ADPS 2025L — Laboratorium 1 (rozwiązania)

Dariusz Kopka

1 Zadanie 1 (1 pkt)

1.1 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.

1.2 Rozwiązanie

(ciąg dalszy na następnej stronie)

1.2.1 Funkcje pomocnicze

Funkcja 'helper' do pobierania danych z ograniczonym zakresem czasowym. Jeśli plik o tym samym zakresie danych został już pobrany *nie zostanie* pobrany ponownie.

```r
# Przykłady URLi:
# https://stooq.pl/q/d/l/?s=googl.us&i=d
# https://stooq.pl/q/d/l/?s=googl.us&f=20120819&t=20220314&i=d
get_stock <- function(ticker, start_date = NULL, end_date = NULL)  {
    base_url <- "https://stooq.com/q/d/l/"
    params <- list(s = ticker, i = "d")
    if (!is.null(start_date)) params$f <- format(as.Date(start_date), "%Y%m%d")
    if (!is.null(end_date)) params$t <-  format(as.Date(end_date), "%Y%m%d")

    query <- paste(names(params), params, sep = "=", collapse = "&")
    directory <- "data/"
    filename <- paste0(directory, gsub("[&]?[a-z]=", "_", query), ".csv")
    web_link <- paste0(base_url, "?", query)

    if (!file.exists(filename)) {
        dir.create(dirname(filename), showWarnings = FALSE, recursive = TRUE)
        download.file(web_link, filename)
    }
    stock_data = read.csv(filename)
    return(stock_data)
}
```

1.2.2 Dla danych z ostatnich 18 miesięcy dotyczących wybranych dwóch spółek giełdowych sporządź wykresy procentowych zmian kursów zamknięcia w zależności od daty

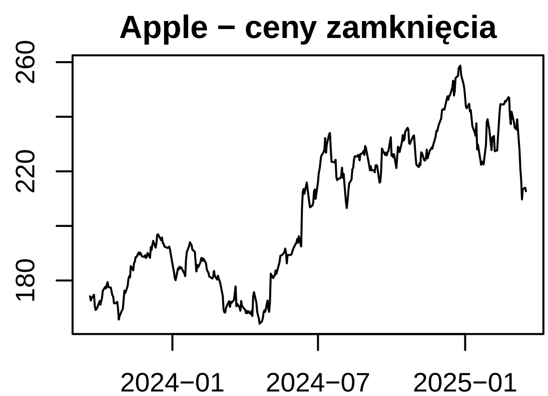
(rozwiązane razem z kolejnym zadaniem)

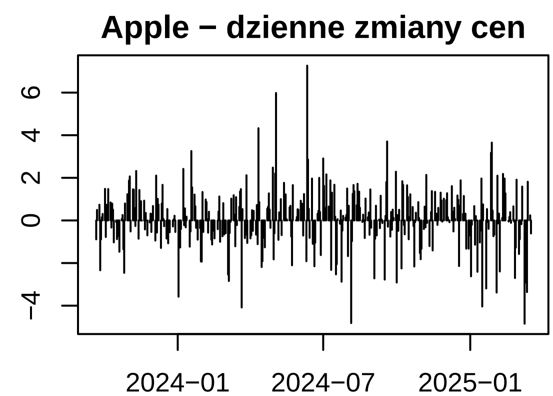
1.2.3 Dla danych z ostatnich 18 miesięcy dotyczących wybranych dwóch spółek giełdowych wykreśl i porównaj histogramy procentowych zmian kursów zamknięcia

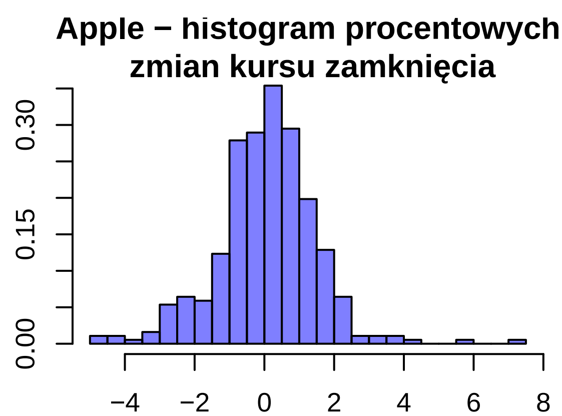
Ostatnie 18 miesięcy jest obliczone jako 78 tygodni (jeden i pół roku), ze względu na brak w `as.difftime` możliwości ustawienia `unit = months`.

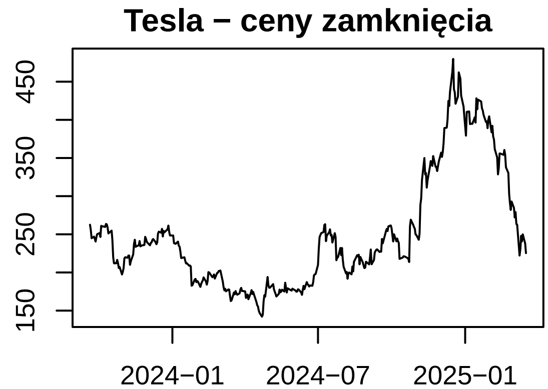
```r
# Pobieranie danych
since <- Sys.Date() - as.difftime(78, unit = "weeks")
until <- Sys.Date() # dziś

# Pobierz dane dla Apple i Tesli
tickers <- list(Apple = "aapl.us", Tesla = "tsla.us")
results <- lapply(tickers, get_stock, start_date = since, end_date = until)

graph_func <- function(df, name) {
    df$Date <- as.Date(df$Date)
    df$Return <- c(NA, 100 * diff(df$Close) / head(df$Close, -1))
    opar <- par(mfrow = c(2,2), mai = c(0.5, 0.5, 0.3, 0.3))
    plot(df$Date,
         df$Close,
```

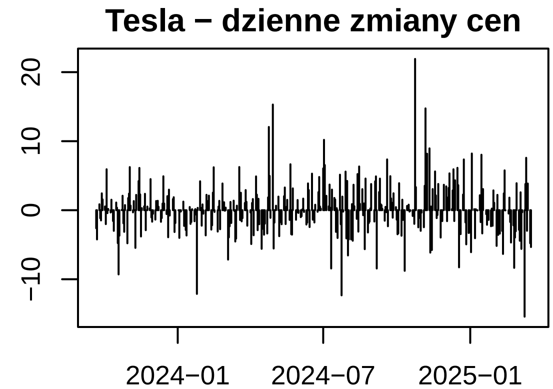
```
         type = "l",
         xlab = "Data",
         ylab = "Cena zamknięcia [USD]",
         main = paste0(name, " - ceny zamknięcia"))
    plot(df$Date,
         df$Return,
         type = "h",
         xlab = "Data",
         ylab = "Stopa zwrotu",
         main = paste0(name, " - dzienne zmiany cen"))
    hist(df$Return,
         breaks = 30,
         col = rgb(0, 0, 1, 0.5),
         border = "black",
         main = paste0(name, " - histogram procentowych\n zmian kursu zamknięcia"),
         ylab = "Gęstość",
         xlab = "Procentowa zmiana (%)", freq = FALSE)
    par(mfrow = c(1, 1))
}
for (name in names(results)) {
    graph_func(results[[name]], name)
}
```

