

# ADPS 2025L — Laboratorium 1 (rozwiązania)

Konrad Lis

## 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

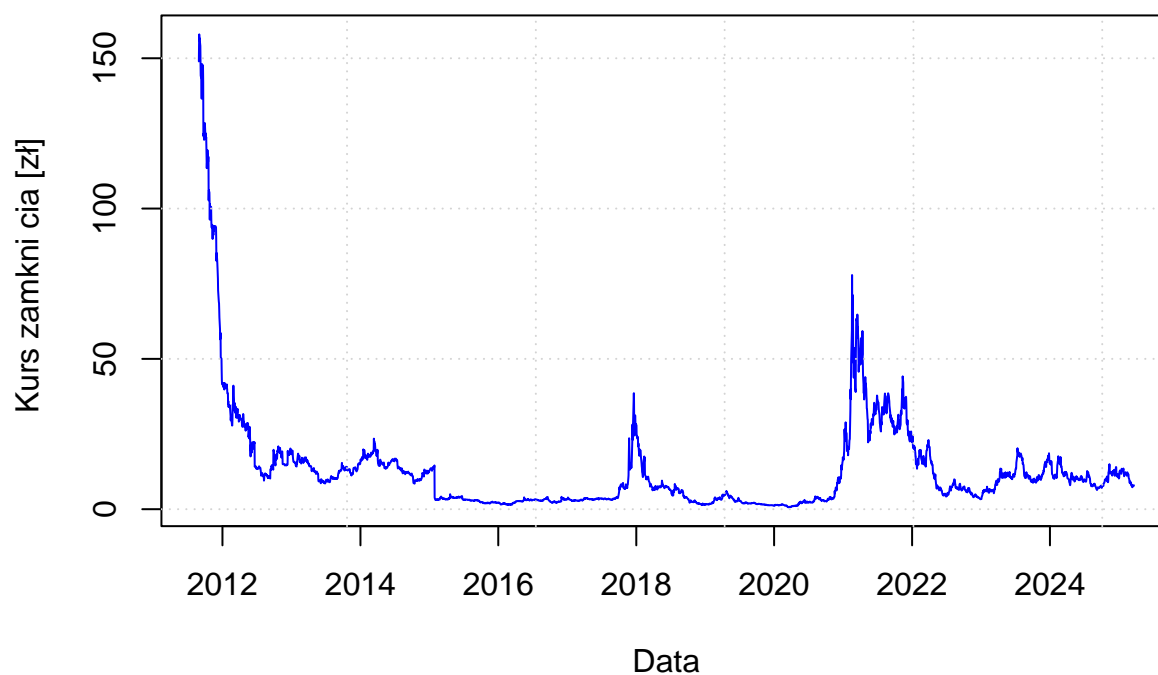
- Wczytanie danych spółek RIOT.US i ograniczenie danych do 18 msc

```
Ticket = 'RIOT.US'
webLink = paste0('https://stooq.pl/q/d/1/?s=', Ticket, '&i=d')
fileName = paste0(Ticket, '.csv')
# if(!file.exists(fileName)) {
  download.file(webLink, fileName)
#}
```

- Kurs RIOT w czasie rzeczywistym

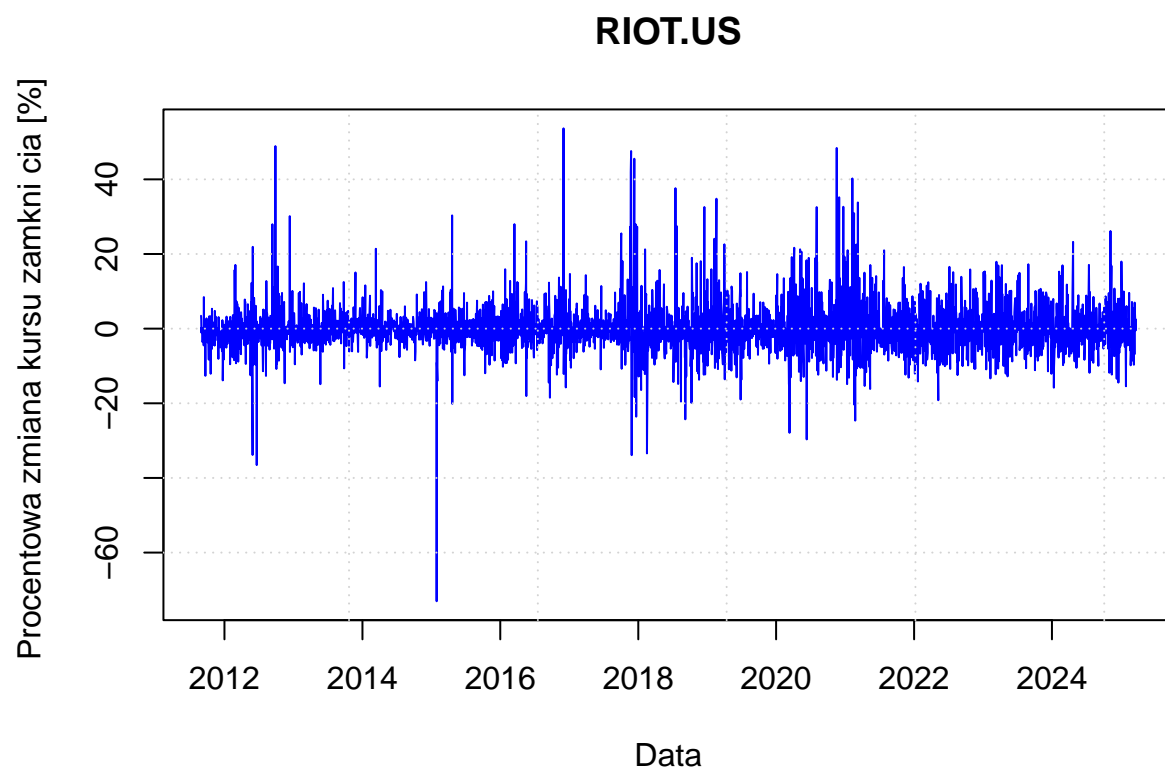
```
df_RIOT.US = read.csv('RIOT.US.csv')
df_RIOT.US$Data = as.Date(df_RIOT.US$Data)
plot(Zamkniecie ~ Data, df_RIOT.US, type = 'l', col = 'blue',
     xlab = 'Data', ylab = 'Kurs zamknięcia [zł]', main = 'RIOT.US w czasie rzeczywistym' )
grid()
```

## RIOT.US w czasie rzeczywistym



- Procentowa dzienna zmiana kursu

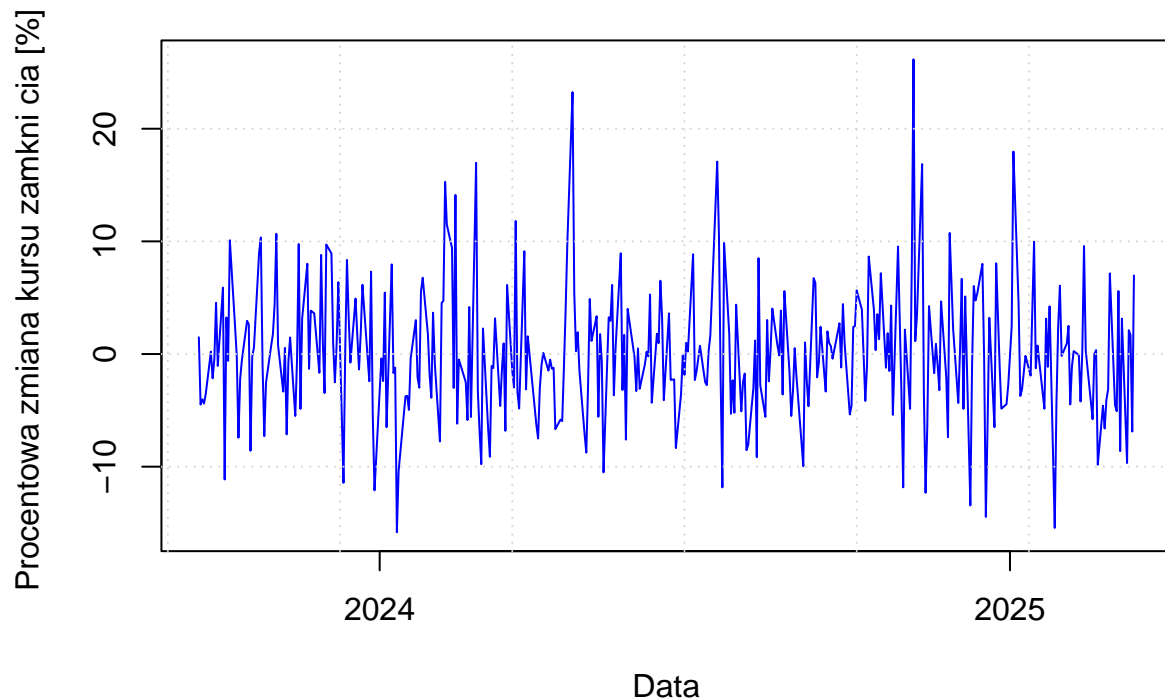
```
df_RIOT.US$Zamkniecie_zm = with(df_RIOT.US, c(NA, 100*diff(Zamkniecie)/Zamkniecie[-length(Zamkniecie)]))
plot(Zamkniecie_zm ~ Data, df_RIOT.US, type = 'l', col = 'blue', xlab = 'Data',
     ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'RIOT.US')
grid()
```



- Ograniczenie czasu do ostatnich 18 miesięcy

```
df_RIOT.US_18msc = df_RIOT.US[which(df_RIOT.US$Data >=
  '2023-09-16' & df_RIOT.US$Data <= '2025-03-15'),]
plot(Zamkniecie_zm ~ Data, df_RIOT.US_18msc, type = 'l', col = 'blue', xlab = 'Data',
  ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'RIOT.US 18 miesięcy')
grid()
```

