Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Tomasz Kulczyński, Błażej Osiński
Nr albumu: 262964, 262963

Wybrane metody statystycznego repróbkowania

Praca licencjacka
na kierunku MATEMATYKA

Praca wykonana pod kierunkiem
dra inż. Przemysława Biecka
Instytut Matematyki Stosowanej i Mechaniki

Październik 2011

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

W pracy przedstawiono testy permutacyjne jako interesujące podejście do testowania hipotez statystycznych, pozwalające abstrahować od założeń o rozkładach, z których pochodzą obserwacje. Opisano także oparte o repróbkowanie metody szacowania precyzji estymatorów z prób, bez jakiejkolwiek wiedzy o rozkładzie poza daną próbą. Następnie pokazano, jak obliczenia potrzebne do użycia tych metod wykonać przy pomocy pakietu R, a także zastosowano je do rzeczywistych danych medycznych, dotyczących zachorowalności na choroby alergiczne.

Słowa kluczowe

testy permutacyjne, bootstrap, jackknife, ECAP

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.2 Statystyka

Klasyfikacja tematyczna

62. Statistics
62.G. Nonparametric inference
62.G.09. Resampling methods

Tytuł pracy w języku angielskim

Selected statistical resampling methods

Spis treści

Wprowadzenie

Statystyka odgrywa coraz większą rolę w wielu dziedzinach ludzkiej aktywności — znajduje zastosowanie w finansach, przemyśle, polityce, a także w praktycznie dowolnej gałęzi nauki. To rosnące zainteresowanie wynika z mnogości informacji, które docierają do nas w ilości niespotykanej wcześniej w historii. Jak ujął to amerykański bibliotekarz Rutherford D. Rogers: *Toniemy w informacjach, desperacko pragnąc wiedzy.* Odpowiedzialną rolę współczesnych Heraklesów, których zadaniem jest uporządkować informacyjną stajnię Augiasza, przyjęli na siebie statystycy. Rosnące wymagania stawiane statystyce powodują prężny rozwój tej dziedziny. Wśród nowych metod, które pojawiły się w ostatnim pięćdziesięcioleciu są te oparte o repróbkowanie (ang. *resampling*), których dotyczy niniejsza praca.

Prace podzieliliśmy na trzy części. W pierwszej (rozdziały 1. – 3.) opisujemy ważniejsze metody repróbkowania: testy permutacyjne, bootstrap i jackknife. Dość szeroko omówiliśmy test najbliższych sąsiadów; w przykładowych eksperymentach numerycznych zajęliśmy się kwestią dobierania optymalnej statystki testowej. W związku z tym, że metody repróbkowania wymagają dużej liczby stosunkowo prostych obliczeń, naturalne wydaje się wykorzystanie w tym celu komputera. Dlatego też druga część pracy (rozdział 4.) dotyczy używania wymienionych wyżej metod w pakiecie statystycznym R. Lektura przedstawionych przez nas implementacji, wraz z dokładnym opisem, pozwoli w przyszłości stosować je gdy zajdzie taka potrzeba. W końcu w trzeciej części (rozdział 5.), podjęliśmy próbę wykorzystania opisanych metod do analizy rzeczywistych danych.

Wykorzystane przez nas dane medyczne pochodzą z projektu Epidemiologia Chorób Alergicznych w Polsce (ECAP), który jest prowadzony przez Zakład Profilaktyki Zagrożeń Środowiskowych i Alergologii Warszawskiego Uniwersytetu Medycznego z inicjatywy Ministra Zdrowia. Projekt ma na celu znalezienie sposobów profilaktyki i wczesnego leczenia chorób alergicznych w Polsce. Zebrane w badaniu dane zawierają informacje o cechach osobowych badanych, czynnikach ryzyka na jakie są wystawieni oraz o występowaniu u nich alergii i astmy.

Rozdział 1

Testy permutacyjne dla obserwacji jednowymiarowych

Do sprawdzania, czy rozkłady, z których pochodzą próby, są takie same, stosuje się rozmaite testy statystyczne. Zawsze polegają one na obliczeniu pewnej funkcji, zwanej *statystyką testową*, której argumentem są zadane próby. Wynik tej funkcji z pomocą tablic (a coraz częściej komputera) przekształca się na p-wartość, która porównana z założonym poziomem istotności pozwala nam stwierdzić istnienie lub brak przesłanek do podważenia prawdziwości hipotezy o równości rozkładów.

Niestety bardzo często (np. w przypadku testu t-Studenta) ich teoretyczne podstawy wymagają dodatkowych założeń o porównywanych rozkładach (np. o normalności rozkładów). Bez tych założeń nie wiedzielibyśmy jaki jest rozkład statystki testowej, a więc i p-wartości uzyskane byłyby w sposób heurystyczny, bez dowodu, że są one poprawne. Niestety w praktyce nie zawsze można formalnie sprawdzić czy zachodzą wymagane założenia o rozkładach.

Okazuje się jednak, że zamiast wielu testów możemy zastosować odpowiadające im testy permutacyjne, które nie wymagają dodatkowych założeń o rozkładach porównywanych prób. Testy te opierają się na spostrzeżeniu, że jeżeli hipoteza zerowa o równości rozkładów jest prawdziwa, to zamiast naszych dwóch prób mogliśmy, z takim samym prawdopodobieństwem, zobaczyć obserwacje o tych samych wartościach, ale innym podziałem na próby. Dla przykładu, gdy naszą obserwacją są dwie dwuelementowe próby: $(\{0{,}2, 0{,}37\}, \{0{,}11, 0{,}29\})$, i zakładamy, że pochodzą z takich samych rozkładów, to równie prawdopodobne było zaobserwowanie prób $(\{0{,}2, 0{,}29\}, \{0{,}11, 0{,}37\})$, czy też $(\{0{,}11, 0{,}2\}, \{0{,}29, 0{,}37\})$.

Powyższe spostrzeżenie pozwala znaleźć rozkład statystki testowej, na podstawie wartości uzyskanych przez tę funkcję przy różnych podziałach na próby. Dzięki temu możemy obliczyć p-wartość i dalej postępować tak, jak w przypadku innych testów statystycznych.

1.1. Porównywanie dwóch rozkładów

Opiszemy teraz bardziej formalnie zastosowanie testu permutacyjnego na przykładzie porównywania dwóch rozkładów. Załóżmy, że mamy dwie próby $X = \{X_1, X_2, \ldots, X_n\}$ i $Y = \{Y_1, Y_2, \ldots, Y_m\}$ z jednowymiarowych rozkładów o dystrybuantach F_X i F_Y odpowiednio. Będziemy chcieli zweryfikować hipotezę zerową $H_0 : F_X = F_Y$ przeciw alternatywnej hipotezie $H_1 : F_X \neq F_Y$.

Jedną z często stosowanych metod w takim przypadku jest test Kołmogorowa-Smirnowa.

Opiera się on o statystykę:

$$D_{n,m} = \sup_{-\inf<z<\inf} |\hat{F_{X}}_n - \hat{F_{Y}}_m|, \tag{1.1}$$

gdzie $\hat{F_{X}}_n$ i $\hat{F_{Y}}_m$ to dystrybuanty empiryczne uzyskane z prób X i Y odpowiednio. Wiemy, że przy założeniu hipotezy zerowej rozkład statystki $D_{n,m}$ nie zależy od rozkładu z jakich wylosowano próby, o ile rozkłady te są ciągłe. Rozkład tej statystyki możemy więc znaleźć dla jakiś prostych rozkładów (np. jednostajnych), a potem stosować go do porównywania rozkładów o nieznanych, ciągłych dystrybuantach. Oznacza to też jednak, że jeżeli nie mamy pewności czy dystrybuanty F_X i F_Y są ciągłe, czy zgoła nic o nich nie wiemy, to nie ma ogólnego sposobu na znalezienie potrzebnego nam rozkładu statystyki testowej.

W omawianym przypadku nie mamy dodatkowych założeń o rozkładach prób X i Y, więc zasadne jest użycie testu permutacyjnego opartego o statystykę $D_{n,m}$. „Duże" wartości statystyki sugerują odrzucenie hipotezy zerowej, wskazują na różniące się rozkłady, a „małe" przeciwnie, potwierdzają równość rozkładów. Za chwilę okaże się, że dzięki zastosowaniu testów permutacyjnych możemy ściśle określić, jakie wartości statystyki uznajemy za „duże", a jakie za „małe". Należy w tym celu formalnie ująć spostrzeżenie zawarte we wstępie, o równym prawdopodobieństwie różnych podziałów na próby.

Niech $Z = (Z_1, Z_2, \ldots, Z_{n+m})$ będzie uporządkowanym wektorem obserwacji zarówno z X, jak i z Y, a przez $N = n + m$ oznaczmy jego długość. Stwórzmy też wektor $g = (g_1, g_2, \ldots, g_N) = (\underbrace{0, 0, \ldots, 0}_{n}, \underbrace{1, 1, \ldots, 1}_{m})$, w którym g_i jest równe 0 gdy Z_i pochodzi z próby X i 1 gdy pochodzi z Y. Zauważmy, że para (Z, g) zawiera wszystkie informacje zawarte w dwóch próbach X i Y. Co więcej, dowolna permutacja g^* wektora g wyznacza podział obserwacji $\{X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_m\}$ na dwa zbiory: n-elementowy i m-elementowy.

Podstawą do stosowania testów permutacyjnych jest następujący fakt:

Twierdzenie 1.1.1 Lemat permutacyjny. *Przy założeniu hipotezy zerowej $H_0 : F_X = F_Y$ oraz otrzymania wektora obserwacji Z, prawdopodobieństwo podziału na próby wyznaczonego wektorem g, lub dowolną jego permutacją, jest takie samo.*

Zauważmy, że jeżeli obserwacje w wektorze Z są parami różne to możliwych podziałów jest $\binom{N}{n}$, a więc prawdopodobieństwo uzyskania któregokolwiek z nich wynosi $\frac{1}{\binom{N}{n}}$. Natomiast jeżeli obserwacje się powtarzają, to różne permutacje wektora g mogą wyznaczać ten sam podział obserwacji. Wówczas jednak każdy podział będzie wyznaczany przez tę samą liczbę permutacji wektora g: jeżeli zbiór obserwacji ma k elementów, a kolejne obserwacje występują $l_1, l_2, \ldots, l_k$ razy ($\sum_{i=1}^{k} l_i = N$) to każdy podział jest wyznaczany przez $l_1! l_2! \ldots l_k!$ permutacji. Aby nie komplikować zbędnie dalszego opisu, a także przykładowych implementacji w rozdziale 4., będziemy rozważali wszystkie możliwe permutacje wektora g. Czasami doprowadzi to do wielokrotnego zliczania tego samego podziału, ale każdego z nich tę samą liczbę razy.

Dowolną statystykę, która porównuje ze sobą rozkłady dwóch prób (np. statystyki Kołmogorowa-Smirnowa, t-Studenta) możemy zapisać w postaci funkcji $S(Z, g^*)$ przyjmującej dwa wektory: jeden zawierający obserwacje (Z), a drugi wskazujący podział na próby (g^*). Z lematu permutacyjnego wynika, że przypisując jednakową wagę wartościom statystyki $S(Z, g^*)$ dla wszystkich wektorów g^* o n zerach i m jedynkach możemy obliczyć rozkład statystyki testowej przy założeniu hipotezy H_0.

