Przemysław GRZEGORZEWSKI
Marek GĄGOLEWSKI
Konstancja BOBECKA-WESOŁOWSKA

Wnioskowanie statystyczne

z wykorzystaniem środowiska R

Warszawa 2014

Opracowanie w systemie LaTeX
Marek Gągolewski

Wydawca: Biuro ds. Projektu „Program Rozwojowy Politechniki Warszawskiej"

ISBN 978-83-937260-1-1

Spis treści

Przedmowa

Niniejszy skrypt jest owocem naszych wieloletnich doświadczeń w nauczaniu statystyki matematycznej na Wydziale Matematyki i Nauk Informacyjnych Politechniki Warszawskiej. Nie należy go jednak uważać za kolejny podręcznik statystyki, których niemało, by wspomnieć [7, 9–11, 16]. Brak w nim typowego dla takiego podręcznika zestawu definicji wprowadzanych pojęć i analizy ich własności, nie podano też twierdzeń ani ich dowodów. Skrypt ten przypomina bardziej poradnik dedykowany Czytelnikowi, któremu nieobce są już podstawy statystyki matematycznej, ale który również chciałby zastosować poznane metody w praktyce. Tak więc niniejszą pozycję należy postrzegać wyłącznie jako uzupełnienie teorii wykładanej podczas tradycyjnego kursu wnioskowania statystycznego.

Każdy, kto zetknął się z wnioskowaniem statystycznym w praktyce, wie, jak pomocnym narzędziem wspomagającym owe wnioskowanie jest komputer, pozwalający przetwarzać duże zbiory danych, wyręczający nas w żmudnych rachunkach i ułatwiający wykonanie rozmaitych przydatnych wizualizacji. To właśnie wspomagana komputerowo analiza danych i statystyka odgrywają istotną rolę w szeroko rozumianej matematyce przemysłowej, obejmującej nie tylko tematykę techniczną, ale i zagadnienia z obszaru biologii, medycyny, nauk społecznych, inżynierii finansowej itd.

Aby móc w pełni skorzystać z dobrodziejstw, jakie oferuje komputer w zakresie wsparcia wnioskowania statystycznego i analizy danych, konieczne jest odpowiednie oprogramowanie. Praktycy korzystają z różnych pakietów statystycznych, począwszy od specjalistycznych programów, ukierunkowanych na wąskie zastosowania, a skończywszy na pakietach oferujących bogate zestawy różnorodnych narzędzi i procedur, które mają potencjalnie zaspokoić potrzeby szerokiego grona analityków.

W ostatnich latach coraz bardziej popularnym narzędziem staje się środowisko R. Można wskazać wiele źródeł sukcesu tego oprogramowania. Po pierwsze, R jest dostępny na wszystkich platformach i systemach operacyjnych. Po wtóre, jest to środowisko otwarte, które dostarczając gotowe funkcje pozwala jednocześnie użytkownikowi tworzyć własne procedury. Dzięki temu różne repozytoria (np. CRAN) są nieustannie wzbogacane o kolejne biblioteki funkcji pisane przez użytkowników R. Nie bez znaczenia jest również fakt, że jest to pakiet bezpłatny i wolnodostępny. Dzięki tym walorom R jest już od lat wykorzystywany także w procesie dydaktycznym realizowanym na Wydziale MiNI. Stąd też wziął się pomysł, aby praktyczne ćwiczenia, którym poświęcony jest niniejszy skrypt, były realizowane właśnie z wykorzystaniem programu R.

Od strony merytorycznej nasz skrypt obejmuje wstęp do programowania w języku R, statystykę opisową, estymację punktową i przedziałową, weryfikację hipotez (z uwzględnieniem testów parametrycznych i nieparametrycznych), badanie związku między cechami (w tym m.in. analizę regresji i analizę wariancji) oraz pewne zagadnienia z zakresu rachunku prawdopodobieństwa i symulacji komputerowych. Każdy z rozdziałów

ma identyczną strukturę: na początku podajemy zwięzłe informacje dotyczące funkcji pakietu R związanych z rozważaną tematyką, po czym zamieszczamy zestaw przykładów zawierających w pełni rozwiązane i omówione zadania. Dodatkowo, w każdym rozdziale publikujemy zbiór zadań pozostawionych do rozwiązania Czytelnikom (do części z nich dołączamy również wskazówki i odpowiedzi).

Mamy nadzieję, że Czytelnik, który prześledzi ze zrozumieniem przykłady zawarte z niniejszym skrypcie, nie tylko poszerzy swą wiedzę i umiejętności w zakresie wnioskowania statystycznego, ale i polubi samo środowisko R. Tych zaś, którzy będą chcieli dowiedzieć się jeszcze więcej na temat programu R odsyłamy do bogatej literatury, m.in. [1–6, 8, 12–14].

Przemysław Grzegorzewski
Marek Gągolewski
Konstancja Bobecka

Warszawa, wrzesień 2014 r.

Programowanie w języku R 1

1.1. Wprowadzenie

1.1.1. Pierwsze kroki z R

Środowisko R[1] jest otwartym, zaawansowanym i coraz bardziej zyskującym uznanie oprogramowaniem służącym m.in. do obliczeń statystycznych i tworzenia wykresów. Jego trzon stanowi wygodny, interpretowany język programowania, który jest podobny składnią do C/C++.

Program ten działa na różnych systemach operacyjnych: Windows, Linux i OS X. Jego instalacja jest łatwa i nie powinna przysporzyć kłopotów nawet niezaawansowanym użytkownikom. Choć nie jest to niezbędne do korzystania z programu R, zachęcamy do zainstalowania dodatkowo bardzo wygodnego edytora RStudio[2], znakomicie ułatwiającego codzienną pracę w R.

Do generowania przykładów w niniejszej książce używamy R w wersji 3.1.1. Jednakże większość zamieszczonych przykładów powinna działać także we wcześniejszych i późniejszych wersjach programu, gdyż w przedstawianych rozwiązaniach nie korzystamy ze złożonych konstrukcji językowych ani z niestandardowych pakietów.

Osobom szczególnie zainteresowanym językiem R polecamy prace [1, 3, 6].

Po uruchomieniu RStudio, konsola R wita użytkownika stosownym komunikatem i drukuje tzw. *znak zachęty*, który standardowo ma następującą postać:

```
>
```

Ten znak wskazuje gotowość do przyjmowania poleceń od użytkownika.

Aby pomóc Czytelnikowi „nawiązać kontakt" z R potraktujmy go najpierw jako kalkulator. Zacznijmy zatem od wydania prostej komendy, której celem będzie obliczenie wyrażenia „$2 + 2$". Wpisujemy owe działanie po znaku zachęty, a następnie zatwierdzamy komendę klawiszem ⟨ENTER⟩. Poniżej zostanie natychmiast wyświetlony wynik działania:

```
> 2+2
[1] 4
```

Niektórzy Czytelnicy mogą pamiętać, że np. w języku C wymaga się, aby każde polecenie kończyć znakiem średnika. Jak widzimy, tutaj nie jest to konieczne. Jednakże tego rodzaju separatora możemy użyć do wprowadzenia więcej niż jednego wyrażenia w jednym wierszu, np.

[1] http://www.r-project.org/
[2] http://www.rstudio.com/ide

```
> 2*3.5; 5+4*2
[1] 7
[1] 13
```

Informacja

W R część całkowitą liczby oddzielamy od części ułamkowej za pomocą kropki.

Ciekawą własnością programu RStudio jest to, że nie trzeba wpisywać całego polecenia w jednym wierszu. Jeżeli wprowadzimy niedokończone wyrażenie, konsola R poprosi o jego uzupełnienie.

```
> 5 / 2 *
+ 8
[1] 20
```

Zwróćmy uwagę, że w tym przypadku następuje zmiana postaci znaku zachęty.

Informacja

Historia ostatnio wywoływanych poleceń dostępna jest za pośrednictwem klawiszy ⟨↑⟩ oraz ⟨↓⟩. Możemy również, podobnie jak w zwykłym edytorze tekstu, dowolnie przesuwać kursor, np. aby zmienić fragment wprowadzanej komendy. Spróbujmy użyć do tego klawiszy ⟨←⟩, ⟨→⟩, ⟨HOME⟩, ⟨END⟩.

Dobrym nawykiem jest komentowanie wpisywanego kodu. Służy do tego symbol „kratki" (#) — wszystkie następujące po nim znaki aż do końca wiersza będą ignorowane przez interpreter poleceń, np.

```
> 2/4  # dzielenie
[1] 0.5
```

Informacja

W RStudio wygodnie jest używać wbudowanego edytora plików (menu *File* → *New File* → R *Script*). Można w nim robić np. notatki lub pisać złożone funkcje. Dowolne fragmenty kodu wysyłamy z tego edytora do konsoli za pomocą kombinacji klawiszy ⟨CTRL+ENTER⟩.

Warto również pamiętać o częstym zapisywaniu tworzonego pliku (*File* → *Save*).

1.1.2. Wektory atomowe i czynniki

1.1.2.1. Wektory liczbowe, tworzenie wektorów, podstawowe operacje

Podstawowym typem danych, z którym będziemy mieli do czynienia w pracy w środowisku R są tzw. *wektory atomowe*.

Zwróćmy uwagę na to, że wywołane w poprzednim punkcie polecenie:

```
> 2+2
[1] 4
```

jest tak naprawdę działaniem na wektorach liczbowych o długości 1, dającym w wyniku wektor o takiej samej liczbie elementów.

Do stworzenia wektora o dowolnej długości używamy funkcji `c()`.

```
> c(4, 6, 5, 3)
[1] 4 6 5 3
```

> **Informacja**
>
> Należy pamiętać o tym, że R rozróżnia wielkość liter. To zaś znaczy, że powyższe wywołanie pisane wielką literą `C()` najprawdopodobniej zakończy się zgłoszeniem błędu.

Skoro już „pojedyncza liczba" jest wektorem, to dlaczego by nie spróbować połączenia kilku dłuższych wektorów w jeden? Oto przykład:

```
> c(1, 2, c(3,4,5), c(6,7), 8)
[1] 1 2 3 4 5 6 7 8
```

Większość operacji arytmetycznych na wektorach jest wykonywana *element po elemencie* (ang. *element-wise*), tzn. dla ciągów $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ oraz $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ wynikiem działania $\mathbf{a} \circledcirc \mathbf{b}$ jest wektor $(a_1 \circledcirc b_1, \ldots, a_n \circledcirc b_n)$. Przykładowo:

```
> c(1, 2, 3)*c(0.5, 0.5, 0.5)
[1] 0.5 1.0 1.5
```

Jeżeli dane operandy są różnej długości (dla ustalenia uwagi niech $n_1 < n_2$), następuje ich uzgodnienie za pomocą tzw. *reguły zawijania* (ang. *recycling rule*). Wektor krótszy $(a_1, a_2, \ldots, a_{n_1})$ jest powielany tyle razy, ile potrzeba, aby dopasował się do dłuższego, według schematu $(a_1, a_2, \ldots, a_{n_1}, a_1, a_2, \ldots)$. Jeżeli rozmiary wektorów nie są zgodne, tzn. $n_2 \mod n_1 \neq 0$, wyświetlane jest ostrzeżenie. W wyniku działania otrzymujemy ciąg o długości n_2. Dla przykładu, następujące wyrażenie jest równoważne działaniu zilustrowanemu powyżej.

```
> c(1, 2, 3)*0.5 # to samo, co wyżej
[1] 0.5 1.0 1.5
```

A oto inne przykłady:

```
> c(1, 2, 3, 4)+c(1, 0.5)
[1] 2.0 2.5 4.0 4.5
> 2^c(0, 1, 2, 3, 4) # potęgowanie
[1]  1  2  4  8 16
```

W tabeli 1.1 przestawiono dostępne operatory arytmetyczne.

Tab. 1.1. Operatory arytmetyczne

Operacja	Znaczenie
`-x`	zmiana znaku
`x + y`	dodawanie
`x - y`	odejmowanie
`x * y`	mnożenie
`x / y`	dzielenie
`x ^ y`	potęgowanie
`x %/% y`	dzielenie całkowite
`x %% y`	reszta z dzielenia

Pewne operacje mogą generować wartości specjalne: nieskończoność (stała `Inf`) i nie-liczbę (stała `NaN`, ang. *not a number*). Na przykład:

```
> 1.0 / 0.0
[1] Inf
> 0.0 / 0.0
[1] NaN
> -1/Inf
[1] 0
> Inf-Inf
[1] NaN
> NaN + 100
[1] NaN
> -Inf + 100000000000
[1] -Inf
```

R ma również wbudowanych wiele funkcji matematycznych. Ich wartości są obliczane oddzielnie dla poszczególnych elementów wektora, tzn.

$$f((a_1, a_2, \ldots, a_n)) = (f(a_1), f(a_2), \ldots, f(a_n)).$$

Przykładowo:

```
> pi # to jest wbudowana stała
[1] 3.141593
> sin(c(0.0, pi*0.5, pi, pi*1.5, pi*20))
```

```
[1]  0.000000e+00  1.000000e+00  1.224647e-16 -1.000000e+00
[5] -2.449294e-15
> round(sin(c(0.0, pi*0.5, pi, pi*1.5, pi*20)), 3) # zaokrąglijmy wynik
[1]  0  1  0 -1  0
```

Informacja

`sin(pi)` to wynik zapisany w tzw. *notacji naukowej*. Np. liczba `3.2e-2` to nic innego niż $3{,}2 \times 10^{-2}$.

```
> 3.2e-2
[1] 0.032
```

Zatem: 10^{-16} to bardzo mała liczba (rezultat błędów arytmetyki zmiennopozycyjnej komputera), czyli prawie 0.

W tabeli 1.2 zamieszczamy wykaz najważniejszych funkcji matematycznych.

Tab. 1.2. Funkcje matematyczne

Funkcja	Znaczenie
`abs(x)`	wartość bezwzględna
`sqrt(x)`	pierwiastek kwadratowy
`cos(x)`	cosinus
`sin(x)`	sinus
`tan(x)`	tangens
`acos(x)`	arcus cosinus
`asin(x)`	arcus sinus
`atan(x)`	arcus tanges
`exp(x)`	e^x
`log(x, base=exp(1))`	logarytm o podstawie `base`
`log10(x)`	logarytm o podstawie 10
`log2(x)`	logarytm o podstawie 2
`round(x, digits=0)`	zaokrąglanie do `digits` cyfr po kropce dziesiętnej
`floor(x)`	największa liczba całkowita nie większa niż `x`
`ceiling(x)`	najmniejsza liczba całkowita nie mniejsza niż `x`

Informacja

W zapisie `log(x, base=exp(1))`, parametr `base` ma określoną *wartość domyślną*. Jeżeli pominiemy jawne nadanie wartości temu argumentowi, przy wywołaniu funkcji przypisana mu będzie liczba e.

Informacja

R posiada bardzo rozwinięty, wygodny i dobrze zorganizowany system pomocy. Aby uzyskać więcej informacji na temat jakiejś funkcji, np. `sin()`, piszemy

```
> ?sin
```

bądź równoważnie

```
> help("sin")
```

W przypadku, gdy nie znamy dokładnej nazwy pożądanej funkcji, możemy też skorzystać z prostej wyszukiwarki:

```
> help.search("standard deviation")
```

Ponadto gdy piszemy polecenia w konsoli, możemy skorzystać z podpowiedzi bądź tzw. autouzupełnienia, dostępnego po wciśnięciu klawisza ⟨TAB⟩:

```
> ce      # tutaj wciskamy klawisz [TAB]...
> ceiling # ...i R "dopowiedział" nazwę funkcji
```

Jako ciekawostkę wpiszmy

```
> cos    # [TAB]
  => cos   cosh
```

Wynika z tego, że R „zna" dwie funkcje o nazwach zaczynających się od `cos`.

W R mamy także dostęp do wielu tzw. funkcji agregujących (zwracających pojedynczą wartość liczbową) oraz dodatkowych funkcji pomocniczych, ułatwiających obliczanie różnego rodzaju wyrażeń arytmetycznych. Kilka takich funkcji zamieściliśmy w tabeli 1.3.

Informacja

Dzięki wydajnym funkcjom takim jak powyższe, w wielu wypadkach nie jest potrzebne używanie warunkowych instrukcji sterujących, takich jak np. pętla `for` znana niektórym Czytelnikom z języka C.

W R można bardzo prosto generować ciągi arytmetyczne. Ciąg o przyroście 1 bądź -1 tworzymy za pomocą dwukropka (`:`), np.

```
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> 1.5:6
[1] 1.5 2.5 3.5 4.5 5.5
> -1:10 # sprawdźmy tzw. priorytet operatora ":"
 [1] -1  0  1  2  3  4  5  6  7  8  9 10
```

Tab. 1.3. Funkcje matematyczne

Funkcja	Znaczenie
`sum(x)`	suma wszystkich elementów, $r_1 := \sum_{i=1}^{n} x_i$
`prod(x)`	iloczyn wszystkich elementów, $r_1 := \prod_{i=1}^{n} x_i$
`diff(x)`	różnica sąsiadujących ze sobą elementów, $r_j := x_{j+1} - x_j; \quad j = 1, 2, \ldots, n-1$
`mean(x)`	średnia arytmetyczna, $r_1 := \frac{1}{n} \sum_{i=1}^{n} x_i$
`var(x)`	wariancja, $r_1 := \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \texttt{mean(x)})^2$
`sd(x)`	odchylenie standardowe, $r_1 := \sqrt{\texttt{var(x)}} = \texttt{sqrt(var(x))}$
`sort(x)`	sortowanie ciągu
`rank(x)`	rangowanie elementów
`min(x)`	minimum, $r_1 := \min_{i=1,2,\ldots,n} x_i$
`max(x)`	maksimum, $r_1 := \max_{i=1,2,\ldots,n} x_i$
`cummin(x)`	minimum skumulowane, $r_j := \min_{i=1,2,\ldots,j} x_i; \quad j = 1, 2, \ldots, n$
`cummax(x)`	maksumum skumulowane, $r_j := \max_{i=1,2,\ldots,j} x_i; \quad j = 1, 2, \ldots, n$
`cumsum(x)`	skumulowana suma, $r_j := \sum_{i=1}^{j} x_i; \quad j = 1, 2, \ldots, n$
`cumprod(x)`	skumulowany iloczyn, $r_j := \prod_{i=1}^{j} x_i; \quad j = 1, 2, \ldots, n$

```
> (-1):10 # tożsame z powyższym
 [1] -1  0  1  2  3  4  5  6  7  8  9 10
> -(1:10)
 [1]  -1  -2  -3  -4  -5  -6  -7  -8  -9 -10
> 5:0
[1] 5 4 3 2 1 0
```

Ciągi o innych przyrostach są konstruowane za pomocą funkcji `seq()`, np.

```
> seq(0, 10, 2) # od 0 do 10 co 2, podobnie:
[1]  0  2  4  6  8 10
> seq(11.5, -3, by=-3.7)
[1] 11.5  7.8  4.1  0.4
> seq(0.0, 1.0, length=5) # ciąg o ustalonej długości, przyrost wyznacza R
[1] 0.00 0.25 0.50 0.75 1.00
```

Możliwe jest także tworzenie wektorów za pomocą powtórzeń (replikacji) zadanych wartości. Rozważmy następujące przykłady:

```
> rep(1, 5)
[1] 1 1 1 1 1
> rep(c(1,2), 5)
 [1] 1 2 1 2 1 2 1 2 1 2
> rep(c(1,2), each=5)
 [1] 1 1 1 1 1 2 2 2 2 2
> rep(c(1,2), each=c(5,4))
Warning: first element used of 'each' argument
 [1] 1 1 1 1 1 2 2 2 2 2
> rep(c(1,2), c(5,4))
[1] 1 1 1 1 1 2 2 2 2
```

Zadanie

Poznaj szczegóły działania tej funkcji, studiując stronę pomocy:

```
> ?rep
```

1.1.2.2. Wektory logiczne

Kolejnym typem wektorów są wektory przechowujące wartości logiczne. Mamy dostęp do dwóch wbudowanych stałych, oznaczających wartość logiczną *prawda* (`TRUE`) oraz wartość *fałsz* (`FALSE`). Wektory te tworzymy za pomocą poznanych już funkcji `c()` bądź `rep()`.

```
> c(TRUE, FALSE, TRUE)
[1]  TRUE FALSE  TRUE
> rep(FALSE, 3)
[1] FALSE FALSE FALSE
```

Mamy dostęp do następujących operacji logicznych:

Operacja	Znaczenie
`! x`	negacja
`x & y`	koniunkcja
`x | y`	alternatywa
`xor(x, y)`	alternatywa wyłączająca

Działają one podobnie do operacji arytmetycznych na wektorach numerycznych, tzn. element po elemencie oraz zgodnie z regułą zawijania. Rozważmy następujące przykłady

```
> !c(TRUE, FALSE)
[1] FALSE  TRUE
```

```
> c(TRUE, FALSE, FALSE, TRUE) | c(TRUE, FALSE, TRUE, FALSE)
[1]  TRUE FALSE  TRUE  TRUE
> c(TRUE, FALSE, FALSE, TRUE) & c(TRUE, FALSE, TRUE, FALSE)
[1]  TRUE FALSE FALSE FALSE
> xor(c(TRUE, FALSE, FALSE, TRUE), c(TRUE, FALSE, TRUE, FALSE))
[1] FALSE FALSE  TRUE  TRUE
```

Informacja

Zwróćmy uwagę, że operatory koniunkcji i alternatywy w języku C są zapisywane, odpowiednio, jako && oraz ||. W języku R są one także dostępne, jednakże nie działają zgodnie z zasadą element-po-elemencie, zwracając pojedynczą wartość. Odkrycie reguły ich działania pozostawiamy jako zadanie dla Czytelnika.

Operatory porównania, które można stosować m.in. do ustalania związków między elementami w wektorach liczbowych, dają w wyniku ciągi wartości logicznych. Oto lista wspomnianych operatorów porównania:

Operator	Znaczenie
`x < y`	mniejsze
`x > y`	większe
`x <= y`	mniejsze lub równe
`x >= y`	większe lub równe
`x == y`	równość
`x != y`	nierówność

Rozważmy następujące przykłady:

```
> (1:10) <= 5
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
> (1:10) <= c(3,7) # ta reguła zawijania...
 [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
> (1:10) %% 2 == 0 # co oznaczał ten operator arytmetyczny?
 [1] FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
> rep(c(TRUE, FALSE), 4) != rep(TRUE, 8)
[1] FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
```

1.1.2.3. Wektory liczb zespolonych

W R mamy też możliwość tworzenia wektorów złożonych z liczb zespolonych. Jednostkę urojoną oznaczamy literą `i`. Stanowi ona przyrostek „doklejany" do liczby, np.

```
> 1i
[1] 0+1i
> 3+2i
[1] 3+2i
```

Wektory zespolone można tworzyć za pomocą funkcji `c()` oraz `rep()`. Dostępne są dla nich operacje arytmetyczne `+`, `-`, `/`, `*`, `^` oraz pewne funkcje dodatkowe. Oto lista podstawowych operatorów dla liczby zespolonej $z = x + iy = |z|(\cos\phi + i\sin\phi)$:

Funkcja	Znaczenie	Wyrażenie
`Re(z)`	część rzeczywista	x
`Im(z)`	część urojona	y
`Mod(z)`	moduł	$\|z\| = \sqrt{x^2+y^2}$
`Arg(z)`	argument	ϕ
`Conj(z)`	sprzężenie	$x - iy$

Rozważmy następujące przykłady:

```
> c(1i, 2i, 3i)
[1] 0+1i 0+2i 0+3i
> rep(3+2i, 3)
[1] 3+2i 3+2i 3+2i
> (1:10)*1i # tzw. sposobem
 [1] 0+ 1i 0+ 2i 0+ 3i 0+ 4i 0+ 5i 0+ 6i 0+ 7i 0+ 8i 0+ 9i 0+10i
> (2i-2)*(3+1i)
[1] -8+4i
> (2i-2)/(3+1i)
[1] -0.4+0.8i
> Re(4i)
[1] 0
> Mod(1+1i)
[1] 1.414214
```

1.1.2.4. Wektory napisów, hierarchia typów

Jeszcze innym ważnym typem elementów, które mogą być przechowywane w wektorach, są napisy bądź łańcuchy znaków (ang. *character strings*). Definiujemy je z użyciem cudzysłowu (`"..."`) bądź (równoważnie) apostrofów (`'...'`), np.

```
> 'ala ma kota'
[1] "ala ma kota"
> c("a", "kot", "ma alę")
[1] "a"      "kot"    "ma alę"
> rep("a", 4)
[1] "a" "a" "a" "a"
```

Oto najważniejsze operacje na wektorach:

```
> paste("aaa", "bbb") # łączenie napisów
[1] "aaa bbb"
> paste("aaa", "bbb", sep="") # brak separatora
[1] "aaabbb"
> paste("a", 1:10, sep="") # reguła zawijania
 [1] "a1"  "a2"  "a3"  "a4"  "a5"  "a6"  "a7"  "a8"  "a9"  "a10"
> paste("a", 1:10, sep="", collapse=", ") # 1 napis
[1] "a1, a2, a3, a4, a5, a6, a7, a8, a9, a10"
> nchar("napis") # liczba znaków
[1] 5
> cat("ala\nma\nkota\n") # wypisywanie; \n to nowy wiersz
ala
ma
kota
```

Do tej pory tworzyliśmy wektory składające się z liczb rzeczywistych, wartości logicznych, liczb zespolonych *albo* napisów. A co by się stało, gdyby tak przechowywać wartości różnych typów w jednym obiekcie? Sprawdźmy:

```
> c(TRUE, 1, 1+1i, "jeden")
[1] "TRUE"  "1"     "1+1i"  "jeden"
> c(TRUE, 1, 1+1i)
[1] 1+0i 1+0i 1+1i
> c(TRUE, 1)
[1] 1 1
> c("jeden", 1, 1+1i, TRUE)
[1] "jeden" "1"     "1+1i"  "TRUE"
```

Z przedstawionych wyżej przykładów łatwo wywnioskować, że typy danych w R tworzą ściśle określoną hierarchię. Wszystkie elementy wektora muszą być tego samego typu. W przypadku próby stworzenia obiektu składającego się z elementów różnych typów, ustalany jest typ wystarczający do reprezentacji wszystkich elementów, w następującej kolejności:

1. typ logiczny,
2. typ liczbowy,
3. typ zespolony,
4. napis.

Informacja

Z formalnego punktu widzenia typ liczbowy dzielimy jeszcze na całkowity (ang. *integer*) i zmiennopozycyjny (ang. *floating point*). Liczby wprowadzane z klawia-

tury są traktowane jako typu zmiennopozycyjnego, nawet jeśli nie podamy jawnie ich części ułamkowej.

Warto także pamiętać, że wartość TRUE jest zawsze przekształcana do liczby 1, zaś FALSE do 0. Korzystając z tego faktu możemy szybko rozwiązać następujące zadanie: mając dany wektor logiczny sprawdzić, ile znajduje się w nim wartości. Okazuje się jednak, że funkcja sum() sama dokonuje stosownego przekształcenia typów.

```
> sum(c(TRUE, FALSE, TRUE, FALSE, FALSE, FALSE, TRUE))
[1] 3
```

1.1.2.5. Obiekty nazwane, indeksowanie wektorów, braki danych

Język programowania byłby bardzo ograniczony, gdyby nie było w nim mechanizmów zapisywania wartości w pamięci. Tego typu funkcję spełniają w R tzw. *obiekty nazwane*. Identyfikowane są one przez nazwę, rozpoczynającą się od litery lub kropki. Nazwy mogą zawierać dowolne kombinacje m.in. liter, cyfr, kropek i znaku podkreślnika (_). Pamiętajmy jednak, że R rozróżnia wielkość liter.

Do tworzenia/zastępowania obiektów nazwanych służy jeden z trzech operatorów: <-, =, ->. Dwa pierwsze[3] oczekują po prawej stronie wyrażenia, a po lewej nazwy zmiennej, ostatni zaś na odwrót. Dla przykładu:

```
> sumka <- 2+2; Sumka <- 3+3
> abs(sumka - Sumka - 1) -> roz_niczka
> roz_niczka
[1] 3
```

Operacje przypisania domyślnie nie powodują wyświetlenia wyniku. Można je jednak wymusić za pomocą wzięcia w nawiasy całego wyrażenia, np.

```
> (zmienna <- 1:50)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
[23] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
[45] 45 46 47 48 49 50
```

Czytelnik z pewnością zwrócił uwagę na oznaczenia typu „[1]", wyświetlane za każdym razem przy okazji wypisywania wyniku. W ostatnim przykładzie znalazły się dodatkowo i inne tego typu oznaczenia. Określają one indeks elementu wektora, na którym leży pierwsza wypisywana w danym wierszu wartość.

Wektory w języku R, w przeciwieństwie do C, są indeksowane od jedynki. Aby poznać długość danego ciągu (a tym samym indeks ostatniego elementu), wywołujemy funkcję length():

[3] Preferowany jest operator <-.

```
> length(-3:3)
[1] 7
```

Interesujące nas elementy znajdujące się na określonych pozycjach możemy pobrać za pomocą operatora indeksowania `[·]`:

```
> w <- -10:10
> w[3] # trzeci element
[1] -8
> w[c(3,5)] # trzeci i piąty (ale wygoda!)
[1] -8 -6
> idx <- 4:10
> w[idx]
[1] -7 -6 -5 -4 -3 -2 -1
```

Można także wykluczać pewne elementy z ciągu za pomocą odwoływania się do ujemnych indeksów:

```
> abc <- 1:10
> abc <- abc[-4]
> abc
[1]  1  2  3  5  6  7  8  9 10
> abc[-c(3,6)]
[1]  1  2  5  6  8  9 10
```

Wektory można również indeksować za pomocą wektorów logicznych. Operator `[·]` przyjmuje wtedy jako argument ciąg o takiej samej długości, co ciąg, do którego wyrazów będziemy się odwoływać (jeżeli jest on krótszy, to działa reguła zawijania). Takie wektory określają, które z elementów mają być umieszczone w podciągu wynikowym (`TRUE`), a które opuszczone (`FALSE`), np.

```
> x <- 1:6
> x[c(TRUE, FALSE, TRUE, FALSE, TRUE, TRUE)]
[1] 1 3 5 6
> x[c(TRUE, FALSE)] # reguła zawijania
[1] 1 3 5
> x > 5 # wynik jest wektorem logicznym
[1] FALSE FALSE FALSE FALSE FALSE  TRUE
> x[x > 5] # cóż za siła wyrazu
[1] 6
> (BardzoLubieR <- x %% 3)
[1] 1 2 0 1 2 0
> x[BardzoLubieR == 0] # równoważnie: x[x %% 3 == 0]
[1] 3 6
```

Istnieje także specjalna stała logiczna służąca do reprezentowania braków danych: `NA` (ang. *not available data*), czyli informacji niedostępnych bądź nieznanych. Wagę jej znaczenia poznamy w rozdziale dotyczącym statystyki opisowej. Tutaj wspomnimy

tylko o funkcjach pozwalających sprawdzać, czy występują braki danych w wektorach i/lub ewentualnie je usuwać. Oto przykłady:

```
> niekompletny1 <- c(TRUE, NA, FALSE, NA, TRUE, TRUE)
> c(3, 4, 5, NA, 2, 3, 1) # NA ma swój odpowiednik w każdym typie
[1]  3  4  5 NA  2  3  1
> is.na(niekompletny1)
[1] FALSE  TRUE FALSE  TRUE FALSE FALSE
> niekompletny1[!is.na(niekompletny1)] # podobnie:
[1]  TRUE FALSE  TRUE  TRUE
> na.omit(niekompletny1)
[1]  TRUE FALSE  TRUE  TRUE
attr(,"na.action")
[1] 2 4
attr(,"class")
[1] "omit"
> niekompletny1[is.na(niekompletny1)] <- FALSE # zastępujemy
> niekompletny1
[1]  TRUE FALSE FALSE FALSE  TRUE  TRUE
```

Czasem mogą się nam przydać także funkcje kontrolne do sprawdzania, czy wynik jest nie-liczbą (NaN), bądź czy jest skończony. Zwracają one wartości logiczne. Przykładowo:

```
> is.nan(Inf/Inf)
[1] TRUE
> is.finite(-Inf)
[1] FALSE
> is.infinite(c(Inf/Inf, Inf*Inf, 1/Inf, Inf/1))
[1] FALSE  TRUE FALSE  TRUE
> is.finite(c(1, NA, NaN, Inf, -Inf))
[1]  TRUE FALSE FALSE FALSE FALSE
```

1.1.2.6. Czynniki

Wyróżnionym typem wektorów są obiekty do przechowywania zmiennych typu jakościowego. Elementy tego typu obiektów mogą występować tylko na z góry określonej, skończonej liczbie poziomów (rzędu kilku czy kilkunastu). Są to tak zwane *wektory czynnikowe* (ang. *factors*). Każda kategoria (klasa, poziom) czynnika może być identyfikowana za pomocą dowolnego ciągu znaków. Dla przykładu:

```
> wek <- c("Ala", "Ola", "Jola", "Ala", "Ala", "Jola", "Ala") # 3 wartości
> fek <- factor(wek) # przekształcenie na wektor czynnikowy
> fek
[1] Ala  Ola  Jola Ala  Ala  Jola Ala
Levels: Ala Jola Ola
```

```
> levels(fek) # wektor nazw poziomów
[1] "Ala"  "Jola" "Ola"
> levels(fek)[1] <- "Michał" # można zmienić
> fek
[1] Michał Ola    Jola   Michał Michał Jola   Michał
Levels: Michał Jola Ola
> (fek2 <- factor(rep(1:5, 2)))
 [1] 1 2 3 4 5 1 2 3 4 5
Levels: 1 2 3 4 5
> levels(fek2) <- c("A", "B", "C", "D", "E")
> fek2
 [1] A B C D E A B C D E
Levels: A B C D E
```

1.1.3. Listy

Zauważyliśmy wcześniej, iż wektory mogą przechowywać elementy tylko jednego typu. Tego ograniczenia pozbawione są *listy*. Dla przykładu:

```
> L <- list(1, "napis", TRUE, 5:10)
> L
[[1]]
[1] 1

[[2]]
[1] "napis"

[[3]]
[1] TRUE

[[4]]
[1]  5  6  7  8  9 10
```

Dostęp do poszczególnych elementów listy uzyskujemy za pomocą podwójnych nawiasów kwadratowych, czyli tzw. *operatora wydobywania*:

```
> L[[1]]
[1] 1
> L[[4]]
[1]  5  6  7  8  9 10
> L[[4]][3] # trzeci element wektora będącego 4 elementem listy
[1] 7
```

Poszczególne elementy listy mogą być także nazwane (identyfikowane za pomocą nazwy):

```
> el <- list(1, komunikat="napis", TRUE, wartosci=5:10)
> el
```

```
[[1]]
[1] 1

$komunikat
[1] "napis"

[[3]]
[1] TRUE

$wartosci
[1]  5  6  7  8  9 10
```

Dostęp do kolejnego (w naszym przypadku — drugiego) elementu możemy uzyskać na jeden z trzech sposobów:

```
> el[[2]]
[1] "napis"
> el$komunikat
[1] "napis"
> el["komunikat"]
$komunikat
[1] "napis"
```

Nazwy elementów można zmieniać za pomocą modyfikacji atrybutu `names`:

```
> names(el)
[1] ""          "komunikat" ""          "wartosci"
> names(el)[1:2] <- c("jedynka", "uwaga")
> el
$jedynka
[1] 1

$uwaga
[1] "napis"

[[3]]
[1] TRUE

$wartosci
[1]  5  6  7  8  9 10
```

Informacja

W podobnym sposób można też nazywać elementy wektorów atomowych tworzonych przy użyciu funkcji `c()`.

1.1.4. Ramki danych

Szczególnego rodzaju listami są *ramki danych* (ang. *data frames*). Są to listy przechowujące wektory o tej samej długości. Przechowywane dane wyświetlane są w postaci dwuwymiarowej tabeli, w której kolumnami (zmiennymi) są wspomniane wektory, a wierszami (obserwacjami) — elementy wektorów o tych samych indeksach. Na przykład:

```
> imiona <- c("Ania", "Kasia", "Janek", "Borys")
> wiek <- c(8, 5, 3, 9)
> lubiaLody <- c(TRUE, TRUE, FALSE, TRUE)
> dzieci <- data.frame(imiona, wiek, lubiaLody)
> dzieci
  imiona wiek lubiaLody
1   Ania    8      TRUE
2  Kasia    5      TRUE
3  Janek    3     FALSE
4  Borys    9      TRUE
```

Nazwy kolumn biorą się domyślnie z nazw argumentów. Możemy je jednak zainicjować w podobny sposób jak w przypadku listy bądź zmienić za pomocą funkcji `names()`:

```
> dzieci <- data.frame(imie=imiona, wiek, lubiaLody)
> dzieci
   imie wiek lubiaLody
1  Ania    8      TRUE
2 Kasia    5      TRUE
3 Janek    3     FALSE
4 Borys    9      TRUE
> names(dzieci)[3] <- "lubiLody"
> dzieci
   imie wiek lubiLody
1  Ania    8     TRUE
2 Kasia    5     TRUE
3 Janek    3    FALSE
4 Borys    9     TRUE
```

A teraz zobaczmy, w jaki sposób uzyskać dostęp do poszczególnych składowych:

```
> dzieci[1,1] # pojedyncza komórka
[1] Ania
Levels: Ania Borys Janek Kasia
> dzieci[2:4, c(1,3)]
   imie lubiLody
2 Kasia     TRUE
3 Janek    FALSE
4 Borys     TRUE
> dzieci[1,] # pierwszy wiersz  (brak wartości w zakresie = cały zakres)
  imie wiek lubiLody
1 Ania    8     TRUE
```

```
> dzieci[,1] # pierwsza kolumna (brak wartości w zakresie = cały zakres)
[1] Ania  Kasia Janek Borys
Levels: Ania Borys Janek Kasia
> dzieci[1] # pierwsza kolumna - nazwana
   imie
1  Ania
2 Kasia
3 Janek
4 Borys
> dzieci$imie
[1] Ania  Kasia Janek Borys
Levels: Ania Borys Janek Kasia
```

Informacja

Widzimy, że kolumna `imie` jest typu czynnikowego. Przy tworzeniu ramek danych napisy są domyślnie automatycznie konwertowane (dla wygody) na czynniki. Możemy temu zapobiec poprzez ustawienie odpowiedniego argumentu funkcji `data.frame()`:

```
> dzieci2 <- data.frame(imiona, wiek, lubiaLody,
+    stringsAsFactors=FALSE)
> dzieci2[,1] # napisy
[1] "Ania"  "Kasia" "Janek" "Borys"
```

1.1.5. Macierze

Ostatnim typem danych, który omówimy, są *macierze* (ang. *matrices*). Do ich tworzenia służy funkcja `matrix()`. Jako parametr pobiera ona wektor pożądanych elementów oraz wymiar macierzy. Na przykład:

```
> matrix(1:6, nrow=2, ncol=3) # dwa wiersze, trzy kolumny
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> matrix(1:6, nrow=2) # dwa wiersze także, liczbę kolumn sam wyznaczy R
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> matrix(1:6, nrow=2, byrow=TRUE) # wprowadzamy dane wierszami
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

Listę ważniejszych operacji i funkcji macierzowych przedstawia poniższa tabela.

Operacja	Znaczenie
`A %*% B`	mnożenie macierzy
`det(A)`	wyznacznik macierzy
`t(A)`	transpozycja
`solve(A, b)`	rozwiązanie układu liniowego postaci $\mathbf{Ax} = \mathbf{b}$
`diag(A)`	diagonala (jako wektor)
`eigen(A)`	wartości własne i wektory własne

Rozważmy następujące przykłady:

```
> (A <- matrix(c(2,0,0,2), nrow=2))
     [,1] [,2]
[1,]    2    0
[2,]    0    2
> (B <- matrix(1:4, nrow=2))
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> A+B   # element po elemencie
     [,1] [,2]
[1,]    3    3
[2,]    2    6
> A*B   # element po elemencie
     [,1] [,2]
[1,]    2    0
[2,]    0    8
> A%*%B # mnożenie macierzy
     [,1] [,2]
[1,]    2    6
[2,]    4    8
> det(B)
[1] -2
> eigen(B) # to akurat łatwo policzyć w pamięci
$values
[1]  5.3722813 -0.3722813

$vectors
           [,1]       [,2]
[1,] -0.5657675 -0.9093767
[2,] -0.8245648  0.4159736
> t(B)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

1.2. Zadania rozwiązane

Zadanie 1.1. Wiedząc, że $\lim_{n\to\infty}\sqrt{\sum_{i=1}^{n}\frac{6}{i^2}} = \pi$, wyznacz przybliżenie ludolfiny (czyli liczby π) dla $n = 100, 1000, 10000$, oraz 100000.

Rozwiązanie. Wyznaczenie każdej z poszukiwanych wartości jest bardzo proste. Wystarczy skorzystać z operatorów arytmetycznych i funkcji `sum()` oraz `sqrt()`.

```
> sqrt(sum(6/(1:100)^2))
[1] 3.132077
> sqrt(sum(6/(1:1000)^2))
[1] 3.140638
> sqrt(sum(6/(1:10000)^2))
[1] 3.141497
> sqrt(sum(6/(1:100000)^2))
[1] 3.141583
```

⊡

Zadanie 1.2. Niech $\mathbf{x} = (4, 1, 3, 2)'$.
a) Wyznacz $\mathbf{x}'\mathbf{x}$ oraz $\mathbf{x}\mathbf{x}'$.
b) Wyznacz długość wektora $\mathbf{x}$, a następnie unormuj ten wektor.
c) Scentruj wektor $\mathbf{x}$.
d) Zestandaryzuj wektor $\mathbf{x}$.
e) Zapisz $\mathbf{x}$ jako macierz i wyznacz transpozycję tej macierzy.

Rozwiązanie. Iloczyn $\mathbf{x}'\mathbf{x}$ możemy wyznaczyć na dwa sposoby:

```
> x <- c(4,1,3,2)
> crossprod(x)
     [,1]
[1,]   30
> t(x) %*% x # równoważnie
     [,1]
[1,]   30
```

Porównajmy ten wynik z iloczynem $\mathbf{x}\mathbf{x}'$, który również możemy wyznaczyć na dwa sposoby:

```
> tcrossprod(x)
     [,1] [,2] [,3] [,4]
[1,]   16    4   12    8
[2,]    4    1    3    2
[3,]   12    3    9    6
[4,]    8    2    6    4
> x %*% t(x)  # równoważnie
```

```
     [,1] [,2] [,3] [,4]
[1,]   16    4   12    8
[2,]    4    1    3    2
[3,]   12    3    9    6
[4,]    8    2    6    4
```

Wyznaczanie długości wektora $\mathbf{x}$ i jego normowanie przebiega nastepująco:

```
> (d <- sqrt(crossprod(x)))  # wyznaczanie długości wektora
         [,1]
[1,] 5.477226
> x/d  # normowanie wektora
[1] 0.7302967 0.1825742 0.5477226 0.3651484
```

Centrowanie i standaryzację wektora $\mathbf{x}$ można przeprowadzić dwiema metodami

```
> x-mean(x)  # centrowanie wektora
[1]  1.5 -1.5  0.5 -0.5
> (x-mean(x))/sd(x)  # standaryzacja wektora
[1]  1.1618950 -1.1618950  0.3872983 -0.3872983
```

lub

```
> scale(x, scale=FALSE)  # centrowanie wektora
     [,1]
[1,]  1.5
[2,] -1.5
[3,]  0.5
[4,] -0.5
attr(,"scaled:center")
[1] 2.5
> scale(x)  # standaryzacja wektora
           [,1]
[1,]  1.1618950
[2,] -1.1618950
[3,]  0.3872983
[4,] -0.3872983
attr(,"scaled:center")
[1] 2.5
attr(,"scaled:scale")
[1] 1.290994
```

Na zakończenie zapiszemy $\mathbf{x}$ jako macierz i wyznaczymy transpozycję tej macierzy:

```
> y <- as.matrix(x)
> t(y)
     [,1] [,2] [,3] [,4]
[1,]    4    1    3    2
```

□

Zadanie 1.3. Wyznacz iloczyn skalarny wektorów $\mathbf{x} = (4, 1, 3, 2)'$ i $\mathbf{y} = (1, -1, 3, 0)'$.

Rozwiązanie. Iloczyn skalarny wektorów $\mathbf{x}$ i $\mathbf{x}$ możemy wyznaczyć na dwa sposoby:

```
> x <- c(4,1,3,2)
> y <- c(1,-1,3,0)
> t(x) %*% y
     [,1]
[1,]   12
> crossprod(x,y)
     [,1]
[1,]   12
```

⊡

Zadanie 1.4. Utwórz macierz diagonalną $\mathbf{D}$ oraz macierz identycznościową $\mathbf{I}_3$, gdzie

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \quad \mathbf{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Rozwiązanie. Aby utworzyć obie macierze wystarczy posłużyć się funkcją `diag()`:

```
> d <- c(1,2,3)
> diag(d)  #macierz diagonalna
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
> diag(rep(1,3))  #macierz identycznościowa
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

⊡

Zadanie 1.5. Wyznacz iloczyny $\mathbf{AB}$ i $\mathbf{BA}'$ oraz iloczyn Kroneckera $\mathbf{A} \otimes \mathbf{B}$ macierzy

$$\mathbf{A} = \begin{pmatrix} 1 & 5 \\ 1 & 2 \\ 3 & 8 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 4 \\ 3 & 2 \end{pmatrix}.$$

Rozwiązanie. Po wprowadzeniu danych wystarczy wywołać odpowiednie funkcje:

```
> A <- matrix(c(1,5,1,2,3,8),3,2, byrow=TRUE)
> B <- matrix(c(1,4,3,2),2,2, byrow=TRUE)
> A %*% B
```

```
     [,1] [,2]
[1,]   16   14
[2,]    7    8
[3,]   27   28
> B %*% t(A)
     [,1] [,2] [,3]
[1,]   21    9   35
[2,]   13    7   25
> kronecker(A,B)
     [,1] [,2] [,3] [,4]
[1,]    1    4    5   20
[2,]    3    2   15   10
[3,]    1    4    2    8
[4,]    3    2    6    4
[5,]    3   12    8   32
[6,]    9    6   24   16
```

⊡

Zadanie 1.6. Wyznacz ślad i wyznacznik macierzy $\mathbf{A}$ oraz macierz odwrotną do macierzy $\mathbf{A}$, gdzie

$$\mathbf{A} = \begin{pmatrix} 3 & 4 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{pmatrix}.$$

Rozwiązanie. Rozwiązania uzyskujemy bezpośrednio po wywołaniu odpowiednich funkcji:

```
> A <- matrix(c(3,4,6,1,2,3,5,7,9),3,3, byrow=TRUE)
> sum(diag(A))  # slad tr(A)
[1] 14
> det(A)  # wyznacznik det(A)
[1] -3
> solve(A)  # macierz odwrotna
     [,1]       [,2]          [,3]
[1,]    1 -2.0000000  7.105427e-16
[2,]   -2  1.0000000  1.000000e+00
[3,]    1  0.3333333 -6.666667e-01
```

⊡

Zadanie 1.7. Niech

$$\mathbf{A} = \begin{pmatrix} 10 & 3 & 9 \\ 3 & 40 & 8 \\ 9 & 8 & 15 \end{pmatrix}.$$

a) Wyznacz wartości własne i wektory własne macierzy $\mathbf{A}$.

b) Wyznacz rozkład Choleskiego macierzy $\mathbf{A}$, tzn. znajdź macierz górną trójkątną $\mathbf{U}$ taką, że $\mathbf{A} = \mathbf{U}'\mathbf{U}$.
c) Wyznacz macierz ortogonalną $\mathbf{H}$ oraz macierz diagonalną $\mathbf{D}$ taką, że $\mathbf{A} = \mathbf{HDH}'$.
d) Wyznacz pierwiastek z macierzy $\mathbf{A}$ tzn. macierz symetryczną $\mathbf{B}$ taką, że $\mathbf{A} = \mathbf{BB}$.

Rozwiązanie. Po wpisaniu odpowiedniego polecenia otrzymamy wartości własne macierzy posortowane od maksymalnej do mininimalnej

```
> A <- matrix(c(10,3,9,3,40,8,9,8,15),3,3, byrow=TRUE)
> eigen(A)$values
[1] 43.27506 18.74543  2.97950
```

oraz unormowane wektory własne (tzn. o długości równej jeden), odpowiadające kolejnym wartościom własnym:

```
> eigen(A)$vectors
           [,1]       [,2]       [,3]
[1,] -0.1701097 -0.6093732  0.77442043
[2,] -0.9326923  0.3531984  0.07304764
[3,] -0.3180373 -0.7098699 -0.62844016
```

Rozkład Choleskiego macierzy $\mathbf{A}$ przedstawia się następująco:

```
> chol(A)
         [,1]      [,2]      [,3]
[1,] 3.162278 0.9486833 2.8460499
[2,] 0.000000 6.2529993 0.8475933
[3,] 0.000000 0.0000000 2.4862795
```

Aby znaleźć macierz ortogonalną $\mathbf{H}$ oraz macierz diagonalną $\mathbf{D}$ taką, że $\mathbf{A} = \mathbf{HDH}'$, wystraczy wykonać następujące polecenia:

```
> (H <- eigen(A)$vectors)
           [,1]       [,2]       [,3]
[1,] -0.1701097 -0.6093732  0.77442043
[2,] -0.9326923  0.3531984  0.07304764
[3,] -0.3180373 -0.7098699 -0.62844016
> (D <- diag(eigen(A)$values))
         [,1]     [,2]   [,3]
[1,] 43.27506  0.00000 0.0000
[2,]  0.00000 18.74543 0.0000
[3,]  0.00000  0.00000 2.9795
```

Łatwo sprawdzić, że po przemnożeniu tych macierzy otrzymamy naszą wyjściową macierz $\mathbf{A}$:

```
> H %*% D %*% t(H)
     [,1] [,2] [,3]
[1,]   10    3    9
[2,]    3   40    8
```

```
[3,]    9    8   15
```

Pierwiastkiem z macierzy $\mathbf{A}$ jest taka macierz symetryczna $\mathbf{B}$, dla której zachodzi $\mathbf{A} = \mathbf{BB}$. Można pokazać, że $\mathbf{B} = \mathbf{H}\,\mathbf{sqrt}(\mathbf{D})\,\mathbf{H}'$, gdzie $\mathbf{sqrt}(\mathbf{D})$ oznacza macierz mającą na diagonali pierwiastki z elementów diagonali macierzy $\mathbf{D}$, tzn.

