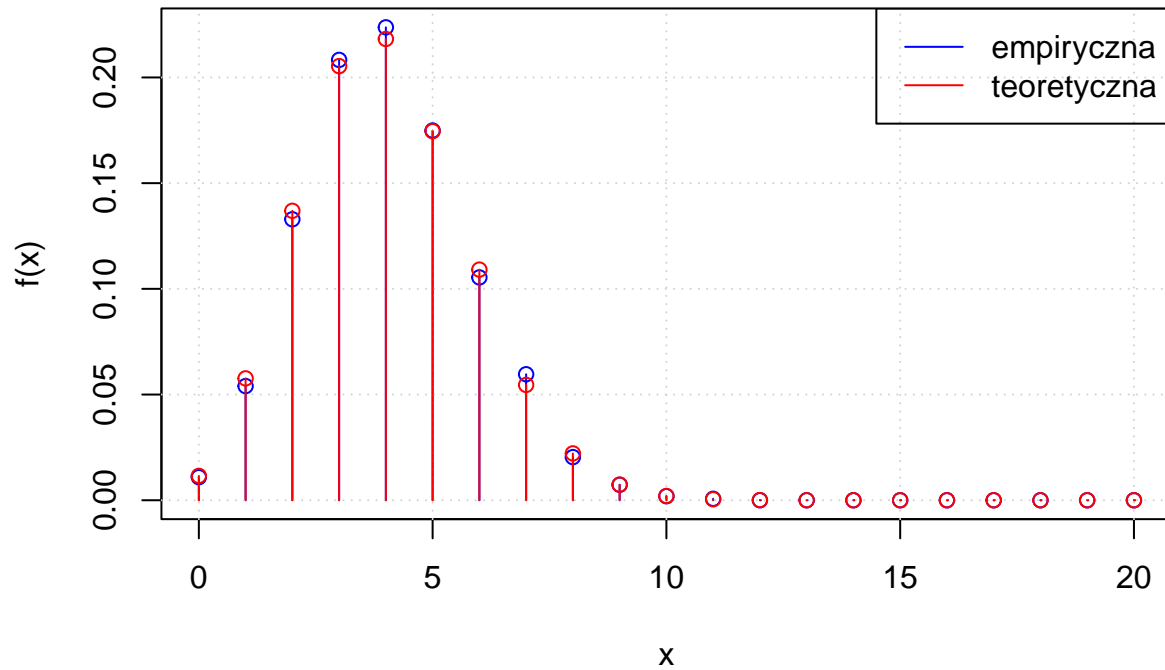
Funkcja prawdopodobieństwa rozkładu_1 dla M = 1000



```
M_10000 = 10000
n = 20
Arg = 0:n
Freq_10000 = as.numeric(table(factor(proba_10000, levels = Arg))) / M_10000
teor_10000 = dbinom(Arg, n, 0.2)

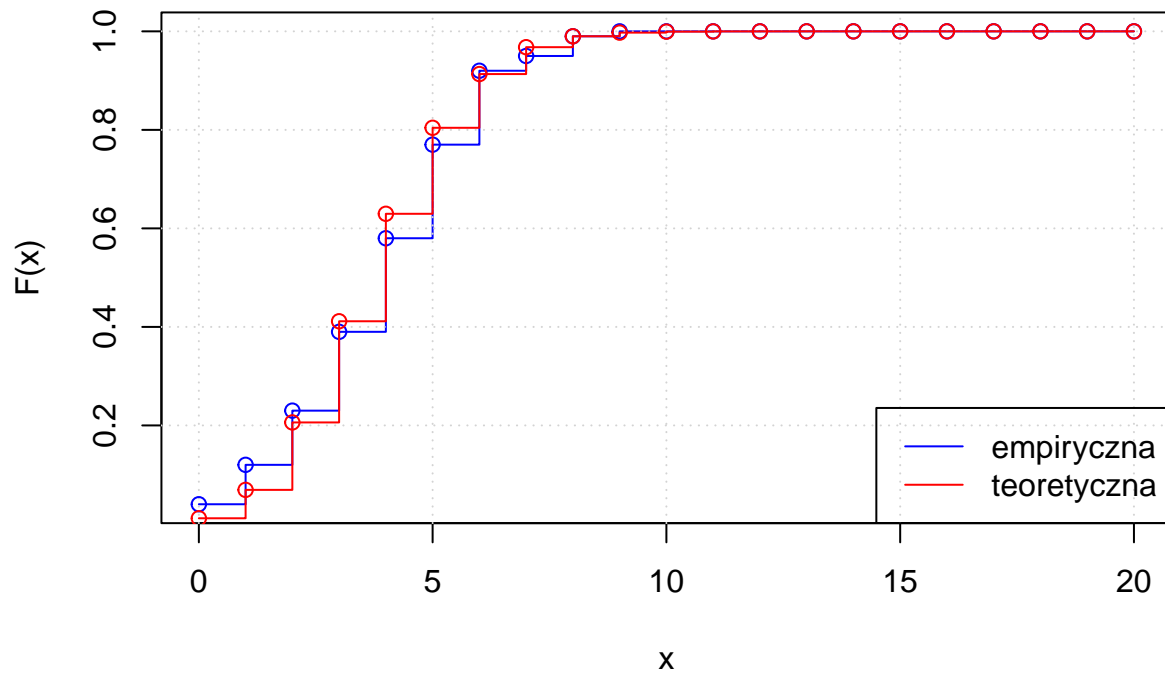
plot(Freq_10000 ~ Arg, type = 'h', col = 'blue', xlab = 'x', ylab = 'f(x)',
main = paste0('Funkcja prawdopodobieństwa rozkładu_1 dla M = ', M_10000))
grid()
points(Freq_10000 ~ Arg, col = 'blue')
lines(teor_10000 ~ Arg, type = 'h', col = 'red',
xlab = 'x', ylab = 'f(x)')
points(teor_10000 ~ Arg, col = 'red')
legend('topright', c('empiryczna', 'teoretyczna'),
col = c('blue', 'red'), lwd = 1)
```