W szczególności aby policzyć p-wartość testu permutacyjnego, czyli prawdopodobieństwo uzyskania bardziej „skrajnej" wartości statystyki testowej przy założeniu H_0, należy sprawdzić wszystkie możliwe permutacje g^* i zliczyć te z nich, dla których $S(Z, g^*)$ jest bardziej

„skrajny" niż $S(Z,g)$. Wartość skrajna oznacza tu bardziej sprzyjająca odrzuceniu hipotezy zerowej, na przykład przy statystyce Kołmogorowa-Smirnowa będzie chodziło po prostu o wartości większe.

Powyższe rozważania prowadzą do zapisania zwartego wzoru na test permutacyjny. Jeżeli duże wartości statystyki S sprzyjają odrzuceniu H_0, to możemy obliczyć p-wartość testu dla danych (Z,g):

$$p\text{-}value = P(S(Z,g^*) \geqslant S(Z,g)) = \frac{1}{\binom{N}{n}} \sum_{i=1}^{\binom{N}{n}} I(S(Z,g_i^*) \geqslant S(Z,g)), \tag{1.2}$$

gdzie $g_1, g_2, \ldots, g_{\binom{N}{n}}$ są wszystkimi permutacjami wektora g.

Podobnie w przypadku gdy małe wartości S sprzyjają odrzuceniu H_0 to p-wartość wyniesie:

$$p\text{-}value = \frac{1}{\binom{N}{n}} \sum_{i=1}^{\binom{N}{n}} I(S(Z,g_i^*) \leqslant S(Z,g)). \tag{1.3}$$

Natomiast dla statystyk, dla których zarówno małe, jak i duże wartości sugerują odrzucenie H_0 (taką statystyką jest na przykład różnica średniej prób) p-wartość definiujemy jako:

$$p\text{-}value = \begin{cases} 2\hat{p} & \hat{p} \leqslant 0.5 \\ 2(1-\hat{p}) & \hat{p} > 0.5 \end{cases}, \tag{1.4}$$

gdzie

$$\hat{p} = \frac{1}{\binom{N}{n}} \sum_{i=1}^{\binom{N}{n}} I(S(Z,g_i^*) \geqslant S(Z,g)). \tag{1.5}$$

1.2. Metoda Monte Carlo

Liczba wszystkich możliwych podzbiorów n-elementowych zbioru N-elementowego, które należałoby sprawdzić w teście permutacyjnym — $\binom{N}{n}$ — rośnie bardzo szybko, wykładniczo[1] wraz ze wzrastającymi N i n. Na przykład już przy dwóch próbach wielkości 12 mielibyśmy do sprawdzenia $\binom{24}{12} = 2\,704\,156$ możliwych podzbiorów. Oznacza to, że dla większych danych w praktyce nie można sobie pozwolić na przeprowadzenie pełnej procedury testu permutacyjnego.

Aby móc sobie poradzić z przeprowadzaniem testów permutacyjnych także dla licznych prób należy zastosować opartą o losowanie metodę Monte Carlo. Polega ona na obliczeniu przybliżonej wartości wyniku testu poprzez ograniczenie się do pewnego losowo wybranego podzbioru podziałów.

[1] O ile tylko n jest rzędu $\frac{N}{2}$, a nie jest bardzo małe, ani bardzo duże.

Mając zatem daną obserwację (Z, g), ustalamy liczbę powtórzeń R, a następnie R-krotnie wybieramy losową permutację g^* wektora g. Dla każdej z nich liczymy wartość statystyki $S(Z, g^*)$ i zliczamy jak często wartość ta była bardziej skrajna od $S(Z, g)$. Gdy oznaczmy kolejne wybierane wektory przez $g_1, \ldots, g_R$, oraz $g_0 = g$, wówczas za oszacowanie p-wartości przyjmiemy:

$$p\text{-}value = \frac{1}{R+1} \sum_{i=0}^{R} I(S(Z, g_i) \geqslant S(Z, g)) = \frac{1}{R+1} \left(1 + \sum_{i=1}^{R} I(S(Z, g_i) \geqslant S(Z, g)) \right), \quad (1.6)$$

w przypadku statystyki S, której duże wartości sugerują odrzucenie hipotezy zerowej (wzory w pozostałych przypadkach będą analogiczne do tych z poprzedniego rozdziału).

Zauważmy jeszcze, że w powyższym wzorze wliczamy do oszacowania p-wartości wkład od zaobserwowanego wektora g. Jest to rozwiązanie sugerowane w książce [Rizzo], jako dające lepsze oszacowanie, które przy okazji zapewnia nam, że uzyskana p-wartość będzie niezerowa.

Rozdział 2

Testowanie równości rozkładów wielowymiarowych

Jednym z zastosowań testów permutacyjnych jest badanie równości rozkładów wielowymiarowych. W ogólnym przypadku problem ten nie daje się rozwiązać metodami znanymi z przypadku jednowymiarowego. Popularne testy, jak np. test Kołmogorowa-Smirnowa, nie mają bezpośrednich odpowiedników wielowymiarowych. Dlatego dobrym pomysłem jest użycie opisanego niżej testu, który ponadto nie wymaga dodatkowych założeń o rozkładach prób.

W całym rozdziale zakładamy, że porównujemy dwa rozkłady, z których mamy próby: $\{X_1, X_2, \ldots, X_{n_1}\}$ oraz $\{Y_1, Y_2, \ldots, Y_{n_2}\}$, które pochodzą z rozkładów d-wymiarowych: $X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2} \in R^d$. Musimy także zdawać sobie sprawę, że opisany test zależy od przyjętej metryki. Będziemy używali funkcji odległości δ bez specyfikowania metryki, w której ta odległość jest mierzona. Naturalnym jest przyjęcie metryki euklidesowej na R^d, ale w zależności od znaczenia zmiennych X i Y możliwe jest przyjęcie dowolnej innej metryki (przy zachowaniu konsekwencji — taka sama metryka dla całego wykonywanego testu).

2.1. Test najbliższych sąsiadów

Test najbliższych sąsiadów w celu zweryfikowania hipotezy zerowej o równości rozkładów, bada wzajemne położenie punktów $X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}$ w przestrzeni R^d. Opiera się na ogólnej idei, że obserwacje zmiennych o jednakowych rozkładach będą leżały blisko siebie, natomiast obserwacje zmiennych o różnych rozkładach — tworzyły mniej lub bardziej oddzielne grupy punktów w R^d. Intuicyjnie, jeśli weźmiemy próby z dwóch rozkładów przesuniętych względem siebie o wektor (odpowiednio długi), obserwacje z pierwszej nich będą sąsiadować w przestrzeni ze sobą, a obserwacje z drugiej ze sobą.

Przejdźmy do formalnego opisu tej intuicji. Tak, jak w przypadku innych testów permutacyjnych, oznaczmy przez $Z = (Z_1, \ldots, Z_{n_1+n_2}) = (X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2})$ wektor wszystkich obserwacji.

Definicja 2.1.1 *Niech $NN_k(Z_i)$ będzie k-tym najbliższym sąsiadem Z_i wśród wszystkich punktów z $Z \setminus \{Z_i\}$. Można to zapisać następująco:*

$$NN_k(Z_i) = Z_j \iff |\{x : x \neq i, j \ \wedge \ \delta(Z_i, Z_x) < \delta(Z_i, Z_j)\}| = k - 1 \tag{2.1}$$

W szczególności, $NN_1(Z_i) = Z_j$ wtedy i tylko wtedy, gdy

$$\forall_x \left(x \neq i, j \ \Rightarrow \ \delta(Z_i, Z_j) < \delta(Z_i, Z_x)\right). \tag{2.2}$$

W takiej definicji nie uwzględniamy możliwości jednakowych odległości pomiędzy różnymi parami punktów. Przy rozważaniu rozkładów ciągłych nie jest to istotne, gdyż prawdopodobieństwo zajścia takiej sytuacji jest równe zero. Jeśli chcemy się przed tym zabezpieczyć (gdy rozważane zmienne mają rozkłady dyskretne), można przyjąć dla $i \neq j$:

$$\delta'(Z_i, Z_j) = \frac{2|Z|}{M}\delta(Z_i, Z_j) + i + j, \tag{2.3}$$

oraz $\delta'(Z_i, Z_i) = 0$, gdzie

$$M = \min_{i,j:\ \delta(Z_i,Z_j)>0} \delta(Z_i, Z_j). \tag{2.4}$$

Taka definicja δ' zapewnia, że z własności metryki δ wynika, że δ' jest również metryką i w dodatku

$$\delta'(Z_i, Z_j) = \delta'(Z_i, Z_k) \ \Rightarrow \ j = k. \tag{2.5}$$

Wprowadźmy dodatkowo oznaczenie: $I_i(k)$, która będzie równa 1 wtedy i tylko wtedy, gdy Z_i i $NN_k(Z_i)$ należały do tej samej próby przed połączeniem (oba do X lub oba do Y), w przeciwnym razie $I_i(k)$ jest równa 0.

Definicja 2.1.2 Statystyką pierwszych najbliższych sąsiadów *nazywamy funkcję*

$$T_1 = \frac{1}{|Z|}\sum_{i=1}^{|Z|} I_i(1). \tag{2.6}$$

Jest to statystyka, którą z powodzeniem możemy zastosować do wykonania permutacyjnego testu równości rozkładów. Przy założeniu hipotezy zerowej (że rozkłady są równe), spodziewamy się małej wartości statystyki w porównaniu do hipotezy alternatywnej (że rozkłady są różne). Dalej postępujemy zgodnie z opisaną wcześniej metodą Monte Carlo: losujemy N podziałów zbioru Z na dwa podzbiory i dla każdego z nich obliczamy wartość tej statystki, a następnie porównujemy z wartością statystyki dla wyjściowego podziału na X i Y. Niech w będzie liczbą wylosowanych podziałów o statystyce nie mniejszej niż wyjściowa. W takim razie za p-wartość przyjmujemy $\frac{w+1}{N+1}$.

Zauważmy, że w takim teście skorzystaliśmy z bardzo małej części informacji o najbliższych sąsiadach: tylko z wartości $NN_i(1)$, pomijając zupełnie kolejność dalszych sąsiadów. Szczególnie przy dużej liczbie obserwacji, ograniczanie się do najbliższego sąsiada wydaje się nierozsądne — przecież drugi z kolei najbliższy sąsiad również jest ważny, a sami najbliżsi sąsiedzi nie mogą przekazać pełnej informacji o zależnościach pomiędzy badanymi rozkładami. Z tego powodu, uogólniamy powyższą definicję:

Definicja 2.1.3 k-tą statystyką najbliższych sąsiadów *nazywamy funkcję*

$$T_k = \frac{1}{k|Z|}\sum_{i=1}^{|Z|}\sum_{j=1}^{k} I_i(j). \tag{2.7}$$

Niewątpliwie, użycie T_k o k większym od 1 powoduje wykorzystanie znacznie większej ilości informacji o położeniu sąsiadów, jednak nie zawsze daje to lepsze wyniki dla całego testu. Więcej na ten temat w kolejnym podrozdziale.