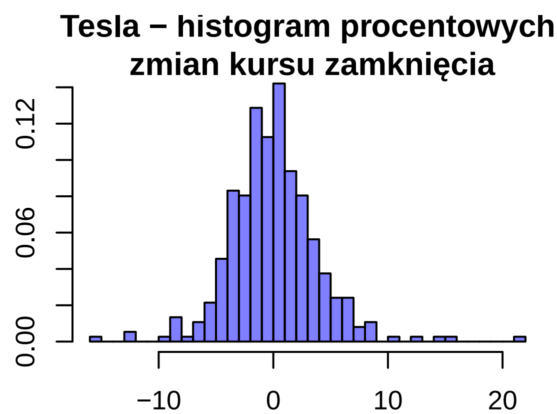








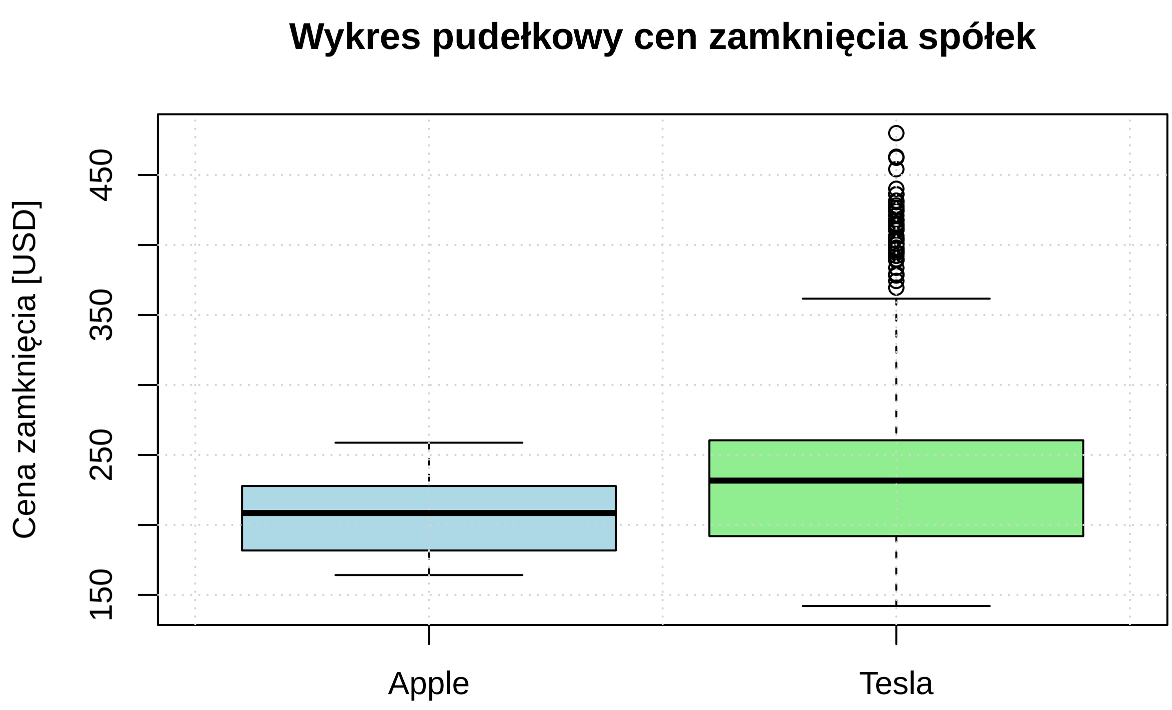




Wykresy zmienności cen dzień do dnia pokazują jak dużą zmiennością charakteryzuje się wartość zamknięcia dla Tesli. Spółka notuje znacznie więcej dużych skoków, co widać po szerokości histogramu procentowych zmian kursu zamknięcia. Dla porównywanej spółki Apple histogram pokazuje sporadyczne epizody zmienności na poziomie powyżej i poniżej 4%. Dla Tesli jest takich przypadków znacznie więcej. Podobnie dla dziennej procentowej zmienności cen. Sama skala i wartości na osi rzędnych świadczą o znacznie większej zmienności (nieprzewidywalności?) ceny akcji Tesli.

1.2.4 Dla danych z ostatnich 18 miesięcy dotyczących wybranych dwóch spółek giełdowych wykonaj jeden wspólny rysunek z wykresami pudełkowymi zmian kursów zamknięcia

```
boxplot(results[[1]]$Close, results[[2]]$Close,
        names = names(results),
        col = c("lightblue", "lightgreen"),
        main = "Wykres pudełkowy cen zamknięcia spółek",
        ylab = "Cena zamknięcia [USD]")
grid()
```



Wykres pudełkowy jeszcze dobitniej pokazuje dużą zmienność cen zamknięcia dla Tesli. Ogromna wręcz ilość wartości odstających (outlierów) w górnej części wykresu pokazuje szybkie dzienne zmiany, typowo w górę, cen akcji. Nie jestem specjalistą giełdowym, ale wydaje mi się że jest to efekt spekulacji i emocjonalnego inwestowania krótkoterminowego.

2 Zadanie 2 (1,5 pkt)

2.1 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:

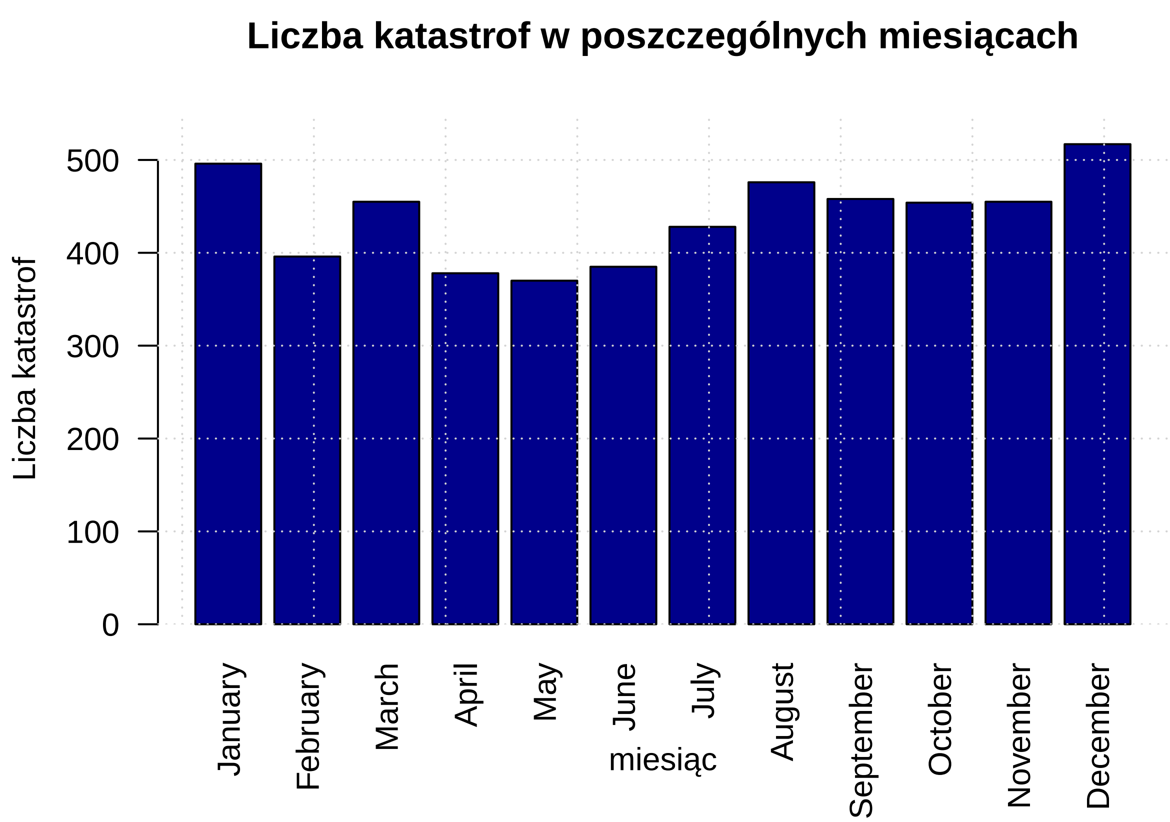
- liczba osób, które przeżyły katastrofy,
- odsetek osób (w procentach), które przeżyły katastrofy.

