O'REILLY®

Wydanie II

Statystyka praktyczna w data science

50 kluczowych zagadnień w językach R i Python

Peter Bruce
Andrew Bruce
Peter Gedeck

Helion

Tytuł oryginału: Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python, 2nd Edition

Tłumaczenie: Krzysztof Sawka, Marta Danch-Wierzchowska

ISBN: 978-83-283-7427-0

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/stpra2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

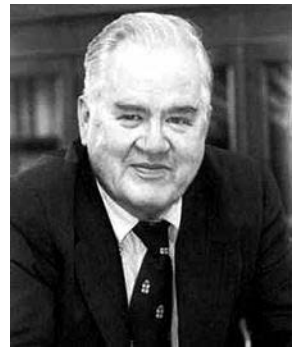
Spis treści

ROZDZIAŁ 1.

Badania eksploracyjne

Pierwszy rozdział skupia się na tym, co zawsze jest pierwszym krokiem w dowolnym projekcie data science: przejrzeniu zebranych danych.

Badania eksploracyjne (**EDA**, ang. *exploratory data analysis*) są względnie nowym zagadnieniem w statystyce. Klasyczna statystyka skupia się niemal wyłącznie na *wnioskach*, czasami skomplikowanych zestawach procedur umożliwiających wyciąganie wniosków o ogromnych populacjach na podstawie niewielkich prób. W 1962 r. John W. Tukey (*https://en.wikipedia.org/wiki/John_Tukey*) (rysunek 1.1) w swoim przełomowym artykule *The Future of Data Analysis* (Przyszłość analizy danych) [Tukey 1962] wezwał do reformy statystyki. Zaproponował nową dyscyplinę, o nazwie **analiza danych** (ang. *data analysis*), której jednym z elementów byłoby wnioskowanie statystyczne (wprowadził terminy *bit*, skrót od cyfr binarnych, i *software*). Oryginalne tezy Tukeya są zaskakująco trwałe i tworzą częściową podstawę data science. Zakres działań badań eksploracyjnych został ustanowiony w klasycznym już dziele Tukeya *Exploratory Data Analysis* [Tukey 1977]. Tukey zaprezentował proste wykresy (np. pudełkowe czy punktowe), które wraz ze statystykami podsumowującymi (średnią, medianą, kwantylami itp.) pomagają zobrazować własności zestawu danych.



Rysunek 1.1. John Tukey, wybitny naukowiec, którego idee, rozwinięte ponad 50 lat temu, utworzyły podstawy data science

Wraz z ułatwieniem dostępu do mocy obliczeniowej i oprogramowania dedykowanego analizie danych badania eksploracyjne wyszły poza ich oryginalne założenia. Głównym motorem napędowym były gwałtowny rozwój nowych technologii, łatwy dostęp do większej liczby dużych zbiorów danych i zwiększenie możliwości wykorzystania analizy ilościowej w różnych dyscyplinach naukowych. David Donoho, profesor statystyki na Uniwersytecie Stanforda i były student Tukeya, jest autorem świetnego artykułu bazującego na jego prezentacji z warsztatów zorganizowanych z okazji setnej rocznicy urodzin Tukeya, które odbyły się na Uniwersytecie Princeton w New Jersey [Donoho 2015]. Donoho prześledził w swojej pracy początki data science, aż do pionierskiej pracy Tukeya dotyczącej analizy danych.

Elementy danych uporządkowanych

Dane mogą pochodzić z różnych źródeł; mogą to być pomiary z czujników, wydarzeń, tekstu, obrazów, filmów. **Internet rzeczy** (**IoT**, ang. *Internet of Things*) wyrzuca potoki informacji. Większość tych danych jest nieuporządkowana: obrazy są zbiorem pikseli, a każdy piksel zawiera informacje o kanałach RGB (ang. *R(ed)* — czerwony, *G(reen)* — zielony, *B(lue)* — niebieski). Tekst jest sekwencją słów i znaków, zazwyczaj uporządkowanych w rozdziały, podrozdziały itd. Z kolei *clickstreams* są sekwencjami przechodzenia przez kolejne elementy aplikacji lub podstrony strony internetowej. Tak naprawdę największym wyzwaniem data science jest przekształcenie wciąż spływającego natłoku danych w użyteczne informacje. Aby zagadnienia statystyczne poruszane w tej książce mogły zostać zastosowane w praktyce, nieuporządkowane dane muszą być przekształcone na uporządkowane. Jedną z najczęściej występujących form danych uporządkowanych jest tabela zawierająca rzędy i kolumny — tak jakby dane pochodziły z relacyjnej bazy danych lub były zbierane w celach badawczych.

Istnieją dwa podstawowe typy danych uporządkowanych: numeryczne i skategoryzowane. Dane numeryczne występują w dwóch formach: *ciągłej*, np. prędkość wiatru, czas trwania, lub *dyskretnej*, np. częstość występowania jakiegoś zjawiska. **Dane skategoryzowane** (ang. *categorical data*) przyjmują jedynie określone zbiory wartości, np. typy ekranów telewizorów (plazmowy, LCD, LED itp.) lub nazwy stanów (Alabama, Alaska itd.). **Dane binarne** (ang. *binary data*) są ważnym, szczególnym przypadkiem danych skategoryzowanych, w którym występują jedynie dwie wartości, tj. 0/1 lub prawda/fałsz. Innym użytecznym rodzajem danych skategoryzowanych są **dane porządkowe** (ang. *ordinal data*), w których kategorie są uporządkowane; przykładem jest ocenianie na podstawie ocen (1, 2, 3, 4 lub 5).

Dlaczego w ogóle przejmujemy się nazewnictwem typów danych? Okazuje się, że w przypadku analizy danych i modelowania predykcyjnego typy danych są istotne dla określenia sposobu ich wizualizacji, analizy i doboru modelu statystycznego. W świecie data science oprogramowanie, jak R lub Python, wykorzystuje te typy do poprawy wydajności obliczeń. Co ważniejsze, określenie typu danych dla zmiennej warunkuje sposób, w jaki oprogramowanie będzie przetwarzać tę zmienną.

Kluczowe pojęcia dotyczące typów danych

Numeryczne

Dane wyrażane w skali numerycznej.

Ciągłe

Dane, które mogą przyjmować dowolną wartość z przedziału (*Synonimy*: przedziałowe, numeryczne, float)

Dyskretne

Dane, które mogą przyjmować jedynie wartości całkowite, np. liczność (*Synonimy*: zliczenia, integer)

Skategoryzowane

Dane, które mogą przyjmować jedynie wartości z konkretnego zbioru, reprezentującego możliwe kategorie (*Synonimy*: czynniki, enum, nominalne, polichotomiczne)

Binarne

Szczególny przypadek danych skategoryzowanych z dwoma wartościami np. 0/1, prawda/fałsz (*Synonimy*: dychotomiczne, logiczne, wskaźnikowe, boolean)

Porządkowe

Dane skategoryzowane według wprowadzonego porządku (*Synonim*: czynniki uporządkowane)

Inżynierowie oprogramowania i programiści baz danych mogą się zastanawiać, do czego analitykom potrzebne są pojęcia danych *skategoryzowanych* i *porządkowych*. Bądź co bądź kategorie są w zasadzie zbiorem wartości tekstowych (lub numerycznych), a podstawowe bazy danych automatycznie przenoszą wewnętrzne typy. Jednakże świadomość różnic pomiędzy danymi skategoryzowanymi a zwykłym tekstem daje pewne korzyści:

- Dzięki wiedzy o tym, że dane są skategoryzowane, można sugerować oprogramowaniu, jak powinny się zachować procedury statystyczne, np. tworząc wykres lub dopasowując model. W szczególności dane porządkowe mogą być reprezentowane jako `ordered.factor` w R i zachowywać specyficzny dla użytkownika porządek w wykresach, tabelach i modelach. W Pythonie moduł `scikit-learn` obsługuje dane porządkowe za pomocą klasy `sklearn.preprocessing.OrdinalEncoder`).
- Przechowywanie i indeksowanie może być zoptymalizowane (jak w relacyjnych bazach danych).
- Wartości, jakie może przyjmować zmienna kategoryzowana, są zdefiniowane w oprogramowaniu (np. enum).

Trzecia z wymienionych korzyści może prowadzić do niezamierzonych lub niespodziewanych zachowań: domyślnym zachowaniem funkcji importowania danych w R (np. `read.csv`) jest automatyczna konwersja kolumn zawierających tekst do typu `factor`. Sprawia to, że jedynymi dopuszczalnymi wartościami dla tych kolumn będą te oryginalnie występujące w danych, a próba przypisania innych wartości tekstowych spowoduje pojawienie się ostrzeżenia i wartości `NA` (brakującej wartości). W Pythonie pakiet `pandas` nie realizuje automatycznie takiego przekształcenia. Można jednak jawnie wyznaczyć kolumnę skategoryzowaną wewnątrz funkcji `read_csv`.

Główne zasady

- Dane są zazwyczaj klasyfikowane przez oprogramowanie ze względu na typ.
- Typy danych można podzielić na numeryczne (ciągłe, dyskretne) i skategoryzowane (binarne, porządkowe).
- Określenie typu danych w oprogramowaniu jest sygnałem dla niego, w jaki sposób dane mają być przetwarzane.

Dla pogłębienia wiedzy

- Dokumentacja pakietu `pandas` (*https://pandas.pydata.org/docs/user_guide/dsintro.html#dsintro*) opisuje różne typy danych i sposób ich wykorzystywania w Pythonie.
- Typy danych nie są jednoznaczne, ich definicje mogą się dublować, a nazewnictwo w jednym oprogramowaniu może się różnić od tego użytego w innym. Strona objaśniająca typy użyte w R to: *http://www.r-tutor.com/r-introduction/basic-data-types*.
- Bazy danych są bardziej dokładne w klasyfikacji typów danych, biorą pod uwagę poziom dokładności, stałą lub zmienną długość pola i wiele innych czynników; patrz podręcznik W3Schools dla SQL-a: *https://www.w3schools.com/sql/sql_datatypes.asp*.

Dane stabelaryzowane

Typowym przykładem analizy w data science są **dane stabelaryzowane** (ang. *rectangular data*), takie jak arkusz kalkulacyjny lub tabela bazy danych.

Dane stabelaryzowane to w gruncie rzeczy dwuwymiarowa macierz, w której wiersze reprezentują rekordy (przypadki), a kolumny określają cechy (zmienne); w R i Pythonie występuje szczególny format, zwany **ramką danych** (ang. *data frame*). Dane nie zawsze od razu wstępują w tej formie: dane nieuporządkowane (np. tekst) muszą być przetworzone w taki sposób, by reprezentowały zestaw cech danych stabelaryzowanych (patrz podrozdział „Elementy danych uporządkowanych" we wcześniejszej części tego rozdziału). Do zastosowań w większości typów analiz i modelowania dane z relacyjnych baz danych muszą przyjąć formę jednej tabeli.

Kluczowe pojęcia dotyczące typów danych

Ramka danych

Dane stabelaryzowane (jak arkusz kalkulacyjny) są podstawową strukturą w statystyce i modelach uczenia maszynowego.

Cecha

Kolumna w tabeli jest najczęściej nazywana *cechą*.

Synonimy

atrybut, wejście, predyktor, zmienna

Wynik

Wiele projektów w data science służy do przewidzenia *wyniku* — zazwyczaj w postaci tak/nie (w tabeli 1.1 mamy np. „aukcja była konkurencyjna lub nie"). *Cechy* są czasem wykorzystywane do przewidzenia *wyniku* w badaniach eksperymentalnych.

Synonimy

zmienna zależna, odpowiedź, cel, wyjście

Rekord

Wiersz w tabeli jest najczęściej nazywany *rekordem*.

Synonimy

przypadek, przykład, instancja, obserwacja, wzorzec, próbka

Tabela.1.1. Typowy przykład ramki danych

Kategoria	Waluta	Wskaźnik sprzedaży	Czas trwania	Dzień zakończenia	Cena zamknięcia	Cena otwarcia	Konkurencyjny?
Muzyka/film/gra	USD	3249	5	Pon.	0,01	0,01	0
Muzyka/film/gra	USD	3249	5	Pon.	0,01	0,01	0
Automotive	USD	3115	7	Wt.	0,01	0,01	0
Automotive	USD	3115	7	Wt.	0,01	0,01	0
Automotive	USD	3115	7	Wt.	0,01	0,01	0
Automotive	USD	3115	7	Wt.	0,01	0,01	0
Automotive	USD	3115	7	Wt.	0,01	0,01	1
Automotive	USD	3115	7	Wt.	0,01	0,01	1

W tabeli 1.1 zamieszczono zestaw danych pomiarowych lub częstości (np. czas trwania i cena) oraz danych skategoryzowanych (np. kategoria, waluta). Jak już wspomniano wcześniej, szczególną formą zmiennej skategoryzowanej jest zmienna binarna (tak/nie, 0/1), przedstawiona w kolumnie po prawej stronie tabeli 1.1 — jest to zmienna wskaźnikowa pokazująca, czy aukcja była konkurencyjna (brało w niej udział wielu uczestników), czy nie. Ta zmienna wskaźnikowa okazuje się być także **zmienną wynikową** (ang. *outcome variable*) w przypadku, gdy zadaniem modelu jest przewidywanie, czy aukcja była konkurencyjna.

Ramki danych i indeksy

Tradycyjne tabele baz danych mają jedną lub więcej kolumn zaprojektowanych jako indeks, będący zasadniczo numerem rzędu. Może to znacznie poprawić wydajność niektórych zapytań bazodanowych. W języku Python z biblioteką `pandas` podstawową formą danych stabelaryzowanych jest obiekt typu `DataFrame`. Dla danych typu `DataFrame` tworzony jest automatycznie domyślny indeks typu `integer`, bazujący na kolejności wierszy. W `pandas` możliwa jest również poprawa wydajności niektórych operacji poprzez ustawienie indeksu wielopoziomowego (hierarchicznego).

W języku R podstawowym typem danych stabelaryzowanych jest obiekt typu `data.frame`. Obiekty typu `data.frame` mają również wprowadzony indeks typu `integer`, bazujący na kolejności wierszy. Podstawowy typ `data.frame` nie wspiera indeksów zdefiniowanych przez użytkownika lub indeksów wielopoziomowych, jednak istnieje możliwość stworzenia własnego klucza poprzez atrybut `row.names`. Do pozbycia się tej niedogodności szeroko stosowane są dwa nowe pakiety: `data.table` i `dplyr`. Oba wspierają indeksowanie wielopoziomowe i umożliwiają znaczne przyspieszenie pracy z danymi typu `data.frame`.