```
> sqrt(D)
         [,1]   [,2]     [,3]
[1,] 6.578379 0.0000 0.000000
[2,] 0.000000 4.3296 0.000000
[3,] 0.000000 0.0000 1.726123
```

Tak więc poszukiwany pierwiastek ma postać:

```
> B <- H %*% sqrt(D) %*% t(H)
> B
          [,1]      [,2]      [,3]
[1,] 2.8332980 0.2095137 1.3887139
[2,] 0.2095137 6.2719540 0.7865728
[3,] 1.3887139 0.7865728 3.5288492
```

Nietrudno sprawdzić, że faktycznie zachodzi $\mathbf{A} = \mathbf{BB}$:

```
> B %*% B
     [,1] [,2] [,3]
[1,]   10    3    9
[2,]    3   40    8
[3,]    9    8   15
```

⊡

Zadanie 1.8. Niech

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Wyznacz rozkład macierzy $\mathbf{A}$ na wartości szczególne, tzn. znajdź macierze ortogonalne $\mathbf{U}$ i $\mathbf{V}$ oraz macierz pseudodiagonalną $\mathbf{D}$ taką, że $\mathbf{A} = \mathbf{UDV}'$.

Rozwiązanie. Rozkład macierzy na wartości szczególne dokonuje się za pomocą jednej funkcji `svd()`:

```
> A <- matrix(c(1,1,0,1,1,0),3,2,byrow=TRUE)
> (svd.A <- svd(A,nu = nrow(A), nv = ncol(A)))
$d
[1] 1.732051 1.000000

$u
           [,1]          [,2]       [,3]
[1,] -0.8164966 -1.665335e-16 -0.5773503
```

```
[2,] -0.4082483 -7.071068e-01  0.5773503
[3,] -0.4082483  7.071068e-01  0.5773503

$v
           [,1]       [,2]
[1,] -0.7071068  0.7071068
[2,] -0.7071068 -0.7071068
```

Warto sprawdzić, czy wyznaczone powyżej macierze spełniają faktycznie równanie $\mathbf{A} = \mathbf{UDV}'$:

```
> (ds <- diag(svd.A$d))
         [,1] [,2]
[1,] 1.732051    0
[2,] 0.000000    1
> (us <- as.matrix(svd.A$u[, 1:2]))
           [,1]          [,2]
[1,] -0.8164966 -1.665335e-16
[2,] -0.4082483 -7.071068e-01
[3,] -0.4082483  7.071068e-01
> (vs <- as.matrix(svd.A$v))
           [,1]       [,2]
[1,] -0.7071068  0.7071068
[2,] -0.7071068 -0.7071068
> us %*% ds %*% t(vs)
             [,1]          [,2]
[1,] 1.000000e+00  1.000000e+00
[2,] 5.551115e-17  1.000000e+00
[3,] 1.000000e+00 -1.665335e-16
```

Okazuje się, że z dokładnością do błędów numerycznych dostaliśmy naszą wyjściową macierz $\mathbf{A}$. Podobnie możemy przekonać się, że zachodzi związek $\mathbf{D} = \mathbf{U}'\mathbf{AV}$:

```
> t(us) %*% A %*% vs
             [,1]         [,2]
[1,] 1.732051e+00 3.330669e-16
[2,] 1.110223e-16 1.000000e+00
```

⊡

1.3. Zadania do samodzielnego rozwiązania

⋆ **Zadanie 1.9.** Indeksem Hirscha dla uporządkowanego nierosnąco ciągu $\mathbf{C} = (c_1, c_2, \ldots, c_n)$, $c_i \geqslant c_j$ dla $i \leqslant j$, $c_1 > 0$, nazywamy wartość:

$$h(\mathbf{C}) = \max\{i : c_i \geqslant i\} = \sum_{i=1}^{n} \mathbf{1}(c_i \geqslant i), \tag{1.1}$$

gdzie $\mathbf{1}(w)$ oznacza tzw. funkcję indykatorową, przyjmującą wartość 1, jeżeli warunek w jest spełniony, oraz 0 — w przeciwnym przypadku. Wyznacz za pomocą R wartość indeksu Hirscha np. dla następujących wektorów: $(43, 12, 9, 4, 3, 2, 0, 0)$, $(1, 1, 1, 1, 1, 1)$, $(32, 74, 24, 64, 123, 6, 0, 35, 1, 1, 1, 3, 64, 0, 0)$.

⋆ **Zadanie 1.10.** Indeksem Egghego dla uporządkowanego nierosnąco ciągu $\mathbf{C} = (c_1, c_2, \ldots, c_n)$, $c_i \geqslant c_j$ dla $i \leqslant j$, $c_1 > 0$, nazywamy wartość

$$g(\mathbf{C}) = \max\left\{i : \sum_{j=1}^{i} c_j \geqslant i^2\right\} = \sum_{i=1}^{n} \mathbf{1}\left(\sum_{j=1}^{i} c_j \geqslant i^2\right). \tag{1.2}$$

Wyznacz za pomocą R wartość indeksu g dla różnych wektorów.

Zadanie 1.11. Wyznacz rozkład na wartości szczególne (osobliwe) macierzy

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \\ 2 & 0 & 1 & 1 \end{pmatrix}$$

i sprawdź, czy wyznaczone macierze $\mathbf{U}$, $\mathbf{D}$ i $\mathbf{V}$ spełniają równanie $\mathbf{A} = \mathbf{UDV}'$.

1.4. Wskazówki i odpowiedzi do zadań

Ad zad. 1.9. W przypadku danych ciągów nieuporządkowanych, posortuj je funkcją `sort()` z odpowiednim parametrem. Skorzystaj z funkcji `sum()`.

Ad zad. 1.10. Użyj m.in. funkcji `cumsum()`.

Ad zad. 1.11.

$$\mathbf{U} = \begin{pmatrix} 0.5417743 & 0.7071068 & -0.4544013 \\ 0.5417743 & -0.7071068 & -0.4544013 \\ 0.6426206 & 2.220446 \times 10^{-16} & 0.7661846 \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} 3.5699528 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1.120463 \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} 0.6635353 & 1.110223 \times 10^{-16} & 0.5565257 & -0.5 \\ 0.303519 & 1.110223 \times 10^{-16} & -0.8110957 & -0.5 \\ 0.4835272 & -0.7071068 & -0.127285 & 0.5 \\ 0.4835272 & 0.7071068 & -0.127285 & 0.5 \end{pmatrix}$$

Statystyka opisowa 2

2.1. Wprowadzenie

Statystyka opisowa stawia sobie za cel stworzenie wstępnego „obrazu" badanego obiektu bądź zjawiska, jaki wyłania się z interesującego nas zbioru danych pochodzących z obserwacji, eksperymentu, bądź będących wynikiem symulacji. Po dokonaniu wstępnego przygotowania danych (m.in. usunięciu błędów, uzgodnieniu formatów, jednostek itp.), statystyka opisowa stanowi podstawowy, eksploracyjny etap analizy danych.

Wspomniane obserwacje (pomiary) mogą dotyczyć jednej bądź wielu własności badanych obiektów, nazywanych dalej cechami statystycznymi. Ogólnie rzecz biorąc rozpatrywać będziemy dwa typy cech: mierzalne i niemierzalne.

Cechy mierzalne, zwane także *ilościowymi* (ang. *quantitative*) są wyrażane za pomocą wartości liczbowych, często w określonych jednostkach. Mogą to być zarówno tzw. cechy *ciągłe*, czyli przyjmujące dowolne wartości z danego przedziału, np. wzrost osoby (w centymetrach), czas bezawaryjnej pracy urządzenia (w godzinach), jak i cechy *dyskretne*, przyjmujące skończoną lub przeliczalną liczbę wartości, np. liczba wypadków na danym skrzyżowaniu.

Cechy niemierzalne, zwane również *jakościowymi* (ang. *qualitative* lub *categorical*) wyrażane są w sposób opisowy poprzez wskazanie kategorii, klasy bądź poziomu, jaki może przyjmować owa cecha. Przykładowo, cecha „płeć" może przyjmować dwie wartości: „kobieta" bądź „mężczyzna", zaś cecha „francuska marka samochodu" którąś wartość ze zbioru: „Citroen", „Peugeot", „Renault".

Typ cechy determinuje możliwe do zastosowania sposoby opisu danej cechy (por. [7, 9]). Poniżej przedstawiamy najczęściej stosowane narzędzia statystyki opisowej wraz ze sposobem ich uzyskania za pomocą środowiska R.

1. Cechy jakościowe:
   (a) Tabele:
      i. empiryczny rozkład liczności (`table(...)`),
      ii. empiryczny rozkład częstości (`prop.table(table(...))`).
   (b) Narzędzia graficzne:
      i. wykres słupkowy (`barplot(...)`),
      ii. wykres kołowy (`pie(...)`).
2. Cechy ilościowe:
   (a) Charakterystyki liczbowe (statystyki próbkowe):
      i. Charakterystyki położenia:
         — średnia (`mean(...)`),
         — średnia ucięta (`mean(..., trim=...)`),
         — moda,

— mediana (`median(...)`),
— kwantyle dowolnego rzędu (`quantile(...)`),
— wartość minimalna (`min(...)`,
— wartość maksymalna (`max(...)`.

ii. Charakterystyki rozproszenia:
— wariancja (`var(...)`),
— odchylenie standardowe (`sd(...)`),
— rozstęp międzykwartylowy (`IQR(...)`),
— rozstęp (`diff(range(...))`).

iii. Charakterystyki kształtu:
— skośność,
— kurtoza.

(b) Narzędzia graficzne:
i. wykres skrzynkowy (pudełkowy) (`boxplot(...)`),
ii. histogram (`hist(...)`),
iii. wykres łodygowo-liściowy (`stem(...)`).

Podczas ćwiczeń zapoznamy się także z metodami ładowania zbiorów danych z plików, transformacji danych i dzielenia zbiorów na podzbiory w zależności od kategorii.

2.2. Zadanie rozwiązane

2.2.1. Dane typu jakościowego

Zadanie 2.1. Pewna grupa studentów Wydziału MiNI PW została poproszona przez pracownicę dziekanatu o wybranie swego przedstawiciela. Kandydatami do tej zaszczytnej funkcji byli: Kasia Złotowłosa (ZK), Jerzy Wąsaty (WJ), Stefan Pulchny (PS) i Cecylia Kowalska (KC). W głosowaniu wzięło udział 25 osób. Jesteś członkiem okolicznościowej komisji i – jako znawca programu R – zostałeś poproszony o wstępne zanalizowanie wyników głosowania celem opublikowania ich na internetowej stronie samorządu.

ZK, ZK, KC, PS, KC, PS, PS, ZK, KC, KC, PS, KC, KC, WJ, PS, KC, PS, ZK, KC, WJ, PS, ZK, WJ, KC, KC

Rozwiązanie. Pierwszą czynnością, którą należy wykonać, jest wprowadzenie danych. Dysponujemy 25 kartkami do głosowania, ich wyniki możemy zapisać w postaci wektora napisów. Używać będziemy przy tym tylko inicjałów w formie takiej, jak podano powyżej w nawiasach.

```
> glosy <- c("ZK", "ZK", "KC", "PS", "KC", "PS", "PS", "ZK", "KC", "KC",
+            "PS", "KC", "KC", "WJ", "PS", "KC", "PS", "ZK", "KC", "WJ",
+            "PS", "ZK", "WJ", "KC", "KC")
> glosy <- factor(glosy) # konwersja na typ czynnikowy-wyodrębnienie klas
```

```
> length(glosy)
[1] 25
```

Są to niewątpliwie dane typu jakościowego: interesująca nas cecha, czyli kandydat na reprezentanta grupy, może przyjąć jedną z 4 wartości, a każda z nich odpowiada osobie, na którą można zagłosować w wyborach.

Zliczenia głosów uzyskanych przez każdego kandydata możemy dokonać za pomocą funkcji `table()`.

```
> glosyTab <- table(glosy) # tabela liczności
> print(glosyTab)
glosy
KC PS WJ ZK
10  7  3  5
```

Jak widać, zwyciężyła Cecylia Kowalska, która otrzymała 10 głosów – najwięcej ze wszystkich kandydatów. Wyniki możemy także przedstawić w postaci *tabeli częstości.*

```
> prop.table(glosyTab)
glosy
  KC   PS   WJ   ZK
0.40 0.28 0.12 0.20
```

Wynika stąd, iż Cecylia – choć zwyciężyła w wyborach – otrzymała tylko 48% wszystkich głosów. Zwróćmy uwagę, że funkcja `prop.table()` przyjmuje jako parametr tabelę liczności, a nie wektor `glosy`.

Jeżeli chcemy mieć dostęp oddzielnie do 4-wyrazowego wektora liczby głosów oraz do wektora odpowiadających głosom kategorii, możemy wykonać następujące polecenia.

```
> osoby <- names(glosyTab) # wektor nazw kategorii (inicjały kandydatów)
> liczbaGlosow <- as.vector(glosyTab) # liczba głosów na kandydata
> print(osoby)
[1] "KC" "PS" "WJ" "ZK"
> print(liczbaGlosow)
[1] 10  7  3  5
> osoby[2]
[1] "PS"
> liczbaGlosow[2] # wyniki drugiej osoby
[1] 7
```

Za ich pomocą da się odtworzyć zawartość wektora `glosy` (z dokładnością do permutacji wyrazów).

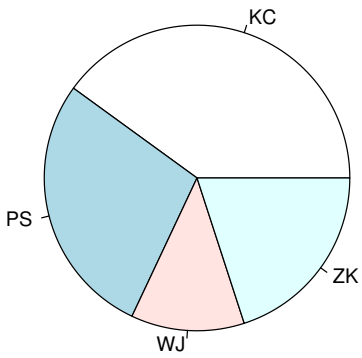
```
> rep(osoby, liczbaGlosow)
 [1] "KC" "KC" "KC" "KC" "KC" "KC" "KC" "KC" "KC" "KC" "PS" "PS" "PS"
[14] "PS" "PS" "PS" "PS" "WJ" "WJ" "WJ" "ZK" "ZK" "ZK" "ZK" "ZK"
```

Umiejętność przechodzenia z różnych postaci danych do innych jest istotna na przykład gdy mamy dostęp do informacji już wstępnie przetworzonych (zliczonych). W takim

wypadku wprowadzilibyśmy je używając właśnie wektorów liczności oraz nazw kategorii.

W praktyce na ogół bardziej przyjaznymi, niż tabele, sposobami opisu danych są wykresy. Chyba najpopularniejszym z nich jest *wykres kołowy* (ang. *pie chart*), który możemy utworzyć za pomocą wywołania:

```
> pie(glosyTab)
```



lub równoważnie

```
> pie(liczbaGlosow, labels=osoby)
```

Informacja

Spróbujmy także innych ustawień:

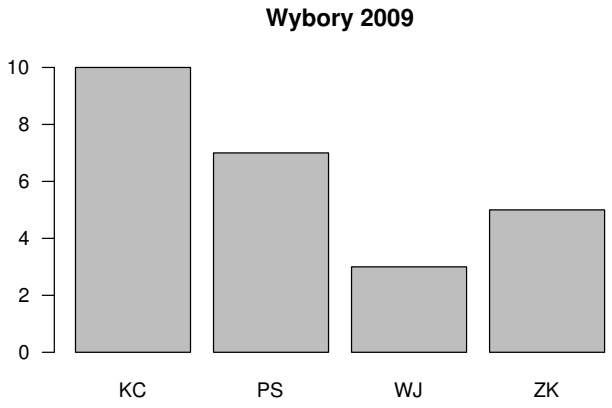
```
> pie(glosyTab, col=c("red", "blue", "yellow", "green"))
> pie(glosyTab, col=heat.colors(4))
> pie(glosyTab, border=NULL, radius=1, main="Wyniki głosowania",
+     labels=c("Cecylia", "Stefan", "Jerzy", "Kasia"))
> pie(glosyTab, labels=paste(osoby, "-", liczbaGlosow))
```

Innym sposobem wizualizacji danych jakościowych jest *wykres słupkowy* (por. rys. 2.1).

```
> barplot(glosyTab, main="Wybory 2009", las=1)
```

Informacja

Rozważmy również:



Rys. 2.1. Wykres słupkowy

```
> barplot(liczbaGlosow, names=osoby)
> barplot(prop.table(glosyTab), names=as.vector(prop.table(glosyTab)),
+     horiz=T, legend=names(glosyTab), col=rainbow(4))
```

Z powyższych wykresów szybko możemy wywnioskować, jak układają się liczby bądź proporcje głosów oddanych na poszczególnych kandydatów. Nie na darmo mówimy, że jeden obraz wart jest tysiąca słów. ⊡

Zadanie 2.2. W pliku `samochody.csv`[1] zamieszczono historyczne dane dotyczące parametrów samochodów kilku wybranych marek.

1. Zmienna `mpg` zawiera dane odpowiadające liczbie mil, przejechanych przez dany samochód na jednym galonie paliwa. Utwórz zmienną `zp` opisującą zużycie paliwa mierzone w litrach na 100 kilometrów.
2. Utwórz nową zmienną, charakteryzującą zużycie paliwa, poprzez kategoryzację na następujące klasy:

Kod kategorii	Zużycie paliwa [l/100 km]
mało	mniejsze niż 7
średnio	nie mniejsze niż 7, lecz mniejsze niż 10
dużo	nie mniejsze niż 10

3. Dla otrzymanych danych utwórz i omów wykres słupkowy.

[1] `http://www.ibspan.waw.pl/~pgrzeg/stat_lab/samochody.csv`

Rozwiązanie. Baza samochodów przechowywana jest w pliku typu CSV (ang. *comma-separated values*). Jest to zwykły plik tekstowy o określonej strukturze. Można go podejrzeć korzystając np. programu *Notatnik*:

```
mpg;cylindry;moc;przysp;rok;waga;producent;marka;model;cena;legenda
43,1;4;48;21,5;78;1985;2;Volkswagen;Rabbit Dl ;2400;America=1
36,1;4;66;14,4;78;1800;1;Ford      ;Fiesta    ;1900;Europe=2
32,8;4;52;19,4;78;1985;3;Mazda     ;GLC Deluxe;2200;Japan =3
39,4;4;70;18,6;78;2070;3;Datsun    ;B210 GX   ;2725;
36,1;4;60;16,4;78;1800;3;Honda     ;Civic CVCC;2250;
...
```

Pierwszy wiersz pliku określa nazwy kolumn (zmiennych). W kolejnych wierszach mamy dostęp do informacji o poszczególnych samochodach. Załadujmy ten plik jako ramkę danych R. Służy do tego funkcja `read.csv2()`, przy czym w nawiasie wskazuje się ścieżkę dostępu do interesującego nas pliku.

```
> samochody <-
+    read.csv2("http://www.ibspan.waw.pl/~pgrzeg/stat_lab/samochody.csv")
```

Sprawdźmy, czy plik został zinterpretowany w oczekiwany sposób:

```
> class(samochody) # czy jest to ramka danych?
[1] "data.frame"
> head(samochody)  # wyświetl kilka pierwszych wierszy
   mpg cylindry moc przysp rok waga producent      marka      model
1 43.1        4  48   21.5  78 1985         2 Volkswagen Rabbit Dl
2 36.1        4  66   14.4  78 1800         1 Ford       Fiesta
3 32.8        4  52   19.4  78 1985         3 Mazda      GLC Deluxe
4 39.4        4  70   18.6  78 2070         3 Datsun     B210 GX
5 36.1        4  60   16.4  78 1800         3 Honda      Civic CVCC
6 19.9        8 110   15.5  78 3365         1 Oldsmobile Cutlass
  cena   legenda
1 2400 America=1
2 1900  Europe=2
3 2200  Japan =3
4 2725
5 2250
6 3300
```

Zauważmy, iż kolumny `cena` oraz `legenda` nie zmieściły się na ekranie, więc zostały wyświetlone poniżej pozostałych.

Informacja

Format CSV jest dość często używany do przechowywania prostych zbiorów danych. Tego typu bazy można edytować np. za pomocą arkusza kalkulacyjnego. Przy odczycie/zapisie należy wybrać tylko odpowiedni format pliku, np. w programie Excel jest to *Plik tekstowy (CSV)*

Więcej interesujących opcji dostępnych w funkcji `read.csv2()` znajduje się oczywiście w systemie pomocy:

```
> ?read.csv
```

Analogiczną funkcją służącą do zapisu ramek danych do pliku jest `write.csv2()`.

W niniejszym zadaniu interesować nas będzie jedynie zmienna `mpg`. Jest ona typu ilościowego, jednakże poddamy ją – celem ćwiczenia – kategoryzacji.

```
> samochody$mpg
  [1] 43.1 36.1 32.8 39.4 36.1 19.9 19.4 20.2 19.2 20.5 20.2 25.1 20.5
 [14] 19.4 20.6 20.8 18.6 18.1 19.2 17.7 18.1 17.5 30.0 27.5 27.2 30.9
 [27] 21.1 23.2 23.8 23.9 20.3 17.0 21.6 16.2 31.5 29.5 21.5 19.8 22.3
 [40] 20.2 20.6 17.0 17.6 16.5 18.2 16.9 15.5 19.2 18.5 31.9 34.1 35.7
 [53] 27.4 25.4 23.0 27.2 23.9 34.2 34.5 31.8 37.3 28.4 28.8 26.8 33.5
 [66] 41.5 38.1 32.1 37.2 28.0 26.4 24.3 19.1 34.3 29.8 31.3 37.0 32.2
 [79] 46.6 27.9 40.8 44.3 43.4 36.4 30.4 44.6 40.9 33.8 29.8 32.7 23.7
 [92] 35.0 23.6 32.4 27.2 26.6 25.8 23.5 30.0 39.1 39.0 35.1 32.3 37.0
[105] 37.7 34.1 34.7 34.4 29.9 33.0 34.5 33.7 32.4 32.9 31.6 28.1   NA
[118] 30.7 25.4 24.2 22.4 26.6 20.2 17.6 28.0 27.0 34.0 31.0 29.0 27.0
[131] 24.0 23.0 36.0 37.0 31.0 38.0 36.0 36.0 36.0 34.0 38.0 32.0 38.0
[144] 25.0 38.0 26.0 22.0 32.0 36.0 27.0 27.0 44.0 32.0 28.0 31.0
> length(samochody$mpg)
[1] 155
```

Wśród 155 obserwacji stwierdzamy jeden brak danych – w miejscu oznaczonym symbolem „NA" (ang. *not available*). Problem brakujących danych jest poważnym wyzwaniem dla statystyków, zwłaszcza w przypadku analizy danych wielowymiarowych. W rozważanym przez nas przypadku warto usunąć ten element ciągu, gdyż obecność oznaczenia „NA" nie wnosi niczego istotnego, a nawet może skomplikować użycie niektórych funkcji statystycznych. Do usuwania wskazanego elementu w ciągu danych służy np. funkcja `na.omit()`.

Aby dokonać konwersji jednostek [mile/galon] na [litry/100 km], należy użyć następującego wzoru:

$$\texttt{zp} = \frac{1}{\texttt{mpg}} \frac{3.785 \cdot 100}{1.609}, \tag{2.1}$$

gdyż 1 mila = 1.609 km, a 1 galon = 3.785 l. Wynikowy ciąg zapiszemy jako wektor `zp`:

```
> # usuwamy braki danych:
> mpg <- na.omit(samochody$mpg)             # albo (lepiej):
> mpg <- as.vector(na.omit(samochody$mpg))  # albo:
> mpg <- samochody$mpg[!is.na(samochody$mpg)]
> # konwersja:
> zp <- 3.785*100/(mpg*1.609)
> print(zp, digits=3)                       # wypisanie
```

```
  [1]  5.46  6.52  7.17  5.97  6.52 11.82 12.13 11.65 12.25 11.48 11.65
 [12]  9.37 11.48 12.13 11.42 11.31 12.65 13.00 12.25 13.29 13.00 13.44
 [23]  7.84  8.55  8.65  7.61 11.15 10.14  9.88  9.84 11.59 13.84 10.89
 [34] 14.52  7.47  7.97 10.94 11.88 10.55 11.65 11.42 13.84 13.37 14.26
 [45] 12.93 13.92 15.18 12.25 12.72  7.37  6.90  6.59  8.59  9.26 10.23
 [56]  8.65  9.84  6.88  6.82  7.40  6.31  8.28  8.17  8.78  7.02  5.67
 [67]  6.17  7.33  6.32  8.40  8.91  9.68 12.32  6.86  7.89  7.52  6.36
 [78]  7.31  5.05  8.43  5.77  5.31  5.42  6.46  7.74  5.27  5.75  6.96
 [89]  7.89  7.19  9.93  6.72  9.97  7.26  8.65  8.84  9.12 10.01  7.84
[100]  6.02  6.03  6.70  7.28  6.36  6.24  6.90  6.78  6.84  7.87  7.13
[111]  6.82  6.98  7.26  7.15  7.44  8.37  7.66  9.26  9.72 10.50  8.84
[122] 11.65 13.37  8.40  8.71  6.92  7.59  8.11  8.71  9.80 10.23  6.53
[133]  6.36  7.59  6.19  6.53  6.53  6.53  6.92  6.19  7.35  6.19  9.41
[144]  6.19  9.05 10.69  7.35  6.53  8.71  8.71  5.35  7.35  8.40  7.59
```

W tej chwili jesteśmy już gotowi na przeprowadzenie kategoryzacji wg. wskazanego w zadaniu klucza. Wyniki umieścimy w wektorze `spalanie`. Będzie on miał, oczywiście, taką samą długość jak `zp`, a każdy jego element, tzn. `spalanie[i]`, będzie oznaczał klasę, do której wpada odpowiadający mu wyraz `zp[i]`, `i` $= 1, ..., 154$.

```
> spalanie <- rep(NA, length(zp))  # "pusty" wektor o żądanym rozmiarze
> spalanie[zp<7] <- "malo"
> spalanie[zp>=7 & zp<10] <- "srednio"
> spalanie[zp>=10] <- "duzo"
> spalanie <- factor(spalanie)  # konwersja na zmienną jakościową
> head(spalanie)
[1] malo    malo    srednio malo    malo    duzo
Levels: duzo malo srednio
```

Po wykonaniu powyższego polecenia otrzymamy zmienną jakościową, charakteryzującą w sposób opisowy zużycie paliwa badanych samochodów. Warto przypomnieć, że wyrażenie „`spalanie[zp<7] <- "malo"`" oznacza „weź te elementy wektora `spalanie`, które odpowiadają elementom wektora `zp`, o wartościach mniejszych niż 7 i przypisz im kategorię `"malo"`". Kolejny raz mamy więc okazję zaobserwować, jak bardzo zwięzłe, a zarazem jak pojemne znaczeniowo są konstrukcje języka R.

R ma także wbudowaną wygodną funkcję o nazwie `cut()`, służącą do kategoryzowania zmiennych ilościowych według podziału dla pewnych ustalonych wartości $b_1 < b_2 < \cdots < b_n$, przy czym możemy uzyskać przedziały domknięte prawostronnie $(b_1, b_2]$, $(b_2, b_3], \ldots, (b_{n-1}, b_n]$, ustalając parametr `right=TRUE`, bądź przedziały domknięte lewostronnie $[b_1, b_2), [b_2, b_3), \ldots, [b_{n-1}, b_n)$, wpisując parametr `right=FALSE`.

```
> cut(zp, c(-Inf, 7, 10, Inf), right=FALSE)
  [1] [-Inf,7)  [-Inf,7)  [7,10)    [-Inf,7)  [-Inf,7)  [10, Inf)
  [7] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [7,10)
 [13] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf)
 [19] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [7,10)    [7,10)
 [25] [7,10)    [7,10)    [10, Inf) [10, Inf) [7,10)    [7,10)
```

```
 [31] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [7,10)    [7,10)
 [37] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf)
 [43] [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf) [10, Inf)
 [49] [10, Inf) [7,10)    [-Inf,7)  [-Inf,7)  [7,10)    [7,10)
 [55] [10, Inf) [7,10)    [7,10)    [-Inf,7)  [-Inf,7)  [7,10)
 [61] [-Inf,7)  [7,10)    [7,10)    [7,10)    [7,10)    [-Inf,7)
 [67] [-Inf,7)  [7,10)    [-Inf,7)  [7,10)    [7,10)    [7,10)
 [73] [10, Inf) [-Inf,7)  [7,10)    [7,10)    [-Inf,7)  [7,10)
 [79] [-Inf,7)  [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)
 [85] [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [7,10)    [7,10)
 [91] [7,10)    [-Inf,7)  [7,10)    [7,10)    [7,10)    [7,10)
 [97] [7,10)    [10, Inf) [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)
[103] [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)
[109] [7,10)    [7,10)    [-Inf,7)  [-Inf,7)  [7,10)    [7,10)
[115] [7,10)    [7,10)    [7,10)    [7,10)    [7,10)    [10, Inf)
[121] [7,10)    [10, Inf) [10, Inf) [7,10)    [7,10)    [-Inf,7)
[127] [7,10)    [7,10)    [7,10)    [7,10)    [10, Inf) [-Inf,7)
[133] [-Inf,7)  [7,10)    [-Inf,7)  [-Inf,7)  [-Inf,7)  [-Inf,7)
[139] [-Inf,7)  [-Inf,7)  [7,10)    [-Inf,7)  [7,10)    [-Inf,7)
[145] [7,10)    [10, Inf) [7,10)    [-Inf,7)  [7,10)    [7,10)
[151] [-Inf,7)  [7,10)    [7,10)    [7,10)
Levels: [-Inf,7) [7,10) [10, Inf)
```

Uzyskanym klasom przydzielimy nazwy inne niż domyślne (zob. powyżej) i narysujemy wykres słupkowy (por. rys. 2.2).

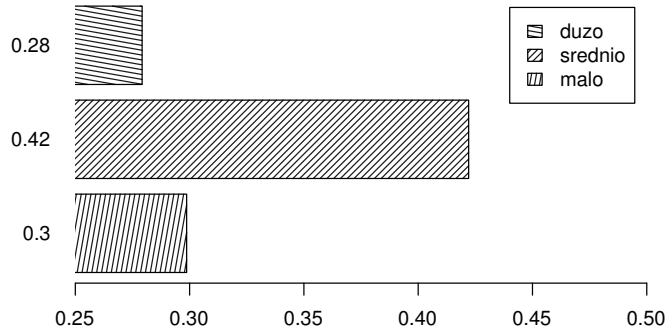
```
> spalanie <- cut(zp, c(-Inf, 7, 10, Inf), right=FALSE,
+    labels=c("malo", "srednio", "duzo"))
> spalTab <- table(spalanie)
> print(spalTab)
spalanie
   malo srednio    duzo
     46      65      43
> barplot(prop.table(spalTab), names=round(prop.table(spalTab), 2),
+    horiz=TRUE, legend=names(spalTab), xlim=c(0.25, 0.5), xpd=FALSE,
+    col="gray10", density=25, angle=c(80,45,-10), las=1)
```

⊡

2.2.2. Dane typu ilościowego

Zadanie 2.3. Przeprowadź wstępną analizę statystyczną zużycia paliwa (w l/100 km) samochodów opisanych w bazie `samochody.csv`.

Rozwiązanie. Tym razem zanalizujemy wektor `zp` (zob. poprzednie zadanie) jako zmienną typu ilościowego. Przypomnijmy, że przechowuje on informacje o zużyciu paliwa (mierzone w litrach na 100 km) samochodów z interesującej nas historycznej bazy



Rys. 2.2. Zmodyfikowany wykres słupkowy

danych.

```
> samochody <-
+    read.csv2("http://www.ibspan.waw.pl/~pgrzeg/stat_lab/samochody.csv")
> zp <- 3.785*100/(samochody$mpg[!is.na(samochody$mpg)]*1.609)
```

Zacznijmy od wyznaczenia charakterystyk liczbowych naszej próby. Zakładamy, że Czytelnik zna definicje i właściwą interpretację podanych poniższych statystyk próbkowych. Gdyby tak nie było, odsyłamy do podręczników ze statystyki, np. [7, 9].

Charakterystyki położenia:

```
> mean(zp)      # średnia arytmetyczna
[1] 8.766693
> median(zp)    # mediana
[1] 8.139865
> min(zp)       # minimum
[1] 5.048053
> max(zp)       # maksimum
[1] 15.17673
> range(zp)     # min. i max. jako jeden wektor
[1]  5.048053 15.176728
> quantile(zp, c(0.1, 0.25, 0.5, 0.75, 0.9)) # kwantyle różnych rzędów
      10%       25%       50%       75%       90%
 6.190507  6.863301  8.139865 10.433264 12.296949
> summary(zp)  # wygodna funkcja, wiele statystyk
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.048   6.863   8.140   8.767  10.430  15.180
```

```
> mean(zp, trim=0.1) # średnia ucięta (po 10% obserwacji z każdej strony)
[1] 8.557733
```

Charakterystyki rozproszenia:

```
> var(zp)          # wariancja
[1] 5.895066
> sd(zp)           # odchylenie standardowe
[1] 2.427976
> IQR(zp)          # rozstęp międzykwartylowy
[1] 3.569962
> diff(range(zp)) # rozstęp
[1] 10.12867
> sd(zp)/mean(zp) # współczynnik zmienności
[1] 0.2769546
```

Charakterystyki kształtu rozkładu:

```
> library("e1071") # musimy załadować dodatkową bibliotekę,
>                  # w niej bowiem znajdują poniższe funkcje:
> skewness(zp)     # współczynnik skośności
[1] 0.6801541
> kurtosis(zp)     # kurtoza
[1] -0.5847272
```

Zachęcamy Czytelników do samodzielnej interpretacji uzyskanych wartości statystyk próbkowych.

Informacja

Jeżeli pakiet e1071 nie został wcześniej zainstalowany, należy to zrobić wydając polecenie:

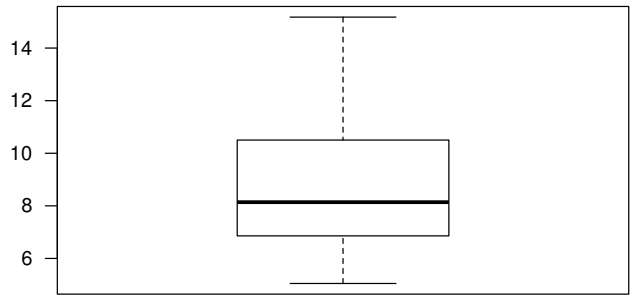
```
> install.packages("e1071")
```

Wykres skrzynkowy (ramkowy, pudełkowy od ang. *box plot* lub „pudełko z wąsami" od ang. *box-and-whisker plot*) jest wygodną metodą graficznej reprezentacji podstawowych statystyk z próby (kwartyli, minimum, maksimum) oraz identyfikacji obserwacji odstających. Domyślnie w programie R obserwacje odstające (ang. *outliers*) definiuje się jako obserwacje mniejsze niż $Q_1 - 1.5\,\mathrm{IQR}$ bądź większe niż $Q_3 + 1.5\,\mathrm{IQR}$, gdzie Q_1 oznacza wartość pierwszego kwartyla z danej próby, Q_3 – trzeciego, zaś $\mathrm{IQR} = Q_3 - Q_1$.

```
> boxplot(zp, las=1)
```

Wynik zamieszczamy na rys. 2.3.

Możliwe są niewielkie modyfikacje postaci uzyskanego wykresu, np. narysowanie go w układzie poziomym. Dodajmy także oznaczenie średniej z próby i odcinka średnia

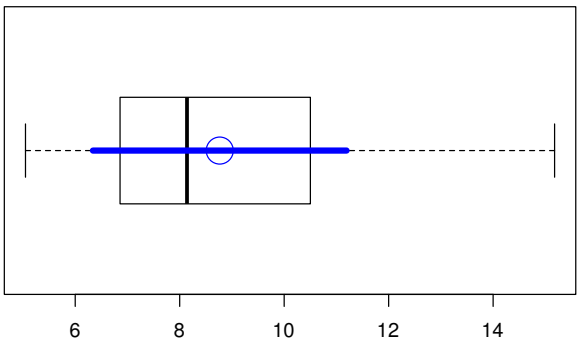


Rys. 2.3. Wykres skrzynkowy

$\pm$ jedno odchylenie standardowe.

```
> boxplot(zp, horizontal=TRUE)
> points(mean(zp), 1, cex=3, col="blue")
> lines(c(mean(zp)-sd(zp), mean(zp)+sd(zp)), c(1,1), col="blue", lwd=5)
```

Wynik zamieszczamy na rys. 2.4.



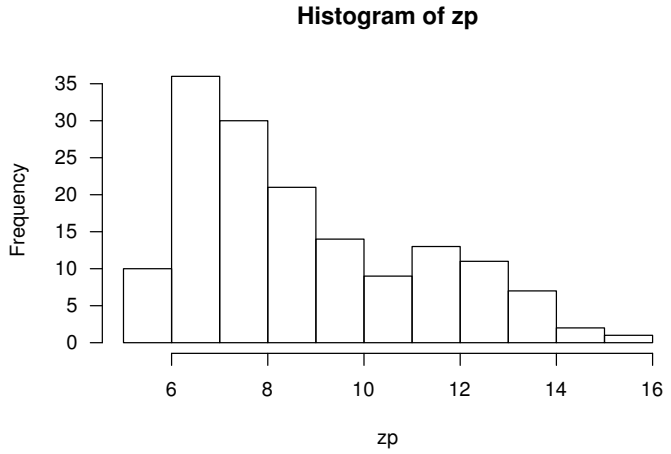
Rys. 2.4. Zmodyfikowany wykres skrzynkowy

Z powyższego rysunku możemy m.in. odczytać prawostronnie skośny charakter rozkładu danych. Świadczy o tym m.in. to, iż obserwacje leżące powyżej mediany rozpro-

szone są na znacznie dłuższym odcinku niż te, które leżą poniżej mediany. Na dodatnią skośność rozkładu wskazuje również kształt „pudełka" utworzonego przez kwartyle. Zauważmy również, że omawiany wykres nie pozwala na stwierdzenie, że w rozważanym zbiorze danych brak obserwacji odstających.

Inną, bardzo często stosowaną graficzną formą prezentacji rozkładu empirycznego jest *histogram*. Zaobserwowane wartości badanej zmiennej grupujemy w rozłączne klasy, tzn. dzielimy prostą na rozłączne przedziały (przeważnie o tej samej długości), po czym zliczamy obserwacje wpadające do każdego przedziału, a następnie tworzymy wykres podobny do słupkowego, por. rys. 2.5.

```
> hist(zp, las=1)
```



Rys. 2.5. Histogram

Na powyższym rysunku dostrzegamy 11 klas, między które zostały rozdzielone wartości badanej zmiennej `zp`. R dokonuje automatycznego doboru liczby klas zgodnie z tzw. regułą Sturgesa. Możemy też wyznaczyć liczbę klas posługując się inną regułą, nadając odpowiednią wartość parametrowi `breaks=`*Identyfikator*, przy czym:

Identyfikator	Nazwa	Wyrażenie
`"Sturges"`	Reguła Sturgesa	$k = \lceil \log_2 n + 1 \rceil$ $(n \geqslant 30)$,
`"Scott"`	Wzór Scotta dla rozkładu normalnego	$h = 3.5s/n^{1/3}$,
`"FD"`	Wzór Freedmana-Diaconisa	$h = 2\,\mathrm{IQR}/n^{1/3}$,

gdzie k – liczba klas, h – szerokość przedziału, n – liczba obserwacji, s – odchylenie standardowe z próby, IQR – rozstęp międzykwartylowy. Przykładowo, wywołanie:

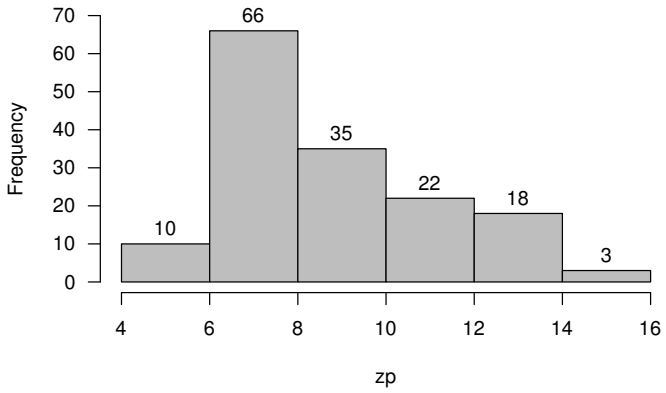
```
> hist(zp, breaks="Scott")
```

daje w wyniku liczbę klas równą 6.

Możemy także *zasugerować* pożądaną liczbę klas, mimo iż R automatycznie optymalizuje ich liczbę.

Warto dodatkowo w tym miejscu zanotować, iż histogram można zapisać jako obiekt, co daje sposobność odczytania jego parametrów.

```
> h <- hist(zp, breaks=5, # wygeneruj histogram z sugerowaną liczbą klas=5
+               labels=TRUE, # dodaj etykietki nad słupkami
+               col="gray", main=NA, ylim=c(0, 70), las=1)
> print(h) # wyświetl definicje tego histogramu
$breaks
[1]  4  6  8 10 12 14 16

$counts
[1] 10 66 35 22 18  3

$density
[1] 0.03246753 0.21428571 0.11363636 0.07142857 0.05844156 0.00974026

$mids
[1]  5  7  9 11 13 15

$xname
[1] "zp"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```



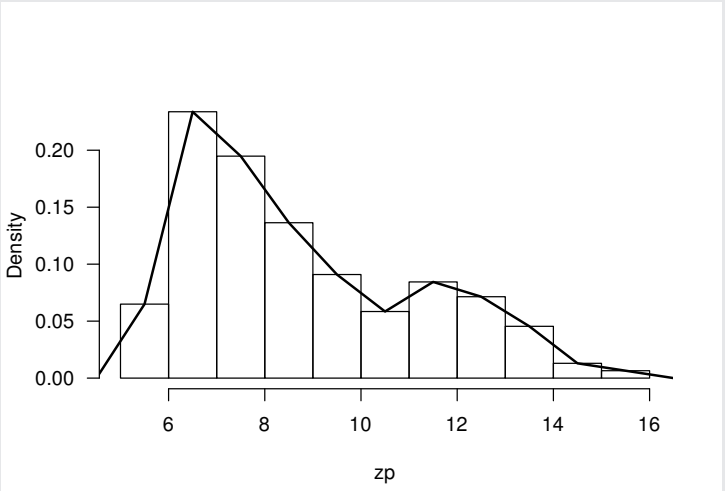
Jak łatwo zauważyć, liczba klas determinuje kształt histogramu. Nie ma w tym wy-

padku jednej, „złotej reguły" dla wszystkich możliwych prób. Właściwą liczbę klas należy dobierać w każdym przypadku indywidualnie, sprawdzając różne możliwości i wybierając ostateczną liczbę klas kierując się walorami estetycznymi wykresu (np. jego regularnością) oraz niesioną przez ten wykres informacją.

Informacja

Możemy pokusić się o narysowanie histogramu częstości (parametr `prob=TRUE`, prowadzący do rysunku, na którym pole pod wykresem wynosi 1) wraz z naniesioną nań łamaną częstości. Będziemy jednak musieli ją skonstruować ręcznie, jak następuje:

```
> h <- hist(zp, prob=TRUE, main=NA, las=1)
> szerPrzedzialu <- h$breaks[2]-h$breaks[1] # podział na równe części
> ileKlas <- length(h$mids) # mids - środki przedziałów
> lamanaX <- c(h$mids[1]-szerPrzedzialu, h$mids,
+    h$mids[ileKlas]+szerPrzedzialu)
> lamanaY <- c(0, h$density, 0)
> lines(lamanaX, lamanaY, lwd=2) # dodaj linie
```

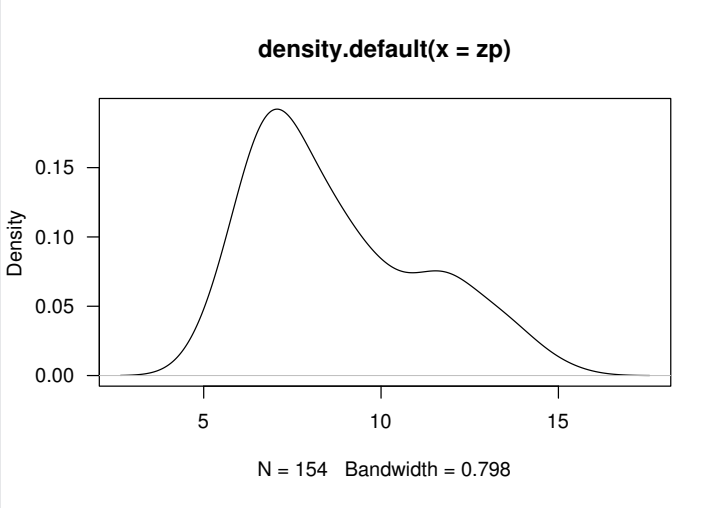


Nietrudno wykazać, że pole pod tak skonstruowaną łamaną częstości także wynosi 1, zatem jest ona pewnym estymatorem gęstości rozkładu.

Informacja

Istnieją także inne sposoby estymowania gęstości na podstawie próby losowej, np. estymatory jądrowe.

```
> plot(density(zp), las=1)
```



Podobną rolę do histogramu spełnia *wykres (diagram) łodygowo-liściowy* (ang. *stem-and-leaf display*). Był on bardziej popularny w czasach, gdy komputery nie miały takich możliwości generowania grafiki, jakie mają dziś (wyświetlany jest w trybie tekstowym). Łatwo się go także rysuje odręcznie.

```
> stem(zp)

  The decimal point is at the |

   5 | 033345788
   6 | 000222222334445555555567788889999999
   7 | 000122233333444444556666778899 9
   8 | 0123444446666677778889
   9 | 0133447788899
  10 | 0012255799
  11 | 1344556666689
  12 | 113333679
  13 | 003444889
  14 | 35
  15 | 2
```

Dzięki niemu można wyrobić sobie nie tylko intuicję odnośnie kształtu funkcji gęstości rozkładu, ale i odczytać przybliżone wartości wszystkich obserwacji, np., kolejno, 5.0, 5,3, 5,3, 5,3, 5,4 itd.

Spójrzmy jak będzie wyglądał diagram tego typu po zmianie parametru skali:

```
> stem(zp, scale=2)

  The decimal point is at the |

   5 | 03334
   5 | 5788
   6 | 00022222233444
   6 | 555555556778888999999
   7 | 000122233333444444
   7 | 5566667788999
   8 | 012344444
   8 | 6666677778889
   9 | 013344
   9 | 7788899
  10 | 00122
  10 | 55799
  11 | 1344
  11 | 556666689
  12 | 113333
  12 | 679
  13 | 003444
  13 | 889
  14 | 3
  14 | 5
  15 | 2
```

⊡

Zadanie 2.4. Korzystając z bazy danych `samochody.csv` przeprowadź wstępną analizę statystyczną zużycia paliwa (w l/100 km) oddzielnie dla samochodów produkowanych w Ameryce, Europie i Japonii. Rozważ tylko auta tańsze niż 10 000$.

Rozwiązanie. Stwórzmy od nowa wektor zp, tym razem nie usuwając z niego braków danych. Będziemy bowiem chcieli podzielić go na rozłączne części, z których każda będzie przechowywać informacje o zuzyciu paliwa przez samochody produkowane w jednym z trzech rozważanych regionów świata.

```
> samochody <-
+    read.csv2("http://www.ibspan.waw.pl/~pgrzeg/stat_lab/samochody.csv")
> zp <- 3.785*100/(samochody$mpg*1.609)
> samochody$producent
  [1] 2 1 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 1 3 1 1 3 2 2 2
 [34] 2 2 3 1 1 1 1 1 1 1 1 1 1 1 1 2 3 1 1 2 1 2 1 1 1 3 2 1 1 1 1 2
 [67] 3 1 3 1 1 1 1 2 3 3 3 3 3 1 3 2 2 2 2 3 2 3 2 3 3 2 1 3 1 1 1 1 1
[100] 3 1 3 3 3 3 3 1 1 1 2 2 3 3 3 3 2 2 2 3 3 1 1 1 1 1 1 1 1 1 1 1 1
[133] 2 3 3 1 1 3 3 3 3 3 3 1 1 1 1 3 1 1 1 2 1 1 1
```

Wartości zmiennej `producent`, mimo iż są cyframi, pełnią de facto funkcję symboli, za pomocą których zakodowano w bazie danych rejon świata, z którego pochodzą poszczególne samochody. Wyjaśnienie znaczenia tych kodów zamieszczono w kolumnie `legenda`, dodanej sztucznie do bazy danych.

```
> head(samochody$legenda)
[1] America=1 Europe=2  Japan =3
Levels:          America=1 Europe=2 Japan =3
```

Zmienną `producent` możemy przekształcić na typ jakościowy (w tym momencie R traktuje ją jak zwykły wektor liczb całkowitych) i przypisać kategoriom liczbowym wartości opisowe:

```
> prod <- factor(samochody$producent)
> levels(prod) <- c("Ameryka", "Europa", "Japonia")
> head(prod)
[1] Europa  Ameryka Japonia Japonia Japonia Ameryka
Levels: Ameryka Europa Japonia
> table(prod)
prod
Ameryka  Europa Japonia
     85      26      44
```

Stwórzmy teraz trzy nowe wektory, każdy odpowiadający zużyciu paliwa aut kosztujących mniej niż 10000$, pochodzących z różnych regionów:

```
> zpA <- zp[prod == "Ameryka" & samochody$cena < 10000]
> zpE <- zp[prod == "Europa"  & samochody$cena < 10000]
> zpJ <- zp[prod == "Japonia" & samochody$cena < 10000]
```

Zwróćmy uwagę, że przywrócenie braków danych w wektorze `zp` było konieczne do podziału wartości na podzbiory, bowiem wektory `zp`, `producent` i `cena` muszą mieć tę samą długość. Pozostaje jeszcze ich powrotne usunięcie (teraz, nie wcześniej) i sprawdzenie, czy zbiory wynikowe reprezentują to, o co zostaliśmy poproszeni.

```
> zpA <- zpA[!is.na(zpA)] # usuwamy braki danych - już nie są potrzebne
> zpE <- zpE[!is.na(zpE)]
> zpJ <- zpJ[!is.na(zpJ)]
> length(zpA) + length(zpE) + length(zpJ) # ile łącznie obserwacji?
[1] 152
> sum(samochody$cena < 10000 & !is.na(zp)) # czy tyle samo?
[1] 152
```

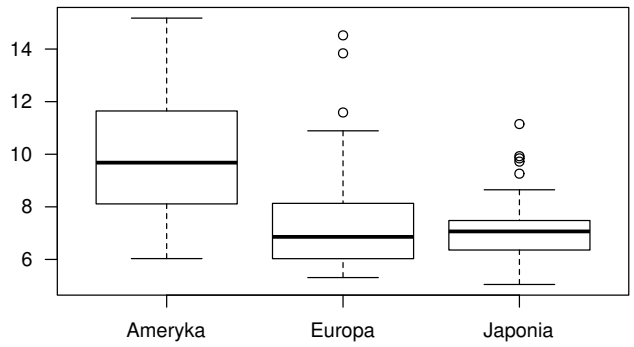
By uzyskać dostęp do podstawowych statystyk próbkowych, należy wywołać stosowne funkcje oddzielnie dla każdego wektora, na przykład:

```
> summary(zpA)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.032   8.112   9.681   9.854  11.650  15.180
> summary(zpE)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.310   6.029   6.858   7.741   8.133  14.520
> summary(zpJ)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.048   6.358   7.065   7.215   7.462  11.150
```

Interesujące też będzie zestawienie i porównanie rozkładów zużycia paliwa za pomocą narzędzi graficznych. Szczególnie przydatne bywa w tym celu umieszczenie na jednym rysunku kilku wykresów skrzynkowych, odpowiadających poszczególnym podzbiorom danych (por. rys. 2.6):

```
> boxplot(zpA, zpE, zpJ, names=c("Ameryka", "Europa", "Japonia"), las=1)
```



Rys. 2.6. Różne wykresy skrzynkowe

I tak rzut oka na wykres wystarcza by stwierdzić, że przeciętne zużycie paliwa samochodów amerykańskich jest większe niż aut wytwarzanych w innych częściach świata (co zresztą jest zgodne z obiegową opinią). Jednocześnie wyniki w tej grupie charakteryzują się bardzo dużą zmiennością. Z kolei samochody produkowane w Europie i w Japonii charakteryzują się przeciętnie podobnym zużyciem paliwa, jednakże w przypadku aut japońskich mamy do czynienia z mniejszym zróżnicowaniem danych, niż w przypadku pojazdów europejskich. Co więcej, w próbkach samochodów pochodzących z Europy i z Japonii stwierdzamy istnienie obserwacji odstających (oznaczonych kółkami powyżej „wąsów" wykresu).