2.2 Rozwiązanie

(ciąg dalszy na następnej stronie)

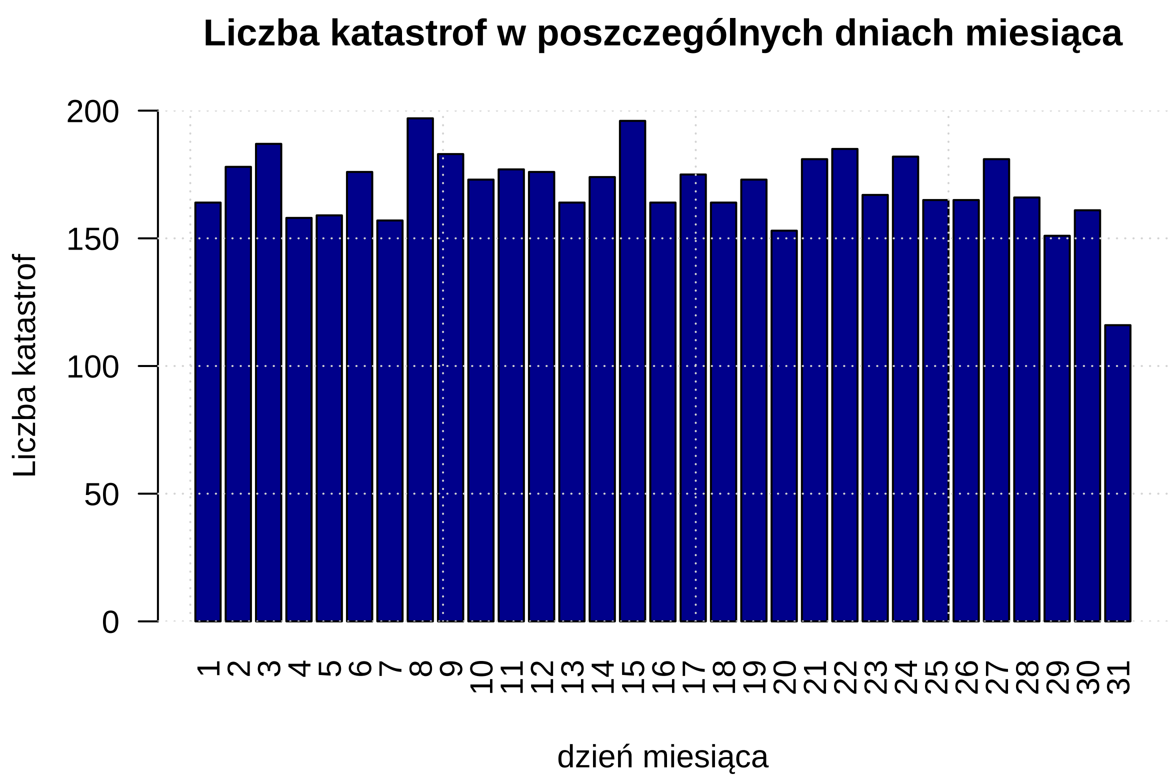
2.2.1 Sporządź wykres liczby katastrof lotniczych w poszczególnych: miesiącach roku (styczeń - grudzień)

```
catast <- read.csv("crashes.csv")
catast$Date <- as.Date(catast$Date, format="%m/%d/%Y")

miesiace <- table(factor(months(catast$Date), levels = month.name))
barplot(miesiace,
        col = "darkblue",
        main = "Liczba katastrof w poszczególnych miesiącach",
        xlab = "miesiąc",
        ylab = "Liczba katastrof",
        ylim = c(0, 550),
        las = 2)
grid()
```



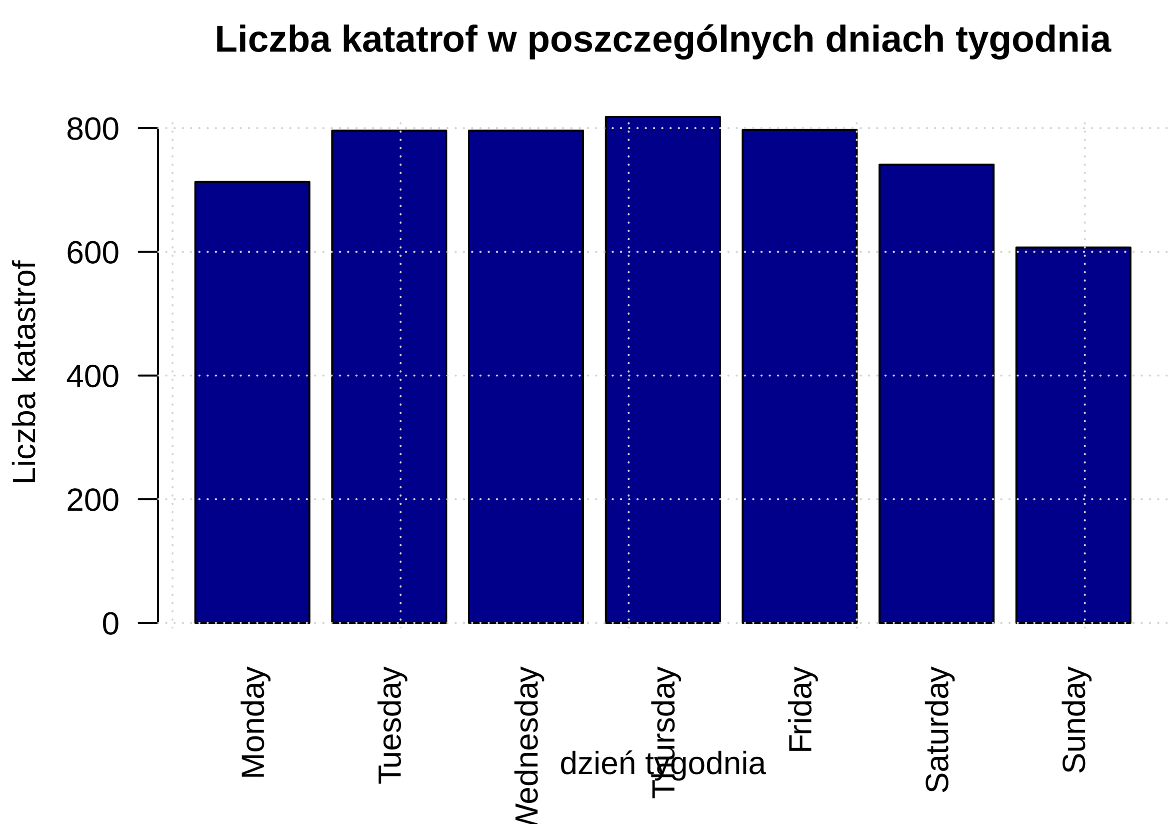
2.2.2 Sporządź wykres liczby katastrof lotniczych w poszczególnych: dniach miesiąca (1-31)

```r
catast <- read.csv("crashes.csv")
catast$Date <- as.Date(catast$Date, format="%m/%d/%Y")

dni <- table(factor(as.numeric(format(catast$Date, "%d"))))
barplot(dni,
        col = "darkblue",
        main = "Liczba katastrof w poszczególnych dniach miesiąca",
        xlab = "dzień miesiąca",
        ylab = "Liczba katastrof",
        ylim = c(0, 200),
        las = 2)
grid()
```



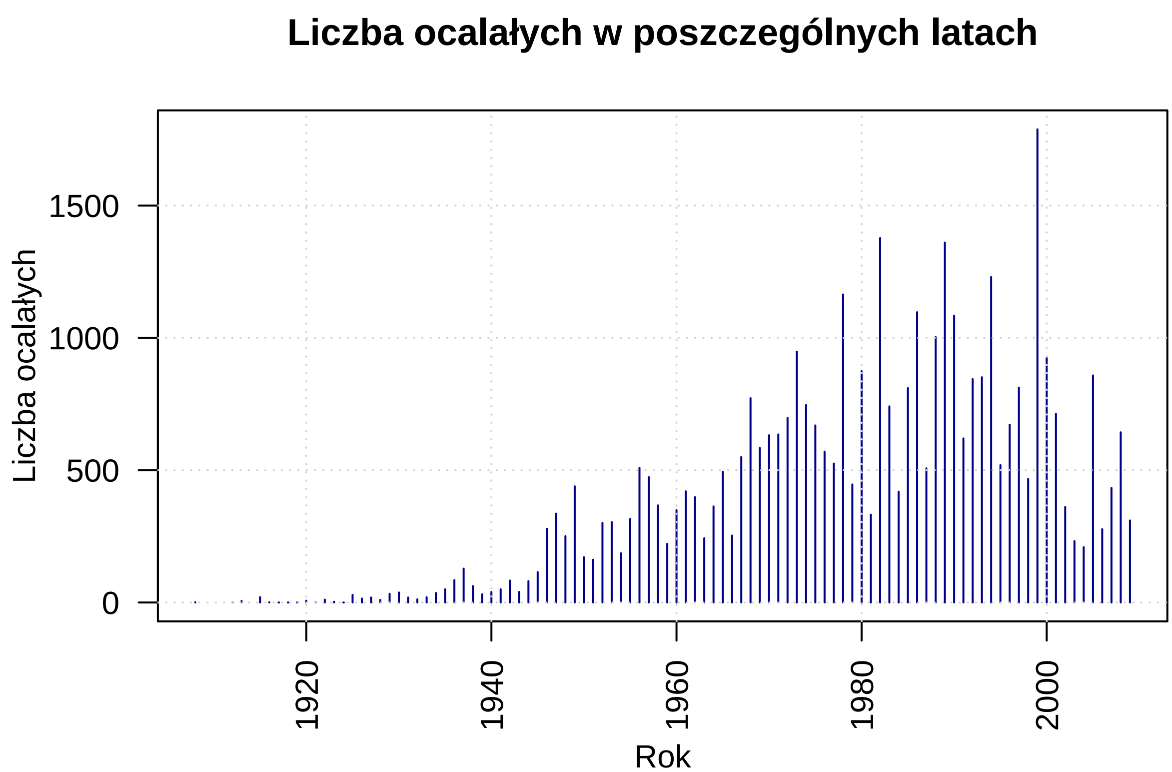
2.2.3 Sporządź wykres liczby katastrof lotniczych w poszczególnych: dniach tygodnia (weekdays())

```
catast <- read.csv("crashes.csv")
catast$Date <- as.Date(catast$Date, format="%m/%d/%Y")

dni_tyg_enum <- c("Monday", "Tuesday", "Wednesday", "Thursday",
                  "Friday", "Saturday", "Sunday")
dni_tyg <- table(factor(weekdays(catast$Date), levels = dni_tyg_enum))
barplot(dni_tyg,
        col = "darkblue",
        main = "Liczba katatrof w poszczególnych dniach tygodnia",
        xlab = "dzień tygodnia",
        ylab = "Liczba katastrof",
        las = 2)
grid()
```