Różnice w terminologii

Terminologia związana z danymi stabelaryzowanymi może być myląca. Statystycy i specjaliści data science używają różnych nazw dla tych samych zagadnień. Dla statystyka *zmienna predykcyjna* jest wykorzystywana w modelu do określenia *odpowiedzi* albo *zmiennej zależnej*. Dla specjalisty data science *cechy* są wykorzystywane do określenia *celu*. Szczególnie mylące jest pojęcie *próby*: specjaliści nauk komputerowych użyją go do określenia pojedynczego wiersza, a statystycy do określenia zbioru wierszy.

Niestabelaryzowane struktury danych

Oprócz danych stabelaryzowanych istnieją jeszcze inne struktury danych, opisane poniżej.

Serie czasowe zawierają dane pomiarowe dla jednej zmiennej. Stanowi to podstawowy materiał dla statystycznych metod prognostycznych, jest również kluczowym komponentem danych tworzonych przez sprzęt — Internet rzeczy.

Przestrzenne dane strukturalne, które są wykorzystywane do mapowania i lokalizacji, są bardziej złożone i zróżnicowane niż dane stabelaryzowane. W reprezentacji obiektowej głównymi elementami danych są **obiekt** (ang. *object*) (np. dom) i jego współrzędne przestrzenne. Widok typu **field** (ang. *field*) dla odmiany skupia się na niewielkich jednostkach przestrzeni i wartości odpowiednich wskaźników (np. poziomie jasności pikseli).

Grafowe (lub sieciowe) struktury danych są wykorzystywane do reprezentacji zależności fizycznych, socjologicznych i abstrakcyjnych. Przykładowo graf sieci społecznej, takiej jak Facebook czy LinkedIn, może reprezentować powiązania pomiędzy osobami w sieci. Centra dystrybucji połączone drogami mogą być przykładem sieci fizycznej. Struktura grafu jest użyteczna w przypadku szczególnego typu zadań, jak optymalizacja sieci i zalecenia systemowe.

Każdy z tych typów danych ma swoją dedykowaną metodologię w data science. W tej książce skupiono się na danych stabelaryzowanych, stanowiących fundament modelowania predykcyjnego.

Grafy w statystyce

W naukach komputerowych i technologii informacyjnej pojęcie *grafu* odnosi się typowo do przedstawienia powiązań pomiędzy bytami albo podstawowymi strukturami danych. W statystyce pojęcie *grafu* jest używane w odniesieniu do grupy wykresów i *wizualizacji*, a nie tylko do połączeń pomiędzy bytami. Samo pojęcie *grafu* odnosi się jedynie do wizualizacji, a nie do struktury danych.

Główne zasady

- Podstawowe struktury danych w data science są macierzami, w których wiersze są rekordami, a kolumny zmiennymi (cechami).
- Terminologia może być myląca; istnieje wiele synonimów, które przeszły do data science z różnych dyscyplin (ze statystyki, z nauki o komputerach i technologii informacyjnej).

Dla pogłębienia wiedzy

- Dokumentacja dotycząca data frame w R: *https://stat.ethz.ch/R-manual/R-devel/library/base/html/data.frame.html*.
- Dokumentacja dotycząca data frame w Pythonie: *https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html*.

Miary położenia

Na zmienne zawierające dane pomiarowe lub liczności mogą się składać setki wartości. Podstawowym zadaniem w badaniach eksploracyjnych jest określenie „typowych wartości" dla każdej z cech (zmiennej): oszacowanie, gdzie zlokalizowana jest większość wartości analizowanych danych (tj. tendencji centralnej).

Kluczowe pojęcia dotyczące miar położenia

Średnia

Suma wszystkich wartości podzielona przez liczbę tych wartości.

Średnia ważona

Suma wszystkich wartości wymnożonych przez wagi, podzielona przez sumę tych wag.

Mediana

Wartość, dla której jedna połowa danych jest od niej większa, a druga połowa od niej mniejsza.

Synonim

50. percentyl

Percentyl

Wartość określająca odsetek P danych, które są od niej mniejsze.

Synonim

kwantyl

Mediana ważona

Wartość, dla której połowa sumy wag należy do próbek większych niż ta wartość, a druga połowa sumy wag należy do próbek mniejszych niż ta wartość.

Średnia ucinana

Średnia ze zbioru powstałego po odrzuceniu stałej liczby wartości skrajnych.

Synonim

Średnia trymowana, średnia obcięta

Odporność

Brak wrażliwości na wartości skrajne.

Wartość odstająca

Wartość, która znacznie się różni od większości wartości występujących w danych.

Synonim

wartość ekstremalna

Na pierwszy rzut oka podsumowanie danych wydaje się trywialne: trzeba po prostu policzyć **średnią** (ang. *mean*) z danych. Tak naprawdę średnia wydaje się najłatwiejsza do obliczenia i wygodna w użyciu, jednakże nie zawsze jest najlepszym określeniem miary położenia. Z tego powodu statystycy stworzyli i rozpropagowali kilka wskaźników alternatywnych.

Miary i estymatory

Statystycy często stosują pojęcie **estymatora/oszacowania** (ang. *estimates*) w odniesieniu do wartości obliczanych z dostępnych danych, by zaznaczyć różnicę pomiędzy tym, co sugerują dane, a teoretycznie właściwym stanem rzeczy. Specjaliści data science i analitycy biznesowi najczęściej używają do tego pojęcia **miary** (ang. *metric*). Różnice w nazewnictwie odzwierciedlają różnice w podejściu statystyki i podejściu data science: obliczanie niepewności jest sercem statystyki, podczas gdy data science koncentruje się na biznesie i celach organizacyjnych. Dlatego właśnie statystycy estymują, a specjaliści data science mierzą.

Średnia

Najbardziej podstawowym estymatorem położenia jest średnia. Średnia jest sumą wszystkich wartości podzieloną przez liczbę tych wartości. Rozważmy następujący zbiór wartości: {3 5 1 2}. Średnia z nich to (3+5+1+2)/4 = 11/4 = 2,75. Możesz się spotkać z symbolem $\bar{x}$, reprezentującym

średnią z próby badanej populacji. Wzór na obliczenie średniej ze zbioru n wartości x_1, x_2, …, x_n jest wyrażony jako:

$$Średnia = \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

N (lub *n*) odnosi się do całkowitej liczby przypadków lub obserwacji. W statystyce *N* odnosi się do populacji, a *n* do próby z populacji. W data science to rozróżnienie jest mniej istotne, dlatego możesz się spotkać z obiema wersjami.

Wariacją średniej jest **średnia ucinana** (ang. *trimmed mean*), którą obliczysz, odrzucając stałą liczbę wartości z każdego końca danych posortowanych, a następnie obliczając średnią z tego, co pozostało. Przyjmując posortowane wartości jako $x_{(1)}$, $x_{(2)}$, …, $x_{(n)}$, gdzie $x_{(1)}$ jest najmniejszą wartością, a $x_{(n)}$ największą, oraz p jako liczbę najmniejszych i największych pomijanych wartości, wzór na obliczenie średniej ucinanej można przedstawić jako:

$$Średnia\ ucinana = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n - 2p}$$

Średnia ucinana eliminuje wpływ wartości skrajnych. Przykładowo w międzynarodowych zawodach w skokach do wody najwyższa i najniższa z ocen pięciu sędziów jest usuwana, a ocena końcowa jest średnią z pozostałych trzech (*https://en.wikipedia.org/wiki/Diving_(sport)#Scoring_the_dive*). Uniemożliwia to pojedynczemu sędziemu manipulację wynikami, np. na korzyść reprezentanta własnego kraju. Średnia ucinana jest szeroko wykorzystywana, w wielu przypadkach chętniej niż klasyczna średnia (patrz punkt „Mediana i estymatory odporne" w dalszej części tego rozdziału).

Innym typem średniej jest **średnia ważona** (ang. *weighted mean*), którą obliczysz, mnożąc każdą wartość w danych x_i przez odpowiednią wagę w_i i dzieląc sumę tych iloczynów przez sumę wag. Wzór na średnią ważoną jest wyrażony jako:

$$Średnia\ ważona = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i\, x_i}{\sum_{i=1}^{n} w_i}$$

Można wyróżnić dwa główne powody użycia średniej ważonej:

- Niektóre wartości są z natury bardziej zmienne niż inne, w takim przypadku wysoce zmienne obserwacje otrzymują mniejsze wagi. Przykładowo jeśli liczymy średnie dla wielu czujników i jeden z nich jest mniej dokładny, możemy obniżyć wpływ danych z tego czujnika na wynik końcowy.
- Zbierane dane nie reprezentują równomiernie różnych grup, które chcemy mierzyć. Przykładowo poprzez sposób, w jaki przeprowadzono eksperyment w sieci, możemy nie uzyskać zbioru danych odzwierciedlającego wszystkie grupy użytkowników w bazie. Aby to skorygować, większe wagi przypisujemy wartościom z grup, które są niedoreprezentowane.

Mediana i estymatory odporne

Mediana (ang. *median*) jest wartością środkową na posortowanej liście danych. Jeśli danych jest parzysta liczba, wartość środkowa nie jest zawarta w zbiorze danych, ale jest średnią z dwóch wartości, które dzielą posortowany zbiór na dolną i górną połowę. W porównaniu do średniej, która wykorzystuje wszystkie obserwacje, mediana zależy od wartości w centrum zbioru danych. Może się to wydawać wadą, jednak średnia jest zdecydowanie bardziej wrażliwa i dla wielu przypadków mediana okazuje się zdecydowanie lepszą miarą położenia. Załóżmy, że chcemy sprawdzić typowe przychody gospodarstw domowych zlokalizowanych wokół jeziora Waszyngton w Seattle. Wykorzystując średnią do porównania obszarów Mediny i Windermere, uzyskamy różne wyniki, ponieważ Bill Gates mieszka w Medinie. Jeśli użyjemy mediany, nie będzie miało znaczenia, jak bogaty jest Bill Gates — wartość środkowa pozostanie taka sama.

Z tego samego powodu zamiast średniej ważonej można wykorzystać **medianę ważoną** (ang. *weighted median*). Jak w przypadku mediany, sortujemy dane, tym razem jednak każda wartość ma przypisaną wagę. Zamiast szukać wartości środkowej, mediana ważona dzieli posortowany ciąg danych na dwie części, z których każda ma taką samą sumę wag. Podobnie jak mediana, mediana ważona jest odporna na wartości odstające.

Wartości odstające

Mediana jest nazywana **odpornym** (ang. *robust*) estymatorem położenia; określenie „odporny" wzięło się od tego, że na ten estymator nie mają wpływu **wartości odstające** (przypadki skrajne) (ang. *outliers*, *extreme cases*), które mogą zniekształcić wynik. Wartość odstająca jest bardzo odległa od innych w zbiorze danych. Dokładna definicja wartości odstającej jest w pewien sposób subiektywna, jednakże istnieją pewne konwencje, wykorzystywane w zestawieniach i wykresach danych (patrz punkt „Percentyle i boxploty" w dalszej części tego rozdziału). To, że wartość jest odstająca, nie oznacza od razu, że jest nieistotna lub błędna (jak w przykładzie z Billem Gatesem). Często wartość odstająca jest jednak wynikiem błędów, np. łączenia danych o różnych jednostkach (metrów i kilometrów) lub niepoprawnego odczytu z czujnika. Kiedy wartość odstająca jest wynikiem zastosowania błędnych danych, średnia będzie kiepskim estymatorem położenia, podczas gdy mediana wciąż będzie wiarygodna. W każdym przypadku wartości odstające powinny być zidentyfikowane i najczęściej warto poddać je dalszej analizie.

Detekcja anomalii

W przeciwieństwie do typowej analizy danych, w której wartości odstające czasami stanowią użyteczną informację, a czasami utrapienie, **detekcja anomalii** dotyczy tylko wartości odstających; znaczna większość danych służy w tym przypadku głównie do zdefiniowania „normalności", wobec której anomalie są określane.

Mediana nie jest jedynym odpornym estymatorem położenia. W rzeczywistości średnia ucinana jest szeroko stosowana do unikania wpływu wartości odstających. Przykładowo odcięcie dolnych i górnych 10% danych (typowy wybór) zapewni ochronę dla wszystkich przypadków poza niewielkimi zbiorami danych. Średnia ucinana może być traktowana jako kompromis pomiędzy medianą a średnią: jest odporna na wartości skrajne, ale wykorzystuje więcej danych do oszacowania położenia.

Inne odporne miary położenia

Statystycy rozwinęli wiele innych estymatorów położenia. Głównym celem było stworzenie estymatora bardziej odpornego niż średnia, ale również bardziej *czułego*, tj. lepiej wychwytującego niewielkie różnice położenia pomiędzy zbiorami danych. Metody te są potencjalnie użyteczne dla niewielkich zbiorów danych, jednakże nie dostarczają wartości dodanej dla dużych lub nawet umiarkowanie dużych zbiorów danych.

Przykład: miara położenia dla wielkości populacji i wskaźnika morderstw

Tabela 1.2 przedstawia kilka pierwszych kolumn zbioru danych zawierającego wartości liczebności populacji i wskaźnika morderstw (w liczbie morderstw na 100 000 mieszkańców na rok) dla każdego stanu USA (spis ludności z 2010 r.).

Tabela 1.2. Kilka kolumn z data.frame zawierającego liczebność populacji i wskaźnik morderstw dla poszczególnych stanów

	State	Population	Murder.Rate	Abbreviation
1	Alabama	4 779 736	5,7	AL.
2	Alaska	710 231	5,6	AK
3	Arizona	6 392 017	4,7	AZ
4	Arkansas	2 915 918	5,6	AR
5	California	37 253 956	4,4	CA
6	Colorado	5 029 196	2,8	CO
7	Connecticut	3 574 097	2,4	CT
8	Delaware	897 934	5,8	DE

Policz średnią, ucinaną średnią i medianę dla populacji, wykorzystując R.

```
> state <- read.csv('state.csv')
> mean(state[['Population']])
[1] 6162876
> mean(state[['Population']], trim=0.1)
[1] 4783697
> median(state[['Population']])
[1] 4436370
```

Do obliczenia średniej i mediany w Pythonie możemy użyć metod ramki danych z pakietu `pandas`. Średnią ucinaną obliczymy za pomocą funkcji `trim_mean` z modułu `scipy.stats`:

```
state = pd.read_csv('state.csv')
state['Population'].mean()
trim_mean(state['Population'], 0.1)
state['Population'].median()
```

Średnia jest większa niż średnia ucinana, która z kolei jest większa od mediany.