Informacja

W przypadku wykresu skrzynkowego istnieje inny, wygodniejszy sposób analizy

zmiennej podzielonej na podzbiory. Możemy w tym celu posłużyć się następującym kodem (należy jednak pamiętać, by wcześniej wykluczyć nieinteresujące nas informacje, co w rozważanym zadaniu oznacza auta nie tańsze niż 10 000$):

```
> zp2   <- zp[samochody$cena < 10000]
> prod2 <- prod[samochody$cena < 10000]
> boxplot(zp2 ~ prod2) # tzn. podziel zp2 według kategorii z prod2
```

Funkcja rysująca histogram nie daje nam, niestety, możliwości automatycznego zilustrowania kilku zbiorów w ramach jednego wykresu. Chcąc zatem porównać kilka histogramów możemy posłużyć się wywołaniem `par(mfrow=c(4,1))`, której spowoduje utworzenie obrazka składającego z czterech podwykresów zamieszczonych jednego pod drugim (oprócz trzech histogramów zużycia paliwa przez samochody amerykańskie, europejskie i japońskie, dodaliśmy czwarty histogram utworzony dla wszystkich samochodów). Pamiętajmy, aby w takim przypadku zadbać o to, aby widoczny zakres na osi x był taki sam na każdym podwykresie.

```
> par(mfrow=c(4,1))   # 4 w 1
> zakres <- range(zp) # od min. do maks. obserwacji w całym zbiorze
> zakres
[1] NA NA
> zakres <- range(na.omit(zp))
> zakres
[1]  5.048053 15.176728
> # 4 histogramy:
> hist(zp,  ylab="Licznosc", xlab="Wszystkie samochody",    xlim=zakres,
+    main="Zuzycie paliwa", las=1)
> hist(zpA, ylab="Licznosc", xlab="Samochody amerykanskie", xlim=zakres,
+    main=NA, las=1)
> hist(zpE, ylab="Licznosc", xlab="Samochody europejskie",  xlim=zakres,
+    main=NA, las=1)
> hist(zpJ, ylab="Licznosc", xlab="Samochody japonskie",    xlim=zakres,
+    main=NA, las=1)
```

Wynik zamieszczamy na rys. 2.7.

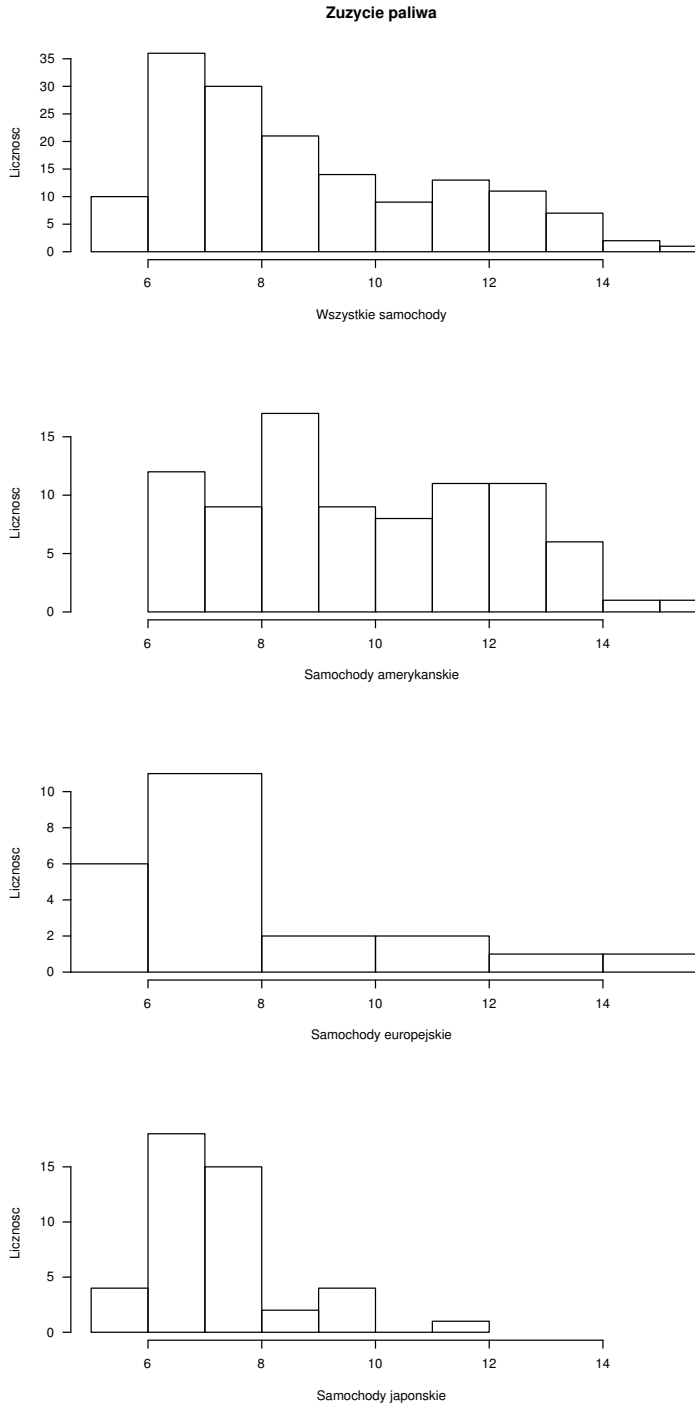
⊡

2.2.3. Szeregi czasowe

Zadanie 2.5. Poniższe dane odpowiadają notowaniom giełdowym pewnej spółki (w PLN) w kolejnych 20 dniach:

23.30, 24.50, 25.30, 25.30, 24.30, 24.80, 25.20, 24.50, 24.60, 24.10,
24.30, 26.10, 23.10, 25.50, 22.60, 24.60, 24.30, 25.40, 25.20, 26.80.

Utwórz wykres cen akcji w funkcji czasu (czyli tzw. *szereg czasowy*).



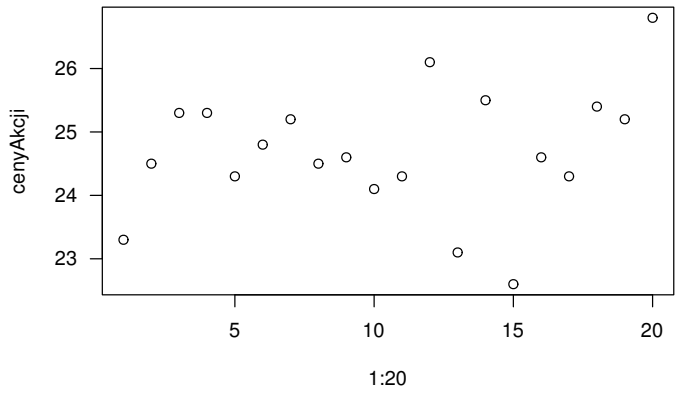
Rys. 2.7. Histogramy

Rozwiązanie. Najpierw należy wprowadzić dane do naszego ulubionego programu:

```
> cenyAkcji <- c(23.30, 24.50, 25.30, 25.30, 24.30, 24.80, 25.20, 24.50,
+ 24.60, 24.10, 24.30, 26.10, 23.10, 25.50, 22.60, 24.60, 24.30, 25.40,
+ 25.20, 26.80)
```

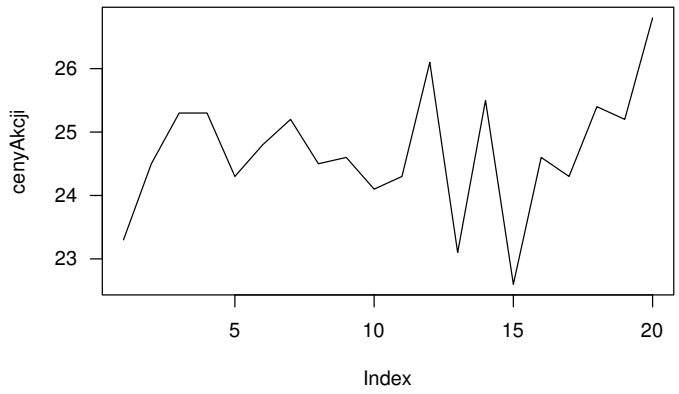
Utworzenie wykresu dokonuje się poprzez wywołanie funkcji `plot`.

```
> plot(cenyAkcji, las=1) # co jest równoważne:
> plot(1:20, cenyAkcji, las=1)
```



Niestety z uzyskanego wykresu niewiele da się wyczytać. Dużo lepszy efekt uzyskamy łącząc punkty łamaną:

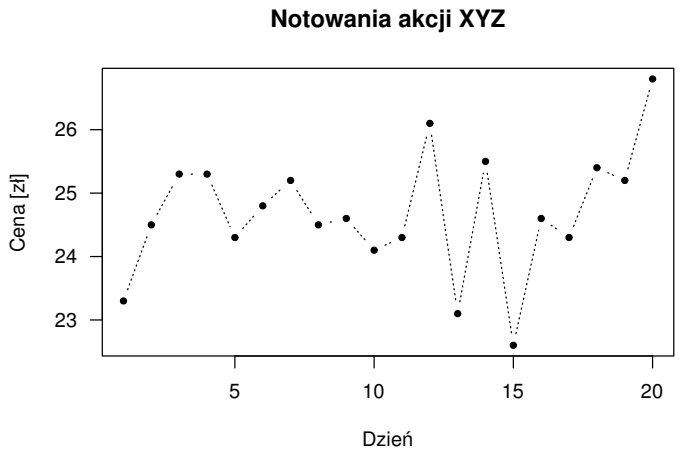
```
> plot(cenyAkcji, type="l", las=1)
```



Zastosowanie funkcji kawałkami liniowej sugeruje ciągły (a w dodatku liniowy) przyrost cen między kolejnymi punktami na osi czasu. Aby podkreślić dyskretny charak-

ter próbkowania można użyć parametr `type="b"` (od *both*), dzięki czemu na wykresie pojawią się zarówno punkty, jak i linie.

```
> plot(cenyAkcji, type="b", pch=20, lty=3, las=1,
+    main="Notowania akcji XYZ", xlab="Dzień", ylab="Cena [zł]")
```



Informacja

Zwróćmy uwagę na argumenty `pch` i `lty` funkcji `plot()`. Mają one za zadanie zmienić sposób rysowania symboli punktów i typów linii, por. `?plot.default`.

Informacja

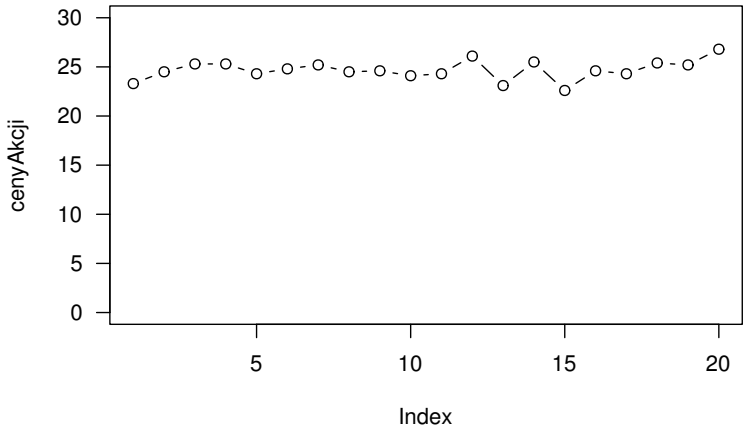
Warto także przekonać się, jak ustawienie zakresu danych na osi y oraz proporcje rozmiarów wykresu, wpływają na percepcję zmienności cen (por. rys. 2.8):

```
> plot(cenyAkcji, ylim=c(0,30), type="b", las=1)
```

⊡

2.3. Zadania do samodzielnego rozwiązania

Zadanie 2.6. Według danych GUS, w 2006 roku liczba urodzeń w Polsce wynosiła 374 244. Poniższa tabela zestawia te dane w zależności od wieku matki.



Rys. 2.8. Zmiana percepcji danych w zależności od zakresu danych na osi y

Wiek matki	Liczba urodzeń
19 lat i mniej	19 230
20–24	93 569
25–29	139 853
30–34	86 825
35–39	28 487
40–44	5 975
45 lat i więcej	305

1. Wprowadź dane do programu R.
2. Utwórz tabelę liczności i częstości liczby urodzeń w zależności od wieku matki.
3. Narysuj wykres kołowy.
4. Narysuj wykres słupkowy.
5. Zinterpretuj uzyskane wyniki.

Zadanie 2.7. Koncern paliwowy planuje otworzyć nową stację benzynową w pewnym mieście. Rozważane są cztery możliwe lokalizacje stacji: w południowej (`S`), północnej (`N`), zachodniej (`W`) i wschodniej (`E`) dzielnicy miasta. W ramach badania opinii społecznej odnośnie preferowanej lokalizacji stacji zapytano o to tysiąc kierowców. Ich odpowiedzi znajdują się w pliku `stacje.csv`[2]. Utwórz wykres słupkowy i wykres kołowy dla badanych preferencji.

[2] `http://www.ibspan.waw.pl/~pgrzeg/stat_lab/stacje.csv`

Zadanie 2.8. Badania demograficzne przeprowadzone w 1988 roku w USA wykazały, że wśród kobiet mających 18 i więcej lat było: 17364 tyś. panien, 56128 tyś. mężatek, 11239 tyś. wdów i 8170 tyś. rozwódek.

1. Utwórz wykres kołowy dla stanu cywilnego danej grupy kobiet. Porównaj różne formy opisu wykresu.
2. Utwórz wykres słupkowy dla stanu cywilnego danej grupy kobiet. Porównaj różne rodzaje wykresów i formy ich opisu.

⋆ **Zadanie 2.9.** Uważa się, że oko ludzkie dobrze rozpoznaje różnice stosunków długości, lecz nie najlepiej radzi sobie ze stosunkami pól. Dlatego, w przypadku danych typu jakościowego, odradza się używania wykresu kołowego, na korzyść np. wykresu słupkowego.

1. Podaj przykład danych, dla których trudno jest ocenić, które kategorie mogą być reprezentowane liczniej od innych.
2. W dowolnym czasopiśmie poruszającym tematykę życia społeczno-politycznego (np. *Polityka, Newsweek*), znajdź przykłady wykresów dla danych typu jakościowego. Których jest najwięcej?

⋆ **Zadanie 2.10.** Zbadaj, jak kształtowała się liczba małżeństw zawartych w ostatnim roku, w zależnosci od miesiąca, w którym miał miejsce ślub. Skorzystaj z aktualnej Bazy Demograficznej publikowanej przez Główny Urząd Statystyczny[3]. Jak wyjaśnisz uzyskane wyniki?

Zadanie 2.11. Wytrzymałość na ciśnienie wewnętrzne szkła butelek jest ich ważną charakterystyką jakościową. W celu zbadania wytrzymałości butelek umieszcza się je w maszynie hydrostatycznej, po czym zwiększa się ciśnienie aż do zniszczenia butelki. Plik `butelki.csv` zawiera dane opisujące graniczną wytrzymałość na ciśnienie wewnętrzne szkła badanej partii butelek mierzone w psi (tzn. funtach na cal kwadratowy).

1. Utwórz zmienną o nazwie `cisnienie`, opisującą wytrzymałość na ciśnienie wewnętrzne szkła butelek mierzone w MPa (wskazówka: 1 psi = 6 894.757 Pa).
2. Utwórz histogram dla danych opisujących wytrzymałość butelek. Prześledź wpływ liczby klas na kształt histogramu. Porównaj różne rodzaje histogramów.
3. Utwórz wykres łamanej liczności i nałóż go na wykres histogramu.
4. Utwórz wykres łodygowo-liściowy.
5. Utwórz i zinterpretuj wykres skrzynkowy dla wytrzymałości butelek.
6. Wyznacz i zinterpretuj podstawowe statystyki próbkowe dla danych opisujących wytrzymałość butelek.
7. Oblicz i zinterpretuj 5, 10, 25, 50, 75, 90 i 95 percentyl dla rozważanych danych.
8. Wyznacz 10% średnią uciętą dla danych opisujących wytrzymałość butelek. Porównaj średnią uciętą ze średnią arytmetyczną i medianą. Prześledź, jak zmienia się wartość średniej wraz ze zmianą stopnia ucięcia próbki.

[3] `http://www.stat.gov.pl`

Zadanie 2.12. Zamieszczone poniżej dane przedstawiają wysokość czynszu płaconego w pewnej spółdzielni mieszkaniowej przez 30 losowo wybranych lokatorów.

334, 436, 425, 398, 424, 429, 392, 428, 339, 389
352, 405, 392, 403, 344, 400, 424, 443, 378, 387
384, 498, 374, 389, 367, 457, 409, 454, 345, 422.

Przeprowadź wstępną analizę statystyczną powyższych danych.

Zadanie 2.13. Przeprowadź wstępną analizę statystyczną danych dotyczących przyspieszenia (zmienna `przysp`) pojazdów z bazy `samochody.csv`, ważących mniej niż 2500 funtów (zmienna `waga`).

Zadanie 2.14. Przeprowadź wstępną analizę statystyczną danych dotyczących przyspieszenia (zmienna `przysp`) pojazdów z bazy `samochody.csv`, oddzielnie dla aut z Ameryki, Europy i Japonii.

Zadanie 2.15. Porównaj dane dotyczące mocy (zmienna `moc`) samochodów posiadających różną liczbę cylindrów (zmienna `cylindry`). Wykorzystaj informacje zawarte w bazie `samochody.csv`.

Zadanie 2.16. Pani Janina bardzo się nudzi od chwili gdy jej pociechy założyły własne rodziny. Całe dnie spędza siedząc na ławce i obserwując życie swojej małej wioski. Jednym z najbardziej fascynujących momentów jej dziennego harmonogramu robót i robótek są wizyty listonosza, pana Sławomira. Pani Janina dowiedziała się od naczelnika poczty, że listonosz powinien pojawiać się w pobliżu jej domu około godziny 10:25. Niestety, jak stwierdził naczelnik, różnego rodzaju okoliczności zewnętrzne wpływają na fluktuacje czasu przyjazdu. Pani Janina postanowiła zbadać frapujący ją problem nie do końca punktualnego listonosza. Zanotowała czasy przyjazdów (w minutach po godz. 10-tej) w kolejnych 33 dniach roboczych. Niestety 3 spośród zapisanych wartości okazały się są nieczytelne z powodu pisania nienaostrzonym ołówkiem.

26, 22, 26, 20, 25, ??, 21, 20, 28, 27, 26,
38, 23, 30, 21, 25, 26, 23, 25, 27, 27, ??,
25, 22, 23, 31, 19, ??, 25, 25, 23, 25, 24.

W miejscowości, w której mieszka pani Janina, już od dawna krążą plotki o wyższości R-a nad innymi programami wspomagającymi analizę danych. Stąd prośba pani Janiny skierowana do Ciebie, wielce pilny studencie, o pomoc w przeprowadzeniu wstępnej analizy załączonego zbioru danych.

Zadanie 2.17. Z danych z poprzedniego zadania usuń obserwacje odstające i braki danych. Następnie przyporządkuj każdej obserwacji jedną z pięciu kategorii:

Kategoria	Czas przyjazdu
ZaWcz	$(-\infty, 23)$,
Wcz	$[23, 25)$,
Punkt	$= 25$,
Pźn	$(25, 27]$,
ZaPźn	$(27, \infty)$.

Opisz wynikowy zbiór za pomocą znanych Ci metod.

⋆ **Zadanie 2.18.** Kilka wykresów, zamieszczonych w poprzednim punkcie, zostało tak narysowanych, aby wywołać u odbiorcy różnego rodzaju wrażenia (np. zmniejszenia zmienności danych). Znajdź inne przykłady manipulacji parametrami wykresów, spotykane w życiu codziennym, np. w czasopismach bądź w materiałach reklamowych.

⋆ **Zadanie 2.19.** Napisz funkcję, która dla danej realizacji $\mathbf{x} = (x_1, \ldots, x_n)$ próby prostej wyznacza wartość nieobciążonego estymatora kurtozy $\kappa_{\mathbf{x}}$:

$$\kappa_{\mathbf{x}} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{n} \left(\frac{x_i - \overline{\mathbf{x}}}{s_{\mathbf{x}}} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}, \tag{2.2}$$

gdzie, $\overline{\mathbf{x}}$ – średnia z próby, $s_{\mathbf{x}}$ – odchylenie standardowe.

⋆ **Zadanie 2.20.** Napisz funkcję wyznaczającą dla danej realizacji próby wartość średniej winsoryzowanej dowolnego rzędu.

2.4. Wskazówki i odpowiedzi do zadań

Ad zad. 2.16. Rozważ nie tylko metody analizy danych jakościowych, wymienione we Wprowadzeniu, ale także potraktuj wyniki pomiarów jako pewien szereg czasowy.

Ad zad. 2.19. Składnia funkcji tworzonych w R przedstawia się następująco:

```
> nazwaFunkcji <- function(argument1, argument2, (...))

   (...) różne operacje (...)
   return(wynik)
```

Na przykład, funkcja licząca średnią arytmetyczną może być utworzona za pomocą operacji:

```
> srednia <- function(probka)
+ {
+    suma <- sum(probka)
+    licznosc <- length(probka)
+    suma/licznosc # ostatnie wyrażenie = wartość zwracana
+ }
```

Sprawdźmy:

```
> X <- c(1,2,3,4,5,6)
> srednia(X)
[1] 3.5
```

Rozkłady prawdopodobieństwa i podstawy symulacji

3

3.1. Wprowadzenie

3.1.1. Wybrane rozkłady prawdopodobieństwa

3.1.1.1. Oznaczenia i dostępne funkcje rozkładów

Środowisko R umożliwia wyznaczanie wartości dystrybuanty i funkcji kwantylowej oraz funkcji prawdopodobieństwa lub gęstości podstawowych rozkładów dyskretnych i ciągłych. Ponadto korzystając z R, możemy generować liczby pseudolosowe z wielu rozkładów. W poniższej tabeli podano rozkłady prawdopodobieństwa, dostępne w R, wraz z przyjętymi identyfikatorami poszczególnych rozkładów i charakteryzującymi je parametrami:

Rozkład	Oznaczenie	Parametry	Identyfikator
dwumianowy	$\mathrm{Bin}(n,p)$	$n \in \mathbb{N}, p \in (0,1)$	`*binom`
geometryczny	$\mathrm{Geom}(p)$	$p \in (0,1)$	`*geom`
hipergeometryczny	$\mathrm{Hyp}(N,n,k)$	$N,n,k \in \mathbb{N}$	`*hyper`
ujemny dwumianowy	$\mathrm{NB}(n,p)$	$n \in \mathbb{N}, p \in (0,1)$	`*nbinom`
Poissona	$\mathrm{Pois}(\lambda)$	$\lambda > 0$	`*pois`
beta	$\mathrm{Beta}(a,b)$	$a > 0, b > 0$	`*beta`
Cauchy'ego	$\mathrm{C}(l=0,s=1)$	$l \in \mathbb{R}, s > 0$	`*cauchy`
chi-kwadrat	${\chi^2}_d$	$d \in \mathbb{N}$	`*chisq`
wykładniczy	$\mathrm{Exp}(\lambda=1)$	$\lambda > 0$	`*exp`
F-Snedecora	$\mathrm{F}^{[d_1,d_2]}$	$d_1, d_2 \in \mathbb{N}$	`*f`
gamma	$\Gamma(a,s)$	$a > 0, s > 0$	`*gamma`
logistyczny	$\mathrm{Logist}(\mu=0,s=1)$	$\mu \in \mathbb{R}, s > 0$	`*logis`
logarytmiczno-normalny	$\mathrm{LN}(\mu=0,\sigma=1)$	$\mu \in \mathbb{R}, \sigma > 0$	`*lnorm`
normalny	$\mathrm{N}(\mu=0,\sigma=1)$	$\mu \in \mathbb{R}, \sigma > 0$	`*norm`
jednostajny	$\mathrm{U}([a=0,b=1])$	$a < b$	`*unif`
t-Studenta	$\mathrm{t}^{[d]}$	$d \in \mathbb{N}$	`*t`
Weibulla	$\mathrm{Weib}(a,s=1)$	$a > 0, s > 0$	`*weibull`

Informacja

Zwróćmy uwagę, że w przypadku kilku rozkładów niektóre parametry posiadają wartości domyślne. Zostały one przedstawione w kolumnie *Oznaczenie* po znaku równości. Przykładowo, symbol $\mathrm{N}(\mu = 0, \sigma = 1)$ wskazuje, że rozkład normalny jest opisany za pomocą dwóch parametrów: μ i σ, przy czym domyślne wartości tych parametrów wynoszą, odpowiednio, 0 i 1.

Aby uzyskać pożądaną wartość (dystrybuanty czy funkcji kwantylowej itp.), należy zastąpić znak $*$, występujący przed identyfikatorem danego rozkładu, odpowiednim przedrostkiem, zgodnie z tym, co podano w poniższej tabeli:

Przedrostek	Znaczenie
`d`	gęstość (ang. *density*) $f(x)$ lub funkcja prawdopodobieństwa $\mathrm{P}(X = x)$
`p`	dystrybuanta (ang. *distribution function*) $F(x) = \mathrm{P}(X \leqslant x)$
`q`	funkcja kwantylowa (ang. *quantile function*) $\simeq F^{-1}(p)$
`r`	generowanie liczb pseudolosowych (ang. *random deviates generation*)

3.1.1.2. Dystrybuanta

Jeśli dla wybranego rozkładu prawdopodobieństwa chcemy wyznaczyć wartość dystrybuanty w danym punkcie x, to wystarczy do identyfikatora rozkładu dostawić przedrostek „p", np.

```
> pnorm(0) # wartość dystrybuanty rozkładu normalnego N(0,1) w punkcie 0
[1] 0.5
```

Funkcja ta jest oczywiście zwektoryzowana, tzn. że pierwszym argumentem tej funkcji może być nie tylko „pojedynczy" punkt, ale też dłuższy niż 1 wektor liczb, dla których chcemy wyznaczyć wartość dystrybuanty, np.

```
> pnorm(c(1, 2, 3)) # a teraz wartość dystrybuanty w punktach 1, 2 oraz 3
[1] 0.8413447 0.9772499 0.9986501
```

Zadając wartość wyłącznie jednego argumentu funkcji (jak w dwóch powyższych przykładach), sugerujemy programowi posłużenie się domyślnymi parametrami rozkładu. W naszych przykładach był to zatem rozkład normalny standardowy, tzn. $N(0, 1)$. Przez podanie dalszych argumentów tej funkcji `pnorm` możemy określić różne od standardowych parametry rozkładu, dla którego chcemy znaleźć wartość dystrybuanty, np.

```
> pnorm(0, 2, 1) # wartość dystrybuanty rozkładu N(2,1) w punkcie 0
[1] 0.02275013
```

Jeśli zamiast wartości dystrybuanty chcemy wyznaczyć wartość funkcji przeżycia (funkcji niezawodności) w punkcie x, czyli wartość $G(x) = 1 - F(x) = \mathrm{P}(X > x)$, to

podajemy dodatkowy argument: `lower.tail=FALSE`, np.

```
> pnorm(0, lower.tail=FALSE) # wartość funkcji przeżycia N(0,1) w 0
[1] 0.5
```

Oczywiście to samo otrzymamy pisząc:

```
> 1-pnorm(0)
[1] 0.5
```

3.1.1.3. Gęstość i prawdopodobieństwo

Przedrostek `d` przed identyfikatorem rozkładu prawdopodobieństwa określa funkcję liczącą wartość gęstości (w przypadku rozkładów absolutnie ciągłych) lub funkcji prawdopodobieństwa (w przypadku rozkładów dyskretnych) w danym punkcie x lub w punktach zadanych przez elementy wektora wejściowego, np.

```
> dexp(0) # wartość f(0), gdzie f jest gęstością rozkładu Exp(1)
[1] 1
> dexp(c(0, 0.5, 1), 0.5) # wartości f(0), f(0.5), f(1) dla Exp(0.5)
[1] 0.5000000 0.3894004 0.3032653
> pr <- dbinom(0:8, 8, 0.25) # Pr(X=i) dla X~Bin(8, 1/4), i=0,1,...,8
> round(pr, 3) # wyświetl zaokrąglone do 3 miejsc po przecinku
[1] 0.100 0.267 0.311 0.208 0.087 0.023 0.004 0.000 0.000
```

3.1.1.4. Funkcja kwantylowa

Wartości kwantyli wyznaczamy pisząc przed nazwą rozkładu przedrostek `q`, przy czym pierwszym argumentem funkcji jest interesujący nas rząd kwantyla. Rozważmy następujące przykłady:

```
> qt(0.95, 5) # kwantyl rzędu 0.95 rozkładu t o 5 stopniach swobody
[1] 2.015048
> qt(0.95, c(1, 5, 10, 15)) # różne stopnie swobody
[1] 6.313752 2.015048 1.812461 1.753050
> qt(0.95, Inf) # to jest rozkład normalny standardowy
[1] 1.644854
> qnorm(0.95)
[1] 1.644854
> qt(0.95, 1) # rozkład Cauchy'ego standardowy
[1] 6.313752
> qcauchy(0.95)
[1] 6.313752
> qt(c(0.95, 0.975, 0.99, 0.995), 5) # różne rzędy kwantyla
[1] 2.015048 2.570582 3.364930 4.032143
> qt(c(0.95, 0.975, 0.99, 0.995), c(1, 5, 10, 15)) # a to co?
```

```
[1] 6.313752 2.570582 2.763769 2.946713
```

W przypadku rozkładów dyskretnych funkcja kwantylowa zmiennej X w punkcie q zwraca najmniejszą wartość $x \in \text{supp}(X)$, dla której $\text{P}(X \leqslant x) \geqslant q$, gdzie $\text{supp}(X)$ oznacza nośnik rozkładu zmiennej losowej X. W poniższym przykładzie zestawiono uzyskane kwantyle z wartościami dystrybuanty:

```
> qbinom(c(0.4, 0.5, 0.6), 5, 0.5)
[1] 2 2 3
> pbinom(0:5, 5, 0.5) # dla porównania
[1] 0.03125 0.18750 0.50000 0.81250 0.96875 1.00000
```

3.1.1.5. Generowanie liczb pseudolosowych

Generowanie liczb pseudolosowych[1] z danego rozkładu prawdopodobieństwa uruchamiamy pisząc przed nazwą rozkładu przedrostek `r`, przy czym pierwszym argumentem tej funkcji jest liczba wartości, które chcemy wygenerować, np.

```
> runif(5) # wygenerowanie 5 obserwacji z U([0,1])
[1] 0.32861008 0.85354830 0.02343164 0.28428211 0.11805379
> runif(10, 0, 5) # realizacja 5-elementowej próby z U([0,5])
 [1] 1.8893373 3.0996468 2.4328564 2.0271740 0.3749589 3.2500657
 [7] 0.3297165 3.7432225 0.3841311 4.7328907
> rpois(20, 4) # wygenerowanie 20 obserwacji z Poi(4)
 [1] 5 8 3 3 6 1 6 2 5 0 6 6 8 2 3 6 2 2 5 1
```

Zauważmy, że kolejne wywołanie np. `runif(5)` da nam inny wynik:

```
> runif(5)
[1] 0.81382298 0.42525204 0.03425531 0.49136714 0.69579459
```

Jeśli jednak – z jakichś względów – chcielibyśmy sprawić, by nasze wyniki były *odtwarzalne*, możemy ustawić ręcznie tzw. ziarno generatora liczb pseudolosowych (ang. *seed*). Dane ziarno zawsze generuje ten sam ciąg liczb.

```
> set.seed(123)
> runif(5)
[1] 0.2875775 0.7883051 0.4089769 0.8830174 0.9404673
> set.seed(123)
> runif(5) # to samo ziarno - ten sam ciąg
[1] 0.2875775 0.7883051 0.4089769 0.8830174 0.9404673
```

[1] Czyli będących rezultatem wykonania pewnego ciągu ściśle deterministycznych operacji arytmetycznych. Operacje te dają w wyniku liczby *przypominające* zachowaniem liczby losowe z zadanego rozkładu, tzn. spełniające pewne statystyczne kryteria, których spełniania spodziewalibyśmy się po liczbach „prawdziwie" losowych (np. powstałych w wyniku pomiaru fizycznego, rzutu monetą itp.). Dalej pomijać będziemy czasem przedrostek „pseudo" i mówić po prostu o liczbach losowych.

3.1.2. Losowanie bez i ze zwracaniem

Do generowania próbek będących wynikiem n-krotnego losowania elementów z danego zbioru S można użyć funkcji `sample()`. Nazwa zbioru S jest pierwszym argumentem tej funkcji, zaś liczność próbki n – drugim argumentem. Domyślnie otrzymujemy wynik losowania bez zwracania. Gdy chcemy losować ze zwracaniem, to jako kolejny argument funkcji `sample()` wpisujemy `replace=TRUE`.

Na przykład, wynik $n = 15$ rzutów monetą można otrzymać następująco:

```
> sample(c("O", "R"), 15, replace=TRUE)
 [1] "O" "R" "R" "R" "O" "R" "O" "R" "R" "O" "R" "O" "O" "O" "R"
```

Parametr n można także pominąć – poniższy kod wygeneruje nam losową permutację zbioru $\{1, 2, \ldots, 10\}$.

```
> sample(1:10) # losowanie bez zwracania 10 elementów z {1,...,10}
 [1]  9  7  6 10  4  8  3  2  1  5
```

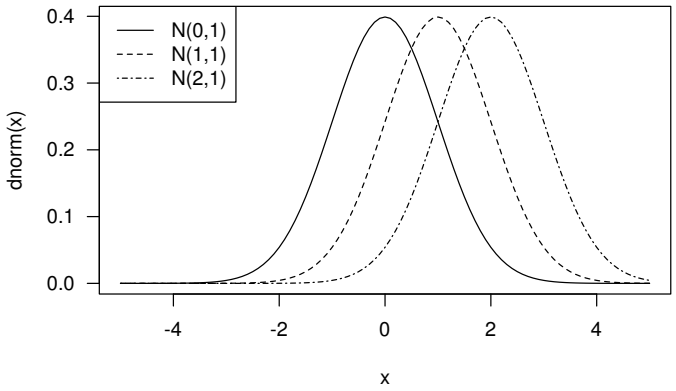
3.2. Zadania rozwiązane

3.2.1. Rozkłady prawdopodobieństwa

Zadanie 3.1. Utwórz wykresy gęstości, dystrybuanty i funkcji przeżycia dla zmiennych losowych z rozkładów normalnych $\mathrm{N}(0, 1)$, $\mathrm{N}(1, 1)$, $\mathrm{N}(2, 1)$.

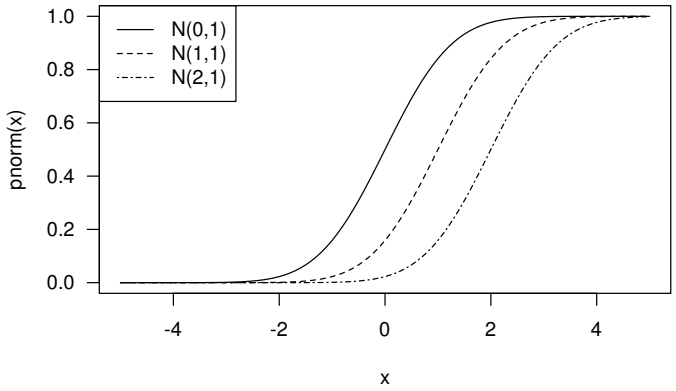
Rozwiązanie. Dla wygody do narysowania wykresów użyjemy funkcji `curve()`. Wywołanie `curve(f(x), from=x1, to=x2)` powoduje próbkowanie wartości funkcji `f()` w wielu punktach `x` z przedziału $[\mathtt{x1}, \mathtt{x2}]$ i przedstawienie ich na rysunku.

Gęstości rozkładów normalnych o różnych parametrach położenia:



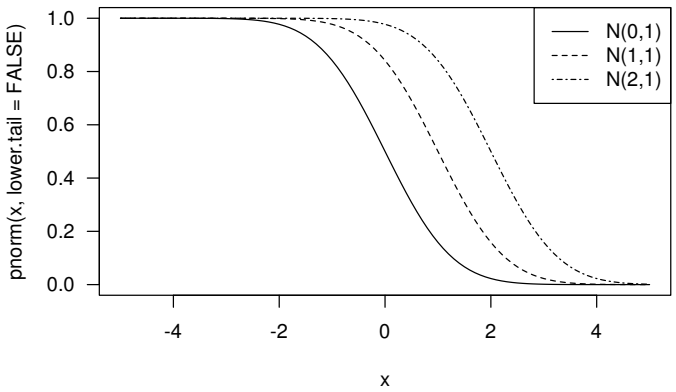
```
> curve(dnorm(x), from=-5, to=5, las=1)
> curve(dnorm(x, 1, 1), lty=2, add=TRUE)  # dodanie kolejnej krzywej
> curve(dnorm(x, 2, 1), lty=4, add=TRUE)  # i jeszcze jednej
> legend("topleft", c("N(0,1)", "N(1,1)", "N(2,1)"), lty=c(1, 2, 4))
```

W podobny sposób możemy sporządzić wykresy dystrybuant rozkładów normalnych o różnych parametrach położenia:




```
> curve(pnorm(x), from=-5, to=5, las=1)
> curve(pnorm(x, 1, 1), lty=2, add=TRUE)
> curve(pnorm(x, 2, 1), lty=4, add=TRUE)
> legend("topleft", c("N(0,1)", "N(1,1)", "N(2,1)"), lty=c(1, 2, 4))
```

lub wykresy funkcji przeżycia tychże rozkładów:



```
> curve(pnorm(x, lower.tail=FALSE), from=-5, to=5, las=1)
> curve(pnorm(x, 1, 1, lower.tail=FALSE), lty=2, add=TRUE)
> curve(pnorm(x, 2, 1, lower.tail=FALSE), lty=4, add=TRUE)
> legend("topright", c("N(0,1)", "N(1,1)", "N(2,1)"), lty=c(1, 2, 4))
```

□

Zadanie 3.2. Sprawdź tzw. regułę 3-sigmową dla rozkładu normalnego. Utwórz graficzną ilustrację tej reguły.

Rozwiązanie. Regułę trzech sigm dla zmiennej losowej $X \sim \mathrm{N}(\mu, \sigma)$ przedstawia się następująco:

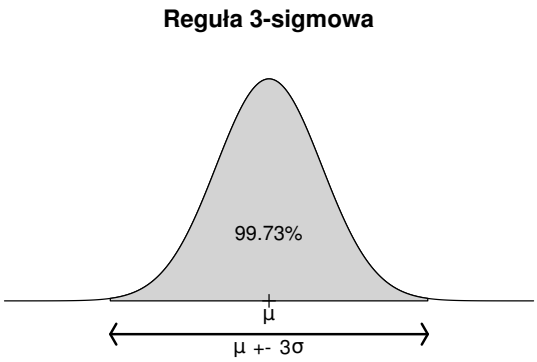
$$\mathrm{P}\left(\mu - 3\sigma \leqslant X \leqslant \mu + 3\sigma\right) \simeq 0.9973. \tag{3.1}$$

Obliczmy to prawdopodobieństwo (czyli np. $\mathrm{P}\left(X \in [-3, 3]\right)$, gdzie $X \sim \mathrm{N}(0, 1)$) przy użyciu R:

```
> pnorm(3)-pnorm(-3)
[1] 0.9973002
```

Reguła 3-sigmowa dla rozkładu $\mathrm{N}(\mu, \sigma)$ może być przykładowo zilustrowana (np. w podręczniku do rachunku prawdopodobieństwa) w sposób następujący.

1. Narysujemy funkcję gęstości.
2. Pokolorujemy odpowiedni fragment pola pod krzywą (za pomocą funkcji `polygon()`, służącej do rysowania wielokątów).
3. Umieścimy odpowiednie etykietki tekstowe (funkcje `arrows()`, `text()`).



Obliczenia przeprowadzimy dla rozkładu normalnego standaryzowanego. Zachęcamy Czytelnika do samodzielnego przestudiowania stron systemu pomocy dotyczących używanych funkcji graficznych.

```
> x <- seq(-5, 5, by=0.01) # wektor argumentów, dla których obliczymy
>                          # wartości gęstości rozkładu N(0,1)
> y <- dnorm(x)
> plot(x, y, type="l", main="Reguła 3-sigmowa", xlab=NA, ylab=NA,
+         axes=FALSE, ylim=c(-max(y)*0.2, max(y))) # wykres gęstości
```

Definiujemy współrzędne punktów, które będą wierzchołkami wielokąta, który następnie wypełnimy kolorem żółtym (p. 2.):

```
> wx <- c(-3, x[x>=-3 & x<=3], 3)
> wy <- c( 0, y[x>=-3 & x<=3], 0)
> polygon(wx, wy, col="lightgray")
```

Dodajemy oznaczenie punktu $x = \mu$, rysujemy podpisaną strzałkę wskazującą interesujący nas przedział na osi poziomej i dodajemy informację o wartości pola pod fragmentem krzywej (p. 3.):

```
> points(0, 0, pch=3) # Oznaczenie punktu x=mi
> text(0, 0, expression(mu), pos=1)
> # Generujemy strzałkę dla przedziału:
> arrows(-3,-max(y)*0.15,3,-max(y)*0.15,angle=60, length=0.1,code=3,lwd=2)
> text(0, -max(y)*0.22, # Etykietka pod strzałką
+  expression(mu~"+-"~3*sigma))
> text(0, max(y)*0.3, paste(round(100*(pnorm(3)-pnorm(-3)), 2),
+    "%", sep=""))
```

⊡

Zadanie 3.3. Wzrost pewnej grupy osób opisany jest rozkładem normalnym o wartości oczekiwanej 173 cm i odchyleniu standardowym 6 cm.

1. Jakie jest prawdopodobieństwo, że losowo wybrana osoba ma nie więcej niż 179 cm wzrostu?
2. Jaka jest frakcja osób mających wzrost pomiędzy 167 i 180 cm?
3. Jakie jest prawdopodobieństwo, że losowo wybrana osoba ma więcej niż 181 cm wzrostu?
4. Wyznacz wartość wzrostu, której nie przekracza 60% badanej populacji osób.

Rozwiązanie. Załóżmy, że mamy do czynienia ze zmienną losową $X \sim \mathrm{N}(173, 6)$, opisującą wzrost losowo napotkanej osoby z danej grupy.

Na początek obliczamy $\mathrm{P}(X \leqslant 179)$, które – jak wiadomo – jest równe dystrybuancie stosownego rozkładu w punkcie 179, tzn.

```
> pnorm(179, 173, 6)
[1] 0.8413447
```

Dalej obliczamy $\mathrm{P}(167 < X \leqslant 180)$, które jest równe różnicy dystrybuant naszego rozkładu na krańcach rozważanego przedziału, czyli

```
> pnorm(180, 173, 6)-pnorm(167, 173, 6)
[1] 0.7196722
```

Trzecie pytanie dotyczy $\mathrm{P}(X > 181)$. Interesującą nas wartość możemy obliczyć dwiema metodami:

```
> 1-pnorm(181, 173, 6); # lub równoważnie:
[1] 0.09121122
> pnorm(181, 173, 6, lower.tail=FALSE)
[1] 0.09121122
```

Wreszcie szukamy kwantyla $q_{0.6}$ rozkładu $\mathrm{N}(173, 6)$, to jest

```
> qnorm(0.6,173,6)
[1] 174.5201
```

⊡

Zadanie 3.4. Utwórz tablicę wartości dystrybuanty rozkładu standardowego normalnego.

Rozwiązanie. Stworzymy wpierw wektor z wartościami dystrybuanty $\Phi(x)$ rozkładu $\mathrm{N}(0, 1)$ dla $x \in \{0, 0.2, \ldots, 3.8\}$.

```
> x <- seq(0, 3.8, 0.2)
> y <- pnorm(x)
> length(x)
[1] 20
```

a następnie przekształćmy ów wektor y na macierz. Można to zrobić w następujący sposób:

```
> dim(y)=c(5,4)
> y
          [,1]      [,2]      [,3]      [,4]
[1,] 0.5000000 0.8413447 0.9772499 0.9986501
[2,] 0.5792597 0.8849303 0.9860966 0.9993129
[3,] 0.6554217 0.9192433 0.9918025 0.9996631
[4,] 0.7257469 0.9452007 0.9953388 0.9998409
[5,] 0.7881446 0.9640697 0.9974449 0.9999277
```

Zauważmy, że w naszej tablicy w pierwszym wierszu mamy wartości $\Phi(x)$ dla $x \in \{0, 0.2, 0.4, 0.6, 0.8\}$, w drugim – dla $x \in \{1, 1.2, 1.4, 1.8\}$ itd. Jeśli chcemy mieć tablicę do czytania „wierszami" możemy zrobić np. tak:

```
> matrix(y, 4, 5, byrow=TRUE)
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.5000000 0.5792597 0.6554217 0.7257469 0.7881446
[2,] 0.8413447 0.8849303 0.9192433 0.9452007 0.9640697
[3,] 0.9772499 0.9860966 0.9918025 0.9953388 0.9974449
[4,] 0.9986501 0.9993129 0.9996631 0.9998409 0.9999277
```

```
> # albo: t(y) # (transpozycja macierzy)
```

Aby było wygodniej korzystać z tak otrzymanej tablicy, warto dodatkowo nadać nazwy poszczególnym wierszom i kolumnom:

```
> matrix(y, 4, 5, byrow=TRUE,
+    dimnames=list(0:3, seq(0, 0.8, by=0.2)))
          0       0.2       0.4       0.6       0.8
0 0.5000000 0.5792597 0.6554217 0.7257469 0.7881446
1 0.8413447 0.8849303 0.9192433 0.9452007 0.9640697
2 0.9772499 0.9860966 0.9918025 0.9953388 0.9974449
3 0.9986501 0.9993129 0.9996631 0.9998409 0.9999277
```

⊡

Zadanie 3.5. Sporządź wykres funkcji prawdopodobieństwa następujących rozkładów dwumianowych: $\mathrm{Bin}(10, 0.5), \mathrm{Bin}(10, 0.25), \mathrm{Bin}(50, 0.25)$.

Rozwiązanie. Wyznaczamy najpierw wartości prawdopodobieństwa $\mathrm{P}(X = k)$ w punktach $k = 0, 1, \ldots, 10$ dla zmiennej losowej $X \sim \mathrm{Bin}(10, 0.5)$, tzn.

```
> x <- dbinom(0:10, 10, 0.5)
```

po czym podobnie postępujemy w przypadku pozostałych rozkładów:

```
> y <- dbinom(0:10, 10, 0.5)
> z <- dbinom(0:50, 50, 0.25)
```

Funkcje prawdopodobieństwa tych rozkładów rysujemy jako wykresy słupkowe, za pomocą komendy

```
> barplot(x, names.arg=0:10)
> barplot(y, names.arg=0:10)
> barplot(z, names.arg=0:50)
```

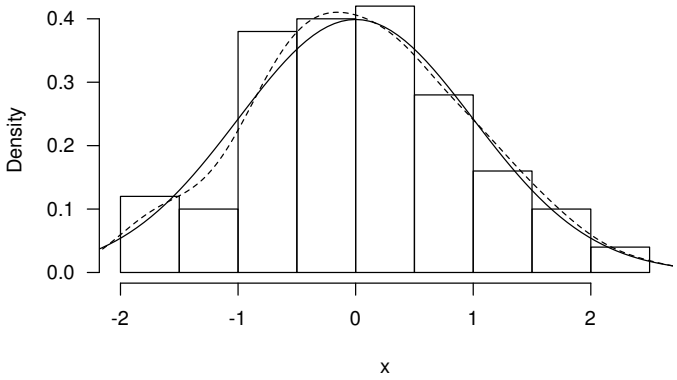
Sporządzenie wynikowych wykresów i wyciągnięcie wniosków pozostawiamy Czytelnikowi. ⊡

3.2.2. Eksperymenty symulacyjne

Zadanie 3.6. Wygeneruj n-elementową ($n = 100$) próbę losową z rozkładu normalnego standardowego. Utwórz histogram oraz estymator jądrowy dla tej próby. Nałóż na uzyskany obraz wykres gęstości teoretycznej rozkładu normalnego.

