

ADPS 20L — Ćwiczenie 1

Gabriel R.

Zadanie nr 1

TREŚĆ ZADANIA:

Dla wybranych dwóch spółek:

- 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:

- Znajduję i pobieram dane historyczne spółek giełdowych z portalu <http://www.bossa.pl> (zakładka Notowania & wykresy; Dane do programów AT -> Metastock -> Wszystkie grupy GPW -> baza danych w formacie tekstowym -> mstall.zip) <http://bossa.pl/pub/metastock/mstock/mstall.zip>
- Zapisuję plik mstall.zip do katalogu ADPS/Lab1.

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

Wybieram spółkę nr 1: 4MOBILITY

- Rozpakowuję w tym katalogu dane spółki 4MOBILITY:

```
unzip('mstall.zip', '4MOBILITY.mst')
```

- Wczytuję dane z pliku 4MOBILITY.mst do środowiska R:

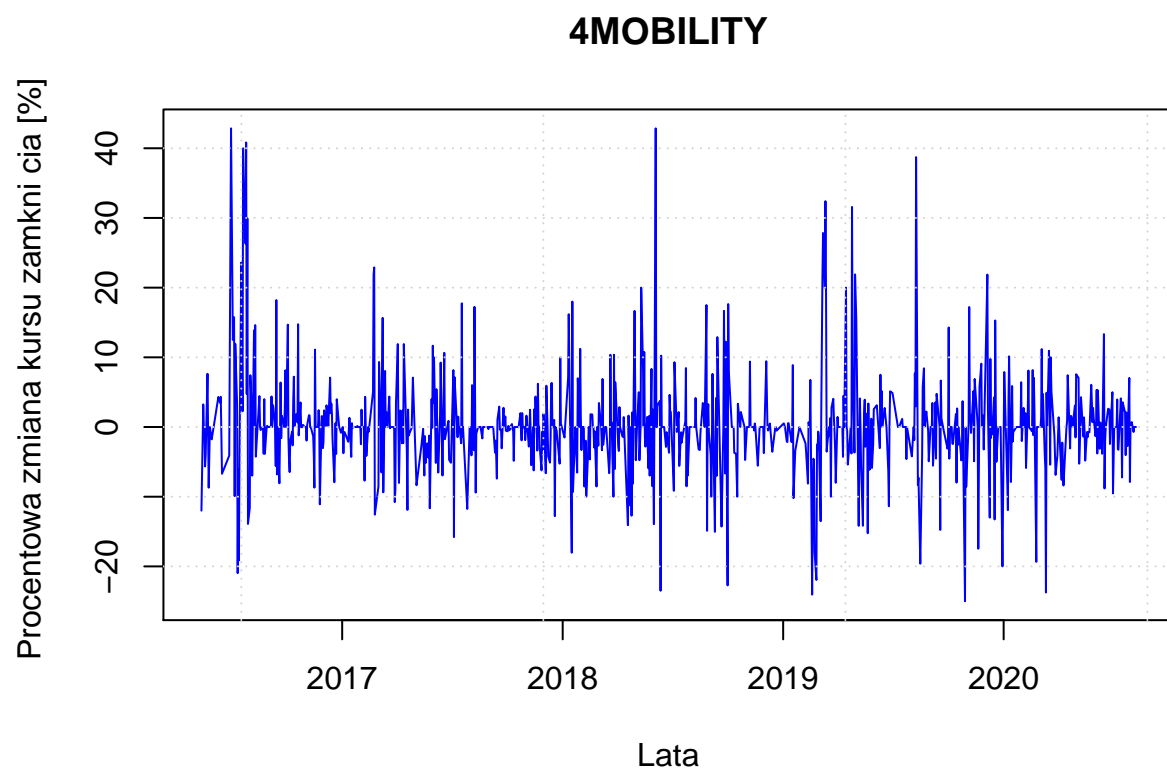
```
df_4MOBILITY = read.csv('4MOBILITY.mst')
```

- Zmieniam nazwy kolumn:

```
names(df_4MOBILITY) = c('ticker', 'date', 'open', 'high', 'low', 'close', 'vol')
```

- Wykres procentowych zmian kursu zamknięcia:

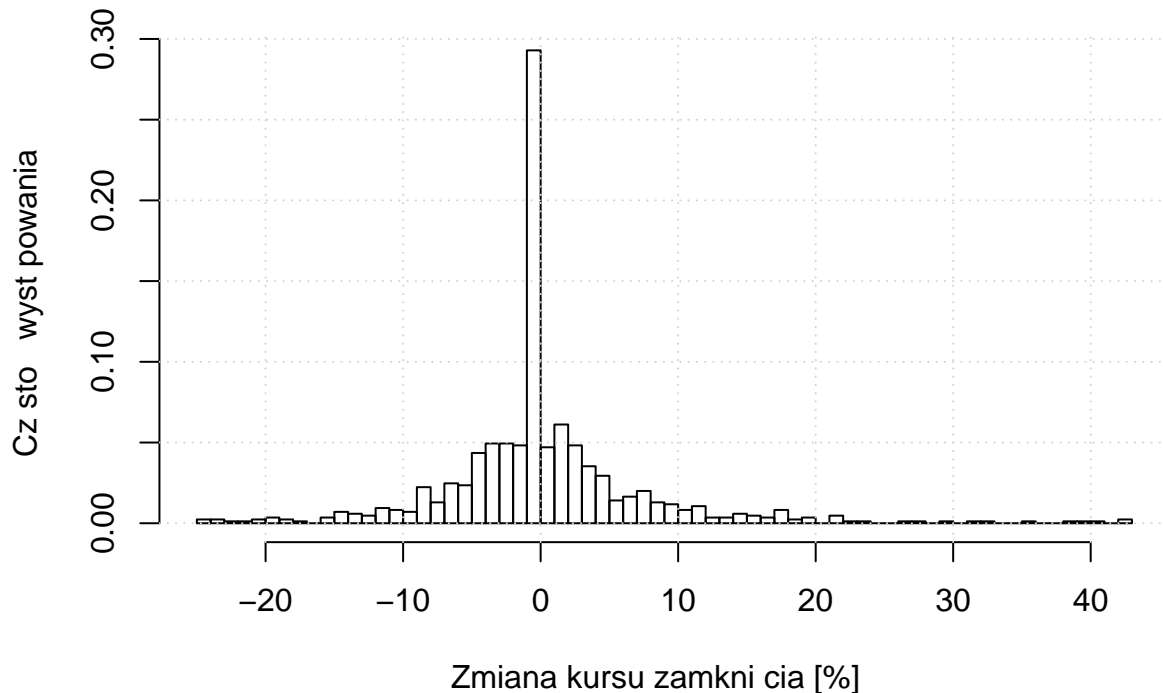
```
df_4MOBILITY$date = as.Date.character(df_4MOBILITY$date, format = '%Y%m%d')  
df_4MOBILITY$close_ch = with(df_4MOBILITY, c(NA, 100*diff(close)/close[-length(close)]))  
plot(close_ch ~ date, df_4MOBILITY, type = 'l', col = 'blue', xlab = 'Lata',  
      ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = '4MOBILITY' )  
grid()
```



- Histogram procentowych zmian kursu spółki 4MOBILITY:

```
hist(df_4MOBILITY$close_ch, breaks = 50, prob = T,  
xlab = 'Zmiana kursu zamknięcia [%] ',  
ylab = 'Częstość występowania',  
main = 'Histogram procentowych zmian kursu 4MOBILITY' )  
grid()
```

Histogram procentowych zmian kursu 4MOBILITY



*Przypisuję wartość close spółki 4MOBILITY do zmiennej SP1:

```
SP1 <- (df_4MOBILITY$close_ch= with(df_4MOBILITY, c(NA, 100*diff(close)/close[-length(close)])))
```

Wybieram spółkę nr 2: MBANK

- Rozpakowuję w tym katalogu dane spółki MBANK:

```
unzip('mstall.zip', 'MBANK.mst')
```

- Wczytuję dane z pliku MBANK.mst do środowiska R:

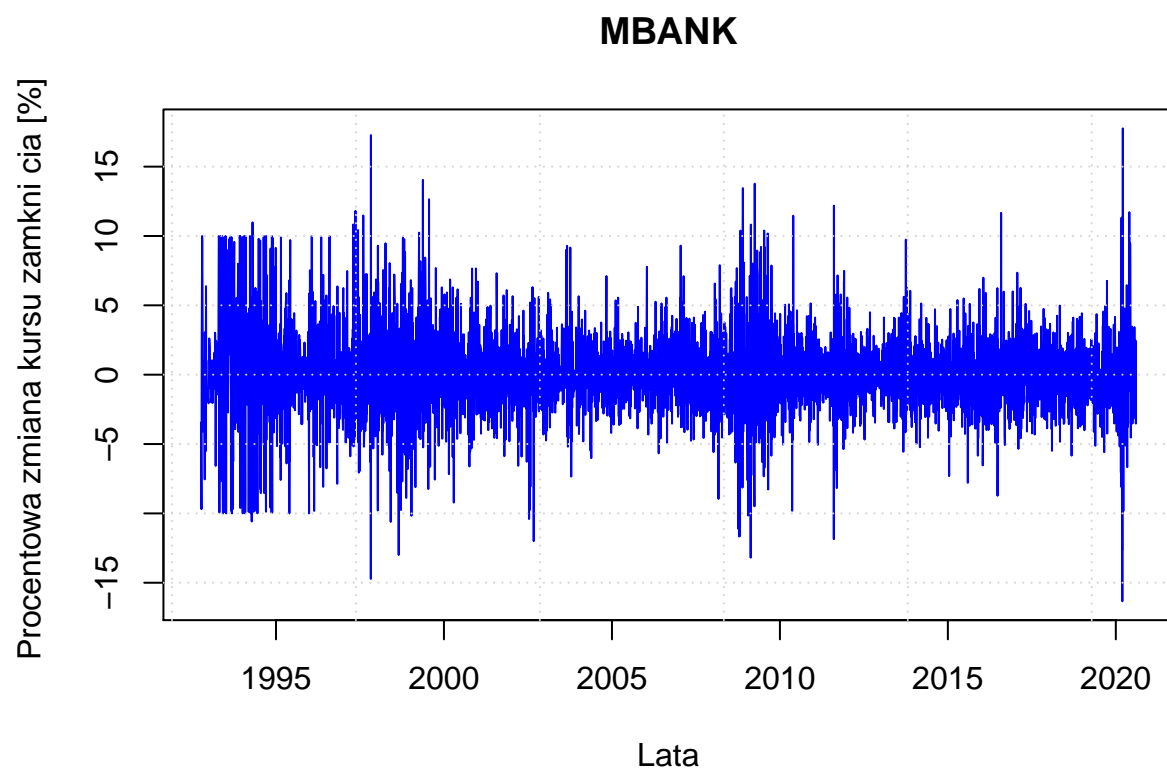
```
df_MBANK = read.csv('MBANK.mst')
```

- Zmieniam nazwy kolumn:

```
names(df_MBANK) = c('ticker', 'date', 'open', 'high', 'low', 'close', 'vol')
```

- Wykres procentowych zmian kursu zamknięcia:

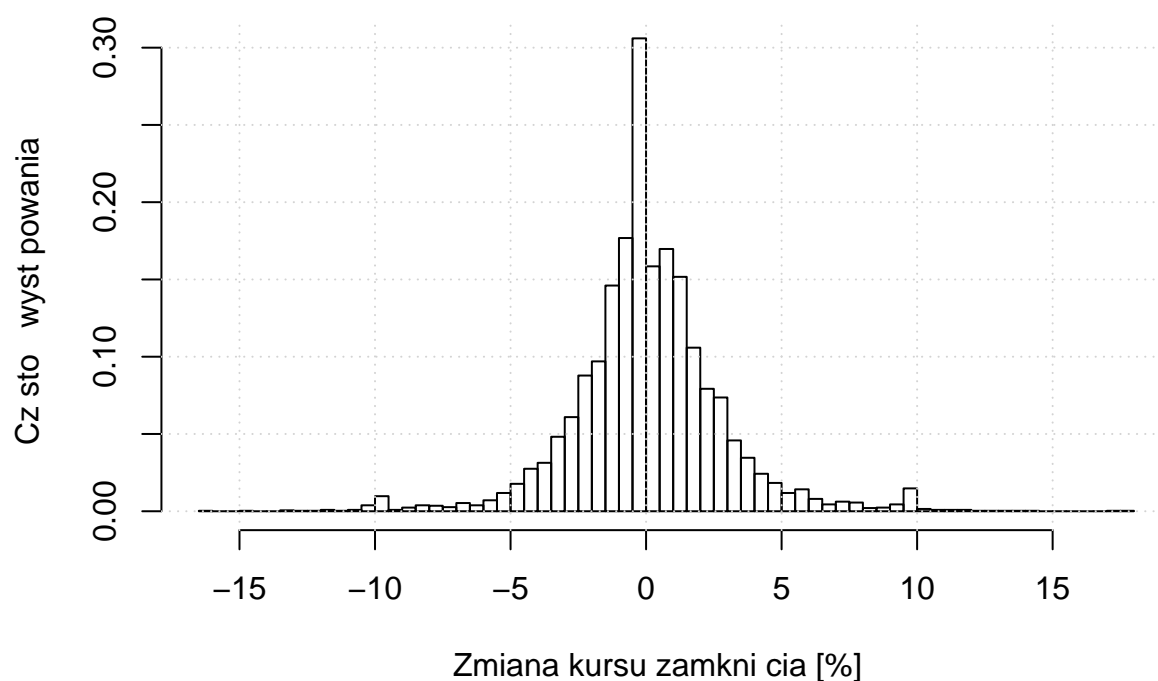
```
df_MBANK$date = as.Date.character(df_MBANK$date, format = '%Y%m%d')
df_MBANK$close_ch= with(df_MBANK, c(NA, 100*diff(close)/close[-length(close)]))
plot(close_ch ~ date, df_MBANK, type = 'l', col = 'blue', xlab = 'Data',
      ylab = 'Procentowa zmiana kursu zamknięcia [%]', main = 'MBANK' )
grid()
```



- Histogram procentowych zmian kursu spółki MBANK:

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

Histogram procentowych zmian kursu MBANK

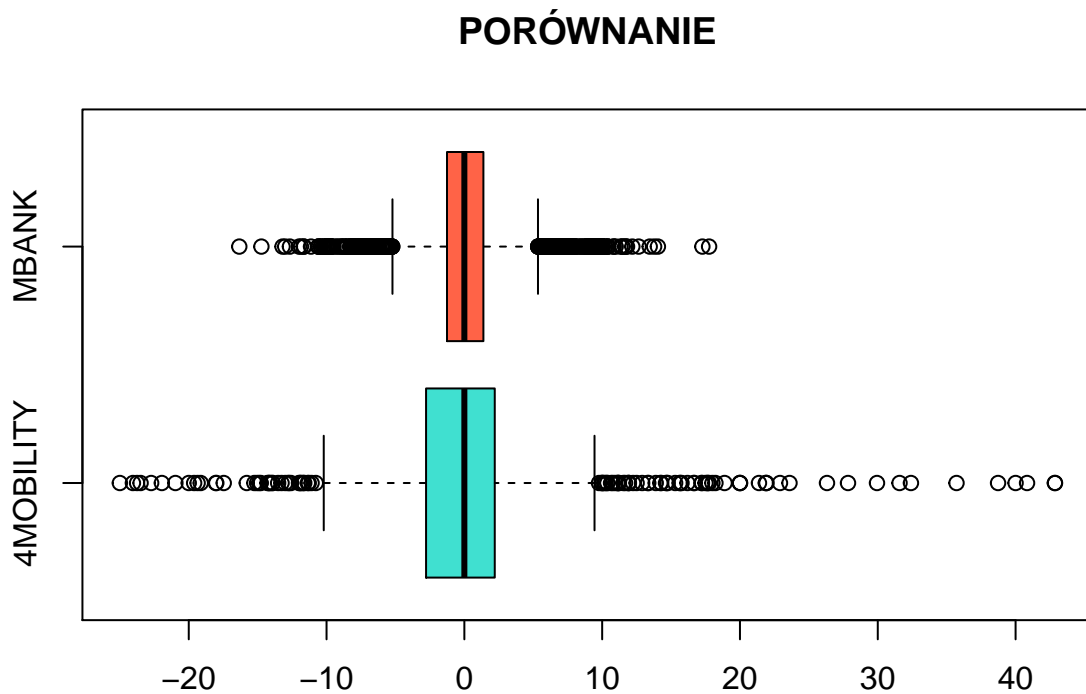


*Przypisuję wartość close spółki MBANK do zmiennej SP2:

```
SP2 <- (df_MBANK$close_ch= with(df_MBANK, c(NA, 100*diff(close)/close[-length(close)])))
```

- Próbuję obie spółki na jednym wykresie w formie pudełkowej:

```
boxplot(SP1, SP2,  
main = 'PORÓWNANIE',  
horizontal = TRUE,  
names = c("4MOBILITY", "MBANK"),  
col = c('turquoise', 'tomato')  
)
```



Zadanie nr 2

TREŚĆ ZADANIA:

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

ROZWIĄZANIE:

- Pobieram i zapisuję plik crashes.csv do katalogu ADPS/Lab1:

```
if(!file.exists('crashes.csv')) {
  download.file('http://elektron.elka.pw.edu.pl/~mrupniew/adps/crashes.csv', 'crashes.csv')
}
```

- Wczytuję dane do środowiska R:

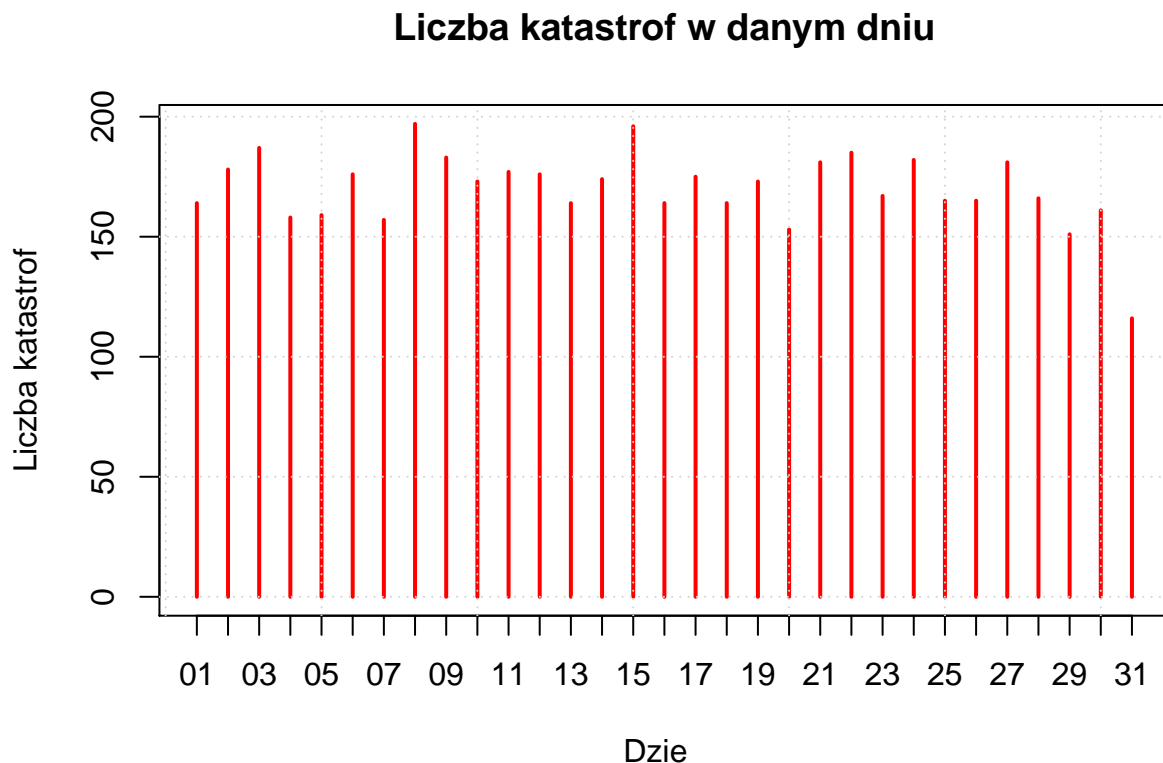
```
kat = read.csv('crashes.csv')
```

- Dodaję do danych kolumny z dniem:

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

- Prezentuję wykres liczby wypadków w danym dniu:

```
plot(table(kat$Day), type = 'h', col = 'red', xlab = 'Dzień',  
ylab = 'Liczba katastrof', main = 'Liczba katastrof w danym dniu')  
grid()
```

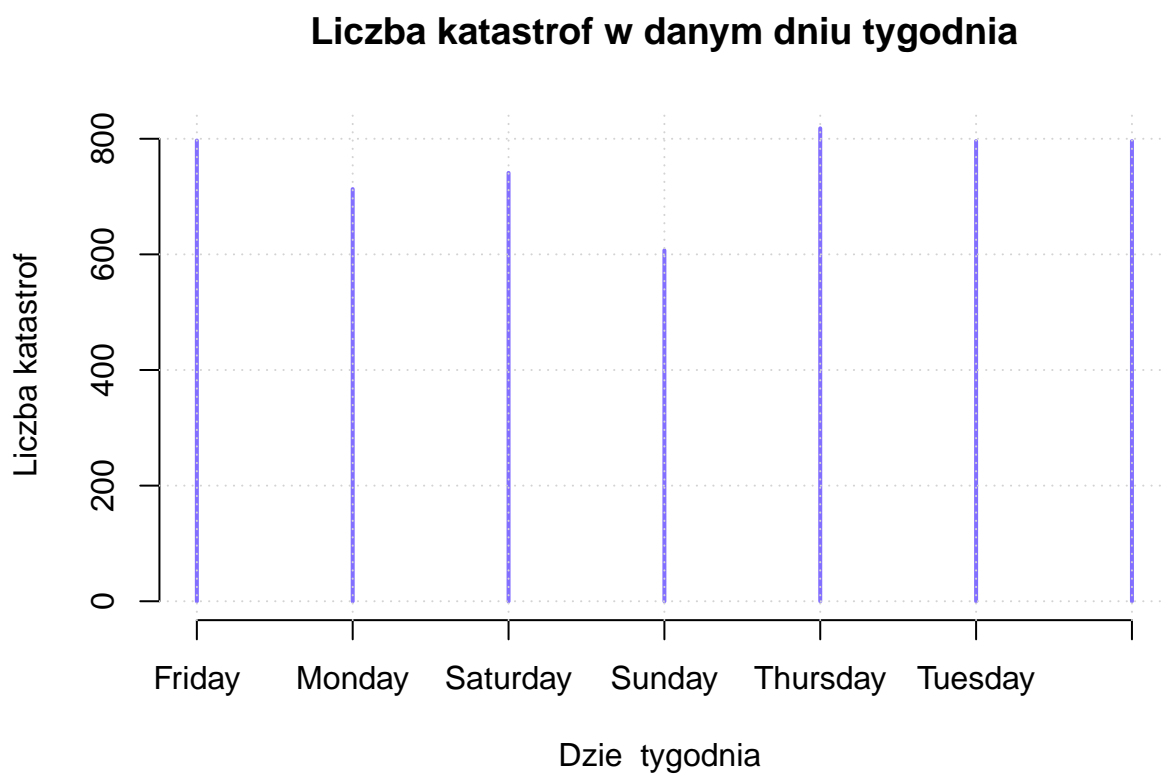


- Tworzę zmienną WeekDay. Dodaję do danych kolumny z dniem tygodnia:

```
kat$WeekDay <- strftime(as.Date(kat$Date, '%m/%d/%Y'), '%d')  
kat$WeekDay = strftime(as.Date(kat$Date, '%m/%d/%Y'), '%A')
```