## RIOT.US 18 miesi cy



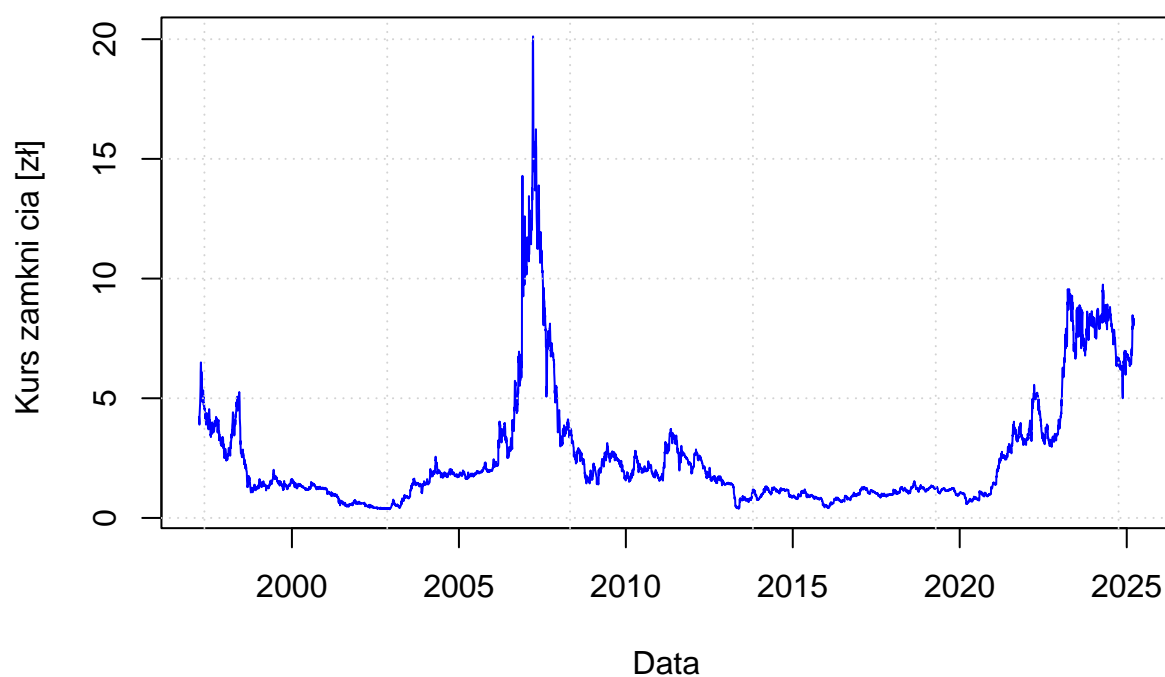
- Wczytanie danych spółki COG i ograniczenie danych do 18 msc

```
Ticket = 'COG'
webLink = paste0('https://stoq.pl/q/d/l/?s=', Ticket, '&i=d')
fileName = paste0(Ticket, '.csv')
# if(!file.exists(fileName)) {
  download.file(webLink, fileName)
#}
```

- Kurs COG w czasie rzeczywistym

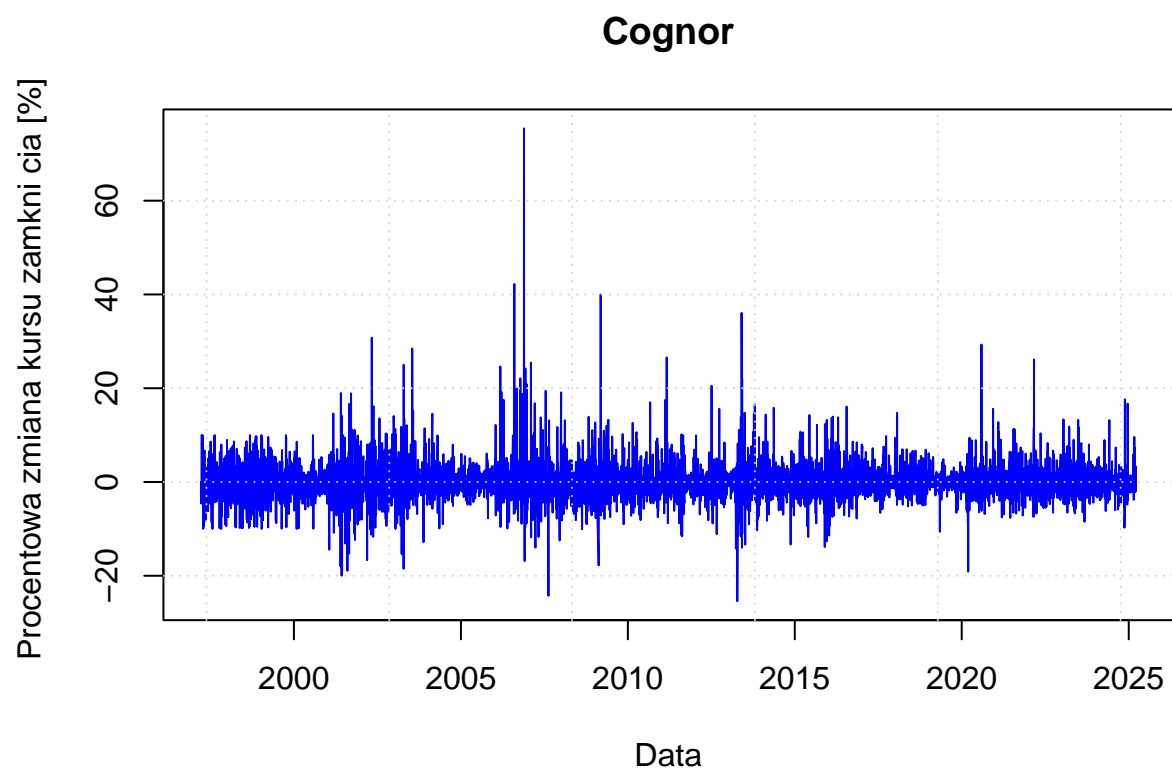
```
df_COG = read.csv('COG.csv')
df_COG$Data = as.Date(df_COG$Data)
plot(Zamkniecie ~ Data, df_COG, type = 'l', col = 'blue',
     xlab = 'Data', ylab = 'Kurs zamknięcia [zł]', main = 'Cognor w czasie rzeczywistym' )
grid()
```

## Cognor w czasie rzeczywistym



- Procentowa dzienna zmiana kursu

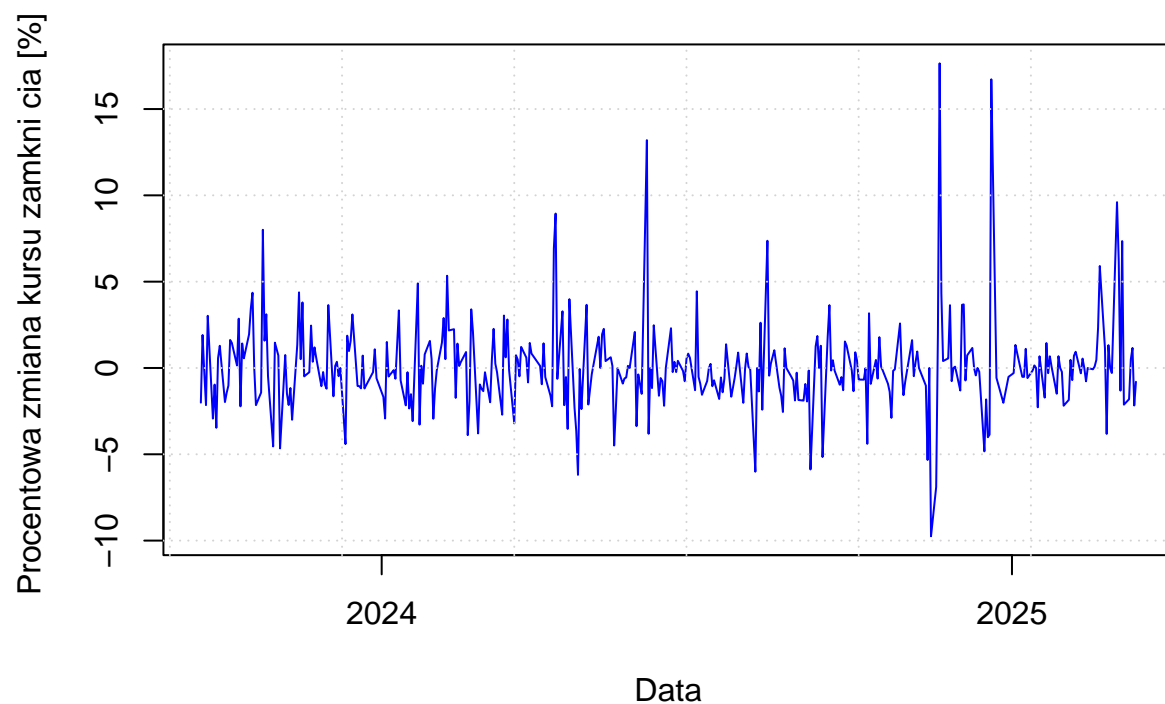
```
df_COG$Zamkniecie_zm = with(df_COG, c(NA, 100*diff(Zamkniecie)/Zamkniecie[-length(Zamkniecie)]))  
plot(Zamkniecie_zm ~ Data, df_COG, type = 'l', col = 'blue', xlab = 'Data',  
      ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'Cognor')  
grid()
```



- Ograniczenie czasu do ostatnich 18 miesięcy

```
df_COG_18msc = df_COG[which(df_COG$Data >= '2023-09-16' & df_COG$Data <= '2025-03-15'),]
plot(Zamkniecie_zm ~ Data, df_COG_18msc, type = 'l', col = 'blue', xlab = 'Data',
      ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'Cognor 18 miesięcy')
grid()
```