Dzieje się tak dlatego, że średnia ucinana wyklucza pięć największych i pięć najmniejszych stanów (`trim=0.1` wyłącza 10% z każdej strony). Jeśli chcemy obliczyć średni wskaźnik morderstw w kraju, musimy użyć średniej ważonej lub mediany ważonej z uwagi na różną liczność populacji w poszczególnych stanach. Ponieważ podstawowy R nie ma funkcji liczącej medianę ważoną, musimy doinstalować pakiet, np. `matrixStats`:

```
> weighted.mean(state[['Murder.Rate']], w=state[['Population']])
[1] 4.445834
> library('matrixStats')
> weightedMedian(state[['Murder.Rate']], w=state[['Population']])
[1] 4.4
```

Średnią ważoną możemy obliczyć za pomocą pakietu NumPy. Do obliczenia mediany ważonej możemy posłużyć się wyspecjalizowanym pakietem wquantiles (*https://pypi.org/project/wquantiles/*):

```
np.average(state['Murder.Rate'], weights=state['Population'])
wquantiles.median(state['Murder.Rate'], weights=state['Population'])
```

W tym przypadku średnia ważona i mediana ważona są niemal identyczne.

Główne zasady

- Średnia jest podstawową miarą położenia, może być jednak wrażliwa na wartości skrajne (odstające).
- Inne miary (mediana, średnia ucinana) są mniej wrażliwe na wartości odstające oraz nietypowe rozkłady, a zatem wykazują się większą odpornością.

Dla pogłębienia wiedzy

- Artykuł Wikipedii poświęcony tendencji centralnej (*https://en.wikipedia.org/wiki/Central_tendency*) opisuje dokładnie różne miary położenia.
- Wciąż jest też czytana książka *Exploratory Data Analysis* (wyd. Pearson) Johna Tukeya z 1977 r.

Miary rozproszenia

Położenie jest tylko jednym z czynników służących do opisania zmiennej. Drugim jest **zmienność** (ang. *variability*), określana również jako **rozproszenie** (ang. *dispersion*), sprawdzająca, czy dane są ciasno zgrupowane, czy rozproszone. W samym sercu statystyki leży rozproszenie: jego pomiar, redukcja, rozróżnienie zmienności losowej i faktycznej, identyfikacja różnych źródeł faktycznej zmienności i podejmowanie decyzji po jej identyfikacji.

Kluczowe pojęcia dotyczące miar rozproszenia

Odchylenie

Różnica pomiędzy wartością obserwowaną a szacowanym położeniem.

Synonimy

błąd, rezyduum, reszta

Wariancja

Suma kwadratów odchylenia od średniej podzielonych przez $n-1$, gdzie n jest liczbą wartości w danych.

Synonim

błąd średniokwadratowy

Odchylenie standardowe

Pierwiastek kwadratowy z wariancji.

Średnie odchylenie bezwzględne

Średnia z wartości bezwzględnych odchyleń zbioru danych od średniej.

Synonimy

norma l1, norma Manhattan

Mediana odchylenia bezwzględnego od mediany

Mediana z wartości bezwzględnych odchyleń zbioru danych od mediany.

Zakres

Różnica pomiędzy najmniejszą a największą wartością w zbiorze danych.

Statystyka porządkowa

Miary oparte na wartościach dla danych posortowanych rosnąco.

Synonim

ranking

Percentyl

Taka wartość, że P procent wartości jest takie samo jak ona bądź mniejsze od niej i (100-P) procent jest takie samo jak ona bądź większe od niej.

Synonim

kwantyl

Przedział międzykwartylowy

Różnica pomiędzy 75. a 25. percentylem.

Synonim

IQR (ang. *interquartile range*)

Jako że istnieje wiele sposobów pomiaru położenia (średnia, mediana itd.), istnieje również wiele sposobów na zmierzenie rozproszenia.

Odchylenie standardowe i powiązane estymatory

Najczęściej stosowane oszacowanie zmienności jest oparte na różnicach lub **odchyleniach** (ang. *deviations*) pomiędzy oszacowaniem położenia a obserwowanymi danymi. Dla zbioru danych {1, 4, 1} średnia wynosi 3, a mediana 4. Odchylenia od średniej są następującymi różnicami: 1–3 = –2, 4–3 = 1, 4–3 = 1. Te różnice mówią nam o tym, jak bardzo dane rozrzucone są wokół wartości centralnej.

Jednym ze sposobów na zmierzenie zmienności jest oszacowanie wartości typowej. Uśrednianie samych odchyleń nie powie zbyt wiele — odchylenia ujemne będą znosić te dodatnie. W rzeczywistości suma odchyleń od średniej dla powyższego przykładu wynosi zero. Zamiast tego w prosty sposób możemy policzyć średnią z wartości bezwzględnych odchyleń od średniej. Wartości bezwzględne odchyleń wynosić będą {2, 1, 1}, a ich średnia wynosi (2+1+1)/3 = 1,33. Obliczona wartość opisuje **średnie odchylenie bezwzględne** (ang. *mean absolute deviation*), które w ogólnym przypadku obliczane jest z następującego wzoru:

$$Średnie\ odchylenie\ bezwzględne = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

gdzie $\bar{x}$ jest średnią z próby.

Najbardziej znanymi estymatorami zmienności są **wariancja** (ang. *variance*) i **odchylenie standardowe** (ang. *standard deviation*), które są oparte na kwadratach odchyleń. Wariancja jest średnią z kwadratów odchyleń, a odchylenie standardowe jest pierwiastkiem kwadratowym z wariancji.

$$Wariancja = s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

$$Odchylenie\ standardowe = s = \sqrt{Wariancja}$$

Odchylenie standardowe jest znacznie prostsze w interpretacji niż wariancja, ponieważ jest przedstawione w tej samej skali co dane oryginalne. W pierwszej chwili zdziwienie budzi fakt, że mimo iż wzór na odchylenie standardowe jest bardziej skomplikowany i mniej intuicyjny niż wzór na średnie odchylenie bezwzględne, to jednak odchylenie standardowe jest częściej stosowane w statystyce niż średnie odchylenie bezwzględne. W statystyce dominuje pogląd, że z punktu widzenia pewnych niuansów matematycznych praca z kwadratami wartości jest zdecydowanie wygodniejsza niż praca z wartościami bezwzględnymi, zwłaszcza w modelach statystycznych.

Stopnie swobody i *n* czy *n*–1?

W podręcznikach do statystyki jakaś część zawsze poświęcona jest rozważaniom o tym, dlaczego w mianowniku wzoru na wariancję występuje *n*–1 zamiast *n*, co bezpośrednio prowadzi do omówienia pojęcia **stopni swobody** (ang. *degrees of freedom*). To rozróżnienie nie jest istotne w przypadku, gdy *n* jest wystarczająco duże, by różnica pomiędzy dzieleniem przez *n* a dzieleniem przez *n*–1 była odczuwalna. Gdybyś jednak był zainteresowany, zapoznaj się z wyjaśnieniem zamieszczonym poniżej. Załóżmy, że chcesz stworzyć estymatory dla populacji, bazując na próbie.

Jeśli użyjesz intuicyjnego mianownika *n* we wzorze na wariancję, nie doszacujesz prawdziwej wartości wariancji i odchylenia standardowego w populacji. Mamy tu do czynienia z **estymatorem obciążonym** (ang. *biased estimate*). Jeśli podzielisz przez *n*–1 zamiast przez *n*, wariancja stanie się **estymatorem nieobciążonym** (ang. *unbiased estimate*).

Żeby w pełni wyjaśnić, dlaczego użycie *n* prowadzi do obciążenia estymatora, należy wprowadzić pojęcie stopni swobody, które biorą pod uwagę liczbę ograniczeń podczas obliczania estymatora. W naszym przykładzie istnieje *n–1* stopni swobody, ponieważ istnieje jedno ograniczenie: odchylenie standardowe zależy od obliczonej średniej z próby. W większości przypadków specjaliści data science nie muszą się przejmować stopniami swobody.

Żadna z przytoczonych powyżej miar, tj. wariancja, odchylenie standardowe czy średnie odchylenie bezwzględne, nie jest odporna na wartości skrajne (patrz punkt „Mediana i estymatory odporne" we wcześniejszej części tego rozdziału). Wariancja i odchylenie standardowe są szczególnie wrażliwe na wartości odstające, ponieważ bazują na różnicy podniesionej do kwadratu.

Odpornym estymatorem rozproszenia jest **mediana odchylenia bezwzględnego od mediany** (**MAD**, ang. *median absolute deviation from the median*):

$$Mediana\ odchylenia\ bezwzględnego = Mediana(|x_1 - m|, |x_2 - m|, \ldots, |x_N - m|)$$

gdzie *m* jest medianą. Podobnie jak mediana, mediana odchylenia bezwzględnego nie jest zależna od wartości ekstremalnych. Możliwe jest również obliczenie ucinanego odchylenia standardowego w taki sposób, w jaki oblicza się średnią ucinaną (patrz punkt „Średnia" we wcześniejszej części tego rozdziału).

Wariancja, odchylenie standardowe, średnie odchylenie bezwzględne i mediana odchylenia bezwzględnego od mediany nie są równoważne, nawet w przypadku, gdy dane pochodzą z rozkładu normalnego. Odchylenie standardowe jest zawsze większe od średniego odchylenia bezwzględnego, które z kolei jest większe od mediany odchylenia bezwzględnego. W przypadku rozkładu normalnego mediana odchylenia bezwzględnego jest czasami mnożona przez stały współczynnik skalujący, żeby można było ją porównać z odchyleniem standardowym. Powszechnie stosowany współczynnik 1,4826 oznacza, że 50% rozkładu normalnego mieści się w zakresie ±MAD (zob. np. *https://en.wikipedia.org/wiki/Median_absolute_deviation#Relation_to_standard_deviation*).

Estymatory oparte na percentylach

Kolejnym podejściem do szacowania rozproszenia jest przyjrzenie się rozrzutowi danych posortowanych. Statystyki bazujące na posortowanych danych są określane mianem **statystyk uporządkowanych** (ang. *order statistics*). Najbardziej podstawową z nich jest **zakres** (ang. *range*), czyli różnica pomiędzy największą a najmniejszą wartością. Znajomość wartości minimalnej i wartości maksymalnej jest przydatna i pomaga w określeniu wartości odstających, jednak jest bardzo wrażliwa na te wartości i niezbyt pomocna w ogólnym określaniu rozrzutu danych.

By uniknąć wrażliwości na wartości odstające, możemy popatrzeć na zakres danych po odrzuceniu wartości z obu końców. Formalnie taki typ estymatora oparty jest na różnicach pomiędzy **percentylami** (ang. *percentiles*). W zbiorze danych *P-ty* percentyl jest taką wartością, że co najmniej *P* procent wartości jest takie samo jak ona bądź od niej mniejsze, a co najmniej *(100–P)* procent wartości jest takie samo jak ona lub od niej większe. Przykładowo żeby znaleźć 80. percentyl, najpierw powinieneś posortować dane. Następnie, począwszy od wartości najmniejszej, przejdź po kolei 80 procent wartości, aż do największej z nich. Zauważ, że mediana jest niczym innym jak 50. percentylem. Percentyl jest w zasadzie tym samym co **kwantyl** (ang. *quantile*), jednakże kwantyle określane są przez ułamki (stąd kwantyl rzędu 0,8 będzie tym samym co 80. percentyl).

Popularną miarą rozproszenia jest różnica pomiędzy 25. percentylem i 75. percentylem, nazywana **przedziałem międzykwartylowym** (ang. *interquartile range*). Prosty przykład to: {3, 1, 5, 3, 6, 7, 2, 9}. Po posortowaniu otrzymujemy: {1, 2, 3, 3, 5, 6, 7, 9}. Ponieważ 25. percentyl to 2,5, a 75. to 6,5, przedział międzykwartylowy wynosi 6,5–2,5 = 4. Oprogramowanie może zaproponować inne podejście, stąd też inne odpowiedzi (patrz kolejna wskazówka); zazwyczaj te różnice są mniejsze.

Dla bardzo dużych zbiorów danych wyliczenie dokładnej wartości percentyla może być obliczeniowo bardzo wymagające, ponieważ zmusza do posortowania wszystkich danych. Metody uczenia maszynowego i oprogramowanie statystyczne wykorzystują specjalne algorytmy [Zhang Wang 2007], żeby oszacować percentyle. Dzięki temu mogą być one obliczane bardzo szybko, z gwarancją zadanej dokładności.

Percentyle: dokładna definicja

Jeśli mamy parzystą liczbę danych (*n* jest parzyste), to — zgodnie z poprzednią definicją — percentyl nie jest jednoznacznie określony. W rzeczywistości może on być dowolną wartością pomiędzy uporządkowanymi statystykami $x_{(j)}$ i $x_{(j+1)}$, gdzie *j* spełnia równanie:

$$100 \cdot \frac{j}{n} \leq P \leq 100 \cdot \frac{j+1}{n}$$

Formalnie percentyl jest średnią ważoną:

$$Percentyl(P) = (1-w)x_{(j)} + wx_{(j+1)}$$

dla wag *w* z przedziału 0 i 1. Oprogramowanie statystyczne wykorzystuje nieco inne podejście przy wyborze *w*. Funkcja w R `quantile` podaje dziewięć różnych sposobów na wyliczenie kwantylu. Z wyjątkiem małych zbiorów danych, nie musisz się przejmować sposobem oszacowania percentyli. W czasie pisania niniejszego wydania klasa `numpy.quantile` ze środowiska Python obsługuje tylko jedno rozwiązanie: interpolację liniową.

Przykład: szacowanie zmienności dla populacji Stanów Zjednoczonych

Tabela 1.3 (powtórzenie tabeli 1.2 zamieszczono dla ułatwienia) ukazuje kilka początkowych wierszy ze zbioru danych zawierającego liczebność populacji i współczynnik morderstw dla każdego stanu.