2.2.4 Narysuj jak w kolejnych latach zmieniały się: liczba osób, które przeżyły katastrofy

```
catast <- read.csv("crashes.csv")
catast$Date <- as.Date(catast$Date, format="%m/%d/%Y")
catast$Year <- as.numeric(format(catast$Date, "%Y"))
catast$Survivors <- catast$Aboard - catast$Fatalities

ocalali <- aggregate(Survivors ~ Year, catast, FUN = sum)

plot(ocalali,
     type = 'h',
     col = "darkblue",
     main = "Liczba ocalałych w poszczególnych latach",
     xlab = "Rok",
     ylab = "Liczba ocalałych",
     las = 2)
grid()
```



2.2.5 Narysuj jak w kolejnych latach zmieniały się: odsetek osób (w procentach), które przeżyły katastrofy

```
catast <- read.csv("crashes.csv")
catast$Date <- as.Date(catast$Date, format="%m/%d/%Y")
catast$Year <- as.numeric(format(catast$Date, "%Y"))
catast$Survivors <- catast$Aboard - catast$Fatalities

ocalali <- aggregate(Survivors ~ Year, catast, FUN = sum)
ofiary <- aggregate(Fatalities ~ Year, catast, FUN = sum)
wszyscy <- merge(ocalali, ofiary, by = "Year", all = TRUE)

wszyscy$procent <- 100 * wszyscy$Survivors / (wszyscy$Survivors + wszyscy$Fatalities)

plot(wszyscy$procent ~ wszyscy$Year,
     type = 'h',
     col = "darkblue",
     main = "Odsetek ocalałych w poszczególnych latach",
     xlab = "Rok",
     ylab = "Odsetek ocalałych",
     las = 2)
grid()
```

3 Zadanie 3 (1 pkt)

3.1 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.

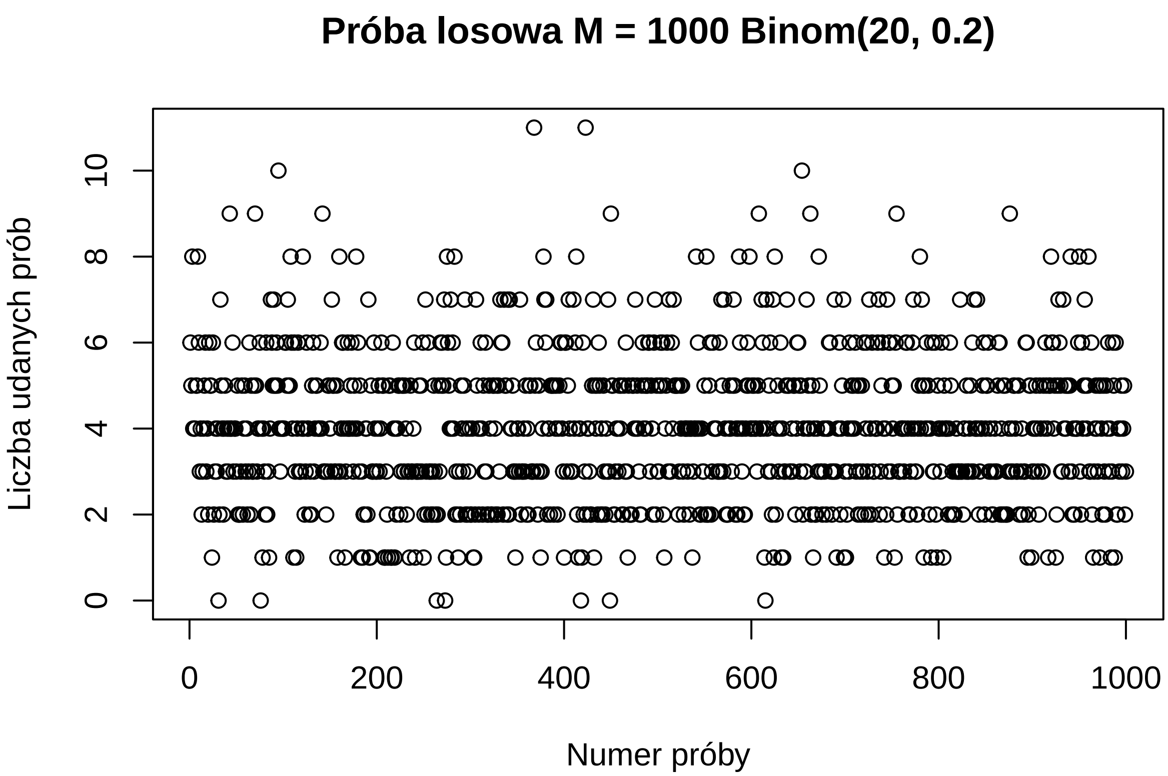
2. Dla każdego z rozkładów narysuj na jednym rysunku empiryczne i teoretyczne (użyj funkcji dbinom) funkcje prawdopodobieństwa, a na drugim rysunku empiryczne i teoretyczne (użyj funkcji pbinom) dystrybuanty. W obu przypadkach wyskaluj oś odciętych od 0 do 20.

3.2 Rozwiązanie

(ciąg dalszy na następnej stronie)

3.2.1 Binom(20,0.2) - wartości wygenerowanych próbek

```
M <- 1000 # Ilość wykonań testu
n <- 20 # Ilość prób w wykonaniu
p <- 0.2 # Prawdopodobieństwo sukcesu dla każdej pojedynczej próby
bi_a <- rbinom(M, n, p)
plot(bi_a,
     type = "p",
     main = "Próba losowa M = 1000 Binom(20, 0.2)",
     xlab = "Numer próby",
     ylab = "Liczba udanych prób")
```