Rozwiązanie. Rozwiązanie tego zadania jest bardzo proste:

```
> n <- 100
> x <- rnorm(n) # próba losowa z rozkładu N(0,1)
> hist(x, prob=TRUE, las=1, main=NA)
> lines(density(x), lty=2)
> curve(dnorm(x), from=-3, to=3, lty=1, add=TRUE)
```



Rys. 3.1

Gęstość teoretyczną na rys. 3.1 narysowano linią ciągłą, natomiast wykres uzyskany za pomocą estymatora jądrowego linią przerywaną.

Oczywiście kolejna wygenerowana próba będzie (z prawdopodobieństwem 1) składała się z innych obserwacji. Warto więc przyjrzeć się wykresom dla kilku realizacji tego eksperymentu, wywołując powyższy kod kilkakrotnie. ⊡

Zadanie 3.7. Wygeneruj próbkę losową z rozkładu Pareto z parametrem $a = 2$, korzystając z generatora liczb losowych o rozkładzie jednostajnym na przedziale $[0, 1]$,

Rozwiązanie. Rozwiązując nasze zadanie posłużymy się następującym twierdzeniem:

Niech F oznacza dystrybuantę zmiennej losowej X. Zdefiniujmy funkcję kwantylową F^{-1} następująco

$$F^{-1}(x) = \inf\{t : F(t) \geqslant x\}. \tag{3.2}$$

Wtedy zmienna losowa X ma taki sam rozkład jak $Y = F^{-1}(U)$, gdzie U jest zmienną losową o rozkładzie jednostajnym $\mathrm{U}([0, 1])$.

Zauważmy, że jeśli F jest dystrybuantą ciągłą, to $Y = F(X)$ ma taki sam rozkład jak U. Zauważmy przy tym, że gdy F jest ciągła i rosnąca, to F^{-1} jest funkcją odwrotną do F w zwykłym sensie. Sposób konstruowania generatorów liczb losowych z zadanego rozkładu z wykorzystaniem generatora rozkładu $\mathrm{U}([0, 1])$ nazywamy *metodą odwracania dystrybuanty*.

Wróćmy jednak do naszego zadania. Zmienna losowa X o rozkładzie Pareto z parametrem a ma rozkład o gęstości:

$$f(x) = \frac{a}{x^{a+1}}, \tag{3.3}$$

dla $x > 1$ oraz dystrybuantę postaci:

$$F(x) = 1 - \frac{1}{x^a}. \tag{3.4}$$

Stąd

$$F^{-1}(x) = (1-x)^{-1/a}, \tag{3.5}$$

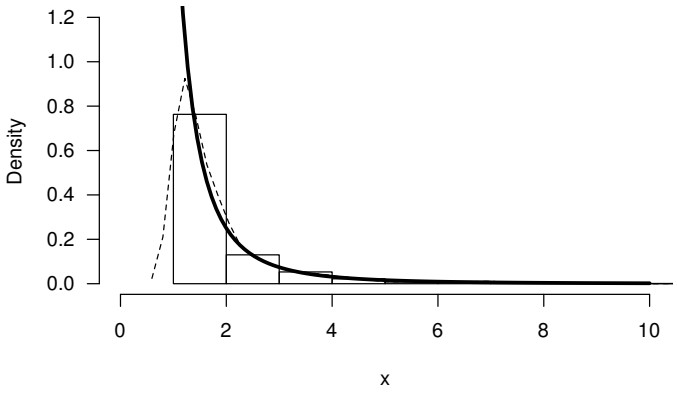
czyli – na mocy przytoczonego wyżej twierdzenia i zakładając, że $U \sim \mathrm{U}([0,1])$ – zmienna losowa $Y = F^{-1}(U) = (1-U)^{-1/a}$ będzie miała rozkład Pareto. Podstawiając $a = 2$ otrzymamy generator poszukiwanego rozkładu.

Próbkę losową generujemy zatem w następujący sposób:

```
> n <- 1000
> u <- runif(n)
> x <- u^(-0.5) # rozkład 1-u jest taki sam jak rozkład u
```

Narysujmy jeszcze histogram dla naszej próbki, wykres estymatora jądrowego gęstości oraz wykres gęstości teoretycznej:

```
> hist(x, prob=TRUE, main=NA, ylim=c(0.0, 1.2),
+    xlim=c(0, 10), breaks=100, las=1)
> lines(density(x), lty=2)
> curve(2/x^3, add=TRUE, lty=1, lwd=3, from=1)
```



⊡

Zadanie 3.8. Posługując się metodą Monte Carlo, oblicz pole powierzchni obszaru

$$A = \left\{(x,y) \in \mathbb{R}^2 : 0 < x < 1;\ 0 < y < x^2\right\}.$$

Porównaj uzyskane w ten sposób wyniki z dokładnymi rezultatami otrzymanymi na drodze analitycznej.

Rozwiązanie. Skorzystamy tu z następującego faktu: *Niech* $X_1, Y_1, X_2, Y_2, \ldots$ *będą niezależnymi zmiennymi losowymi o rozkładzie jednostajnym* $U([0,1])$. *Dla funkcji borelowskiej* $f : [0,1] \to [0,1]$ *definiujemy*

$$Z_i = \mathbf{1}\left(Y_i \leqslant f(X_i)\right), \quad i = 1, 2, \ldots, \tag{3.6}$$

gdzie $\mathbf{1}(\cdot)$ *jest funkcją indykatorową. Wówczas, z mocnego prawa wielkich liczb (MPWL), z prawdopodobieństwem równym 1 zachodzi*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} Z_i = \int_0^1 f(x)\, dx. \tag{3.7}$$

Informacja

Możliwe jest uogólnienie powyższego faktu na funkcje zdefiniowane dla innych dziedzin i przeciwdziedzin niż przedział jednostkowy. Pozostawiamy to jako ćwiczenie dla Czytelnika.

Wróćmy do naszego zadania. Pole obszaru A można obliczyć nastepująco:

$$\int_A dx\, dy = \int_0^1 \left(\int_0^{x^2} dy \right) dx = \int_0^1 x^2\, dx = \frac{1}{3}.$$

Skonfrontujemy ten wynik z wartością uzyskaną w sposób przybliżony, obliczając całkę $\int_0^1 x^2\, dx$ metodą omówioną powyżej, zwaną *całkowaniem Monte Carlo*[2].

Generujemy najpierw próbkę losową $(U_1, V_1, \ldots, U_n, V_n)$ z rozkładu jednostajnego:

```
> n <- 10000
> u <- runif(n)
> v <- runif(n)
```

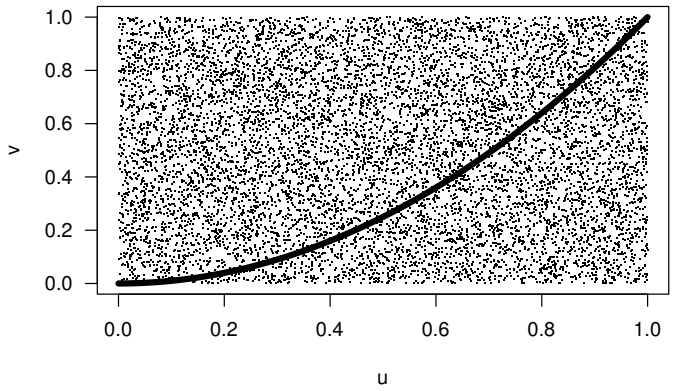
Zaznaczmy te punkty na obrazku i nałóżmy na niego wykres funkcji $y = x^2$ dla $x \in [0,1]$.

```
> plot(u, v, xlim=c(0, 1), ylim=c(0, 1), las=1,
+    pch='.') # punkty oznaczamy "kropką"
> curve(x*x, type="l", lwd=5, add=T)
```

Zliczmy teraz punkty z naszej próbki, które znalazły się poniżej wykresu funkcji $y = x^2$:

```
> z <- (v <= u*u)
> sum(z) # przypominamy, TRUE ma wartość 1, FALSE - 0
[1] 3264
```

[2]Metoda całkowania Monte Carlo została zaproponowana przez polskiego matematyka Stanisława Ulama, biorącego udział w tzw. projekcie Manhattan.



Dzieląc otrzymaną liczbę punktów przez liczbę wszystkich wygenerowanych punktów (w naszym przypadku wynosi ona 10000), otrzymamy, że pole interesującego nas obszaru wynosi w przybliżeniu:

```
> mean(z)
[1] 0.3264
```

Informacja

W R dostępna jest także funkcja do całkowania numerycznego o nazwie `integrate()`. Oto przykład zastosowania tejże funkcji:

```
> integrate(dnorm, -1.96, 1.96)
0.9500042 with absolute error < 1e-11
> pnorm(1.96)-pnorm(-1.96)
[1] 0.9500042
> integrate(dnorm, -Inf, Inf)
1 with absolute error < 9.4e-05
```

Pamiętajmy, że jako pierwszy argument `integrate()` należy podać funkcję do scałkowania. Chcąc policzyć całki występujące w naszym zadaniu, należy stworzyć własną funkcję o składni:

```
> nazwaFunkcji <- function(argument1, argument2, (...))
{
   (...) różne operacje (...)
   return(wynik) # bądź po prostu 'wynik'
}
```

Spróbujmy więc:

```
> integrate(function(x) { return(x^2) }, 0, 1)
0.3333333 with absolute error < 3.7e-15
```

□

3.3. Zadania do samodzielnego rozwiązania

Zadanie 3.9. Utwórz wykresy gęstości, dystrybuanty i funkcji przeżycia dla zmiennych losowych z następujących rozkładów normalnych $\mathrm{N}(0,1)$, $\mathrm{N}(0,0.5)$, $\mathrm{N}(0,2)$.

Zadanie 3.10. Utwórz tablicę podstawowych kwantyli (tzn. rzędu $0.9, 0.95, 0.975, 0.99, 0.995$) rozkładu standardowego normalnego.

Zadanie 3.11. Utwórz tablicę podstawowych kwantyli rozkładu chi-kwadrat o różnych stopniach swobody (tzn. kwantyli rzędu $0.005, 0.01, 0.025, 0.05, 0.1, 0.9, 0.95, 0.975, 0.99, 0.995$).

Zadanie 3.12. Utwórz wykresy gęstości zmiennych losowych z rozkładu chi-kwadrat o 5, 10 oraz 40 stopniach swobody. Przeanalizuj, jak zmienia się gęstość rozkładu χ^2 wraz ze wzrostem liczby stopni swobody.

⋆ **Zadanie 3.13.** Przeprowadź eksperyment symulacyjny pokazujący, że rozkład chi-kwadrat, wraz ze wzrostem liczby stopni swobody, zbiega do rozkładu normalnego.

Zadanie 3.14. Utwórz tablicę podstawowych kwantyli (tzn. rzędu $0.9, 0.95, 0.975, 0.99, 0.995$) rozkładu t-Studenta o różnych stopniach swobody.

Zadanie 3.15. Utwórz wykresy gęstości zmiennych losowych o rozkładzie t-Studenta z 1, 5 i 30 stopniami swobody. Porównaj otrzymane wykresy z wykresem gęstości zmiennej losowej o rozkładzie normalnym standardowym.

⋆ **Zadanie 3.16.** W wielu tablicach statystycznych sugeruje się, że rozkład t-Studenta już od 30 stopni swobody można dobrze przybliżać rozkładem normalnym standardowym. Niech Φ oznacza dystrybuantę rozkładu $\mathrm{N}(0,1)$, a F_d – dystrybuantę rozkładu $\mathrm{t}^{[d]}$. Dla różnych liczb stopni swobody d zbadaj wartości funkcji błędu:

$$e(d) = \sup_{x \in \mathbb{R}} |F_d(x) - \Phi(x)|,$$

którą można aproksymować za pomocą wyrażenia

$$e(d) \simeq \max_{x=-\lambda,-\lambda+\delta,\ldots,\lambda} |F_d(x) - \Phi(x)|,$$

gdzie np. $\lambda = 5$ oraz $\delta = 0.001$.

Zadanie 3.17. Utwórz wykresy gęstości zmiennych losowych o następujących rozkładach gamma:

1. $\Gamma(1,1), \Gamma(0.5,1), \Gamma(2,1), \Gamma(3,1)$,
2. $\Gamma(2,1), \Gamma(2,2), \Gamma(2,3)$.

Sformułuj wnioski dotyczące wpływu poszczególnych parametrów rozkładu na kształt wykresu gęstości.

Zadanie 3.18. Utwórz wykresy gęstości zmiennych losowych o następujących rozkładach beta: $Beta(1,1), Beta(2,2), Beta(5,2)$ i $Beta(2,5)$. Sformułuj wnioski dotyczące wpływu poszczególnych parametrów rozkładu na kształt wykresu gęstości.

Zadanie 3.19. Utwórz wykresy gęstości zmiennych losowych o następujących rozkładach F-Snedecora: $\mathrm{F}^{[10,5]}, \mathrm{F}^{[10,10]}, \mathrm{F}^{[10,20]}$.

Zadanie 3.20. Średnio jedna na dziesięć osób mijających pewien sklep wchodzi do tego sklepu. Niech X oznacza numer pierwszej osoby, która weszła do sklepu, podczas gdy $X-1$ osób, które wcześniej mijały ów sklep, nie weszło do środka. Oblicz prawdopodobieństwa $\mathrm{P}(X=1), \mathrm{P}(X=2), \mathrm{P}(X=3), \mathrm{P}(X=4)$ oraz $\mathrm{P}(X>11)$.

Zadanie 3.21. W partii towaru liczącej 200 sztuk znajduje się 5 sztuk niespełniających wymagań jakościowych. Jakie jest prawdopodobieństwo, że w losowej próbie 10 sztuk pobranych z tej partii nie znajdzie się ani jedna sztuka wadliwa?

Zadanie 3.22. Czas poprawnej pracy aparatu telefonicznego ma rozkład wykładniczy o intensywności awarii 0.0001 [1/h].

1. Oblicz prawdopodobieństwo, że aparat ten nie uszkodzi się w ciągu: 1000, 10000, 30000 godzin pracy.
2. Ile godzin, co najmniej, powinien przepracować bezawaryjnie ten aparat z prawdopodobieństwem 0.9?

Zadanie 3.23. Z dotychczasowych obserwacji wynika, że liczba klientów przybywających w ciągu godziny do oddziału banku ma rozkład Poissona o średniej 4 [klientów/h].

1. Jaki jest rozkład prawdopodobieństwa czasu między przyjściem kolejnych klientów?
2. Jaki jest średni czas oraz odchylenie standardowe czasu pomiędzy chwilami przybycia kolejnych klientów?
3. Jeżeli w danej chwili do oddziału wszedł klient, to jakie jest prawdopodobieństwo, że kolejny klient przybędzie do oddziału w ciągu najbliższych 30 minut?
4. Jakie jest prawdopodobieństwo, że w ciągu godziny do oddziału banku nie przyjdzie ani jeden klient?

Zadanie 3.24. Wygeneruj $n = 100$ liczb z rozkładu $U([0,10])$. Znajdź maksimum i minimum otrzymanej próbki.

Zadanie 3.25. Wygeneruj $n = 100$ liczb z rozkładu $\mathrm{N}(3, 3)$. Ile z nich jest ujemnych?

Zadanie 3.26. Wygeneruj $n = 1000$ liczb z rozkładu $\mathrm{N}(1, 2)$. Ile z nich różni się od średniej o więcej niż 2 odchylenia standardowe?

Zadanie 3.27. Napisz program symulujący $n = 20$ rzutów symetryczną monetą. Ile razy wypadła reszka?

Zadanie 3.28. W urnie jest $n = 60$ kul, ponumerowanych od 1 do n. Wylosuj spośród nich bez zwracania $m = 30$ kul. Jaki jest największy i najmniejszy numer wylosowanej kuli? Powtórz eksperyment losując ze zwracaniem.

Zadanie 3.29. Napisz program symulujący $n = 100$ rzutów symetryczną kostką do gry. Ile razy wypadła „szóstka" lub „piątka"?

Zadanie 3.30. Wylosuj ze zwracaniem $n = 1000$ kart do gry. Ile otrzymaliśmy asów?

Zadanie 3.31. Rzucamy $n = 1000$ razy dwiema symetrycznymi monetami. Wygeneruj odpowiednią próbkę za pomocą R. Ile razy otrzymaliśmy dwa orły?

Zadanie 3.32. Korzystając z generatora liczb losowych o rozkładzie jednostajnym na przedziale $[0, 1]$ wygeneruj próbkę losową z rozkładu wykładniczego z parametrem $\lambda = 5$. Narysuj histogram dla uzyskanych danych.

Zadanie 3.33. Korzystając z generatora liczb losowych o rozkładzie jednostajnym na przedziale $[0, 1]$ wygeneruj próbkę losową z rozkładu logistycznego. Narysuj histogram dla uzyskanych danych.

⋆ **Zadanie 3.34.** Posługując się metodą Monte Carlo oblicz pole powierzchni obszaru $B = \{(x, y) \in \mathbb{R}^2 : x^2 < y < 1 - x^2\}$. Porównaj uzyskane w ten sposób wyniki z dokładnymi rezultatami otrzymanymi na drodze analitycznej.

⋆ **Zadanie 3.35.** Posługując się metodą Monte Carlo wyznacz przybliżoną wartość liczby π.

Zadanie 3.36. Niech X oznacza zmienną losową o rozkładzie normalnym standardowym. Oblicz wartości następujących prawdopodobieństw:

1. $P(-1 < X < 1)$,
2. $P(-2 < X < 2)$,
3. $P(-3 < X < 3)$.

Następnie wygeneruj $n = 10000$ elementową próbę $(X_1, \ldots, X_n)$ z rozkładu normalnego standardowego i porównaj częstości wystąpienia zdarzeń: $A : X_i \in (-1, 1)$, $B : X_i \in (-2, 2)$, $C : X_i \in (-3, 3)$ z wartościami odpowiednich prawdopodobieństw wyznaczonych powyżej.

Zadanie 3.37. Wygeneruj $m = 100$ próbek $n = 200$ elementowych $(U_1, \ldots, U_n)$ z rozkładu jednostajnego na przedziale $[0, 1]$. Utwórz histogramy dla zmiennych $(Z_1, \ldots, Z_m)$, gdzie

$$Z_k = \frac{\sum_{i=1}^{k} U_i - k/2}{\sqrt{k/12}},$$

dla $k = 1, \ldots, m$. Nałóż na histogram wykres gęstości rozkładu normalnego standardowego. Sformułuj wnioski odnośnie zmiany kształtu histogramu zmiennej Z_k wraz ze wzrostem k.

Zadanie 3.38. Niech X oznacza zmienną losową o rozkładzie dwumianowym $\mathrm{Bin}(n, p)$. Wyznacz tablice prawdopodobieństw $\mathrm{P}(X \leqslant k)$ dla kilku wybranych wartości k. Porównaj te prawdopodobieństwa z wartościami prawdopodobieństw otrzymanymi za pomocą aproksymacji

1. rozkładem Poissona,
2. rozkładem normalnym (tw. Moivre'a-Laplace'a),
3. rozkładem normalnym z korektą ciągłości.

Porównaj również wykres dystrybuanty zmiennej losowej X z wykresami dystrybuant rozkładów użytych do aproksymacji X. Sformułuj wnioski dotyczące jakości aproksymacji, biorąc pod uwagę różne wartości parametrów n oraz p np. $n = 20, 30, 50$ oraz $p = 0.1, 0.2, 0.3, 0.5$.

3.4. Wskazówki i odpowiedzi do zadań

Ad zad. 3.20. X ma rozkład geometryczny.

Ad zad. 3.21. X ma rozkład hipergeometryczny.

Ad zad. 3.23. $X \sim \mathrm{Exp}(\lambda)$, gdzie $\mathbb{E}X = \frac{1}{4}$.

Ad zad. 3.33. Dystrybuanta rozkładu logistycznego ma postać: $F(x) = 1/(1 + e^{-x})$.

Estymacja punktowa i przedziałowa

4

4.1. Wprowadzenie

4.1.1. Dystrybuanta empiryczna

Dystrybuantą empiryczną (ang. *empirical distribution function*) zbudowaną na podstawie próbki $\mathbf{X} = (X_1, \ldots, X_n)$ nazywamy funkcję

$$\widehat{F}_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{(-\infty,t]}(X_i), \quad t \in \mathbb{R}. \tag{4.1}$$

Dla zaobserwowanej realizacji próbki $\mathbf{x} = (x_1, \ldots, x_n)$ funkcja $\widehat{F}_n$ jest dystrybuantą schodkową, mającą skoki o wysokości $1/n$ w punktach $x_1, \ldots, x_n$.

Do narysowania wykresu dystrybuanty empirycznej w programie R można użyć funkcji `ecdf()`. Wektor `x`, zawierający wartości próbki $x_1, \ldots, x_n$, podajemy jako argument tej funkcji: `ecdf(x)`.

4.1.2. Obciążenie i błąd średniokwadratowy estymatora

Przypomnijmy, że obciążenie estymatora $\hat{\theta}$ parametru θ definiujemy następująco

$$b\left(\hat{\theta}\right) = \mathbb{E}\left(\hat{\theta}\right) - \theta, \tag{4.2}$$

natomiast błąd średniokwadratowy estymatora jako

$$\mathrm{MSE}\left(\hat{\theta}\right) = \mathbb{E}\left(\hat{\theta} - \theta\right)^2 = \mathrm{Var}\left(\hat{\theta}\right) + \left[b\left(\hat{\theta}\right)\right]^2. \tag{4.3}$$

W praktyce te dwie ważne charakterystyki estymatora nie zawsze dają się łatwo wyznaczyć analitycznie. Wówczas mogą okazać się pomocne badania symulacyjne. Zauważmy, że zarówno obciążenie, jak i błąd średniokwadratowy są, de facto, wartościami oczekiwanymi pewnych statystyk. Stąd też naturalnymi kandydatami na empiryczne oszacowanie tych wielkości będą średnie z odpowiednio przygotowanych próbek (w przypadku MSE możemy się również posłużyć wariancją próbkową). Uzasadnieniem takiego postępowania jest choćby fakt, iż średnia arytmetyczna jest estymatorem nieobciążonym, zgodnym (i często efektywnym) wartości oczekiwanej.

4.1.3. Przedziały ufności

W R mamy do dyspozycji funkcje pozwalające wyznaczać przedziały ufności (ang. *confidence intervals*) m.in. dla wartości oczekiwanej w modelu normalnym z nieznanym odchyleniem standardowym oraz dla wskaźnika struktury (proporcji; prawdopodobieństwa sukcesu w rozkładzie Bernoulliego).

Niech $(X_1, \ldots, X_n)$ będzie próbą z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$ o nieznanych parametrach μ i σ. Do wyznaczenia przedziału ufności dla wartości oczekiwanej μ można użyć funkcji `t.test()`. Pierwszym argumentem tej funkcji jest wektor reprezentujący zaobserwowaną próbę, na podstawie których szacujemy μ. Poziom ufności (ang. *confidence level*) podajemy jako drugi argument tej funkcji, np. `conf.level=0.9` (domyślnym poziomem ufności jest 0.95).

Niech $(X_1, \ldots, X_n)$ będzie próbą z rozkładu dwupunktowego $\mathrm{Bern}(p)$. Do wyznaczenia przedziału ufności dla prawdopodobieństwa sukcesu p (wskaźnika struktury, proporcji) można użyć funkcji `binom.test()` lub `prop.test()`, przy czym w drugim przypadku dostajemy asymptotyczny przedział ufności. Pierwszym argumentem obu tych funkcji jest liczba jedynek w naszej próbie (odpowiadająca liczbie umownych sukcesów w rozważanym doświadczeniu, a więc elementów posiadających interesującą nas cechę), a drugim – liczność próby n. Poziom ufności podajemy jako kolejny argument, np. `conf.level=0.9` (domyślnym poziomem ufności jest 0.95).

Inne przedziały ufności wyznaczamy sami, implementując odpowiedni kod.

4.2. Zadania rozwiązane

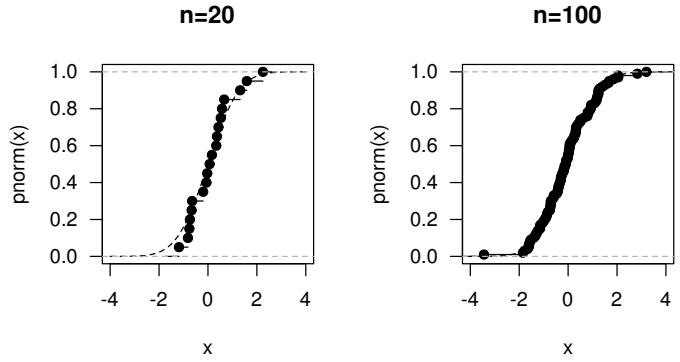
Zadanie 4.1. Wygeneruj dwie próby losowe z rozkładu standardowego normalnego: 20 i 100 elementową. Narysuj dla obu prób dystrybuanty empiryczne i porównaj je z odpowiednią dystrybuantą teoretyczną.

Rozwiązanie. Generujemy próby losowe: 20 i 100 elementową z rozkładu normalnego standardowego.

```
> y20  <- rnorm(20)
> y100 <- rnorm(100)
```

Dla każdej z próbek rysujemy wykres dystrybuanty rozkładu $\mathrm{N}(0,1)$ i nakładamy na niego wykres dystrybuanty empirycznej, por. rys. 4.1:

```
> par(mfrow=c(1,2)) # 2 wykresy na jednym rysunku
> curve(pnorm(x), from=-4, to=4, lty=2, main="n=20", las=1)
> plot(ecdf(y20), add=TRUE)
> curve(pnorm(x), from=-4, to=4, lty=2, main="n=100", las=1)
> plot(ecdf(y100), add=TRUE)
```



Rys. 4.1. Dystrybuanty empiryczne; $n = 20$ (po lewej) oraz $n = 100$ (po prawej)

Informacja

Zwróćmy uwagę, że powyższe zadanie jest empiryczną ilustracją lematu Glivienki-Cantellego, zwanego też *podstawowym twierdzeniem statystyki matematycznej*. Zachęcamy Czytelnika do zastanowienia się nad uzasadnieniem drugiej nazwy tego twierdzenia.

□

Zadanie 4.2. Wygeneruj $n = 500$-elementową próbę $Y_1, \ldots, Y_n$ z rozkładu standardowego Cauchy'ego (tzn. z parametrem położenia $a = 0$ i parametrem rozproszenia $b = 1$).

1. Dla każdej podpróbki zawierającej i początkowych elementów próbki wyjściowej, tzn. dla $X_i = (Y_1, \ldots, Y_i)$, gdzie $i = 1, \ldots, n$, wyznacz średnią $\bar{X}_i$ oraz medianę Med_i Następnie przedstaw na wspólnym wykresie zbiory $\left\{\bar{X}_i : i = 1, \ldots, n\right\}$ oraz $\{\mathrm{Med}_i : i = 1, \ldots, n\}$. Przeanalizuj wpływ liczności próby na zachowanie się średniej oraz mediany z próby. Czy statystyki te wydają się być sensownymi estymatorami parametru położenia a w tym modelu?
2. Dla każdej podpróbki zawierającej $i = 2, \ldots, n$ początkowych elementów próbki wyjściowej wyznacz odchylenie standardowe s_i oraz odchylenie ćwiartkowe $r_i = \mathrm{IQR}(X_i)/2$ (czyli połowę rozstępu międzykwartylowego). Następnie przedstaw na wspólnym wykresie zbiory $\{s_i : i = 2, \ldots, n\}$ oraz $\{r_i : i = 2, \ldots, n\}$. Przeanalizuj wpływ liczności próby na zachowanie się s_i oraz r_i. Czy statystyki te wydają się być sensownymi estymatorami parametru rozproszenia b w tym modelu?

Rozwiązanie. Zaczniemy od porównania estymatorów parametru położenia $a = 0$ rozkładu Cauchy'ego $\mathrm{C}(0, 1)$. W tym celu konieczne będzie stworzenie jakiegoś mechanizmu, który pozwoli nam na rozpatrzenie podpróbek X_i o różnej liczności, a następnie wyznaczenie dla każdej z nich odpowiednich statystyk.

Możemy to zrobić w następujący sposób: dla każdego $i = 1, 2, \ldots, n$ chcemy wyznaczyć średnią i medianę dla podpróbki X_i złożonej z elementów $(Y_1, \ldots, Y_i)$. Informacje te mogą być przechowywane jako elementy ciągów wyjściowych $\bar{X}_i$ oraz Med_i. Do zapisania powyższego algorytmu użyjemy pętli `for`.

Informacja

Pętla `for` w języku R (podobnie jak i w innych językach programowania) służy do cyklicznego wykonywania ciągu wyrażeń. Jej składnia jest następująca:

```
for (Zmienna in Wektor) {
   ... WyrazeniaDoWykonania ...
}
```

Instrukcja `WyrazeniaDoWykonania` zostanie wykonana zadaną liczbę razy, równą `length(WektorWartosci)`, z tym że w każdym kolejnym powtórzeniu `Zmienna` będzie przyjmować kolejną wartość z ciągu `WektorWartosci`, czyli w porządku: `WektorWartosci[1]`, `WektorWartosci[2]`, ...

Działanie pętli `for` najprościej jest pokazać na przykładzie:

```
> for (i in 1:5)            # dla każdego i=1,2,3,4,5
+ {                         # wykonaj:
+    print(i)               # wypisz i
+ }                         # koniec
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Jeżeli do wykonania jest tylko jedna instrukcja to można pominąć nawiasy klamrowe `{·}`, które grupują wiele poleceń w jedno.

```
> for (i in 1:5) print(2^i)
[1] 2
[1] 4
[1] 8
[1] 16
[1] 32
```

W wielu sytuacjach użycie pętli `for` w języku R jest nieuzasadnione, jako mało wydajne. Tam, gdzie się da (i gdzie się potrafi) należy starać się korzystać z innych konstrukcji językowych. Ilustracją takiej sytuacji może być powyższy przykład, który byłoby lepiej zaimplementować następująco:

```
> print(2^(1:5)) # tylko działania na wektorach
[1]  2  4  8 16 32
```

Rozważmy jeszcze jeden przykład. Spróbujmy wyznaczyć wektor liczb $a_1, \ldots, a_n$, gdzie $a_i = \left(1 + \frac{1}{i}\right)^i$, tworzących kolejne przybliżenia liczby e. Uczynimy to zarówno za pomocą pętli `for`, jak i operacji na wektorach. Do porównania obu metod posłużymy się czasami wykonania obu operacji, które uzyskamy wywołując funkcję `system.time()`.

```
> n <- 1000000
> a1 <- numeric(n) # pusty wektor o rozmiarze n
> system.time(      for (i in 1:n) a1[i] <- (1+1/i)^i  ) # sposób I
> system.time(      a2 <- (1+1/(1:n))^(1:n)            ) # sposób II
```

Poszukiwane czasy wykonywania operacji na naszym komputerze, wyrażone w sekundach, wynoszą odpowiednio (kolumna `user`):

```
# sposób I:
   user  system elapsed
  5.045   0.007   5.081
# sposób II:
   user  system elapsed
  0.232   0.013   0.248
```

Wnioski pozostawiamy Czytelnikowi.

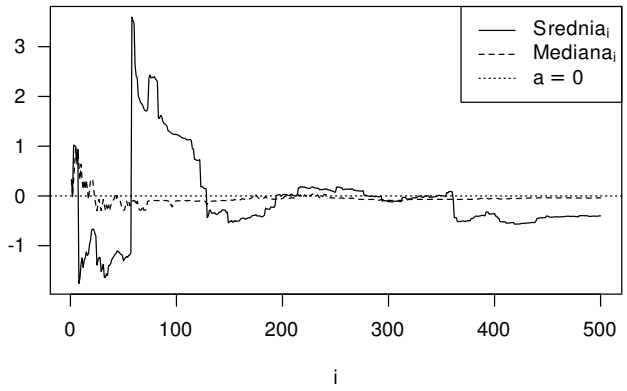
Wróćmy do naszego zadania. Posługując się pętlą `for` rozwiązanie sformułowanego problemu można przedstawić następująco:

```
> n <- 500
> X <- rcauchy(n)   # n-elementowa próba z rozkładu Cauchy'ego
> mn <- numeric(n)  # tu będziemy przechowywać średnie z podpróbek
> md <- numeric(n)  # tu będziemy przechowywać mediany z podpróbek
> for (i in 1:n) {
+    mn[i] <- mean(X[1:i])
+    md[i] <- median(X[1:i])
+ }
```

```
> plot (1:i, mn, type="l", xlab="i", ylab="", las=1)
> lines(1:i, md, lty=2)
> abline(h=0, lty=3)
> legend("topright", expression(Srednia[i], Mediana[i], a==0), lty=1:3)
```

Wynik zamieszczamy na rys. 4.2.

Pamiętajmy, że rozpatrywanie różnych prób losowych prowadzi za każdym razem do innego wykresu. Stąd też warto powtórzyć kilkukrotnie symulacje i przyjrzeć się otrzymanym wykresom. A jakie wnioski płyną z powyższego obrazka? Odpowiedź na to pytanie, jak i teoretyczne uzasadnienie wniosków pozostawiamy Czytelnikowi.



Rys. 4.2

Zajmijmy się teraz porównaniem estymatorów parametru skali $b = 1$ w rozkładzie standaryzowanym Cauchy'ego.

```
> n <- 500
> X <- rcauchy(n)
> s <- numeric(n-1)
> r <- numeric(n-1)
> for (i in 2:n) {
+    s[i-1] <- sd(X[1:i])
+    r[i-1] <- IQR(X[1:i])/2
+ }
```

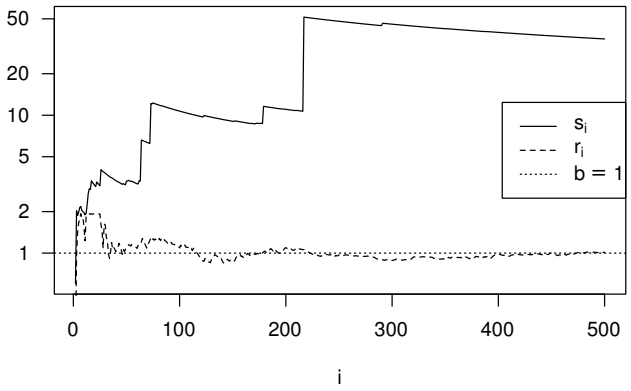
```
> plot (2:i, s, type="l", xlab="i", ylab="", log="y", las=1)
> lines(2:i, r, lty=2)
> abline(h=1, lty=3)
> legend("right", expression(s[i], r[i], b==1), lty=1:3)
```

Wynik zamieszczamy na rys. 4.3. Zwróćmy uwagę na zastosowanie skali logarytmicznej na osi pionowej tego wykresu.

Wyciągnięcie wniosków dotyczących estymacji parametru skali w rozkładzie Cauchy'ego, jakie płyną z zamieszczonego powyżej wykresu (a także uzasadnienie tych wniosków) pozostawiamy Czytelnikowi.

⊡

Zadanie 4.3. Wygeneruj $m = 10000$ n-elementowych próbek ($n = 20$) z rozkładu jednostajnego $U([0, \theta])$, gdzie $\theta > 0$. Porównaj empirycznie obciążenie i błąd średniokwadratowy estymatora momentów i estymatora największej wiarogodności parametru θ w rozkładzie jednostajnym $U([0, \theta])$.



Rys. 4.3

Rozwiązanie. Niech $X_1, \ldots, X_n$ będzie próbą z interesującego nas rozkładu jednostajnego $U([0, \theta])$. Można pokazać, że estymatory parametru θ otrzymany metodą momentów (EMM) ma postać

$$\hat{\theta}_1 = 2\bar{X}, \tag{4.4}$$

podczas gdy estymator największej wiarogodności (ENW) dany jest wzorem

$$\hat{\theta}_2 = X_{n:n}. \tag{4.5}$$

Korzystając ze wzorów (4.2) i (4.3) można wykazać, że obciążenia oraz odchylenia standardowe rozważanych przez nas estymatorów wynoszą, odpowiednio, $b\left(\hat{\theta}_1\right) = 0$, $\mathrm{MSE}\left(\hat{\theta}_1\right) = \frac{\theta^2}{3n}$, $b\left(\hat{\theta}_2\right) = -\frac{\theta}{n+1}$ i $\mathrm{MSE}\left(\hat{\theta}_2\right) = \frac{2\theta^2}{(n+1)(n+2)}$.

Jak widać, estymator największej wiarogodności $\hat{\theta}_2$ obarczony jest mniejszym błędem średniokwadratowym niż estymator metody momentów $\hat{\theta}_1$, ale w przeciwieństwie do tego ostatniego jest obciążony. Nietrudno jednak zauważyć, iż estymator $\hat{\theta}_2$ można w łatwy sposób „poprawić", tak aby mieć zapewnioną nieobciążoność, otrzymując

$$\hat{\theta}_3 = \frac{n+1}{n}\hat{\theta}_2 = \frac{n+1}{n}X_{n:n}. \tag{4.6}$$

Błąd średniokwadratowy tego estymatora wynosi $\mathrm{MSE}\left(\hat{\theta}_3\right) = \frac{\theta^2}{n(n+2)}$.

Przejdźmy teraz do symulacyjnego badania własności podanych wyżej estymatorów. Możemy założyć, bez straty ogólności, że $\theta = 1$. Wygenerujmy 10000 20-elementowych próbek z rozkładu $U([0, 1])$ i porównajemy estymatory parametru θ. Wyniki zapiszemy w macierzy o rozmiarze $m \times 3$, w której każdy wiersz będzie odpowiadał innemu estymatorowi.

```
> m <- 10000
> n <- 20
> theta <- 1
```

```
> wyniki <- matrix(nrow=3, ncol=m,          # tu będą przechowywane wyniki
+    dimnames=list(c("emm", "enw", "enmw"))) # nadajemy nazwy wierszom
>
> for (k in 1:m) { # :-(
+    X <- runif(n, 0, theta) # w każdej iteracji pętli nowa próbka
+    wyniki[1, k] <- 2*mean(X)
+    wyniki[2, k] <- max(X)
+    wyniki[3, k] <- max(X)*(n+1)/n
+ }
```

Rozwiązanie naszego zadania za pomocą pętli `for` nie wydaje się być najszczęśliwsze. Zobaczmy, że przebiega ono według schematu:

```
k razy
{
   wykonaj eksperyment losowy
   zapisz wynik
}
```

Informacja

Bardziej wydajna implementacja powyższej metody może być stworzona za pomocą funkcji `replicate()`. Służy ona np. do wielokrotnego przeprowadzania pewnego eksperymentu losowego i zapisywania wyników w wektorze bądź macierzy wyjściowej. Oto składnia tej funkcji:

```
replicate(IleRazy,
{
   ... różne operacje, np. losowanie próby, działania arytmetyczne...
   wyznacz wynik eksperymentu (wynik: wektor)
}
```

A zatem zamiast pętli `for` posłużymy się następującą konstrukcją:

```
> wyniki <- replicate(m,
+ {
+    X <- runif(n, 0, theta)
+    c(2*mean(X), max(X), max(X)*(n+1)/n) # wynik eksperymentu
+ })
>
> wyniki[,1:5] # zobaczmy wyniki 5 pierwszych eksperymentów
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.8793367 1.0869522 0.8018722 0.8370622 0.8537034
[2,] 0.9793075 0.9790389 0.8779666 0.9820345 0.9541463
```

```
[3,] 1.0282729 1.0279909 0.9218649 1.0311362 1.0018536
```

Dzięki temu rozwiązanie uzyskuje się nieco szybciej (dla $m = 100000, n = 20$ otrzymaliśmy czas 3.5 zamiast 4.3 s). Ponadto, co chyba ważniejsze, kod jest bardziej zwięzły i zrozumiały.

W tym momencie możemy już oszacować obciążenia naszych trzech estymatorów

```
> mean(wyniki[1, ]) - theta
[1] 0.001274957
> mean(wyniki[2, ]) - theta
[1] -0.04682591
> mean(wyniki[3, ]) - theta
[1] 0.000832798
```

oraz ich błędy średniokwadratowe:

```
> var(wyniki[1, ]) + (mean(wyniki[1, ]) - theta)^2
[1] 0.01672622
> var(wyniki[2, ]) + (mean(wyniki[2, ]) - theta)^2
[1] 0.004266179
> var(wyniki[3, ]) + (mean(wyniki[3, ]) - theta)^2
[1] 0.002286743
```

Uzyskane wyniki warto porównać z teoretycznymi wartościami obciążenia i błędów średniokwadratowych, obliczonych przez podstawienie $\theta = 1$ oraz $n = 20$ do podanych wcześniej wzorów:

— obciążenie $b\left(\hat{\theta}_2\right)$:

```
> -theta/(n+1)
[1] -0.04761905
```

— MSE $\left(\hat{\theta}_1\right)$:

```
> (theta^2)/(3*n)
[1] 0.01666667
```

— MSE $\left(\hat{\theta}_2\right)$:

```
> 2*(theta^2)/(n+1)/(n+2)
[1] 0.004329004
```

— MSE $\left(\hat{\theta}_3\right)$:

```
> (theta^2)/n/(n+2)
[1] 0.002272727
```

⊡

Zadanie 4.4. Wygeneruj $m = 50$ próbek n-elementowych ($n = 10$) z rozkładu normalnego $\mathrm{N}(1, 2)$. Przedstaw na jednym wykresie przedziały ufności dla wartości oczekiwa-

nej μ na poziomie ufności 0.95. Ile z nich powinno zawierać wartość $\mu = 1$?

Rozwiązanie. Dla próby $X_1, \ldots, X_n$ z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$, o nieznanych parametrach μ, σ, przedział ufności dla μ na poziomie ufności $1 - \alpha$ ma postać

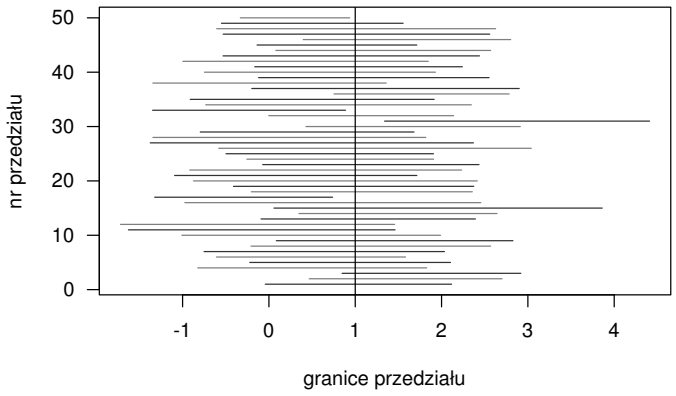
$$\left(\bar{X} - t_{1-\alpha/2}^{[n-1]} \frac{s}{\sqrt{n}}, \bar{X} + t_{1-\alpha/2}^{[n-1]} \frac{s}{\sqrt{n}} \right), \tag{4.7}$$

gdzie $t_{1-\alpha/2}^{[n-1]}$ oznacza kwantyl rzędu $1 - \alpha/2$ rozkładu t-Studenta z $n - 1$ stopniami swobody.

Generujemy m próbek o liczności n z rozkładu normalnego $\mathrm{N}(1, 2)$ i tworzymy dla nich wektor `mn`, zawierający średnie $\bar{X}$ oraz wektor `d`, zawierający wartości $s/\sqrt{n}$:

```
> m  <- 50
> n  <- 10
> mi <- 1
> sigma <- 2
> wyniki <- replicate(m,
+ {
+    X <- rnorm(n, mi, sigma)
+    c(mean(X), sd(X)/sqrt(n))
+ })
> mn <- wyniki[1,]
> d  <- wyniki[2,]
```

Następnie wyznaczamy granice przedziałów ufności i rysujemy uzyskane przedziały na jednym wykresie:




```
> alfa <- 0.05
> q <- qt(1-alfa/2,n-1)
> matplot(rbind(mn-q*d,mn+q*d), rbind(1:m,1:m), type="l", lty=1,
```

```
+    col=c("gray20", "gray50"), las=1, xlab='granice przedziału',
+    ylab='nr przedziału')
> abline(v=mi)
```

Jakie wnioski płyną z powyższego obrazka? Jak się ma frakcja przedziałów pokrywających wartość $\mu = 1$ do założonego poziom ufności? Zachęcamy Czytelnika do chwili refleksji.

□

Zadanie 4.5. Wygeneruj $m = 10000$ próbek n-elementowych ($n = 10$) z rozkładu normalnego. Następnie, zakładając, iż o próbkach wiemy tylko tyle, że pochodzą one z rozkładu normalnego o nieznanych parametrach, wyznacz dla każdej próbki przedział ufności dla wartości oczekiwanej na poziomie ufności 0.95. Porównaj frakcję pokryć przez przedział ufności faktycznej wartości oczekiwanej z założonym poziomem ufności.

Rozwiązanie. Wprowadzamy dane do eksperymentu

```
> m <- 10000
> n <- 10
> alpha <- 0.05
> q <- qt(0.975,9)
```

a następnie generujemy próbki i sprawdzamy, czy przedziały pokrywają teoretyczną wartość oczekiwaną, tzn. $\mu = 0$:

```
> ileWpada <- replicate(m, {
+    X  <- rnorm(n)
+    mn <- mean(X)
+    s  <- sd(X)
+    # to jest wynik eksperymentu:
+    (mn-q*s/sqrt(n) < 0) & (mn+q*s/sqrt(n) > 0)
+    # TRUE (mi wpada do przedziału ufności) albo FALSE
+ })
> ileWpada[1:10] # pierwszych 10 wyników
 [1]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

W końcu obliczamy frakcję pokryć liczby $\mu = 0$ przez wygenerowane przedziały ufności:

```
> sum(ileWpada)/m
[1] 0.9516
```

□

Zadanie 4.6. Średnia cena 50 losowo wybranych podręczników akademickich wyniosła 48.40 zł. Wiadomo, że odchylenie standardowe cen podręczników wynosi 4.75 zł. Wyznacz 95% przedział ufności dla średniej ceny podręcznika akademickiego zakładając, że rozkład cen jest normalny.

Rozwiązanie. Dla rozważanego modelu nie zaimplementowano w R procedury wyznaczania przedziału ufności. Dlatego też musimy zrobić to samodzielnie. Przypomnijmy zatem postać przedziału ufności dla wartości oczekiwanej μ na poziomie ufności $1-\alpha$ dla próby $X_1, \ldots, X_n$ z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$, o znanym odchyleniu standardowym σ:

$$\left(\bar{X} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}, \bar{X} + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right), \tag{4.8}$$

gdzie $z_{1-\alpha/2}$ oznacza kwantyl rzędu $1-\alpha/2$ rozkładu $\mathrm{N}(0,1)$.

W naszym zadaniu: $\sigma = 4.75$, $n = 50$, $\bar{\mathbf{x}} = 48.4$, $1-\alpha = 0.95$. Zatem granice poszukiwanego przedziału ufności wynoszą:

```
> 48.4-qnorm(0.975)*4.75/sqrt(50)
[1] 47.08339
> 48.4+qnorm(0.975)*4.75/sqrt(50)
[1] 49.71661
```

⊡

Zadanie 4.7. Przeprowadzono 18 niezależnych pomiarów temperatury topnienia ołowiu i otrzymano następujące wyniki (w stopniach Celsjusza):

$$330.0, 322.0, 345.0, 328.6, 331.0, 342.0,$$
$$342.4, 340.4, 329.7, 334.0, 326.5, 325.8,$$
$$337.5, 327.3, 322.6, 341.0, 340.0, 333.0.$$

Zakładamy, że temperatura topnienia ołowiu ma rozkład normalny. Wyznacz dwustronny przedział ufności dla wartości oczekiwanej i odchylenia standardowego temperatury topnienia ołowiu na poziomie ufności 0.95.

Rozwiązanie. Do wyznaczenia przedziału ufności dla wartości oczekiwanej μ w rozważanym modelu (tzn. dla próby losowej z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$ o nieznanych parametrach μ i σ) można użyć funkcji `t.test()`:

```
> x <- c(330.0, 322.0, 345.0, 328.6, 331.0, 342.0,
+ 342.4, 340.4, 329.7, 334.0, 326.5, 325.8,
+ 337.5, 327.3, 322.6, 341.0, 340.0, 333.0)
> mean(x)
[1] 333.2667
> t.test(x, conf.level=0.95)$conf.int
[1] 329.6482 336.8851
attr(,"conf.level")
[1] 0.95
```

Informacja

Szczegóły dotyczące funkcji `t.test()` poznamy w części dotyczącej weryfikacji hipotez.

Porównajmy uzyskany wynik z przedziałem ufności wyznaczonym wg wzoru (4.7):

```
> mean(x)-qt(0.975, length(x)-1)*sd(x)/sqrt(length(x))
[1] 329.6482
> mean(x)+qt(0.975, length(x)-1)*sd(x)/sqrt(length(x))
[1] 336.8851
```

Przedział ufności dla odchylenia standardowego σ nie został zaimplementowany w R, a zatem wyznaczamy go bezpośrednio ze wzoru

$$\left(\sqrt{\frac{(n-1)\,s^2}{\chi^2_{1-\alpha/2,n-1}}}, \sqrt{\frac{(n-1)\,s^2}{\chi^2_{\alpha/2,n-1}}} \right), \tag{4.9}$$

otrzymując dolny kraniec szukanego przedziału ufności równy

```
> sqrt((length(x)-1)*var(x) / qchisq(0.975, (length(x)-1)))
[1] 5.460114
```

i górny kraniec przedziału

```
> sqrt((length(x)-1)*var(x) / qchisq(0.025, (length(x)-1)))
[1] 10.90836
```

⊡

Zadanie 4.8. Wygeneruj $m = 10$ próbek $n = 100$-elementowych z rozkładu dwupunktowego $\mathrm{Bern}\,(p)$. Przedstaw na jednym wykresie przedziały ufności dla parametru $p = 0.5$ na poziomie ufności 0.9. Ile z nich powinno zawierać wartość $p = 0.5$?