Tabela 1.3. Kilka kolumn z data.frame zawierającego liczebność populacji i wskaźniki morderstw dla poszczególnych stanów

	State	Population	Murder.Rate	Abbreviation
1	Alabama	4 779 736	5,7	AL.
2	Alaska	710 231	5,6	AK
3	Arizona	6 392 017	4,7	AZ
4	Arkansas	2 915 918	5,6	AR
5	California	37 253 956	4,4	CA
6	Colorado	5 029 196	2,8	CO
7	Connecticut	3 574 097	2,4	CT
8	Delaware	897 934	5,8	DE

Wykorzystując wbudowaną funkcję R dla odchylenia standardowego, IQR i mediany odchylenia bezwzględnego dla mediany, możemy obliczyć estymatory rozproszenia dla populacji poszczególnych stanów:

```
> sd(state[['Population']])
[1] 6848235
> IQR(state[['Population']])
[1] 4847308
> mad(state[['Population']])
[1] 3849870
```

Ramka danych `pandas` zawiera metody umożliwiające obliczanie odchylenia standardowego i kwantyli. Dzięki kwantylom możemy z łatwością określić IQR. W przypadku odpornej mediany odchylenia bezwzględnego dla mediany używamy funkcji `robust.scale.mad` z pakietu `statsmodel`:

```
state['Population'].std()
state['Population'].quantile(0.75) - state['Population'].quantile(0.25)
robust.scale.mad(state['Population'])
```

Odchylenie standardowe jest niemal dwukrotnie większe od mediany odchylenia bezwzględnego (w R wyniki są domyślnie skalowane, tak by mediana odchylenia bezwzględnego mogła być porównana ze średnią). Nie jest to zaskoczeniem, zważywszy na to, że odchylenie standardowe jest wrażliwe na wartości odstające.

Główne zasady

- Wariancja i odchylenie standardowe są najpowszechniej wykorzystywanymi statystykami rozproszenia.
- I wariancja, i odchylenie standardowe są wrażliwe na wartości odstające.
- Bardziej odporne metryki to średnie odchylenie bezwzględne i mediana odchylenia bezwzględnego z mediany i percentyle (kwantyle).

Dla pogłębienia wiedzy

- Poświęcona statystyce strona Davida Lane'a zawiera sekcję dotyczącą percentyli: *http://onlinestatbook.com/2/introduction/percentiles.html*.
- Post Kevina Davenporta na R-Bloggers o odchyleniu od mediany i jej własnościach: *https://www.r-bloggers.com/2013/08/absolute-deviation-around-the-median/*.

Badanie rozkładu danych

Każdy z omówionych tu estymatorów podsumowuje dane w postaci jednej liczby opisującej ich położenie albo rozproszenie. Użyteczne jest również zbadanie, jak wygląda ogólny rozkład danych.

Kluczowe pojęcia dotyczące badania rozkładu danych

Boxplot

Wykres zaproponowany przez Tukeya jako szybka metoda wizualizacji rozkładu danych.

Synonim

wykres pudełkowy

Tabela liczności

Zestawienie częstości wystąpienia danych numerycznych rozdzielonych na interwały.

Histogram

Wykres tabeli częstości obrazujący interwały na osi x, a liczność (lub proporcje) na osi y. Wykresy kolumnowe z wyglądu bardzo przypominają histogramy, nie należy jednak ich ze sobą mylić. Różnice pomiędzy nimi zostały opisane w podrozdziale „Badanie danych binarnych i skategoryzowanych".

Wykres gęstości

Wygładzona wersja histogramu, zazwyczaj bazująca na **jądrowym estymatorze gęstości**.

Percentyle i boxploty

W punkcie „Estymatory oparte na percentylach" we wcześniejszej części tego rozdziału badaliśmy, jak percentyle mogą być wykorzystane do określenia rozrzutu danych. Percentyle są także wartościowym narzędziem do podsumowania całego rozkładu. Powszechnie używa się kwartyli (25., 50. i 75. percentyla) i decyli (10., 20., …, 90. percentyla). Percentyle są szczególnie przydatnym wskaźnikiem opisującym **ogony** rozkładu (końce zakresu) (ang. *tails*, *outer range*). Amerykańska kultura popularna stworzyła pojęcie *one-percenters*, które odnosi się do ludzi na szczycie — powyżej 99. percentyla — zamożności.

Tabela 1.4 przedstawia wybrane percentyle wskaźnika morderstw dla poszczególnych stanów. Korzystając z języka R, można ją stworzyć za pomocą funkcji `quantile`:

```
> quantile(state[['Murder.Rate']], p=c(.05, .25, .5, .75, .95))
    5%   25%   50%   75%   95%
1.600 2.425 4.000 5.550 6.510
```

Tabela 1.4. Percentyle wskaźnika morderstw dla poszczególnych stanów

5%	25%	50%	75%	95%
1,60	2,42	4,00	5,55	6,51

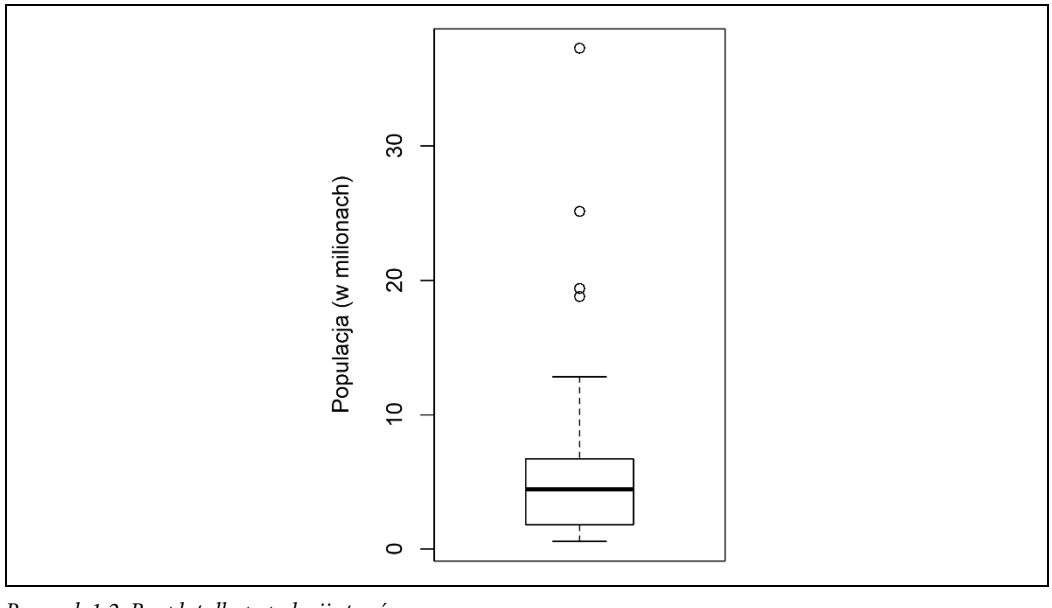
W Pythonie tę samą operację można wykonać za pomocą metody `quantile` dostępnej w ramce danych `pandas`:

```
state['Murder.Rate'].quantile([0.05, 0.25, 0.5, 0.75, 0.95])
```

Mediana wynosi 4 morderstwa na 100 000 osób, jednakże mamy tu do czynienia ze sporym rozproszeniem: 5. percentyl wynosi jedynie 1,6, a 95. — 6,51.

Boxploty (wykresy pudełkowe) (ang. *boxplots*), zaproponowane przez Tukeya [Tukey 1977], oparte są na percentylach i umożliwiają wizualizację rozkładu danych w szybki sposób. Rysunek 1.2 przedstawia stworzony w R boxplot dla populacji według stanów:

```
> boxplot(state[['Population']]/1000000, ylab='Populacja (w milionach)')
ax.set_ylabel('Populacja (w milionach)')
```



Rysunek 1.2. Boxplot dla populacji stanów

Moduł `pandas` zawiera wiele podstawowych wykresów przeznaczonych dla ramek danych; wśród nich znajdziemy także wykresy pudełkowe:

```
ax = (state['Population']/1_000_000).plot.box()
```

Z tego wykresu można od razu odczytać, że mediana populacji stanu wynosi mniej więcej pięć milionów, połowa stanów mieści się w zakresie od ok. 2 do ok. 7 milionów, a także że występują elementy odstające o dużych wartościach. Górna i dolna krawędź pudełka reprezentują, odpowiednio, 75. i 25. percentyl. Mediana jest przedstawiona jako poprzeczna linia wewnątrz pudełka. Linie wychodzące w górę i w dół poza pudełko, określane jako **wąsy** (ang. *whiskers*), wskazują zakres większości danych. Istnieje wiele wariacji na temat boxplotu; dla przykładu możesz przejrzeć dokumentację funkcji R `boxplot` [*R base* 2015]. Domyślnie funkcja R rozciąga wąsy do najdalszych punktów poza pudełkiem, z pominięciem tych, które przekraczają IQR 1,5 raza Ta sama implementacja jest wykorzystywana w pakiecie Matplotlib; inne oprogramowanie może działać zgodnie z innymi zasadami.

Każdy element poza zakresem wąsów jest przedstawiany za pomocą punktu lub kółka (w ten sposób często są symbolizowane wartości odstające).

Tablica częstości i histogramy

Tablica częstości zmiennej dzieli jej zakres na równoodległe przedziały i określa, jak wiele wartości znajduje się w każdym z nich. Tabela 1.5 przedstawia tablicę częstości dla populacji stanów stworzoną w R:

```
breaks <- seq(from=min(state[['Population']]),
              to=max(state[['Population']]), length=11)
pop_freq <- cut(state[['Population']], breaks=breaks,
              right=TRUE, include.lowest=TRUE)
table(pop_freq)
```

Tabela 1.5. Tablica częstości populacji według stanów

BinNumber	BinRange	Count	States
1.	563 626 – 4 232 658	24	WY,VT,ND,AK,SD,DE,MT,RI,NH,ME,HI,ID,NE,WV,NM,NV,UT,KS,AR,MS,IA,CT,OK,OR
2.	4 232 659 – 7 901 691	14	KY,LA,SC,AL,CO,MN,WI,MD,MO,TN,AZ,IN,MA,WA
3.	7 901 692 – 11 570 724	6	VA,NJ,NC,GA,MI,OH
4.	11 570 725 – 15 239 757	2	PA,IL
5.	15 239 758 – 18 908 790	1	FL
6.	18 908 791 – 22 577 823	1	NY
7.	22 577 824 – 26,246,856	1	TX
8.	26 246 857 – 29 915 889	0	
9.	29 915 890 – 33 584 922	0	
10.	33 584 923 – 37 253 956	1	CA

Funkcja `pandas.cut` tworzy szereg odwzorowujący wartość na segmenty. Dzięki metodzie `value_counts` otrzymujemy tabelę częstości:

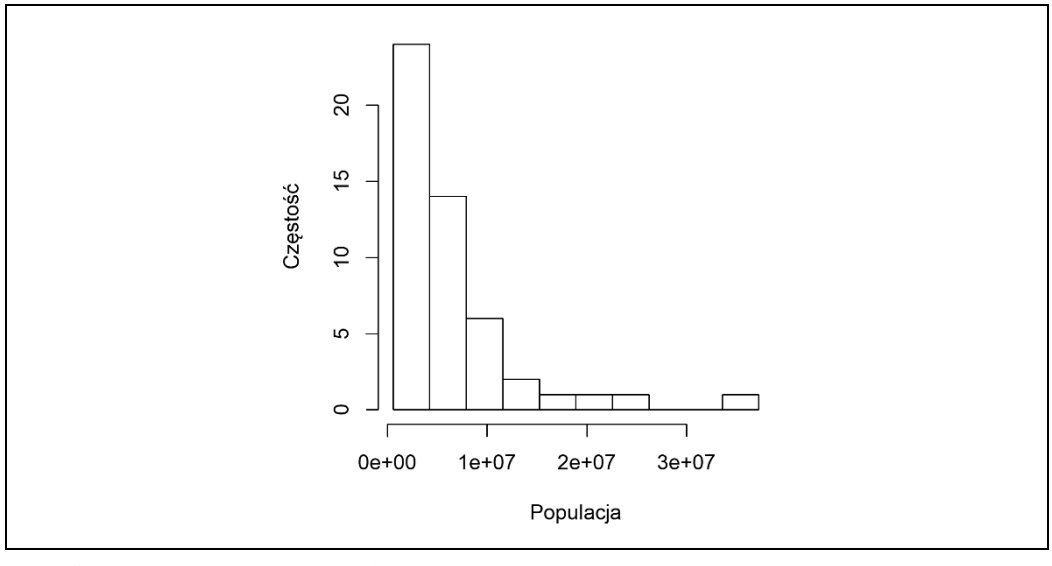
```
binnedPopulation = pd.cut(state['Population'], 10)
binnedPopulation.value_counts()
```

Najmniej licznym ze stanów jest Wyoming, gdzie zamieszkuje 563 626 osób, a najbardziej licznym ze stanów jest Kalifornia, gdzie mieszka 37 253 956 osób, co daje nam przedział 37 253 956 – 563 626 = = 36 690 330. Tę wartość musimy podzielić na równe przedziały, niech będzie ich 10, w wyniku czego otrzymujemy przedziały o zakresie 3 669 033. Pierwszy przedział będzie się rozciągał pomiędzy 563 626 a 4 232 658. Dla kontrastu najwyższy przedział ma zakres od 33 584 923 do 37 253 956 i zawiera jedynie jeden stan: Kalifornię. Dwa przedziały umieszczone nad nim są puste; zmiana następuje w kolejnym, gdzie znajduje się Teksas. Niezwykle ważne jest pozostawienie tych pustych przedziałów. To, że nie ma w nich żadnych wartości, jest również istotną informacją. Praktyczne może być również eksperymentowanie z różną liczbą przedziałów, co zmieni obejmowane przez nie zakresy. Jeśli będą zbyt duże, istotne cechy rozkładu mogą zostać rozmyte. Jeśli będą zbyt małe, wynik będzie zbyt poszatkowany i stracimy szerszy obraz całości.

Zarówno tablice częstości, jak i percentyle podsumowują dane poprzez tworzenie przedziałów. W ogólnym przypadku kwartyle i decyle będą miały równe liczby wartości w każdym przedziale (przedziały równoliczne), ale zakresy poszczególnych przedziałów będą się od siebie różniły. Natomiast tablica częstości będzie zawierać różne liczby wartości w przedziałach o tym samym zakresie (jednakowego rozmiaru), a rozmiary przedziałów będą takie same.

Histogram jest metodą wizualizacji tablicy częstości; przedziały znajdują się na osi x, a liczność danych jest na osi y. Na przykład na rysunku 1.3 przedział mający środek w punkcie 10 milionów (1e+07) mieści się w przybliżonym zakresie od 8 milionów do 12 milionów i uwzględnia sześć stanów. Żeby stworzyć histogram odpowiadający tabeli 1.5 w R, należy wykorzystać funkcję `hist` z argumentem `breaks`:

```
hist(state[['Population']], breaks=breaks)
```



Rysunek 1.3. Histogram populacji według stanów

Moduł pandas obsługuje histogramy ramek danych za pomocą metody DataFrame.plot.hist. Argument bins definiuje liczbę przedziałów. Różne metody zwracają obiekt osi, dzięki któremu można dalej uszczegóławiać wizualizację za pomocą pakietu Matplotlib:

```
ax = (state['Population'] / 1_000_000).plot.hist(figsize=(4, 4))
ax.set_xlabel('Populacja (w milionach)')
```

Wynik działania w postaci histogramu przedstawiono na rysunku 1.3.