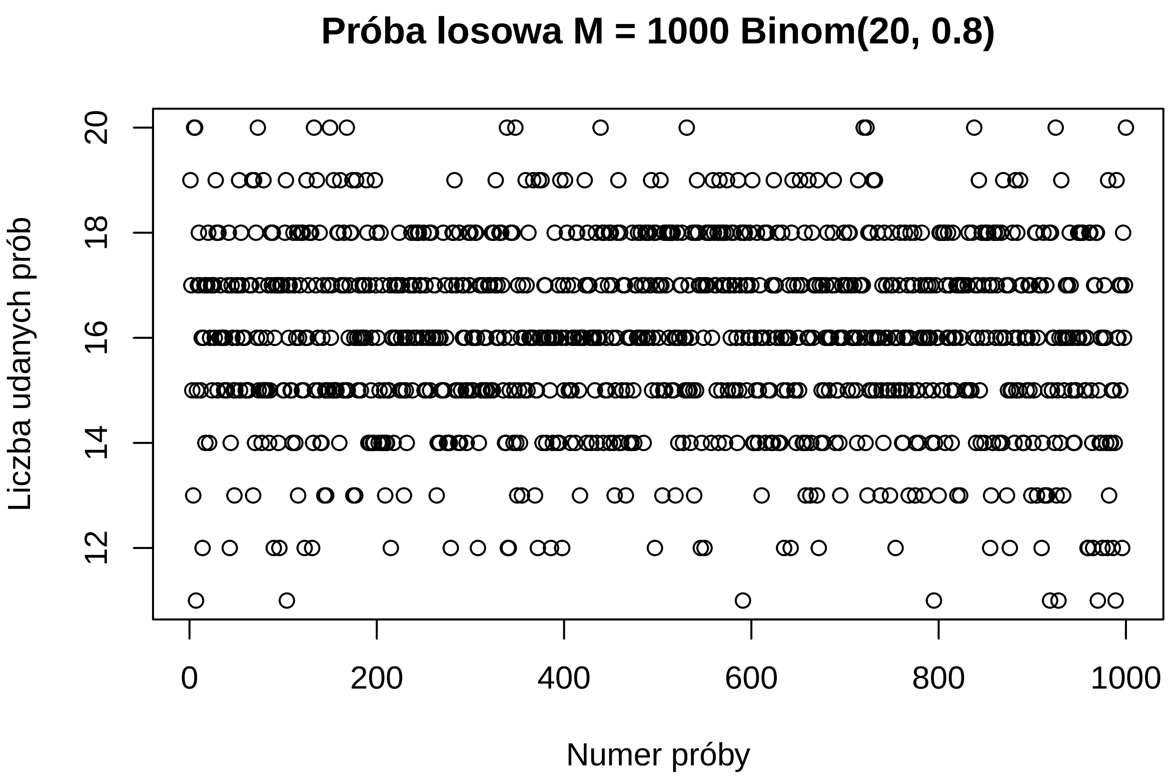
Wykres przedstawia ilość sukcesów w każdym z tysiąca wykonań danego testu. Parametry `M = 1000`, `n = 20` i `p = 0.2` rozkładu dwumianowego pokazują ile z 20 prób zakończyło się 'sukcesem' jeśli prawdopodobieństwo sukcesu w pojedynczej próbie wynosi `p = 0.2`.

Obrazowo można to przedstawić jako wykonanie 1000 razy serii 20 rzutów kostką pięciościenną i odnotowanie po każdej serii ile razy wyrzucono liczbę '1' (lub inną dowolnie wybraną liczbę).

Pomimo, że zakładamy wykonanie 20 rzutów w każdej próbie, przez wzgląd na stosunkowo niskie prawdopodobieństwo sukcesu bardzo rzadko zdarza się wyrzucenie 8, 9 lub więcej 'jedynek'.

3.2.2 Binom(20,0.8) - wartości wygenerowanych próbek

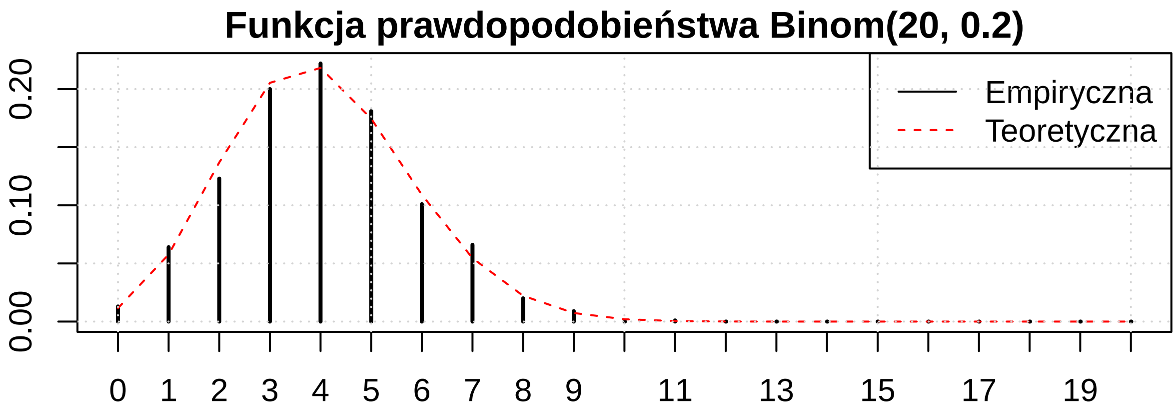
```
M <- 1000 # Ilość wykonań testu
n <- 20 # Ilość prób w wykonaniu
p <- 0.8 # Prawdopodobieństwo sukcesu dla każdej pojedynczej próby
bi_b <- rbinom(M, n, p)
plot(bi_b,
     type = "p",
     main = "Próba losowa M = 1000 Binom(20, 0.8)",
     xlab = "Numer próby",
     ylab = "Liczba udanych prób")
```

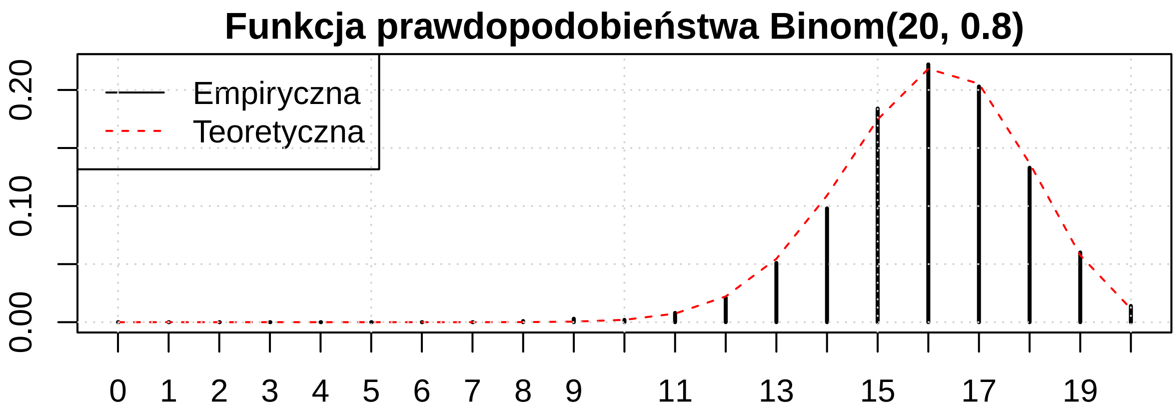


Drugi przykład, gdzie prawdopodobieństwo sukcesu w pojedynczej próbie wynosi `p = 0.8` pokazuje, że liczba udanych prób nieczęsto jest niższa od 10. Zdarzają się przypadki, gdzie każda z 20 prób dawała pozytywny wynik. Dla kontrastu (z wykresem gdzie `p = 0.2`) zakres wartości jest 'przesunięty' w górę.

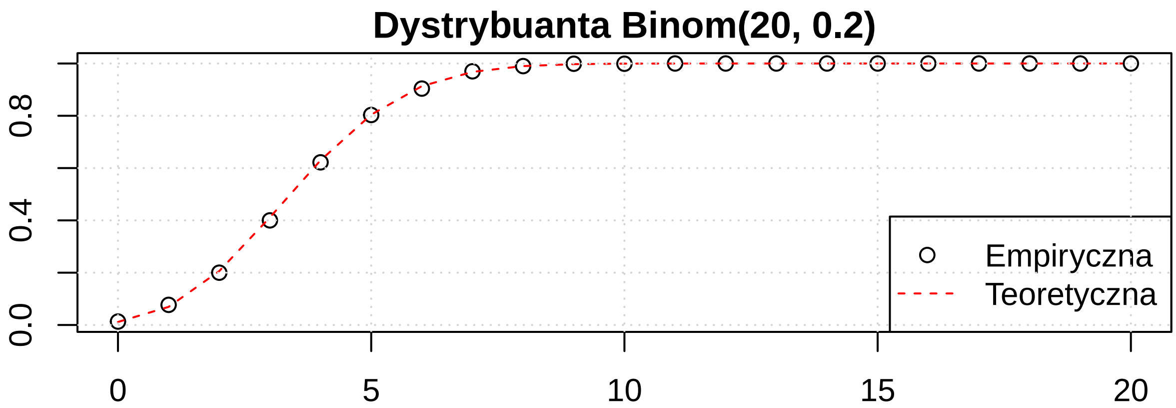
3.2.3 Dla każdego z rozkładów narysuj na jednym rysunku empiryczne i teoretyczne (użyj funkcji dbinom) funkcje prawdopodobieństwa, a na drugim rysunku empiryczne i teoretyczne (użyj funkcji pbinom) dystrybuanty. W obu przypadkach wyskaluj oś odciętych od 0 do 20.