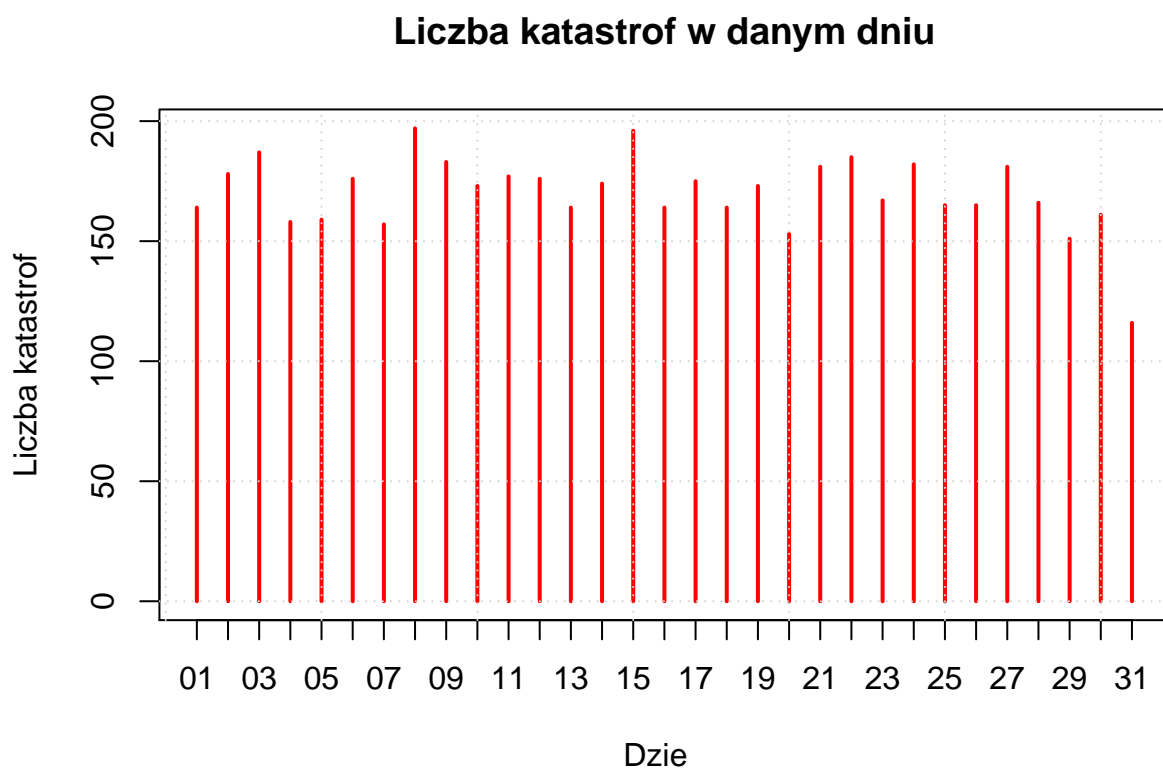
- Prezentuję wykres liczby wypadków w danym dniu tygodnia:

```
plot(table(kat$WeekDay), type = 'h', col = 'slateblue1', xlab = 'Dzień tygodnia',  
ylab = 'Liczba katastrof', main = 'Liczba katastrof w danym dniu tygodnia')  
grid()
```



- Prezentuję wykres liczby wypadków w danym dniu:

```
plot(table(kat$Day), type = 'h', col = 'red', xlab = 'Dzień',  
ylab = 'Liczba katastrof', main = 'Liczba katastrof w danym dniu')  
grid()
```

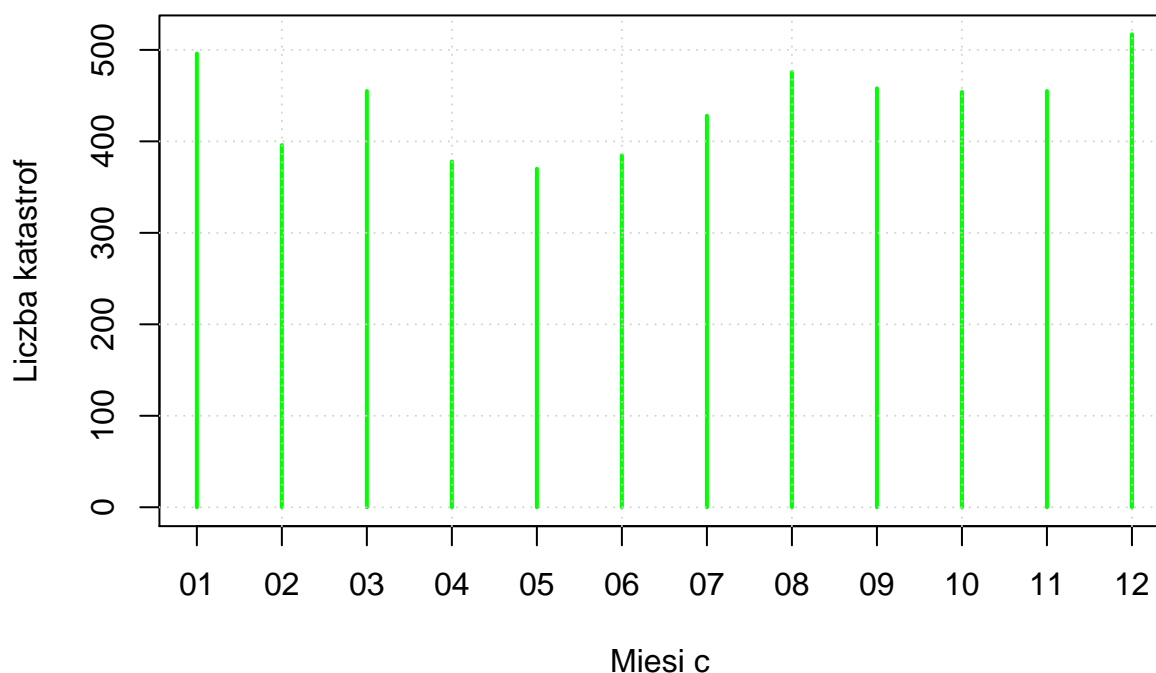
- Dodaję do danych kolumny z miesiącem:

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

- Prezentuję wykres liczby wypadków w danym miesiącu:

```
plot(table(kat$Month), type = 'h', col = 'green', xlab = 'Miesiąc',
ylab = 'Liczba katastrof', main = 'Liczba katastrof w danym miesiącu')
grid()
```

Liczba katastrof w danym miesi cu



- Dodaje do danych kolumny z rokiem:

```
kat$Year = strptime(as.Date(kat$Date, '%m/%d/%Y'), '%Y')
```

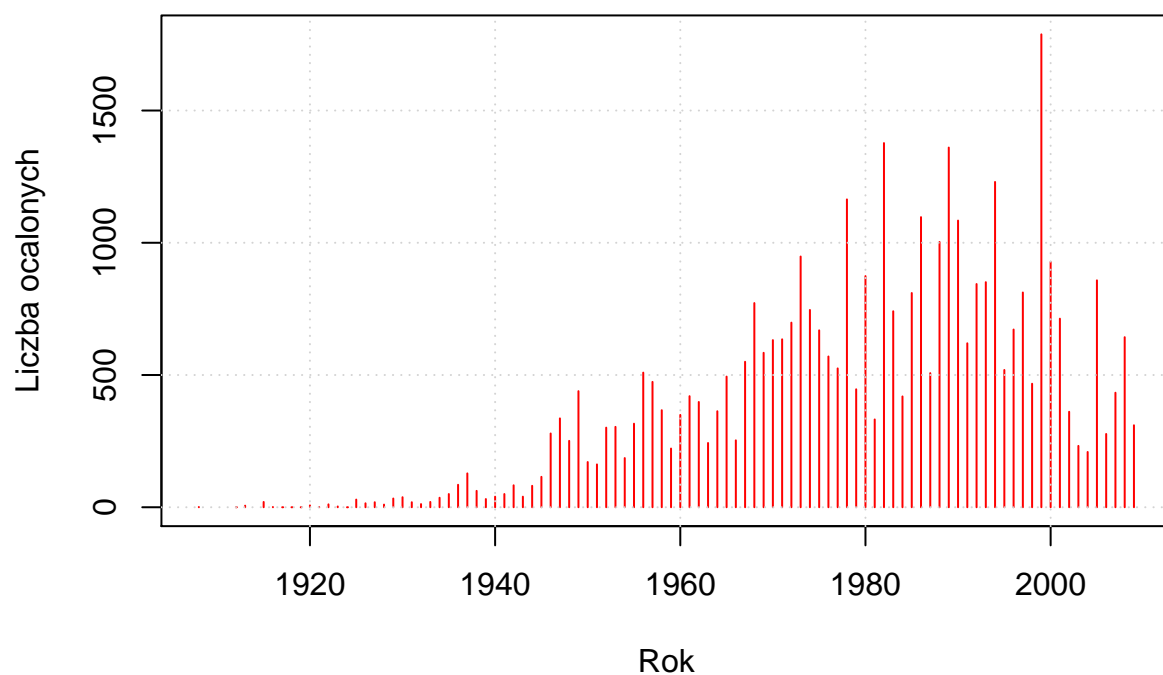
- Tworzę kategorię Saved i agreguję liczbę ocalałych osób:

```
kat$Saved = (kat$Aboard - kat$Fatalities)
Saved_agr = aggregate((Aboard-Fatalities) ~ Year, kat, FUN = sum)
Saved_perc = aggregate((Aboard-Fatalities)/Aboard ~ Year, kat, FUN = sum)
```

- Agreguję liczbę ocalałych osób w poszczególnych latach:

```
plot(Saved_agr, type = 'h', col = 'red', xlab = 'Rok',
ylab = 'Liczba ocalałych', main = 'Liczba osób ocalałych z katastrof w roku')
grid()
```

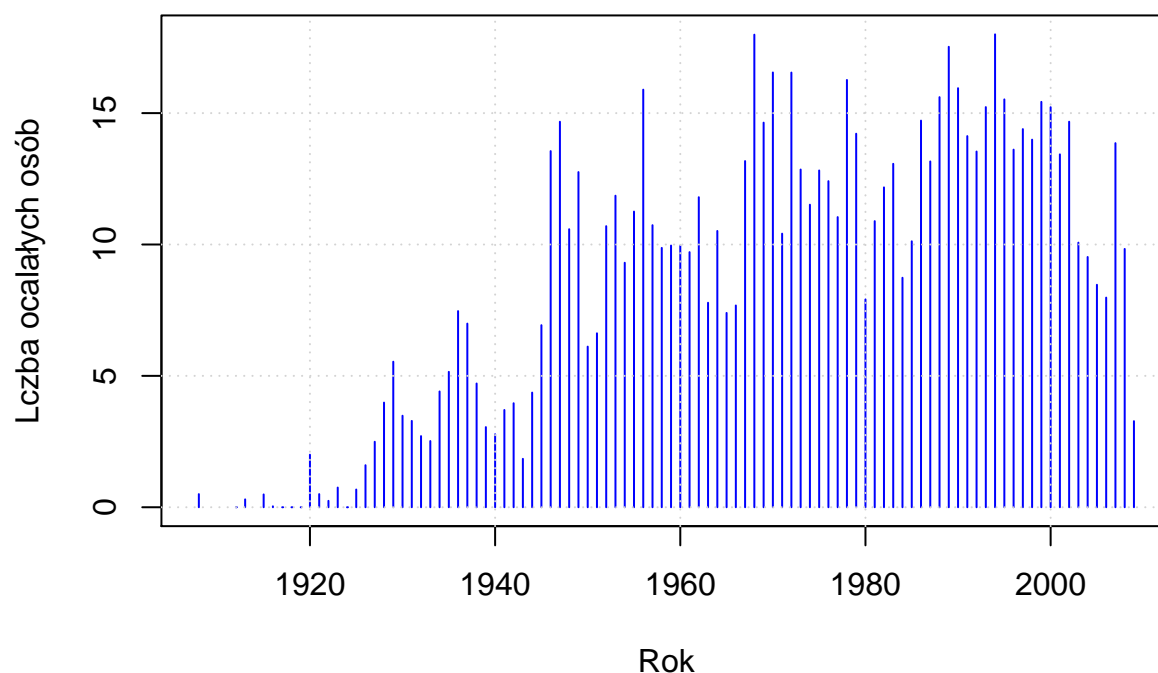
Liczba osób ocalałych z katastrof w roku



- Wykres procentowych liczby osób ocalałych z katastrof w roku:

```
plot(Saved_perc, type = 'h', col = 'blue', xlab = 'Rok',  
ylab = 'Liczba ocalałych osób', main = 'Liczba ocalałych osób w roku')  
grid()
```

Liczba ocalałych osób w roku



Zadanie nr 3

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 poszczególnych rozkładów (zestawów parametrów) narysuj na jednym rysunku empiryczne i teoretyczne (`dbinom`) funkcje prawdopodobieństwa, a na drugim rysunku empiryczne i teoretyczne (`pbinom`) dystrybuanty. W obu przypadkach wyskaluj oś odciętych od 0 do 20.