- Puste przedziały włączamy do wykresu.
- Przedziały mają jednakową szerokość.
- Liczbę przedziałów (jest to równoznaczne z rozmiarem przedziałów) dobiera użytkownik.
- Kolejne słupki wykresu przylegają do siebie — nie ma pustych przestrzeni, chyba że istnieją puste przedziały.

W ogólnym przypadku histogramy rysowane są według następujących zasad:

Momenty statystyczne

W teorii statystyki położenie i rozproszenie są określane jako pierwszy i drugi **moment centralny** (ang. *moments*) rozkładu, a trzeci i czwarty moment centralny to **skośność** (ang. *skewness*) i **kurtoza** (ang. *kurtosis*). Skośność określa, czy dane są skośne do mniejszej lub większej wartości, a kurtoza wskazuje na obecność skrajnych wartości w danych. W ogólnym przypadku zamiast miar do obliczania skośności i kurtozy wykorzystuje się wizualne przedstawienia rozkładów do ich określenia, tak jak pokazano na rysunkach 1.2 i 1.3.

Szacowanie i wykresy gęstości rozkładu

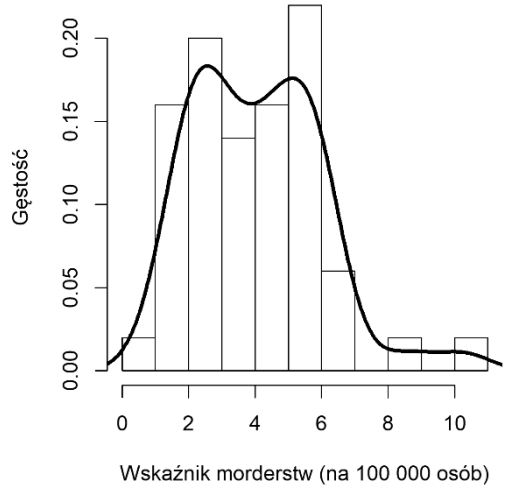
Bezpośrednio powiązany z histogramem jest wykres gęstości rozkładu, który przedstawia rozkład wartości danych jako linię ciągłą. Wykres gęstości można potraktować jako wygładzony histogram, mimo że jest on wyliczany bezpośrednio z danych, poprzez **estymator jądrowy gęstości** (ang. *kernal density estimate*) (krótki opis tego procesu znajdziesz w [Duong 2001]). Rysunek 1.4 przedstawia gęstość rozkładu złożoną z histogramem. W R możesz obliczyć gęstość rozkładu, wykorzystując funkcję density:

```
hist(state[['Murder.Rate']], freq=FALSE)
lines(density(state[['Murder.Rate']]), lwd=3, col='blue')
```

Moduł pandas zawiera metodę density służącą do tworzenia wykresu gęstości rozkładu. Argument bw_method określa stopień wygładzenia krzywej gęstości:

```
ax = state['Murder.Rate'].plot.hist(density=True, xlim=[0,12], bins=range(1,12))
state['Murder.Rate'].plot.density(ax=ax)      ❶
ax.set_xlabel('Wskaźnik morderstw (na 100 000 osób)')
```

❶ Funkcje wykresów często przyjmują dodatkowy argument osi (ax), dzięki któremu wykres jest dodawany do tego samego grafu.



Rysunek 1.4. Gęstość rozkładu wskaźnika morderstw dla poszczególnych stanów

Podstawową różnicą w stosunku do rysunku 1.3 jest skala na osi y: wykres gęstości wykorzystuje proporcje, nie liczności jak histogram (określasz to w R za pomocą argumentu `freq=FALSE`). Należy zauważyć, że całkowita powierzchnia pod krzywą gęstości wynosi 1, a zamiast liczby przedziałów obliczana jest tu powierzchnia pod krzywą pomiędzy dwoma dowolnymi punktami na osi x, co jest równoznaczne proporcji rozkładów znajdujących się pomiędzy tymi dwoma punktami.

Szacowanie gęstości rozkładu

Szacowanie gęstości to temat szeroki, o długiej historii w literaturze statystycznej. Ponad 20 dostępnych pakietów R zawiera funkcję do szacowania gęstości rozkładu. W swoim artykule z 2011 r. H. Deng i H. Wickham przedstawiają wszechstronny przegląd pakietów R, ze szczególnym uwzględnieniem `ASH` i `KernSmooth`. Metody oszacowywania gęstości w modułach `pandas` i `scikit-learn` również cechują się dobrymi implementacjami. Jednak w wielu sytuacjach w przypadku data science nie ma potrzeby roztrząsania niuansów różnych metod szacowania gęstości rozkładu; wystarczające są funkcje podstawowe.

Główne zasady

- Histogram częstości przedstawia częstość występowania na osi y, a wartości zmiennych na osi x, co w prosty sposób pozwala zobrazować rozkład danych.
- Tablica liczności jest stabelaryzowaną wersją częstości występowania, jaką przedstawia histogram.
- Boxplot — z górną i dolną wartością na, odpowiednio, 75. i 25. percentylu — także łatwo wizualizuje rozkład danych; najczęściej rysuje się boxploty jeden obok drugiego, by móc porównywać rozkłady.
- Wykres gęstości jest wygładzoną wersją histogramu i wymaga funkcji do oszacowania krzywej na podstawie danych (oczywiście istnieje wiele metod szacowania).

Dla pogłębienia wiedzy

- Wykładowca z SUNY Oswego udostępnił przewodnik, który krok po kroku opisuje, w jaki sposób tworzyć boxploty, dostępny pod adresem: *http://www.oswego.edu/~srp/stats/bp_con.htm*.
- Szacowanie gęstości w R jest przedstawione w artykule Henry'ego Denga i Hadleya Whickhama, dostępnym pod adresem: *http://vita.had.co.nz/papers/density-estimation.pdf*.
- R-Bloggers opublikowali praktyczny post o histogramach w R (*https://www.r-bloggers.com/2012/12/basics-of-histograms/*), zawierający również elementy personalizacji, takie jak binning (podziały).
- R-Bloggers opracowali również podobny artykuł o boxplotach, dostępny pod adresem: *https://www.r-bloggers.com/2013/06/box-plot-with-r-tutorial/*.
- Matthew Conlen opublikował interaktywną prezentację (*https://mathisonian.github.io/kde/*) ukazującą wpływ doboru różnych jąder i szerokości na jądrowe oszacowanie gęstości.

Badanie danych binarnych i skategoryzowanych

Dla danych skategoryzowanych prosty odsetek lub procent opowiada historię danych.

Kluczowe pojęcia dotyczące badania danych skategoryzowanych

Moda

Wartość występująca najczęściej w zbiorze danych.

Wartość oczekiwana

Jeśli kategorie można przypisać do wartości numerycznych, będzie to średnia wartość bazująca na prawdopodobieństwie wystąpienia danej kategorii.

Wykres słupkowy

Częstość lub odsetek dla każdej kategorii przedstawione w postaci słupków.

Wykres kołowy

Częstość lub odsetek dla każdej kategorii przedstawione jako wycinek koła.

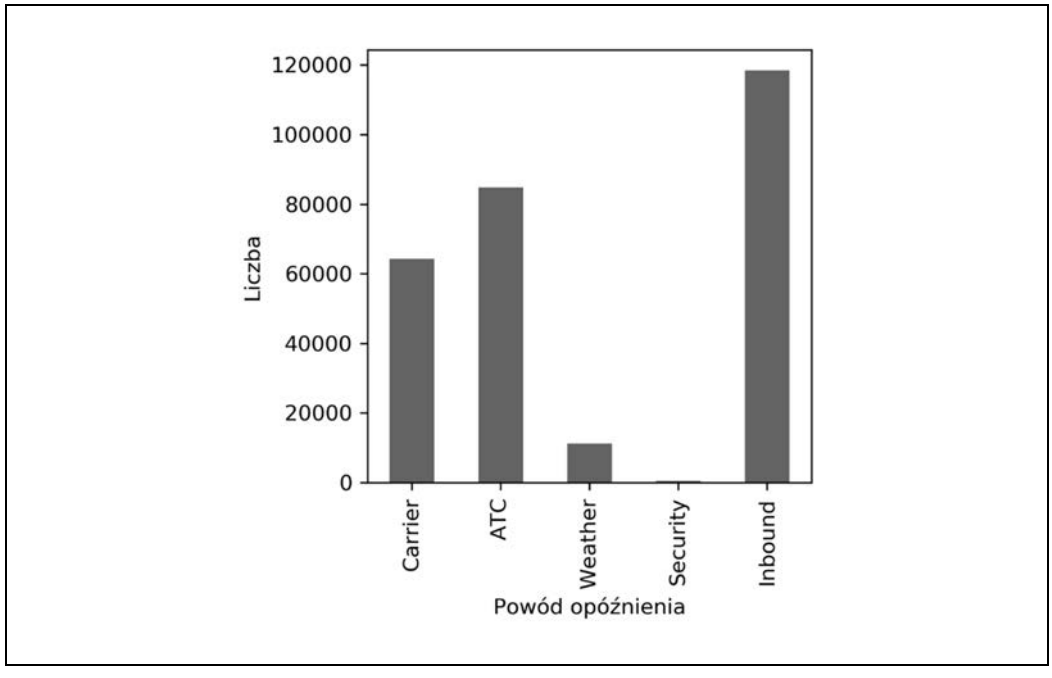
Stworzenie opisu dla zmiennej binarnej lub skategoryzowanej z kilkoma kategoriami jest całkiem proste: trzeba znaleźć odsetek jedynek albo najważniejszej kategorii. Tabela 1.6 przedstawia przykładowy procent opóźnionych lotów na lotnisku Fort Worth w Dallas od 2010 r. w zależności od przyczyny opóźnienia. Opóźnienia są podzielone według następujących kategorii: z winy przewoźnika (Carrier), z winy systemu kontroli lotów (ATC), przez pogodę (Weather), ze względów bezpieczeństwa (Security), przez opóźniony przylot (Inbound).

Tabela 1.6. Procent opóźnień na lotnisku Fort Worth w Dallas ze względu na ich przyczynę

Carrier	ATC	Weather	Security	Inbound
23,02	30,40	4,03	0,12	42,43

Wykresy słupkowe są powszechną metodą przedstawiania pojedynczych danych skategoryzowanych, często można je znaleźć w prasie popularnej. Kategorie są wypisane na osi x, a częstość występowania lub odsetki na osi y. Rysunek 1.5 przedstawia dane dla opóźnień lotów na lotnisku Fort Worth w Dallas (DFW) dla jednego roku w zależności od przyczyny; można go stworzyć za pomocą funkcji R `barplot`:

```
barplot(as.matrix(dfw) / 6, cex.axis=0.8, cex.names=0.7,
        xlab='Powód opóźnienia', ylab='Liczba')
```



Rysunek 1.5. Wykres słupkowy dla opóźnień na lotnisku DFW ze względu na przyczynę

Moduł `pandas` także pozwala wyświetlać ramki danych za pomocą wykresów słupkowych:

```
ax = dfw.transpose().plot.bar(figsize=(4, 4), legend=False)
ax.set_xlabel('Powód opóźnienia')
ax.set_ylabel('Liczba')
```

Zwróć uwagę na to, że wykres słupkowy przypomina histogram; w wykresie słupkowym na osi x przedstawione są poszczególne kategorie, a w histogramie oś x reprezentuje wartości numeryczne jednej zmiennej. W histogramie słupki wykresu najczęściej przylegają do siebie; przerwa sugeruje wartości, które nie występują w danych. Na wykresie słupkowym natomiast słupki są od siebie oddzielone.

Wykres kołowy jest alternatywą dla wykresu słupkowego, jednakże statystycy i eksperci w dziedzinie wizualizacji danych unikają go, gdyż wizualnie jest trudniejszy w interpretacji [Few 2007].

Dane numeryczne jako dane skategoryzowane

W punkcie „Tablica częstości i histogramy" we wcześniejszej części tego rozdziału prezentowane były tablice częstości oparte na przedziałach danych, co wprost nadaje danym numerycznym cechę uporządkowania. W tym sensie histogram i wykres słupkowy są podobne, jedynie kategorie na osi x wykresu słupkowego nie są uporządkowane. Zamiana danych numerycznych na dane kategoryzowane jest ważnym i szeroko stosowanym aspektem w analizie danych, ułatwiającym zmniejszenie złożoności (rozmiaru) danych. Pomaga to w wykrywaniu zależności pomiędzy cechami, zwłaszcza na początkowym etapie analizy.

Moda

Moda jest wartością — lub, w przypadkach łączonych, wartościami — która pojawia się najczęściej w danych. Modą w przykładzie dotyczącym opóźnień lotów z lotniska Fort Worth w Dallas jest „przylot". Modą dla większości regionów Stanów Zjednoczonych w przypadku wyznania będzie „chrześcijanin". Moda jest prostą statystyką opisową dla danych kategoryzowanych i nie jest używana dla danych numerycznych.

Wartość oczekiwana

Specjalnym typem danych skategoryzowanych są dane, w których przypadku kategorie reprezentują dyskretne wartości na jednej skali lub mogą być do nich przypisane. Załóżmy, że handlowiec proponuje dwa typy usług dla nowej technologii w chmurze: jeden za 300 $/miesiąc, a drugi za 50 $/miesiąc. Oferuje on również darmowe webinarium wprowadzające. Firma obliczyła, że 5% uczestników webinarium wybrało ofertę za 300 $, 15% za 50 $, a 80% nie skorzystało z żadnej oferty. Powyższe dane mogą być opisane jako cel finansowy z pojedynczą „wartością oczekiwaną", przedstawioną w postaci średniej ważonej; wagami są w tym przypadku prawdopodobieństwa.

Wartość oczekiwaną możesz obliczyć następująco:

1. Przemnóż wszystkie wyniki przez prawdopodobieństwa ich występowania.
2. Zsumuj otrzymane wartości.

W przykładzie dotyczącym usług w chmurze wartość oczekiwana dla uczestnika webinarium wynosi 22,5 $/miesiąc, co obliczamy następująco:

$$WO = (0{,}05)(300) + (0{,}15)(50) + (0{,}80)(0) = 22{,}5$$

Wartość oczekiwana jest w rzeczywistości formą średniej ważonej: oddaje idee przyszłych oczekiwań i wag dotyczących prawdopodobieństwa, zazwyczaj bazuje na subiektywnej ocenie. Wartość oczekiwana jest podstawowym pojęciem w wycenie biznesowej i budżetowaniu — chodzi o wartość oczekiwaną z pięciu lat dla zysków z nowej inwestycji lub oczekiwane oszczędności z nowego oprogramowania do zarządzania w klinice.