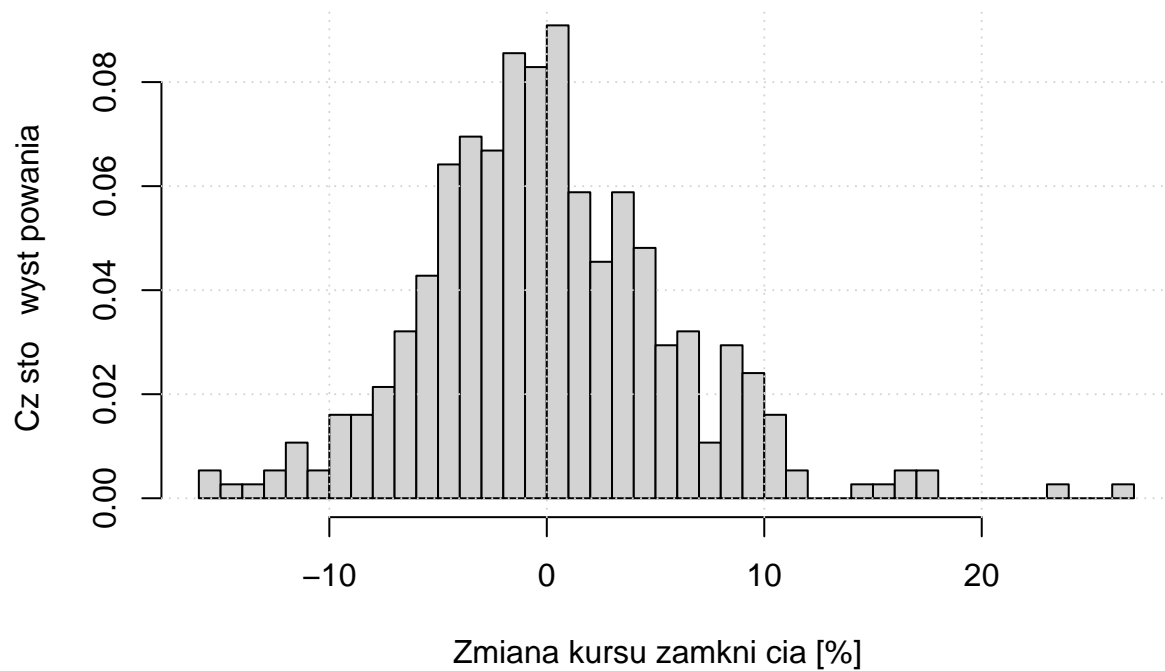
## Cognor 18 miesi cy



- Porównanie histogramów procentów kursów zamknięcia obu spółek

```
hist(df_RIOT.US_18mcs$Zamkniecie_zm, breaks = 50, prob = T,  
     xlab = 'Zmiana kursu zamknięcia [%] ',  
     ylab = 'Częstość występowania',  
     main = paste('Histogram procentowych zmian kursu', 'RIOT.US') )  
grid()
```

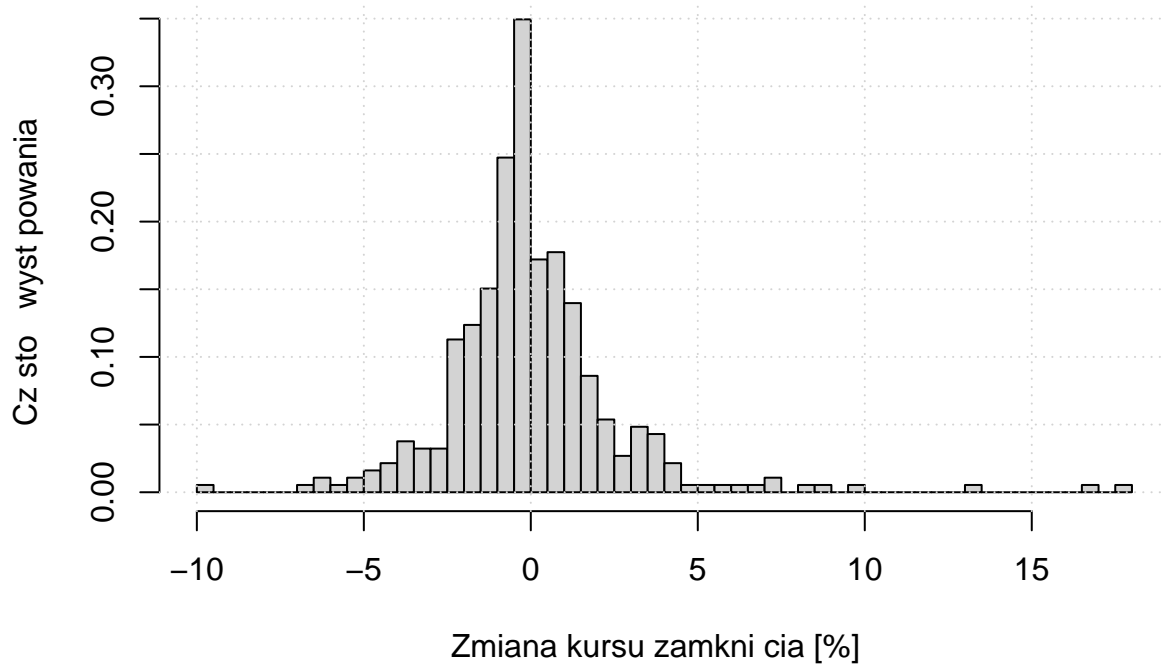
## Histogram procentowych zmian kursu RIOT.US



```
hist(df_COG_18msc$Zamkniecie_zm, breaks = 50, prob = T,  
     xlab = 'Zmiana kursu zamknięcia [%] ',  
     ylab = 'Częstość występowania',  
     main = paste('Histogram procentowych zmian kursu', 'COG') )  
grid()
```



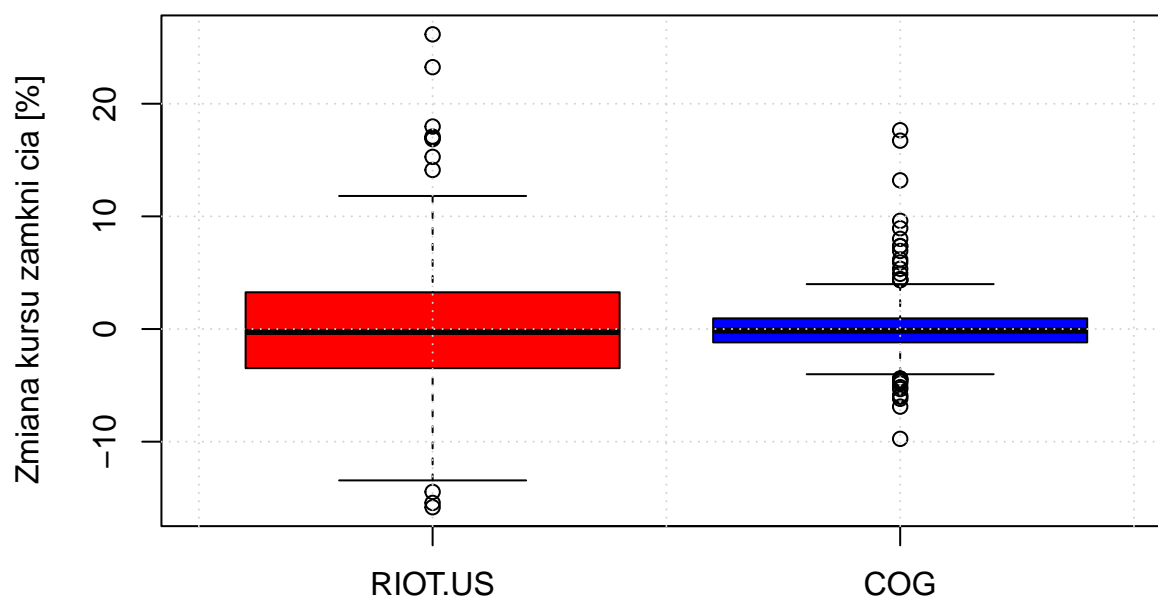
## Histogram procentowych zmian kursu COG



- KOMENTARZ
- RIOT.US charakteryzuje się większymi wahaniami dziennymi oraz częstość występowania jest większa niż w COG.
- Wykres pudełkowy procentowych zmian kursu zamknięcia COG i RIOT.US

```
box_RIOT_18msc = df_RIOT.US_18msc$Zamknięcie_zm
box_COG_18msc = df_COG_18msc$Zamknięcie_zm
boxplot(box_RIOT_18msc, box_COG_18msc, col = c('red','blue'),
        names = c('RIOT.US', 'COG'), ylab = 'Zmiana kursu zamknięcia [%] ',
        main = 'RIOT.US i COG')
grid()
```

## RIOT.US i COG



- KOMENTARZ
- Spółka RIOT.US ma większe wahania kursowe niż spółka COG.

## 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

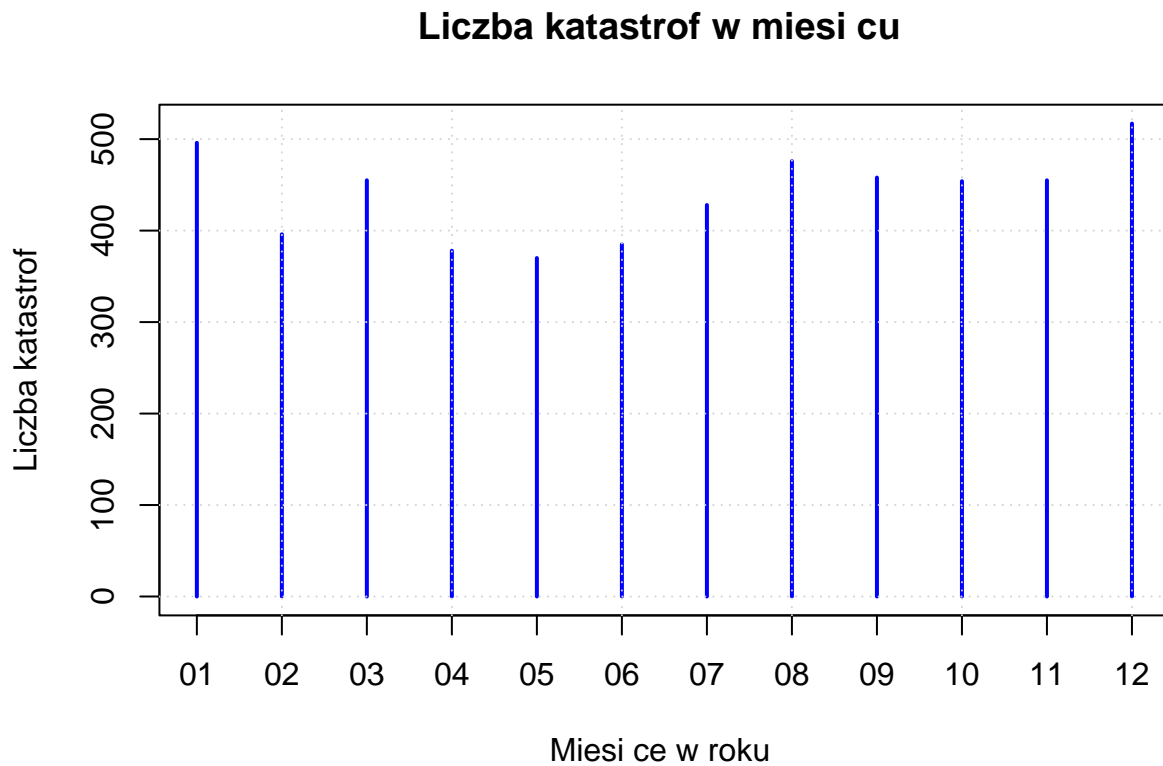
- Wykres liczby katastrof w miesiącach

```
kat = read.csv('crashes.csv')
kat$Year = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%Y')
```

```
kat$Month = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%m')
kat$Day = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%d')
kat$weekDay = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%A')
```