ROZWIĄZANIE

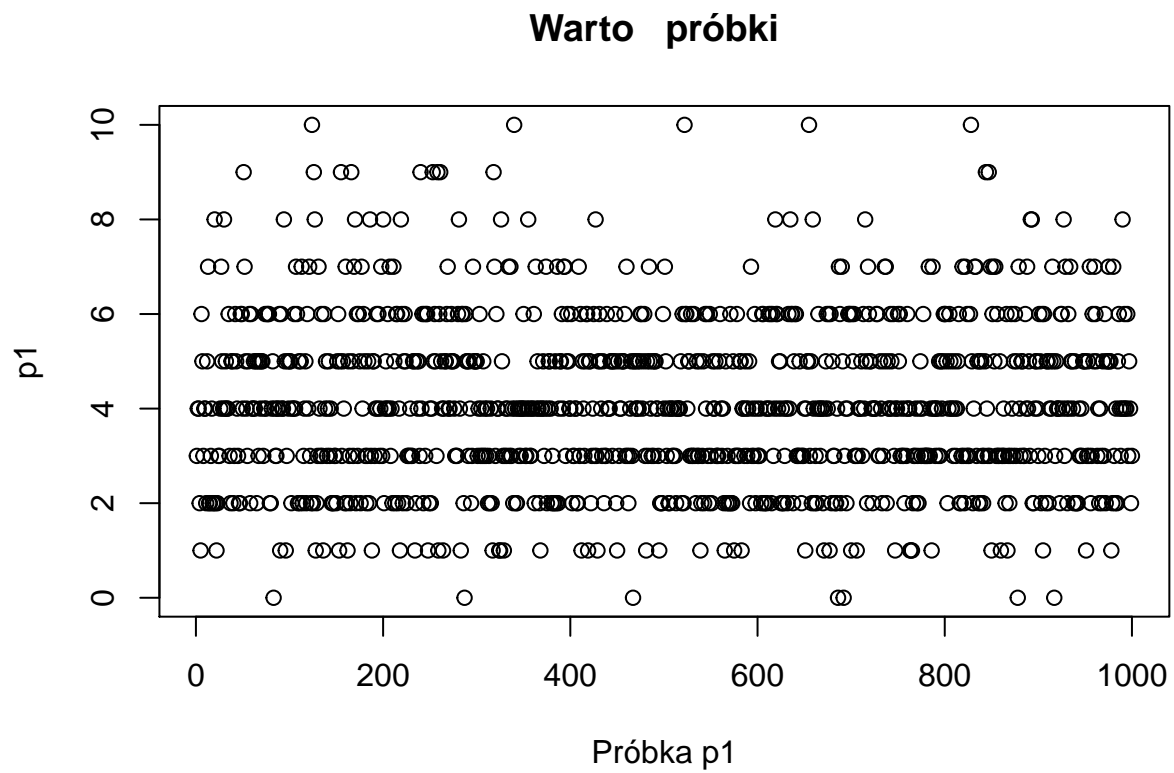
AD. 1

- $M = 1000$ próbek, rozkłady dwumianowe `Binom(20, 0.2)` i `Binom(20, 0.8)`:

```
p1 = rbinom(1000, size = 20, prob = 0.2)
p2 = rbinom(1000, size = 20, prob = 0.8)
```

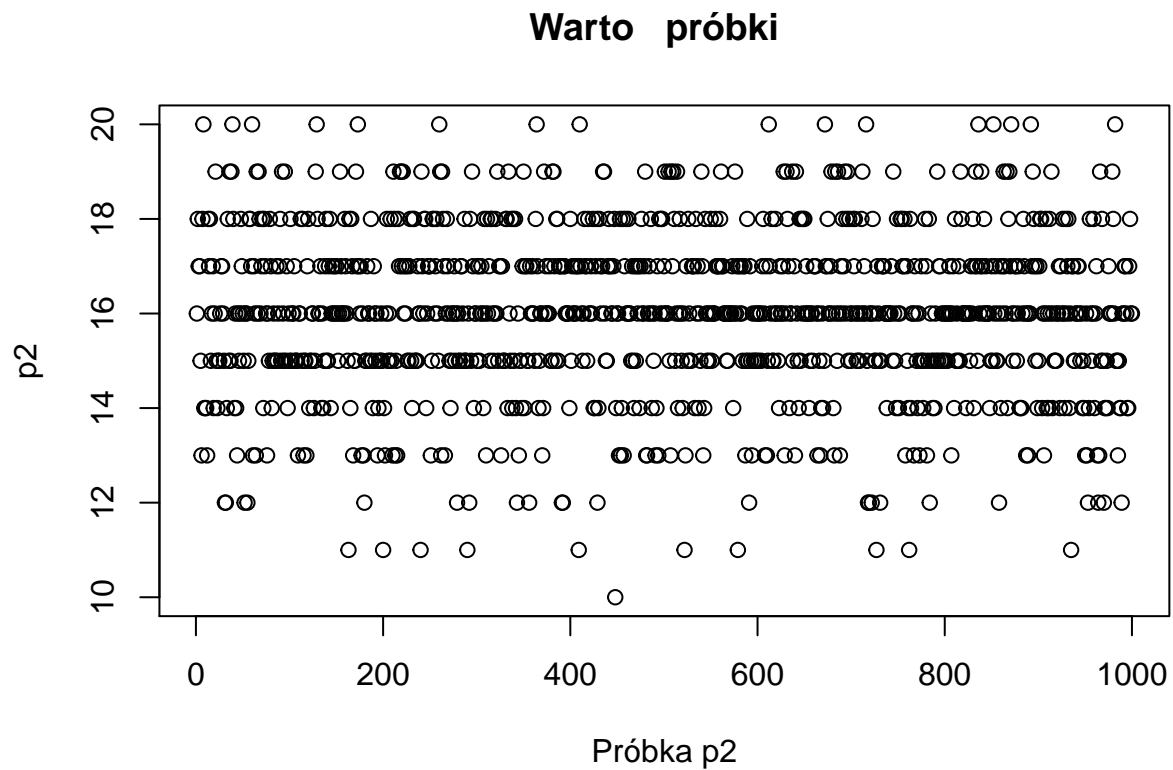
- Wykres pierwszej próbki - `p1`:

```
plot(p1, main = 'Wartość próbki', xlab = 'Próbka p1')
```



- Wykres drugiej próbki - p2:

```
plot(p2, main = 'Wartość próbki', xlab = 'Próbka p2')
```



AD. 2

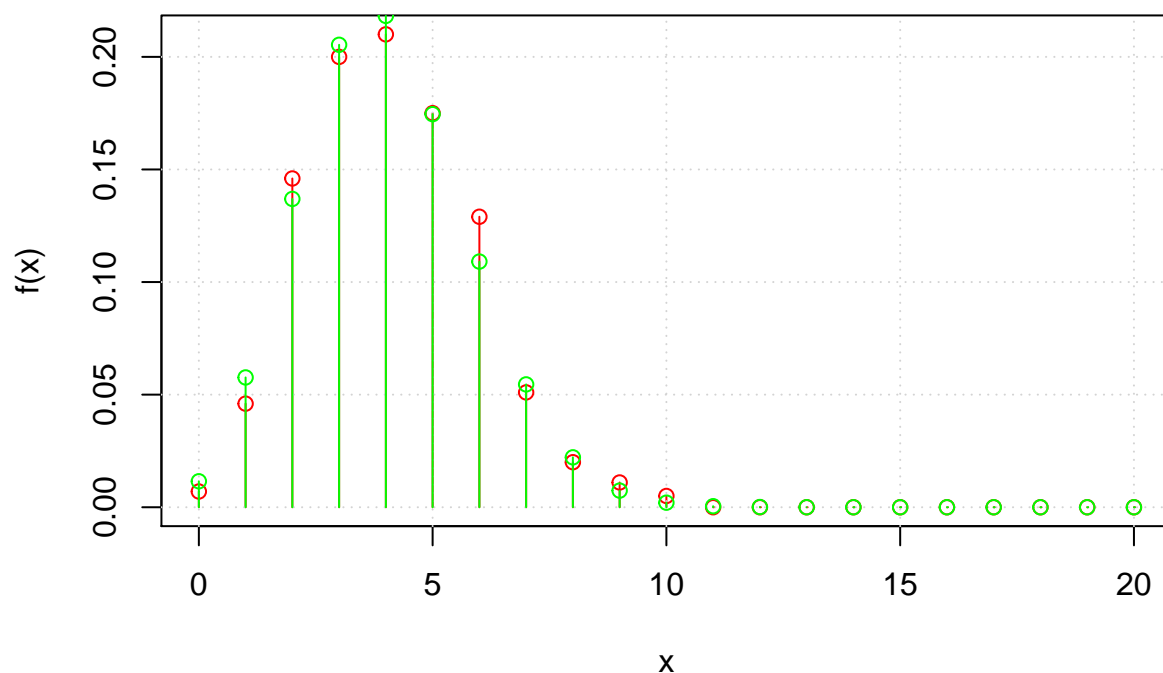
- Funkcja prawdopodobieństwa dla próbki p1:

```
Arg = 0:20
M = 1000
Arg = 0:20
Freq = as.numeric(table(factor(p1, levels = Arg))) / M
plot(Freq ~ Arg, type = 'h', col = 'red', xlab = 'x',
ylab = 'f(x)',
main = paste0('Funkcja prawdopodobieństwa dla p1'))

grid()
points(Freq ~ Arg, col = 'red')

lines(dbinom(Arg,20,0.2) ~ Arg, type = 'h', col = 'green', xlab = 'wartość x',
ylab = 'f(x)')
points(dbinom(Arg,20,0.2) ~ Arg, col = 'green')
```

Funkcja prawdopodobieństwa dla p1

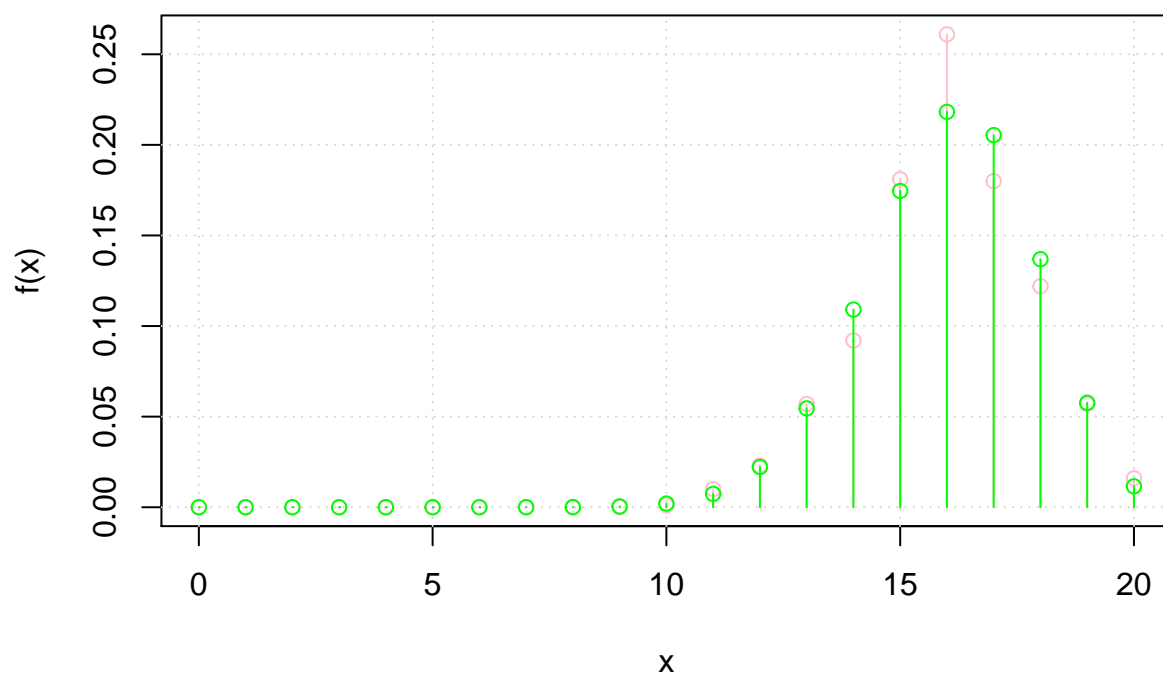


- Funkcja prawdopodobieństwa dla próbki p2:

```
Arg = 0:20
M = 1000
Freq = as.numeric(table(factor(p2, levels = Arg))) / M
plot(Freq ~ Arg, type = 'h', col = 'pink', xlab = 'x', ylab = 'f(x)',
main = paste0('Funkcja prawdopodobieństwa dla p2'))
grid()
points(Freq ~ Arg, col = 'pink')

lines(dbinom(Arg,20,0.8) ~ Arg, type = 'h', col = 'green', xlab = 'wartość x',
ylab = 'f(x)')
points(dbinom(Arg,20,0.8) ~ Arg, col = 'green')
```