Prawdopodobieństwo

Wspomnieliśmy powyżej o **prawdopodobieństwie** (ang. *probability*) wystąpienia wartości. Większość ludzi rozumie pojęcie prawdopodobieństwa intuicyjnie i spotyka się z nim często podczas oglądania prognozy pogody (możliwość wystąpienia opadów) lub wydarzeń sportowych (prawdopodobieństwo zwycięstwa). W sporcie i grach częściej stosuje się pojęcie **szans** (ang. *odds*), które można z łatwością przekształcić w prawdopodobieństwo (jeżeli szanse zespołu na zwycięstwo wynoszą dwa do jednego, prawdopodobieństwo wygranej jest równe 2/(2+1) = 2/3. Co ciekawe, próby zdefiniowania prawdopodobieństwa stanowią źródło ożywionej i głębokiej dyskusji filozoficznej. Na szczęście nie potrzebujemy tutaj formalnej definicji matematycznej ani filozoficznej. W naszym przypadku prawdopodobieństwo wystąpienia jakiegoś zdarzenia zdefiniujemy jako odsetek przypadków, w jakich to zdarzenie wystąpi, przy założeniu, że dana sytuacja będzie powtarzana niezliczoną liczbę razy. Najczęściej taki konstrukt jest czysto abstrakcyjny, ale pozwala dobrze zrozumieć pojęcie prawdopodobieństwa.

Główne zasady

- Dane skategoryzowane są zazwyczaj opisywane za pomocą procentów i mogą być przedstawiane na wykresach słupkowych.
- Kategorie mogą reprezentować odrębne podmioty (jabłka i pomarańcze, mężczyzn i kobiety), poziomy zmiennej czynnikowej (niski, średni, wysoki) lub dane numeryczne, które można zaklasyfikować do przedziałów.
- Wartość oczekiwana jest sumą wartości pomnożonych przez ich prawdopodobieństwo wystąpienia; najczęściej sumuje się poziomy zmiennej czynnikowej.

Dla pogłębienia wiedzy

Kurs statystyki nie jest pełny bez rozdziału o metodach manipulacji wykresem w celu wprowadzenia czytelnika w błąd (*https://en.wikipedia.org/wiki/Misleading_graph*); często dotyczy to wykresów słupkowych i kołowych.

Korelacja

Badania eksploracyjne danych w wielu przypadkach (czy to w data science, czy w nauce) wymagają sprawdzenia korelacji pomiędzy zmiennymi predykcyjnymi, jak również między zmiennymi predykcyjnymi a zmienną decyzyjną. Można powiedzieć, że zmienne X i Y (każda z nich zawiera zmierzone dane) są pozytywnie skorelowane, jeśli wysokie wartości X odpowiadają wysokim wartościom Y, a niskie wartości X odpowiadają niskim wartościom Y. Jeśli wysokie wartości X odpowiadają niskim wartościom Y i odwrotnie, to mówimy, że zmienne są negatywnie skorelowane.

Kluczowe pojęcia dotyczące korelacji

Współczynnik korelacji

Metryka, która mierzy, do jakiego stopnia zmienne numeryczne są ze sobą powiązane (zmienia się od –1 do +1).

Macierz korelacji

Tabela, w której zmienne powtarzają się w wierszach i kolumnach, a wartości w komórkach odpowiadają korelacji pomiędzy nimi.

Wykres punktowy

Wykres, w którym na osi x przedstawione są wartości jednej zmiennej, a na osi y wartości drugiej zmiennej.

Rozważmy przypadek dwóch zmiennych idealnie skorelowanych:

```
v1: {1, 2, 3}
v2: {4, 5, 6}
```

Wektor sumy iloczynów wynosi 1·4+2·5+3·6 = 32. Spróbuj przemieszać jeden z nich i policzyć od nowa — wektor sumy iloczynów nigdy nie będzie większy niż 32. Obliczona suma iloczynów może być wykorzystana jako metryka; oznacza to, że obserwowana suma 32 może być porównywana do wielu innych losowo przemieszanych wyników (w rzeczywistości idea ta nawiązuje do szacowania opartego na testach randomizacyjnych; patrz punkt „Test permutacyjny" w rozdziale 3.). Jednak wartości obliczone za pomocą tej metryki nie są zbyt znaczące, poza odniesieniem do rozkładu randomizowanego.

Bardziej użyteczny jest wariant standaryzowany: **współczynnik korelacji** (ang. *correlation coefficient*), który odpowiada oszacowaniu korelacji pomiędzy dwoma zmiennymi, przedstawionymi w tej samej skali. Aby obliczyć **współczynnik korelacji Pearsona** (ang. *Pearson's correlation coefficient*), należy pomnożyć odchylenia od średniej dla zmiennej 1, pomnożyć przez odchylenia od średniej dla zmiennej 2, a następnie podzielić przez iloczyn odchyleń standardowych:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

Zwróć uwagę na to, że dzielimy przez *n–1* zamiast przez *n* (patrz ramka „Stopnie swobody i n czy n–1" we wcześniejszej części tego rozdziału). Współczynnik korelacji zawsze przyjmuje wartości od +1 (idealna dodatnia korelacja) do –1 (idealna negatywna korelacja); 0 wskazuje na brak korelacji.

Zmienne mogą być powiązane również w sposób nieliniowy; w takim przypadku współczynnik korelacji może nie być przydatną metryką. Związek pomiędzy wysokością podatków a ich ściągalnością może być tego przykładem: wysokość podatków rośnie od 0, ściągalność również. Jednakże kiedy wysokość podatków osiąga wysoki poziom, zbliżając się do 100%, rośnie liczba osób uchylających się od ich płacenia, dlatego dochód z podatków w zasadzie maleje.

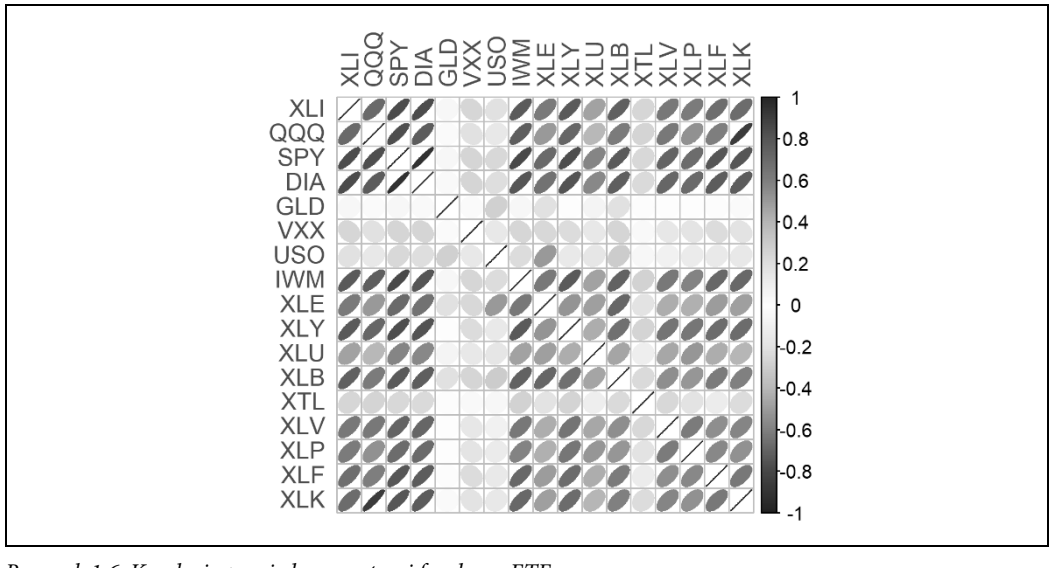
Tabela 1.7, nazywana **macierzą korelacji** (ang. *correlation matrix*), przedstawia korelacje pomiędzy dziennymi zwrotami dla akcji telekomunikacyjnych od lipca 2012 r. do czerwca 2015 r. Z tabeli możesz odczytać, że Verizon (VZ) i ATT (T) mają największą korelację z pozostałymi akcjami. Level Three (LVLT), które jest przedsiębiorstwem zajmującym się infrastrukturą, ma najniższą korelację w stosunku do pozostałych. Zauważ, że na przekątnej macierzy wartości wynoszą 1 (korelacja akcji z samą sobą), a wartości w macierzy nad i pod przekątną powtarzają się.

Tabela 1.7. Korelacja pomiędzy zwrotami z akcji telekomunikacyjnych

	T	CTL	FTR	VZ	LVLT
T	1,000	0,455	0,359	0,681	0,082
CLT	0,455	1,000	0,435	0,448	0,096
FTR	0,359	0,435	1,000	0,349	0,111
VZ	0,681	0,448	0,349	1,000	0,096
LVLT	0,082	0,096	0,111	0,096	1,000

Tabela korelacji, taka jak tabela 1.7, jest najczęściej przedstawiana w postaci wykresu, by można było zwizualizować zależności pomiędzy wieloma zmiennymi. Rysunek 1.6 przedstawia korelację pomiędzy dziennymi zwrotami dla głównych funduszy **ETF** (ang. *exchange-traded fund*). W R można w prosty sposób stworzyć taki wykres za pomocą funkcji `corrplot`:

```
etfs <- sp500_px[row.names(sp500_px) > '2012-07-01',
                 sp500_sym[sp500_sym$sector == 'etf', 'symbol']]
library(corrplot)
corrplot(cor(etfs), method='ellipse')
```



Rysunek 1.6. Korelacja pomiędzy zwrotami funduszy ETF

Jest możliwe utworzenie takiego samego wykresu w Pythonie, ale żaden ze standardowych pakietów nie zawiera odpowiedniej implementacji. Jednakże większość z nich obsługuje wizualizację macierzy korelacji za pomocą map cieplnych. W poniższym listingu wykorzystamy w tym celu pakiet `seaborn.heatmap`. W materiałach dodatkowych umieściliśmy kod Pythona generujący bardziej zaawansowaną wizualizację:

```
etfs = sp500_px.loc[sp500_px.index > '2012-07-01',
                    sp500_sym[sp500_sym['sector'] == 'etf']['symbol']]
sns.heatmap(etfs.corr(), vmin=-1, vmax=1,
            cmap=sns.diverging_palette(20, 220, as_cmap=True))
```

Wyniki ETF dla S&P 500 (SPY) i dla Dow Jones Index (DIA) mają wysoką korelację. Podobnie QQQ i XLK, złożone głownie ze spółek technologicznych, są pozytywnie skorelowane. Defensywne ETF-y, takie, które śledzą np. ceny złota (GLD), ceny ropy naftowej (USO) czy zmienność rynku (VXX), zdają się słabo lub negatywnie skorelowane z pozostałymi wskaźnikami. Orientacja elipsy wskazuje, czy dwie zmienne są skorelowane dodatnio (elipsa skierowana w prawo i do góry), czy ujemnie (elipsa skierowana w lewo i do góry). Odcień i grubość elipsy wskazują na siłę powiązania: cieńsza i ciemniejsza elipsa odpowiada silniejszym relacjom.

Podobnie jak średnia i odchylenie standardowe, współczynnik korelacji jest wrażliwy na dane odstające. Pakiety oprogramowania oferują wydajne alternatywy dla klasycznego współczynnika korelacji, np. dostępny w R pakiet `robust` (*https://cran.r-project.org/web/packages/robust/robust.pdf*) wykorzystuje funkcję `covRob` do obliczania odpornego oszacowania korelacji. Metody dostępne w module `sklearn.covariance` (*https://scikit-learn.org/stable/modules/classes.html#module-sklearn.covariance*) pakietu scikit-learn implementują różne możliwe rozwiązania.

Inne oszacowania korelacji

Statystycy wiele lat temu zaproponowali inne współczynniki korelacji, takie jak **rho Spearmana** (*Spearman's rho*) czy **tau Kendalla** (*Kendall's tau*). Te współczynniki korelacji bazują na szeregach danych. Ponieważ obliczane są na podstawie rankingów, a nie wartości, oszacowania te są odporne na wartości odstające i pewne typy nieliniowości. W data science wystarczą współczynnik korelacji Pearsona i jego alternatywne, wydajniejsze wersje. Odnoszenie się do rankingów oszacowań jest stosowane w przypadku mniejszych zbiorów danych i specyficznych testów hipotez.

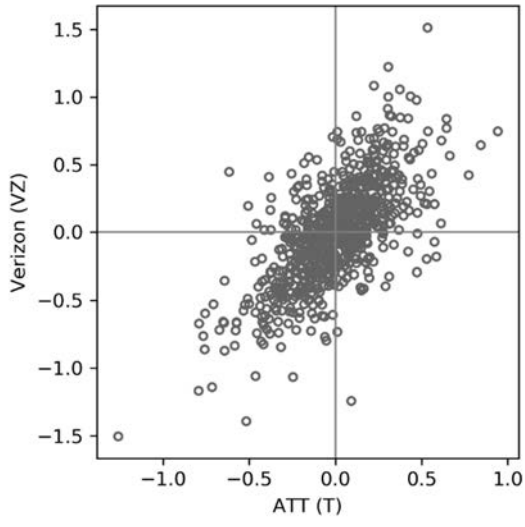
Wykres punktowy

Standardowym sposobem wizualizacji związku pomiędzy dwoma mierzonymi zmiennymi jest **wykres punktowy** (ang. *scatterplot*). Oś x reprezentuje jedną zmienną, a oś y drugą; każdy punkt wykresu jest wierszem. Rysunek 1.7 przedstawia wykres korelacji pomiędzy dziennymi zwrotami dla firm ATT i Verizon. Można go stworzyć za pomocą komendy w R:

```
plot(telecom$T, telecom$VZ, xlab='ATT (T)', ylab='Verizon (VZ)')
```

Taki sam wykres można wygenerować w Pythonie za pomocą metody `scatter` z modułu pandas:

```
ax = telecom.plot.scatter(x='T', y='VZ', figsize=(4, 4), marker='$\u25EF$')
ax.set_xlabel('ATT (T)')
ax.set_ylabel('Verizon (VZ)')
ax.axhline(0, color='grey', lw=1)
ax.axvline(0, color='grey', lw=1)
```



Rysunek 1.7. Wykres punktowy korelacji pomiędzy zwrotami dla ATT (T) i Verizon (VZ)

Zwroty mają pozytywne powiązanie: gdy dane są zgrupowane w pobliżu środka układu współrzędnych, przez większość dni wartości obu akcji rosły lub malały wspólnie (ćwiartki pierwsza i trzecia). Było tylko kilka dni, kiedy wartości jednej akcji znacząco malały, a drugiej rosły i odwrotnie (ćwiartki druga i czwarta).