Rozwiązanie. Asymptotyczny przedział ufności (tzn. skonstruowany przy założeniu, że próbka ma dużą liczność) dla prawdopodobieństwa sukcesu p w rozkładzie dwupunktowym $\mathrm{Bern}\,(p)$, na poziomie ufności $1-\alpha$, ma postać

$$\left(\hat{p} - z_{1-\alpha/2}\sqrt{\frac{\hat{p}\,(1-\hat{p})}{n}}, \hat{p} + z_{1-\alpha/2}\sqrt{\frac{\hat{p}\,(1-\hat{p})}{n}} \right), \tag{4.10}$$

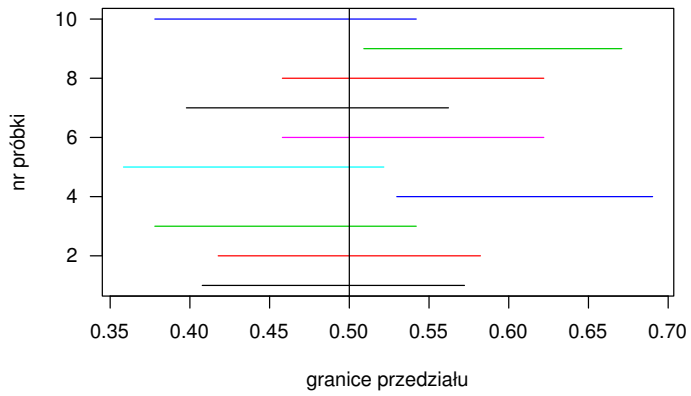
gdzie $z_{1-\alpha/2}$ oznacza kwantyl rzędu $1-\alpha/2$ rozkładu $\mathrm{N}(0,1)$.

Tworzymy wektor pp, zawierający wartości $\hat{p}$ wyliczone dla m próbek oraz wektor d, zawierający wartości $\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$ wyliczone dla m próbek z rozkładu $\mathrm{Bern}(p)$ o licznościach n:

```
> m <- 10
> n <- 100
> p <- 0.5
> alfa <- 0.1
> z <- qnorm(1-alfa/2)
> pp <- rbinom(m, n, p)/n
> d  <- sqrt(pp*(1-pp)/n)
```

Następnie rysujemy otrzymane przedziały na jednym wykresie:

```
> matplot(rbind(pp-z*d,pp+z*d), rbind(1:m,1:m), type="l", lty=1,
+    las=1, xlab='granice przedziału', ylab='nr próbki')
> abline(v=p)
```



⊡

Zadanie 4.9. W sondażu przeprowadzonym przez magazyn „Time" (22 czerwca 1987) 578 spośród 1014 dorosłych respondentów stwierdziło, że dla dobra dzieci lepiej jest, gdy matka nie pracuje poza domem. Wyznacz 95% przedział ufności dla odsetka dorosłych podzielających tę opinię.

Rozwiązanie. Właściwym modelem matematycznym doświadczenia opisanego powyżej jest ciąg niezależnych obserwacji z tego samego rozkładu Bernoulliego $\mathrm{Bern}(p)$, o nieznanym prawdopodobieństwie sukcesu p. Dysponując próbką o dostatecznie dużej liczności (a z takim przypadkiem mamy do czynienia w tym zadaniu) możemy wyznaczyć przedział ufności dla parametru p korzystając ze wzoru (4.10):

```
> p <- 578/1014
> n <- 1014
> p + c(-1,1)*qnorm(0.975)*sqrt(p*(1-p)/n)
[1] 0.5395479 0.6004915
```

A teraz wyznaczmy przedział ufności dla p używając funkcji `prop.test`:

```
> prop.test(578, 1014, conf.level=0.95)$conf.int
[1] 0.5388446 0.6006578
attr(,"conf.level")
[1] 0.95
```

Wynik uzyskany za pomocą funkcji `prop.test` różni się nieco od tego, który otrzymaliśmy implementując samodzielnie znany wzór na asymptotyczny przedział ufności dla p. Spowodowane jest to, w szczególności, stosowaniem przez R korekty na ciągłość. Można ją, oczywiście, wyłączyć z obliczeń, a wówczas otrzymamy:

```
> prop.test(578, 1014, conf.level=0.95, correct=F)$conf.int
[1] 0.5393401 0.6001709
attr(,"conf.level")
[1] 0.95
```

Jak widzimy, nieznaczna różnica wyników nadal się utrzymuje. Wskazuje to, że procedura służąca do wyznaczania przedziału ufności dla proporcji, zaimplementowana w naszym programie, nie korzysta z przybliżenia rozkładem normalnym zgodnie z centralnym twierdzeniem granicznym Moivre'a-Laplace'a. Istotnie, stosowana jest tutaj poprawka zaproponowana przez Wilsona (1927). Zainteresowanego Czytelnika odsyłamy do literatury. ⊡

Zadanie 4.10. Na 12 oddanych niezależnie rzutów kostką otrzymano 3 „szóstki". Wyznacz 95% przedział ufności dla prawdopodobieństwa otrzymania „szóstki" w pojedynczym rzucie kostką.

Rozwiązanie. Tym razem próbka, którą dysponujemy, jest małej liczności, co nie pozwala posłużyć się wzorem na asymptotycznym przedział ufności dla parametru p w rozkładzie dwupunktowym $\mathrm{Bern}\,(p)$. Dysponujemy jednakże przydatną w tym przypadku funkcją `binom.test`:

```
> binom.test(3, 12, conf.level=0.95)$conf.int
[1] 0.05486064 0.57185846
attr(,"conf.level")
[1] 0.95
```

⊡

4.3. Zadania do samodzielnego rozwiązania

Zadanie 4.11. Wygeneruj $n = 500$-elementową próbę $(Y_1, \ldots, Y_n)$ z rozkładu normalnego standardowego.

1. Dla każdej podpróbki zawierającej i początkowych elementów próbki wyjściowej, tzn. dla $X_i = (Y_1, \ldots, Y_i)$, gdzie $i = 1, \ldots, n$, wyznacz średnią $\bar{X}_i$ oraz medianę

Med_i. Następnie przedstaw na wspólnym wykresie zbiory $\left\{\bar{X}_i : i = 1, \ldots, n\right\}$ oraz $\{\mathrm{Med}_i : i = 1, \ldots, n\}$. Przeanalizuj wpływ liczności próby na zachowanie się średniej oraz mediany z próby. Czy statystyki te wydają się być sensownymi estymatorami parametru wartości oczekiwanej w tym modelu?

2. Dla każdej podpróbki zawierającej $i = 2, \ldots, n$ początkowych elementów próbki wyjściowej wyznacz odchylenie standardowe s_i oraz $d_i = \mathrm{IQR}(X_i)/1.35$ (czyli rozstęp międzykwartylowy podzielony przez 1.35). Następnie przedstaw na wspólnym wykresie zbiory $\{s_i : i = 2, \ldots, n\}$ oraz $\{d_i : i = 2, \ldots, n\}$. Przeanalizuj wpływ liczności próby na zachowanie się s_i oraz d_i. Czy statystyki te wydają się być sensownymi estymatorami odchylenia standardowego w tym modelu?

Zadanie 4.12. Na podstawie danych zawartych w pliku `samochody.csv` oszacuj przedziałowo średnie zużycie paliwa i odchylenie standardowe zużycia paliwa samochodów o przyspieszeniu mniejszym niż 20 m/s^2 (wykorzystaj zmienne `mpg` i `przysp`). Załóż, że badana cecha ma rozkład normalny. Przyjmij poziom ufności 0.95.

Zadanie 4.13. Pewien ichtiolog pobrał losową próbę 15 ryb i zmierzył ich długość. Otrzymał następujące wyniki (w mm):

$$92, 88, 85, 82, 89, 86, 81, 66, 75, 61, 78, 76, 91, 82, 82.$$

Zakładając, że rozkład długości ryb badanego gatunku jest normalny, oszacuj przedziałowo na poziomie ufności 0.95 średnią długość ryb tego gatunku.

Zadanie 4.14. Według sondażu przeprowadzonego przez CBOS w związku z dziesięcioleciem członkowstwa Polski w Unii Europejskiej poparcie dla obecności naszego kraju w zjednoczonej Europie osiągnęło rekordowy w historii poziomu 89%. Badanie przeprowadzono na reprezentatywnej próbie losowej 2118 dorosłych mieszkańców Polski. Wyznacz 90% przedział ufności dla odsetka Polaków popierających członkowstwo Polski w UE.

Zadanie 4.15. W celu oszacowania niezawodności pewnego urządzenia dokonano 8 pomiarów czasu bezawaryjnej pracy tego urządzenia i otrzymano następujące wyniki (w godzinach):

$$1034, 2720, 482, 622, 2624, 420, 342, 703.$$

Zakładamy, że czas bezawaryjnej pracy tego urządzenia ma rozkład wykładniczy. Oszacuj prawdopodobieństwo, że dane urządzenie nie ulegnie awarii w ciągu 750 godzin pracy.

Zadanie 4.16. Poniższe obserwacje:

$$3.91, 2.57, 1.18, 2.21, 2.25, 2.52, 4.78, -15.26, 9.93, 6.22,$$
$$9.11, 4.33, 3.10, 4.70, -5.70, 0.39, 0.76, -1.14, -3.63, 3.44;$$

stanowią próbę losową z rozkładu Cauchy'ego $\mathrm{C}(a, b)$. Oszacuj parametry a i b posługując się estymatorami wyznaczonymi metodą kwantyli.

Zadanie 4.17. Poniższe dane przedstawiają zarejestrowaną przez radar drogowy prędkość 10 losowo wybranych pojazdów, jadących pewną autostradą (w kilometrach na godzinę):

$$106, 115, 99, 109, 122, 119, 104, 125, 107, 111.$$

Zakładając normalność rozkładu prędkości, wyznacz liczność próby, potrzebną do wyestymowania średniej prędkości, z dokładnością ± 2 km/h, na poziomie ufności 0.95.

Zadanie 4.18. Jak dużą próbę należy pobrać, aby z maksymalnym błędem $\pm 2\%$ oszacować na poziomie ufności 0.99 procent kierowców nie zapinających pasów bezpieczeństwa? Uwzględnij rezultaty wstępnych badań, z których wynika, że interesująca nas wielkość jest rzędu 16%. Porównaj otrzymaną liczność próby z licznością, jaka byłaby wymagana, gdyby pominąć rezultaty badań wstępnych.

$\star$ **Zadanie 4.19.** Niech $X_1, \ldots, X_n$ oznacza ciąg obserwacji czasów poprawnego działania n urządzeń pracujących niezależnie. Zakładamy, że czas poprawnej pracy każdego urządzenia ma rozkład wykładniczy z nieznanym parametrem θ. Urządzenia te nie są obserwowane w sposób ciągły, lecz kontrola dokonywana jest w dyskretnych chwilach $1, 2, \ldots, k$. Stąd też, de facto, obserwujemy jedynie $Y_1, \ldots, Y_n$, gdzie

$$Y_j = \begin{cases} i & \text{gdy} \quad i-1 < X_j \leqslant i, \quad \text{dla pewnego } i = 1, \ldots, k, \\ k+1 & \text{gdy} \quad X_j > k, \end{cases}$$

przy czym $j = 1, \ldots, n$. Niech $N_i = \#\{j : Y_j = i\}$, $i = 1, \ldots, k+1$. Wyznacz estymator największej wiarogodności parametru θ. Dokonaj obliczeń dla przypadku, gdy $n = 10$, $k = 2$ oraz $N_1 = 5$, $N_2 = 2$ i $N_3 = 3$.

4.4. Wskazówki i odpowiedzi do zadań

Ad zad. 4.12. $\mu \in (27.30534, 29.69748)$, $\sigma \in (6.45733, 8.16175)$.

Ad zad. 4.13. $\mu \in (76.08535, 85.78131)$.

Ad zad. 4.14. $p \in (0.87806, 0.90091)$.

Ad zad. 4.15. Chcemy oszacować prawdopodobieństwo $q = P(T > 750)$. Wystarczy zatem wyestymować parametr rozkładu wykładniczego – np. ENW obliczony dla zadanej próbki wynosi $\hat{\lambda} = \bar{X} = 1118,375$. I stąd $\hat{q} = 0,511$.

Ad zad. 4.16. $\hat{a} = \frac{Q_1+Q_3}{2} = 2.73$, $\hat{b} = \frac{IQR}{2} = 1.97$

Ad zad. 4.17. $n = 90$

Ad zad. 4.18. Wymagana liczność próbki, wyznaczona z uwzględnieniem informacji wcześniejszych badań, wynosi $n_1 = 2237$, podczas gdy bez uwzględnienia tych badań konieczne jest $n_2 = 4161$ obserwacji.

Weryfikacja hipotez: Podstawowe testy parametryczne

5

5.1. Wprowadzenie

5.1.1. Test dla wartości oczekiwanej pojedynczej próby

Niech $X_1, \ldots, X_n$ będzie próbą prostą z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$ o nieznanych parametrach μ i σ. Do weryfikacji hipotez dotyczących wartości oczekiwanej μ postaci $H : \mu = \mu_0$ względem jednej z alternatyw $K : \mu \neq \mu_0$, $K' : \mu < \mu_0$ lub $K'' : \mu > \mu_0$, można użyć funkcji `t.test()`.

Pierwszym argumentem tej funkcji jest wektor obserwacji. Wartość μ_0 podajemy jako drugi argument pisząc: `mu` $= \mu_0$, zaś postać hipotezy alternatywnej jako trzeci argument pisząc: `alternative="two.sided"` (dla K), `"less"` (dla K') lub `"greater"` (dla K''). Domyślnie przeprowadzany jest test dwustronny.

Przykładowo, dla wektora obserwacji x wywołanie funkcji `t.test(x, mu=1)` oznacza testowanie hipotezy $H : \mu = 1$ przeciwko $K : \mu \neq 1$. Z kolei `t.test(x, mu=1, alternative="less")` mówi o tym, że testujemy hipotezę $H : \mu = 1$ przeciwko $K : \mu < 1$.

5.1.2. Test do porównywania wartości oczekiwanych dwóch prób

Rozważmy dwie niezależne próbki proste: $X_1, \ldots, X_{n_1}$ oraz $Y_1, \ldots, Y_{n_2}$ z rozkładów, odpowiednio, $\mathrm{N}(\mu_1, \sigma_1)$ oraz $\mathrm{N}(\mu_2, \sigma_2)$. Za pomocą funkcji `t.test()` możemy weryfikować hipotezy o równości dwóch wartości średnich, tzn. $H : \mu_1 = \mu_2$ względem wybranej alternatywy $K : \mu_1 \neq \mu_2$, $K' : \mu_1 < \mu_2$ lub $K'' : \mu_1 > \mu_2$.

Dwoma pierwszymi argumentami funkcji `t.test()` w tym przypadku są etykiety wektorów obserwacji `x` i `y`. Trzeci argument służy do wyboru hipotezy alternatywnej (domyślnie przeprowadzany jest test dwustronny). Jako kolejny argument funkcji `t.test()` podajemy informację dotyczącą wariancji obu rozkładów: czy wariancje σ_1^2 i σ_2^2 są sobie równe, czy też nic o tym nie wiadomo. Jeśli wariancje są równe, to podajemy: `var.equal=TRUE` i wówczas mamy do czynienia z testem t-Studenta (domyślnie równość wariancji nie jest zakładana).

Przykładowo, dla wektora x zawierającego obserwacje pierwszej próbki i wektora y z obserwacjami drugiej próbki, wywołaniem `t.test(x, y)` zlecamy testowanie hipotezy $H : \mu_1 = \mu_2$ przeciwko $K : \mu_1 \neq \mu_2$, bez zakładania równości wariancji obu rozkładów. Natomiast `t.test(x, y, alternative="less",var.equal=TRUE)` jest poleceniem wywołującym testowanie hipotezy $H : \mu_1 = \mu_2$ przeciwko $K' : \mu_1 < \mu_2$, przy

założeniu $\sigma_1^2 = \sigma_2^2$.

Jeśli zamiast prób niezależnych $X_1, \ldots, X_n$ z rozkładu $\mathrm{N}(\mu_1, \sigma_1)$ oraz $Y_1, \ldots, Y_n$ z rozkładu $\mathrm{N}(\mu_2, \sigma_2)$, mamy obserwacje zależne w parach, to jako argument funkcji `t.test()` podajemy dodatkowo: `paired=TRUE`.

5.1.3. Test do porównywania wariancji dwóch prób

Do weryfikacji hipotezy o równości wariancji $H : \sigma_1^2 = \sigma_2^2$ przeciw wybranej hipotezie alternatywnej: $K : \sigma_1^2 \neq \sigma_2^2$, $K' : \sigma_1^2 < \sigma_2^2$ bądź $K'' : \sigma_1^2 > \sigma_2^2$, dla dwóch niezależnych próbek prostych $X_1, \ldots, X_{n_1}$ z rozkładu $\mathrm{N}(\mu_1, \sigma_1)$ oraz $Y_1, \ldots, Y_{n_2}$ z rozkładu $\mathrm{N}(\mu_2, \sigma_2)$, można posłużyć się funkcją `var.test()`.

Jej pierwszym i drugim argumentem są wektory obserwacji `x` i `y`. Jako trzeci argument podajemy postać hipotezy alternatywnej (domyślnie przeprowadzany jest test dwustronny).

Przykładowo, aby przetestować hipotezę $H : \sigma_1^2 = \sigma_2^2$ przeciwko $K : \sigma_1^2 \neq \sigma_2^2$, na podstawie wektorów obserwacji `x` i `y`, wpisujemy instrukcję `var.test(x, y)`. Z kolei wyrażenie `var.test(x, y, alternative="less")` wywołuje test hipotezy $H : \sigma_1^2 = \sigma_2^2$ przeciwko $K' : \sigma_1^2 < \sigma_2^2$.

5.1.4. Test dla wskaźnika struktury (proporcji) dla pojedynczej próby

Niech $X_1, \ldots, X_n$ oznacza próbke prostą z rozkładu dwupunktowego $\mathrm{Bern}(p)$. Do weryfikacji hipotezy $H : p = p_0$ względem jednej z alternatyw $K : p \neq p_0$, $K' : p < p_0$ bądź $K'' : p > p_0$ można posłużyć się funkcją `binom.test()` lub `prop.test()`, przy czym w drugim przypadku przeprowadzany jest test asymptotyczny.

Pierwszym argumentem obu tych funkcji jest liczba jedynek w naszej próbie (odpowiadająca liczbie umownych sukcesów, tzn. elementów posiadających interesującą nas cechę), a drugim – liczność próby n. Wartość p_0 podajemy jako trzeci argument pisząc: `p` $= p_0$, zaś postać hipotezy alternatywnej jako czwarty argument (domyślnie przeprowadzany jest test dwustronny).

Dla przykładu, polecenie `prop.test(4, 100, p=0.05)` oznacza test hipotezy $H : p = 0.05$ przeciwko $K : p \neq 0.05$, na podstawie 100 elementowej próbki, w której zaobserwowano 4 elementy mające interesującą nas cechę. Natomiast instrukcja `prop.test(4, 100, p=0.05, alternative="less")` wywołuje test do weryfikacji hipotezy $H : p = 0.05$ przeciwko $K' : p < 0.05$.

5.1.5. Test do porównania wskaźników struktury dwóch prób

Niech $X_1, \ldots, X_{n_1}$ i $Y_1, \ldots, Y_{n_2}$ będą niezależnymi próbkami prostymi z rozkładów, odpowiednio, $\mathrm{Bern}(p_1)$ oraz $\mathrm{Bern}(p_2)$. Do weryfikacji hipotezy zerowej $H : p_1 = p_2$ przeciw wybranej alternatywie $K : p_1 \neq p_2$, $K' : p_1 < p_2$ bądź $K'' : p_1 > p_2$, służy funkcja `prop.test()`, przeprowadzająca test asymptotyczny dla dwóch wskaźników struktury.

Jako pierwszy argument tej funkcji należy podać wektor (k_1, k_2), gdzie k_i oznacza liczbę jedynek (elementów posiadających interesującą nas cechę) w i-tej próbce ($i = 1, 2$). Drugim argumentem jest wektor liczności próbek (n_1, n_2), natomiast trzecim – postać hipotezy alternatywnej (domyślnie przeprowadzany jest test dwustronny).

Na przykład, chcąc zweryfikować hipotezę $H : p_1 = p_2$ przeciwko $K : p_1 \neq p_2$, na podstawie próbek 100 i 120 elementowej, w których zaobserwowano, odpowiednio, 4 i 6 elementów wyróżnionych, należy wywołać funkcję `prop.test(c(4, 6), c(100, 120))`. Natomiast wywołując `prop.test(c(4, 6), c(100, 120), alternative= "less")` testujemy – dla analogicznego wyniku, jak poprzednio – hipotezę $H : p_1 = p_2$ przeciwko $K' : p_1 < p_2$.

5.2. Zadania rozwiązane

Zadanie 5.1. Nominalna waga netto kawy sprzedawanej w opakowaniu szklanym powinna wynosić 150 g. Występuje jednakże duża zmienność wagi. Istotnie, próba losowa siedmiu słoiczków kawy konkretnej marki sprzedawanej w sieci handlowej Żuczek wykazała następujące wagi netto (w gramach):

$$142, 151, 148, 151, 145, 150, 141.$$

Zakładając normalność rozkładu wagi, przetestuj hipotezę głoszącą, że przeciętna waga netto tej marki kawy wynosi faktycznie 150 g. Przyjmij poziom istotności $\alpha = 0.05$.

Rozwiązanie. Obserwacje, którymi dysponujemy, są realizacją próby $X_1, \ldots, X_n$ z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$ o dwóch nieznanych parametrach μ oraz σ. Nasz problem decyzyjny sprowadza się do weryfikacji hipotezy zerowej mówiącej, iż przeciętna waga netto słoiczka kawy wynosi 150 g, czyli $H : \mu = 150$, wobec alternatywy głoszącej, że średnia waga różni sie istotnie od wagi nominalnej, tzn. $K : \mu \neq 150$. Przyjęty model matematyczny upoważnia nas do posłużenia się testem t. Jest on zaimplementowany w pakiecie R jako funkcja `t.test()`:

```
> kawa <- c(142, 151, 148, 151, 145, 150, 141)
> mean(kawa) # zobaczmy, jaka jest średnia waga kawy w pobranej próbce
[1] 146.8571
> t.test(kawa, mu=150)

One Sample t-test

data:  kawa
t = -1.9704, df = 6, p-value = 0.0963
alternative hypothesis: true mean is not equal to 150
95 percent confidence interval:
```

```
 142.9542 150.7601
sample estimates:
mean of x
 146.8571
```

Z otrzymanego wydruku możemy odczytać m.in. wartość statystyki testowej (która w rozważanym przypadku wynosi $t = -1.9704$) oraz p-wartość $= 0.0963$, która jest wielkością kluczową do podjęcia decyzji. Ponieważ w zadaniu przyjęto poziom istotności testu $\alpha = 0.05$, a otrzymana p-wartość $> \alpha$, zatem nie ma podstaw do odrzucenia hipotezy zerowej. Oznacza to, że średnia waga słoiczka kawy nie różni się istotnie od 150 g.

Informacja

Przypomnijmy, że funkcja `t.test()` zwraca nam m.in. granice przedziału ufności dla wartości średniej wagi kawy (przy domyślnym poziomie ufności 95%). Zwróćmy uwagę, że dysponując tym przedziałem ufności jesteśmy również w stanie zweryfikować rozważaną hipotezę zerową. Dlaczego? Albowiem istnieje wzajemnie równoważny związek między przedziałami ufności na poziomie ufności $1 - \alpha$ i testami na poziomie istotności α. A jaki to związek i co z niego wynika – odpowiedzi na to pytanie zostawiamy Czytelnikowi (odsyłając go, w razie potrzeby, do literatury przedmiotu).

□

Zadanie 5.2. Wytrzymałość na ciśnienie wewnętrzne jest ważną charakterystyką jakościową szkła butelek. Pewna rozlewnia chce zamówić butelki, których średnia wytrzymałość przewyższa 1.20 N/mm^2. Na podstawie dotychczasowych doświadczeń wiadomo, że rozkład wytrzymałości jest normalny z odchyleniem standardowym 0.07 N/mm^2. Pobrano próbę losową 20 butelek, które następnie umieszczono w maszynie hydrostatycznej, zwiększając ciśnienie aż do zniszczenia butelki i otrzymano następujące wyniki (w N/mm^2):

1.36, 1.14, 1.27, 1.15, 1.20, 1.29, 1.27, 1.18, 1.23, 1.36,
1.38, 1.37, 1.30, 1.21, 1.33, 1.28, 1.32, 1.29, 1.33, 1.25.

Na poziomie istotności 0.04 stwierdź, czy dana partia butelek spełnia postawione wymagania jakościowe.

Rozwiązanie. Mamy tutaj do czynienia z próbą $X_1, \ldots, X_n$ z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$ o nieznanym parametrze μ, ale znanym odchyleniu standardowym $\sigma = 0.07$.

Ponieważ nasz zleceniodawca chce być przekonany o tym, że zamówiona partia butelek wytrzyma ciśnienie wewnętrzne przewyższające 1.20 N/mm^2, naszym celem będzie przetestowanie hipotezy zerowej $H_0 : \mu = 1.2$ przeciw alternatywie $K : \mu > 1.2$. Skorzystamy tu z tzw. testu z, ponieważ ma on większą moc niż test t (dysponujemy

bowiem informacją o rozproszeniu rozkładu). Niestety, ten test nie został on zaimplementowany w R, a więc przeprowadzimy go „ręcznie".

Jak wiadomo (a tych Czytelników, którzy o tym nie wiedzą, odsyłamy do literatury), hipotezę zerową $H_0 : \mu = \mu_0$ w teście z przy tzw. alternatywie prawostronnej $K : \mu > \mu_0$ odrzucamy, gdy statystyka testowa

$$z = \frac{\overline{X} - \mu_0}{\sigma}\sqrt{n} \tag{5.1}$$

przyjmie wartość z obszaru krytycznego: $W_\alpha = [z_{1-\alpha}, +\infty)$, gdzie $z_{1-\alpha}$ jest kwantylem rzędu $1 - \alpha$ rozkładu normalnego standardowego $\mathrm{N}(0, 1)$.

W naszym zadaniu: $n = 20$, $\alpha = 0.04$, $\mu_0 = 1.2$ i $\sigma = 0.07$. Podstawiamy te dane do wzoru (5.1) i wyznaczając średnią z naszej próby, w rezultacie otrzymujemy wartość statystyki testowej:

```
> x <- c(1.36, 1.14, 1.27, 1.15, 1.20, 1.29, 1.27, 1.18, 1.23, 1.36,
+ 1.38, 1.37, 1.30, 1.21, 1.33, 1.28, 1.32, 1.29, 1.33, 1.25)
> mu0 <- 1.2
> sigma <- 0.07
> (z <- (mean(x)-mu0)/sigma*sqrt(length(x)))
[1] 4.823518
```

Aby wyznaczyć obszar krytyczny potrzebna nam wartość odpowiedniego kwantyla, która wynosi:

```
> qnorm(0.96)
[1] 1.750686
```

Ponieważ wartość statystyki testowej z należy do przedziału krytycznego $W_{0.04} = [1.75069, +\infty)$, odrzucamy hipotezę zerową, co pozwala nam stwierdzić, iż wytrzymałość na ciśnienie badanych butelek istotnie ($\alpha = 0.04$) przewyższa 1.2 N/mm^2. ⊡

Zadanie 5.3. Wylosowana niezależnie z partii żarówek 12-elementowa próba dała następujące wyniki pomiaru czasu świecenia żarówek aż do przepalenia (w godzinach): 2852, 3060, 2631, 2819, 2805, 2835, 2955, 2595, 2690, 2723, 2815, 2914.

a) Czy średni czas świecenia żarówek jest istotnie krótszy od 2900 godzin? Przyjmij poziom istotności $\alpha = 0.05$.
b) Wyznacz 97% przedział ufności dla średniego czasu świecenia żarówek.

Rozwiązanie. W przeciwieństwie do dwóch poprzednich zadań, w niniejszym nie mamy informacji dotyczącej rozkładu, z którego pochodzi próba. Gdyby czas świecenia żarówek miał rozkład normalny, moglibyśmy posłużyć się znanym nam testem t. Sprawdzimy zatem, czy można przyjąć założenie o jego normalności, tzn. przetestujemy hipotezę H_0 : *rozkład próby jest normalny* wobec alternatywy K : *rozkład ten nie jest normalny*. Posłużymy się w tym celu testem Shapiro-Wilka (więcej informacji na temat testowania zgodności z danym rozkładem znajdzie Czytelnik w następnym rozdziale):

```
> zar <- c(2852, 3060, 2631, 2819, 2805, 2835, 2955,
+          2595, 2690, 2723, 2815, 2914)
> shapiro.test(zar)

Shapiro-Wilk normality test

data:  zar
W = 0.9747, p-value = 0.9532
```

Ponieważ p-wartość obliczona dla statystyki testowej $W = 0.9747$ wynosi 0.9532, a więc jest wystarczająco duża (w szczególności większa od $\alpha = 0.05$), to nie mamy podstaw do odrzucenia hipotezy zerowej na zadanym poziomie istotności, a więc możemy przyjąć, że nasza próba pochodzi z rozkładu normalnego.

Dzięki temu do wyznaczenia jednostronnego przedziału ufności dla μ oraz zweryfikowania hipotezy $H : \mu = 2900$ przeciw $K : \mu < 2900$, możemy teraz użyć funkcji `t.test()` (gdyby założenie o normalności nie było spełnione, powinniśmy posłużyć się jakimś testem nieparametrycznym, zob. nast. rozdział):

```
> t.test(zar, conf.level=0.97, mu=2900, alternative="less")

One Sample t-test

data:  zar
t = -2.3855, df = 11, p-value = 0.01807
alternative hypothesis: true mean is less than 2900
97 percent confidence interval:
     -Inf 2888.819
sample estimates:
mean of x
 2807.833
```

Wartość statystyki testowej t wynosi -2.3855. A ponieważ p-wartość $= 0.0181 < \alpha = 0.05$, więc odrzucamy hipotezę zerową na rzecz alternatywnej, co oznacza, że średni czas świecenia żarówek jest istotnie krótszy niż 2900 godzin. Na marginesie, zauważmy, że podanie parametru `conf.level` nie wpływa na wynik testu. Służy on tylko do określania poziomu ufności dla przedziału ufności. Natomiast parametr `alternative` determinuje zarówno postać hipotezy alternatywnej, jak i rodzaj przedziału ufności. W naszym przypadku, z uwagi na rozważaną jednostronną hipotezę alternatywną, uzyskaliśmy również jednostronny przedział ufności.

Gdybyśmy chcieli otrzymać dwustronny przedział ufności dla przeciętnego czasu świecenia żarówki, musielibyśmy ponownie skorzystać z funkcji `t.test()` pomijając parametr `alternative` (co odpowiada domyślnie opcji „`two.sided`").

```
> t.test(zar, conf.level=0.97, mu=2900)

One Sample t-test
```

```
data:  zar
t = -2.3855, df = 11, p-value = 0.03615
alternative hypothesis: true mean is not equal to 2900
97 percent confidence interval:
 2711.605 2904.062
sample estimates:
mean of x
 2807.833
```

Zatem 97% przedział ufności dla średniego czasu świecenia żarówek (w godzinach) ma następującą postać: $[2711.605, 2904.062]$.

□

Zadanie 5.4. W stopie metalicznym pewnego typu zastosowano dwa różne pierwiastki utwardzające. Wyniki pomiarów twardości stopów utwardzanych obiema metodami wyglądają następująco:

Metoda I	$145, 150, 153, 148, 141, 152, 146, 154, 139, 148$
Metoda II	$152, 150, 147, 155, 140, 146, 158, 152, 151, 143, 153$

Przyjmuje się, że twardość ma rozkład normalny oraz że wariancje dla obu metod są równe. Czy na podstawie przeprowadzonych pomiarów można stwierdzić, że średnia twardość stopu utwardzanego drugą metodą przewyższa średnią twardość stopu utwardzanego pierwszą metodą? Przyjmij poziom istotności $\alpha = 0.05$.

Rozwiązanie. Mamy tutaj do czynienia z dwoma niezależnymi próbami prostymi: $X_1, \ldots, X_{n_1}$ z rozkładu $\mathrm{N}(\mu_1, \sigma_1)$ oraz $Y_1, \ldots, Y_{n_2}$ z rozkładu $\mathrm{N}(\mu_2, \sigma_2)$, przy czym wiadomo, że $\sigma_1^2 = \sigma_2^2$.

Weryfikujemy $H : \mu_1 = \mu_2$ przeciw $K : \mu_1 < \mu_2$. Użyjemy testu t dla dwóch prób niezależnych o takich samych wariancjach. W praktyce oznacza to posłużenie się znaną nam funkcją `t.test()` z odpowiednio zadanymi parametrami:

```
> m1 <- c(145,150,153,148,141,152,146,154,139,148)
> m2 <- c(152,150,147,155,140,146,158,152,151,143,153)
> t.test(m1, m2, alternative="less", var.equal=TRUE)

Two Sample t-test

data:  m1 and m2
t = -0.9466, df = 19, p-value = 0.1779
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf 1.758418
sample estimates:
mean of x mean of y
 147.6000  149.7273
```

Nietrudno zauważyć, że nie ma podstaw do twierdzenia, iż średnia twardość stopu utwardzanego drugą metodą przewyższa średnią twardość stopu utwardzanego pierwszą metodą. Uzasadnienie tej decyzji pozostawiamy Czytelnikowi.

My natomiast sprawdźmy, jakie uzyskalibyśmy wyniki, gdybyśmy nie wiedzieli o równości wariancji obu rozkładów. W tym celu wystarczy zmienić w funkcji `t.test()` wartość argumentu dotyczącego wariancji:

```
> t.test(m1, m2, alternative="less") # domyślnie: var.equal=FALSE

Welch Two Sample t-test

data:  m1 and m2
t = -0.9496, df = 18.973, p-value = 0.1771
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf 1.746503
sample estimates:
mean of x mean of y
 147.6000  149.7273
```

W tym przypadku także nie mielibyśmy podstaw do odrzucenia hipotezy zerowej.

Informacja

Jeżeli założenie o równości wariancji, tzn. $\sigma_1^2 = \sigma_2^2$, jest uzasadnione, to warto je uwzględnić dobierając właściwy test (pierwszy z pokazanych w tym przykładzie), który jest mocniejszy od drugiego z rozważanych testów, nie wymagających tego założenia.

□

Zadanie 5.5. Spośród pracowników pewnego przedsiębiorstwa wylosowano niezależnie 15 pracowników fizycznych i 9 pracowników umysłowych. Otrzymano następujące dane dotyczące stażu pracy (w latach):

Pracownicy umysłowi	$14, 17, 7, 33, 2, 24, 26, 22, 12$
Pracownicy fizyczni	$13, 15, 3, 2, 25, 4, 1, 18, 6, 9, 20, 11, 5, 1, 7$

Wiadomo, że rozkład stażu pracy w przedsiębiorstwie jest normalny. Zweryfikuj hipotezę głoszącą, że średni staż pracy pracowników fizycznych jest istotnie krótszy niż staż pracy pracowników umysłowych. Przyjmij poziom istotności $\alpha = 0.05$.

Rozwiązanie. Mamy dwie niezależne próbki proste: $X_1, \ldots, X_{n_1}$ z rozkładu $\mathrm{N}(\mu_1, \sigma_1)$ oraz $Y_1, \ldots, Y_{n_2}$ z rozkładu $\mathrm{N}(\mu_2, \sigma_2)$. Naszym celem jest weryfikacja hipotezy zerowej $H : \mu_1 = \mu_2$ przeciw $K : \mu_1 > \mu_2$. Zastosujemy tu, podobnie jak w poprzednim zadaniu, test t-Studenta dla dwóch prób.

Zanim jednak to uczynimy, sprawdzimy wpierw, czy można przyjąć założenie o równości wariancji obu rozkładów. Do weryfikacji hipotezy $H : \sigma_1^2 = \sigma_2^2$ przeciw $K : \sigma_1^2 \neq \sigma_2^2$ użyjemy funkcji `var.test()`:

```
> um <-  c(14,17,7,33,2,24,26,22,12)
> fiz <- c(13,15,3,2,25,4,1,18,6,9,20,11,5,1,7)
> var.test(um, fiz)

F test to compare two variances

data:  um and fiz
F = 1.725, num df = 8, denom df = 14, p-value = 0.3557
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5250833 7.1238787
sample estimates:
ratio of variances
           1.72505
```

Zatem można przyjąć, że nie ma podstaw do odrzucenia hipotezy o równości wariancji. Założenie to, tzn. $\sigma_1^2 = \sigma_2^2$, uwzględniamy w wywołaniu funkcji `t.test()`:

```
> t.test(um, fiz, alternative="greater", var.equal=TRUE)

Two Sample t-test

data:  um and fiz
t = 2.2937, df = 22, p-value = 0.01587
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 2.038754      Inf
sample estimates:
mean of x mean of y
17.444444  9.333333
```

Przy założonym poziomie istotności 0.05 stwierdzamy, że średni staż pracy pracowników fizycznych jest istotnie krótszy od stażu pracy pracowników umysłowych. ⊡

Zadanie 5.6. Grupę 10 dzieci poddano pewnemu badaniu pamięci. Po pewnym czasie, w którym dzieci wykonywały w domu ćwiczenia usprawniające pamięć, poddano je ponownemu badaniu. Na podstawie wyników zamieszczonych w tabeli stwierdź, czy zaproponowane ćwiczenia w istotny sposób usprawniają pamięć. Przyjmij poziom istotności równy 5%.

Dziecko	1	2	3	4	5	6	7	8	9	10
Wynik przed	27	21	34	24	30	27	33	31	22	27
Wynik po	29	32	29	27	31	26	35	30	29	28

Rozwiązanie. Tym razem mamy do czynienia z dwiema próbkami prostymi $X_1, \ldots, X_n$ oraz $Y_1, \ldots, Y_n$, odpowiadającymi wynikom poszczególnych dzieci przed i po ćwiczeniach, które są zależne parami. Istotnie, X_i oraz Y_i są obserwacjami tego samego i-tego dziecka, odpowiednio, przed i po ćwiczeniach mających usprawnić pamięć. Ta sytuacja różni się zatem od rozważanych w poprzednich zadaniach, w których mieliśmy do czynienia z próbkami niezależnymi.

Zakładając, że μ_1 i μ_2 oznaczają średni wynik osiągnięty podczas badania, odpowiednio, przed i po odbyciu ćwiczeń, nasz problem sprowadzi się do weryfikacji hipotezy zerowej $H_0 : \mu_1 = \mu_2$ względem $K : \mu_1 < \mu_2$. Faktycznie, odrzucenie hipotezy zerowej na rzecz tak sformułowanej alternatywy świadczyłoby o poprawie pamięci, a wiec o skuteczności ćwiczeń.

Załóżmy chwilowo, że nasze próbki pochodzą z rokładów normalnych. W celu weryfikacji postawionej hipotezy posłużymy się ponownie funkcją `t.test()`, z odpowiednio ustalonym argumentem:

```
> przed <- c(27,21,34,24,30,27,33,31,22,27)
> po    <- c(29,32,29,27,31,26,35,30,29,28)
> t.test(przed, po, alternative="less", paired=TRUE)

Paired t-test

data:  przed and po
t = -1.4302, df = 9, p-value = 0.09322
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf 0.5634468
sample estimates:
mean of the differences
                     -2
```

Porównując p-wartość z przyjętym poziomem istotności $\alpha = 0.05$ stwierdzamy brak podstaw do odrzucenia H_0, co oznacza, że zaproponowane ćwiczenia nie wydają się istotnie usprawniać pamięci dzieci.

Informacja

Zauważmy, że w celu weryfikacji naszej hipotezy możemy również użyć testu t dla jednej próby utworzonej jako różnica obserwacji dwóch próbek:

```
> t.test(przed-po, mu=0, alternative="less")

One Sample t-test

data:  przed - po
t = -1.4302, df = 9, p-value = 0.09322
alternative hypothesis: true mean is less than 0
95 percent confidence interval:
      -Inf 0.5634468
sample estimates:
mean of x
       -2
```

Zwróćmy uwagę, że stosowany przez nas test t zakłada normalność rozkładów. Pominęliśmy ten problem przedstawiając sposób postępowania z próbkami zależnymi parami. Teraz jednak sprawdzimy, czy faktycznie byliśmy uprawnieni do posłużenia się testem t. Dokładniej, zweryfikujemy hipotezę o normalności rokładu różnic obserwacji w parach. Posłużymy sie w tym celu testem Shapiro-Wilka:

```
> shapiro.test(przed-po) # test normalności dla różnic w parach

Shapiro-Wilk normality test

data:  przed - po
W = 0.9355, p-value = 0.504
```

⊡

Zadanie 5.7. W czasie poprawnej pracy maszyny frakcja wytwarzanych przez nią elementów wadliwych nie powinna przekraczać 4%. Jeżeli liczba ta będzie większa, wówczas należy podjąć czynności mające na celu wyregulowanie procesu produkcji. Pracownik zajmujący się kontrolą jakości pobrał próbkę losową 200 elementów i znalazł w niej 14 elementów wadliwych. Czy zaistniała sytuacja wymaga wyregulowania procesu produkcji? Zweryfikować odpowiednią hipotezę na poziomie istotności 0.05.

Rozwiązanie. Mamy próbę $X_1, \ldots, X_n$ z rozkładu dwupunktowego $\mathrm{Bern}(p)$, w którym parametr p interpretujemy jako wadliwość procesu produkcji. Aby móc podjąć decyzję, czy wymagane jest podjęcie czynności mających na celu wyregulowanie procesu, zweryfikujemy hipotezę $H_0 : p = 0.04$ przeciw $K > 0.04$. Możemy to zrobić kilkoma metodami. Po pierwsze, możemy posłużyć się dokładnym testem, zaimplementowanym w funkcji `binom.test()`:

```
> binom.test(14, 200, p=0.04, alternative="greater")
```

```
Exact binomial test

data:  14 and 200
number of successes = 14, number of trials = 200, p-value =
0.03121
alternative hypothesis: true probability of success is greater than 0.04
95 percent confidence interval:
 0.04281265 1.00000000
sample estimates:
probability of success
                  0.07
```

Z uwagi na to, iż dysponujemy stosunkowo liczną próbką ($n = 200$), możemy skorzystać z testu asymptotycznego. Dostęp do tego testu uzyskujemy poprzez funkcję `prop.test()`. Zaimplementowana wersja domyślna tego testu wykorzystuje tzw. korektę ciągłości:

```
> prop.test (14, 200, p=0.04, alternative="greater")

1-sample proportions test with continuity correction

data:  14 out of 200, null probability 0.04
X-squared = 3.9388, df = 1, p-value = 0.02359
alternative hypothesis: true p is greater than 0.04
95 percent confidence interval:
 0.04371856 1.00000000
sample estimates:
   p
0.07
```

Ze wspomnianej korekty ciągłości możemy zrezygnować, podając odpowiedni argument funkcji `prop.test()`:

```
> prop.test (14, 200, p=0.04, alternative="greater", correct=FALSE)

1-sample proportions test without continuity correction

data:  14 out of 200, null probability 0.04
X-squared = 4.6875, df = 1, p-value = 0.01519
alternative hypothesis: true p is greater than 0.04
95 percent confidence interval:
 0.04570864 1.00000000
sample estimates:
   p
0.07
```

Wyniki wszystkich testów wskazują, że należy odrzucić hipotezę zerową na poziomie istotności 0.05, co oznacza, że proces produkcji został rozregulowany i należy podjąć stosowne działania naprawcze. ⊡

Zadanie 5.8. Dawno, dawno temu... (no może nie tak dawno), obowiązywały egzaminy wstępne na wyższe uczelnie. Otóż pewnego roku 455 spośród 700 absolwentów techników i 517 spośród 1320 absolwentów liceów nie zdało egzaminu wstępnego z matematyki na Politechnikę. Czy na podstawie powyższych wyników można stwierdzić, że absolwenci techników byli słabiej przygotowani do egzaminu z matematyki niż absolwenci liceów?

Rozwiązanie. Modelem matematyczny rozważanej sytuacji są dwie niezależne próbki proste $X_1, \ldots, X_{n_1}$ i $Y_1, \ldots, Y_{n_2}$ z rozkładów dwupunktowych, odpowiednio, $\mathrm{Bern}(p_1)$ oraz $\mathrm{Bern}(p_2)$. Parametry p_1 i p_2 interpretujemy jako prawdopodobieństwa nie zdania egzaminu wstępnego z matematyki przez absolwenta, odpowiednio, technikum i liceum.

Naszym zadaniem jest weryfikacja hipotezy zerowej $H : p_1 = p_2$ przy alternatywie $K : p_1 > p_2$. Posłużymy się w tym celu testem dla dwu proporcji, zaimplementowanym w funkcji `prop.test()`:

```
> prop.test(c(455,517), c(700,1320), alternative="greater")

2-sample test for equality of proportions with continuity
correction

data:  c(455, 517) out of c(700, 1320)
X-squared = 121.2477, df = 1, p-value < 2.2e-16
alternative hypothesis: greater
95 percent confidence interval:
 0.2202584 1.0000000
sample estimates:
   prop 1    prop 2
0.6500000 0.3916667
```

Zwróćmy uwagę, że w zadaniu nie wskazano poziomu istotności. Nie jest to jednak przeszkodą w podjęciu decyzji, bowiem p-wartość przeprowadzonego testu jest mniejsza niż jakikolwiek, dający sie uzasadnić, poziom istotności. Tym samym decyzja wydaje się oczywista: faktycznie, absolwenci techników okazali się słabiej przygotowani z matematyki niż absolwenci liceów.

⊡

Zadanie 5.9. Niech $X_1, \ldots, X_n$ będzie próbą niezależnych obserwacji z tego samego rozkładu dwupunktowego $\mathrm{Bern}(\theta)$, gdzie $\theta \in (0, 1)$.

1. Przyjmując, że $n = 20$, podaj postać najmocniejszego testu na poziomie istotności $\alpha = 0.02$ do weryfikacji hipotezy $H : \theta = 0.2$ wobec hipotezy alternatywnej $K : \theta = 0.3$.
2. Oblicz rozmiar tego testu.
3. Wyznacz metodą symulacyjną rozmiar rozważanego testu.
4. Podaj postać testu zrandomizowanego rozmiaru 0.02.

5. Zbadaj metodą symulacyjną, czy rozmiar testu zrandomizowanego wynosi faktycznie 0.02.

Rozwiązanie. Korzystając z lematu Neymana-Pearsona można wykazać, że najmocniejszy test na zadanym poziomie istotności α do weryfikacji hipotezy $H : \theta = 0.2$ wobec hipotezy alternatywnej $K : \theta = 0.3$ ma postać

$$\varphi(X_1, \ldots, X_n) = \begin{cases} 1 & \text{gdy} \quad T(X_1, \ldots, X_n) > c, \\ 0 & \text{gdy} \quad T(X_1, \ldots, X_n) \leqslant c, \end{cases} \tag{5.2}$$

gdzie

$$T(X_1, \ldots, X_n) = \sum_{i=1}^{n} X_i. \tag{5.3}$$

W rozważanym przez nas przypadku, tzn. dla $n = 10$ i $\theta = 0.2$ statystyka (5.3) ma rozkład dwumianowy, a dokładniej: $T(X_1, \ldots, X_{10}) = \sum_{i=1}^{10} X_i \sim \mathrm{Bin}(10, 0.2)$.

Przy prawdziwości hipotezy zerowej stała c występująca w teście (5.2) powinna spełniać warunek

$$P_H\left(T(X_1, \ldots, X_n) > c\right) = \alpha. \tag{5.4}$$

Ponieważ mamy do czynienia z rozkładem dyskretnym, może się okazać, że przy zadanym poziomie istotności nie istnieje stała c spełniająca warunek (5.4). Wówczas zastąpimy go nieco słabszym warunkiem postaci

$$P_H\left(T(X_1, \ldots, X_n) > c\right) \leqslant \alpha. \tag{5.5}$$

W celu wyznaczenia stałej c, przyjrzyjmy się wartościom $\mathrm{P}(Y > y)$, gdzie $Y \sim \mathrm{Bin}(10, 0.2)$:

```
> y <- 0:20
> p <- 1-pbinom(y, 20, 0.2)
> data.frame(p=round(p, 10), row.names=y)
              p
0  0.9884707850
1  0.9308247097
2  0.7939152811
3  0.5885511380
4  0.3703517361
5  0.1957922145
6  0.0866925136
7  0.0321426631
8  0.0099817863
9  0.0025948274
10 0.0005634137
11 0.0001017288
12 0.0000151628
```

```
13 0.0000018450
14 0.0000001803
15 0.0000000138
16 0.0000000008
17 0.0000000000
18 0.0000000000
19 0.0000000000
20 0.0000000000
```

W istocie szukamy największego c, dla którego zachodzi warunek (5.5). Nietrudno zauważyć, że $c = 8$. Stąd szukany test ma więc postać

$$\varphi(X_1, \ldots, X_n) = \begin{cases} 1 & \text{gdy} \quad T(X_1, \ldots, X_n) > 8, \\ 0 & \text{gdy} \quad T(X_1, \ldots, X_n) \leqslant 8, \end{cases} \tag{5.6}$$

Ponieważ, jak wynika z zamieszczonego wyżej wydruku, $P_H\left(T(X_1, \ldots, X_n) > 8\right) = 0.0099818$, zatem rozmiar naszego testu wynosi właśnie 0.0099818.