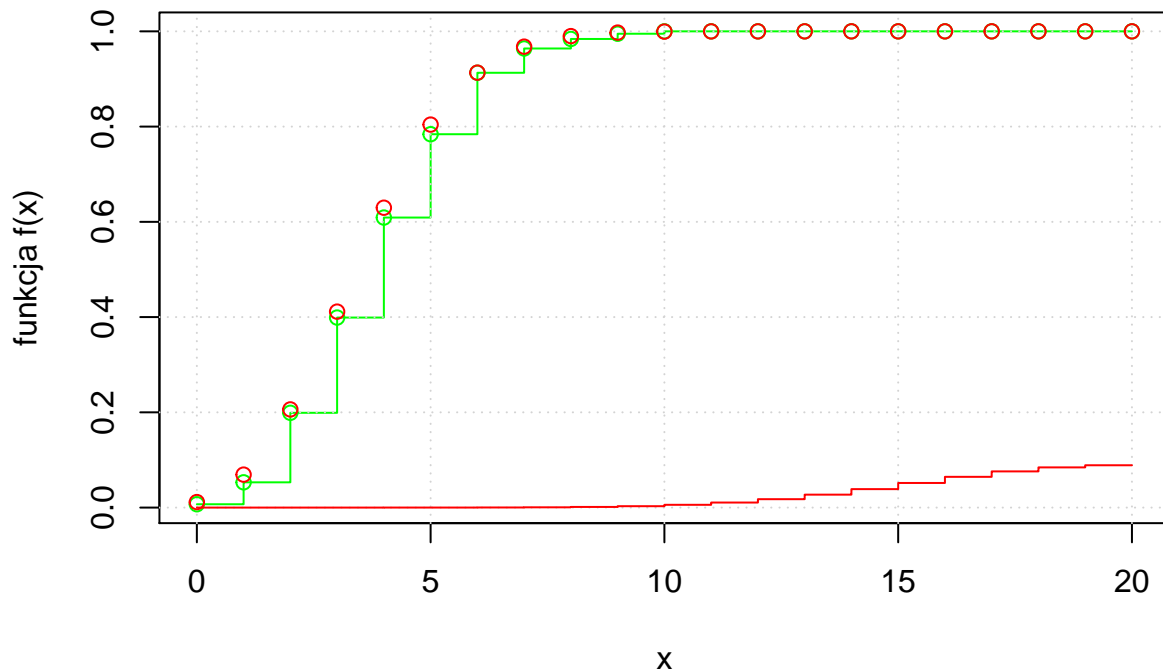
Funkcja prawdopodobieństwa dla p2



- Wyznaczam dystrybucję teoretyczną i empiryczną dla próbki p1:

```
Arg = 0:20
Freq = as.numeric(table(factor(p1, levels = Arg))) / M
plot(cumsum(Freq) ~ Arg, type = 's', col = 'green', xlab = 'x',
     ylab = 'funkcja f(x)',
     main = 'Dystrybucja empiryczna i teoretyczna dla próbki p1')
grid()
points(cumsum(Freq) ~ Arg, col = 'green')
lines(dpois(Arg,20,0.2) ~ Arg, type = 's', col = 'red',
     xlab = 'wartość x', ylab = 'funkcja f(x)')
points(pbinom(Arg,20,0.2) ~ Arg, col = 'red')
```

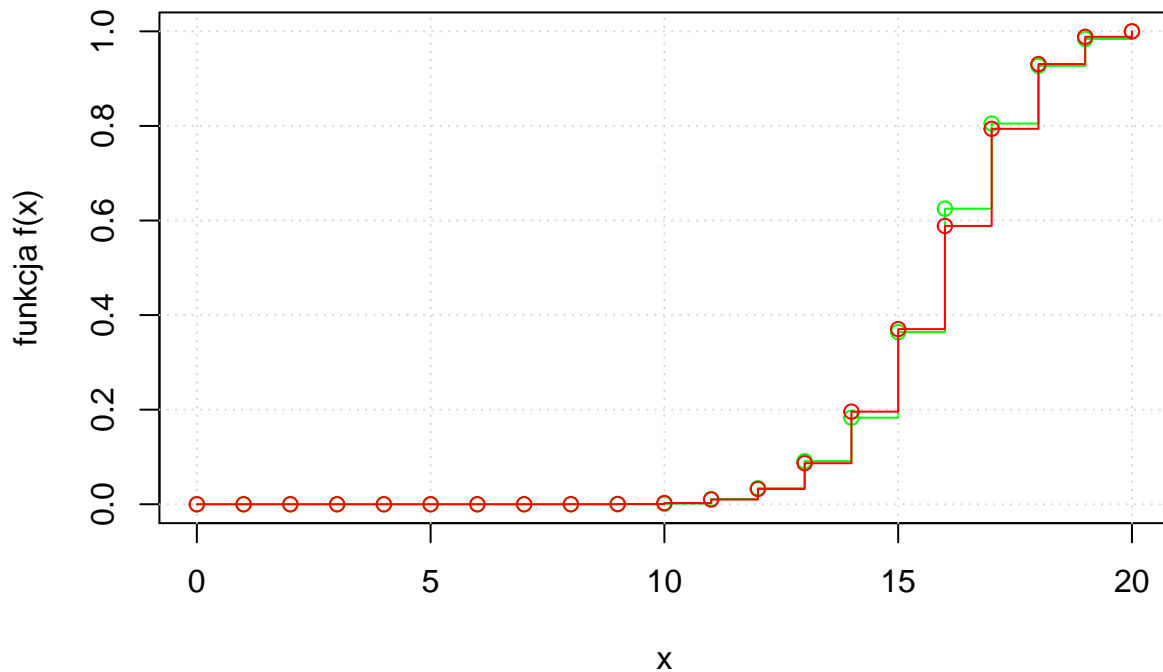

Dystrybuanta empiryczna i teoretyczna dla próbki p1



- Wyznaczam dystrybuantę teoretyczną i empiryczną dla próbki p2:

```
Arg = 0:20
Freq = as.numeric(table(factor(p2, levels = Arg))) / M
plot(cumsum(Freq) ~ Arg, type = 's', col = 'green', xlab = 'x',
     ylab = 'funkcja f(x)',
     main = 'Dystrybuanta empiryczna i teoretyczna dla próbki p2')
grid()
points(cumsum(Freq) ~ Arg, col = 'green')
lines(pbinom(Arg,20,0.8) ~ Arg, type = 's', col = 'red',
     xlab = 'wartość x', ylab = 'funkcja f(x)')
points(pbinom(Arg,20,0.8) ~ Arg, col = 'red')
```

Dystrybuanta empiryczna i teoretyczna dla próbki p2



Zadanie nr 4

TREŚĆ ZADANIA:

1. Dla rozkładu dwumianowego $\text{Binom}(20, 0.8)$ wygeneruj trzy próby losowe składające się z $M = 100$, $M = 1000$ i $M = 10000$ 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 z wartościami teoretycznymi dla rozkładu $\text{Binom}(20, 0.8)$.

ROZWIĄZANIE

AD. 1

- Próba losowa dla $M = 100$ próbek

```
p3 = rbinom(100, size = 20, prob = 0.8)
```

- Próba losowa dla $M = 1000$ próbek

```
p4 = rbinom(1000, size = 20, prob = 0.8)
```

- Próba losowa dla $M = 10000$ próbek

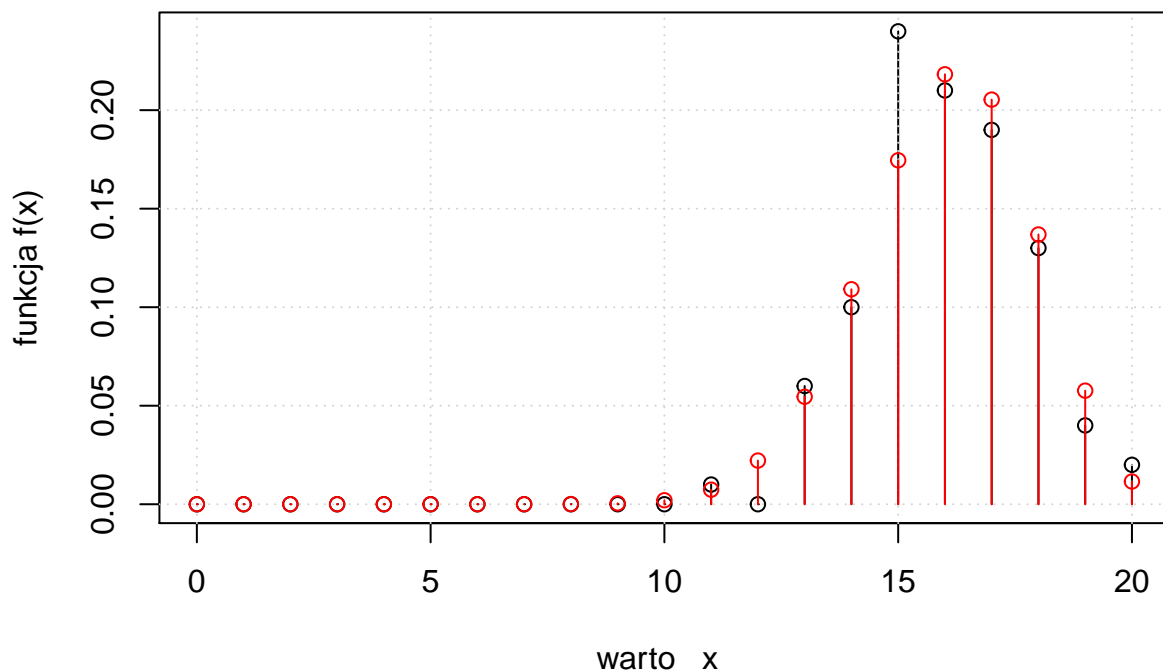
```
p5 = rbinom(10000, size = 20, prob = 0.8)
```

AD. 2

- Funkcja prawdopodobieństwa (empiryczna i teoretyczna) $M = 100$ dla próbki p3:

```
M=100
Arg = 0:max(p3)
Freq = as.numeric(table(factor(p3, levels = Arg))) / M
plot(Freq ~ Arg, type = 'h', col = 'black', xlab = 'wartość x',
     ylab = 'funkcja f(x)',
     main = 'Funkcje prawdopodobieństwa dla M = 100 próbki p3 ')
grid()
points(Freq ~ Arg, col = 'black')
lines(dbinom(Arg, size = 20, prob = 0.8) ~ Arg, type = 'h', col = 'red',
     xlab = 'wartość x', ylab = 'funkcja f(x)')
points(dbinom(Arg, size = 20, prob = 0.8) ~ Arg, col = 'red')
```

Funkcje prawdopodobieństwa dla $M = 100$ próbki p3



- Funkcja prawdopodobieństwa (empiryczna i teoretyczna) $M = 1000$ dla próbki p3:

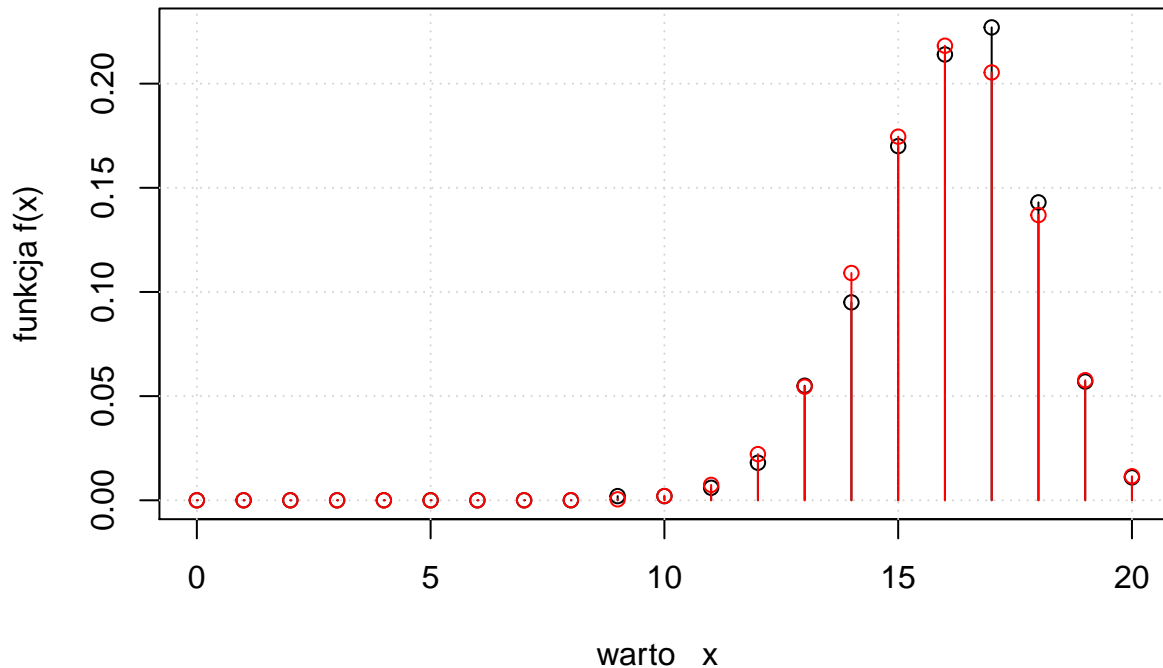
```
M=1000
Arg = 0:max(p4)
Freq = as.numeric(table(factor(p4, levels = Arg))) / M
plot(Freq ~ Arg, type = 'h', col = 'black', xlab = 'wartość x',
     ylab = 'funkcja f(x)',
     main = 'Funkcje prawdopodobieństwa dla M = 1000 próbki p4 ')
grid()
```

```

grid()
points(Freq ~ Arg, col = 'black')
lines(dbinom(Arg, size = 20, prob = 0.8) ~ Arg, type = 'h', col = 'red',
xlab = 'wartość x', ylab = 'funkcja f(x)')
points(dbinom(Arg, size = 20, prob = 0.8) ~ Arg, col = 'red')

```

Funkcje prawdopodobieństwa dla M = 1000 próbek p4



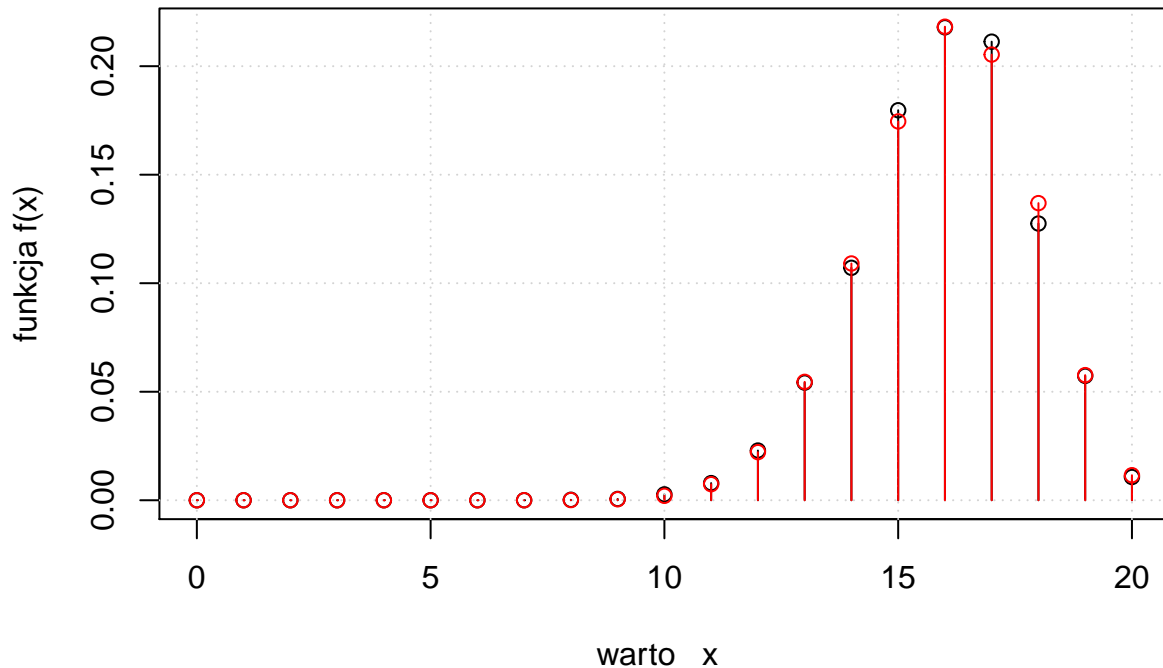
- Funkcja prawdopodobieństwa (empiryczna i teoretyczna) M = 100 dla próbek p3:

```

M=10000
Arg = 0:max(p5)
Freq = as.numeric(table(factor(p5, levels = Arg))) / M
plot(Freq ~ Arg, type = 'h', col = 'black', xlab = 'wartość x',
ylab = 'funkcja f(x)',
main = 'Funkcje prawdopodobieństwa dla M = 10000 próbek p5 ')
grid()
points(Freq ~ Arg, col = 'black')
lines(dbinom(Arg, size = 20, prob = 0.8) ~ Arg, type = 'h', col = 'red',
xlab = 'wartość x', ylab = 'funkcja f(x)')
points(dbinom(Arg, size = 20, prob = 0.8) ~ Arg, col = 'red')

```

Funkcje prawdopodobieństwa dla M = 10000 próbek p5

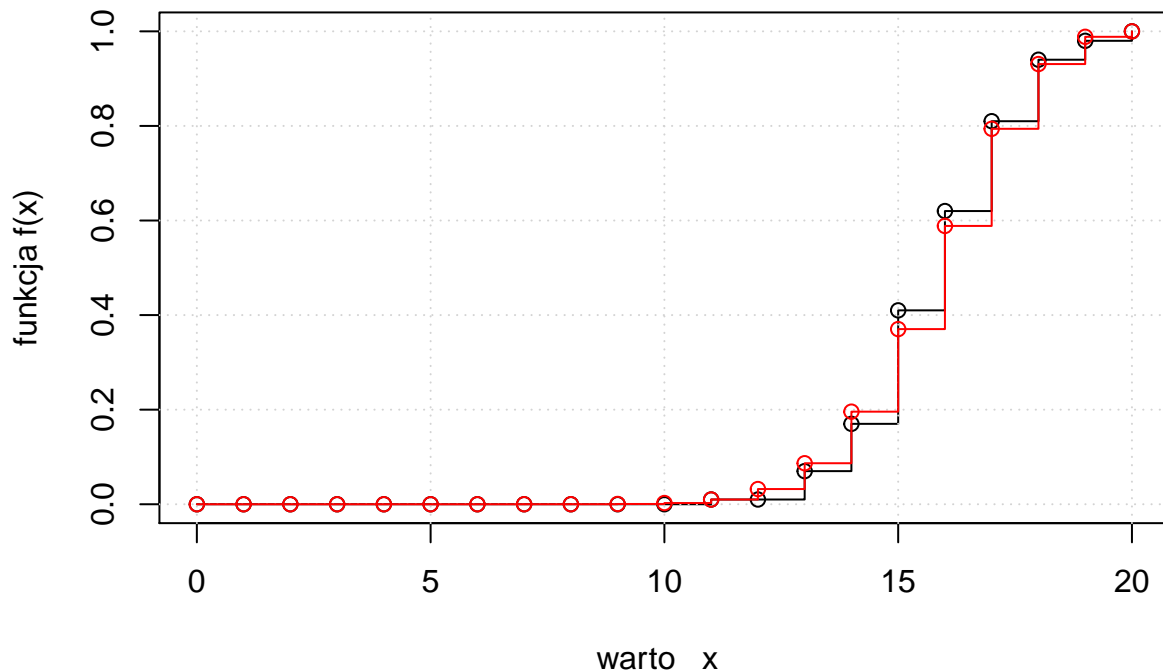


- Dystrybuanta (empiryczna i teoretyczna) M = 100 dla próbek p3:

```
M = 100
Arg = 0:max(p3)
Freq = as.numeric(table(factor(p3, levels = Arg))) / M
plot(cumsum(Freq) ~ Arg, type = 's', col = 'black',
     xlab = 'wartość x', ylab = 'funkcja f(x)',
     main = 'Dystrybuanta dla M = 100 próbek p3')
grid()
points(cumsum(Freq) ~ Arg, col = 'black')

lines(pbinom(Arg, size = 20, prob = 0.8) ~ Arg, type = 's', col = 'red',
     xlab = 'wartość x', ylab = 'funkcja f(x)')
points(pbinom(Arg, size = 20, prob = 0.8) ~ Arg, col = 'red')
```

Dystrybuanta dla M = 100 próbek p3

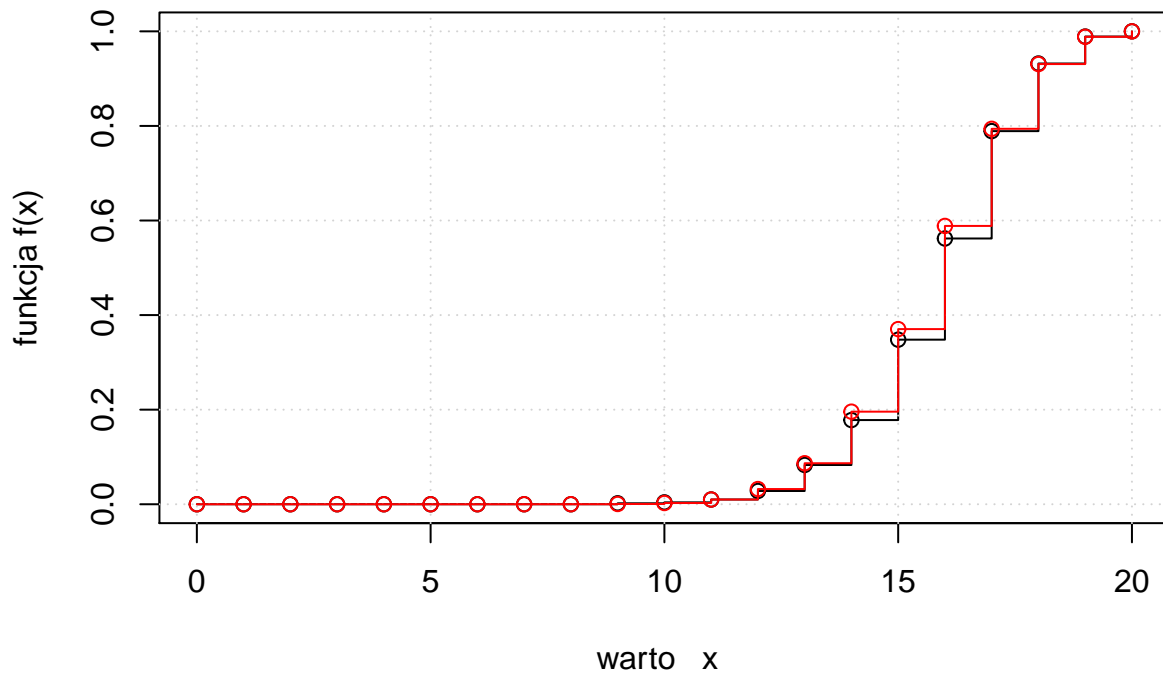


- Dystrybuanta (empiryczna i teoretyczna) M = 100 dla próbek p4:

```
M = 1000
Arg = 0:max(p4)
Freq = as.numeric(table(factor(p4, levels = Arg))) / M
plot(cumsum(Freq) ~ Arg, type = 's', col = 'black',
     xlab = 'wartość x', ylab = 'funkcja f(x)',
     main = 'Dystrybuanta dla M = 1000 próbek p4')
grid()
points(cumsum(Freq) ~ Arg, col = 'black')

lines(pbinom(Arg, size = 20, prob = 0.8) ~ Arg, type = 's', col = 'red',
     xlab = 'wartość x', ylab = 'funkcja f(x)')
points(pbinom(Arg, size = 20, prob = 0.8) ~ Arg, col = 'red')
```