Wykres widoczny na rysunku 1.7 zawiera jedynie 754 punkty danych, ale jest oczywiste, że rozpoznawanie szczegółów w środku wykresu jest niemal niemożliwe. W dalszej części rozdziału pokażemy, że dodawanie przezroczystości do punktów danych lub stosowanie przedziałów heksagonalnych i wykresów gęstości ułatwia wykrywanie ukrytych struktur danych.

Główne zasady

- Współczynnik korelacji mierzy stopień, w jakim dwie zmienne (np. wzrost i waga człowieka) są powiązane ze sobą.
- Kiedy wraz z wysoką wartością v1 występuje wysoka wartość v2, wtedy v1 i v2 są powiązane dodatnio.
- Kiedy wraz z wysoką wartością v1 występuje niska wartość v2, wtedy v1 i v2 są powiązane ujemnie.
- Współczynnik korelacji jest standardową metryką, która zawsze przyjmuje wartości z zakresu –1 (idealna negatywna korelacja) do +1 (idealna pozytywna korelacja).
- Współczynnik korelacji o wartości 0 wskazuje na brak korelacji, ale musisz być świadomy, że losowe zbiory danych mogą przez przypadek dawać pozytywny bądź negatywny współczynnik korelacji.

Dla pogłębienia wiedzy

Bardzo dobre omówienie zagadnienia korelacji znajdziesz w: David Freedman, Robert Pisani, Roger Purves, *Statistics, 4th ed.*, W.W. Norton, 2007.

Badanie dwóch lub więcej zmiennych

Znane nam już estymatory, takie jak średnia czy wariancja, biorą pod uwagę tylko jedną zmienną naraz [jest to **analiza jednoczynnikowa** (ang. *univariate analysis*)]. Analiza korelacji (patrz podrozdział „Korelacja" we wcześniejszej części tego rozdziału) jest ważną metodą, która porównuje dwie zmienne [jest to **analiza dwuczynnikowa** (ang. *bivariate analysis*)]. W tej części zajmiemy się dodatkowymi oszacowaniami i wykresami dla więcej niż dwóch zmiennych [czyli **analizą wieloczynnikową** (ang. *multivariate analysis*)].

Kluczowe pojęcia dotyczące badania dwóch lub więcej zmiennych

Tablica kontyngencji

Zestawienie liczności dla dwóch lub więcej zmiennych skategoryzowanych.

Wykres hexagon binning

Wykres dwóch zmiennych numerycznych z wynikami przedstawionymi w postaci sześciokątów.

Wykres konturowy

Wykres przedstawiający gęstość dwóch zmiennych numerycznych podobnie jak mapy topograficzne.

Wykres skrzypcowy

Wykres podobny do boxplotu, ale przedstawiający szacowaną gęstość.

Podobnie jak analiza jednoczynnikowa, analiza wieloczynnikowa obejmuje zarówno obliczanie statystyk opisowych, jak i wizualizację wyników. Odpowiedni rodzaj analizy dwuczynnikowej lub wieloczynnikowej zależy od rodzaju danych: numerycznych lub skategoryzowanych.

Wykres przedziałów heksagonalnych i wykres konturowy (przedstawianie danych numerycznych względem danych numerycznych)

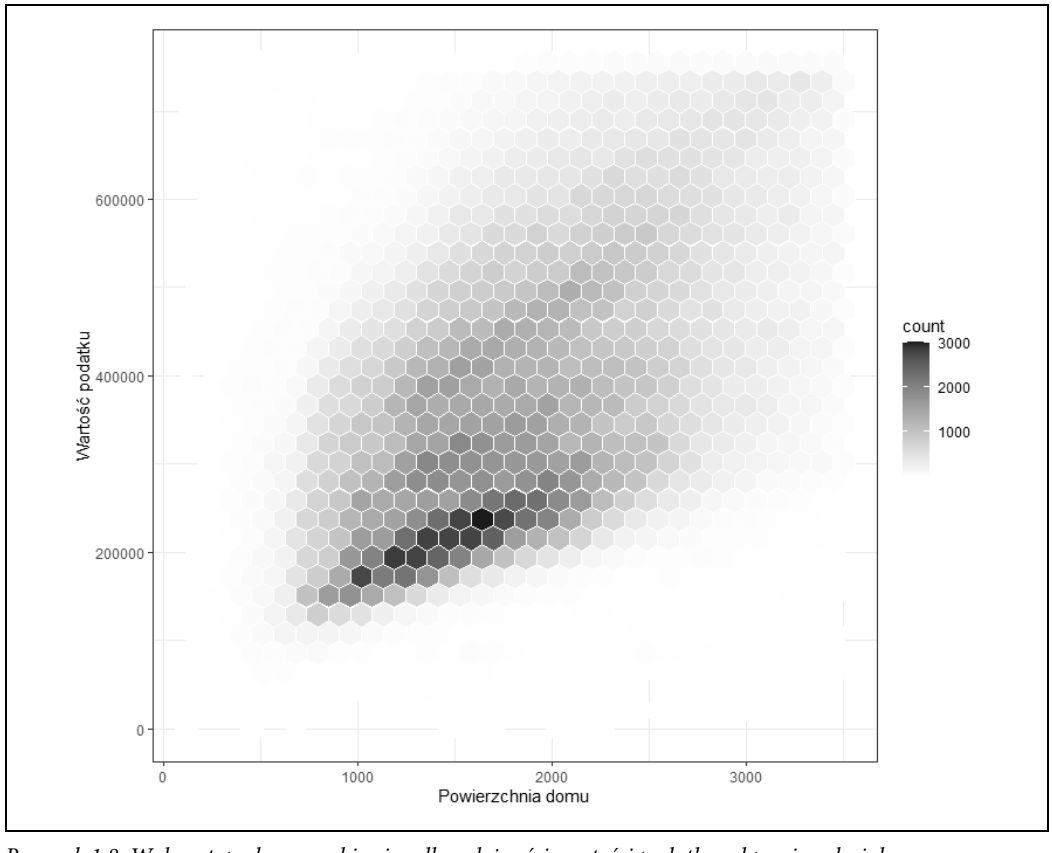
Wykresy punktowe są dobre, gdy musisz przedstawić względnie niewielki zbiór danych. Wykres zwrotu z akcji z rysunku 1.7 zawierał około 750 punktów. Dla każdego zbioru danych z setkami tysięcy lub milionami wpisów wykres punktowy byłby zbyt gęsty, dlatego potrzebujemy innego sposobu na wizualizację tych zależności. Dla zobrazowania problemu rozważmy zbiór danych `kc_tax`, który zawiera szacowany podatek dla domów mieszkalnych w King County w stanie Waszyngton. Żeby móc się skupić na najważniejszej części danych, pozbędziemy się bardzo drogich, bardzo małych lub bardzo dużych posiadłości; wykorzystamy do tego funkcję `subset`:

```
kc_tax0 <- subset(kc_tax, TaxAssessedValue < 750000 &
                  SqFtTotLiving > 100 &
                  SqFtTotLiving < 3500)
nrow(kc_tax0)
432693
```

W module pandas filtrujemy zestaw danych następująco:

```
kc_tax0 = kc_tax.loc[(kc_tax.TaxAssessedValue < 750000) &
                     (kc_tax.SqFtTotLiving > 100) &
                     (kc_tax.SqFtTotLiving < 3500), :]
kc_tax0.shape
(432693, 3)
```

Rysunek 1.8 jest wykresem **przedziałów heksagonalnych** (ang. *hexagonal binning*, *hexbin*) dla zależności pomiędzy rozmiarem końcowym a obliczoną wartością podatku dla domów w King County. Zamiast wykreślać punkty, które stworzyłyby jednolitą ciemną plamę, pogrupowaliśmy wpisy w sześcienne komórki i narysowaliśmy je w taki sposób, żeby kolor odpowiadał liczbie elementów w danej komórce. Na tym wykresie dodatnia korelacja pomiędzy wielkością a naliczonym podatkiem jest oczywista. Interesującą cechą jest ślad dodatkowych pasm powyżej pasma głównego (czarnego) znajdującego się u spodu, które wskazuje na domy o podobnej powierzchni, jak w głównym paśmie, ale z przypisanym większym podatkiem.



Rysunek 1.8. Wykres typu hexagon binning dla zależności wartości podatku od powierzchni domu

Rysunek 1.8 został wygenerowany przez potężny pakiet R ggplot2, stworzony przez Hadleya Wickhama [ggplot2]. `ggplot2` jest jedną z nowych bibliotek stworzonych do zaawansowanej, wizualnej analizy danych (patrz punkt „Wizualizacja wielu zmiennych" w dalszej części tego rozdziału).

```
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(color='white') +
  theme_bw() +
  scale_fill_gradient(low='white', high='black') +
  labs(x='Powierzchnia domu', y='Wartość podatku')
```

W Pythonie wykresy typu hexbin są dostępne w metodzie `hexbin` ramki danych pandas:

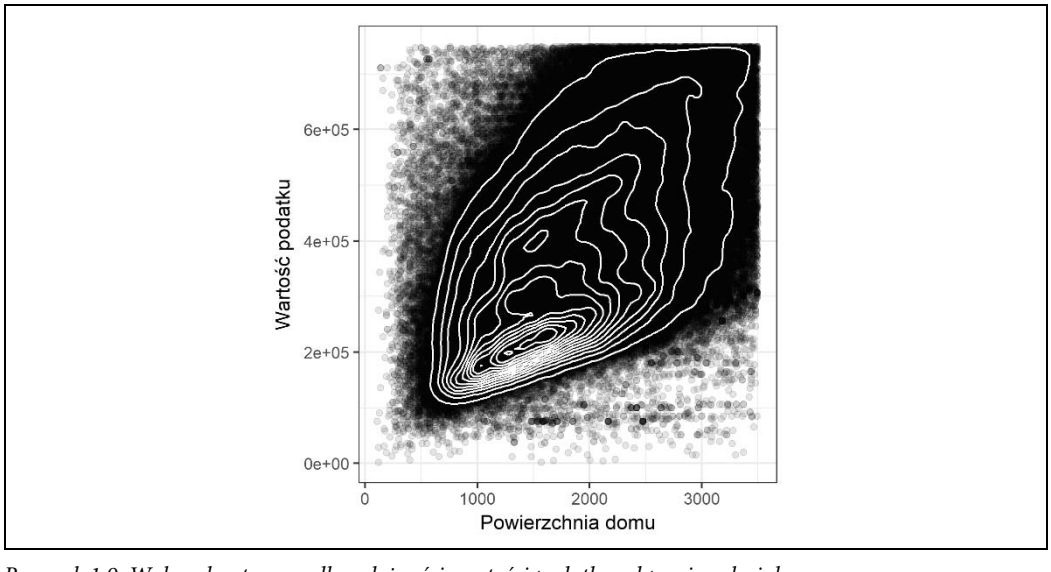
```
ax = kc_tax0.plot.hexbin(x='SqFtTotLiving', y='TaxAssessedValue',
                         gridsize=30, sharex=False, figsize=(5, 4))
ax.set_xlabel('Powierzchnia domu')
ax.set_ylabel('Wartość podatku')
```

Rysunek 1.9 wykorzystuje kontury nałożone na wykres punktowy, by zobrazować związki pomiędzy dwoma zmiennymi numerycznymi. Kontury są zasadniczo mapą topograficzną dwóch zmiennych; każdy kontur zawiera konkretne zagęszczenie punktów rosnących w kierunku „szczytu". Przedstawiony wykres obrazuje podobny przykład jak rysunek 1.8: widoczny jest drugi wierzchołek „na północ" od głównego szczytu. Ten wykres stworzono za pomocą funkcji `geom_density2d`, również z biblioteki `ggplot2`.

```
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue)) +
  theme_bw() +
  geom_point(alpha=0.1) +
  geom_density2d(color='white') +
  labs(x='Powierzchnia domu', y='Wartość podatku')
```

Funkcja `kdeplot` z pakietu `seaborn` w Pythonie tworzy wykres konturowy:

```
ax = sns.kdeplot(kc_tax0.SqFtTotLiving, kc_tax0.TaxAssessedValue, ax=ax)
ax.set_xlabel('Powierzchnia domu')
ax.set_ylabel('Wartość podatku')
```



Rysunek 1.9. Wykres konturowy dla zależności wartości podatku od powierzchni domu

Istnieją również inne rodzaje wykresów wykorzystywane do obrazowania zależności pomiędzy dwoma zmiennymi numerycznymi, np. **heatmapy** (ang. *heat maps*). Heatmapy, wykresy hexagon binning i konturowe obrazują gęstość w dwóch wymiarach. Dzięki temu są naturalnymi analogami do histogramów i wykresów gęstości.

Dwie zmienne skategoryzowane

Prostym sposobem na opisanie dwóch zmiennych skategoryzowanych jest tablica kontyngencji — tablica liczności według kategorii. Tabela 1.8 przedstawia tablicę kontyngencji pomiędzy notami kredytów osobistych a uzyskanym wynikiem. Dane są udostępnione przez firmę Lending Club, lidera na rynku bezpośrednich usług kredytowych. Skala zaczyna się od A (wysoki) do G (niski). Wynikami mogą być: spłacony, trwający, opóźniony, oddalony (nie jest spodziewana spłata zadłużenia). Tabela przedstawia liczności i wartości procentowe. Wysoko notowane pożyczki mają niewielki odsetek opóźnionych spłat lub oddaleń w porównaniu do pożyczek niżej ocenionych.