2.2. Dobór optymalnej liczby sąsiadów

Aby optymalnie użyć omawiany test należałoby wiedzieć dla jakiego k, k-ta statystyka najbliższych sąsiadów najlepiej rozstrzyga pomiędzy hipotezą zerową i alternatywną. W ogólności wydaje się to dalece nieoczywiste — optymalna liczba sąsiadów z pewnością będzie zależała od bardzo wielu czynników. Postanowiliśmy sprawdzić za pomocą eksperymentu numerycznego jakie będą najlepsze wartości k w pewnych konkretnych przypadkach.

2.2.1. Plan eksperymentów

W eksperymentach $N = 1000$ razy wykonamy doświadczenie składające się z następujących kroków:

- Wylosujmy dwie próby trzydziestoelementowe z rozkładów o dystrybuantach F_X i F_Y.
- Dla różnych k wykonujemy test permutacyjny k-najbliższych sąsiadów i zapisujemy uzyskane p-wartości.

W połowie doświadczeń dystrybuanty F_X i F_Y będą identyczne, a w pozostałych będą różniły się jakimś parametrem. Przy założeniu, że hipotezą zerową jest równość tych rozkładów, chcielibyśmy, by test odrzucał hipotezę zerową tylko gdy rozkłady rzeczywiście się różnią. Pytaniem, na które będziemy szukali odpowiedzi jest „Dla jakiego k test najbliższych sąsiadów najlepiej rozstrzyga o prawdziwości hipotezy zerowej?".

Do eksperymentów wybraliśmy zmienne dwuwymiarowe, o współrzędnych niezależnie losowanych z rozkładów normalnych. W połowie doświadczeń były to zawsze standardowe rozkłady normalne (o średniej 0 i odchyleniu standardowym 1), natomiast w drugiej połowie doświadczeń w jednej z prób rozkładowi pierwszej współrzędnej dobraliśmy inne parametry.

2.2.2. Porównywanie testów

Pytanie, jak porównać ze sobą testy samo w sobie nie jest oczywiste. Poniżej opiszemy przyjęty przez nas sposób oceny testu, zaczynając od ustalenia pewnych standardowych definicji i oznaczeń.

Dla każdego testu (przy ustalonej wartości k) wynikiem eksperymentu jest ciąg N p-wartości uzyskanych w kolejnych doświadczeniach. O każdym z nich wiemy, czy spełniona jest w nim hipotezy zerowa, czy alternatywna. Wszystkim testom ustaliliśmy jeden wspólny poziom istotności $\alpha = 0{,}05$. Jeżeli uzyskana w teście p-wartość jest mniejsza bądź równa od α to wynik testu jest *dodatni*, czyli sugeruje odrzucenie hipotezy zerowej. W przeciwnym przypadku wynik testu jest *ujemny*. W zależności od wyniku testu i od tego, czy hipoteza zerowa jest prawdziwa czy fałszywa, możemy mieć do czynienia z wynikiem *prawdziwie dodatnim*, *fałszywie dodatnim*, *fałszywie ujemnym*, bądź też *prawdziwie ujemnym*. Poniższa tabelka ilustruje, kiedy stosuje się poszczególne terminy, a także popularne angielskie nazwy i skróty.

Wynik testu	Prawdziwa hipoteza alternatywna	Prawdziwa hipoteza zerowa
Dodatni (odrzucenie hipotezy zerowej)	Prawdziwie dodatni *true positive* — **TP**	Fałszywie dodatni *false positive* — **FP**
Ujemny (utrzymanie hipotezy zerowej)	Fałszywe ujemny *false negative* — **FN**	Prawdziwie ujemny *true negative* — **TN**

Powyższe dwuliterowe skróty będziemy używali do oznaczania liczby wyników danego typu.

Mocą testu nazywamy prawdopodobieństwo otrzymywania wyniku prawdziwie dodatniego przy założeniu hipotezy alternatywnej (estymatorem mocy jest więc $\frac{TP}{TP+FN}$). Do porównywania naszych testów będziemy używali właśnie mocy, przy ustalonym poziomie istotności ($\alpha = 0{,}05$).

Ponieważ sami zaprojektowaliśmy eksperyment i wiemy, że w testy wykonujemy na próbach dwuwymiarowych, których drugie współrzędne pochodzą z takich samych rozkładów, możemy hipotezę o równości rozkładów sprawdzać tylko dla pierwszych współrzędnych prób, za pomocą jakiegoś testu jednowymiarowego. Postanowiliśmy skorzystać z testu Kołmogorowa-Smirnowa, obliczyć jego moc przy ustalonym poziomie istotności i porównać jego skuteczność z testem najbliższych sąsiadów.

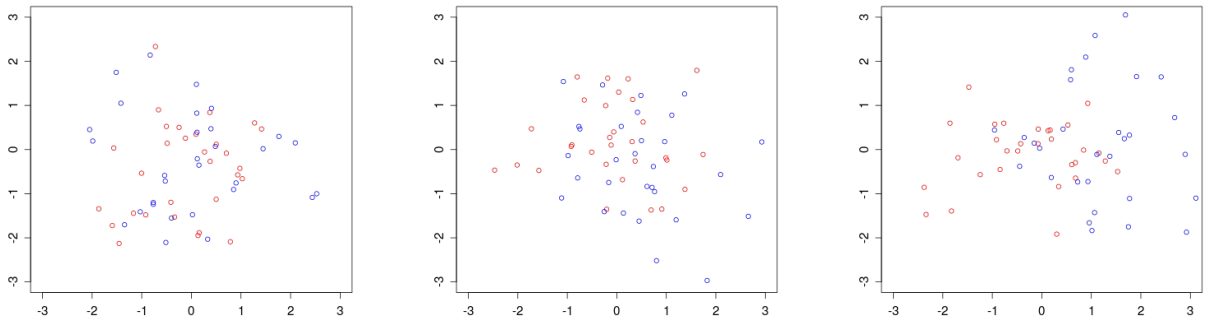
2.2.3. Wyniki eksperymentów

Poniżej prezentujemy opis i wyniki przeprowadzonych eksperymentów. Różnią się one rozkładami, z których losowano próby. Sprawdzaliśmy jak dobrze test najbliższych sąsiadów rozróżnia zmianę średniej oraz zmianę odchylenia standardowego.

Rozróżnianie średniej

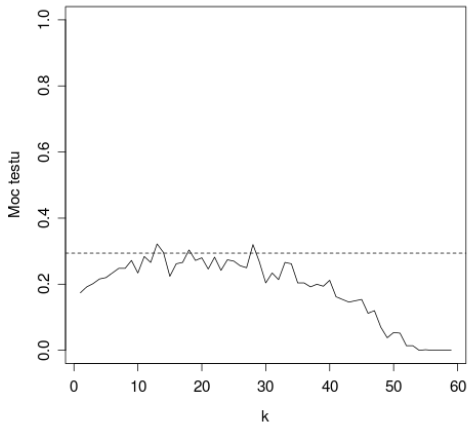
Wykonaliśmy dwa eksperymenty, w każdym w połowie doświadczeń pierwsza współrzędna jednej z prób była wybierana z rozkładu normalnego o średniej μ. W pierwszym eksperymencie przyjęliśmy $\mu = 0{,}5$, a w drugim $\mu = 1$. Oznacza to, że w tych doświadczeniach jedna z prób była trochę przesunięta, a więc nie była też spełniona hipoteza zerowa o równości rozkładów.

Przykładowe próby porównywane w eksperymentach:

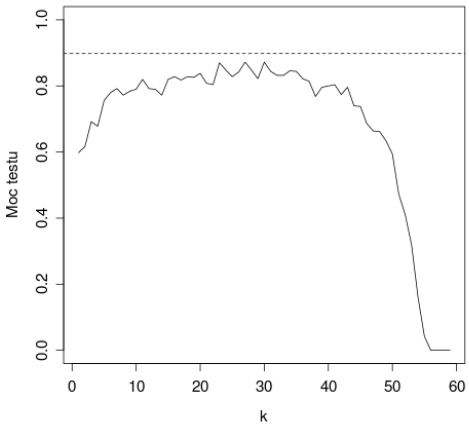


$\mu = 0$ $\mu = 0{,}5$ $\mu = 1$

Porównywaliśmy moc testu najbliższych sąsiadów dla k równego $1, 2, \ldots, 59$. Wyniki obu eksperymentów prezentujemy na poniższych wykresach.



$\mu = 0,5$



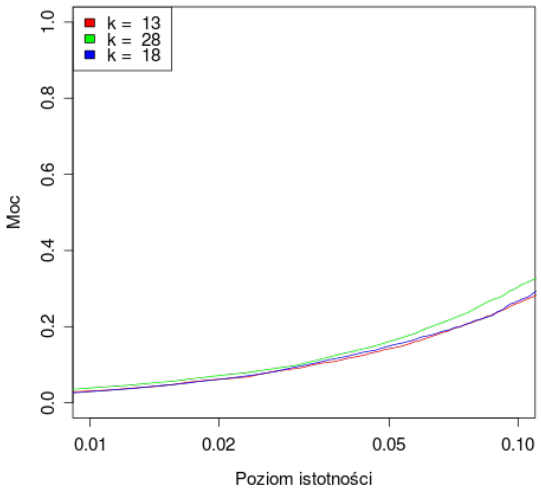
$\mu = 1$

Po pierwsze, widać, że testy radzą sobie dużo lepiej z rozróżnianiem prób o średniej oddalonej o 1. Tego należało się spodziewać, interesujące jest jednak to, że różnica jest aż tak duża, osiągana jest nieomal trzykrotnie większa moc.

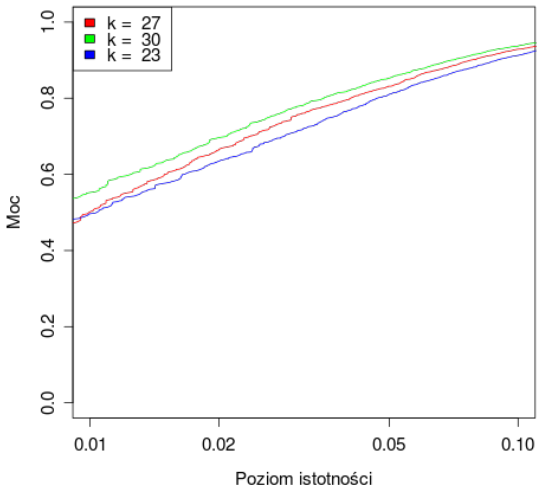
Jeżeli chodzi o wybór optymalnego k to w obu przykładach wartości w przedziale od około 10 do 35 sprawdzały się najlepiej. Natomiast przy zwiększaniu k powyżej 40 jakość przewidywania zdecydowanie się pogarszała. Przy sprawdzaniu powyżej 55 najbliższych sąsiadów mamy bardzo słabe wyniki, ale nie jest to zaskakujące — przy łącznej wielkości prób 60 o wyniku testu w takim wypadku decyduje tylko kilku *najdalszych* sąsiadów każdego punktu.

Przerywane linie na powyższych wykresach prezentują średnie moce testu Kołmogorowa-Smirnowa, użytego do porównywania tylko pierwszej współrzędnej. Możemy zaobserwować, że test ten sprawdził się lepiej niż test k najbliższych sąsiadów, dla większości k.