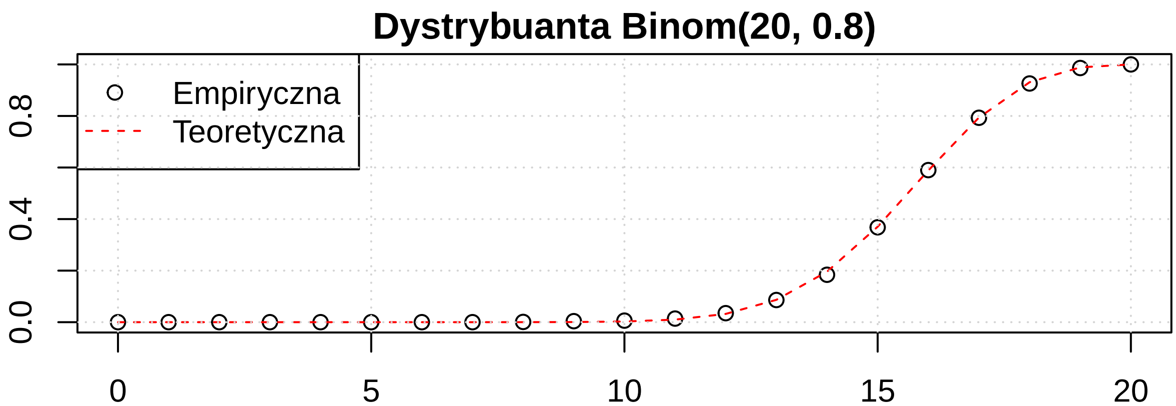
```r
# Funkcje prawdopodobieństwa

M <- 1000 # Ilość wykonań testu
n <- 20 # Ilość prób w wykonaniu
bi_a <- rbinom(M, n, 0.2)
bi_b <- rbinom(M, n, 0.8)
odciete <- seq(0, n)
# Empiryczna funkcja prawdopodobieństwa
empiryczna_func_a <- table(factor(bi_a, levels = odciete)) / M
empiryczna_func_b <- table(factor(bi_b, levels = odciete)) / M
# Teoretyczna funkcja prawdopodobieństwa
teoretyczna_func_a <- dbinom(odciete, n, 0.2)
teoretyczna_func_b <- dbinom(odciete, n, 0.8)

par(mfrow = c(2, 1), mai = c(0.5, 0.5, 0.3, 0.3))
# Empiryczna funkcja prawdopodobieństwa
plot(empiryczna_func_a,
     main = "Funkcja prawdopodobieństwa Binom(20, 0.2)",
     xlab = "Liczba udanych prób",
     ylab = "Prawdopodobieństwo")
# Teoretyczna funkcja prawdopodobieństwa
lines(teoretyczna_func_a ~ odciete,
      col = "red",
      lty = 2)
legend("topright",
       legend = c("Empiryczna", "Teoretyczna"),
       col = c("black", "red"),
       lty = c(1, 2)) # ciągła i przerywana
grid()

# Empiryczna funkcja prawdopodobieństwa
plot(empiryczna_func_b,
     main = "Funkcja prawdopodobieństwa Binom(20, 0.8)",
     xlab = "Liczba udanych prób",
     ylab = "Prawdopodobieństwo")
# Teoretyczna funkcja prawdopodobieństwa
lines(teoretyczna_func_b ~ odciete,
      col = "red",
      lty = 2)
legend("topleft",
       legend = c("Empiryczna", "Teoretyczna"),
       col = c("black", "red"),
       lty = c(1, 2)) # ciągła i przerywana
grid()
```


Funkcja prawdopodobieństwa Binom(20, 0.2)
Empiryczna
Teoretyczna
0.00
0.10
0.20
0 1 2 3 4 5 6 7 8 9 11 13 15 17 19


Funkcja prawdopodobieństwa Binom(20, 0.8)
Empiryczna
Teoretyczna
0.00
0.10
0.20
0 1 2 3 4 5 6 7 8 9 11 13 15 17 19

```r
# Dystrybuanty

M <- 1000 # Ilość wykonań testu
n <- 20 # Ilość prób w wykonaniu
odciete <- seq(0, n)
empiryczna_dyst_a <- ecdf(bi_a) # ecdf() zwraca funkcję!
empiryczna_dyst_b <- ecdf(bi_b)
teoretyczna_dyst_a <- pbinom(odciete, n, 0.2)
teoretyczna_dyst_b <- pbinom(odciete, n, 0.8)

par(mfrow = c(2, 1), mai = c(0.5, 0.5, 0.3, 0.3))
# Empiryczna dystrybuanta
plot(empiryczna_dyst_a(odciete) ~ odciete,
     main = "Dystrybuanta Binom(20, 0.2)",
     xlab = "Liczba udanych prób",
     ylab = "Prawdopodobieństwo")
# Teoretyczna dystrybuanta
lines(teoretyczna_dyst_a ~ odciete,
      col = "red",
      lty = 2)
legend("bottomright",
       legend = c("Empiryczna", "Teoretyczna"),
       col = c("black", "red"),
       pch = c(1, NA), # małe kółko w legendzie i brak symbolu
       lty = c(NA, 2)) # brak linii i linia przerywana
grid()

# Empiryczna dystrybuanta
plot(empiryczna_dyst_b(odciete) ~ odciete,
     main = "Dystrybuanta Binom(20, 0.8)",
     xlab = "Liczba udanych prób",
     ylab = "Prawdopodobieństwo")
# Teoretyczna dystrybuanta
lines(teoretyczna_dyst_b ~ odciete,
      col = "red",
      lty = 2)
legend("topleft",
       legend = c("Empiryczna", "Teoretyczna"),
       col = c("black", "red"),
       pch = c(1, NA), # małe kółko w legendzie i brak symbolu
       lty = c(NA, 2)) # brak linii i linia przerywana
grid()
```


Dystrybuanta Binom(20, 0.2)
Empiryczna
Teoretyczna
0.0
0.4
0.8
0
5
10
15
20


Dystrybuanta Binom(20, 0.8)
Empiryczna
Teoretyczna
0.0
0.4
0.8
0
5
10
15
20

4 Zadanie 4 (1,5 pkt)

4.1 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).

4.2 Rozwiązanie

(ciąg dalszy na następnej stronie)

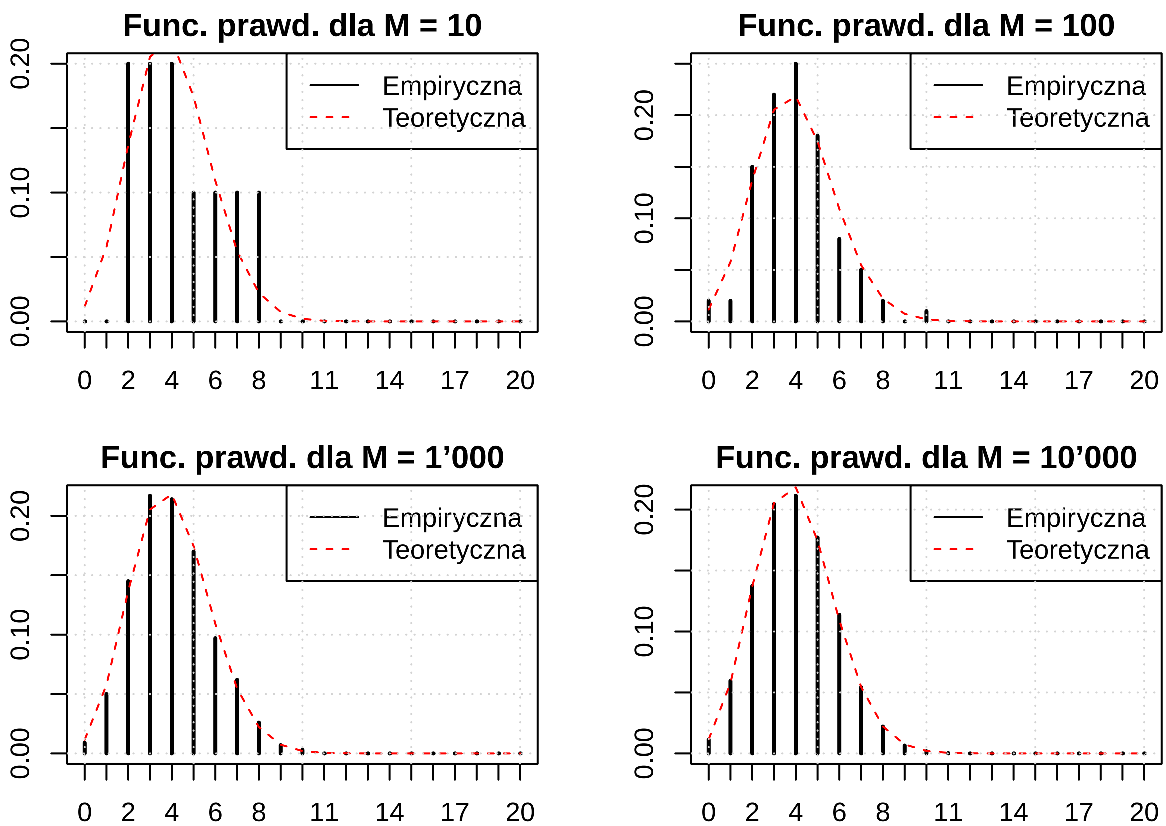
4.2.1 Dla rozkładu dwumianowego Binom(20, 0.2) wygeneruj trzy próby losowe składające się z M = 100, 1000 i 10000 próbek