Funkcja prawdopodobieństwa rozkładu_1 dla M = 10000



```
teor_100b = pbinom(Arg, n, 0.2, lower.tail = TRUE, log.p = FALSE)
Freq_100b = cumsum(as.numeric(table(factor(proba_100, levels = Arg)))) / M_100
plot(Freq_100b ~ Arg, type = 's', col = 'blue',
     xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybucja dla M = ', M_100))
grid()
points(Freq_100b ~ Arg, col = 'blue')
lines(teor_100b ~ Arg, type = 's', col = 'red',
     xlab = 'x', ylab = 'F(x)')
points(teor_100b ~ Arg, col = 'red')
legend('bottomright', c('empiryczna', 'teoretyczna'),
     col = c('blue', 'red'), lwd = 1)
```

Dystrybuanta dla M = 100

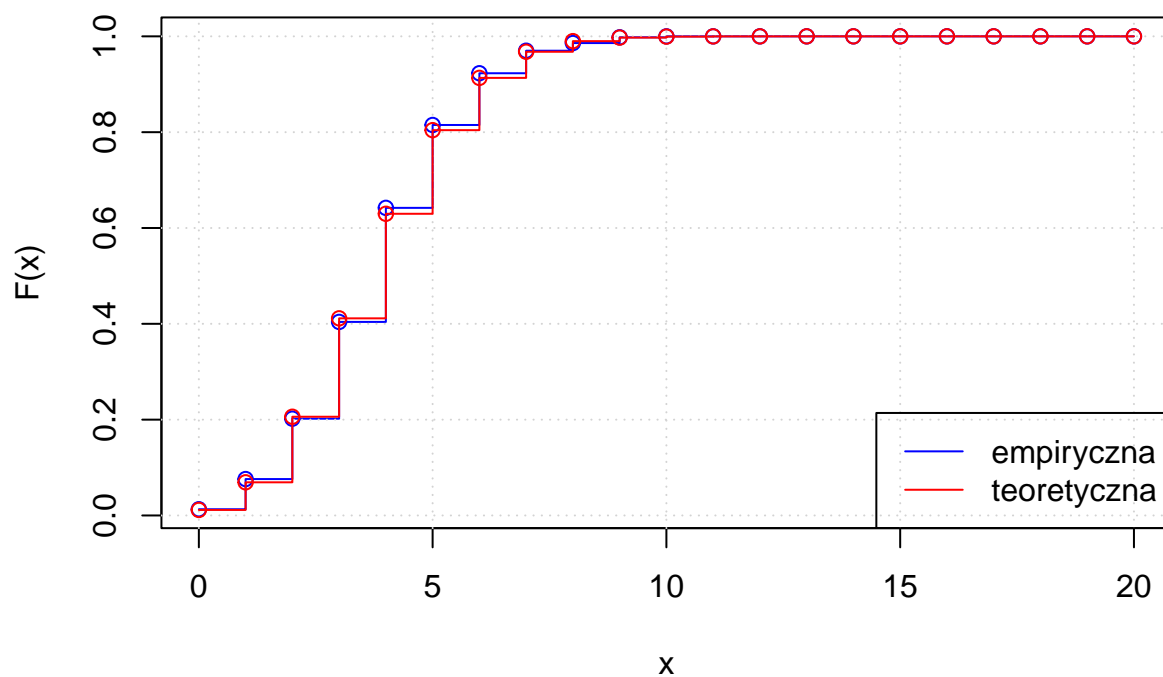


```

teor_1000b = pbinom(Arg, n, 0.2, lower.tail = TRUE, log.p = FALSE)
Freq_1000b = cumsum(as.numeric(table(factor(proba_1000, levels = Arg)))) / M_1000
plot(Freq_1000b ~ Arg, type = 's', col = 'blue',
     xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybuanta dla M = ', M_1000))
grid()
points(Freq_1000b ~ Arg, col = 'blue')
lines(teor_1000b ~ Arg, type = 's', col = 'red',
     xlab = 'x', ylab = 'F(x)')
points(teor_1000b ~ Arg, col = 'red')
legend('bottomright', c('empiryczna', 'teoretyczna'),
     col = c('blue', 'red'), lwd = 1)