Dla trzech najlepszych wartości k przyjrzeliśmy się dokładniej zależności między poziomem istotności, a mocą. Wyniki w przedziale poziomu istotności używanego w praktyce (między 0,01, a 0,1) są prezentowane poniżej. Aby zwiększyć dokładność do ich obliczenia użyliśmy osobnego eksperymentu, w którym doświadczenia powtarzaliśmy 20 000 razy.



$\mu = 0,5$



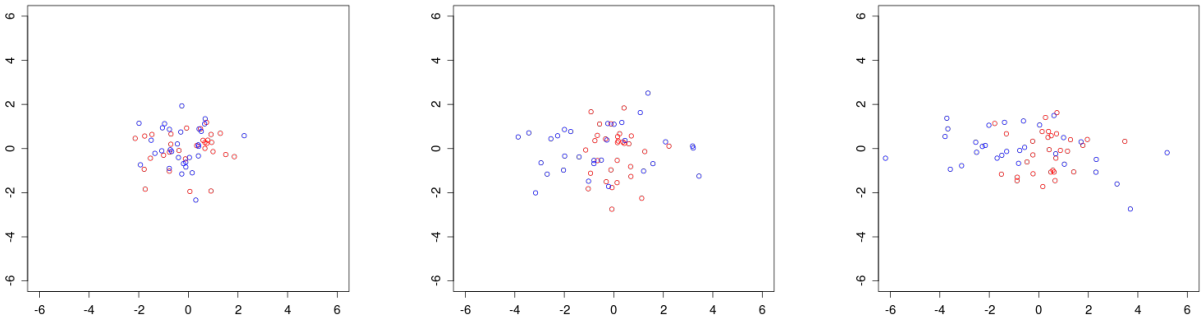
$\mu = 1$

Można zaobserwować dwie rzeczy. Po pierwsze, wykresy dla różnych k są do siebie bardzo podobne. Po drugie, różnice między mocami w początkowych eksperymentach były na tyle małe, że przy większej liczbie powtórzeń inne wartości k okazały się najlepsze.

Rozróżnianie odchylenia standardowego

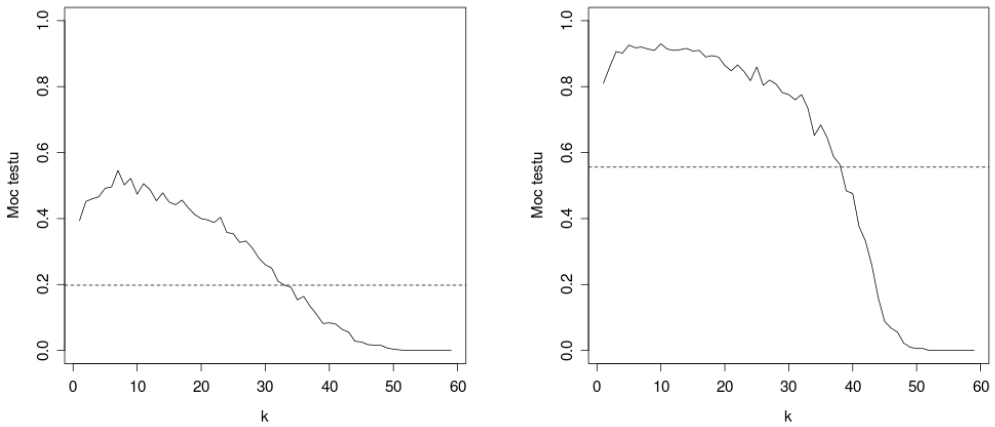
Tym razem w eksperymentach, w połowie doświadczeń losowaliśmy pierwszą współrzędną jednej z prób z rozkładu normalnego o średniej 0, ale o odchyleniu standardowym σ. Używaliśmy σ równego 2 i 3.

Przykładowe próby porównywane w eksperymentach:



$\sigma = 1$ $\sigma = 2$ $\sigma = 3$

Poniżej wykresy mocy uzyskanych przez test najbliższych sąsiadów przy różnych k, wraz z zaznaczoną mocą testu Kołmogorowa-Smirnowa porównującego tylko pierwszą współrzędną.

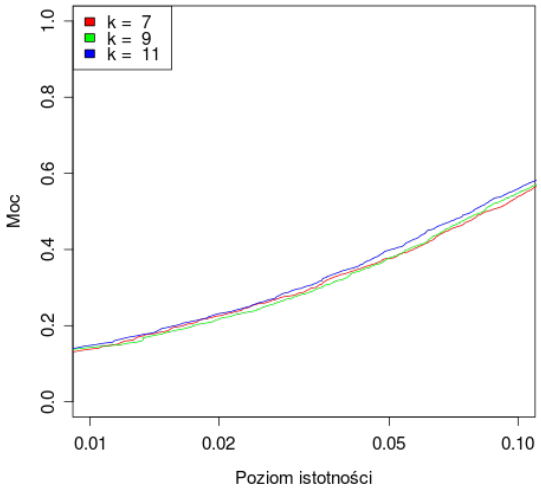


$\sigma = 2$ $\sigma = 3$

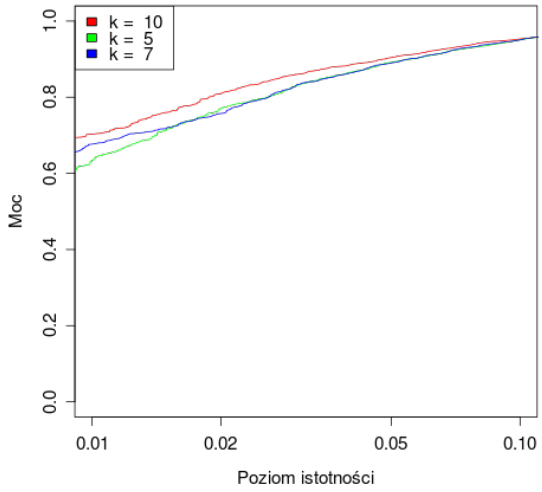
W przypadku $\sigma = 2$ najlepiej sprawdzają się k od 5 do 12 (moc powyżej 0,47), natomiast dla $\sigma = 3$ najlepiej sprawdzają się k od 3 do 16 (moc powyżej 0,9). W obu przypadkach dla k większego od 20 moc zaczyna się zmniejszać, dla $\sigma = 3$ i k w przedziale od 30 do 45 dość gwałtownie. Podobnie jak przy rozróżnianiu oddalenia średniej, testy dla k większego od 50 są bardzo słabe.

W przypadku rozróżniania zmiany odchylenia standardowego test najbliższych sąsiadów okazał się lepszy od testu Kołmogorowa-Smirnowa, o ile nie rozważamy za dużej liczby sąsiadów (k mniejsze od 30 sprawdza się lepiej dla obu sprawdzanych wartości σ).

Tak jak ostatnio, przyjrzyjmy się zależności między poziomem istotności, a mocą, uzyskanymi w eksperymencie z większą liczbą powtórzeń doświadczeń.



$\sigma = 2$



$\sigma = 3$

Ponownie możemy zauważyć, że zależności te nie różnią się zasadniczo dla najlepszych wartości k.

Kod źródłowy skryptu użytego do przeprowadzenia opisanych wyżej testów numerycznych można znaleźć w dodatku A. Został on napisany w postaci algorytmu równoległego, dlatego na maszynie z wieloma procesorami działa kilkukrotnie szybciej. W opisanych wyżej eksperymentach wykonaliśmy na tyle dużo doświadczeń, że zrównoleglanie okazało się istotne — na czterordzeniowym procesorze trwały łącznie około 14 godzin.

Rozdział 3

Metody szacowania precyzji estymatorów

Opiszemy teraz dwie metody, które również zaliczamy do klasy metod repróbkowania, a które służą do szacowania precyzji estymatorów. Wiemy wszakże, że ocena pewnego parametru, obliczona na podstawie jednej próby, może być obarczona pewnym błędem. Chcielibyśmy w jakiś sposób stwierdzić, czy spodziewamy się, że błąd ten jest duży czy mały.

Gdybyśmy wiedzieli, z jakiego rozkładu pochodziła próba, a nie znali jedynie jego parametrów, być może moglibyśmy stwierdzić także jaki jest rozkład wartości naszego estymatora, a więc także jak mocno ocena parametru może odbiegać od jego prawdziwej wartości. Zakładając jednak, że nie mamy takiej wiedzy, spróbujemy znaleźć przybliżony rozkład estymatora na podstawie dostępnej próby poprzez repróbkowanie.

3.1. Bootstrap

Przyjrzyjmy się możliwemu zastosowaniu metody bootstrap. Niech $T(F)$ będzie statystyką, która wyznacza z rozkładu wartość θ charakterystyczną dla tego rozkładu. Mając próbę X z tego rozkładu możemy wyznaczyć przy pomocy T ocenę $\widehat{\theta}$ równą po prostu $T(\widehat{F})$, gdzie $\widehat{F}$ jest rozkładem empirycznym powstałym na podstawie X. Szukamy oszacowania błędu tego estymatora.

Przede wszystkim, przy samym przyjmowaniu estymatora podstawiliśmy po prostu $\widehat{F}$ zamiast F. Jest to podejście jak najbardziej uzasadnione i ogólnie stosowane (w angielskim ma nawet swoją nazwę: *plug-in principle*), gdyż przybliżenie $\widehat{F}$ jest w istocie jedyną informacją o F jaką posiadamy. W takim razie nie jest zaskoczeniem, że do szacowania błędu estymatora (który to błąd jest przecież też cechą właściwą danemu rozkładowi F) także się tym podejściem posłużymy.

3.1.1. Błąd standardowy

Jednym z najbardziej popularnych sposobów na ocenienie jakości estymatora, jest policzenie jego błędu standardowego. Spróbujmy więc właśnie tę wartość obliczyć przy pomocy metody bootstrap.

Przeanalizujmy, jaka jest geneza naszych działań, oraz co chcemy obliczyć. Otóż, z pewnego rozkładu F (nieznanego!) została wylosowana próba $X_1, X_2, \ldots, X_n$, a następnie na jej podstawie obliczyliśmy ocenę $\widehat{\theta}$ interesującej nas statystki T. Brakuje nam ostatniego ogniwa,

przedstawionego na schemacie poniżej, tzn. błędu standardowego:

$$F \quad \longrightarrow \quad X_1, X_2, \ldots, X_n \quad \longrightarrow \quad \widehat{\theta} = T(X) \quad \longrightarrow \quad se_F\left(\widehat{\theta}\right) = \sqrt{Var_F\left[T(X)\right]}.$$

Wartości błędu standardowego nie da się jednak po prostu policzyć, nie znając F. Korzystamy więc teraz z pomysłu, o którym już wspomnieliśmy, a więc zastąpimy F przez $\widehat{F}$. W tym celu musimy zacząć od początku, tzn. założyć, że interesującym nas rozkładem jest właśnie $\widehat{F}$, wylosować z niego próbę, a następnie obliczyć na jej podstawie ocenę naszego parametru używając statystyki T. Błąd standardowy takiego estymatora $(se_{\widehat{F}}(\widehat{\theta^*}))$ będzie naszym przybliżeniem błędu standardowego, którego szukamy $(se_F(\widehat{\theta}))$. O rozkładzie $\widehat{F}$ mamy praktycznie pełną informację, dlatego uda się nam znaleźć przybliżenie rozkładu estymatora $se_{\widehat{F}}(\widehat{\theta^*})$. Drugi schemat przedstawia tzw. świat bootstrap, którego używamy jako przybliżenie świata rzeczywistego:

$$\widehat{F} \quad \longrightarrow \quad X_1^*, X_2^*, \ldots, X_n^* \quad \longrightarrow \quad \widehat{\theta^*} = T(X^*) \quad \longrightarrow \quad se_{\widehat{F}}\left(\widehat{\theta^*}\right) = \sqrt{Var_{\widehat{F}}\left[T(X^*)\right]}.$$

Pozostaje nadal pytanie, jak obliczyć wariancję zmiennej $T(X^*)$, przy założeniu że X^* ma rozkład $\widehat{F}$. I tu właśnie pojawia się istota metody bootstrap. Naszym sposobem będzie bowiem wielokrotne wylosowanie X^* ze znanego rozkładu $\widehat{F}$, obliczenie wartości $T(X^*)$ i podanie oszacowania na wariancję tej zmiennej na podstawie dużej próby z jej rozkładu.