Dystrybuanta dla M = 1000 próbek p4

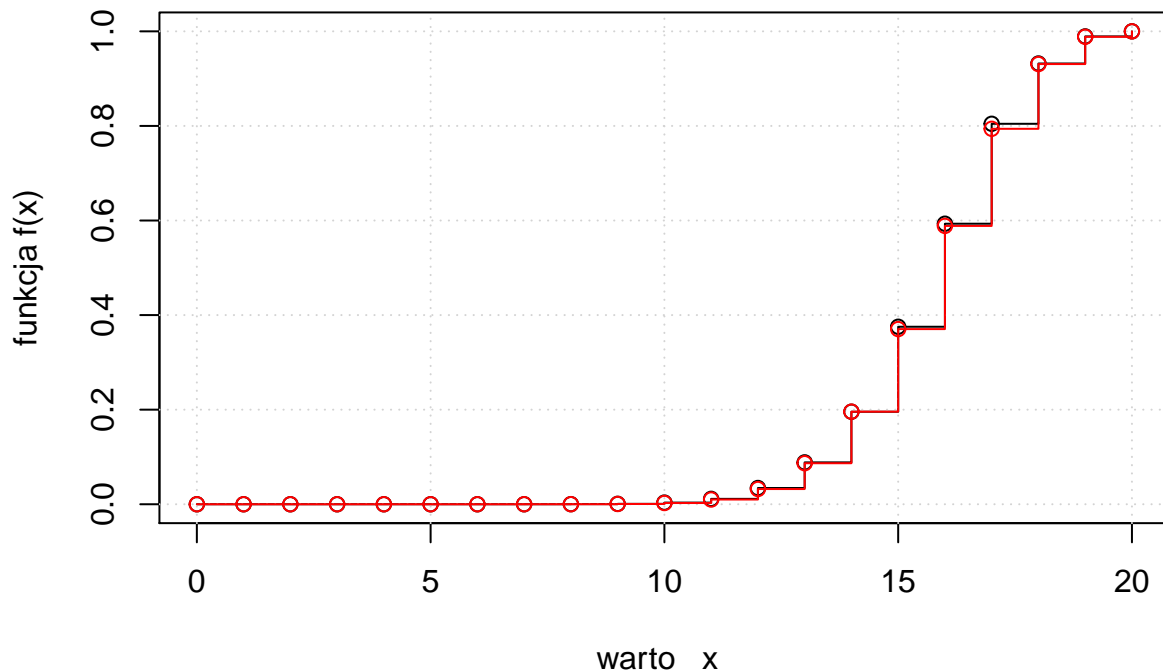


- Dystrybuanta (empiryczna i teoretyczna) M = 100 dla próbek p5:

```
M = 10000
Arg = 0:max(p5)
Freq = as.numeric(table(factor(p5, levels = Arg))) / M
plot(cumsum(Freq) ~ Arg, type = 's', col = 'black',
     xlab = 'wartość x', ylab = 'funkcja f(x)',
     main = 'Dystrybuanta dla M = 10000 próbek p5')
grid()
points(cumsum(Freq) ~ Arg, col = 'black')

lines(pbinom(Arg, size = 20, prob = 0.8) ~ Arg, type = 's', col = 'red',
     xlab = 'wartość x', ylab = 'funkcja f(x)')
points(pbinom(Arg, size = 20, prob = 0.8) ~ Arg, col = 'red')
```

Dystrybuanta dla M = 10000 próbek p5



AD. 3

*Obliczenia dla wartości średniej z prób p3, p4 i p5:

```
c(100,1000,10000)->M
c(mean(p3), mean(p4), mean(p5))->wartosc_oczekiwana
c(var(p3), var(p4), var(p5))->wariancja
data.frame(M,wartosc_oczekiwana,wariancja)->parametry
print(parametry)
```

```
##      M wartosc_oczekiwana wariancja
## 1   100             15.9800  2.848081
## 2  1000             16.0750  3.138514
## 3 10000             15.9714  3.206903
```

- Obliczam wartości teoretycznymi dla rozkładu $\text{Binom}(20, 0.8)$:

- 1) wartość oczekiwana: $EX = n * p$, stąd: $EX = 20 * 0.8 = 16$
- 2) wariancja: $\text{Var}X = n * p * (1 - p)$, stąd $\text{Var}X = 20 * 0.8 * 0.2 = 3.2$

- Wniosek: Im większa wartość M tym wynik wartości empirycznej bardziej zbliżony do wartości teoretycznej.

Zadanie nr 5

TREŚĆ ZADANIA:

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

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

ROZWIĄZANIE

AD. 1, 2

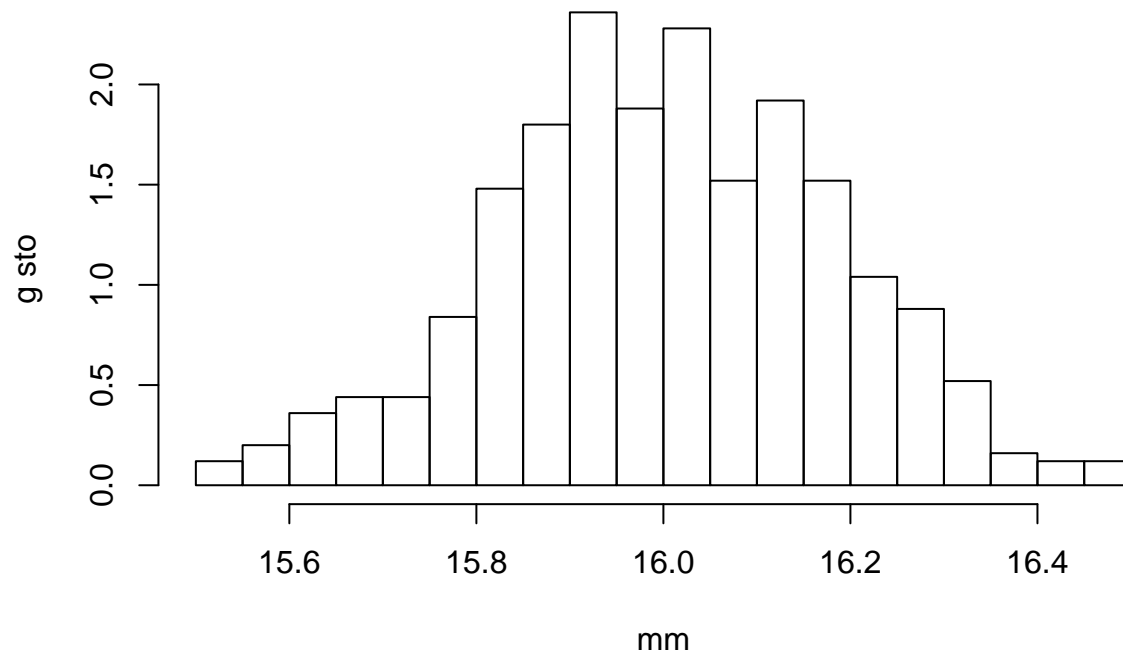
- Generuję $K = 500$ prób losowych złożonych z $M = 100$ próbek $\text{Binom}(20, 0.8)$, w tym wartość średnią i wariancję dla wszystkich realizacji:

```
M = 100
mm = replicate(500, mean(rbinom(M, 20, 0.8)))
mv = replicate(500, var(rbinom(M, 20, 0.8)))
```

- Tworzę histogram dla wartości średniej:

```
hist(mm, breaks = 20, prob = T, ylab = 'gęstość',
main = 'histogram wartości średniej z 500 prób dla M = 100')
```

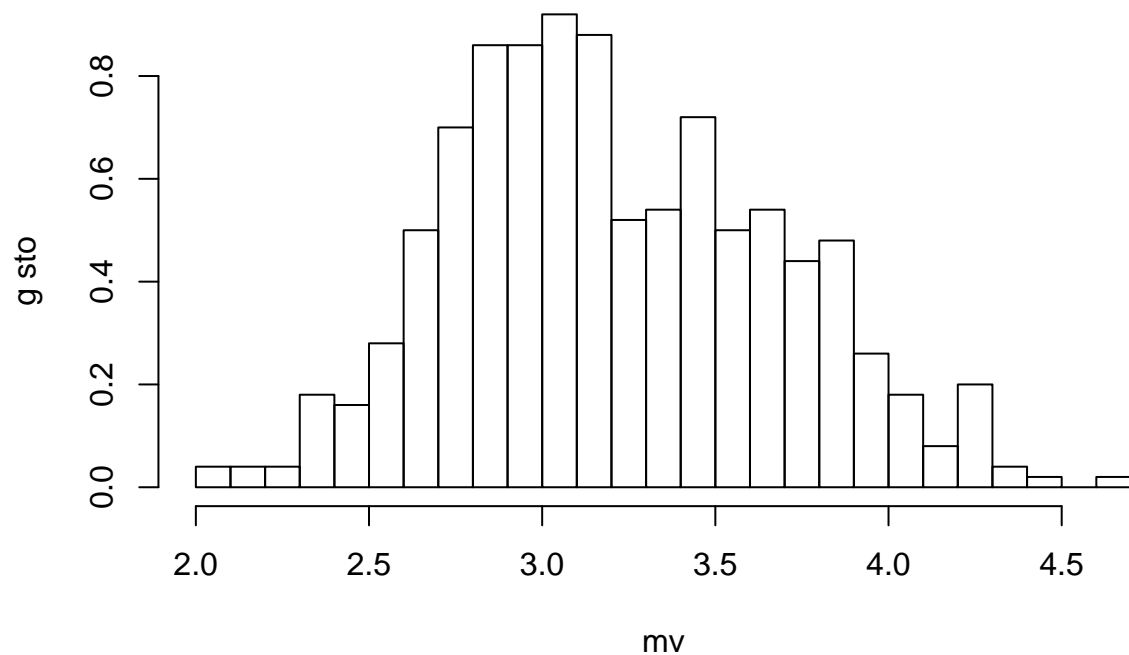
histogram wartości redniej z 500 prób dla $M = 100$



- Tworzę histogram dla wariancji:

```
hist(mv, breaks = 20, prob = T, ylab = 'gęstość',  
main = 'histogram wartości wariancji z 500 prób dla M = 100')
```

histogram wartości wariancji z 500 prób dla $M = 100$



AD. 3

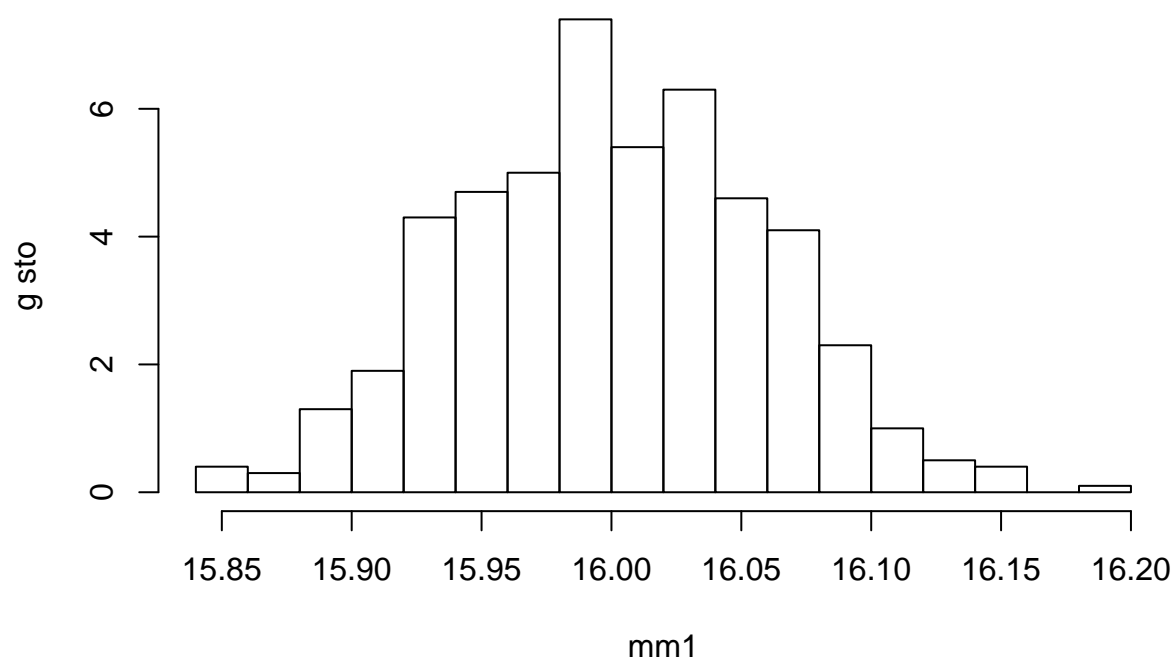
- Generuję $K = 500$ prób losowych złożonych z $M = 1000$ próbek $\text{Binom}(20, 0.8)$, w tym wartość średnią i wariancję dla wszystkich realizacji:

```
M = 1000
mm1 = replicate(500, mean(rbinom(M, 20, 0.8)))
mv1 = replicate(500, var(rbinom(M, 20, 0.8)))
```