Za chwilę przekonamy się, czy w praktyce uda się otrzymać tę wartość rozmiaru testu, obliczoną analitycznie. W tym celu przeprowadzimy eksperyment symulacyjny i wyznaczymy przybliżoną wartość rozmiaru testu jako iloraz liczby przypadków, w których wartość statystyki testowej wpada do obszaru krytycznego oraz liczby przeprowadzonych doświadczeń:

```
> n <- 10000 # liczba replikacji eksperymentu
> wyniki <- replicate(n, {
+    (rbinom(1, 20, 0.2) > 8)
+ })
> (rozmiar <- sum(wyniki)/n)
[1] 0.0102
```

Chcąc otrzymać test o żądanym rozmiarze 0.02 musimy zdecydować się na randomizację. W rozważanej przez nas sytuacji test zrandomizowany będzie miał następującą postać:

$$\varphi(X_1, \ldots, X_n) = \begin{cases} 1 & \text{gdy} \quad \sum_{i=1}^{10} X_i > 8, \\ \gamma & \text{gdy} \quad \sum_{i=1}^{10} X_i = 8, \\ 0 & \text{gdy} \quad \sum_{i=1}^{10} X_i < 8, \end{cases} \tag{5.7}$$

gdzie stałą γ wyznaczamy z równania:

$$\alpha = \mathrm{P}\left(\sum_{i=1}^{10} X_i > 8\right) + \gamma \mathrm{P}\left(\sum_{i=1}^{10} X_i = 8\right), \tag{5.8}$$

przy czym $\alpha = 0.02$ oznacza wymagany rozmiar testu.

Po odpowiednich przekształceniach otrzymamy:

```
> (gamma <- (0.02-1+pbinom(8,20,0.2))/dbinom(8,20,0.2))
[1] 0.4520676
```

co oznacza, że poszukiwana stała γ jest równa 0.4520676.

Teraz możemy już wyjaśnić, w jaki sposób należy korzystać z testu zrandomizowanego. Otóż, hipotezę zerową odrzucamy zawsze, gdy wartość statystyki testowej T przekracza 8. Natomiast gdy $T(X_1, \ldots, X_{10}) = 8$, wówczas przeprowadzamy doświadczenie Bernoulliego (niezwiązane z rozważanym zagadnieniem), z prawdopodobieństwem sukcesu równym γ. Zaobserwowanie w tym doświadczeniu sukcesu prowadzi również do odrzucenia hipotezy zerowej. We wszystkich pozostałych sytuacjach nie mamy podstaw do odrzucenia H.

Na zakończenie tego zadania sprawdźmy jeszcze rozmiar podanego testu zrandomizowanego metodą symulacyjną:

```
> n <- 100000 # liczba replikacji eksperymentu
> wyniki <- replicate(n, {
+    probka <- rbinom(1, 20, 0.2)
+    (probka > 8) || (probka == 8 && runif(1)<gamma)
+ })
> (rozmiar <- sum(wyniki)/n) # wartość przybliżona
[1] 0.01949
```

⊡

Zadanie 5.10. Niech $X_1, \ldots, X_n$ będzie próbą niezależnych obserwacji z rozkładu normalnego $\mathrm{N}(\mu, \sigma)$. Wyznacz metodą symulacyjną funkcję mocy testu do weryfikacji hipotezy $H : \mu = 2$ względem $K : \mu > 2$. Przyjmij, że liczność próby wynosi 10, natomiast poziom istotności jest równy 0.05.

Rozwiązanie. Test na poziomie istotności α do weryfikacji hipotezy $H : \mu = 2$ względem $K : \mu > 2$ ma postać

$$\varphi(X_1, \ldots, X_n) = \begin{cases} 1 & \text{gdy} \quad T(X_1, \ldots, X_n) > c, \\ 0 & \text{gdy} \quad T(X_1, \ldots, X_n) \leqslant c, \end{cases}$$

gdzie

$$T(X_1, \ldots, X_n) = \frac{\bar{X} - 2}{S}\sqrt{n}$$

oraz

$$c = t_{1-\alpha}^{[n-1]}.$$

Zatem funkcja mocy tego testu jest dana wzorem:

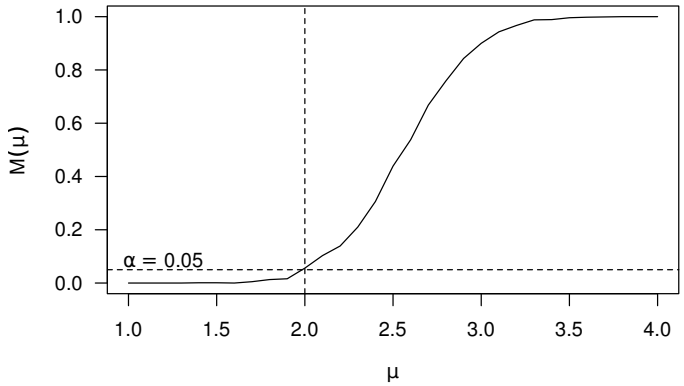
$$M_\varphi(\mu) = \mathrm{P}_\mu\left(\frac{\bar{X} - 2}{S}\sqrt{n} > t_{1-\alpha}^{[n-1]}\right). \tag{5.9}$$

Załóżmy bez straty ogólności, że $\sigma = 1$. Empiryczne wyznaczanie funkcji mocy testu φ na poziomie istotności $\alpha = 0.05$ można przeprowadzić w następujący sposób:

```
> mi   <- seq(1.00, 4.00, 0.1) # dla tych mi wyznaczamy moc
> moce <- rep(0, length(mi))   # tu będziemy przechowywać moc testów
> c <- qt(0.95, 9)
>
> n <- 1000
>
> for (i in 1:length(mi))
+ {
+    wynik <- replicate(n,
+    {
+       x <- rnorm(10, mi[i], 1)
+       T <- (mean(x)-2)*sqrt(10)/sd(x)
+       T > c # TRUE(1) albo FALSE(0)
+    })

+    moce[i] <- sum(wynik) / n
+ }
```

A oto wykres funkcji mocy testu:

```
> plot(mi, moce, xlab=expression(mu), ylab=expression(M(mu)),
+    type='l', ylim=c(0,1), las=1)
> abline(v=2, lty=2)
> abline(h=0.05, lty=2)
> text(1.2, 0.08, expression(alpha==0.05))
```



⊡

5.3. Zadania do samodzielnego rozwiązania

Zadanie 5.11. Pewien ichtiolog pobrał losową próbę 15 ryb i zmierzył ich długość. Otrzymał następujące wyniki (w mm):

$$92, 88, 85, 82, 89, 86, 81, 66, 75, 61, 78, 76, 91, 82, 82.$$

Zakładając, że rozkład długości ryb badanego gatunku jest normalny, zweryfikuj hipotezę, że średnia długość ryb tego gatunku przekracza 78 mm. Przyjmij poziom istotności $\alpha = 0.01$.

Zadanie 5.12. Pewien księgowy przypuszcza, że przeciętne saldo na kontach klientów jego firmy jest mniejsze niż 31 tys. €. Żeby to sprawdzić, pobrał losową próbę kont, otrzymując następujące wyniki dotyczące przeciętnego salda (w tys. €):

30.0, 30.0, 29.9, 31.3, 32.0, 32.0, 32.1, 30.5, 32.3, 29.5, 27.8, 27.3, 31.1, 30.7, 24.5, 28.3, 31.3, 32.7, 33.3, 26.8.

Czy prawdziwe jest przypuszczenie księgowego? Zweryfikuj stosowną hipotezę na poziomie istotności 0.01.

Zadanie 5.13. Na podstawie danych zawartych w pliku `samochody.csv` zweryfikuj przypuszczenie, że średnia moc silnika samochodów wyprodukowanych w latach 1979–1981 wynosi 84 KM (wykorzystaj zmienne `moc` i `rok`). Przyjmij poziom istotności 0.01.

Zadanie 5.14. Oszacowano przeciętną długość życia w wybranych losowo 18 krajach. Wyniki przedstawia poniższa tabela:

Kraj	Długość życia	Kraj	Długość życia	Kraj	Długość życia
Argentyna	70.5	Japonia	79	Sudan	53
Etiopia	51.5	Kenia	61	Tajwan	75
Niemcy	76	Meksyk	72	Tajlandia	68.5
Indie	57.5	Maroko	64.5	Turcja	70
Iran	64.5	RPA	64	Ukraina	70.5
Włochy	78.5	Hiszpania	78.5	USA	75.5

Czy na podstawie tych danych możemy twierdzić, że średnia długość życia przekracza 62 lata? Przyjmij poziom istotności 0.05.

Zadanie 5.15. Na podstawie danych dotyczących parametrów kilku wybranych marek samochodów (plik `samochody.csv`) stwierdź, czy występuje statystycznie istotna różnica w przyspieszeniu samochodów produkowanych w USA i w Japonii. Przyjmij poziom istotności $\alpha = 0.05$.

Zadanie 5.16. Badano wytrzymałość 20 losowo wybranych wsporników betonowych, przy czym 10 z nich wykonano metodą tradycyjną, a pozostałe niedawno opatentowaną, nową metodą. Wyniki pomiarów (w MPa) podano w poniższej tabeli:

Metoda tradycyjna	$53, 51, 62, 55, 59, 56, 61, 54, 47, 57$
Nowa metoda	$62, 55, 61, 58, 54, 49, 56, 60, 52, 63$

Czy na podstawie tych danych można stwierdzić, że wytrzymałość wsporników wykonanych nową metodą przewyższa istotnie wytrzymałość wsporników wykonanych metodą tradycyjną? Przyjmij poziom istotności 0.04.

Zadanie 5.17. Badano liczbę recept wypisywanych w ciągu 14 losowo wybranych dni przez pewnych dwóch lekarzy. Otrzymano następujące wyniki:

Lekarz I	$19, 21, 15, 17, 24, 12, 19, 14, 20, 18, 23, 21, 17, 12$
Lekarz II	$17, 15, 12, 12, 16, 15, 11, 13, 14, 21, 19, 15, 11, 10$

Zakładając, że badana cecha ma rozkład normalny, zweryfikuj przypuszczenie, że lekarz I wypisuje średnio więcej recept niż lekarz II. Przyjmij poziom istotności 0.05.

Zadanie 5.18. Badano przeciętną długość filmów produkowanych przez dwie konkurujące ze sobą firmy. W tym celu wylosowano do badania kilka filmów i otrzymano następujące dane (w minutach):

Długości filmów produkcji A	$102, 86, 98, 109, 92, 102, 95, 120$
Długości filmów produkcji B	$81, 165, 97, 134, 92, 87, 114, 120, 95, 136, 170$

Czy można twierdzić, że przeciętna długość filmów produkcji A przewyższa przeciętną długość filmów produkcji B? Zweryfikuj stosowną hipotezę na poziomie istotności 0.01.

Zadanie 5.19. Badano wpływ nowego leku na zmianę poziomu pewnej substancji we krwi (w mg/ml). W tym celu zmierzono poziom tej substancji u 8 losowo wybranych osób, a następnie, po upływie 30 minut od podania owego leku, powtórzono badanie na tej samej grupie osób. Otrzymano następujące wyniki:

Pacjent	1	2	3	4	5	6	7	8
Poziom przed	2.76	5.18	2.68	3.05	4.10	7.05	6.60	4.79
Poziom po	7.02	3.10	5.44	3.99	5.21	10.26	13.91	14.53

Czy na podstawie powyższych danych można stwierdzić, że nowy lek powoduje istotne podwyższenie poziomu owej substancji we krwi? Przyjmij poziom istotności $\alpha = 0.05$ oraz założenie o normalności rozkładu badanej cechy.

Zadanie 5.20. Badano wagę (w kilogramach) losowo wybranych kobiet palących papierosy przed i 5 tygodni po rzuceniu przez nich palenia. Otrzymano następujące wyniki:

Palaczka	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Waga przed	67	65	62	62	66	65	61	63	64	71	69	65	61	60
Waga po	69	71	65	67	74	62	69	64	70	68	73	71	67	62

Czy na podstawie powyższych danych można stwierdzić, że rzucenie palenia wpływa na wzrost wagi palącej papierosy kobiety? Przyjmij poziom istotności $\alpha = 0.05$.

Zadanie 5.21. Wśród pracowników naukowych pewnej uczelni przeprowadzono ankietę dotyczącą stażu pracy i stanu cywilnego. Otrzymano następujące wyniki liczby osób wedle stanu i stażu pracy (w latach):

Staż pracy	Panna/kawaler	Mężatka/żonaty
0–5	6	20
5–10	8	20
10–15	3	60
15–20	2	25
20–25	1	15

Zweryfikuj hipotezę, że w grupie mężatek i żonatych odsetek osób pracujących na owej uczelni dłużej niż 15 lat wynosi 0.3. Przyjmij poziom istotności $\alpha = 0.05$.

Zadanie 5.22. Na podstawie danych zawartych w pliku `samochody.csv`:

1. Podaj przedział ufności dla odsetka samochodów mających moc większą niż 80 KM (wykorzystaj zmienną `moc`). Przyjmij poziom ufności 0.95.
2. Zweryfikuj hipotezę, że ponad 50% samochodów ma moc większą niż 80 KM. Przyjmij poziom istotności 0.06.
3. Rozpatrz te problemy ponownie ograniczając się do samochodów produkowanych wyłącznie w Ameryce i Japonii (wykorzystaj zmienne `producent` i `legenda`).

5.4. Wskazówki i odpowiedzi do zadań

Ad zad. 5.12. UWAGA! W tym zadaniu, jak i w kilku następnych, należy zweryfikować hipotezę o normalności rozkładu, z którego pochodzi próbka.

Test normalności Shapiro -Wilka: wartość statystyki testowej to $T = 0.932238$, zaś p-wartość wynosi 0.1705, czyli możemy przyjąć, że próba pochodzi z rozkładu normalnego.

Do weryfikacji hipotezy $H : \mu = 31$ przeciw $K : \mu < 31$ wykorzystujemy test t. Wartość statystyki testowej to $T = -1.65$, zaś p-wartość $= 0.06$. Zatem odrzucamy H na rzecz K, co oznacza, że przypuszczenie księgarza okazało sie prawdziwe.

Ad zad. 5.14. Test normalności Shapiro-Wilka: wartość statystyki testowej to $T = 0,9305$ zaś p-wartość $= 0.1985$, czyli możemy przyjąć, że próba pochodzi z rozkładu normalnego.

Do weryfikacji hipotezy $H : \mu = 62$ przeciw $K : \mu > 62$ wykorzystujemy test t. Wartość statystyki testowej to $T = 3.148$, zaś p-wartość $= 0.00295$. Zatem odrzucamy H na rzecz K, tzn. średnia długość życia w badanych krajach przekracza 62 lata.

Ad zad. 5.15. UWAGA! W tym zadaniu (jak i w niektórych następnych) należy zweryfikować hipotezę o równości wariancji w obu próbach.

Ad zad. 5.16. Łatwo sprawdzić, że próby pochodzą z rozkładów normalnych o równych wariancjach.

Do weryfikacji hipotezy $H : \mu_1 = \mu_2$ przeciw $K : \mu_1 < \mu_2$ wykorzystujemy test t. Wartość statystyki testowej to $T = -0,73$, zaś p-wartość $= 0.23695$. Zatem nie mamy podstaw do odrzucenia H, co oznacza, że wytrzymałość wsporników wykonanych nową metodą nie przewyższa istotnie wytrzymałości wsporników wykonanych metodą tradycyjną.

Ad zad. 5.18. Próby pochodzą z rozkładów normalnych, ale nie można przyjąć założenia o równych wariancjach.

Do weryfikacji hipotezy $H : \mu_1 = \mu_2$ przeciw $K : \mu_1 > \mu_2$ wykorzystujemy test t Welcha. Wartość statystyki testowej to $T = -1.69$, zaś p-wartość $= 0.94$. Zatem nie mamy podstaw do odrzucenia H, co oznacza, że przeciętna długość filmów produkcji A nie przewyższa przeciętnej długości filmów produkcji B.

Ad zad. 5.19. Obserwacje parami zależne.

Ad zad. 5.20. Próby parami zależne, pochodzące z rozkładów normalnych. Do weryfikacji hipotezy $H : \mu_1 = \mu_2$ przeciw $K : \mu_1 < \mu_2$ wykorzystujemy test t. Wartość statystyki testowej to $T = -3.85$, zaś p-wartość $= 0.001$. Zatem odrzucamy H na rzecz K, tzn. średnio rzecz biorąc, rzucenie palenia wpływa na wzrost wagi.

Weryfikacja hipotez: Podstawowe testy nieparametryczne

6

6.1. Wprowadzenie

6.1.1. Testy dla mediany

6.1.1.1. Test rangowanych znaków

Niech $X_1, \ldots, X_n$ będzie próbką prostą z rozkładu ciągłego i symetrycznego. Do weryfikacji hipotezy zerowej dotyczącej mediany $K : \mathrm{Med} = m_0$ można posłużyć się tzw. testem rangowanych znaków (ang. *signed rank test*), który w R jest zaimplementowany jako funkcja `wilcox.test()`.

Wektor obserwacji `x` podajemy jako pierwszy argument funkcji `wilcox.test`, natomiast jako drugi argument podajemy hipotetyczną wartość mediany (parametr `mu`). Wybór hipotezy alternatywnej, tzn. $K : \mathrm{Med} \neq m_0$, $K' : \mathrm{Med} < m_0$ lub $K'' : \mathrm{Med} > m_0$, można określić podając jako kolejny argument `alternative="twosided"` (wartość domyślna), `"less"` lub `"greater"`.

Dla przykładu, gdy chcemy zweryfikować hipotezę $H : \mathrm{Med} = 0$ przeciwko $K : \mathrm{Med} \neq 0$, piszemy:

```
> wilcox.test(x, mu=0)
```

6.1.1.2. Test Wilcoxona

Niech $X_1, \ldots, X_{n_1}$ i $Y_1, \ldots, Y_{n_2}$ będą niezależnymi próbkami prostymi z rozkładów ciągłych o medianach, odpowiednio, Med_1 i Med_2. Test Wilcoxona (zwany też testem Manna-Whitneya-Wilcoxona) służy do weryfikacji hipotez o równości median dwóch populacji, tzn. $H : \mathrm{Med}_1 = \mathrm{Med}_2$. Ten test jest dostępny w R za pośrednictwem funkcji `wilcox.test()`.

Jako pierwszy i drugi argument tej funkcji podajemy wektory obserwacji `x` i `y`. Za pomocą kolejnego argumentu: `alternative="twosided"` (wartość domyślna), `"less"` lub `"greater"`, możemy ustalić hipotezę alternatywną, tzn. $K : \mathrm{Med}_1 \neq \mathrm{Med}_2$, $K' : \mathrm{Med}_1 < \mathrm{Med}_2$ lub $K'' : \mathrm{Med}_1 > \mathrm{Med}_2$.

Dla ilustracji, chcemy zweryfikować hipotezę $H : \mathrm{Med}_1 = \mathrm{Med}_2$ przeciwko $K : \mathrm{Med}_1 \neq \mathrm{Med}_2$, wywołujemy funkcję:

```
> wilcox.test(x, y)
```

6.1.2. Testy zgodności

Do weryfikacji hipotez o zgodności rozkładu badanej cechy z interesującym nas rozkładem, można korzystać w R z testu chi-kwadrat (funkcja `chisq.test()`) lub testu Kołmogorowa (funkcja `ks.test()`), przy czym test Kołmogorowa używamy w przypadku prób pochodzących z populacji o rozkładach typu ciągłego. Funkcja `ks.test()` służy też do przeprowadzania testu Kołmogorowa-Smirnowa o identyczności rozkładów badanej cechy w dwóch populacjach o rozkładach typu ciągłego. W przypadku $k > 2$ populacji o rozkładach typu ciągłego używa się w tym celu testu Kruskala-Wallisa (funkcja `kruskal.test()`). Do porównania rozkładów w wielu populacjach o rozkładach dyskretnych, można użyć funkcji `prop.test()`, za pomocą której przeprowadzany jest test jednorodności chi-kwadrat.

6.1.2.1. Test chi-kwadrat Pearsona

Niech $X_1, \ldots, X_n$ będzie próbką prostą z rozkładu o dystrybuancie F. Test zgodności chi-kwadrat do weryfikacji hipotezy $H : F = F_0$ przeciw $K : F \neq F_0$ przeprowadza się dla prób pogrupowanych w szereg rozdzielczy. Jako pierwszy argument funkcji `chisq.test()`, podajemy wektor zaobserwowanych liczności poszczególnych klas szeregu. Drugim argumentem tej funkcji jest wektor prawdopodobieństw p przynależności do poszczególnych klas obliczonych przy założeniu prawdziwości hipotezy zerowej. Należy pamiętać, że test zgodności chi-kwadrat jest testem asymptotycznym, tzn. wymagającym dostatecznie wielu obserwacji.

6.1.2.2. Test Kołmogorowa

Niech $X_1, \ldots, X_n$ będzie próbką prostą z rozkładu o ciągłej dystrybuancie F. Weryfikujemy hipotezę $H : F = F_0$ przeciw $K : F \neq F_0$. W teście zgodności Kołmogorowa porównuje się wartość dystrybuanty empirycznej, zbudowanej na podstawie próby, z dystrybuantą teoretyczną. Jako pierwszy argument funkcji `ks.test()`, podajemy wektor obserwacji `x`. Drugim argumentem jest nazwa funkcji wyznaczającej dystrybuantę rozkładu, z którym chcemy badać zgodność, np. `"punif"` – gdy testujemy zgodność z rozkładem jednostajnym lub `"pnorm"` – gdy testujemy zgodność z rozkładem normalnym. Jako kolejne argumenty podajemy parametry interesującego nas rozkładu.

Przykładowo, zgodność z rozkładem $\mathrm{U}[0, 1]$ badamy wpisując:

```
> ks.test(x, "punif")
```

Testowanie zgodności z rozkładem normalnym $\mathrm{N}(0.5, 1)$ dokonuje się poleceniem:

```
> ks.test(x, "pnorm", 0.5, 1)
```

natomiast zgodność z rozkładem wykładniczym $\mathrm{Exp}(2)$ testujemy następująco:

```
> ks.test(x, "pexp", 2)
```

6.1.2.3. Testy normalności

Do weryfikacji hipotezy o normalności rozkładu badanej cechy można korzystać z obu wymienionych wyżej testów. Jednak istnieją testy zaprojektowane specjalnie w celu badania zgodności z rozkładem normalnym. Takim testem jest np. test Shapiro-Wilka dostępny w programie R za pośrednictwem funkcji `shapiro.test()`.

Inne dostępne w R testy normalności są zamieszczone w pakiecie nortest. Są to: test Cramera-von Misesa (`cvm.test()`), test Andersona-Darlinga (`ad.test()`), test Lillieforsa (`lillie.test()`), test chi-kwadrat Pearsona (`pearson.test()`), test Shapiro-Francii (`sf.test()`). Pierwszym argumentem wyżej wymienionych funkcji jest wektor obserwacji `x`.

6.1.2.4. Wykres kwantylowy

Przydatną, poglądową metodą badania zgodności z rozkładem normalnym jest wykres kwantylowy dla rozkładu normalnego (ang. *normal Q-Q plot* bądź *quantile-quantile plot*), zaimplementowany w funkcji `qqnorm()`.

Na wykresie tym porównujemy kwantyle obliczone dla próbki z kwantylami teoretycznymi rozkładu normalnego. Wektor `x`, zawierający wartości próbki, podajemy jako argument tej funkcji. Za pomocą wywołania metody `qqline(x)`, można do wykresu kwantylowego dorysować prostą przechodzącą przez kwantyle rozkładu teoretycznego. Jeśli nasza próba pochodzi z rozkładu normalnego, to punkty na wykresie będą układać się wzdłuż tej prostej. Możemy to zrealizować sekwencją komend:

```
> qqnorm(x)
> qqline(x)
```

6.1.2.5. Test Kołmogorowa-Smirnowa

Niech $X_1, \ldots, X_{n_1}$ i $Y_1, \ldots, Y_{n_2}$ będą niezależnymi próbkami prostymi z rozkładów ciągłych o dystrybuantach, odpowiednio, F i G. Do weryfikacji hipotezy $H : F = G$ można posłużyć się testem Kołmogorowa-Smirnowa, który porównuje wartości dystrybuant empirycznych, zbudowanych na podstawie obu prób. Jako pierwszy argument funkcji `ks.test()` podajemy wektor `x`, zawierający wartości pierwszej próbki, a jako drugi – wektor `y`, zawierający wartości drugiej próbki.

6.1.2.6. Test Kruskala-Wallisa

Test Kruskala-Wallisa służy do weryfikacji hipotezy o identyczności rozkładów badanej cechy w $k > 2$ populacjach, tzn. $H : F_1 = F_2 = \ldots = F_k$. Jako argumenty funkcji `kruskal.test()` podajemy wektory obserwacji $\mathbf{x}_1, \ldots, \mathbf{x}_k$ pochodzących z kolejnych próbek.

6.1.2.7. Test jednorodności chi-kwadrat

Test jednorodności chi-kwadrat służy do weryfikacji hipotezy o jednakowym rozkładzie badanej cechy kilku populacji w przypadku dyskretnym. Przeprowadzany jest na podstawie tablicy częstości pojawień się interesujących nas obserwacji w kolejnych próbach. Jako pierwszy argumenty stosowanej w tym celu funkcji `prop.test()` podajemy wektor x zawierający liczności elementów wyróżnionych w obserwowanych próbkach, natomiast drugim argumentem jest wektor y podający liczności prób.

6.2. Zadania rozwiązane

Zadanie 6.1. Wygeneruj 200-elementowe próbki z rozkładu normalnego, Cauchy'ego, jednostajnego, Laplace'a, wykładniczego oraz z rozkładu o gęstości

$$f(x) = \begin{cases} e^x & \text{dla} \quad x \leqslant 0, \\ 0 & \text{dla} \quad x > 0. \end{cases} \tag{6.1}$$

Następnie utwórz dla każdej próbki tzw. wykres normalności (czyli wykresy kwantylowe dla rozkładu normalnego). Jak zmienia się kształt wykresu w zależności od typu rozkładu?

Rozwiązanie. Zanim będziemy mogli zająć się porównywaniem wykresów, wpierw wygenerujemy próbki z następujących rozkładów:

1. normalnego $\mathrm{N}(0, 1)$,
2. Cauchy'ego $\mathrm{C}(0, 1)$,
3. jednostajnego $\mathrm{U}[0, 1]$,
4. Laplace'a $\mathrm{La}(0, 1)$,
5. wykładniczego $\mathrm{Exp}(1)$
6. o gęstości (6.1).

W środowisku R zaimplementowano funkcje pozwalające generować próbki z pierwszych trzech rozkładów oraz rozkładu wykładniczego (por. rozdział 3), tzn. `rnorm()`, `rcauchy()`, `runif()` i `rexp()`. Funkcje generujące pozostałe dwa rozkłady zaimplementujemy samodzielnie.

Przypomnijmy, że rozkład Laplace'a $\mathrm{La}(a, b)$, gdzie $a \in \mathbb{R}$, $b > 0$, dany jest dystrybuantą

$$F(x) = 0.5 + 0.5\,\mathrm{sgn}\,(x-a)\left(1 - \exp\left(-\frac{|x-a|}{b}\right)\right). \tag{6.2}$$

Łatwo pokazać, że jeżeli $U \sim \mathrm{U}[-0.5, 0.5]$, wtedy $a - b\,\mathrm{sgn}(U)\ln(1-2|U|) \sim \mathrm{La}(a, b)$ (rozkład ten wybraliśmy nieprzypadkowo, bowiem jest on symetryczny i ma taką samą kurtozę jak rozkład normalny).

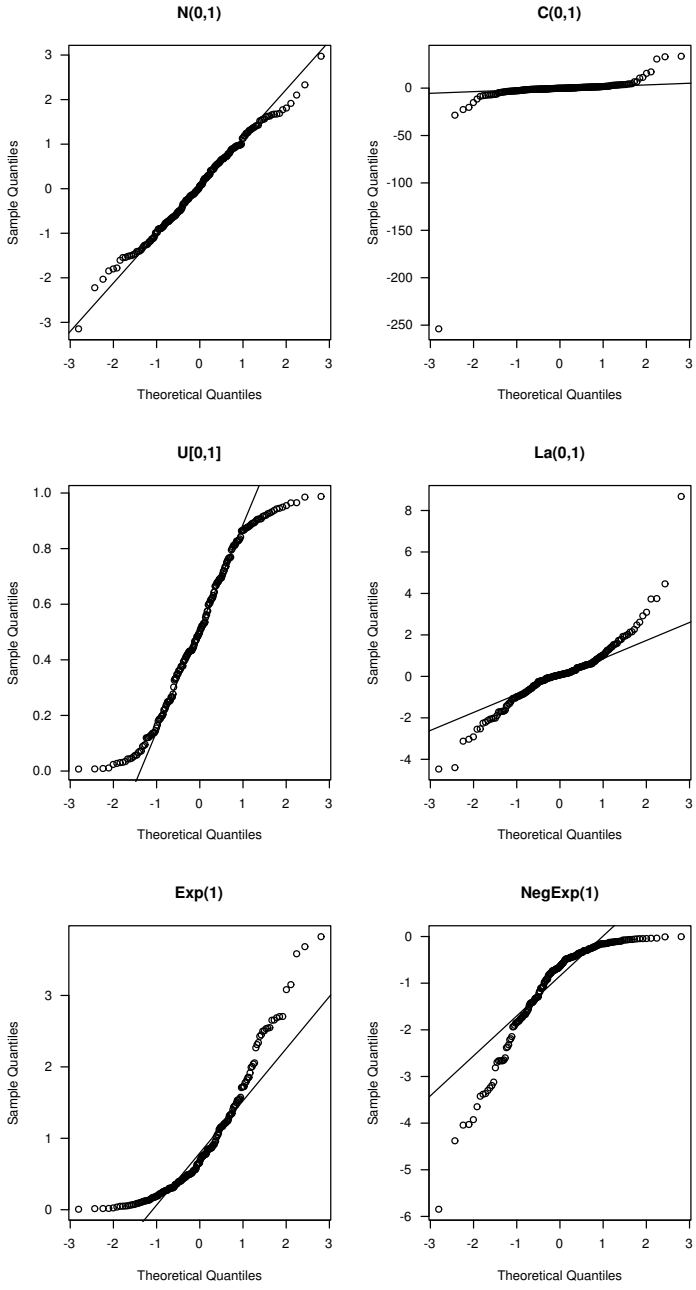
Nietrudno zauważyć, że wykres funkcji danej wzorem (6.1) dostajemy w wyniku przekształcenia symetrycznego względem osi $0Y$ wykresu gęstości rozkładu wykładniczego $\mathrm{Exp}(1)$. Tym samym próbkę z rozkładu o gęstości (6.1) możemy otrzymać generując liczby losowe z rozkładu wykładniczego $\mathrm{Exp}(1)$, a następnie mnożąc je przez (-1).

Zatem generowanie próbek z zadanych rozkładów przebiega następująco:

```
> n <-  200 # liczba obserwacji
> a <-  rnorm(n)
> b <-  rcauchy(n)
> c <-  runif(n)
> U <-  runif(n, -0.5, 0.5)
> d <- -sign(U)*log(1-2*abs(U))
> e <-  rexp(n, 1)
> f <- -rexp(n, 1)
```

po czym pozostaje „wrzucić" każdą z próbek na siatkę wykresu kwantylowego utworzonego dla rozkładu normalnego:

```
> par(mfrow=c(3,2)) # 3x2 podwykresy
>
> qqnorm(a, main="N(0,1)", las=1)
> qqline(a)
>
> qqnorm(b, main="C(0,1)", las=1)
> qqline(b)
>
> qqnorm(c, main="U[0,1]", las=1)
> qqline(c)
>
> qqnorm(d, main="La(0,1)", las=1)
> qqline(d)
>
> qqnorm(e, main="Exp(1)", las=1)
> qqline(e)
>
> qqnorm(f, main="NegExp(1)", las=1)
> qqline(f)
```



Rys. 6.1. Wykresy kwantylowe (ilustracja do zadania 6.1).

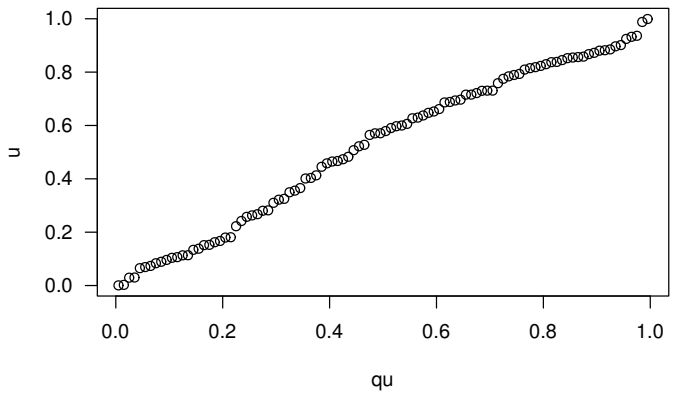
Szczegółową analizę powyższych obrazków pozostawiamy Czytelnikowi. Zwróćmy tylko uwagę, że zastosowanie wykresu kwantylowego rozkładu normalnego wykracza daleko poza cel, dla którego ów wykres został stworzony, tzn. do sprawdzania, czy badana próbka pochodzi właśnie z rozkładu normalnego. Choćby pobieżne przyjrzenie się otrzymanym obrazkom pozwala stwierdzić, że w sytuacjach, w których mieliśmy do czynienia z rozkładami innymi niż normalny, na wykresie kwantylowym ujawniają się charakterystyczne cechy rozkładów informujące nas, czy jest to rozkład symetryczny, czy asymetryczny (i o jakim znaku owej asymetrii), czy ma „cięższe ogony" niż rozkład normalny itd. ⊡

Zadanie 6.2. Utwórz wykres „jednostajności" i „wykładniczości", czyli wykresy kwantylowe do porównywania losowych prób, odpowiednio, z rozkładu jednostajnego i rozkładu wykładniczego, z kwantylami teoretycznymi tych rozkładów.

Rozwiązanie. Żądane wykresy utworzymy w ten sposób, że na osi odciętych położymy kwantyle teoretyczne danego rozkładu, natomiast na osi rzędnych uporządkowane rosnąco wartości wygenerowanej próbki.

Oto kod źródłowy i wykres kwantylowego dla rozkładu jednostajnego, por. rys. 6.2:

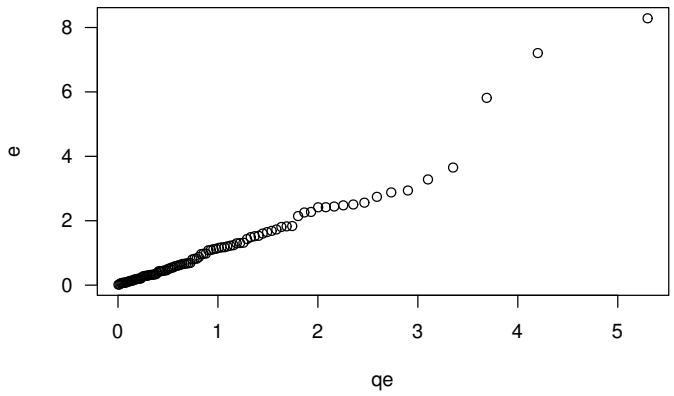
```
> u <- runif(100)             # losujemy próbę
> qu <- qunif(ppoints(100))   # kwantyle teoretyczne (zob. ?ppoints)
> qqplot(qu, u, las=1)
```



Rys. 6.2. Wykres kwantylowy dla rozkładu jednostajnego

a poniżej kod i wykres kwantylowy dla rozkładu wykładniczego, zob. rys. 6.3:

```
>  e <- rexp(100)
> qe <- qexp(ppoints(100))
> qqplot(qe, e, las=1)
```



Rys. 6.3. Wykres kwantylowy dla rozkładu wykładniczego

⊡

Zadanie 6.3. Badania grupy krwi 200 osób dały następujące wyniki: grupę O miały 73 osoby, grupę A – 74 osoby, grupę B – 34 osoby, natomiast grupę AB miało 19 osób.

1. Czy na podstawie tych wyników można przyjąć hipotezę o równomiernym rozkładzie wszystkich grup krwi?
2. Zweryfikuj hipotezę, że grupa krwi O występuje średnio u 36.7% ludzi, grupa A – u 37.1%, B – u 18.6%, natomiast grupa AB występuje u 7.6% ogółu ludzi.

W obu przypadkach przyjmij poziom istotności $\alpha = 0.05$.

Rozwiązanie. Zaczniemy od weryfikacji hipotezy o równomierności rozkładu grup krwi. Mówiąc nieco bardziej formalnie, mamy do czynienia z rozkładem czteropunktowym, przyjmującym poszczególne wartości (odpowiadające grupom krwi) z prawdopodobieństwami: p_0, p_A, p_B i p_{AB}. Tym samym pierwsze z rozpatrywanych zagadnień sprowadza się do weryfikacji hipotezy $H : p_0 = p_A = p_B = p_{AB} = \frac{1}{4}$, przy alternatywie $K : \neg H$ mówiącej, iż nie mamy do czynienia z rozkładem równomiernym.

Ze względu na to, że dysponujemy w miarę liczną próbką, możemy posłużyć się testem zgodności chi-kwadrat:

```
> krew <- c(73, 74, 34, 19) # wyniki badań
> praw <- rep(0.25, 4)      # testowane prawdopodobieństwa
> chisq.test(krew, p=praw)  # test chi-kwadrat
```

```
Chi-squared test for given probabilities

data:  krew
X-squared = 46.44, df = 3, p-value = 4.572e-10
```

Wynik testowania jest oczywisty – odrzucamy hipotezę zerową. Oznacza to, że rozkład grupy krwi nie jest równomierny. Warto zwrócić uwagę na to, że w zasadzie decyzja ta nie zależy od przyjętego poziomu ufności, bowiem p-value < 0.0001.

Rozważmy teraz hipotezę $H : p_0 = 0.367, p_A = 0.371, p_B = 0.186, p_{AB} = 0.076$, przy alternatywie $K : \neg H$ mówiącej, iż faktyczny rozkład grup krwi różni się istotnie od wskazanego w hipotezie zerowej. Ponowne zastosowanie testu zgodności chi-kwadrat da nam następujący rezultat:

```
> praw2 <- c(0.367, 0.371, 0.186, 0.076)
> chisq.test(krew, p=praw2)

Chi-squared test for given probabilities

data:  krew
X-squared = 1.228, df = 3, p-value = 0.7463
```

Jak widać, tym razem nie mamy podstaw do odrzucenia hipotezy zerowej. ⊡

Zadanie 6.4. Postanowiono zbadać, ile zadań rozwiązują studenci w czasie egzaminu ze statystyki. Poniższa tabela zawiera wyniki badania przeprowadzonego w grupie 120 losowo wybranych studentów:

Liczba rozwiązanych zadań	0	1	2	3	4
Liczba studentów	10	32	46	26	6

Na poziomie istotności 0.05 zweryfikuj hipotezę, że liczba rozwiązanych zadań ma rozkład dwumianowy $\mathrm{Bin}(4, 0.5)$.

Rozwiązanie. Niech X oznacza zmienną losową wskazującą liczbę zadań rozwiązanych przez studenta podczas egzaminu ze statystyki. Naszym celem będzie Weryfikacja hipotezy $H : X \sim \mathrm{Bin}(4, 0.5)$, wobec alternatywy $K : \neg H$ mówiącej, iż rozkład liczby rozwiązanych zadań różni się istotnie od wskazanego przez hipotezę H. A oto kod źródłowy procedury wykorzystującej test zgodności chi-kwadrat:

```
> licznosci <- c(10, 32, 46, 26, 6)
> # Pr(X=i); X~Bin(4, 0.5), i=0,1,...,4:
> (pstwa <- dbinom(0:4, 4, 0.5))
[1] 0.0625 0.2500 0.3750 0.2500 0.0625
> chisq.test(licznosci, p=pstwa)

Chi-squared test for given probabilities
```

```
data:  licznosci
X-squared = 1.8222, df = 4, p-value = 0.7684
```

W świetle uzyskanych wyników nie ma podstaw do odrzucenia hipotezy, że liczba rozwiązanych zadań ma rozkład $\mathrm{Bin}(4, 0.5)$. ⊡

Zadanie 6.5. Wśród studentów, którzy nie zdali egzaminu z rachunku prawdopodobieństwa przeprowadzono sondaż mający wskazać, ile zadań rozwiązali oni samodzielnie w ramach przygotowań do tego egzaminu. Poniższa tabelka przedstawia wyniki badania przeprowadzonego w grupie 120 losowo wybranych studentów:

Liczba rozwiązanych zadań	0	1	2	$\geqslant 3$
Liczba studentów	64	30	18	8

Na poziomie istotności 0.05 zweryfikuj hipotezę, że liczba rozwiązanych samodzielnie zadań ma rozkład Poissona.

Rozwiązanie. Niech X oznacza zmienną losową opisującą liczbę samodzielnie rozwiązanych zadań. Naszym celem jest weryfikacja hipotezy zerowej H : *X ma rozkład Poissona*, względem hipotezy alternatywnej $K : \neg H$ mówiącej, iż rozkład liczby rozwiązanych samodzielnie zadań nie jest rozkładem Poissona. Zwróćmy uwagę, że w przeciwieństwie do rozważanych poprzednio zadań dotyczących testowania zgodności, tym razem hipoteza zerowa jest złożona (a nie prosta). Oznacza to, iż testowanie zgodności musimy poprzedzić estymacją parametrów rozkładu występującego w hipotezie zerowej.

W naszym przypadku, z uwagi na to, iż jest to rozkład Poissona, mamy do oszacowania tylko jeden parametr λ. Łatwo pokazać, że estymatorem metody największej wiarogodności tego parametru jest średnia arytmetyczna, tzn. $\hat{\lambda} = \bar{X}$. Dopiero po wyestymowaniu tego parametru możemy przystąpić do weryfikacji hipotezy o zgodności próbki z rozkładem doprecyzowanym dzięki dokonanej estymacji. A oto kod naszej procedury testowej:

```
> zadania <- c(64, 30, 18, 8) # uprowadzamy dane
> zadania / sum(zadania)
[1] 0.53333333 0.25000000 0.15000000 0.06666667
> (lambda <- sum(zadania*(0:3))/sum(zadania))
[1] 0.75
> pstwa <- dpois(0:2, lambda)
> (pstwa <- c(pstwa, 1-sum(pstwa)))
[1] 0.47236655 0.35427491 0.13285309 0.04050544
> chisq.test(zadania, p=pstwa)
Warning: Chi-squared approximation may be incorrect

Chi-squared test for given probabilities
```

```
data:  zadania
X-squared = 6.9204, df = 3, p-value = 0.07448
```

Zwróćmy uwagę, że – formalnie rzecz biorąc – zaimplementowany w R test nie uwzględnia zmniejszenia o 1 stopni swobody (`df`, ang. *degrees of freedom*), spowodowanego estymacją parametru z próby. W związku z tym wyznaczymy samodzielnie obszar krytyczny, który pozwoli nam podjąć ostateczną decyzję:

```
> qchisq(0.95, 3) # lewa granica obszaru krytycznego dla df=3
[1] 7.814728
> qchisq(0.95, 2) # lewa granica obszaru krytycznego dla df=2
[1] 5.991465
```

Z uwagi na to, że otrzymana wartość statystyki testowej $\chi^2 = 6.9204$ wpada do obszaru krytycznego $K_{0.05} = [5.9915, +\infty)$, odrzucamy hipotezę zerową, co oznacza, iż rozkład liczby samodzielnie rozwiązywanych zadań nie jest rozkładem Poissona. ⊡

Zadanie 6.6. Wygeneruj $n = 20$-elementową próbę losową z rozkładu Cauchy'ego $\mathrm{C}(m_0, 1)$. Za pomocą testu znaków zweryfikuj hipotezę, że mediana rozkładu, z którego pochodzi próba, wynosi $m_0 = 4$. Porównaj otrzymany rezultat z wynikiem testu rangowanych znaków.

Rozwiązanie. Niech $X_1, .., X_n$ oznacza próbkę prostą z rozkładu ciągłego w otoczeniu mediany. Do weryfikacji hipotezy dotyczącej mediany $H : \mathrm{Med} = m_0$ przeciw alternatywie $K : \mathrm{Med} \neq m_0$ służy np. test znaków (ang. *sign test*), którego statystyka testowa ma postać

$$T = \sum_{i=1}^{n} \mathbb{I}(X_i > m_0), \tag{6.3}$$

zaś obszar krytyczny jest dany wzorem

$$K_\alpha = \left[0, r_{\alpha/2}\right] \cup \left[n - r_{\alpha/2}, n\right], \tag{6.4}$$

gdzie $r_{\alpha/2}$ jest kwantylem rzędu $\alpha/2$ rozkładu $\mathrm{Bin}(n, 0.5)$. W naszym zadaniu: $n = 20$, $\alpha = 0.05$, $m_0 = 4$.

Po wygenerowaniu próbki

```
> n  <- 20
> m0 <- 4
> x  <- rcauchy(n, m0)
> median(x)
[1] 4.636915
```

wyznaczamy wartość statystyki testowej oraz kwantyl wymagany przez obszar krytyczny

```
> (T <- sum(x>m0))
[1] 13
> alfa <- 0.05
> (r <- qbinom(alfa*0.5, n, 0.5))
[1] 6
```

Z uwagi na to, że otrzymana wartość statystyki testowej $T = 13$ nie wpada do obszaru krytycznego $K_{0.05} = [0, 6] \cup [14, 20]$, nie mamy podstaw do odrzucenia hipotezy zerowej, iż mediana rozkładu jest równa 4.

Dla porównania, zastosujmy test rangowanych znaków (ang. *signed rank test*):

```
> wilcox.test(x, mu=4)

Wilcoxon signed rank test

data:  x
V = 125, p-value = 0.4749
alternative hypothesis: true location is not equal to 4
```

Informacja

Należy pamiętać, że jednym z założeń (obok ciągłości rozkładu), które powinno być spełnione, jeśli chcemy posłużyć się testem rangowanych znaków, jest symetria rozkładu.

⊡

Zadanie 6.7. Zmierzono czas trwania siedmiu rozmów telefonicznych i otrzymano następujące dane (w minutach):

$$2.5; 1.8; 6.0; 0.5; 8.75; 1.2; 3.75.$$

Na poziomie istotności $\alpha = 0.01$ zweryfikuj hipotezę, że czas trwania rozmowy ma rozkład wykładniczy o wartości średniej 4 minuty.

Rozwiązanie. Niech X oznacza zmienną losową, której wartość odpowiada długości rozmowy telefonicznej. Naszym zadaniem jest weryfikacja hipotezy, że czas trwania rozmowy ma rozkład wykładniczy o wartości średniej 4 minuty, co jest równoważne stwierdzeniu, że jest to rozkład wykładniczy z parametrem $\lambda = \frac{1}{4}$. Tak więc będziemy testować hipotezę $H : X \sim \mathrm{Exp}(\frac{1}{4})$ przeciw hipotezie alternatywnej $K : \neg H$. Ponieważ dysponujemy stosunkowo nieliczną próbką z rozkładu ciągłego, możemy posłużyć się testem zgodności Kołmogorowa:

```
> telefony <- c(2.5, 1.8, 6, 0.5, 8.75, 1.2, 3.75)
> ks.test(telefony, "pexp", 0.25)
```

```
One-sample Kolmogorov-Smirnov test

data:  telefony
D = 0.1175, p-value = 0.9997
alternative hypothesis: two-sided
```

Jak widać, w świetle uzyskanych danych, nie mamy podstaw do odrzucenia hipotezy zerowej. ⊡

Zadanie 6.8. Na podstawie danych dotyczących parametrów kilku wybranych marek samochodów (plik `samochody.csv`), zweryfikuj hipotezę o jednakowym rozkładzie zużycia paliwa przez samochody produkowane w USA i w Japonii (wykorzystaj zmienne `mpg` i `producent`). Przyjmij poziom istotności $\alpha = 0.05$.

Rozwiązanie. Dysponujemy dwiema niezależnymi próbkami losowymi $X_1, \ldots, X_{n_1}$ i $Y_1, \ldots, Y_{n_2}$ z rozkładów o ciągłych dystrybuantach, odpowiednio, F i G. Weryfikacja hipotezy o jednakowym rozkładzie zużycia paliwa przez samochody amerykańskie i japońskie, zapisana formalnie, sprowadza się do testowania hipotezy zerowej $H : F(x) = G(x)$ *dla każdego* $x \in \mathbb{R}$, przeciw alternatywie $K : F(x) \neq G(x)$ *dla przynajmniej jednego* x.

Po załadowaniu zbioru danych

```
> auta <- read.csv(
+    "http://www.ibspan.waw.pl/~pgrzeg/stat_lab/samochody.csv",
+    head=TRUE, dec=",", sep=";")
> head(auta) # czy OK?
   mpg cylindry moc przysp rok waga producent      marka      model
1 43.1        4  48   21.5  78 1985         2 Volkswagen Rabbit Dl
2 36.1        4  66   14.4  78 1800         1 Ford       Fiesta
3 32.8        4  52   19.4  78 1985         3 Mazda      GLC Deluxe
4 39.4        4  70   18.6  78 2070         3 Datsun     B210 GX
5 36.1        4  60   16.4  78 1800         3 Honda      Civic CVCC
6 19.9        8 110   15.5  78 3365         1 Oldsmobile Cutlass
  cena   legenda
1 2400 America=1
2 1900  Europe=2
3 2200  Japan =3
4 2725
5 2250
6 3300
```

przechodzimy do weryfikacji postawionej hipotezy. Możemy posłużyć się w tym celu testem Kołmogorowa-Smirnowa:

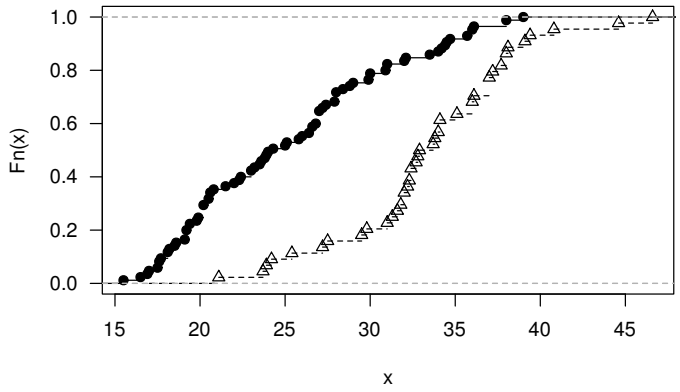
```
> mpga <- auta$mpg[auta$producent == 1]
> mpgj <- auta$mpg[auta$producent == 3]
> ks.test(mpga, mpgj)
```

```
Warning: cannot compute exact p-value with ties

Two-sample Kolmogorov-Smirnov test

data:  mpga and mpgj
D = 0.5963, p-value = 2.229e-09
alternative hypothesis: two-sided
```

którego wynik jednoznacznie wskazuje na odrzucenia H oznaczające, iż występują istotne różnice między rozkładami zużycia paliwa przez samochody produkowane w Ameryce i w Japonii.