Jeśli przyjmiemy, że repróbkujemy R razy, to z prób tych dostaniemy oceny:

$$\widehat{\theta_i^*} = T\left(X^{*,i}\right), \quad i = 1, 2, \ldots, R. \tag{3.1}$$

Mając próbę $\widehat{\theta_1^*}$, $\widehat{\theta_2^*}$, $\ldots$, $\widehat{\theta_R^*}$ obliczamy z niej błąd standardowy w zwykły sposób:

$$\widehat{se} = \sqrt{\frac{1}{R-1}\sum_{i=1}^{R}\left(\widehat{\theta_i^*} - \overline{\widehat{\theta^*}}\right)^2}, \tag{3.2}$$

gdzie $\overline{\widehat{\theta^*}} = \frac{1}{R}\sum_{i=1}^{R}\widehat{\theta_i^*}$. Szacujemy, że interesujące nas $se_F(\widehat{\theta})$ jest w przybliżeniu równe $\widehat{se}$.

Warto jeszcze zaznaczyć, że losowanie próby X^* z rozkładu $\widehat{F}$ oznacza w praktyce losowanie n elementów ze zwracaniem z próby X, gdyż $\widehat{F}$ jest rozkładem empirycznym powstałym z tejże właśnie próby. Możemy w ten sposób wielokrotne wylosować próbę X^*, np. przy pomocy komputera, o czym więcej w rozdziale 4.

3.1.2. Obciążenie estymatora

Podobnie możemy szacować obciążenie estymatora. Przypomnijmy, że estymatorem nieobciążonym nazywamy taki, którego wartość oczekiwana jest równa estymowanej wartości. W ogólnym zaś przypadku obciążeniem estymatora $\widehat{\theta}$ statystyki o prawdziwej wartości θ nazywamy liczbę

$$bias\left(\widehat{\theta}\right) = E\left[\widehat{\theta} - \theta\right] = E\left[\widehat{\theta}\right] - \theta. \tag{3.3}$$

Zupełnie analogicznie do błędu standardowego, naturalnym oszacowaniem obciążenia przy wykorzystaniu metody bootstrap jest

$$\widehat{bias}\left(\widehat{\theta}\right) = E\left[\widehat{\theta^*} - \widehat{\theta}\right] = \overline{\widehat{\theta^*}} - \widehat{\theta}, \tag{3.4}$$

przy wszystkich oznaczeniach jak przy liczeniu błędu standardowego.

3.2. Jackknife

Jackknife to metoda bardzo zbliżona do bootstrap, w której jednak zamiast losować podpróby z dostępnej próby, opieramy się na usuwaniu kolejno po jednym z elementów próby.

Z próby $X = (X_1, X_2, \ldots, X_n)$ otrzymujemy więc n prób jackknife, przy czym i-ta z nich ma postać:

$$X_{(i)} = (X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n).$$

Podobnie, jak w metodzie bootstrap, dla wartości θ powstałej przez obliczenie pewnej statystyki na podstawie dystrybuanty rozkładu ($\theta = T(F)$), chcemy oszacować rozkład błędu estymatora $\widehat{\theta} = T(\widehat{F}(X))$. W tym celu przyjmijmy $\widehat{\theta}_{(i)} = T(\widehat{F}(X_{(i)}))$.

Przedstawiona powyżej metoda jackknife ma pewne ograniczenia. Może ona nie działać dobrze w przypadku statystyk, które nie są gładkie, tzn. drobna zmiana próby może nieść dużą zmianę wartości estymatora. Przykładem może być mediana, której estymatorem miałaby być mediana próby. Nawet przy bardzo dużej liczbie elementów próby n, usunięcie elementu będącego medianą znacznie zmienia medianę, co powoduje że wyniki otrzymane metodą jackknife nie są wiarygodne.

3.2.1. Błąd standardowy

Korzystając z metody jackknife możemy także szacować błąd standardowy estymatora:

$$\widehat{se}_{jack} = \sqrt{\frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\theta}_{(i)} - \overline{\hat{\theta}_{(\cdot)}}\right)^2}. \tag{3.5}$$

3.2.2. Obciążenie estymatora

Tym razem szacowane obciążenie wyraża się wzorem:

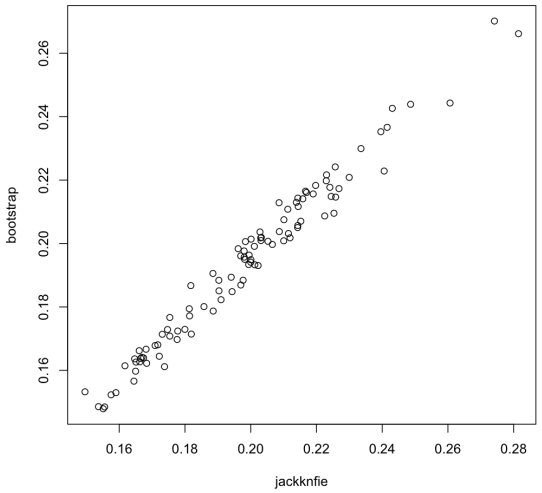
$$\widehat{bias}_{jack} = (n-1)\left(\overline{\hat{\theta}_{(\cdot)}} - \hat{\theta}\right), \tag{3.6}$$

gdzie naturalnie $\overline{\hat{\theta}_{(\cdot)}} = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta}_{(i)}$. Aby sprawdzić, skąd bierze się czynnik $n-1$, warto rozpatrzeć przypadek estymatora wariancji.

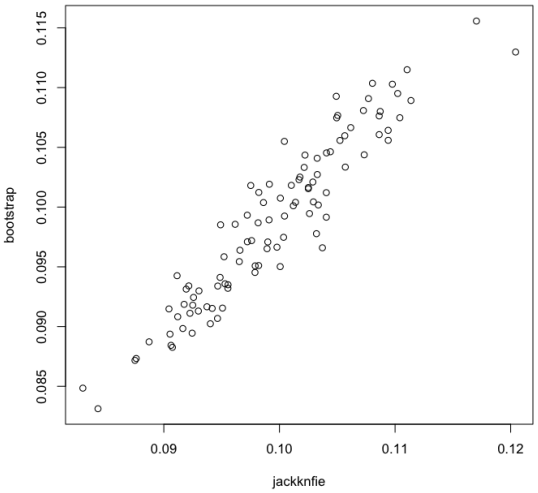
3.3. Porównanie metod

Sprawdźmy teraz, jak obie opisane metody zachowują się w praktyce. W tym celu, wylosujemy pewną próbę X ze standardowego rozkładu normalnego, a następnie sprawdzimy, jak mają się oceny błędu standardowego obliczone metodami bootstrap i jackknife do prawdziwej wartości tego błędu.

Niech N będzie wielkością próby X. Nasz pojedynczy eksperyment polega na wylosowaniu próby X oraz obliczeniu obiema metodami ocen błędu standardowego estymatora średniej. Poniżej przedstawiono wykresy obrazujące wyniki stu takich eksperymentów dla $N = 25$ oraz stu dla $N = 100$. Wiemy jednocześnie, że rzeczywistym błędem standardowym estymatora jest $\frac{1}{\sqrt{N}}$, a więc odpowiednio 0,2 i 0,1. Każdy eksperyment jest symbolizowany przez punkt na wykresie. Podobieństwo tych wykresów do funkcji identycznościowej pokazuje, że obie metody mają bardzo zbliżone do siebie wyniki dla poszczególnych losowań próby X.

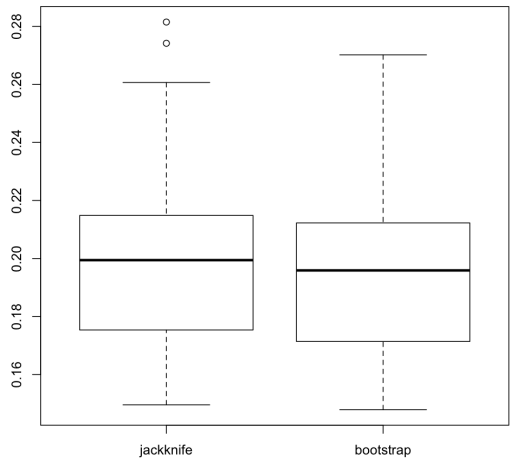


$N = 25$

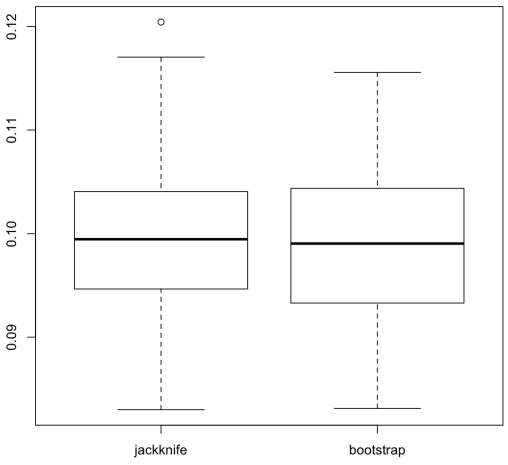


$N = 100$

Dodatkowo, poniżej przedstawiamy wykresy pudełkowe tych samych danych. Widzimy, że nasze oszacowania błędów standardowych nie odbiegają od ich prawdziwych wartości o więcej niż 20-30%, a w większość przypadków — poniżej 10%.



$N = 25$



$N = 100$

Rozdział 4

Zastosowanie pakietu R

Używanie opisanych w pracy metod opartych o repróbkowanie, ze względu na dużą liczbę obliczeń, wymaga w praktyce użycia komputera. W niniejszym rozdziale przedstawiamy jak używać ich w pakiecie statystycznym R. Wybór tego narzędzia nie jest przypadkowy — pakiet R staje się coraz bardziej popularny, a jego możliwości bardzo szybko rosną.

4.1. Permutacyjny test Kołmogorowa-Smirnowa

Wcześniej opisaliśmy permutacyjną wersję testu Kołmogorowa-Smirnowa służącą do sprawdzania równości dwóch rozkładów. Teraz pokażemy jak przeprowadzić taki test w pakiecie R.