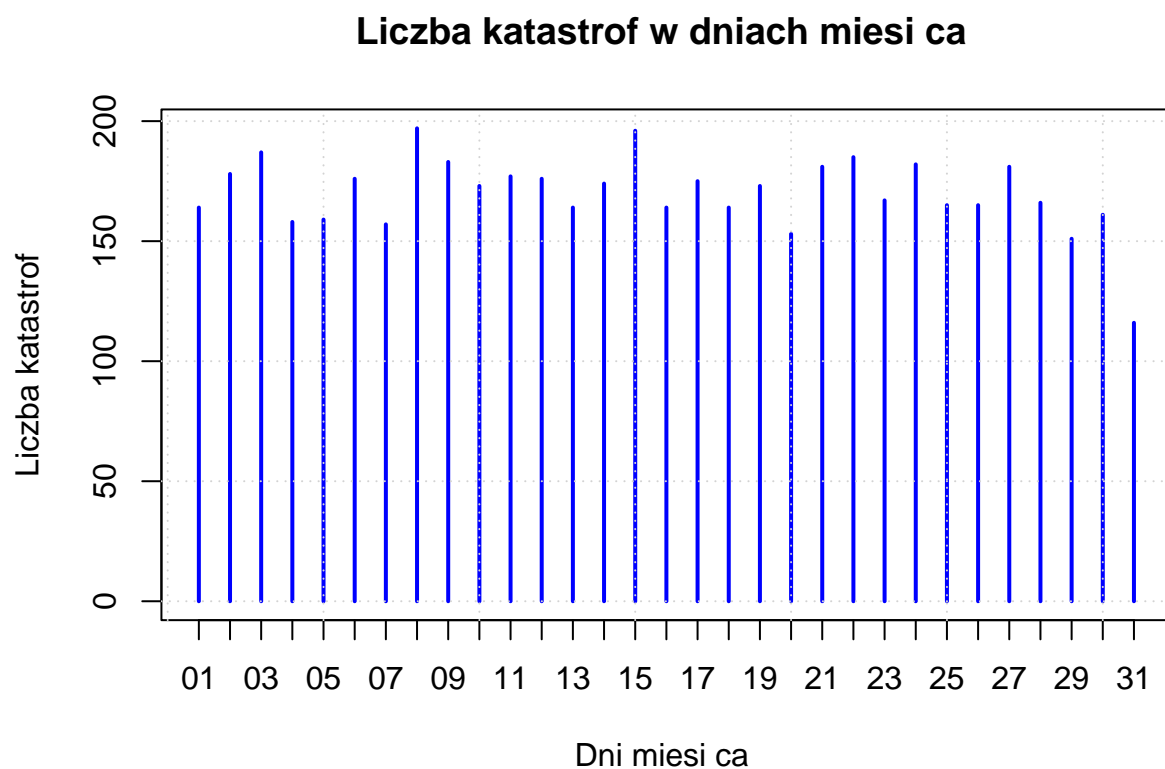
- Wykres liczby katastrof w miesiącach

```
plot(table(kat$Month), type = 'h', col = 'blue', xlab = 'Miesiące w roku',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w miesiącu' )
grid()
```



- Wykres liczby katastrof w dniach miesiąca

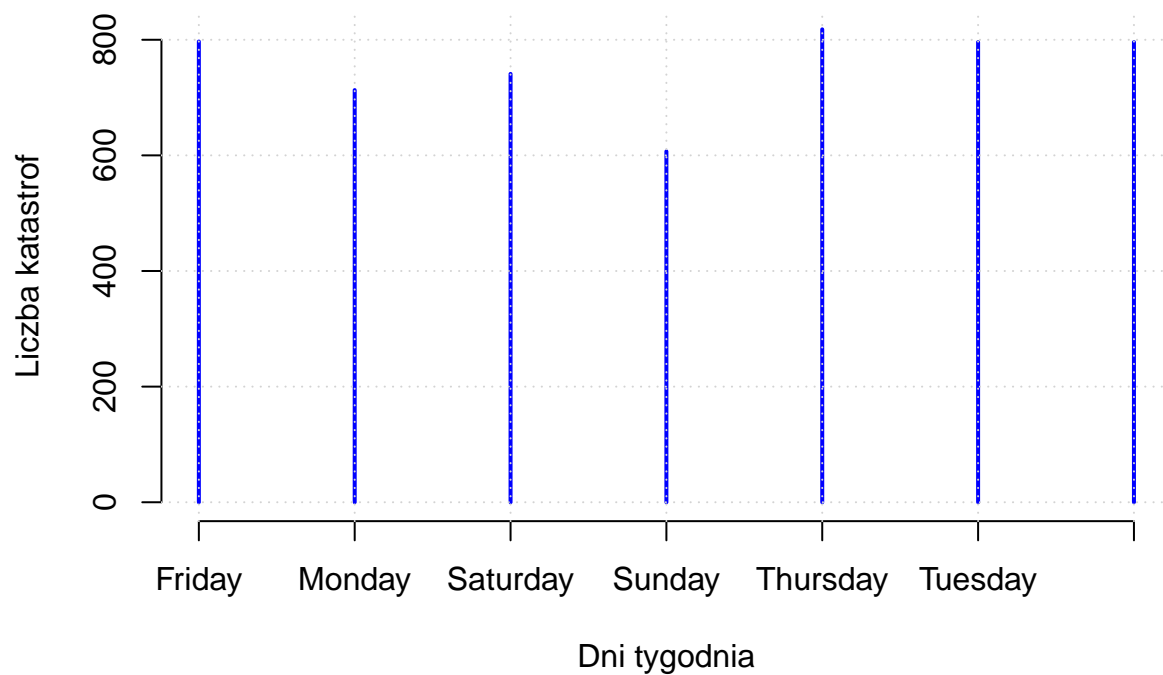
```
plot(table(kat$Day), type = 'h', col = 'blue', xlab = 'Dni miesiąca',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w dniach miesiąca' )
grid()
```



- Wykres liczby katastrof w dniach tygodnia

```
plot(table(kat$weekDay), type = 'h', col = 'blue', xlab = 'Dni tygodnia',  
ylab = 'Liczba katastrof', main = 'Liczba katastrof w dniach tygodnia' )  
grid()
```

## Liczba katastrof w dniach tygodnia

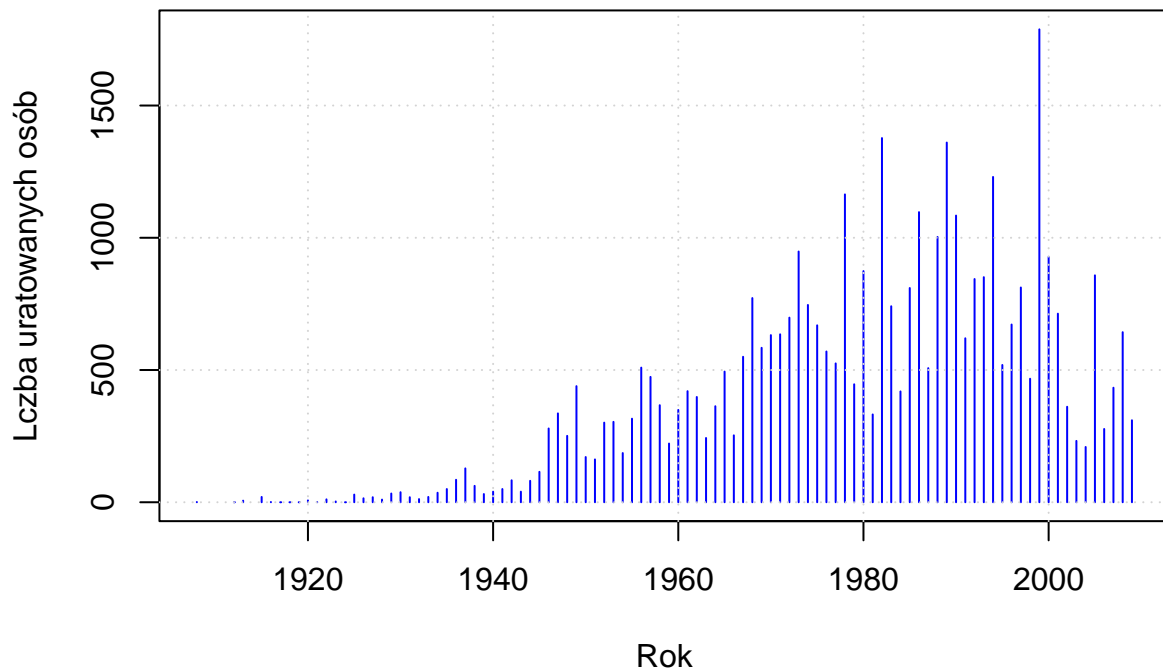


- Liczba osób, które przeżyły katastrofy

```
kat$Rescue = (kat$Aboard - kat$Fatalities)
Rescue_agr = aggregate(Rescue ~ Year, kat, FUN = sum)

plot(Rescue_agr, type = 'h', col = 'blue', xlab = 'Rok',
      ylab = 'Liczba uratowanych osób', main = 'Liczba uratowanych osób w katastrofach w danym roku' )
grid()
```