Jak pamietamy, w teście Kołmogorowa-Smirnowa mierzymy odległość między dystrybuantami empirycznymi badanych prób. Wykresy dystrybuant empirycznych obu prób możemy uzyskać następująco:

```
> plot(ecdf(mpga), xlim=range(c(mpga,mpgj)), main="", las=1)
> plot(ecdf(mpgj), add=TRUE, pch=2, lty=2)
```



Innym testem, który może być wykorzystany do weryfikacji rozważanej hipotezy jest test Wilcoxona, zwany także testem Manna-Whitneya-Wilcoxona:

```
> wilcox.test(mpga, mpgj)

Wilcoxon rank sum test with continuity correction

data:  mpga and mpgj
W = 607, p-value = 3.547e-10
alternative hypothesis: true location shift is not equal to 0
```

Ten test potwierdza decyzję sugerowana przez test Kołmogorowa-Smirnowa. Mamy zatem mocne podstawy do odrzucenia hipotezy o równości badanych rozkładów. ⊡

Zadanie 6.9. W celu zbadania, czy nowy rodzaj paliwa lotniczego ma istotny wpływ na zasięg lotu pewnego samolotu sportowego, wykonano 10 pomiarów zasięgu samolotów napędzanych stosowanym dotąd paliwem oraz 10 pomiarów dla samolotów zasilanych nowym paliwem. Otrzymano następujące wyniki (w km):

Stosowane dotąd paliwo	1039, 1168, 1008, 1035, 1035, 1025, 1059, 1012, 1212, 1039
Nowy rodzaj paliwa	1096, 1161, 1210, 1088, 1154, 1111, 1103, 1094, 1059, 1177

Czy na podstawie tych danych można stwierdzić, że nowy rodzaj paliwa lotniczego ma istotny wpływ na wzrost przeciętnego zasięgu samolotu? Przyjmij $\alpha = 0.05$.

Rozwiązanie. Jesteśmy zainteresowani doborem jak najmocniejszego testu. Przykładowo, gdyby okazało się, że rozkłady, z których pochodzą próbki, są normalne, wówczas mielibyśmy mocny test do weryfikacji odpowiedniej hipotezy dotyczącej średniego zasięgu. Sprawdźmy zatem, czy zasadne byłoby przyjęcie założenia o normalności obydwu rozkładów:

```
> stare <- c(1039, 1168, 1008, 1035, 1035, 1025, 1059, 1012, 1212, 1039)
> nowe  <- c(1096, 1161, 1210, 1088, 1154, 1111, 1103, 1094, 1059, 1177)
> shapiro.test(stare)

Shapiro-Wilk normality test

data:  stare
W = 0.7223, p-value = 0.001641
> shapiro.test(nowe)

Shapiro-Wilk normality test

data:  nowe
W = 0.9355, p-value = 0.5043
```

Jako że pierwsza próbka nie pochodzi z rozkładu normalnego, nie możemy skorzystać z testu t dla średnich w dwóch próbach niezależnych. Możemy jednak problem porównania przeciętnego zasięgu wyrazić za pośrednictwem median, po czym przetestować odpowiednią hipotezę testem nieparametrycznym. Dokładniej, oznaczając mediany obu próbek, odpowiednio, przez m_1 i m_2, zweryfikujemy hipotezę $H : m_1 = m_2$, przeciw $K : m_1 < m_2$, testem Wilcoxona:

```
> mean(stare)
[1] 1063.2
> mean(nowe)
[1] 1125.3
> wilcox.test(stare, nowe, alternative="less")
Warning: cannot compute exact p-value with ties

Wilcoxon rank sum test with continuity correction
```

```
data:  stare and nowe
W = 18.5, p-value = 0.009487
alternative hypothesis: true location shift is less than 0
```

Uzyskana p-wartość 0.00949 wskazuje odrzucenie hipotezy zerowej, co oznacza, iż zastosowanie nowego rodzaju paliwa wpływa na wzrost zasięgu lotu.

⊡

Zadanie 6.10. W celu porównania trzech metod nauki stenografii, przeprowadzono sprawdzian na losowych próbach osób szkolonych poszczególnymi metodami. Otrzymano następujące wyniki:

Metoda A	$147, 188, 162, 144, 157, 179, 165, 180$
Metoda B	$153, 161, 157, 155, 163, 160, 154$
Metoda C	$173, 152, 194, 186, 166, 194, 178, 192, 186$

Zbadaj, czy te trzy metody są tak samo efektywne. Przyjmij poziom istotności $\alpha = 0.05$.

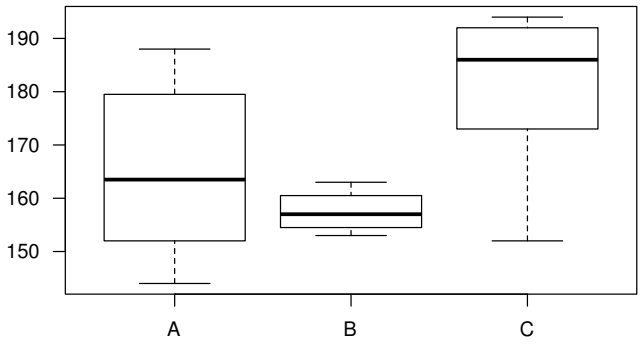
Rozwiązanie. Niech m_1, m_2 i m_3 oznaczają, odpowiednio, mediany rozkładów, z których pochodzą rozważane trzy próbki. Zadanie weryfikacji hipotezy o jednakowej efektywności metod A, B i C ujmiemy formalnie jako problem testowania hipotezy $H : m_1 = m_2 = m_3$ wobec alternatywy $K : \neg H$ orzekającej, iż co najmniej dwie z median się różnią. Do rozstrzygnięcia tego problemu użyjemy testu Kruskala-Wallisa:

```
> A <- c(147, 188, 162, 144, 157, 179, 165, 180)
> B <- c(153, 161, 157, 155, 163, 160, 154)
> C <- c(173, 152, 194, 186, 166, 194, 178, 192, 186)
> kruskal.test(list(A,B,C))

Kruskal-Wallis rank sum test

data:  list(A, B, C)
Kruskal-Wallis chi-squared = 7.7436, df = 2, p-value = 0.02082
```

Przyjmując poziom istotności 0.05 odrzucamy hipotezę zerową, czyli stwierdzamy, że rozważane metody nie są jednakowo efektywne.

Warto również zilustrować uzyskane wyniki za pomocą wykresu skrzynkowego porównującego na jednym obrazku rozkłady trzech próbek:

```
> boxplot(list(A,B,C), names=c("A", "B", "C"), las=1)
```



Wyciągnięcie wniosków płynących z porównania powyższych wykresów pozostawiamy Czytelnikowi. ⊡

Zadanie 6.11. Spośród studentów czterech wydziałów, na których pan Iksiński wykłada najciekawszy przedmiot świata (osobom niezorientowanym wyjaśniamy, że mowa tu oczywiście o statystyce matematycznej), pobrano próbki losowe i zliczono studentów (zwanych dalej „szczęśliwcami"), którym udało się zdać egzamin z tego przedmiotu. Wyniki zamieszczono w poniższej tabeli:

Wydział	Liczność próbki	Liczba szczęśliwców
Nauk niepotrzebnych	206	61
Mniemanologii stosowanej	164	34
Nauk ciekawych	98	38
Nauk przydatnych	102	35

Czy w świetle zebranych danych można stwierdzić, że występują istotne różnice między odsetkami osób na poszczególnych wydziałach, które zdały statystykę? Przyjmij poziom istotności $\alpha = 0.05$.

Rozwiązanie. Niech p_i, gdzie $i = 1, \ldots, 4$, oznacza prawdopodobieństwo zdania egzaminu ze statystyki przez studenta i-tego wydziału. Testujemy hipotezę $H : p_1 = p_2 = p_3 = p_4$ przeciw hipotezie alternatywnej $K : \neg H$. Posłużymy się w tym celu testem jednorodności chi-kwadrat:

```
> prop.test(c(61,34,38,35), c(206,164,98,102))

4-sample test for equality of proportions without continuity
correction
```

```
data:  c(61, 34, 38, 35) out of c(206, 164, 98, 102)
X-squared = 11.2601, df = 3, p-value = 0.0104
alternative hypothesis: two.sided
sample estimates:
   prop 1    prop 2    prop 3    prop 4
0.2961165 0.2073171 0.3877551 0.3431373
```

Na zadanym poziomie istotności odrzucamy hipotezę zerową, co oznacza, że występują istotne różnice między odsetkami studentów, które zdały egzamin ze statystyki na poszczególnych wydziałach.

□

6.3. Zadania do samodzielnego rozwiązania

Zadanie 6.12. W losowo wziętym tygodniu wydarzyło się w Warszawie 414 wypadków i kolizji drogowych, przy czym ich rozkład w poszczególnych dniach tygodnia wyglądał następująco:

Dzień	Pon.	Wt.	Śr.	Czw.	Pt	Sob.	Ndz
Liczba wypadków	78	56	52	58	83	42	45

Zbadaj, czy rozkład liczby wypadków w poszczególne dni tygodnia jest równomierny. Przyjmij poziom istotności 0,05.

Zadanie 6.13. W celu zbadania, czy program generujący liczby losowe z rozkładu dwumianowego o parametrach 3 i 0,5 działa prawidłowo, wygenerowano 100 liczb i otrzymano następujące wyniki:

Wygenerowana liczba losowa	0	1	2	3
Liczba uzyskanych wyników	12	37	38	13

Zweryfikuj odpowiednią hipotezę na poziomie istotności 0,05.

Zadanie 6.14. Policzono liczbę błędów drukarskich na 100 losowo wziętych stronach encyklopedii i otrzymano następujące wyniki:

Liczba błędów	Liczba stron
0	50
1	36
2	14

Czy można uznać, że rozkład liczby błędów na stronie jest rozkładem Poissona? Przyjmij poziom istotności 0,05.

Zadanie 6.15. Zweryfikuj hipotezę, że poniższa próbka pochodzi z rozkładu wykładniczego o wartości oczekiwanej 100.

$$112, 66, 81, 124, 140, 72, 155, 94, 145, 116.$$

Przyjmij poziom istotności 0,05.

Zadanie 6.16. Dla 200 próbek betonu przeprowadzono badanie wytrzymałości na ściskanie i otrzymano wyniki (w MPa):

Wytrzymałość	Liczba próbek
$19-20$	10
$20-21$	26
$21-22$	56
$22-23$	64
$23-24$	30
$24-25$	14

Zweryfikuj hipotezę głoszącą, że wytrzymałość na ściskanie ma rozkład normalny. Przyjmij poziom istotności $\alpha = 0.05$.

Zadanie 6.17. Na podstawie danych zawartych w pliku `samochody.csv`, zweryfikuj przypuszczenie, że rozkład przyspieszenia samochodów o wadze 2500–3000 funtów jest normalny (wykorzystaj zmienne `przysp` i `waga`). Czy można twierdzić, że przeciętne przyspieszenie tych samochodów przekracza 15 ft/s^2? Przyjmij poziom istotności 0.01.

Zadanie 6.18. Postanowiono zbadać zdawalność egzaminu na prawo jazdy w różnych województwach. W ramach prowadzonego badania wylosowano 100 osób, które w ostatnim roku ubiegały się o prawo jazdy w województwie mazowieckim i okazało się, że tylko 25 spośród nich zdało egzamin przy pierwszym podejściu. Na 80 wybranych losowo osób w województwie łódzkim egzamin przy pierwszym podejściu zdało 21 osób. Natomiast dla województwa małopolskiego i dolnośląskiego liczność próby oraz liczba tych, którzy przy pierwszym podejściu zdali egzamin wyniosły, odpowiednio, 110 i 40 osób oraz 90 i 29 osób. Na poziomie istotności 0,05 stwierdź, czy frakcje osób, które zdały przy pierwszym podejściu egzamin na prawo jazdy w tych czterech województwach różnią się istotnie

Zadanie 6.19. Wytrzymałość pewnych elementów konstrukcji lotniczej zależy w dużym stopniu od zawartości tytanu w stopie, z którego te elementy są wykonane. Przeciętna zawartość tytanu w stopie o pożądanych własnościach powinna wynosić 8,5%. Poniższe dane przedstawiają zawartość tytanu (w procentach) w 20 losowo wziętych próbkach:

8,32; 8,05; 8,93; 8,65; 8,25; 8,46; 8,52; 8,35; 8,36; 8,41; 8,42; 8,30; 8,71; 8,75; 8,60; 8,83; 8,50; 8,38; 8,29; 8,46.

1. Posługując się testem znaków stwierdź, czy stop, z którego zostały pobrane próbki, spełnia postawione wymagania jakościowe. Przyjmij poziom istotności 0,05.
2. Zakładając, że rozkład zawartości tytanu w stopie jest ciągły i symetryczny, zweryfikuj rozważaną hipotezę za pomocą testu rangowanych znaków.

Zadanie 6.20. W celu porównania dwóch układów wtrysku paliwa w silniku wysokoprężnym przeprowadzono następujący eksperyment. W silnikach 12 losowo wybranych samochodów zainstalowano najpierw jeden z układów wtrysku, po czym zmierzono zużycie paliwa na ustalonym dystansie. Następnie w tych samych samochodach zmieniono układ wtrysku paliwa na układ drugiego typu i ponownie zmierzono zużycie paliwa na tym samym dystansie. Otrzymane wyniki (w mpg) przedstawia poniższa tabela

Samochód	1	2	3	4	5	6
Układ I	17,6	19,4	19,5	17,1	15,3	15,9
Układ II	16,8	20,0	18,2	16,4	16,0	15,4
Samochód	7	8	9	10	11	12
Układ I	16,3	18,4	17,3	19,1	17,8	18,2
Układ II	16,5	18,0	16,4	20,1	16,7	17,9

1. Posługując się testem znaków stwierdź, czy występują istotne różnice w przeciętnym zużyciu paliwa między samochodami wyposażonymi w układy wtrysku paliwa obu typów. Przyjmij poziom istotności 0,05.
2. Zakładając, że spełnione są wymagane założenia, zweryfikuj rozważaną hipotezę za pomocą testu rangowanych znaków.

6.4. Wskazówki i odpowiedzi do zadań

Ad zad. 6.12. Test zgodności chi-kwadrat: wartość statystyki testowej to $T = 25.0386$, zaś p-wartość $= 0.0003359$. Zatem odrzucamy hipotezę o równomierności rozkładu liczby wypadków.

Ad zad. 6.13. Test zgodności chi-kwadrat: wartość statystyki testowej to $T = 0.0533$, zaś p-wartość $= 0.9968$. Zatem nie mamy podstaw do odrzucenia hipotezy, że generator działa prawidłowo.

Ad zad. 6.15. Test zgodności Kołmogorowa: wartość statystyki testowej to $T = 0.4831$, zaś p-wartość $= 0.01137$. Zatem, na poziomie istotnośći 0.05 nie możemy przyjąć, że próba pochodzi z rozkładu wykładniczego Exp(0,01).

Ad zad. 6.16. Posłuż się testem zgodności chi-kwadrat. W tym celu rozpatrz następujące klasy wartości wytrzymałości: $(-\infty, 20], (20, 21], (21, 22], \ldots, (25, +\infty)$. Wyestymuj z próby wartość oczekiwaną i odchylenie standardowe. Następnie, zakładając

że zmienna losowa X, oznaczająca wytrzymałość, ma rozkład normalny o wyestymowanych parametrach, wyznacz prawdopodobieństwa przyjęcia przez X wartości należącej do poszczególnych klas, np. $\mathrm{P}(20 < X \leqslant 21)$.

Ad zad. 6.17. Test normalności Shapiro-Wilka: wartość statystyki testowej to $T = 0.858$, zaś p-wartość $= 0.8471$. Zatem możemy przyjąć, że próba pochodzi z rozkładu normalnego.

Do weryfikacji hipotezy $H : \mu = 15$ przeciw $K : \mu > 15$ wykorzystujemy test t. Wartość statystyki testowej to $T = 2.5702$, zaś p-wartość $= 0.006812$. Zatem odrzucamy H na rzecz K, co oznacza, że przeciętne przyspieszenie tych samochodów przekracza 15 ft/s^2.

Testowanie niezależności i analiza regresji

7

7.1. Wprowadzenie

7.1.1. Test niezależności

Do weryfikacji hipotezy o niezależności dwóch cech badanej populacji służy np. test niezależności chi-kwadrat (ang. *chi-square test*). Jest on przeprowadzany na podstawie danych zapisanych w tablicy kontyngencji. W R test ten jest zaimplementowany pod postacią funkcji `chisq.test()`. Jako argument tej funkcji podajemy odpowiednią tablica kontyngencji.

Test chi-kwadrat jest testem asymptotycznym. Mając do dyspozycji niedużą liczbę obserwacji można użyć tzw. dokładnego testu Fishera (ang. *exact Fisher test*), dostępnego poprzez funkcję `fisher.test()`.

7.1.2. Współczynniki korelacji liniowej i rangowej

W środowisku R współczynniki korelacji wyznaczamy za pomocą funkcji `cor()`. Domyślnie wyliczany jest próbkowy współczynnik korelacji liniowej Pearsona. Wektory `x` i `y`, zawierające obserwacje należące do badanych prób, podajemy jako pierwszy i drugi argument tej funkcji.

Jeśli chcemy obliczyć współczynnik korelacji rangowej Spearmana bądź Kendalla, jako trzeci argument funkcji `cor()`, podajemy, odpowiednio, `method="spearman"` albo `method="kendall"`.

Do weryfikacji hipotezy o istotności współczynnika korelacji liniowej ϱ, tzn. $H : \varrho = 0$ przeciw $K : \varrho \neq 0$, służy funkcja `cor.test()`. Wektory zawierające obserwacje należące do badanych prób, na podstawie których przeprowadzamy test, podajemy jako pierwszy i drugi argument tej funkcji. Jeśli chcemy testować istotność współczynnika korelacji rangowej Spearmana albo Kendalla, jako trzeci argument podajemy, odpowiednio, `method="spearman"` bądź `"kendall"`.

7.1.3. Regresja prosta liniowa

Rozważmy model regresji liniowej postaci

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon, \tag{7.1}$$

gdzie Y jest tzw. zmienną objaśnianą (zależną), $X_1, \ldots, X_n$ – zmiennymi objaśniającymi (niezależnymi), zaś ε odpowiada błędowi losowemu.

Estymatory współczynników $\beta_0, \beta_1, \ldots, \beta_n$ w równaniu regresji wyznaczamy za pomocą funkcji `lm()` (ang. *linear model*). Ponadto procedura ta umożliwia weryfikację dopasowania modelu.

Pierwszym argumentem funkcji `lm()` jest tzw. *formuła* opisująca model, tzn. symboliczny opis zależności między zmiennymi. Jej składnia jest następująca:

$$\mathtt{Y} \sim \mathtt{X_1} + \mathtt{X_2} + \cdots + \mathtt{X_n}. \tag{7.2}$$

Więcej informacji na temat formuł znajdziemy w dokumentacji: `?lm`.

Nas w dalszym ciągu interesować będzie przypadek $n = 1$, zwany regresją liniową prostą. Zmienną zależną wyrażamy jako funkcję liniową jednej zmienne niezależnej:

$$Y = a + bX + \varepsilon. \tag{7.3}$$

Informację o dopasowanym modelu, zwracanym przez `lm()`, można zapisać jako zmienną, co z kolei pozwala na wykonywanie na tej zmiennej m.in. następujących funkcji: `summary()` – informacje szczegółowe, `predict()` – prognozowanie, `plot()` – diagnostyka za pomocą wykresów.

Po wpisaniu wektorów danych, oznaczonych przykładowo jako wektory `x` i `y`, funkcję `lm()` w zagadnieniu regresji liniowej prostej wywołuje się następująco:

```
> lm(y~x)
```

Jeśli zaobserwowane wartości `x` i `y` przechowywane są jako kolumny w pewnej ramce danych, to nazwę tej ramki podajemy jako argument parametru `data` funkcji `lm()`, np.

```
> lm(y~x, data=dane) # co jest równoważne następującemu zapisowi:
> lm(dane$y~dane$x)
```

Aby otrzymać pełen opis dopasowania modelu, używamy funkcji `summary()`.

```
> model <- lm(y~x)
> summary(model)
```

Możemy też odwoływać się bezpośrednio do wybranych elementów tego opisu, np.

```
> model$coefficients # oszacowania współczynników równania regresji
> model$residuals    # reszty
> opis <- summary(model)
> opis$r.squared          # współczynnik determinacji R^2
```

Funkcja `predict()` służy do prognozowania wartości zmiennej zależnej na podstawie dopasowanego wcześniej modelu. Gdy, na przykład, chcemy przewidzieć wartość zmiennej objaśnianej Y dla zmiennej objaśniającej $x = 8$, możemy napisać:

```
> nowy <- data.frame(x=8)
> predict(model, nowy) # prognoza y dla nowy$x na podstawie 'model'
```

7.2. Zadania rozwiązane

Zadanie 7.1. W celu zbadania, czy istnieje związek pomiędzy dochodem i posiadanym wykształceniem, przeprowadzono badanie na 450 osobowej próbie losowej i otrzymano następujące wyniki:

	Roczny dochód (tys. zł.)		
	poniżej 100	$10-200$	powyżej 200
Wykształcenie wyższe	80	115	55
Brak ukończonych studiów	95	70	35

Zweryfikuj odpowiednią hipotezę na poziomie istotności $\alpha = 0.01$.

Rozwiązanie. Do weryfikacji hipotezy H : *nie ma związku między dochodem i posiadanym doświadczeniem*, przeciw hipotezie K : *badane cechy są zależne*, użyjemy testu niezależności chi-kwadrat. Wymaga on zapisania danych w formie tzw. tablicy kontyngencji:

```
> ww <- c(80, 115, 55)
> bw <- c(95, 70, 35)
> (ct <- rbind(ww, bw)) # zob. ?rbind
   [,1] [,2] [,3]
ww   80  115   55
bw   95   70   35
```

po czym możemy już przejść do testu chi-kwadrat:

```
> chisq.test(ct)

Pearson's Chi-squared test

data:  ct
X-squared = 11.2596, df = 2, p-value = 0.003589
```

Otrzymana p-wartość wskazuje na istnienie związku między wykształceniem i osiąganymi dochodami. □

Informacja

Warto pamiętać, że test niezależności chi-kwadrat orzeka wyłącznie o istnieniu bądź nieistnieniu zależności między badanymi cechami, a nie o charakterze ewentualnego związku. Oznacza to, w szczególności, że odrzucenie hipotezy zerowej o niezależności cech nie oznacza tym samym, iż jedna z cech wpływa bezpośrednio na drugą (być może istnieje jeszcze inna zmienna, zwana ukrytą, która ma wpływ na obie badane zmienne). Odnosząc to do naszego przykładu, choć udało się nam stwierdzić związek między wykształceniem i dochodem, nie należy z tego automatycznie wyciągać np. wniosku, że dochody rosną wraz z poziomem wykształcenia.

Nawet, gdyby faktycznie tak było w badanej populacji, nie byłby to wniosek płynący wprost z testu chi-kwadrat.

Zadanie 7.2. Wygeneruj 200-elementową próbę z rozkładu dwuwymiarowego normalnego

1. $\mathrm{N}_2(0, 1, 0, 1, 0)$,
2. $\mathrm{N}_2(0, 1, 2, 1, 0.6)$.

Dla uzyskanych próbek oszacuj współczynniki korelacji, a następnie zweryfikuj hipotezę o niezależności zmiennych X i Y oraz o istotności współczynnika korelacji liniowej.

Rozwiązanie. Realizacje wektora losowego (X, Y) z rozkładu $\mathrm{N}_2(0, 1, 0, 1, 0)$ można stworzyć generując dwa niezależne wektory danych z rozkładów $\mathrm{N}(0, 1)$:

```
> n <- 200
> x <- rnorm(n)
> y <- rnorm(n)
```

Wartości współczynników korelacji Pearsona, Spearmana i Kendalla wyznacza się za pomocą funkcji `cor()`:

```
> cor(x, y)
[1] 0.0584777
> cor(x, y, method="pearson") # to samo co wyżej
[1] 0.0584777
> cor(x, y, method="spearman")
[1] 0.0561689
> cor(x, y, method="kendall")
[1] 0.0358794
```

Test istotności współczynnika korelacji liniowej Pearsona, tzn. zadanie weryfikacji hipotezy $H : \rho = 0$ względem $K : \rho \neq 0$, przeprowadza się w następujący sposób:

```
> cor.test(x, y)

Pearson's product-moment correlation

data:  x and y
t = 0.8243, df = 198, p-value = 0.4108
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.08091984  0.19563150
sample estimates:
      cor
0.0584777
```

Informacja

Pamiętajmy, że przeprowadzony powyżej test istotności współczynnika korelacji liniowej Pearsona zakłada normalność rozkładu (dwuwymiarowego), z którego pochodzą obserwacje.

Generowanie próbek wektora losowego (X, Y) w przypadku, gdy zmienne X i Y są zależne, wymaga odpowiedniego modelowania danego rodzaju zależności. W przypadku dwuwymiarowego rozkładu normalnego jest to zadanie stosunkowo proste, bowiem całkowita informacja dotycząca relacji między zmiennymi brzegowymi jest zawarta we współczynniku korelacji.

Informacja

Niech U i V oznaczają dwie niezależne zmienne losowe o rozkładzie $\mathrm{N}(0,1)$. Można pokazać, że wówczas wektor losowy (X, Y), dla którego $X = \mu_1 + \sigma_1 U$ oraz $Y = \mu_2 + \sigma_2 \left(\varrho U + \sqrt{1-\varrho^2} V\right)$, ma rozkład $\mathrm{N}_2(\mu_1, \sigma_1, \mu_2, \sigma_2, \varrho)$.

Korzystając z przytoczonego wyżej stwierdzenia wygenerujmy próbkę z rozkładu $\mathrm{N}_2(0, 1, 2, 1, 0.6)$.

```
> u <- rnorm(n)
> v <- rnorm(n)
> x2 <- u
> y2 <- 2+(0.6*u+sqrt(1-0.6^2)*v)
```

Informacja

Macierz kowariancji rozkładu $\mathrm{N}_2(\mu_1, \sigma_1, \mu_2, \sigma_2, \varrho)$ ma postać:

$$\mathbf{C} = \begin{bmatrix} \sigma_1^2 & \varrho\sigma_1\sigma_2 \\ \varrho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}, \tag{7.4}$$

czyli w naszym przypadku:

```
> (C <- matrix(c(1, 0.6, 0.6, 1), nrow=2))
     [,1] [,2]
[1,]  1.0  0.6
[2,]  0.6  1.0
```

Do wygenerowania n-elementowej próby z rozkładu $\mathrm{N}_2(0, 1, 2, 1, 0.6)$ można również wykorzystać funkcję `mvrnorm()` z pakietu MASS.

```
> library(MASS)
> mvrnorm(n, c(0, 2), C)
```

Wyznaczamy, jak poprzednio, wartości współczynników korelacji – tym razem dla naszej nowej próbki:

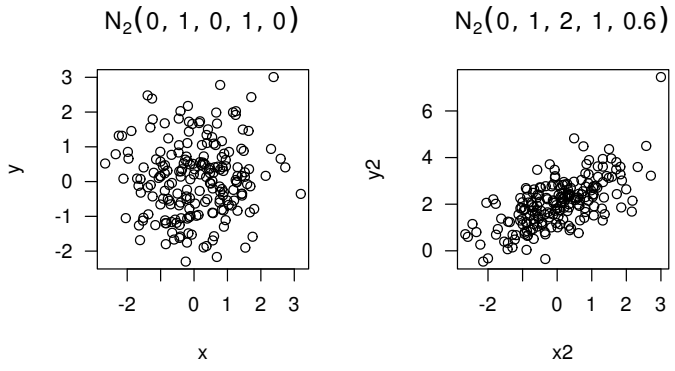
```
> cor(x2, y2)
[1] 0.6475526
> cor(x2, y2, method="spearman")
[1] 0.6279967
> cor(x2, y2, method="kendall")
[1] 0.4510553
```

oraz przeprowadzamy test istotności współczynnika korelacji liniowej Pearsona:

```
> cor.test(x2, y2)

Pearson's product-moment correlation

data:  x2 and y2
t = 11.9575, df = 198, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5590366 0.7214748
sample estimates:
      cor
0.6475526
```

Na zakończenie porównajmy wykresy rozproszenia obu prób:

```
> par(mfrow=c(1,2))
> plot(x, y, main=expression(N[2](0,1, 0,1, 0.0)), las=1)
> plot(x2, y2, main=expression(N[2](0,1, 2,1, 0.6)), las=1)
```



□

Zadanie 7.3. Dwaj profesorowie postanowili ocenić zdolności swoich 11 studentów dyplomantów. W tym celu uszeregowali oni studentów od najzdolniejszego do najmniej zdolnego. Numery w poniższej tabeli wskazują rangi nadane poszczególnym studentom

przez każdego z profesorów.

Student	A	B	C	D	E	F	G	H	I	J	K
Profesor X	1	7	8	3	6	10	9	2	11	4	5
Profesor Y	4	8	10	1	5	9	11	3	7	2	6

Czy można uznać, że istnieje zależność między opiniami obu profesorów? Przyjmij poziom istotności 0.05.

Rozwiązanie. W niniejszym zadaniu mamy do czynienia z porównaniem dwóch systemów preferencji (rankingów). Do analizy tego typu danych okazuje się przydatny współczynnik korelacji rangowej Spearmana. Tak więc po wprowadzeniu danych

```
> x <- c(1, 7, 8, 3, 6, 10, 9, 2, 11, 4, 5)
> y <- c(4, 8, 10, 1, 5, 9, 11, 3, 7, 2, 6)
```

obliczamy wartość tego współczynnika:

```
> cor(x, y, method="spearman")
[1] 0.7909091
```

Otrzymana wartość sugeruje na stosunkowo silną korelację między opiniami obu profesorów. Dodatkowo, możemy się pokusić o test istotności dla współczynnika korelacji rangowej Spearmana:

```
> cor.test(x, y, method="spearman")

Spearman's rank correlation rho

data:  x and y
S = 46, p-value = 0.006061
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.7909091
```

Oczywiście, na poziomie istotności $\alpha = 0.05$ odrzucamy hipotezę o braku zależności między badanymi zmiennymi. Reasumując, można więc przyjąć, że istnieje istotny, dodatni związek między opiniami przez profesorów. ⊡

Zadanie 7.4. W zamieszczonej poniżej tabeli podano wysokość rocznego dochodu (w tys. zł.) dziewięciu rodzin wybranych w sposób losowy spośród mieszkańców pewnego osiedla oraz wartość domu (w mln zł.) posiadanego przez dana rodzinę:

Dochód	360	640	490	210	280	470	580	190	320
Wartość domu	1.49	3.10	2.60	0.92	1.26	2.42	2.88	0.81	1.34

1. Wyznacz prostą regresji wartości domu względem dochodu.
2. Przeanalizuj dopasowanie modelu.
3. Oszacuj wartość domu rodziny, której roczny dochód wynosi 400 000 zł.

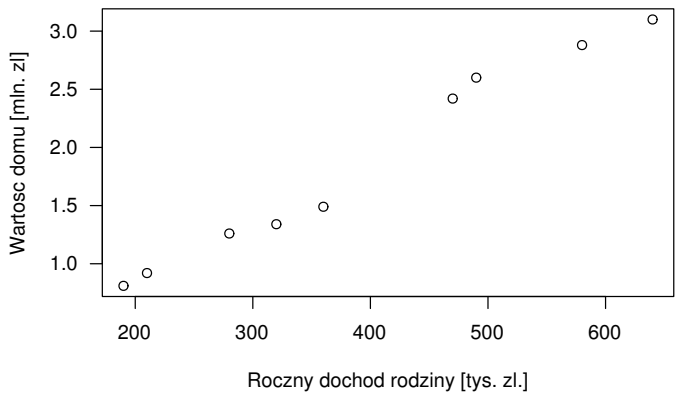
4. Wyznacz 95% przedział ufności dla szacowanej wartości domu tej rodziny.

Rozwiązanie. Zaraz po wprowadzeniu danych

```
> dochod <- c(360, 640, 490, 210, 280, 470, 580, 190, 320)
> dom    <- c(1.49, 3.10, 2.60, 0.92, 1.26, 2.42, 2.88, 0.81, 1.34)
```

sprawdzamy na wykresie, czy zależność między zmienną niezależną X (dochód), a objaśnianą Y (wartość domu) jest typu liniowego:

```
> plot(dochod, dom, xlab="Roczny dochod rodziny [tys. zl.]",
+      ylab="Wartosc domu [mln. zl]", las=1)
```



Uzyskany wykres sugeruje, że rozpatrywanie w tym przypadku zależności liniowej między badanymi zmiennymi, wydaje się być uzasadnione. Dopasowujemy zatem model prostej regresji liniowej:

```
> (opis <- summary(modl <- lm(dom~dochod)))

Call:
lm(formula = dom ~ dochod)

Residuals:
      Min        1Q    Median        3Q       Max
-0.197759 -0.109247  0.006951  0.047323  0.205835

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.2684395  0.1281678  -2.094   0.0745 .
dochod       0.0054339  0.0003042  17.863 4.25e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.1378 on 7 degrees of freedom
Multiple R-squared:  0.9785,Adjusted R-squared:  0.9755
F-statistic: 319.1 on 1 and 7 DF,  p-value: 4.254e-07
```

Wiersz zaczynający się od `(Intercept)` zawiera estymator $\hat{a}$ wyrazu wolnego oraz pewne informacje dotyczące tego estymatora. Podobnie w wierszu zaczynającym się od nazwy zmiennej objaśniającej (w naszym przypadku `dochod`) – współczynnik $\hat{b}$ stojący przy zmiennej X wraz ze stosownymi informacjami. Wyestymowane parametry modelu $Y = a + bX + \varepsilon$ możemy odczytać z kolumny `Estimate` tabelki `Coefficients`, bądź za pomocą wywołania:

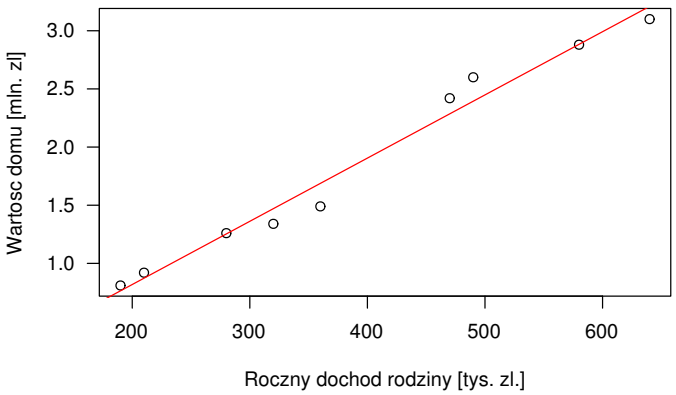
```
> modl$coefficients
 (Intercept)       dochod
-0.268439463  0.005433886
```

Tak więc poszukiwana funkcja regresji przyjmuje postać:

$$\hat{Y} = -0.2684 + 0.0054X.$$

A oto wykres obserwacji wraz z otrzymaną prostą regresji:

```
> plot(dochod, dom, xlab="Roczny dochod rodziny [tys. zl.]",
+      ylab="Wartosc domu [mln. zl]", las=1)
> abline(modl, col="red")
```



Informacja

Pamiętajmy, że zaimplementowana funkcja wyestymuje „optymalny" (w sensie najmniejszych kwadratów) model regresji liniowej, nawet w sytuacji, gdy rozpatrywanie tego typu zależności jest ewidentnie pozbawione sensu. Stąd też estymacja parametrów modelu jest wyłącznie pierwszym krokiem analizy regresji. Niezbęd-

nym, drugim krokiem, powinna być weryfikacja poprawności modelu.

Weryfikację poprawności modelu możemy przeprowadzić, odczytując i interpretując wyniki zwracane przez funkcję `summary()`.

Analiza współczynnika determinacji. Współczynnik R^2 wskazuje, jaka część zmienności zmiennej zależnej jest wyjaśniana przez rozpatrywany model:

```
> opis$r.squared
[1] 0.9785324
```

Test istotności dla współczynnika korelacji. Testujemy hipotezę $H : \rho = 0$ wobec $K : \rho \neq 0$:

```
> cor.test(dochod, dom)

Pearson's product-moment correlation

data:  dochod and dom
t = 17.8626, df = 7, p-value = 4.254e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9476475 0.9978124
sample estimates:
      cor
0.9892079
```

Test F (analiza wariancji w analizie regresji). Celem tego testu jest stwierdzenie adekwatności modelu. Formalnie rzecz biorąc rozpatrujemy hipotezę zerową $H : b = 0$ (nie ma zależności liniowej między zmiennymi) przeciwko $K : b \neq 0$:

```
> opis$fstatistic
   value    numdf    dendf
319.0721   1.0000   7.0000
> anova(modl)
Analysis of Variance Table

Response: dom
          Df Sum Sq Mean Sq F value    Pr(>F)
dochod     1 6.0590   6.059  319.07 4.254e-07 ***
Residuals  7 0.1329   0.019
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> anova(modl)$"Pr(>F)"[1] # p-value
[1] 4.254379e-07
```

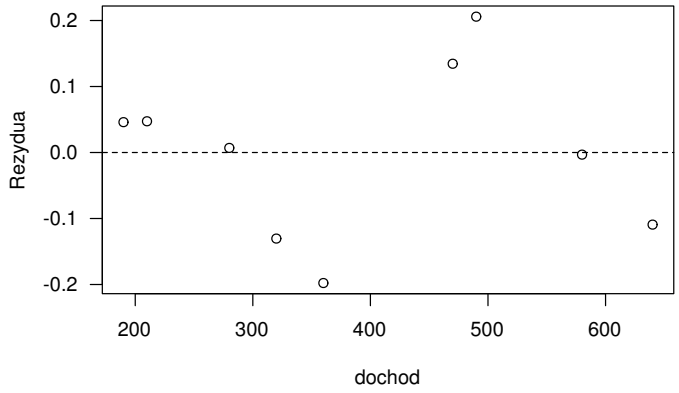
Test t (test istotności współczynników regresji). Testujemy hipotezy o nieistotności współczynników przy alternatywie, że są one istotnie różne od zera, tzn. $H_1 : a = 0$

przeciwko $K_1 : a \neq 0$ oraz $H_2 : b = 0$ względem $K_2 : b \neq 0$:

```
> opis$coefficients
                Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) -0.268439463 0.1281677764 -2.094438 7.448037e-02
dochod       0.005433886 0.0003042048 17.862589 4.254379e-07
```

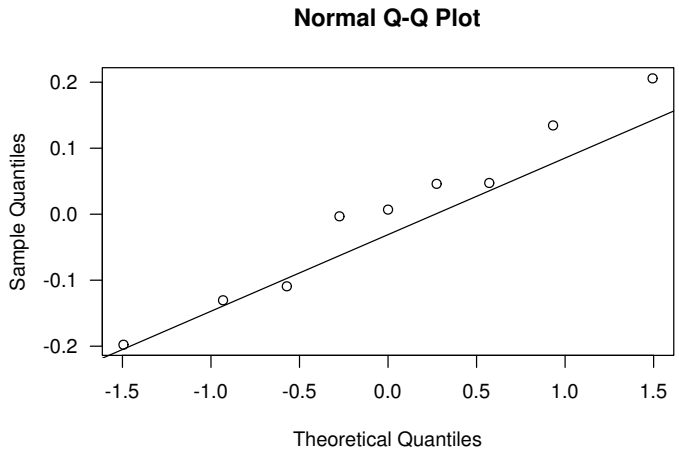
Analiza reszt. Wykreślamy wykres reszt w zależności od X:

```
> plot(dochod, modl$residuals, ylab='Rezydua', las=1)
> abline(h=0, lty=2)
```



Narysujmy jeszcze wykres normalności reszt:

```
> qqnorm(modl$residuals, las=1)
> qqline(modl$residuals)
```



i zweryfikujmy hipotezę o normalności rozkładu reszt testem Shapiro-Wilka:

```
> shapiro.test(modl$residuals)

Shapiro-Wilk normality test

data:  modl$residuals
W = 0.9706, p-value = 0.9002
```

Reasumując, biorąc pod uwagę wyniki wszystkich wspomnianych wyżej procedur służących weryfikacji poprawności modelu, możemy stwierdzić, że posługiwanie się w rozważanej przez nas sytuacji modelem liniowym jest uzasadnione.

Dysponując już pozytywnie zweryfikowanym modelem regresji możemy wykorzystać ów model np. do prognozowania.

W naszym zadaniu chcielibyśmy oszacować wartość domu rodziny mającej dochód równy 400 tys. zł. W tym celu wywołujemy następującą funkcję:

```
> (nowe <- data.frame(dochod=400)) # uwaga na nazwę kolumny!
  dochod
1    400
> predict(modl, nowe)
       1
1.905115
```

Oprócz wartości estymatora punktowego warto również wyznaczyć przedziały ufności dla prognozy. W naszym przypadku 95% przedział ufności dla prognozy ma postać:

```
> # przedział dla predykcji:
> predict(modl, nowe, interval="prediction", level=0.95)
       fit      lwr      upr
1 1.905115 1.561606 2.248624
> # przedział dla wartości średniej:
> predict(modl, nowe, interval="confidence", level=0.95)
       fit      lwr      upr
1 1.905115 1.796393 2.013837
```

Granice przedziałów odczytujemy z kolumn `lwr` (ang. *lower* – dolna) i `upr` (ang. *upper* – górna). ⊡

Zadanie 7.5. W poniższej tabeli podano liczbę ludności USA (w milionach) w latach 1890-2007:

Rok	1890	1900	1910	1920	1930	1940	1950
Ludność	62 947	75 994	91 972	105 710	122 775	131 669	150 697
Rok	1960	1970	1980	1990	2000	2007	
Ludność	179 323	203 235	226 542	248 718	281 422	301 140	

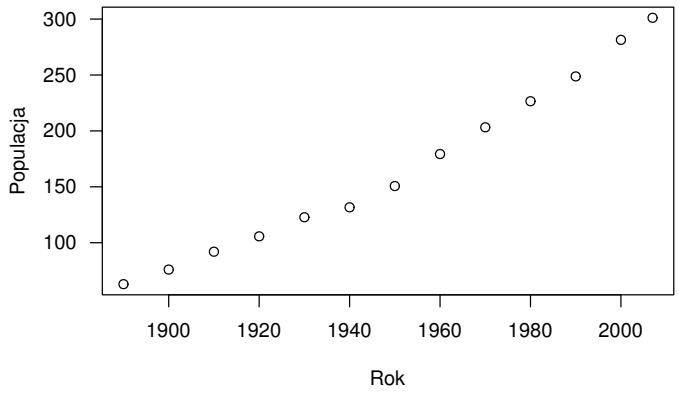
1. Przyjmując wykładniczy model wzrostu populacji, oszacuj parametry tego modelu i zweryfikuj jego dopasowanie.
2. Oszacuj przewidywaną wielkość populacji USA w 2020 i w 2025 roku.

Rozwiązanie. Wprowadźmy dane wejściowe:

```
> rok <- c(1890, 1900, 1910, 1920, 1930, 1940, 1950,
+    1960, 1970, 1980, 1990, 2000, 2007)
> pop <- c(62.947, 75.994, 91.972, 105.710, 122.775, 131.669, 150.697,
+    179.323, 203.235, 226.542, 248.718, 281.422, 301.140)
```

i sporządźmy dla nich wykres:

```
> plot(rok, pop, xlab="Rok", ylab="Populacja", las=1)
```



Informacja

Zadania regresji nieliniowej rozwiązuje się dokonując uprzedniej linearyzacji modelu. Oznacza ona takie przekształcenie wyjściowego modelu, które sprowadzi go do funkcji liniowej. Dysponując już pożądanym przekształceniem transformujemy za jego pomocą dane, a następnie stosujemy dobrze nam już znaną funkcję `lm()`. Należy pamiętać, że wyznaczone przez tę funkcję estymatory współczynników regresji mogą nieraz wymagać odpowiedniego przekształcenia (odwrotnego do transformacji prowadzonej w celu linearyzacji).

Dopasowanie do danych funkcji wykładniczej $y = \exp(a + bx)$ poprzedzimy linearyzacją modelu wykładniczego: $z := \ln(y)$ i stąd $z = a + bx$. Dla przekształconych we wskazany sposób danych otrzymujemy:

```
> logpop <- log(pop)
> usa    <- lm(logpop~rok)
> summary(usa)

Call:
lm(formula = logpop ~ rok)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.08869 -0.02785  0.00062  0.03788  0.05693

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.045e+01  6.516e-01  -31.39 4.07e-12 ***
rok          1.306e-02  3.341e-04   39.09 3.72e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04464 on 11 degrees of freedom
Multiple R-squared:  0.9929,Adjusted R-squared:  0.9922
F-statistic:  1528 on 1 and 11 DF,  p-value: 3.719e-13
```

Zwróćmy uwagę, że otrzymane wartości estymatorów współczynników regresji nie wymagają dokonania przekształcenia i wynoszą, odpowiednio:

```
> usa$coefficients
 (Intercept)          rok
-20.45450963   0.01306111
```

Tym samym otrzymaliśmy następującą funkcję regresji:

$$\hat{Y} = \exp(-20.4545 + 0.0131X).$$

Nietrudno też zauważyć, że model wykładniczy jest dobrze dopasowany do danych.

Aby dokonać prognozy wielkości populacji USA w roku 2020 i 2025, wywołujemy następującą funkcję:

```
> nowyrok <- data.frame(rok=c(2020,2025));
> (nowylogpop <- predict(usa, nowyrok, interval="prediction"))
       fit      lwr      upr
1 5.928929 5.814638 6.043219
2 5.994234 5.878235 6.110233
> exp(nowylogpop)   # przekształcenie odwrotne do log()
       fit      lwr      upr
1 375.7517 335.1700 421.2469
2 401.1093 357.1784 450.4435
```

⊡

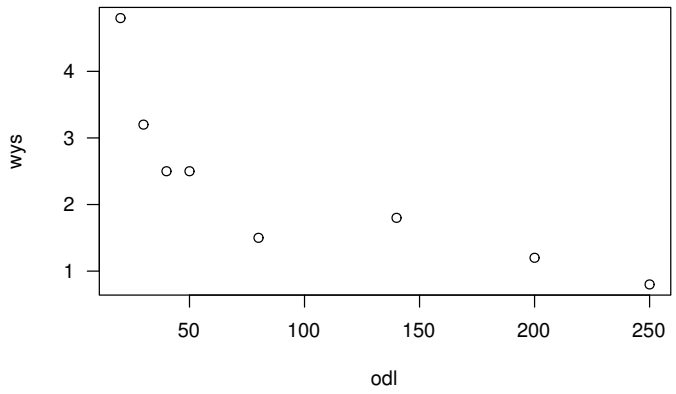
Zadanie 7.6. Dokonano ośmiu niezależnych pomiarów wielkości drgań pionowych (w cm) gruntu, powstałych w wyniku trzęsienia ziemi, w różnej odległości od epicentrum trzęsienia (w km). Otrzymano następujące wyniki:

Odległość	20	30	40	50	80	140	200	250
Wielkość drgań	4.8	3.2	2.5	2.5	1.5	1.8	1.2	0.8

1. Wyznacz funkcję regresji wielkości drgań gruntu względem odległości od epicentrum.
2. Zweryfikuj dopasowanie modelu.
3. Oszacuj wielkość drgań w odległości 100 km od epicentrum.

Rozwiązanie. Wprowadzamy dane i rysujemy wykres rozrzutu:

```
> odl <- c(20,  30,  40,  50,  80,  140, 200, 250)
> wys <- c(4.8, 3.2, 2.5, 2.5, 1.5, 1.8, 1.2, 0.8)
> plot(odl, wys, las=1)
```



Z powyższego rysunku wynika, że interesującą nas zależność możemy modelować przy użyciu różnych funkcji malejących. Przystępujemy zatem do poszukiwania najlepszego modelu. Rozpatrzymy w tym celu kilka modeli i porównamy się je ze sobą.

Zaczniemy od funkcji wykładniczej: $y = \exp(a + bx)$. Po dokonaniu linearyzacji $z := \ln(y)$ otrzymamy $z = a + bx$ i dalej stosujemy metody regresji liniowej:

```
> yp <- log(wys)
> t1 <- lm(yp~odl)
> summary(t1)

Call:
lm(formula = yp ~ odl)

Residuals:
     Min       1Q   Median       3Q      Max
-0.41291 -0.09903  0.01523  0.10161  0.38665

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.303166   0.142135   9.169 9.48e-05 ***
```

```
odl         -0.006060   0.001099  -5.516  0.00149 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2503 on 6 degrees of freedom
Multiple R-squared:  0.8353,Adjusted R-squared:  0.8078
F-statistic: 30.42 on 1 and 6 DF,  p-value: 0.001493
```

Kolejnym rozważanym przez nas modelem będzie model multiplikatywny (potęgowy), czyli interesować nas będzie funkcja postaci: $y = ax^b$. Linearyzacja tej funkcji przebiega następująco: $z := \ln(y)$, $u := \ln(x)$, $a' := \ln(a)$ i stąd otrzymujemy $z = a' + bu$. Po odpowiednim przekształceniu danych uruchamiamy znaną nam procedurę regresji liniowej:

```
> xp <- log(odl)
> t2 <- lm(yp~xp)
> summary(t2)

Call:
lm(formula = yp ~ xp)

Residuals:
     Min       1Q   Median       3Q      Max
-0.21649 -0.12952 -0.01131  0.10844  0.29685

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.21390    0.33119   9.704 6.87e-05 ***
xp          -0.59150    0.07607  -7.776 0.000238 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1853 on 6 degrees of freedom
Multiple R-squared:  0.9097,Adjusted R-squared:  0.8947
F-statistic: 60.46 on 1 and 6 DF,  p-value: 0.0002382
```

Zwróćmy uwagę, że wyestymowana wyżej wartość nie jest wartością wyrazu wolnego modelu potęgowego, ale jest to estymator a'. Zatem, aby otrzymać estymator parametru a musimy zastosować przekształcenie odwrotne, tzn. $a' := \ln(a)$.