Dodałem wartość M=10 próbek aby dobitniej pokazać jak mało reprezentatywny jest model zbudowany na małej próbie. Również dlatego, żeby mieć 4 wykresy zamiast trzech.

```
M_vec <- c(10, 100, 1000, 10000)
n <- 20
p <- 0.2
# Ustawienie seed w thm przypadku ma sens, ponieważ w ostatnim
# punkcie tego zadania odnoszę się w opisach do konkretnych wartości
# średnich i wariancji.
set.seed(1205012)
results <- lapply(M_vec, function(M) rbinom(M, n, p))
names(results) <- as.character(M_vec)
```

4.2.2 Dla poszczególnych prób wykreśl empiryczne i teoretyczne funkcje prawdopodobieństwa, ...

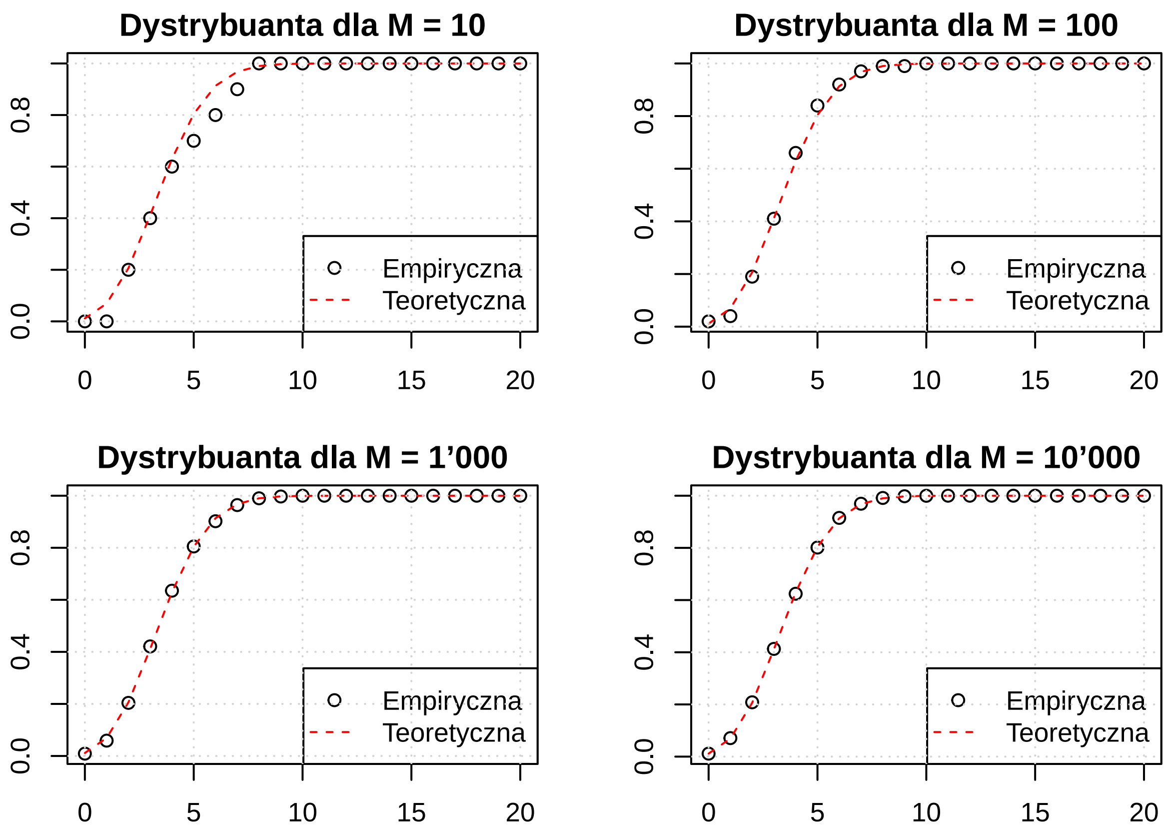
```
odciete <- seq(0, n)

empiryczne_func <- lapply(M_vec, function(M)
    table(factor(results[[as.character(M)]], levels = odciete)) / M)
names(empiryczne_func) <- as.character(M_vec)
# Teoretyczna funkcja prawdopodobieństwa jest jedna
teoretyczna_func <- dbinom(odciete, n, p)

par(mfrow = c(2, 2), mai = c(0.5, 0.5, 0.3, 0.3))
for (M_val in M_vec) {
    df <- empiryczne_func[[as.character(M_val)]]
    plot(df,
         main = paste0("Func. prawd. dla M = ",
                       format(M_val,
                       big.mark = "'",
                       scientific = FALSE)),
         xlab = "Liczba udanych prób",
         ylab = "Prawdopodobieństwo")
    lines(teoretyczna_func ~ odciete,
          col = "red",
          lty = 2)
    legend("topright",
           legend = c("Empiryczna", "Teoretyczna"),
           col = c("black", "red"),
           lty = c(1, 2)) # ciągła i przerywana
    grid()
}
```



Doskonale widać jak wraz ze wzrostem ilości próbek otrzymane rezultaty coraz lepiej dają się zamodelować teoretyczną funkcją prawdopodobieństwa.