## Liczba uratowanych osób w katastrofach w danym roku



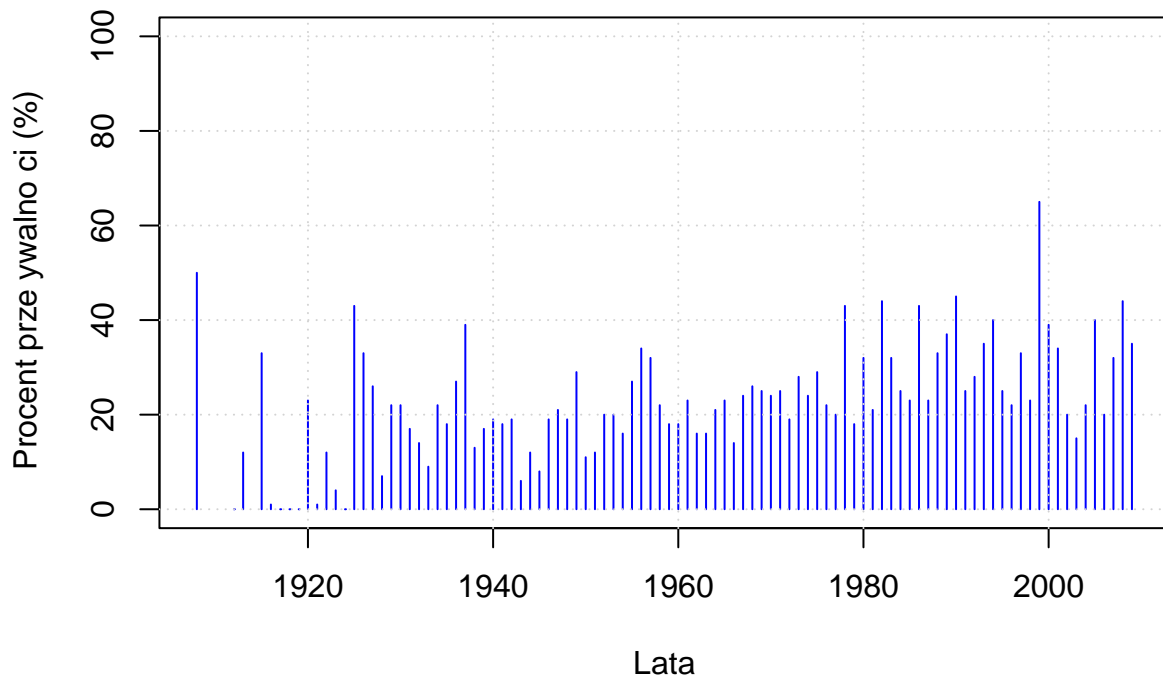
- Odsetek osób (w procentach), które przeżyły katastrofy

```
Fatalities_agr = aggregate(Fatalities ~ Year, kat, FUN = sum)
Aboard_agr = aggregate(Aboard ~ Year, kat, FUN = sum)

kat_agr_ratio = merge(Aboard_agr, Rescue_agr, by = "Year")
kat_agr_ratio$Ratio = round((kat_agr_ratio$Rescue / kat_agr_ratio$Aboard) * 100, digits = 0)

plot(kat_agr_ratio$Year, kat_agr_ratio$Ratio, type = "h", col = "blue", pch = 16, ylim = c(0, 100),
     main = "Odsetek osób, które przeżyły katastrofy", xlab = "Lata", ylab = "Procent przeżywalności (%)")
grid()
```

## Odsetek osób, które przeżyły katastrofy



- KOMENATRZ
- Wraz z biegiem lat liczba ocalałych wzrasta

## 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.

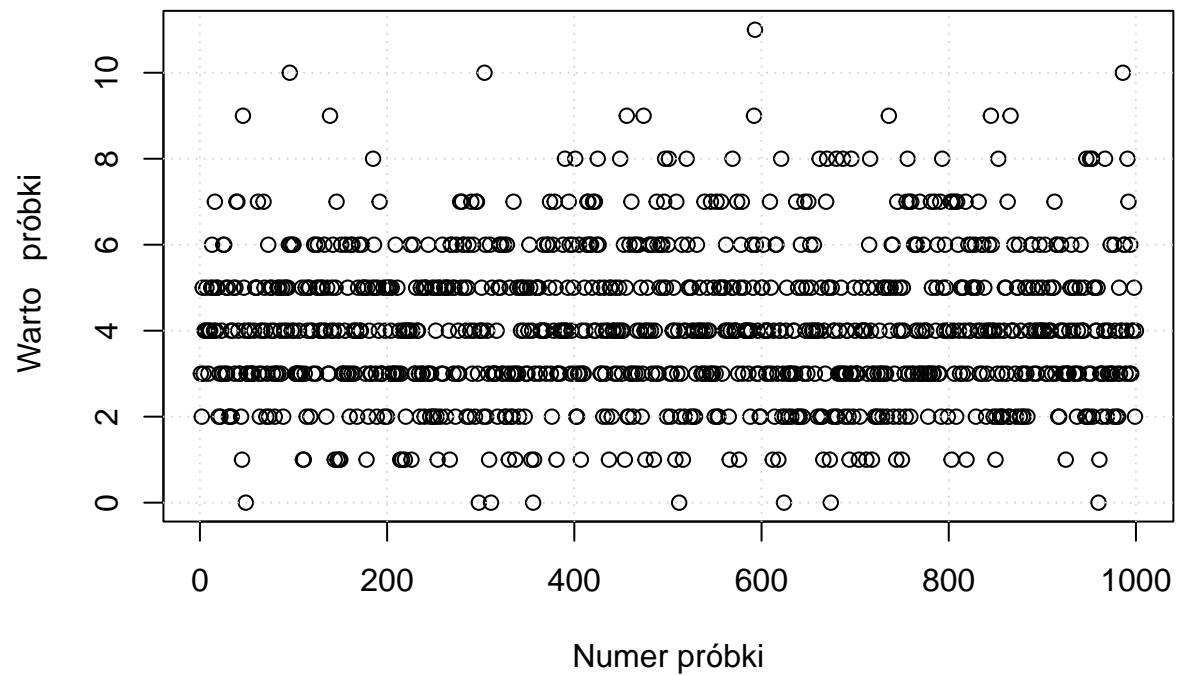
### Rozwiązanie

- Próby losowe rBinom

```
M = 1000
proba1 = rbinom(M, size = 20, prob = 0.2)
plot(proba1, xlab = 'Numer próbek', ylab = 'Wartość próbek',
```

```
main = 'Wartości wygenerowanych próbek Binom (20, 0.2)'\ngrid()
```

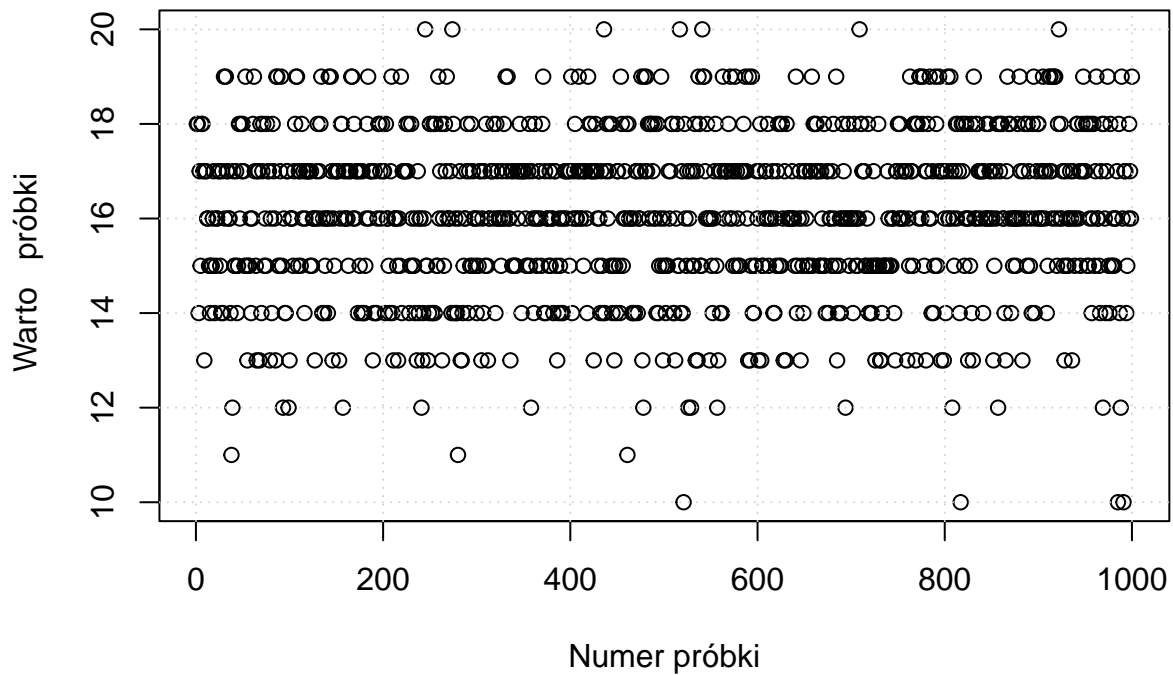
## Warto ci wygenerowanych próbek Binom (20, 0.2)



```
proba2 = rbinom(M, size = 20, prob = 0.8)\nplot(proba2, xlab = 'Numer próbki', ylab = 'Wartość próbki',\nmain = 'Wartości wygenerowanych próbek Binom (20, 0.8)'\ngrid())
```



## Warto ci wygenerowanych próbek Binom (20, 0.8)



- Empiryczne i teoretyczne funkcje prawdopodobieństwa (dBinom)