- Tworzę histogram dla wartości średniej:

```
hist(mm1, breaks = 20, prob = T, ylab = 'gęstość',
main = 'histogram wartości średniej z 500 prób dla M = 1000')
```

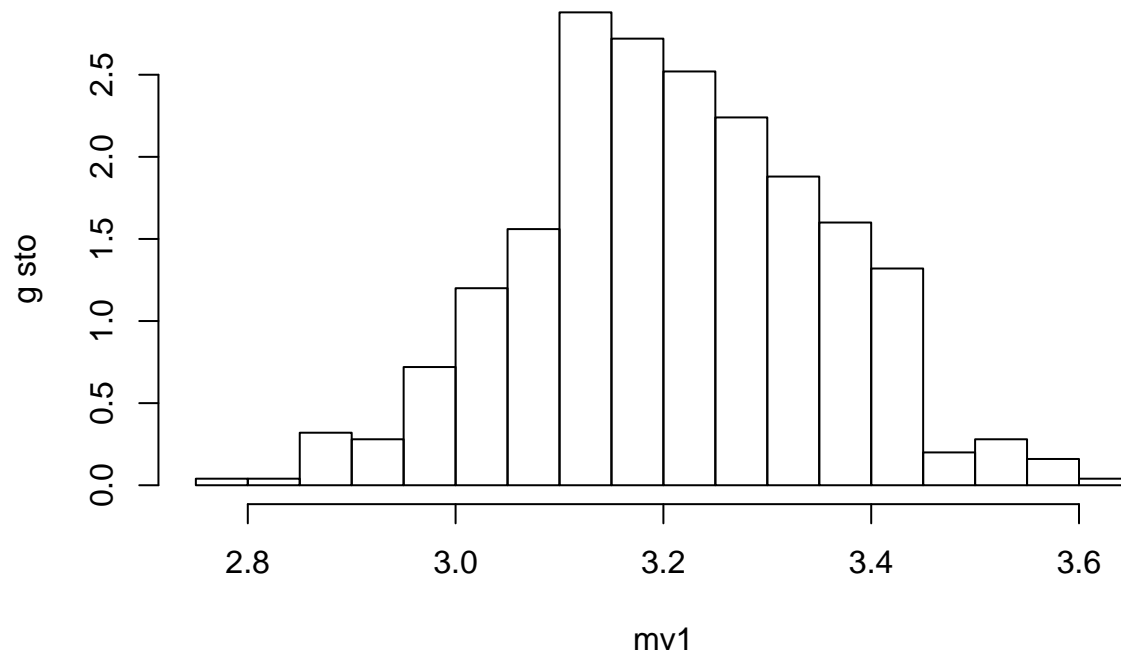
histogram wartości redniej z 500 prób dla $M = 1000$



- Tworzę histogram dla wariancji:

```
hist(mv1, breaks = 20, prob = T, ylab = 'gęstość',  
main = 'histogram wartości wariancji z 500 prób dla M = 1000')
```

histogram wartości wariancji z 500 prób dla $M = 1000$



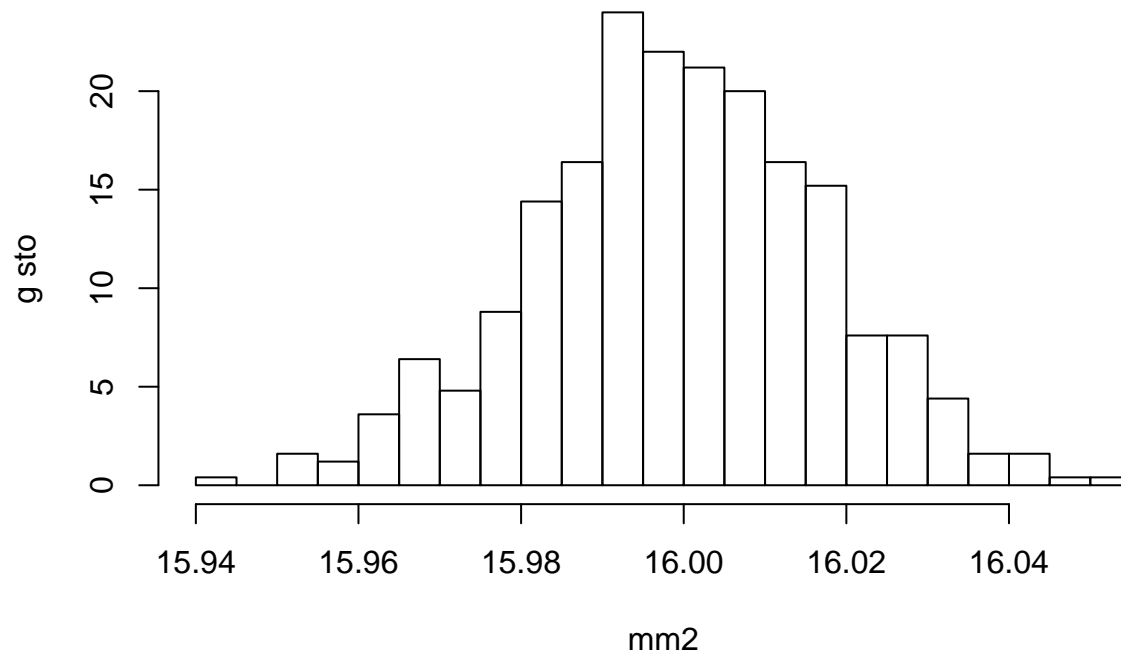
- Generuję $K = 500$ prób losowych złożonych z $M = 1000$ próbek $\text{Binom}(20, 0.8)$, w tym wartość średnią i wariancję dla wszystkich realizacji:

```
M = 10000  
mm2 = replicate(500, mean(rbinom(M, 20, 0.8)))  
mv2 = replicate(500, var(rbinom(M, 20, 0.8)))
```

- Tworzę histogram dla wartości średniej:

```
hist(mm2, breaks = 20, prob = T, ylab = 'gęstość',  
main = 'histogram wartości średniej z 500 prób dla M = 10000')
```

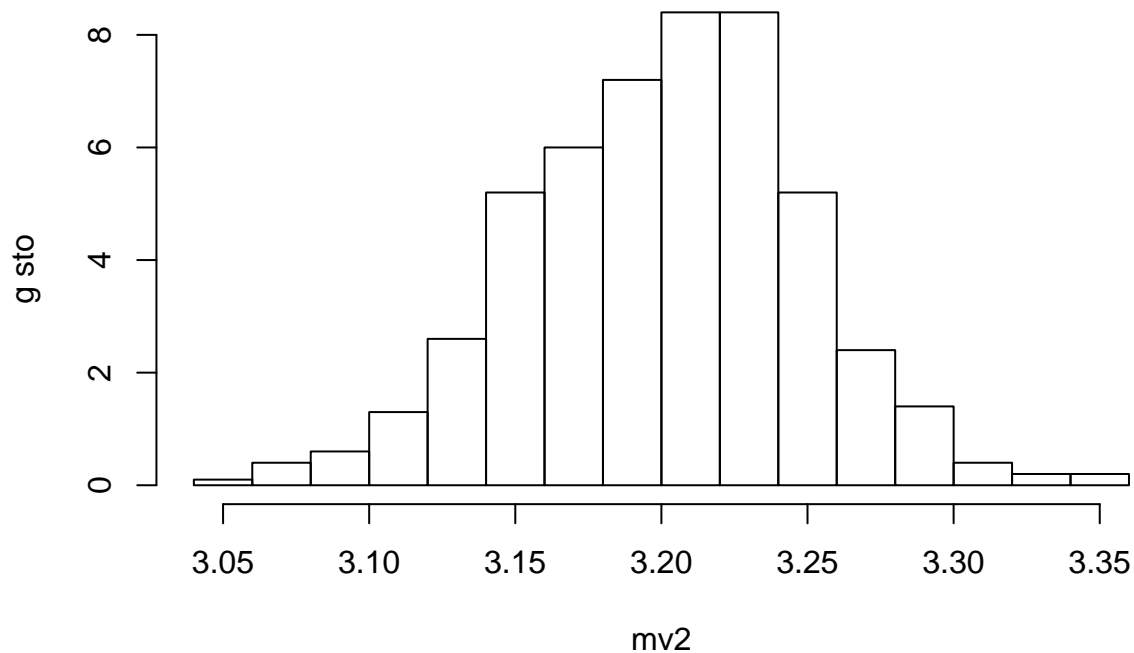
histogram wartości redniej z 500 prób dla M = 10000



- Tworzę histogram dla wariancji:

```
hist(mv2, breaks = 20, prob = T, ylab = 'gęstość',  
main = 'histogram wartości wariancji z 500 prób dla M = 10000')
```

histogram warto ci wariancji z 500 prób dla M = 10000



- Obliczenia (podsumowanie):

```
sredniaMM <- mean(mm)
wariancjaMV <- var(mv)

sredniaMM1 <- mean(mm1)
wariancjaMV1 <- var(mv1)

sredniaMM2 <- mean(mm2)
wariancjaMV2 <- var(mv2)

srednia <- c(sredniaMM, sredniaMM1, sredniaMM2)
wariancja <- c(wariancjaMV, wariancjaMV1, wariancjaMV2)

M = c(100,1000,10000)

PODSUMOWANIE <- data.frame(M, srednia, wariancja)
print(PODSUMOWANIE)
```

```
##      M  srednia  wariancja
## 1   100 16.00896 0.219469868
## 2   1000 16.00145 0.020687355
## 3 10000 15.99928 0.002287057
```

- Jak zmieniają się histogramy ze zmianą liczby próbek:

a) M = 100 dla średniej (mm):

skala x ma rozpiętość od 15.6 do 16.4 (różnica: 0.8)
skala y ma rozpiętość od 0.0 do 2.0 (różnica: 2.0)

dla wariancji (mv):

skala x ma rozpiętość od 2.0 do 4.5 (różnica: 2.5)
skala y ma rozpiętość od 0.0 do 0.8 (różnica: 0.8)

b) $M = 1000$ dla średniej (mm1):

skala x ma rozpiętość od 15.8 do 16.1 (różnica: 0.3)
skala y ma rozpiętość od 0.0 do 7.0 (różnica: 7.0)

dla wariancji (mv1):

skala x ma rozpiętość od 2.8 do 3.6 (różnica: 0.8)
skala y ma rozpiętość od 0.0 do 2.5 (różnica: 2.5)

c) $M = 10000$ dla średniej (mm2):

skala x ma rozpiętość od 15.94 do 16.06 (różnica: 0.12)
skala y ma rozpiętość od 0.0 do 20.0 (różnica: 20.0)

dla wariancji (mv2):

skala x ma rozpiętość od 3.10 do 3.35 (różnica: 0.25)
skala y ma rozpiętość od 0.0 do 10.0 (różnica: 10.0)

```
M = c(100,1000,10000)
srednie_podsumowanie <- c(0.8, 0.3, 0.12)
srednie_zageszczenie <- c(2.0, 7.0, 20.0)
wariancja_podsumowanie <- c(2.5, 0.8, 0.25)
wariancja_zageszczenie <- c(0.8, 2.5, 10.0)
```

```
PODSUMOWANIE_2 <- data.frame(M, srednie_podsumowanie, srednie_zageszczenie, wariancja_podsumowanie, wari  
print(PODSUMOWANIE_2)
```

```
##      M srednie_podsumowanie srednie_zageszczenie wariancja_podsumowanie
## 1   100                0.80                 2                2.50
## 2  1000                0.30                 7                0.80
## 3 10000                0.12                20                0.25
##   wariancja_zageszczenie
## 1                    0.8
## 2                    2.5
## 3                   10.0
```

- Wnioski: Zarówno w przypadku średniej jak i wariancji wraz ze wzrostem wartości M :
- wartości przedziałów na osi X maleją,
- wartości przedziałów na osi Y (gęstości) rosną.