4.2.3 ... a także empiryczne i teoretyczne dystrybuanty

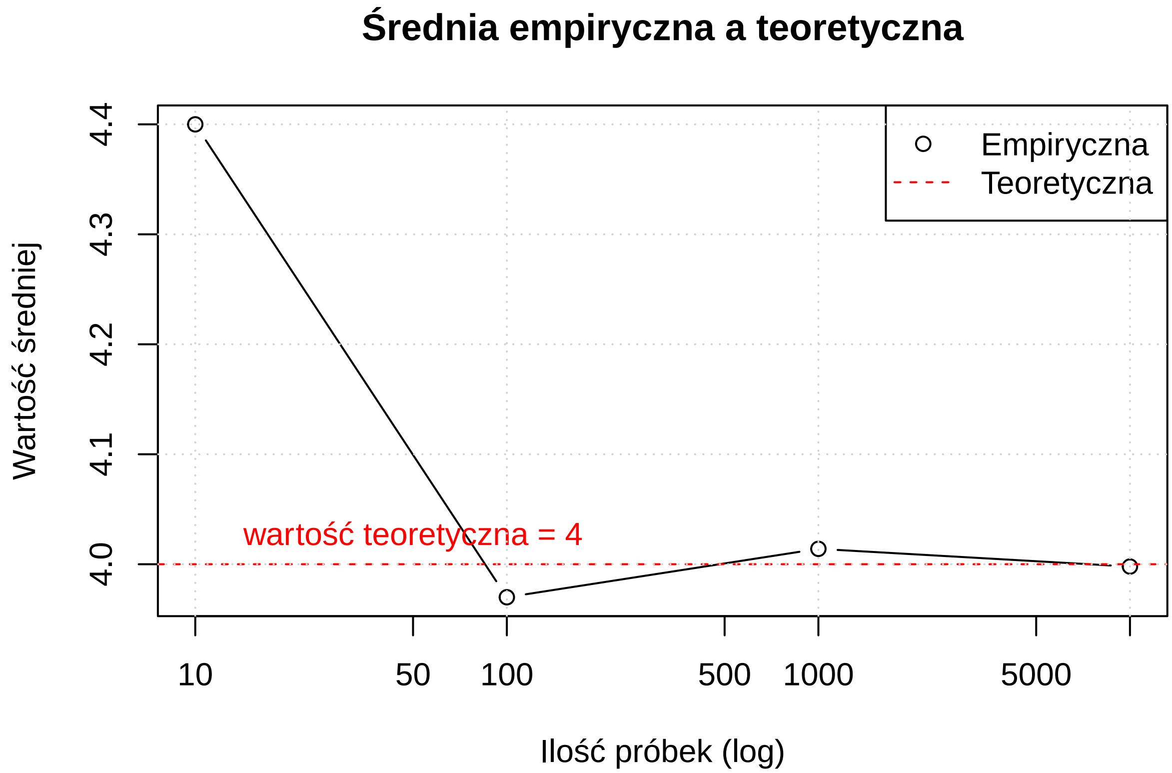
```r
# ecdf() zwraca funkcję!
empiryczne_dyst <- lapply(M_vec, function(M) ecdf(results[[as.character(M)]]))
names(empiryczne_dyst) <- as.character(M_vec)
teoretyczna_dyst <- pbinom(odciete, n, p)

par(mfrow = c(2, 2), mai = c(0.5, 0.5, 0.3, 0.3))

for (M_val in M_vec) {
    df <- empiryczne_dyst[[as.character(M_val)]]
    plot(df(odciete) ~ odciete,
         main = paste0("Dystrybuanta dla M = ",
                       format(M_val,
                       big.mark = "'",
                       scientific = FALSE)),
         xlab = "Liczba udanych prób",
         ylab = "Prawdopodobieństwo")
    # Teoretyczna dystrybuanta
    lines(teoretyczna_dyst ~ odciete,
          col = "red",
          lty = 2)
    legend("bottomright",
           legend = c("Empiryczna", "Teoretyczna"),
           col = c("black", "red"),
           pch = c(1, NA), # małe kółko w legendzie i brak symbolu
           lty = c(NA, 2)) # brak linii i linia przerywana
    grid()
}
```



Podobnie jak w przypadku funkcji prawdopodobieństwa tak i tutaj wraz ze wzrostem ilości prób wartości otrzumane empirycznie coraz bardziej przypominają model teoretyczny.

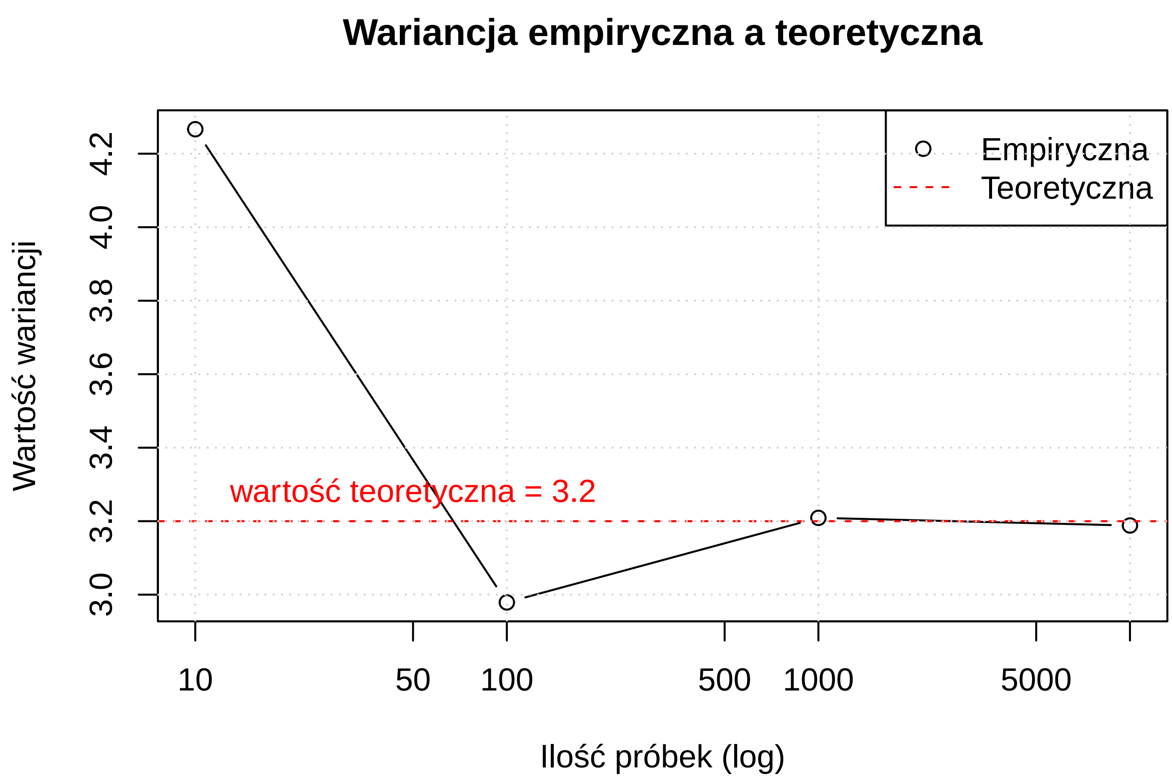
4.2.4 We wszystkich przypadkach oblicz empiryczne wartości średnie i wariancje. Porównaj je ze sobą oraz z wartościami teoretycznymi dla rozkładu Binom(20, 0.2)

```
# teoretyczna średnia wartość próby to prawdopodobieństwo * ilość prób
# przy 20 rzutach kostką pięciościenną teoretyczna wartość średniej
# wynosi 20 * 0.2 = 4
teoretyczna_srednia <- n * p
teoretyczna_wariancja <- n * p * (1 - p)

empiryczne_srednie <- lapply(M_vec, function(M) mean(results[[as.character(M)]]))
empiryczne_wariancje <- lapply(M_vec, function(M) var(results[[as.character(M)]]))

par(mfrow = c(1, 1))
plot(M_vec,
     empiryczne_srednie,
     main = "Średnia empiryczna a teoretyczna",
     type = 'b',
     log = 'x',
     xlab = "Ilość próbek (log)",
     ylab = "Wartość średniej")
abline(h = teoretyczna_srednia,
       col = 'red',
       lty = 2,)
legend("topright",
       legend = c("Empiryczna", "Teoretyczna"),
       col = c("black", "red"),
       pch = c(1, NA),
       lty = c(NA, 2))
text(x = 50,
     y = teoretyczna_srednia,
     labels = paste("wartość teoretyczna =", round(teoretyczna_srednia, 2)),
     col = "red",
     pos = 3)
grid()
```



Co wcześniej obserwowaliśmy dla funkcji prawdopodobieństwa i dystrybuanty tutaj możemy zaobserwować na podstawie średniej wartości próby. Dla jasno zdefiniowanej wartości prawdopodobieństwa sukcesu `p = 0.2` statystycznie powinniśmy widzieć 4 sukcesy na 20 testów. Jeśli wykonamy 10 lub 100 prób rozrzut wyników jest niereprezentatywny i średnia znacznie odbiega od wartości teoretycznej. Przy zwiększaniu ilości prób wartość coraz bardziej zbliża się do wartości teoretycznej.

```r
plot(M_vec,
     empiryczne_wariancje,
     main = "Wariancja empiryczna a teoretyczna",
     type = 'b',
     log = 'x',
     xlab = "Ilość próbek (log)",
     ylab = "Wartość wariancji")
abline(h = teoretyczna_wariancja,
       col = 'red',
       lty = 2,)
legend("topright",
       legend = c("Empiryczna", "Teoretyczna"),
       col = c("black", "red"),
       pch = c(1, NA),
       lty = c(NA, 2))
text(x = 50,
     y = teoretyczna_wariancja,
     labels = paste("wartość teoretyczna =", round(teoretyczna_wariancja, 2)),
     col = "red",
     pos = 3)
grid()
```



Analogicznie do średniej wartości z próby wariancja również zbliża się do wartości teoretycznej przy zwiększaniu ilości prób.