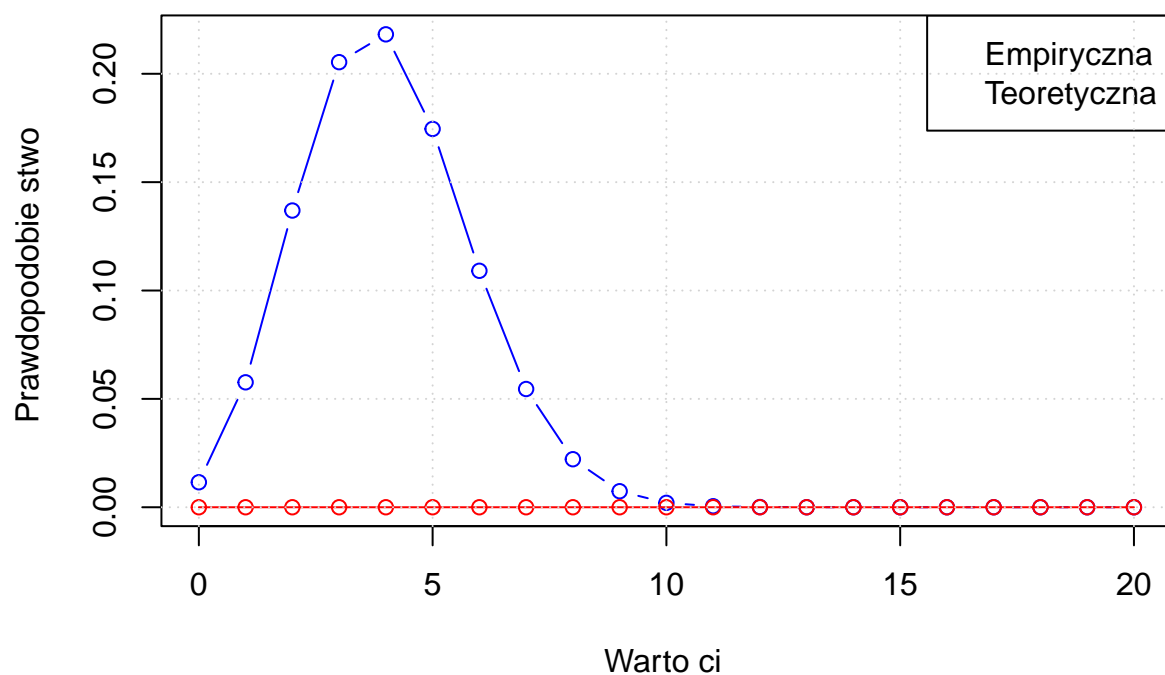
```
size = 20
x = 0:size

proba3 = dbinom(x, size = 20, prob = 0.2, log = FALSE)
proba4 = dbinom(x, size = 20, prob = 0.8, log = FALSE)

Emp3 = table(factor(proba3, levels = x)) / M
Emp4 = table(factor(proba4, levels = x)) / M

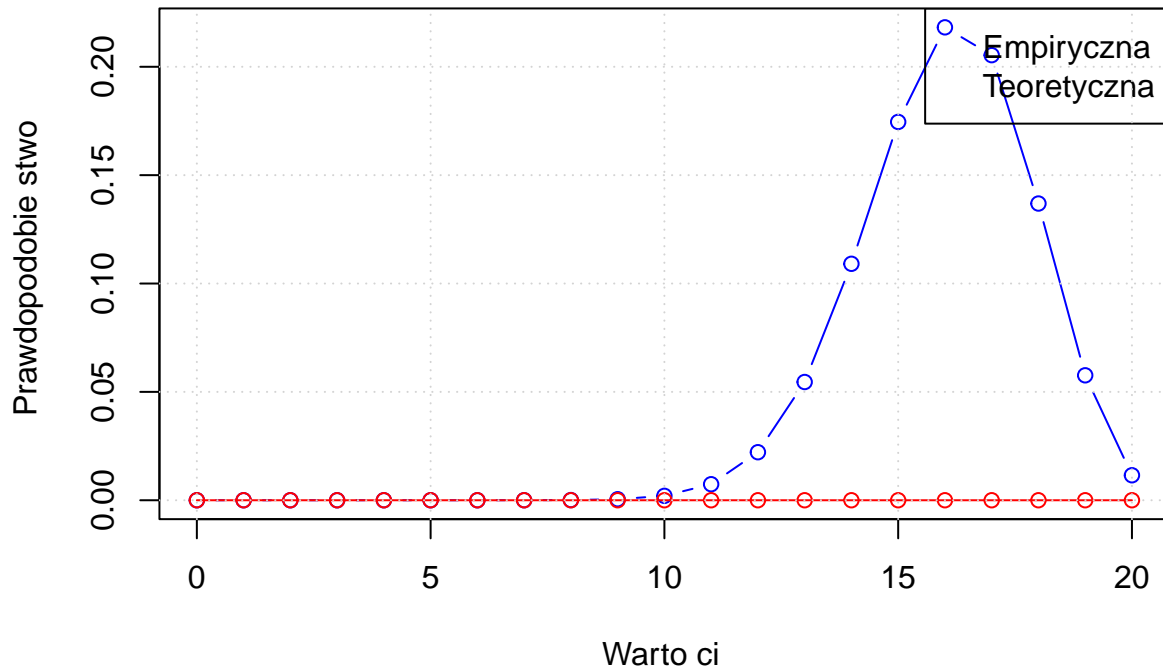
plot(x, proba3, type = 'b', col = 'blue',
     ylim = c(0, max(proba3, Emp3)),
     main = "Funkcja prawdopodobieństwa: Binom(20, 0.2)", xlab = "Wartości",
     ylab = "Prawdopodobieństwo")
lines(x, Emp3, col = "red", type = "o")
legend("topright", legend = c("Empiryczna", "Teoretyczna"), col = c("blue", "red"))
grid()
```

## Funkcja prawdopodobieństwa: Binom(20, 0.2)



```
plot(x, proba4, type = 'b', col = 'blue',
     ylim = c(0, max(proba4, Emp4)),
     main = "Funkcja prawdopodobieństwa: Binom(20, 0.2)", xlab = "Wartości",
     ylab = "Prawdopodobieństwo")
lines(x, Emp4, col = "red", type = "o")
legend("topright", legend = c("Empiryczna", "Teoretyczna"), col = c("blue", "red"))
grid()
```

## Funkcja prawdopodobieństwa: Binom(20, 0.8)



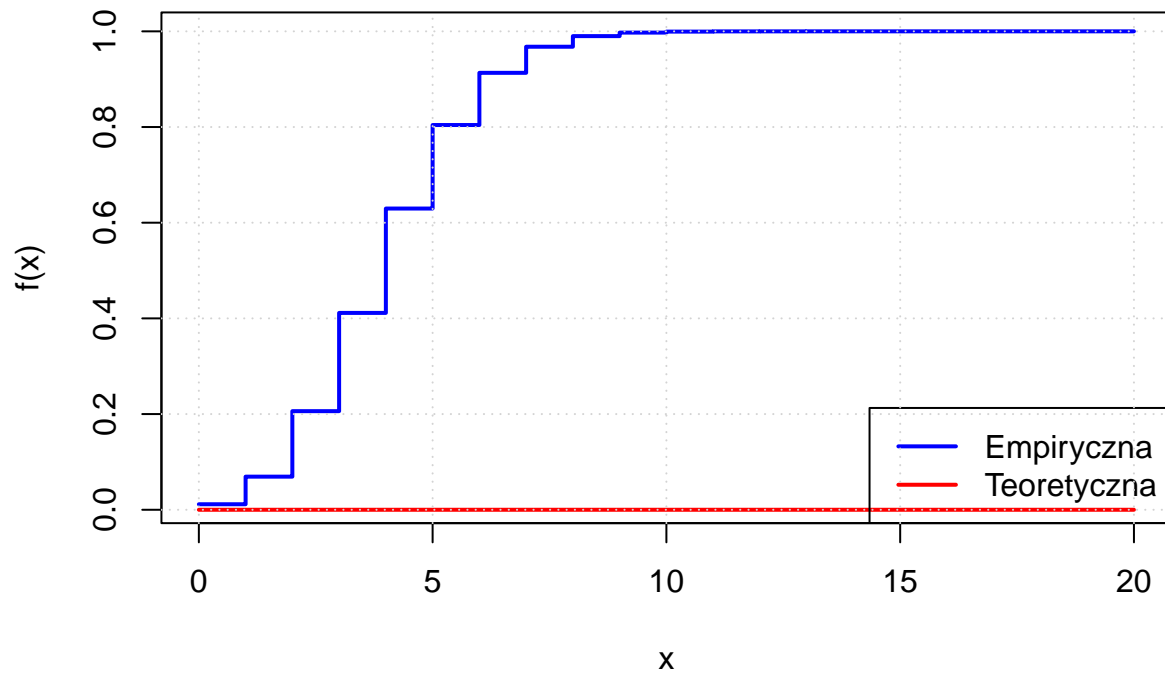
- Empiryczne i teoretyczne dystrybuanty (pBinom)

```
proba5 = pbinom(x, size = 20, prob = 0.2, log = FALSE)
proba6 = pbinom(x, size = 20, prob = 0.8, log = FALSE)
```

```
Emp3_dyst <- cumsum(Emp3)
Emp4_dyst <- cumsum(Emp4)
```

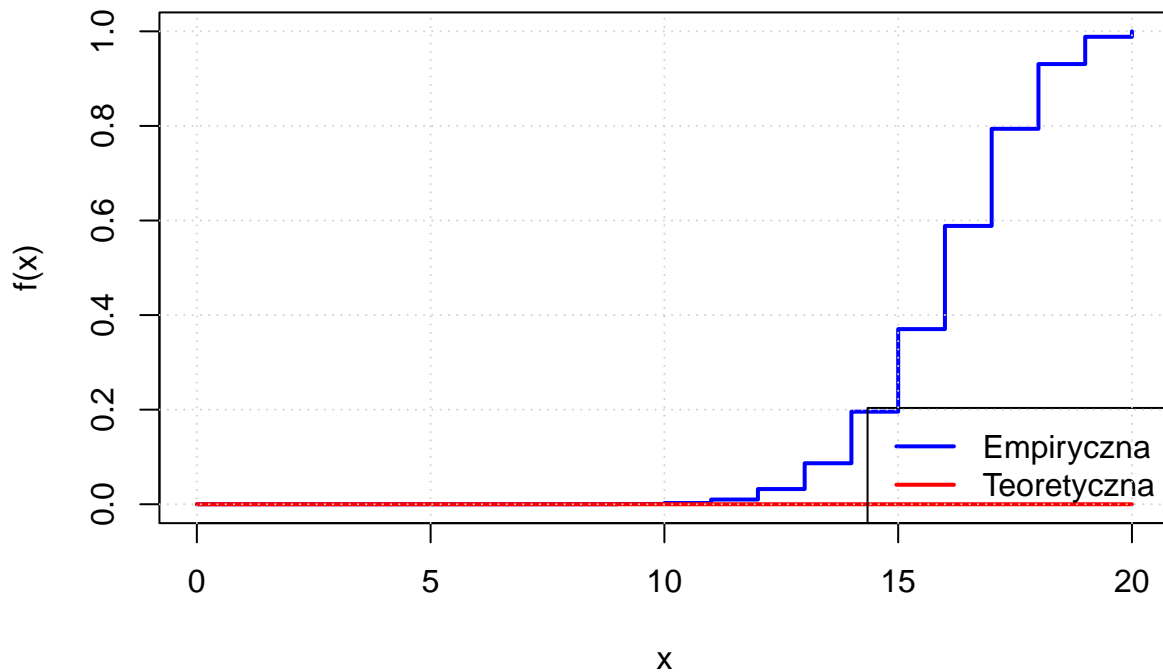
```
plot(x, proba5, type = "s", col = "blue", lwd = 2,
     main = "Dystrybuanta: Binom(20, 0.2)", xlab = "x", ylab = "f(x)")
lines(x, Emp3_dyst, type = "s", col = "red", lwd = 2)
legend("bottomright", legend = c("Empiryczna", "Teoretyczna"), col = c("blue", "red"), lwd = 2)
grid()
```