```

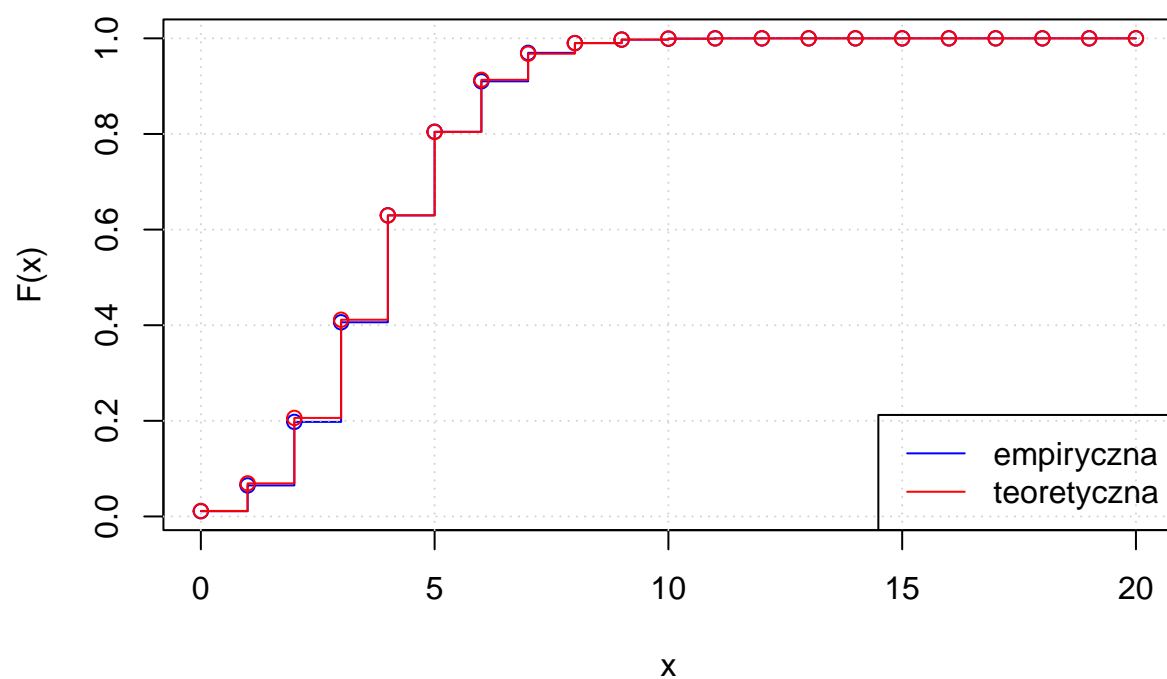

Dystrybuanta dla M = 1000



```

teor_10000b = pbinom(Arg, n, 0.2, lower.tail = TRUE, log.p = FALSE)
Freq_10000b = cumsum(as.numeric(table(factor(proba_10000, levels = Arg)))) / M_10000
plot(Freq_10000b ~ Arg, type = 's', col = 'blue',
     xlab = 'x', ylab = 'F(x)', main = paste0('Dystrybuanta dla M = ', M_10000))
grid()
points(Freq_10000b ~ Arg, col = 'blue')
lines(teor_10000b ~ Arg, type = 's', col = 'red',
     xlab = 'x', ylab = 'F(x)')
points(teor_10000b ~ Arg, col = 'red')
legend('bottomright', c('empiryczna', 'teoretyczna'),
     col = c('blue', 'red'), lwd = 1)
    
```

Dystrybuanta dla $M = 10000$



```
mean(proba_100)
```

```
## [1] 4.01
```

```
var(proba_100)
```

```
## [1] 3.949394
```

```
mean(proba_1000)
```

```
## [1] 3.971
```

```
var(proba_1000)
```

```
## [1] 3.139298
```

```
mean(proba_10000)
```

```
## [1] 4.0192
```

```
var(proba_10000)
```

```
## [1] 3.144746
```

```
n <- 20
```

```
p <- 0.2
```

```
wyniki <- data.frame(  
  Rozmiar_próby = c(100, 1000, 10000),  
  Srednia_empiryczna = c(mean(proba_100), mean(proba_1000), mean(proba_10000)),  
  Wariancja_empiryczna = c(var(proba_100), var(proba_1000), var(proba_10000)),  
  Srednia_teoretyczna = n * p,  
  Wariancja_teoretyczna = n * p * (1 - p)  
)  
wyniki_t <- as.data.frame(t(wyniki))  
colnames(wyniki_t) <- paste0("M=", wyniki$Rozmiar_próby)  
wyniki_t <- wyniki_t[-1, ]  
wyniki_t <- round(wyniki_t, 3)  
print(wyniki_t)
```

```
##           M=100 M=1000 M=10000  
## Srednia_empiryczna  4.010  3.971  4.019  
## Wariancja_empiryczna 3.949  3.139  3.145  
## Srednia_teoretyczna  4.000  4.000  4.000  
## Wariancja_teoretyczna 3.200  3.200  3.200
```

Empiryczne wartości (z próby) będą się zbliżały do teoretycznych wraz ze wzrostem liczby prób (M).

Dla małej próby (np. 100) mogą wystąpić większe odchylenia od wartości teoretycznych.