Załóżmy, że mamy dane dwa wektory `x` i `y` zawierające `n1` i `n2` obserwacji jednowymiarowych zmiennych o rozkładach F_X i F_Y odpowiednio. Będziemy testować hipotezę $H_0 : F_X = F_Y$ przeciw alternatywnej $H_1 : F_X \neq F_Y$.

4.1.1. Sprawdzenie wszystkich permutacji

Jeżeli liczba obserwacji jest mała, to możemy próbować obliczyć wartość statystyki testowej dla wszystkich możliwych podziałów zbioru prób. W tym celu możemy wykorzystać funkcję `combn(n, m)`, która generuje wszystkie m-elementowe permutacje zbioru $\{1, 2, \ldots, n\}$ i umieszcza je w kolejnych kolumnach wynikowej macierzy.

Poniżej znajduje się kod funkcji zwracającej p-wartość testu permutacyjnego. `t0` to wartość statystyki testowej dla wyjściowego podziału obserwacji na zbiory `x` i `y`, natomiast w wektorze `reps` odkładane są wartości statystyki testowej dla kolejnych podziałów obserwacji. Podziały te są generowane na podstawie tablicy `combinations`, która zawiera wszystkie kombinacje `n1`-elementowe zbioru $\{1, 2, \ldots,$ `n1+n2`$\}$.

```
ksp_test <- function(x, y) {
  n1 <- length(x)
  n2 <- length(y)
  options(warn = -1)
  z <- c(x, y)
  t0 <- ks.test(x, y)$statistic

  combinations = combn((n1 + n2), n1)
  reps <- numeric(ncol(combinations))
  for (i in 1:ncol(combinations)){
```

```
    x1 <- z[combinations[,i]]
    y1 <- z[-combinations[,i]]
    reps[i] <- ks.test(x1, y1)$statistic
  }
  options(warn = 0)
  mean(reps >= t0)
}
```

Działanie polecenia `mean(reps >= t0)` jest odrobinę bardziej skomplikowane. `reps >= t0` zwraca wektor wartości logicznych, który zawiera `TRUE` odpowiadające podziałom zbioru obserwacji, które dają wartość statystyki Kołmogorowa-Smirnowa nie mniejszą niż `t0` i `FALSE` odpowiadające pozostałym podziałom. Przy obliczaniu średniej następuje automatyczna konwersja tych wartości do 1 i 0 odpowiednio, a więc sumą tego wektora jest liczba tych podziałów, które mają wartość statystyki testowej nie mniejszą niż `t0`. Po podzieleniu tej sumy przez liczbę wszystkich możliwych podziałów otrzymujemy, zgodnie ze wzorem (1.2), p-wartość testu permutacyjnego Kołmogorowa-Smirnowa.

Zwróćmy jeszcze uwagę na fakt, że na czas obliczeń zmieniona została flaga ustawień `warn` na wartość -1. W przeciwnym przypadku każdorazowe wywołanie funkcji `ks.test` mogłoby ostrzegać nas, że obliczona przez nie p-wartość może być niepoprawna[1]. Możemy spokojnie zignorować to ostrzeżenie, gdyż nie będziemy korzystali ze zwróconej p-wartości, a jedynie z wartości statystyki testowej.

Jak wspomnieliśmy w rozdziale 1.2, sprawdzanie wszystkich podziałów zbioru obserwacji jest możliwe tylko dla bardzo małych danych. Dla przykładu już przy porównywaniu prób wielkości 14 i 12 ze zbioru danych `chickwts`[2] trzeba sprawdzić $\binom{26}{14} = 9\,657\,700$ podziałów. Obliczenia te na nowoczesnym komputerze osobistym zajęły ponad dwie godziny. Przykład ten pokazuje, że w praktyce nie ma możliwości sprawdzania wszystkich możliwych podziałów gdy próby mają duży rozmiar, gdyż czas będzie rósł wykładniczo względem wielkości prób.

4.1.2. Metoda Monte Carlo

Zaprezentujemy teraz implementację opisanej wcześniej metody Monte Carlo. Zamiast sprawdzać wszystkie podziały możemy losowo wybierać pewną liczbę podziałów i obliczyć przybliżony osiągnięty poziom istotności na ich podstawie. W R możemy tę metodę zaimplementować bardzo podobnie do poprzedniej, wybierając losowy podział za pomocą funkcji `sample`:

```
ksp_test_MC <- function(x, y, R = 999) {
  n1 <- length(x)
  n2 <- length(y)
  options(warn = -1)
  z <- c(x,y)
  t0 <- ks.test(x, y)$statistic
  reps <- numeric(R)

  for (i in 1:R) {
    k <- sample((n1 + n2), size = n1, replace = FALSE)
    x1 <- z[k]
```

[1] Autorzy implementacji testu Kołmogorowa-Smirnowa założyli, że powtarzające się wartości obserwacji przeczą założeniu o ciągłości rozkładów, które jest potrzebne do obliczenia p-wartości testu.

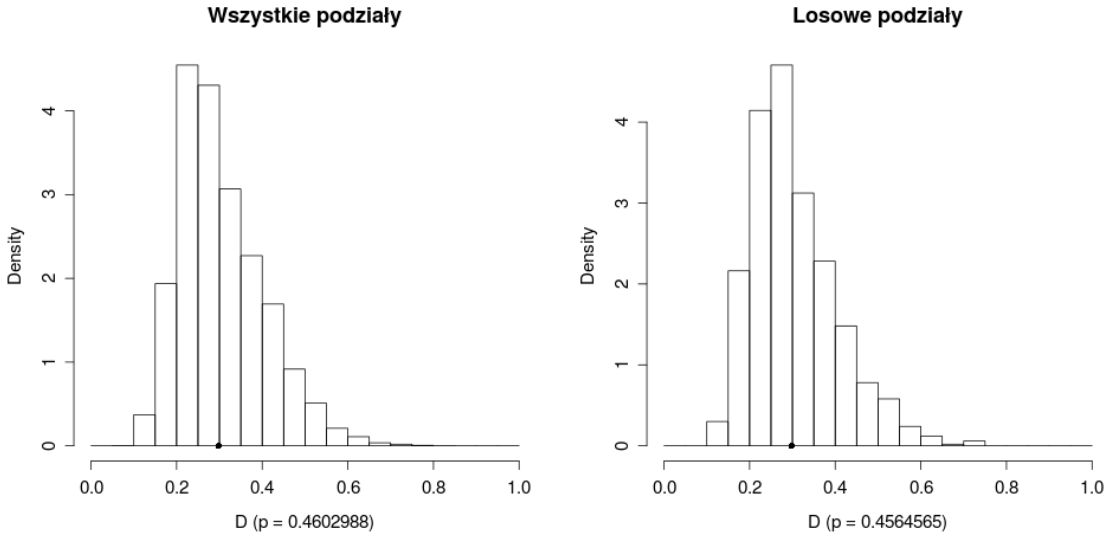
[2] Standardowy zbiór danych dołączony do pakietu R. Zawiera wagi kurczaków karmionych różnymi paszami.

```
    y1 <- z[-k]
    reps[i] <- ks.test(x1, y1)$statistic
  }

  options(warn = 0)
  mean(c(t0, reps) >= t0)
}
```

W powyższej funkcji domyślną liczbę losowań (`R`) ustaliliśmy na 999. W książce [Rizzo] rekomendowana jest liczba iteracji od 99 do 999.

Czas wykonania powyższej procedury wynosi mniej niż sekundę, tak więc poprawa jest wyraźna. Zastanawiać może, czy wyniki powyższych dwóch metod (dokładnej i Monte Carlo) różnią się od siebie. Poniżej prezentujemy histogramy rozkładów statystyki testowej uzyskanych przy porównywaniu dwóch populacji ze wspomnianego zbioru danych `chickwts`.



Widzimy, że są one do siebie podobne. Bliskie siebie są też p-wartości uzyskane w testach permutacyjnych. Wynoszą 0.4602988 i 0.4564565 dla sprawdzenia odpowiednio wszystkich podziałów i tylko 999 losowych.

4.1.3. Wykorzystanie funkcji *boot*

Aby zrealizować test permutacyjny z losowym wybieraniem podziałów możemy też skorzystać z funkcji `boot` z pakietu o tej samej nazwie. Funkcja ta służy do przeprowadzania testów opartych o repróbkowanie. Poniżej opis ważniejszych argumentów funkcji:

- `data` — zbiór obserwacji, na którym będzie przeprowadzany test,
- `statistic` — funkcja obliczająca wartość statystyki na podstawie zbioru danych i pewnej permutacji (być może z powtórzeniami) numerów obserwacji. Przy omawianych teraz testach testach będzie to permutacja bez powtórzeń wyznaczająca podział obserwacji na dwa zbiory,
- `R` — liczba permutacji indeksów, które zostaną wylosowane, czyli innymi słowy liczba iteracji algorytmu.

- `sim` — ciąg znakowy określający typ symulacji, a więc sposób w jaki będą losowane tablice indeksów przekazywane wyżej opisanej funkcji. Użycie argumentu *permutation* spowoduje generowanie losowych ciągów indeksów bez powtórzeń, które właśnie potrzebujemy do testu permutacyjnego.

- dodatkowe argumenty, których nie spodziewa się funkcja `boot` zostaną bez zmian przekazane do funkcji liczącej statystyki.

Funkcja `boot` przeprowadza test, a jego wyniki zwraca w postaci obiektu. Najważniejszymi jego polami są dla nas `t0` i `t`. Pierwszy z nich jest wartością statystyki testowej dla początkowego podziału (czyli dla wektora indeksów $\{1, 2, \ldots, n1 + n2\}$, natomiast drugi to wektor wartości tej statystyki dla kolejnych wylosowanych permutacji indeksów.

Poniższa funkcja w języku R wykonuje ten sam test co poprzedni, ale przy pomocy funkcji `boot`:

```
ksp_test_MC_2 <- function(x, y, R = 999) {
  ks_stat <- function(data, ix, n1, n2){
      x <- data[ix[1:n1]]
      y <- data[ix[(n1+1):(n1+n2)]]
      ks.test(x, y)$statistic
  }

  library(boot)
  z <- c(x, y)
  boot.obj <- boot(data = z, statistic = ks_stat,
         sim = "permutation", R = 999, n1 = nrow(x), n2 = nrow(y))
  mean(c(boot.obj$t, boot.obj$t0) >= boot.obj$t0)
}
```

Funkcja `ks_stat` oblicza statystykę testową Kołmogorowa-Smirnowa dla wektora danych `data`, przy czym podział na dwie populacje wyznacza permutacja indeksów `ix`. Parametry `n1` i `n2` określają wielkości prób i są właśnie tymi nadmiarowymi argumentami, które zostają przekazane funkcji liczącej statystkę testową — w powyższym przykładzie będą to zawsze długości wektorów x i y odpowiednio.

4.2. Test najbliższych sąsiadów

Pokażemy teraz jak zaimplementować permutacyjny test najbliższych sąsiadów w pakiecie R. Tak jak w powyższych przykładach będziemy chcieli skorzystać z funkcji `boot`, więc będziemy musieli napisać tylko metodę obliczającą odpowiednią statystykę. Funkcja o nazwie `ann` z pakietu `yaImpute`, która znajduje właśnie kolejnych najbliższych sąsiadów dla każdego punktu w podanym zbiorze, wykona za nas większość pracy. Przyjrzyjmy się jej kilku wybranym parametrom:

- `ref` – macierz zawierająca w kolejnych wierszach współrzędne punktów, wśród których będą szukani najbliżsi sąsiedzi,

- `target` – macierz z punktami dla których będziemy szukali najbliższych sąsiadów spośród zbioru `ref`, sformatowana tak jak poprzednia,

- `k` – liczba najbliższych sąsiadów, których należy znaleźć.