### Dystrybuanta: Binom(20, 0.2)



```
plot(x, proba6, type = "s", col = "blue", lwd = 2,
     main = "Dystrybuanta: Binom(20, 0.8)", xlab = "x", ylab = "f(x)")
lines(x, Emp4_dyst, type = "s", col = "red", lwd = 2)
legend("bottomright", legend = c("Empiryczna", "Teoretyczna"), col = c("blue", "red"), lwd = 2)
grid()
```

## Dystrybuanta: Binom(20, 0.8)



### Zadanie 4 (1,5 pkt)

#### Treść zadania

1. Dla rozkładu dwumianowego 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 Binom(20, 0.2).

#### Rozwiązanie

- Próby losowe

```
M1 = 100
M2 = 1000
M3 = 10000

proba1 = rbinom(M1, size = 20, prob = 0.2)
proba2 = rbinom(M2, size = 20, prob = 0.2)
proba3 = rbinom(M3, size = 20, prob = 0.2)
```

- Empiryczne i teoretyczne funkcje prawdopodobieństwa (dBinom)

```

size = 20
x = 0:size

proba = dbinom(x, size = 20, prob = 0.2)

Emp1 = table(factor(proba1, levels = x)) / M1 # M100
Emp2 = table(factor(proba2, levels = x)) / M2 # M1000
Emp3 = table(factor(proba3, levels = x)) / M3 # M10000

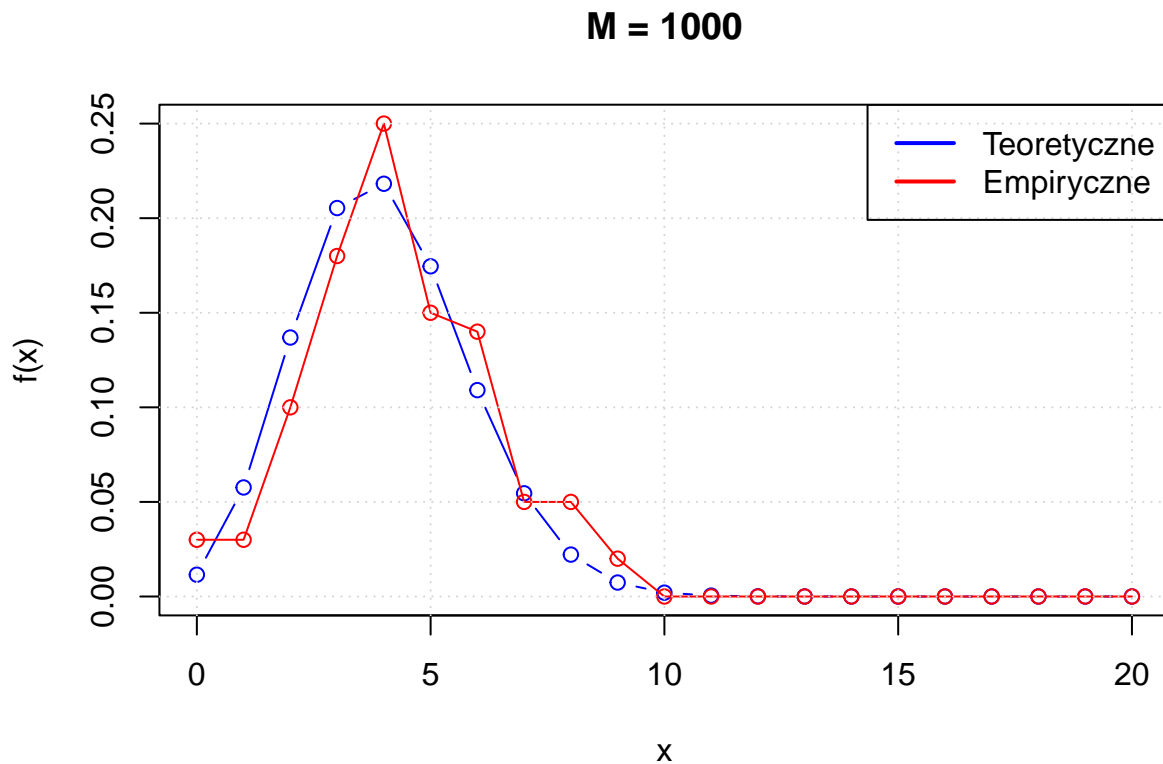
```

- Wykresy powyższych funkcji prawdopodobieństwa (dBinom)

```

plot(x, proba, type = "b", col = "blue",
     ylim = c(0, max(proba, Emp1)),
     main = "M = 1000", xlab = "x", ylab = "f(x)")
lines(x, Emp1, col = "red")
points(x, Emp1, col = "red")
grid()
legend("topright", legend = c("Teoretyczne", "Empiryczne"), col = c("blue", "red"), lwd = 2)

```

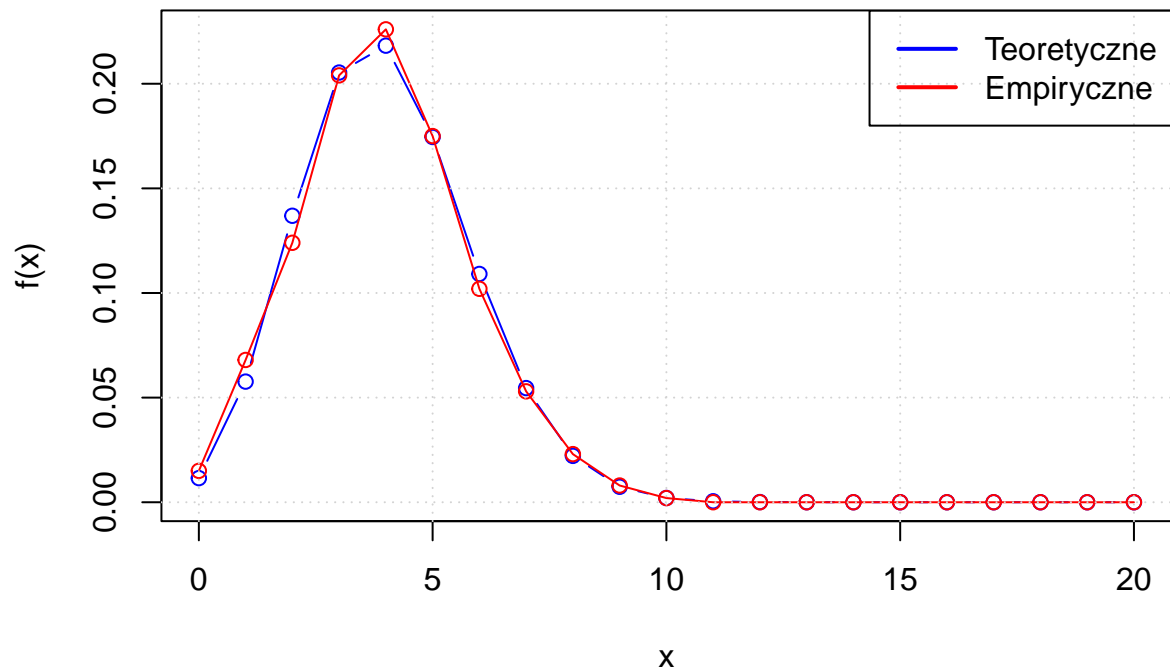


```

plot(x, proba, type = "b", col = "blue",
     ylim = c(0, max(proba, Emp2)),
     main = "M = 1000", xlab = "x", ylab = "f(x)")
lines(x, Emp2, col = "red")
points(x, Emp2, col = "red")
grid()
legend("topright", legend = c("Teoretyczne", "Empiryczne"), col = c("blue", "red"), lwd = 2)

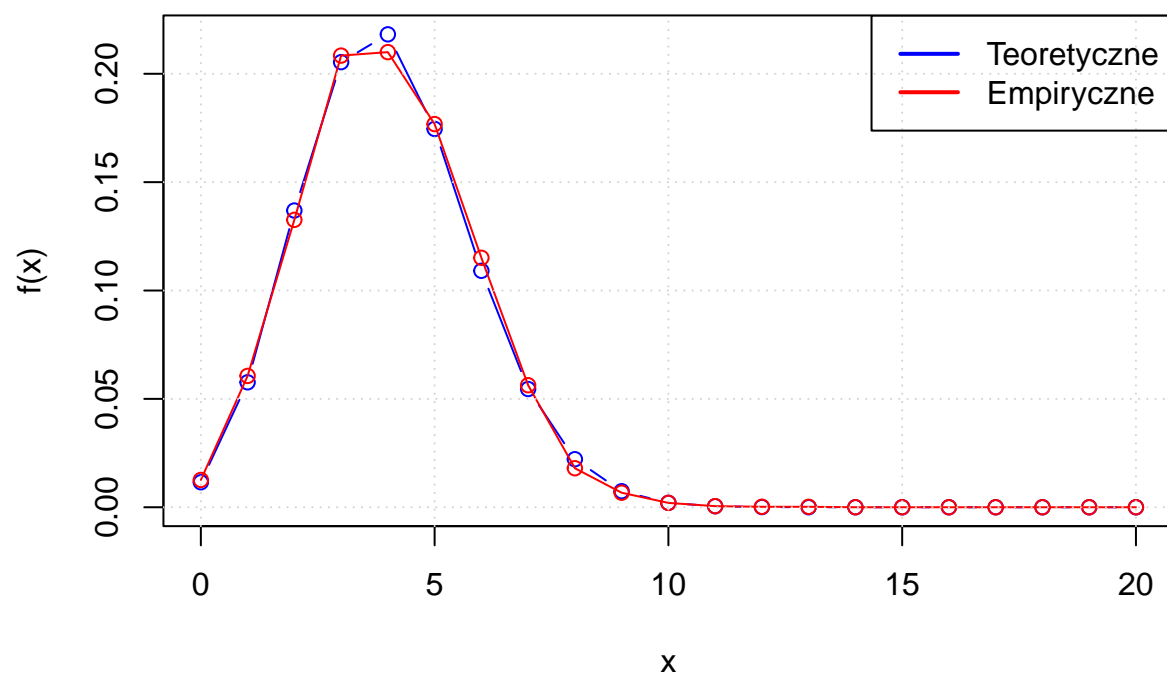
```

**M = 1000**



```
plot(x, proba, type = "b", col = "blue",  
     ylim = c(0, max(proba, Emp3)),  
     main = "M = 10000", xlab = "x", ylab = "f(x)")  
lines(x, Emp3, col = "red")  
points(x, Emp3, col = "red")  
grid()  
legend("topright", legend = c("Teoretyczne", "Empiryczne"), col = c("blue", "red"), lwd = 2)
```