Następnym rozważanym modelem będzie tzw. model odwrotnościowy względem zmiennej zależnej, tzn. będzie nas interesować funkcja postaci: $y = \frac{1}{a+bx}$. Linearyzacja modelu w tym przypadku sprowadza się do przekształcenia: $v := 1/y$, dzięki któremu otrzymujemy $v = a + bx$. Przekształcamy odpowiednio dane i uruchamiamy procedurę regresji liniowej:

```
> yb <- 1/wys
> t3 <- lm(yb~odl)
```

```
> summary(t3)

Call:
lm(formula = yb ~ odl)

Residuals:
      Min        1Q    Median        3Q       Max
-0.165680 -0.080000  0.003884  0.066478  0.166753

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.2048174  0.0694234   2.950 0.025604 *
odl         0.0036887  0.0005366   6.874 0.000467 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1222 on 6 degrees of freedom
Multiple R-squared:  0.8873,Adjusted R-squared:  0.8686
F-statistic: 47.25 on 1 and 6 DF,  p-value: 0.0004672
```

Jako ostatni rozpatrzymy tzw. model odwrotnościowy względem x, reprezentowany przez funkcje postaci: $y = a + \frac{b}{x}$. Linearyzacja tego modelu za pomocą funkcji $w := 1/x$ prowadzi do pożądanego modelu liniowego $y = a + bw$, co pozwala na uruchomienie procedury regresji liniowej:

```
> xb <- 1/odl
> t4 <- lm(wys~xb)
> summary(t4)

Call:
lm(formula = wys ~ xb)

Residuals:
     Min       1Q   Median       3Q      Max
-0.26756 -0.21344 -0.05194  0.15094  0.48701

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.7552     0.1656   4.559  0.00385 **
xb           78.0908     6.7037  11.649 2.41e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2847 on 6 degrees of freedom
Multiple R-squared:  0.9577,Adjusted R-squared:  0.9506
F-statistic: 135.7 on 1 and 6 DF,  p-value: 2.411e-05
```

Zwróćmy uwagę, że każdy z rozważanych wyżej modeli mógłby służyć jako model regresji wielkości drgań względem odległości od epicentrum, o czym świadczy niezłe dopasowanie udokumentowane na wydrukach (szczegóły pozostawiamy Czytelnikowi). Który z nich powinniśmy zatem wybrać? Aby odpowiedzieć na to pytanie musimy zdecydować się na jakieś rozsądne kryterium. Może nim być np. wartość współczynnika determinacji R^2. Kierując się tym właśnie kryterium stwierdzamy, że spośród zbadanych modeli najlepiej dopasowany jest ostatni, tzn. model odwrotnościowy względem x: $y = a + b/x$ o współczynnikach:

```
> t4$coefficients
(Intercept)          xb
  0.7551998  78.0908318
```

Tym samym otrzymaliśmy następującą funkcję regresji:

$$\hat{Y} = 0.7552 + \frac{78.0908}{X}.$$

Uzyskana na tej podstawie prognoza wielkości drgań w odległości 100 km od epicentrum jest następująca:

```
> pw <- data.frame(xb=1/100)
> predict(t4, pw, interval="confidence")
       fit      lwr     upr
1 1.536108 1.243547 1.82867
```

⊡

Zadanie 7.7. Metodą najmniejszych kwadratów dopasuj proste regresji opisujące zależność każdej z par zmiennych (X_i, Y_i), $i = 1, 2, 3, 4$, ze zbioru `anscombe`. Wyznacz wartości współczynników korelacji próbkowych i współczynników determinacji R^2. Sporządź wykresy rozrzutu oraz nanieść na nie wykresy dopasowanych prostych.

Rozwiązanie. Rozważany w tym zadaniu zestaw danych, zwany niekiedy kwartetem Anscombe'a (ang. *Anscombe's quartet*), składa się z czterech próbek dwuwymiarowych o niemal identycznych charakterystykach próbkowych. Został on stworzony w 1973 roku przez amerykańskiego statystyka, Francisa Anscombe'a, aby pokazać, jak ważne jest sporządzanie wykresów rozrzutu zanim przystąpi się do analizy danych. Dane te ilustrują zarazem jak obserwacje odstające (ang. *outliers*) mogą wpływać na estymatory uzyskiwane metodą najmniejszych kwadratów.

Zacznijmy od porównania średnich i wariancji zmiennych objaśniających X_i dla $i = 1, \ldots, 4$:

```
> data(anscombe)
> sapply(anscombe[c("x1", "x2", "x3", "x4")], mean)
x1 x2 x3 x4
 9  9  9  9
```

```
> sapply(anscombe[c("x1", "x2", "x3", "x4")], var)
x1 x2 x3 x4
11 11 11 11
```

Jak widać, zarówno średnie, jak i wariancje z tych próbek są sobie równe. Podobnie ma się sprawa ze zmiennymi objaśnianymi:

```
> sapply(anscombe[c("y1", "y2", "y3", "y4")], mean)
      y1       y2       y3       y4
7.500909 7.500909 7.500000 7.500909
> sapply(anscombe[c("y1", "y2", "y3", "y4")], var)
      y1       y2       y3       y4
4.127269 4.127629 4.122620 4.123249
```

gdzie różnice, w przypadku średnich, pojawiają się na czwartym, a w przypadku wariancji – na trzecim miejscu po przecinku. Wyznaczmy teraz wartości współczynników korelacji dla każdej z par (X_i, Y_i), $i = 1, \ldots, 4$:

```
> cor(anscombe$x1, anscombe$y1)
[1] 0.8164205
> cor(anscombe$x2, anscombe$y2)
[1] 0.8162365
> cor(anscombe$x3, anscombe$y3)
[1] 0.8162867
> cor(anscombe$x4, anscombe$y4)
[1] 0.8165214
```

Widzimy, że wartości współczynników korelacji Pearsona dla wszystkich czterech par zmiennych zgadzają się aż do trzeciego miejsca po przecinku. Wyznaczmy zatem również proste regresji dla każdej pary zmiennych:

```
> model1 <- lm(y1~x1, anscombe)
> summary(model1)

Call:
lm(formula = y1 ~ x1, data = anscombe)

Residuals:
     Min       1Q   Median       3Q      Max
-1.92127 -0.45577 -0.04136  0.70941  1.83882

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
x1            0.5001     0.1179   4.241  0.00217 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.237 on 9 degrees of freedom
```

```
Multiple R-squared:  0.6665,Adjusted R-squared:  0.6295
F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

```
> model2 <- lm(y2~x2, anscombe)
> summary(model2)

Call:
lm(formula = y2 ~ x2, data = anscombe)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9009 -0.7609  0.1291  0.9491  1.2691

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.001      1.125   2.667  0.02576 *
x2             0.500      0.118   4.239  0.00218 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared:  0.6662,Adjusted R-squared:  0.6292
F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

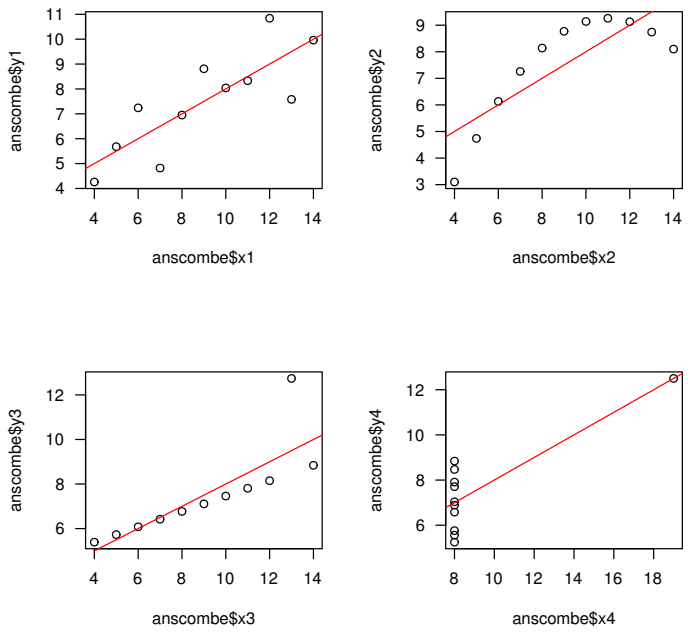
```
> model3 <- lm(y3~x3, anscombe)
> summary(model3)

Call:
lm(formula = y3 ~ x3, data = anscombe)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1586 -0.6146 -0.2303  0.1540  3.2411

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0025     1.1245   2.670  0.02562 *
x3            0.4997     0.1179   4.239  0.00218 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.236 on 9 degrees of freedom
Multiple R-squared:  0.6663,Adjusted R-squared:  0.6292
F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
```

```
> model4 <- lm(y4~x4, anscombe)
> summary(model4)
```

```
Call:
lm(formula = y4 ~ x4, data = anscombe)

Residuals:
   Min     1Q Median     3Q    Max
-1.751 -0.831  0.000  0.809  1.839

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0017     1.1239   2.671  0.02559 *
x4            0.4999     0.1178   4.243  0.00216 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.236 on 9 degrees of freedom
Multiple R-squared:  0.6667,Adjusted R-squared:  0.6297
F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

Okazuje się, że w czterech rozważanych przypadkach prosta regresji wyznaczona metodą najmniejszych kwadratów ma w przybliżeniu równanie $\hat{Y} = 3.00 + 0.005X$, przy czym różnice między wartościami estymatorów pojawiają się, odpowiednio, na trzecim i czwartym miejscu po przecinku. Współczynniki determinacji R^2 są sobie równe z dokładnością do czwartego miejsca po przecinku.

Te wszystkie zaobserwowane podobieństwa nie powinny jednakże sugerować, że mamy do czynienia z niemal identycznymi zestawami danych. Najlepiej pokażą nam to rysunki rozrzutu dla każdej pary zmiennych:



7.3. Zadania do samodzielnego rozwiązania

Zadanie 7.8. Psycholog pracujący w poradni rodzinnej zebrał dane dotyczące przyczyn kryzysów małżeńskich, które wymieniane były przez przychodzące do poradni pary. Dane te, zamieszczone w poniższej tabeli, pokazują źródła kryzysu postrzegane przez każde z małżonków.

Żona \ Mąż	Pieniądze	Dzieci	Zainteresowania	Inne
Pieniądze	86	31	132	19
Dzieci	17	64	43	13
Zainteresowania	54	39	132	33
Inne	30	17	37	54

Czy na podstawie zebranych danych można stwierdzić, że istnieje zależność między poglądami mężów i żon co do przyczyn kryzysu w ich małżeństwach? Przyjmij poziom istotności $\alpha = 0.05$.

Zadanie 7.9. Badano istnienie związku między ciśnieniem krwi a nadwagą. W poniższej tabeli zebrano dane na temat losowo wybranej grupy osób:

	Ciśn. ++	Ciśn. OK
Nadwaga	57	18
Brak nadwagi	24	91

Czy na podstawie tych danych można stwierdzić istnienie takiej zależności? Przyjmij poziom istotności 0.05.

Zadanie 7.10. Badano, czy istnieje zależność między zawodami ojców i ich dorosłych synów. W tym celu zbadano losowo wybraną grupę ojców i synów. Otrzymano następujące wyniki:

Ojciec \ Syn	Polityk	Prawnik	Lekarz
Polityk	33	48	17
Prawnik	21	38	12
Lekarz	7	8	68

Na podstawie powyższych danych stwierdzić, czy istnieje taka zależność. Przyjmij poziom istotności testu 0.01.

Zadanie 7.11. Pewien przedsiębiorca zainteresowany jest oceną ryzyka planowanych inwestycji. Dwóch zatrudnionych przez niego analityków uszeregowało planowane inwestycje od tej o największym ryzyku (10) do tej o najmniejszym (1):

Inwestycja	A	B	C	D	E	F	G	H	I	J
Analityk I	1	4	9	8	6	3	5	7	2	10
Analityk II	1	5	6	2	9	7	3	10	4	8

Czy można uznać, że istnieje zależność między opiniami obu analityków? Przyjmij poziom istotności 0.05.

Zadanie 7.12. Wyznacz prostą regresji poziomu cholesterolu względem wieku dziesięciu losowo wziętych mężczyzn. Zweryfikuj dopasowanie modelu.

Wiek	58	69	43	39	63	52	47	31	74	36
Cholesterol	189	235	193	177	154	191	213	175	198	181

Zadanie 7.13. Niech X oznacza przeciętną liczbę samochodów poruszających się autostradą w ciągu dnia (w tys.), natomiast Y – liczbę wypadków samochodowych, która miała miejsce w ciągu miesiąca na autostradzie.

X	2.0	2.3	2.5	2.6	2.8	3.0	3.1	3.4	3.7	3.8	4.0	4.6	4.8
Y	15	27	20	21	31	26	22	23	32	39	27	43	53

Na podstawie danych zamieszczonych w tabeli wyestymuj parametry funkcji regresji

$$\sqrt{Y} = a + bX,$$

opisującej zależność liczby wypadków od natężenia ruchu na autostradzie. Oszacuj liczbę wypadków, jakiej można się spodziewać przy natężeniu ruchu odpowiadającemu 3500 samochodom poruszającym się autostradą w ciągu dnia.

Zadanie 7.14. Korzystając z danych zawartych w poniższej tabeli, wyznacz funkcję regresji, opisującą zależność między liczbą cykli do zniszczenia pewnego detalu (w mln), a wywieranym na ten detal naprężeniem (w MPa):

Naprężenie	55	50.5	43.5	42.5	42	41	35.7	34.5	33	32
Cykle	0.223	0.925	6.75	18.1	29.1	50.5	126	215	445	420

Oszacuj liczbę cykli do zniszczenia detalu, na który wywierane jest naprężenie 40 MPa.

Zadanie 7.15. Badano wpływ dawki pewnego leku na puls pacjenta. Oto wyniki uzyskane dla 10 losowo wybranych osób:

Dawka leku	2	2	4	4	8	8	16	16	32	32
Puls	68	58	63	62	67	65	70	70	74	73

Dopasuj model regresji do powyższych danych.

.

Zadanie 7.16. Zbadano zależność zużycia paliwa (mile/galon) od mocy silnika 15 wybranych losowo samochodów pewnej marki. Wyniki przedstawia poniższa tabela:

Zużycie paliwa	43.1	20.3	17	21.6	16.2	31.5	31.9	25.4
Moc	48	103	125	115	133	71	71	77
Zużycie paliwa	27.2	37.3	41.5	34.3	44.3	43.4	36.4	
Moc	71	69	76	78	48	48	67	

Dopasuj optymalny model regresji do powyższych danych. Zweryfikuj jego dopasowanie. Podaj przewidywane zużycie paliwa samochodu o mocy 150.

Zadanie 7.17. Zapytano dziesięciu losowo wybranych mężczyzn, stojących pod osiedlową Pijalnią Jogurtu, ile litrów tego napoju mlecznego wypijają w ciągu tygodnia. Wyniki przedstawia poniższa tabela:

Wiek	37	42	25	14	48	78	18	34	20	57
Spożycie	3	2	4	5	1	0.3	8	2.5	7	0.5

Dopasuj wykładniczy model regresji do powyższych danych. Zweryfikuj jego dopasowanie. Podaj przewidywaną wielkość tygodniowego spożycia jogurtu przez 40-latka.

Zadanie 7.18. Poniższa tabela zawiera informacje o miesięcznych wydatkach na rozrywki i o wysokości miesięcznych dochodów 7 losowo wybranych mieszkańców pewnego miasta:

Wydatki	186	700	490	385	266	357	613
Dochody	2800	4200	3500	3150	2975	3175	3850

1. Wyznacz optymalny model regresji opisujący zależność miesięcznych wydatków na rozrywki od dochodów.
2. Zweryfikuj dopasowanie modelu.
3. Podaj przewidywaną wysokość wydatków na rozrywki osoby o miesięcznych dochodach wysokości 4000 zł.

Zadanie 7.19. Badano zależność między liczbą wypalanych dziennie papierosów a prawdopodobieństwem zachorowania na raka płuc w populacji 40-letnich palaczy, palących od 10 lat. Uzyskane dane zamieszczono w poniższej tabeli:

Liczba papierosów	5	10	20	30	40	50	60
Prawdopodobieństwo	0.061	0.113	0.192	0.259	0.339	0.401	0.461

1. Wyznacz potęgowy model regresji opisujący badaną zależność.
2. Przeanalizuj dopasowanie modelu.
3. Oszacuj prawdopodobieństwo zachorowania na raka płuc przez palacza wypalającego 35 papierosów dziennie.

7.4. Wskazówki i odpowiedzi do zadań

Ad zad. 7.12. Dopasowana liniowa funkcja regresji:

$$y = 162.7852 + 0.5433x$$

Model liniowy jest źle dopasowany: $R^2 = 0.13$.

Ad zad. 7.3. Model odwrotnościowy względem zmiennej zależnej. Dopasowana funkcja regresji:

$$y = 1/(-0.0009 + 0.00045x)$$

Dobre dopasowanie: $R^2 = 0.89$. Prognozowana wartość y dla $x = 150$ to 14.8668.

Ad zad. 7.17. Model wykładniczy. Dopasowana funkcja regresji:

$$y = \exp(2.77 - 0.05x)$$

Dobre dopasowanie: $R^2 = 0.94$. Prognozowana wartość y dla $x = 40$ to 2.15.

Ad zad. 7.18. Model odwrotnościowy względem zmiennej niezależnej. Dopasowana funkcja regresji:

$$y = 1736,91 - 4343442/x$$

Dobre dopasowanie: $R^2 = 0.99$.

Ad zad. 7.19. Model multiplikatywny. Dopasowana funkcja regresji:

$$y = \exp(-4.07449)x^{0.80771}$$

Dobre dopasowanie: $R^2 = 0.99$.

Analiza wariancji 8

8.1. Wprowadzenie

8.1.1. ANOVA

Analizę wariancji – zarówno jednoczynnikową, jak i dwuczynnikową – realizujemy w środowisku R za pośrednictwem funkcji `anova()`. Wszystkie dostępne obserwacje muszą być zapisane w postaci ramki danych, dla przykładu:

```
> dane <- data.frame(y, poziomy)
```

gdzie pierwszy argument, czyli `y`, zawiera obserwacje zmiennej objaśnianej, natomiast drugi argument, tj. `poziomy`, kody poziomów, na których występuje badany czynnik, dzięki którym możliwe jest jednoznaczne zidentyfikowanie, która z obserwacji wektora y należy do danego poziomu czynnika.

Wywołanie funkcji `anova()` wygląda następująco:

```
> anova(lm(dane$y~dane$poziomy))
```

W rezultacie otrzymujemy tzw. tablicę analizy wariancji, której kolejne kolumny zawierają liczbę stopni swobody (`Df`), sumę kwadratów odchyleń (`Sum Sq`), średni kwadrat odchyleń (`Mean Sq`), wartość statystyki testowej (`F`) oraz p-wartość testu F.

W przypadku odrzucenia hipotezy zerowej testem F, przeprowadzamy testy porównań wielokrotnych, np. metodą Tukeya. W R dokonujemy tego przy użyciu funkcji `TukeyHSD()`:

```
> TukeyHSD(aov(dane$y~dane$poziomy))
```

W przypadku dwuczynnikowej analizy wariancji, uwzględniającej interakcje między poziomami, funkcję `anova()` wywołujemy w następujący sposób:

```
> dane <- data.frame(y, A, B)
> anova(lm(dane$y~dane$A*dane$B))
```

przy czym argument y oznacza tu obserwacje zmiennej objaśnianej, natomiast `A` i `B` są wektorami kodów poziomów pierwszego i drugiego czynnika. W tym przypadku wywołanie procedury porównań wielokrotnych wyglądałoby następująco:

```
> TukeyHSD(aov(dane$y~dane$A*dane$B))
```

Do analizy ewentualnych interakcji czynników przydają się również tzw. wykresy interakcji, które uzyskujemy przy użyciu funkcji `interaction.plot()`.

8.1.2. Weryfikacja założeń ANOVY

Zanim przystąpimy do testu F powinniśmy sprawdzić, czy są spełnione założenia pozwalające posłużyć się wspomnianą metodą:

1. niezależność obserwacji,
2. normalność rozkładów w każdej z podpopulacji wyznaczonych przez poziomy czynników,
3. jednorodność wariancji podpopulacji wyznaczonych przez poziomy czynników.

Weryfikację założenia o normalności rozkładów przeprowadzimy za pomocą odpowiedniego testu zgodności. W szczególności może to być test Shapiro-Wilka, wyznaczany przy użyciu funkcji `shapiro.test()`.

Weryfikacja założenia o jednorodności wariancji sprowadza się do przetestowania hipotezy o równości wariancji podpróbek wyznaczonych przez poziomy czynników. Mamy tu do wyboru kilka testów, z których najpopularniejszy jest test Bartletta, por. `bartlett.test()`.

8.1.3. Nieparametryczna ANOVA

W R mamy do dyspozycji funkcje pozwalające przeprowadzić wnioskowanie w sytuacji, gdy założenia analizy wariancji nie są spełnione. Nieparametrycznym odpowiednikiem jednoczynnikowej ANOVY jest test Kruskala-Wallisa, który w środowisku R dostępny jest przy użyciu wywołania funkcji `kruskal.test()`.

8.2. Zadania rozwiązane

Zadanie 8.1. Wykonano po cztery niezależne pomiary wytrzymałości na ściskanie trzech rodzajów betonu. Otrzymano następujące wyniki (w kG/cm^2):

I	204	200	198	204
II	197	205	213	209
III	190	208	202	210

Stwierdź, czy badane gatunki betonu różnią się istotnie pod względem średniej wytrzymałości na ściskanie. Przyjmij poziom istotności 0.05.

Rozwiązanie. Testujemy hipotezę zerową $H : \mu_1 = \mu_2 = \mu_3$, mówiącą, iż – średnio rzecz biorąc – badane gatunki betonu nie różnią się istotnie pod względem wytrzymałości na ściskanie, wobec hipotezy alternatywnej $K : \neg H$ orzekającej, iż te gatunki różnią się wytrzymałością.

Postawiony problem decyzyjny sugeruje zastosowanie analizy wariancji. Zanim jednak przystąpimy do ANOVY, sprawdźmy, czy są spełnione założenia wspomniane w par. 8.1.2, tzn. niezależność obserwacji, normalność rozkładów i jednorodność wariancji.

Odnośnie pierwszego wymagania przyjmijmy, że w trakcie przeprowadzania eksperymentu zadbano o to, by obserwacje były niezależne. Weryfikację założenia o normalności rozkładów przeprowadzimy za pomocą testu Shapiro-Wilka:

```
> wyt <- c(204, 200, 198, 204, 197, 205, 213, 209, 190, 208, 202, 210)
> gat <- gl(3, 4, labels=1:3) # wektor identyfikujący gatunki betonu
> beton <- data.frame(wyt, gat) # ramka danych naszego zadania
> simplify2array(tapply(beton$wyt, beton$gat,
+    function(x) shapiro.test(x)[1:2]))
          1         2         3
statistic 0.8494024 0.9713737 0.8945062
p.value   0.2242305 0.8499708 0.4042863
```

Otrzymane p-wartości sugerują, że nie ma podstaw do odrzucenia hipotezy o normalności rozkładów żadnej z trzech podpopulacji.

Weryfikacja założenia o jednorodności wariancji sprowadza się do przetestowania hipotezy $H : \sigma_1^2 = \sigma_2^2 = \sigma_3^2$ względem $K : \neg H$. W tym celu skorzystamy z testu Bartletta:

```
> bartlett.test(beton$wyt,beton$gat)

Bartlett test of homogeneity of variances

data:  beton$wyt and beton$gat
Bartlett's K-squared = 2.6706, df = 2, p-value = 0.2631
```

Duża p-wartość (p-value $= 0.2631$) świadczy o tym, że nie ma podstaw do odrzucenia hipotezy o jednorodności wariancji. Po pozytywnym zweryfikowaniu założeń testu F możemy przystąpić do weryfikacji hipotezy $H : \mu_1 = \mu_2 = \mu_3$, względem hipotezy alternatywnej $K : \neg H$:

```
> anova(lm(beton$wyt~beton$gat))
Analysis of Variance Table

Response: beton$wyt
          Df Sum Sq Mean Sq F value Pr(>F)
beton$gat  2  44.67  22.333  0.4902 0.6279
Residuals  9 410.00  45.556
```

Otrzymana p-wartość testu F wskazuje na brak podstaw do odrzucenia hipotezy zerowej. Oznacza to, że badane trzy gatunki betonu nie różnią się istotnie pod względem średniej wytrzymałości na ściskanie.

Na marginesie, aby przeprowadzić analizę wariancji możemy także wywołać:

```
> summary(aov(beton$wyt~beton$gat))
            Df Sum Sq Mean Sq F value Pr(>F)
beton$gat    2   44.7   22.33    0.49  0.628
Residuals    9  410.0   45.56
```

⊡

Zadanie 8.2. Zbadano czas reakcji trzech rodzajów układów stosowanych w kalkulatorach elektronicznych i otrzymano następujące wyniki (w mikrosekundach):

I	19	22	20	18	25	21	24	17
II	20	21	33	27	29	30	22	23
III	16	15	18	26	17	23	20	19

Sprawdź, czy istnieje statystycznie istotna różnica między przeciętnymi czasami reakcji badanych trzech układów. Przyjmij poziom istotności 0.05.

Rozwiązanie. Testujemy hipotezę zerową o braku istotnych różnic między średnimi czasami reakcji badanych układów elektronicznych, tzn. $H : \mu_1 = \mu_2 = \mu_3$, przeciw hipotezie $K : \neg H$ wskazującej na istnienie różnic między czasami reakcji tych układów.

Podobnie jak w poprzednim przykładzie, sprawdzamy wpierw, czy są spełnione założenia analizy wariancji: niezależność obserwacji, normalność rozkładów i jednorodność wariancji. Ten fragment zadania pozostawiamy Czytelnikowi. My zaś – zakładając, iż owe założenia są spełnione – przejdziemy już bezpośrednio do testu F.

```
> czas <- c(19, 22, 20, 18, 25, 21, 24, 17, 20, 21, 33, 27,
+           29, 30, 22, 23, 16, 15, 18, 26, 17, 23, 20, 19)
> typ <- gl(3, 8, labels=c(1, 2, 3))
> uklad <- data.frame(czas, typ)
> anova(lm(uklad$czas~uklad$typ))
Analysis of Variance Table

Response: uklad$czas
          Df Sum Sq Mean Sq F value   Pr(>F)
uklad$typ  2 177.75  88.875  6.0036 0.008668 **
Residuals 21 310.88  14.804
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mała p-wartość testu F wskazuje na konieczność odrzucenia hipotezy zerowej, tzn. na istnienie istotnych różnic między czasami reakcji układów.

Ponieważ mamy tu do czynienia z trzema rodzajami układów, warto sprawdzić, czy każdy z układów różni się od pozostałych, czy też może dwa spośród nich tworzą tzw. grupę jednorodną. Ponadto ciekawe byłoby stwierdzenie, który z badanych układów byłby najlepszy z punktu widzenia konstrukcji kalkulatora, tzn. którego czas reakcji jest najkrótszy.

W celu uzyskania odpowiedzi na pierwsze z pytań zastosujemy tzw. porównania wielokrotne metodą Tukeya:

```
> TukeyHSD(aov(uklad$czas~uklad$typ))
  Tukey multiple comparisons of means
    95% family-wise confidence level
```

```
Fit: aov(formula = uklad$czas ~ uklad$typ)

$`uklad$typ`
      diff          lwr       upr     p adj
2-1  4.875   0.02600137  9.723999 0.0486299
3-1 -1.500  -6.34899863  3.348999 0.7192770
3-2 -6.375 -11.22399863 -1.526001 0.0088750
```

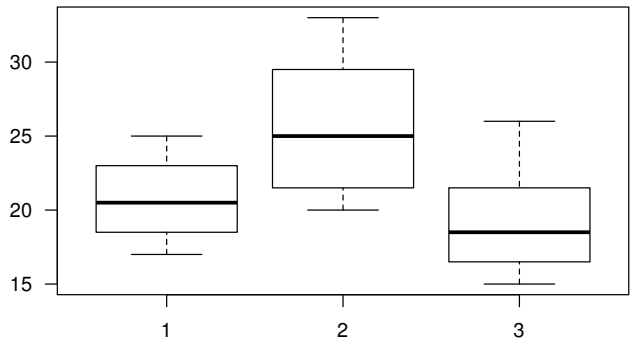
Zamieszczone w ostatniej kolumnie p-wartości testów porównujących parami poziomy badanego czynnika sugerują, iż nie ma istotnych różnic między I i III rodzajem układu, jednakże układy te różnią się istotnie od układu II rodzaju.

Informacja

Metoda Tukeya wymaga, by plan badania był zrównoważony, tzn. aby we wszystkich „klatkach" macierzy eksperymentu było tyle samo obserwacji. Jeśli warunek ten nie jest spełniony możemy posłużyć się np. metodą LSD (ang. *Least Significant Difference*) Fishera.

Z kolei odpowiedź na drugie z postawionych wcześniej pytań, możemy uzyskać, analizując wykres skrzynkowy:

```
> boxplot(split(uklad$czas,uklad$typ), las=1)
```



Z wykresu wynika, że niewątpliwie układy II rodzaju są najgorsze z naszego punktu widzenia, bo przeciętnie charakteryzują się najdłuższym czasem reakcji. Ponadto przeciętnie najkrótszy czas reakcji wykazywały układu III rodzaju, te jednak nie są istotnie lepsze od układów I rodzaju.

□

Zadanie 8.3. W pliku `zarobki.csv` zamieszczono historyczne dane dotyczące wysokości miesięcznych zarobków wybranych losowo osób w czterech miastach: w Warszawie, Krakowie, Wrocławiu i Katowicach. Zbadaj, czy wysokość miesięcznych zarobków w tych miastach różni się istotnie (przyjmij poziom istotności 0.05).

Rozwiązanie. Po załadowaniu pliku z bazy danych

```
> salary <-
+    read.csv2("http://www.ibspan.waw.pl/~pgrzeg/stat_lab/zarobki.csv")
```

możemy sprawdzić zawartość pliku:

```
> summary(salary)
    zarobki          miasto
 Min.   :1070   Katowice:11
 1st Qu.:1394   Krakow  :10
 Median :1752   Warszawa:13
 Mean   :2214   Wroclaw :10
 3rd Qu.:2525
 Max.   :7900
```

Następnie przystępujemy do weryfikacji założeń analizy wariancji:

```
> simplify2array(tapply(salary$zarobki, salary$miasto,
+    function(podprobka)
+       shapiro.test(podprobka)[c("p.value", "statistic")]))
          Katowice   Krakow     Warszawa   Wroclaw
p.value   0.00145374 0.09815871 0.04345169 0.03952855
statistic 0.7380793  0.8693207  0.8639536  0.8360177
> bartlett.test(salary$zarobki,salary$miasto)

Bartlett test of homogeneity of variances

data:  salary$zarobki and salary$miasto
Bartlett's K-squared = 15.9819, df = 3, p-value = 0.001144
```

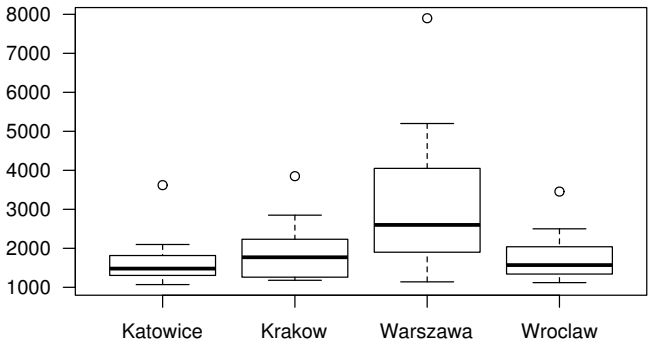
Czytelnik bez trudu stwierdzi, że założenia ANOVY nie są spełnione (brak jednorodności wariancji oraz „kłopoty z normalnością" rozkładów niektórych podpróbek). Tak więc zamiast testu F posłużymy się jego nieparametrycznym odpowiednikiem, tj. testem Kruskala-Wallisa. Zajmiemy się zatem weryfikacją hipotezy $H : F_1 = F_2 = F_3 = F_4$, gdzie F_i oznacza rozkład zarobków w i-tym mieście, przeciw $K : \neg H$:

```
> kruskal.test(salary$zarobki~salary$miasto)

Kruskal-Wallis rank sum test

data:  salary$zarobki by salary$miasto
Kruskal-Wallis chi-squared = 8.0529, df = 3, p-value = 0.04493
```

Okazuje się, że rozkłady wysokości miesięcznych zarobków w rozważanych czterech

miastach różnią się istotnie (na poziomie 0.05). Aby zobaczyć, gdzie są one przeciętnie wyższe, a gdzie niższe, narysujemy wykres skrzynkowy:

```
> boxplot(split(salary$zarobki, salary$miasto), las=1)
```



Zadanie 8.4. W pewnych zakładach lotniczych stosuje się dwie metody nakładania farby podkładowej na części aluminiowe: malowanie zanurzeniowe i natryskowe. Czyni się tak w celu zwiększenia przylegania właściwej farby nawierzchniowej, którą później są malowane owe części. We wspomnianych zakładach stosowano do tej pory trzy rodzaje farb podkładowych. Inżynier technolog, odpowiedzialny za ten etap produkcji, postanowił zbadać, czy rodzaj farby podkładowej oraz sposób jej nakładania na detal mają istotny wpływ na siłę przylegania właściwej farby nawierzchniowej. W tym celu przeprowadzono eksperyment, w którym zmierzono siłę przylegania farby nawierzchniowej do kilku detali malowanych najpierw różnymi farbami podkładowymi, nanoszonymi obiema metodami. Wyniki pomiarów zamieszczono w poniższej tabeli.

Rodzaj farby	Malowanie zanurzeniowe	Malowanie natryskowe
A	4.0; 4.5; 4.3	5.4; 4.9; 5.6
B	5.6; 4.9; 5.4	5.8; 6.1; 6.3
C	3.8; 3.7; 3.9	6.5; 6.0; 5.0

Jakie wnioski powinien wyciągnąć inżynier na podstawie powyższych wyników?

Rozwiązanie. Tym razem mamy przeprowadzimy dwuczynnikową (dwukierunkową) analizą wariancji: pierwszym czynnikiem jest rodzaj farby, natomiast drugim – sposób malowania. Pierwszy czynnik występuje na trzech, zaś drugi na dwóch poziomach.

Zacznijmy od przypomnienia modelu dwuczynnikowej analizy wariancji, w której k-tą obserwację występującą na i-tym poziomie pierwszego czynnka oraz na j-tym po-

ziomie drugiego czynnika, wyobrażamy sobie jako

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk},$$

gdzie μ jest tzw. średnią ogólną, α_i oznacza swoisty wpływ i-tego poziomu pierwszego czynnika, β_j — swoisty wpływ j-tego poziomu drugiego czynnika, γ_{ij} — interakcję i-tego poziomu pierwszego czynnika oraz j-tego poziomu drugiego czynnika, natomiast ε_{ijk} jest błędem losowym obserwacji Y_{ijk}, przy czym $i = 1, \ldots, r$, $j = 1, \ldots, s$ oraz $k = 1, \ldots, n$. W naszym przypadku $r = 3$, $s = 2$ oraz $n = 3$.

Stojący przed nami problem dwuczynnikowej analizy wariancji sprowadza się do weryfikacji następujących hipotez:

1. $H_1 : \alpha_1 = \alpha_2 = \alpha_3 = 0$ przeciw $K_1 : \neg H_1$,
2. $H_2 : \beta_1 = \beta_2 = 0$ przeciw $K_2 : \neg H_2$,
3. $H_3 : \gamma_{11} = \gamma_{12} = \ldots = \alpha_{32} = 0$ przeciw $K_3 : \neg H_3$.

Po pozytywnym zweryfikowaniu założeń analizy wariancji (ten etap pozostawiamy Czytelnikowi, por. zadanie 8.1) przechodzimy bezpośrednio do weryfikacji postawionych powyżej hipotez ANOVY.

```
> X <- c(4.0, 4.5, 4.3, 5.4, 4.9, 5.6, 5.6, 4.9, 5.4,
+      5.8, 6.1, 6.3, 3.8, 3.7, 3.9, 6.5, 6.0, 5.0)
> farba <-gl(3, 6, 18, labels=c("A", "B", "C"))
> m <- gl(2, 3, 18, labels=c("mz", "mn"))
> dane <- data.frame(X, farba, m)
> anova(lm(dane$X~dane$farba*dane$m))
Analysis of Variance Table

Response: dane$X
                 Df Sum Sq Mean Sq F value    Pr(>F)
dane$farba        2 3.1244  1.5622  9.5646  0.003282 **
dane$m            1 7.3472  7.3472 44.9830 2.174e-05 ***
dane$farba:dane$m  2 1.3378  0.6689  4.0952  0.044077 *
Residuals        12 1.9600  0.1633
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Z tablicy analizy wariancji wnioskujemy, że na poziomie istotności 0.05 należy odrzucić wszystkie trzy hipotezy zerowe, co oznacza, że zarówno rodzaj używanej farby, jak i sposób malowania, maja wpływ na siłę przylegania farby nawierzchniowej. Co więcej, występują interakcje między rodzajem farby i sposobem jej nakładania. Tym samym nie zakończyliśmy jeszcze rozwiązywania naszego zadania, albowiem odrzucenie hipotez zerowych skłania nas m.in. do przeprowadzenia porównań wielokrotnych, których celem będzie bliższe przyjrzenie się wpływowi poszczególnych poziomów badanych czynników na zmienną objaśniającą.

Do przeprowadzenia porównań wielokrotnych posłużymy się metodą Tukeya:

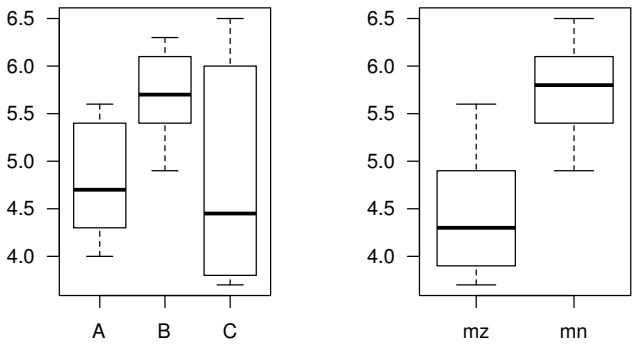
```
> TukeyHSD(aov(dane$X~dane$farba*dane$m), which=c("dane$farba","dane$m"))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = dane$X ~ dane$farba * dane$m)

$`dane$farba`
           diff        lwr        upr     p adj
B-A  0.90000000  0.2774985  1.5225015 0.0059684
C-A  0.03333333 -0.5891682  0.6558349 0.9888224
C-B -0.86666667 -1.4891682 -0.2441651 0.0076976

$`dane$m`
          diff       lwr      upr    p adj
mn-mz 1.277778 0.8626794 1.692876 2.17e-05
```

Z otrzymanej tabeli wynika, że farby typu A i C stanowią grupę jednorodną, tzn. różnice między nimi są statystycznie nieistotne. Z kolei farba typu B istotnie różni się zarówno od farby A jak i farby C.

Tabela potwierdza również istotną różnice między sposobami nakładania farby, ale o tym już wiedzieliśmy z tablicy ANOVY (z uwagi na to, że drugi czynnik występuje tylko na dwóch poziomach, porównania wielokrotne w tym przypadku nie wnoszą niczego nowego).

Relacje między poziomami poszczególnych czynników i ich wpływ na zmienną objaśnianą można zilustrować wykresami skrzynkowymi:

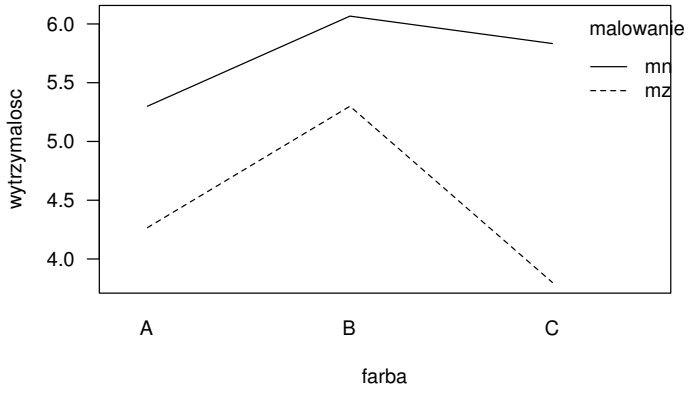
```
> par(mfrow=c(1,2))
> boxplot(split(dane$X,dane$farba), las=1)
> boxplot(split(dane$X,dane$m), las=1)
```



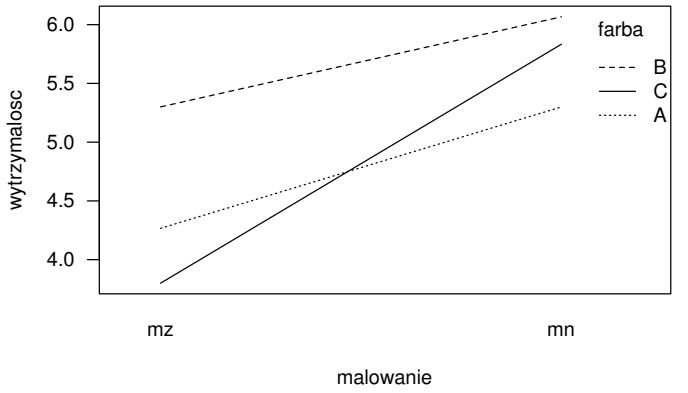
Do oceny interakcji między czynnikami posłużymy się wykresami, w których na osi

odciętych mamy poziomy wybranego czynnika, natomiast na osi rzędnych wartości badanej cechy. Z uwagi na to, że mamy do czynienia z analizą dwuczynnikową, uzyskamy w ten sposób dwa wykresy.

```
> interaction.plot(dane$farba, dane$m, dane$X, xlab="farba",
+     ylab="wytrzymalosc", trace.label="malowanie", las=1)
```




```
> interaction.plot(dane$m, dane$farba, dane$X, xlab="malowanie",
+     ylab="wytrzymalosc", trace.label="farba", las=1)
```



O istnieniu interakcji świadczy brak „równoległości" odcinków (łamanej) tworzących powyższe wykresy.

Reasumując stwierdzamy, że najlepsze wyniki, jeśli chodzi o przyleganie farby nawierzchniowej, daje malowanie natryskowe farbą typu B. ⊡

8.3. Zadania do samodzielnego rozwiązania

Zadanie 8.5. Do gimnazjum osiedlowego trafiają uczniowie z trzech okolicznych szkół podstawowych. Wylosowano niezależnie po czterech uczniów wywodzących się z każdej z tych szkół i okazało się, że mieli oni następujące średnie z ocen na świadectwie ukończenia szóstej klasy:

Szkoła A	Szkoła B	Szkoła C
4.2	4.4	3.8
4.4	4.0	3.6
4.3	4.7	4.0
4.5	4.3	3.7

Zbadaj, czy istnieją statystycznie istotne różnice między przeciętnymi wynikami absolwentów tych trzech szkół podstawowych.

Zadanie 8.6. Przeprowadzono następujące doświadczenie: 18 mężczyzn i 18 kobiet rozmieszczono losowo w 9 pokojach w ten sposób, że w każdym pokoju były po dwie osoby tej samej płci. W pokojach tych utrzymywano stałą temperaturę: 18, 21 albo 24 stopnie Celsjusza (przydział temperatur poszczególnym pokojom był także losowy). Po upływie trzech godzin oceniano samopoczucie każdej z badanych osób (zastosowano ocenę punktową, w której 1 = zbyt zimno, 8 = idealna temperatura, 15 = zbyt ciepło).

	M	K
18	5, 4, 5, 4, 4, 2	1, 2, 5, 5, 1, 3
21	8, 8, 6, 3, 5, 7	10, 7, 8, 8, 7, 8
24	12, 8, 8, 7, 6, 6	11, 13, 8, 8, 6, 7

Zbadaj wpływ, jaki na samopoczucie osób wywiera temperatura panująca w danym pokoju. Czy ocena samopoczucia zależy od płci? Czy występują tu istotne interakcje między badanymi czynnikami (tzn. temperaturą i płcią)?

Zadanie 8.7. W celu zbadania wpływu czterech dawek nawożenia azotowego (w ilościach 0, 40, 80 i 120 kg/ha) na wysokość plonów lucerny przy trzech sposobach siewu (siew czysty C oraz dwa rodzaje wsiewu M i P w jęczmień jary) założono doświadczenie w czterech powtórzeniach. Dla każdej kombinacji nawożenia ze sposobem siewu zmierzono plon zielonej masy (w kg z poletka). W pierwszym pokosie uzyskano następujące obserwacje:

	0	40	80	120
C	33.2; 36.2; 44.2; 51.0	42.2; 41.4; 50.6; 45.2	50.2; 53.0; 52.6; 45.0	46.2; 52.4; 49.0; 43.6
M	18.6; 13.0; 14.6; 18.8	18.0; 20.0; 14.2; 19.1	24.2; 21.6; 16.4; 19.0	34.2; 17.2; 15.5; 22.2
P	20.4; 14.4; 11.0; 22.6	21.9; 42.0; 16.2; 25.6	18.2; 21.0; 27.3; 27.6	16.4; 15.0; 21.6; 27.8

Ustal, który z badanych czynników miał istotny wpływ na wysokość plonów masy zielonej.

Zadanie 8.8. W celu porównania trzech środków antykorozyjnych pobrano po 10 próbek losowych drutu zabezpieczanego każdym z tych środków i zmierzono głębokość zaistniałej korozji (razy 10^{-3} mm). Wyniki pomiarów przestawia poniższa tabelka:

Środek A	Środek B	Środek C
98.5	100.2	56.7
98.5	99.2	82.0
98.5	99.9	67.8
97.5	97.8	58.3
99.3	99.8	61.2
102.0	100.5	67.8
101.8	99.8	117.4
98.3	99.0	103.4
102.0	101.1	43.8
101.2	100.8	86.1

Czy na podstawie tych danym można stwierdzić, że środki te różnią się istotnie pod względem jakości tworzonego przez nie zabezpieczenia antykorozyjnego?

8.4. Wskazówki i odpowiedzi do zadań

Ad zad. 8.5. Założenia jednoczynnikowej ANOVY są spełnione.
Test F: wartość statystyki testowej $F = 10.239$, p-wartość $= 0.004802$. Zatem średnie wyniki absolwentów tych trzech szkół różnią się istotnie.
Test HSD Tukeya: istotna różnica między średnimi wynikami uczniów szkół A i C oraz B i C.

Ad zad. 8.6. Założenia dwuczynnikowej ANOVY są spełnione.
Test F istotności czynnika temperatura: wartość statystyki testowej $F = 21.902$, p-wartość $= 1.37e-06$.
Test F istotności czynnika płeć: wartość statystyki testowej $F = 0.776$, p-wartość $= 0.385$.

Test F istotności interakcji czynników temperatura i płeć: wartość statystyki testowej $F = 2.011$, p-wartość $= 0.152$.
Zatem istotny wpływ na samopoczucie ma tylko czynnik temperatura.
Test HSD Tukeya: istotna różnica między średnim samopoczuciem tylko przy temperaturach 18 i 21 oraz 18 i 24.

Ad zad. 8.7. Założenia dwuczynnikowej ANOVY są spełnione.
Test F istotności czynnika a (rodzaj wysiewu): wartość statystyki testowej $F = 101.369$, p-wartość $= 1.62e - 15$.
Test F istotności czynnika b (sposób nawożenia): wartość statystyki testowej $F = 2.842$, p-wartość $= 0.0514$.
Test F istotności interakcji czynników a i b: wartość statystyki testowej $F = 0.819$, p-wartość $= 0.5625$.
Zatem istotny wpływ na średnią wysokość plonów ma rodzaj wysiewu oraz nawożenia ale brak jest istotności interakcji tych czynników.
Test HSD Tukeya dla czynnika a: istotna różnica między średnią wysokością plonów tylko dla rodzaju wysiewu C i M oraz C i P.
Test HSD Tukeya dla czynnika b: istotna różnica między średnią wysokością plonów tylko dla nawożenia w dawce 0 i 80.

Ad zad. 8.8. Założenia jednoczynnikowej ANOVY nie są spełnione.
Test Kruskala-Wallisa: wartość statystyki testowej $T = 7.0019$, p-wartość $= 0.03017$.
Zatem skuteczność badanych środków antykorozyjnych jest istotnie różna.

Bibliografia

[1] P. Biecek. *Przewodnik po pakiecie* R. GiS, Wrocław, 2011.

[2] M.J. Crawley. *Statistics: An Introduction Using* R. John Wiley & Sons, 2005.

[3] M.J. Crawley. *The* R *Book*. John Wiley & Sons, 2007.

[4] P. Dalgaard. *Introductory Statistics with* R. Springer-Verlag, 2008.

[5] B.S. Everitt, T. Hothorn. *A Handbook of Statistical Analyses Using* R. Chapman & Hall, 2006.

[6] M. Gągolewski. *Programowanie w języku* R. *Analiza danych, obliczenia, symulacje*. Wydawnictwo Naukowe PWN, Warszawa, 2014.

[7] P. Grzegorzewski, K. Bobecka, A. Dembińska, J. Pusz. *Rachunek prawdopodobieństwa i statystyka*. WSISiZ, Warszawa, 2008.

[8] G.J. Kerns. *Introduction to Probability and Statistics Using* R. 2011.

[9] J. Koronacki, J. Mielniczuk. *Statystyka*. WNT, Warszawa, 2001.

[10] R. Magiera. *Modele i metody statystyki matematycznej. Część I. Rozkłady i symulacja stochastyczna*. GiS, Wrocław, 2007.

[11] R. Magiera. *Modele i metody statystyki matematycznej. Część II. Wnioskowanie statystyczne*. GiS, Wrocław, 2007.

[12] N.S. Matloff. *The Art of* R *Programming: A Tour of Statistical Software Design*. No Starch Press, 2011.

[13] H.V. Mittal. R *Graphs Cookbook*. Packt Publishing, 2011.

[14] P. Murrell. R *Graphics*. Chapman & Hall/CRC, 2006.

[15] R. Wieczorkowski, R. Zieliński. *Komputerowe generatory liczb losowych*. WNT, Warszawa, 1997.

[16] R. Zieliński. *Siedem wykładów wprowadzających do statystyki matematycznej*. PWN, Warszawa, 1990.