My będziemy szukali najbliższych sąsiadów dla każdej obserwacji w próbie, wśród pozostałych obserwacji, dlatego też zbiory `ref` i `target` będą takie same. Powyższa funkcja zwraca macierz o $2k$ kolumnach. Pierwsze k kolumn zawiera indeksy najbliższych sąsiadów, a kolejne k kolumny zawierają odległości danego punktu od tych sąsiadów. Co więcej, każdy punkt będzie uznany za swojego najbliższego sąsiada, co nie jest zbyt interesujące. Dlatego też będziemy przekazywać do funkcji k o jeden większe i ignorować pierwszych najbliższych sąsiadów (pierwszą kolumnę).

Kompletna funkcja do liczenia k–tej statystyki najbliższych sąsiadów jest prezentowana poniżej. Argument `z` to macierz zawierająca wszystkie obserwacje, wektor `ix` wskazuje podział na próby, a `n1` i `n2` to liczebności tych prób.

```
Tk <- function(z, ix, n1, n2, k) {
    z = z[ix,]
    NN <- yaImpute::ann(ref=z, target=z, k=k+1)$knnIndexDist[,2:(k+1)]
    I1 <- NN[1:n1,] < n1 + 0.5
    I2 <- NN[(n1+1):(n1+n2),] > n1 + 0.5
    I <- rbind(I1, I2)
    sum(I) / (k * (n1 + n2))
}
```

Macierze `I1` i `I2` zawierają wartości funkcji oznaczonych przez $I_i(k)$ w rozdziale 2.1. Dokładnie rzecz biorąc w wektorze `I1` wartość `TRUE` będzie ustawiona w i–tym rzędzie ($i <$ `n1`) i j–tej kolumnie wtedy i tylko wtedy, gdy j–ty najbliższy sąsiad obserwacji i pochodził z pierwszej próby. Podobnie w wektorze `I2` wartości `TRUE` odpowiadają tym spośród najbliższych sąsiadów obserwacji z drugiej próby, które też należą do drugiej próby. Nietrudno zauważyć, że wartości `TRUE` odpowiadają składnikom sumy we wzorze (2.7). Wystarczy je zatem dodać i podzielić przez odpowiedni mianownik, by otrzymać szukaną wartość statystyki najbliższych sąsiadów.

Teraz wystarczy już uruchomić funkcję `boot` i z jej wyników obliczyć p–wartość testu permutacyjnego najbliższych sąsiadów. Poniżej użyjemy tego testu na sztucznych danych: losowych punktów na płaszczyźnie o współrzędnych będących próbami z rozkładów normalnych.

```
x <- matrix(rnorm(20), nrow=10, ncol=2)
y <- matrix(rnorm(20,1), nrow=9, ncol=2)
z <- rbind(x, y)
k <- 2

library(boot)
R <- 999
boot.obj <- boot(data = z, R = R, sim="permutation", statistic=Tk,
                                    n1 = 10, n2 = 9, k = k)
p <- sum(boot.obj$t >= boot.obj$t0) / (R+1)
p
```

Wykonując powyższy kod uzyskaliśmy p–wartość równą 0,011 (naturalnie jest to wynik losowy i może się nieco różnić przy każdym kolejnym uruchomieniu), co pozwoliłoby odrzucić hipotezę zerową o jednakowości rozkładów (przy założeniu poziomu istotności 0,05). Jest to wynik zgodny z oczekiwaniami, bo przecież X i Y zostały wylosowane z rozkładów normalnych o różnych średnich!

4.3. Bootstrap i jackknife

Poznaliśmy już dokładnie funkcję `boot`, także przeprowadzenie metody bootstrap nie przedstawi nam żadnych trudności. Dla wygody napiszmy funkcję, która będzie zwracała próbę $\widehat{\theta^{*i}}$ (oznaczenia jak w rozdziale 3. — `X` to dane, a `ES` statystyka której estymator nas interesuje):

```
bootstrap <- function(X, ES, R = 1000) {
  indices_ES <- function(D, ix) {
      ES(D[ix])
  }
  boot.obj <- boot(data = X, statistic = indices_ES, R)
  boot.obj$t
}
```

Zauważmy, że tym razem nie musieliśmy ustawić parametru `sim` w funkcji `boot`. Jego domyślna wartość powoduje losowanie ze zwracaniem ze zbioru obserwacji, którego właśnie potrzebujemy w tej metodzie.

Błąd standardowy, jaki uzyskuje nasz estymator będziemy przybliżać przez wynik obliczenia

```
sd(bootstrap(X, ES)).
```

Pokażemy teraz jak przeprowadzić metodę jackknife w takiej samej sytuacji jak ta opisana powyżej, także do obliczenia błędu standardowego. Implementacja jest bardzo prosta, polega na n-krotnym obliczeniu sprawdzanego estymatora, a następnie obliczeniu błędu standardowego zgodnie ze wzorem (3.5).

```
jackknife_sd <- function(X, ES) {
  n <- length(X)
  ests <- numeric(n)
  for (i in 1:n)
    ests[i] <- ES(X[-i])
  sqrt((n-1) * mean((ests - mean(ests))^2))
}
```

Rozdział 5

Zastosowanie metod repróbkowania do danych ECAP

Dane, które analizujemy, pochodzą z projektu o nazwie „Epidemiologia Chorób Alergicznych w Polsce". W jego trakcie przeprowadzono ankietę na prawie 23 tysiącach osób, a część z nich poddano dalszemu badaniu lekarskiemu. Zbierane były informacje o genetycznych i środowiskowych czynnikach, które mogą mieć wpływ na występowanie chorób alergicznych, w tym astmy.

Dane, do których mieliśmy dostęp, zawierały wyniki dla 5000 osób, głównie dzieci. Wśród zebranych cech można było znaleźć informacje o:

- cechach indywidualnych: płci, wzroście, wadze, wieku rozpoczęcia szkoły,
- ośrodku przeprowadzającym badania,
- występowania alergii u matki i ojca,
- sposobu ogrzewania mieszkania,
- posiadanych zwierzętach domowych: psach, kotach, ptakach.

5.1. Zależność zachorowalności od płci

Postanowiliśmy sprawdzić, czy dane sugerują niezależność występowania astmy u osób różnych płci. Poniżej przedstawiam tablicę kontyngencji reprezentującą zależność między tymi dwoma cechami u badanych osób.

		Astma	
		Nie	Tak
Płeć	Kobieta	2312	113
	Mężczyzna	2403	165

Na pierwszy rzut oka wydaje się, że cechy te mogą być zależne: przy podobnej liczbie kobiet i mężczyzn, ci drudzy wyraźnie częściej chorują na astmę. Sprawdzimy teraz, czy ta zaobserwowana zależność jest statystycznie istotna.

W tym celu można się posłużyć, którymś z testów porównujących niezależność cech w tablicy kontyngencji, na przykład testem chi-kwadrat. Przy liczeniu p-wartości w tym teście używa się jednak przybliżenia rozkładu statystyki testowej rozkładem asymptotycznym. To przybliżenie, przy małej liczbie obserwacji, może być niedokładne. My jednak poznaliśmy metodę testów permutacyjnych, która pozwala na obliczenie p-wartości bez znajomości rozkładu statystki testowej i nią się tu posłużymy.

Dane do testu można wyobrazić sobie jako macierz o dwóch kolumnach, zawierających dane o płci i zachorowalności na astmę, w której każdy wiersz reprezentuje pojedynczego badanego. Wyznaczenie statystyki testowej polega na zagregowaniu danych do postaci tablicy kontyngencji, a następnie obliczenie wartości statystki chi-kwadrat. Zatem przeprowadzenie testu permutacyjnego (metodą Monte Carlo) będzie polegało na wielokrotnym losowym permutowaniu drugiej kolumny, przy zachowaniu pierwszej, a następnie obliczaniu statystki testowej. Na koniec zliczymy te losowania, w których uzyskaliśmy statystykę większą od początkowej, aby zgodnie ze wzorem (1.6) otrzymać p-wartość.

Po przeprowadzeniu tej procedury okazało się, że uzyskana została p-wartość równa 0,006. Sugeruje ona odrzucenie hipotezy zerowej, którą jest niezależność rozkładów. Jest to zgodne z naszym początkowym spostrzeżeniem, że mężczyźni częściej chorują na astmę.

Wynik ten znalazł swoje potwierdzenie w literaturze medycznej, na przykład w publikacji [Pirożyński] można znaleźć informacje (opartą o inny zbiór danych), że płeć męska należy do jednej z cech powiązanych[1] z częstszą zachorowalnością na astmę.

5.2. Porównanie wzrostu

Aby zbadać, czy chorowanie na astmę jest w jakiś sposób związane ze wzrostem osób, postanowiliśmy wykorzystać test najbliższych sąsiadów. Nie jest to jedyny możliwy do wykonania test, szczególnie że zmienna jest jedno- a nie wielowymiarowa. Nic nie stoi jednak na przeszkodzie, by także takie zmienne badać tym testem.

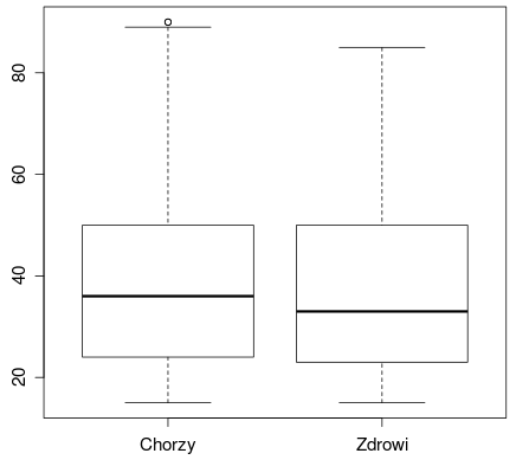
Przede wszystkim, z dostępnych danych wybieramy dwie interesujące nas próby: X — listę wzrostów osób z astmą, oraz Y — listę wzrostów osób bez tej choroby. Dla tak przygotowanych danych wykonujemy test k najbliższych sąsiadów dla $k = 10$. Wartość k została przez nas wybrana przy uwzględnieniu obserwacji z rozdziału 2.2, biorąc też pod uwagę dość dużą liczbę obserwacji w próbach.

Otrzymana p-wartość równa 0,909 sugeruje, iż nie ma podstaw do odrzucenia hipotezy zerowej o równości rozkładów.

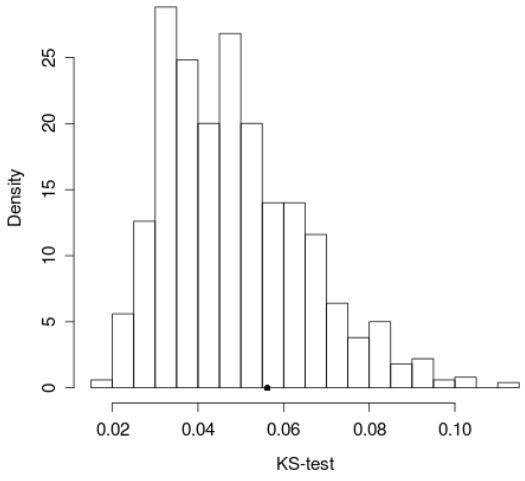
[1] Płeć została określona jako *słabiej powiązana*, wskazano też inne, ważniejsze czynniki.