**M = 10000**

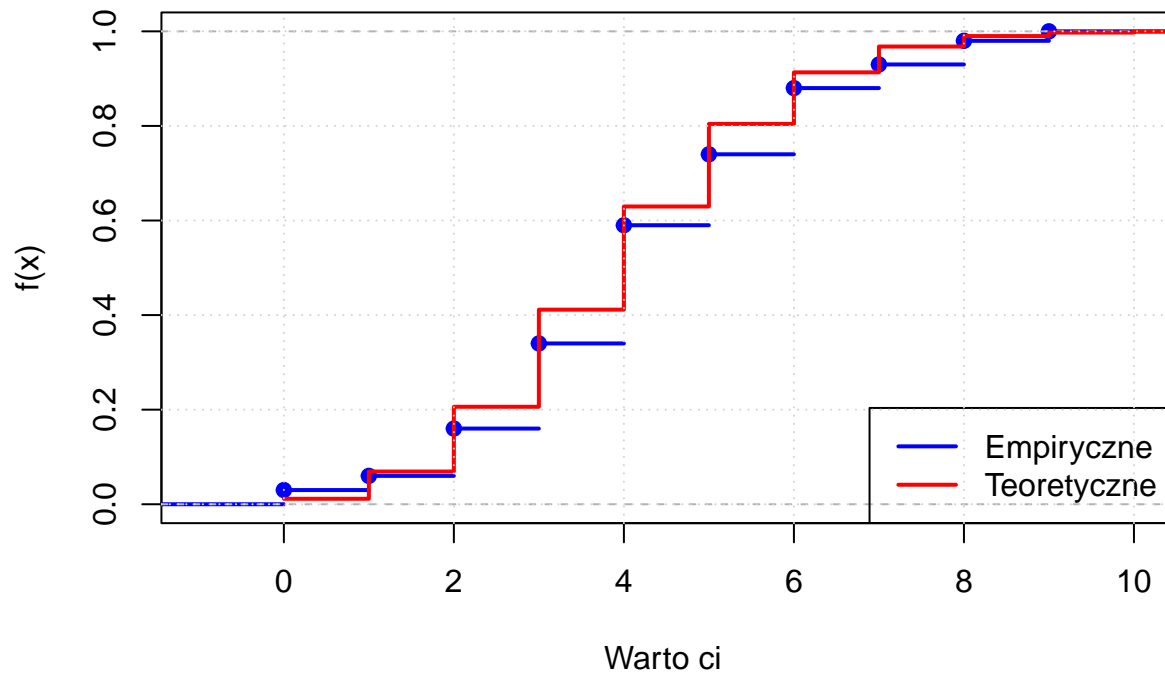


- Empiryczne i teoretyczne dystrybuanty

```
plot(ecdf(proba1), xlab = "Wartości", ylab = "f(x)",  
main = "Dystrybuanta (M = 100)", col = "blue", lwd = 2)  
lines(x, pbinom(x, size = 20, prob = 0.2), type = "s", col = "red", lwd = 2)  
legend("bottomright", legend = c("Empiryczne", "Teoretyczne"),  
col = c("blue", "red"), lwd = 2)  
grid()
```

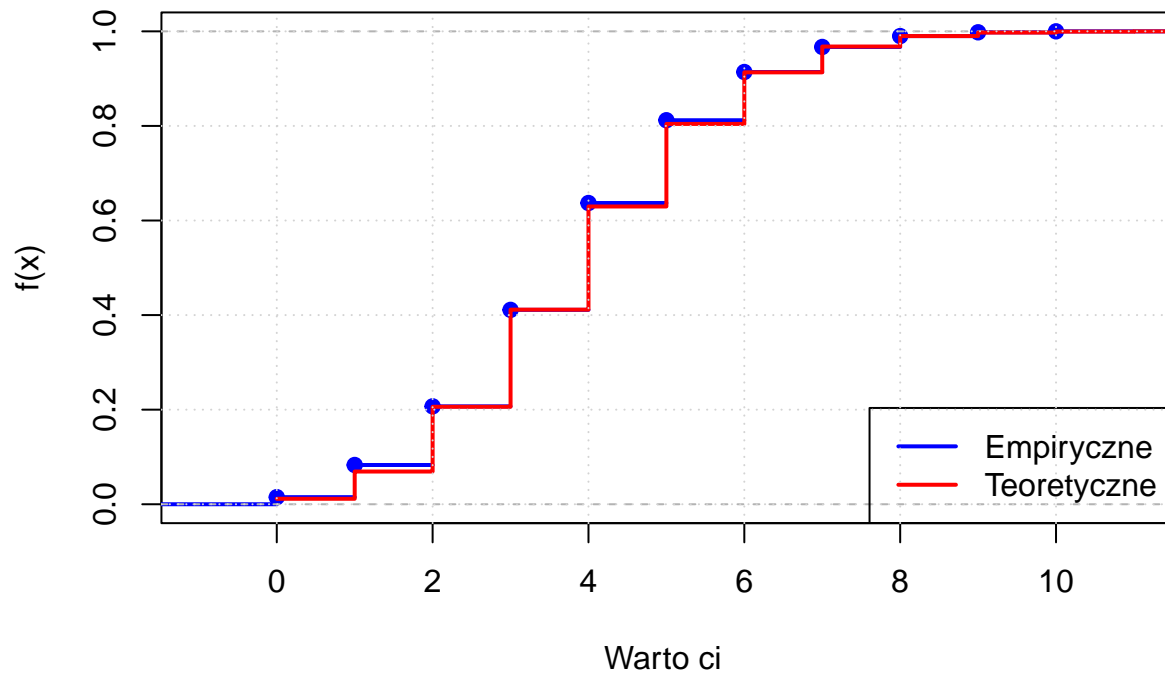


## Dystrybuanta (M = 100)



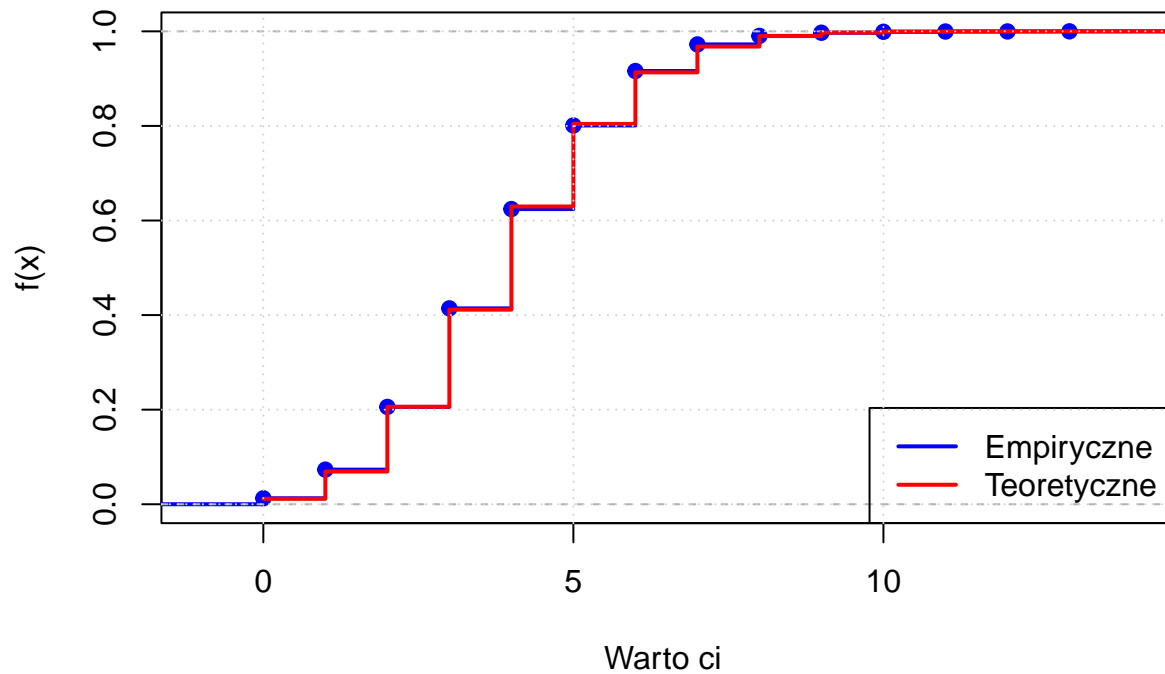
```
plot(ecdf(proba2), xlab = "Wartości", ylab = "f(x)",  
main = "Dystrybuanta (M = 1000)", col = "blue", lwd = 2)  
lines(x, pbinom(x, size = 20, prob = 0.2), type = "s", col = "red", lwd = 2)  
legend("bottomright", legend = c("Empiryczne", "Teoretyczne"),  
col = c("blue", "red"), lwd = 2)  
grid()
```

## Dystrybuanta (M = 1000)



```
plot(ecdf(proba3), xlab = "Wartości", ylab = "f(x)",  
main = "Dystrybuanta (M = 10000)", col = "blue", lwd = 2)  
lines(x, pbinom(x, size = 20, prob = 0.2), type = "s", col = "red", lwd = 2)  
legend("bottomright", legend = c("Empiryczne", "Teoretyczne"),  
col = c("blue", "red"), lwd = 2)  
grid()
```

## Dystrybuanta (M = 10000)



- Wartości średnie i wariancje
- Próba M = 100

```
Emp_mean_100 = mean(proba1)
print(Emp_mean_100)
```

```
## [1] 4.29
```

```
Emp_var_100 = var(proba1)
print(Emp_var_100)
```

```
## [1] 3.743333
```

- Próba M = 1000

```
Emp_mean_1000 = mean(proba2)
print(Emp_mean_1000)
```

```
## [1] 3.966
```

```
Emp_var_1000 = var(proba2)
print(Emp_var_1000)
```

```
## [1] 3.254098
```

- Próba M = 10000

```
Emp_mean_10000 = mean(proba3)
print(Emp_mean_10000)
```

```
## [1] 3.9945
Emp_var_10000 = var(proba3)
print(Emp_var_10000)

## [1] 3.210991
    • Próba teoretyczna
Proba_mean = 20 * 0.2
print(Proba_mean)

## [1] 4
Proba_var = 20 * 0.2 * (1 - 0.2)
print(Proba_var)

## [1] 3.2
```