5.3. Porównanie wagi

Postanowiliśmy także sprawdzić czy fakt bycia chorym na astmę wpływa w sposób istotny na wagę danej osoby.



Po obejrzeniu wykresu pudełkowego doszliśmy do wniosku, że rozkłady wagi wśród osób chorych i zdrowych są do siebie bardzo podobne. Za pomocą testu permutacyjnego sprawdziliśmy hipotezę zerową, że mają one ten sam rozkład, przeciw alternatywnej, że się różnią. Podobnie jak we wcześniejszych rozdziałach korzystaliśmy ze statystyki używanej w teście Kołmogorowa-Smirnowa. Oto histogram przedstawiający uzyskany przez nas rozkład statystyki testowej:



Uzyskany poziom istotności wyniósł 0,295, a więc nie daje podstaw do odrzucenia hipotezy zerowej.

5.4. Błąd standardowy estymatora średniej wagi

Możemy też podejść do tematu wagi pacjentów z innej strony i zastanowić się, jaka jest średnia waga dziecka chorego na astmę. Nie jest to może zbyt praktyczny przykład, szczególnie że nie znany jest nam wiek badanych. Celowo używamy go jednak, aby zaprezentować najważniejsze zastosowanie metod bootstrap i jackknife.

Przede wszystkim, generujemy wektor X wag osób chorych na astmę z naszych danych, a następnie liczymy ich średnią, która wynosi 37,44 kilograma. Jest to naturalny estymator średniej wagi chorego. Możemy teraz spróbować obliczyć błąd standardowy tego estymatora. W celu jego oszacowania używamy obu poznanych metod i porównujemy wyniki. Błąd standardowy obliczony za pomocą:

- bootstrap to 0,9655,
- a jackknife to 0,9551.

Rozdział 6

Wkład autorów

Praca była pisana przez nas wspólnie, obaj mieliśmy wpływ na jej całość. Powstała ona na podstawie materiałów przygotowywanych i prezentowanych przez na nas wspólnie na proseminarium *Zastosowania statystyki w biologii i medycynie* w roku akademickim 2010/11. Z powyższych powodów dość trudno jest określić wkład poszczególnych autorów. W związku z formalnym wymaganiem stworzenia tego rozliczenia, poniżej przypisujemy osobę spisującą poszczególne rozdziały:

Wprowadzanie	Błażej Osiński
1. Testy permutacyjne dla obserwacji jednowymiarowych	Błażej Osiński
2. Testowanie równości rozkładów wielowymiarowych	Tomasz Kulczyński
2.2. Dobór optymalnej liczby sąsiadów	Błażej Osiński
3. Metody szacowania precyzji estymatorów	Tomasz Kulczyński
4. Zastosowanie pakietu R	
4.1. Permutacyjny test Kołmogorowa-Smirnowa	Błażej Osiński
4.2. Test najbliższych sąsiadów	Tomasz Kulczyński
4.3. Bootstrap i jackknife	Błażej Osiński
5. Zastosowanie metod repróbkowania do danych ECAP	
5.1. Zależność zachorowalności od płci	Błażej Osiński
5.2. Porównanie wzrostu	Tomasz Kulczyński
5.3. Porównanie wagi	Błażej Osiński
5.4. Błąd standardowy estymatora średniej wagi	Tomasz Kulczyński

Dodatek A

Skrypt w R wykonujący eksperymenty numeryczne

```
library(boot)
library(ROCR)

exhaustive_test <- function(N, M, k, mod_rnorm){
  knntest <- function(x, y, K){
    # Wykonuje test K najbliższych sąsiadów dla próbek x i y.
    k <- K+1
    z <- rbind(x,y)
    NN.idx <- function(x, k=NROW(x)) {
        x <- as.matrix(x)
        k <- min(c(k+1, NROW(x)))
        NN <- yaImpute::ann(ref=x, target=x,
                  tree.type="kd", k=k, verbose=FALSE)
        idx <- NN$knnIndexDist[,1:k]

        # W pierwszej kolumnie jest zawsze dany wierzchołek,
        # jako swój najbliższy sąsiad, nas interesują pozostali sąsiedzi.
        nn.idx <- idx[,-1]
        row.names(nn.idx) <- idx[,1]
        nn.idx
    }
    nn.idx <- NN.idx(z, k=k)

    fperm <- function(x, perm){
      return (perm[x]);
    }

    TnK <- function(z, ix=1:NROW(z), sizes) {
        # Oblicza statystykę najbliższych sąsiadów T(n,k).
        z <- as.matrix(z)
        n1 <- sizes[1]
        n2 <- sizes[2]
        n <- n1 + n2
```

```
        z <- as.matrix(z[ix, ])
        block1 <- fperm(nn.idx[ix[1:n1], ], ix)
        block2 <- fperm(nn.idx[ix[(n1+1):n], ], ix)
        i1 <- sum(block1 < n1 + .5)
        i2 <- sum(block2 > n1 + .5)
        return ((i1 + i2) / (k * n))
    }

    boot.obj <- boot(data = z, statistic = TnK,
      sim = "permutation", R = 999, sizes = c(M,M))
    # Im wiekszą statystyka, tym bardziej prawdopodobne,
    # ze próbki pochodzą z różnych rozkładów.
    return(mean(c(boot.obj$t,boot.obj$t0) >= boot.obj$t0))
  }

  results <- numeric(N)
  for (i in 1:N){
    X <- cbind(rnorm(M), rnorm(M))
    # W połowie testów losujemy próbki z tego samego rozkładu,
    # a w połowie jedna współrzedna w jednej próbce jest inna.
    if (i <= N/2) {
      Y <- cbind(rnorm(M), rnorm(M))
    }
    else {
      Y <- cbind(mod_rnorm(M), rnorm(M))
    }
    results[i] <- knntest(X, Y, k)
  }
  return (list(k=k, pvals=results))
}

mod_mean = function(d) {
  function (N) {
    return (rnorm(N, d))
  }
}

mod_sd = function(sd) {
  function (N) {
    return (rnorm(N, sd=sd))
  }
}

mods = c(mod_sd(2), mod_sd(3), mod_mean(0.5), mod_mean(1))

M = 30
N = 1000
KK = 59;
```

```
perform_test = function (mod) {
  res = list()
  for (k in 1:KK){
    res = c(res, exhaustive_test(N, M, k, mod))
  }
  return(res)
}

results = mclapply(mods, perform_test)
```

Dodatek B

Skrypty w R użyte do doświadczeń

B.1. Permutacyjny test chi-kwadrat (rozdział 5.1)

```
dane <- read.table("ecap.csv", header=TRUE, sep=";")
dane = subset(dane, astma != "Kategoryczna odmowa odpowiedzi")
attach(dane)

chisq_stat <- function(data, ix){
  chisq.test(table(data$Plec, data[ix,]$astma)[,-1], correct=FALSE)$statistic
}

boot.obj <- boot(data = dane, statistic = chisq_stat,
  sim = "permutation", R = 999)
mean(c(boot.obj$t, boot.obj$t0) >= boot.obj$t0)
```

B.2. Test najbliższych sąsiadów (rozdział 5.2)

```
dobrywzrost = subset(dane, Wzrost > 0)
X = subset(dobrywzrost, astma == 'Tak')$Wzrost
Y = subset(dobrywzrost, astma == 'Nie')$Wzrost

Tk <- function(z, ix, n1, n2, k) {
    z = z[ix,]
        ANN <- yaImpute::ann(ref=matrix(z), target=matrix(z), k=k+1)
        NN <- ANN$knnIndexDist[,2:(k+1)]
        I1 <- NN[1:n1,] < n1 + 0.5
        I2 <- NN[(n1+1):(n1+n2),] > n1 + 0.5
                        I <- rbind(I1, I2)
                        sum(I) / (k * (n1 + n2))
}

z <- rbind(matrix(X), matrix(Y))
k <- 10
R <- 999

library(boot)
```

```
boot.obj <- boot(data = z, R = R, sim="permutation", statistic=Tk,
                 n1 = length(X), n2 = length(Y), k = k)
```

B.3. Test Kołmogorowa-Smirnowa (rozdział 5.3)

```
dobrawaga = subset(dane, Waga > 0)
bezastmy = subset(dobrawaga, astma == 'Nie')$Waga
zastma = subset(dobrawaga, astma == 'Tak')$Waga

boxplot(bezastmy, zastma, names=c('Chorzy','Zdrowi'))

R <- 999
z <- c(bezastmy, zastma)
n <- length(bezastmy)
m <- length(zastma)
K <- 1:(n+m)
reps <- numeric(R)

options(warn = -1)
t0 <- ks.test(bezastmy, zastma)$statistic

for (i in 1:R){
  # generate indices k for the first sample
  k <- sample(K, size = n, replace = FALSE)
  x1 <- z[k]
  y1 <- z[-k]
  reps[i] <- ks.test(x1,y1,exact=FALSE)$statistic
}
p <- mean(c(t0, reps) >= t0)

options(warn = 0)

hist(reps, main = "", freq=FALSE, breaks = "scott",, xlab = "KS-test")
points(t0, 0, cex = 1, pch = 16)
```

B.4. Bootstrap i jackknife (rozdział 5.4)

```
dobrawaga = subset(dane, Waga > 0)
zastma = subset(dobrawaga, astma == 'Tak')$Waga

library(boot)

bootstrap <- function(X, ES, R = 1000) {
    indices_ES <- function(D, ix) {
        ES(D[ix])
    }
    boot.obj <- boot(data = X, statistic = indices_ES, R)
    boot.obj$t
```

```
}

jackknife_sd <- function(X, ES) {
    n <- length(X)
    ests <- numeric(n)
    for (i in 1:n)
        ests[i] <- ES(X[-i])
    sqrt((n-1) * mean((ests - mean(ests))^2))
}

sd(bootstrap(zastma, mean))
jackknife_sd(zastma, mean)
```

B.5. Porównanie metod (rozdział 3.3)

```
one_test <- function(N, R=100) {
    res = matrix(nrow=R, ncol=2)
    for(i in 1:R) {
        x = rnorm(N)
        res[i,1] = jackknife_sd(x, mean)
        res[i,2] = sd(bootstrap(x, mean))
    }
    res
}

res25 = one_test(25)
res100 = one_test(100)
```

Bibliografia

[Rizzo] Rizzo M. L., *Statistical Computing with R*, 2008

[Efron] Efron B., Tibshirani R. J., *An Introduction to the Bootstrap*, 1993

[Sing] Sing T., Sander O., Beerenwinkel N., Lengauer T., *ROCR: visualizing classifier performance in R*, Bioinformatics 21 (2005) 3940–3941.

[Venables] Venables W. N., Smith D. M. et al., *An Introduction to R*, 2010

[Pirożyński] Pirożyński M., Solarski Z., *Astma oskrzelowa*, Postępy Nauk Medycznych 11 (2007), s. 466-478.