Tabela 1.8. Tablica kontyngencji dla noty pożyczki i jej statusu

Grade	Charged Off	Current	Fully Paid	Late	Total
A	1562	50 051	20 408	469	72 490
	0,022	0,69	0,282	0,006	0,161
B	5302	93 852	31 160	2056	132 370
	0,04	0,709	0,235	0,016	0,294
C	6023	88 928	23 147	2777	120 875
	0,05	0,736	0,191	0,023	0,268
D	5007	53 281	13 681	2308	74 277
	0,067	0,717	0,184	0,031	0,165
E	2842	24 639	5949	1374	34 804
	0,082	0,708	0,171	0,039	0,077
F	1526	8444	2328	606	12 904
	0,118	0,654	0,18	0,047	0,029
G	409	1990	643	199	3241
	0,126	0,614	0,198	0,061	0,007
Suma	22 671	321 185	97 316	9789	450 961

Tabele kontyngencji mogą przechowywać tylko zliczenia albo mogą zawierać także wartości procentowe dla każdej kolumny oraz wartość sumaryczną. Tabela przestawna w Excelu jest najprawdopodobniej najpopularniejszym narzędziem do tworzenia tablicy kontyngencji. W R funkcja `CrossTable` z pakietu `descr` tworzy tablice kontyngencji, a do stworzenia tabeli 1.8 wykorzystano następującą funkcję:

```
library(descr)
x_tab <- CrossTable(lc_loans$grade, lc_loans$status,
                    prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)
```

Metoda `pivot_table` tworzy tabelę przestawną w Pythonie. Za pomocą argumentu `aggfunc` uzyskujemy zliczenia. Obliczanie wartości procentowych okazuje się nieco bardziej skomplikowane:

```
crosstab = lc_loans.pivot_table(index='grade', columns='status',
                                aggfunc=lambda x: len(x), margins=True)   ❶

df = crosstab.loc['A':'G',:].copy()   ❷
df.loc[:,'Charged Off':'Late'] = df.loc[:,'Charged Off':'Late'].div(df['All'],
                                                                   axis=0)   ❸
df['All'] = df['All'] / sum(df['All'])   ❹
perc_crosstab = df
```

❶ Argument `margins` dodaje sumy kolumn i rzędów.

❷ Tworzymy kopię tabeli przestawnej.

❸ Dzielimy rzędy przez sumę rzędów.

❹ Dzielimy kolumnę `'All'` przez jej sumę.

Dane kategoryzowane i numeryczne

Zestaw boxplotów (patrz punkt „Percentyle i boxploty" we wcześniejszej części tego rozdziału) jest prostą metodą porównywania rozkładów danych numerycznych pogrupowanych według zmiennej kategorycznej. Możemy chcieć porównać odsetek opóźnionych lotów w zależności od przewoźnika. Rysunek 1.10 przedstawia odsetek lotów opóźnionych z winy przewoźnika na miesiąc.

```
boxplot(pct_carrier_delay ~ airline, data=airline_stats, ylim=c(0,50))
```

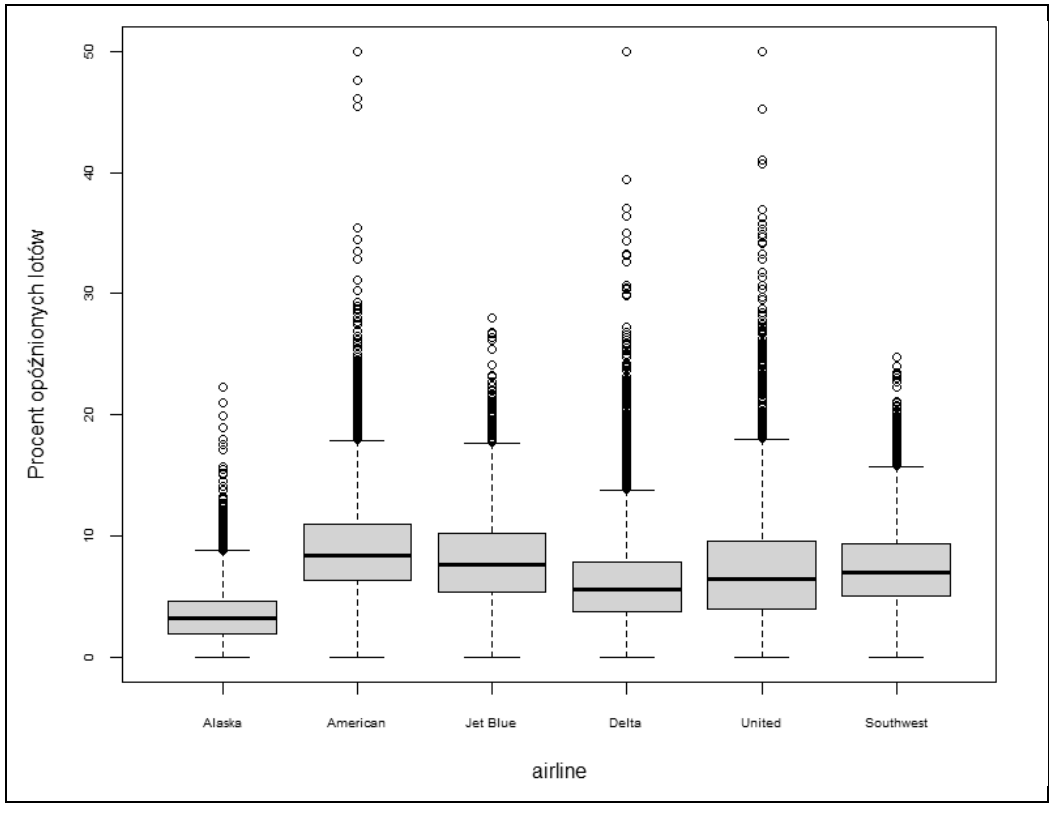
Metoda `boxplot` z modułu pandas przyjmuje argument `by` rozdzielający zestaw danych na grupy, a następnie tworzy poszczególne wykresy pudełkowe:

```
ax = airline_stats.boxplot(by='airline', column='pct_carrier_delay')
ax.set_xlabel('')
ax.set_ylabel('Procent opóźnionych lotów')
plt.suptitle('')
```

Alaska wydaje się mieć najmniejsze problemy z opóźnieniami, a American ma ich najwięcej: dolny kwartyl dla firmy American jest wyżej niż górny dla firmy Alaska.

Wykres skrzypcowy (ang. *violin plot*) wprowadzony w 1998 r. przez Hintzego i Nelsona jest rozszerzoną wersją boxplotu i obrazuje szacowanie gęstości wraz z gęstością na osi y. Gęstość jest odbita i odwrócona, a kształt wynikowy tworzy obraz przypominający skrzypce. Zaletą wykresu skrzypcowego jest ukazywanie niuansów w różnicach rozkładów, które nie są zauważalne w boxplotach. Jednocześnie boxplot przedstawia wartości odstające w sposób bardziej przejrzysty. W pakiecie `ggplot2` jest funkcja `geom_violin`, przy pomocy której można stworzyć wykres skrzypcowy:

```
ggplot(data=airline_stats, aes(airline, pct_carrier_delay))  +
  ylim(0, 50) +
  geom_violin() +
  labs(x='', y='Procent opóźnionych lotów')
```



Rysunek 1.10. Boxplot dla odsetka lotów opóźnionych z winy przewoźnika

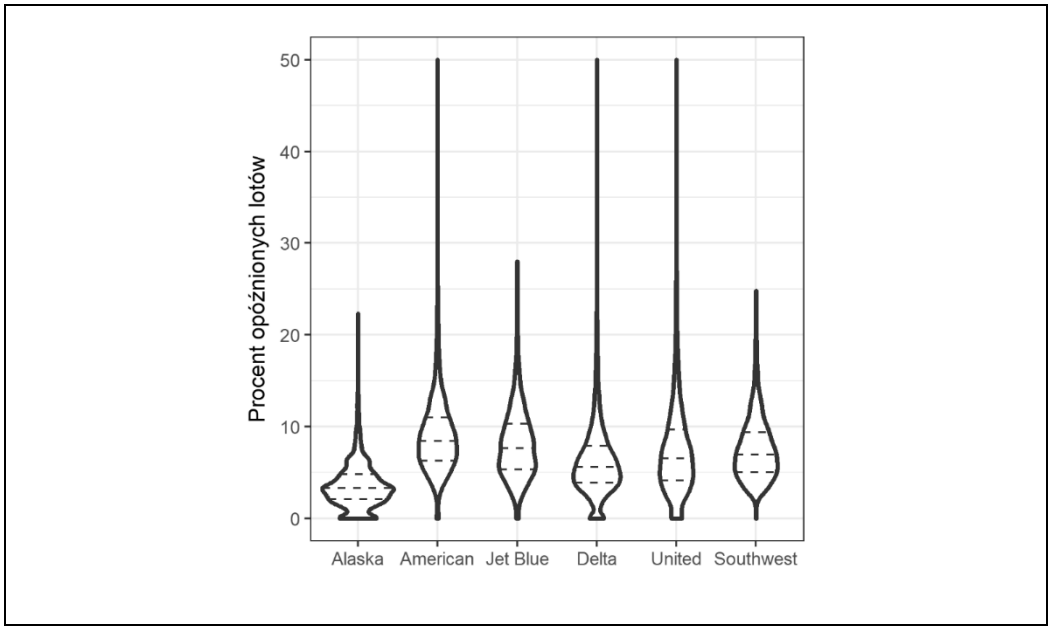
Wykresy skrzypcowe są dostępne poprzez metodę `violinplot` z pakietu `seaborn`:

```
ax = sns.violinplot(airline_stats.airline, airline_stats.pct_carrier_delay,
                    inner='quartile', color='white')
ax.set_xlabel('')
ax.set_ylabel('Procent opóźnionych lotów')
```

Odpowiadający temu fragmentowi wykres został przedstawiony na rysunku 1.11. Wykres skrzypcowy pokazuje koncentrację wokół zera dla Alaski i w mniejszym zakresie dla Delty. Takie zjawisko nie jest oczywiste w boxplocie. Możesz połączyć wykres skrzypcowy z boxplotem, używając `geom_boxplot` w wykresie (wynik będzie bardziej przejrzysty, jeśli wykorzystasz kolory).

Wizualizacja wielu zmiennych

Wykresy wykorzystywane do porównywania dwóch zmiennych — wykres punktowy, hexagon binning i boxploty — mogą być z łatwością rozszerzone na więcej zmiennych dzięki pojęciu **warunkowania** (ang. *conditioning*). Przykładem może być rysunek 1.8, na którym przedstawiono związek pomiędzy powierzchnią domów a opodatkowaniem. Natomiast na rysunku 1.12 uwzględniono wpływ lokalizacji poprzez przedstawienie danych z uwzględnieniem kodu pocztowego. W tym przypadku rysunek jest bardziej przejrzysty: opodatkowanie jest dla niektórych kodów (98105, 98126) znacznie wyższe niż dla innych (98108,98188). Ta różnica powoduje tworzenie klastrów obserwowanych na rysunku 1.8.



Rysunek 1.11. Wykres skrzypcowy dla odsetka lotów opóźnionych z winy przewoźnika

Rysunek 1.12 stworzyliśmy, wykorzystując pakiet `ggplot2` i ideę **aspektów** (ang. *facets*) lub warunkowania zmiennej (w tym przypadku jest to kod pocztowy):

```
ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
       aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(color='white') +
  theme_bw() +
  scale_fill_gradient(low='white', high='blue') +
  labs(x='Powierzchnia domu', y='Wartość podatku') +
  facet_wrap('ZipCode')    ❶
```

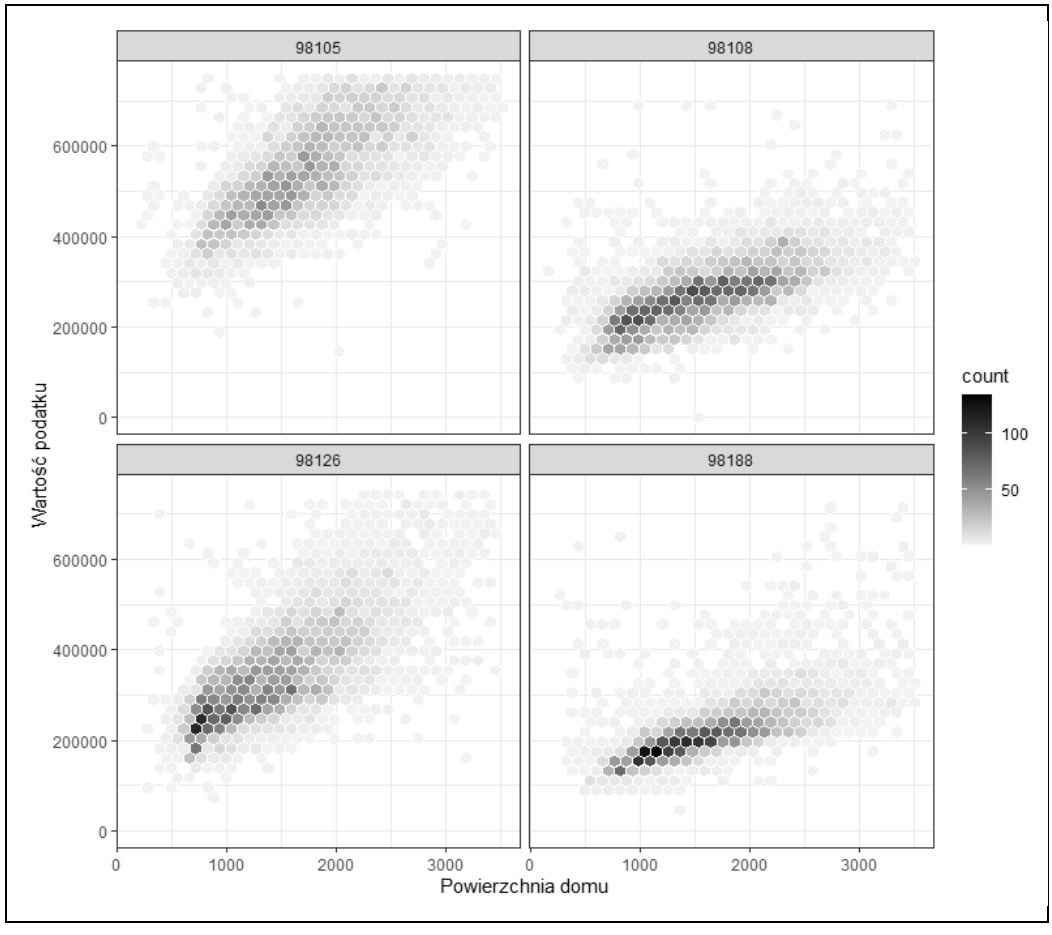
❶ Za pomocą funkcji `facet_wrap` i `facet_grid` z pakietu `ggplot` określamy warunkowanie zmiennej.

Wizualizacje większości pakietów Pythona bazują na możliwościach pakietu `Matplotlib`. Teoretycznie jest możliwe tworzenie wykresów aspektowych za pomocą pakietu `Matplotlib`, ale kod może być bardzo skomplikowany. Na szczęście prostsze, alternatywne rozwiązanie jest dostępne poprzez pakiet `seaborn`:

```
zip_codes = [98188, 98105, 98108, 98126]
kc_tax_zip = kc_tax0.loc[kc_tax0.ZipCode.isin(zip_codes),:]
kc_tax_zip

def hexbin(x, y, color, **kwargs):
    cmap = sns.light_palette(color, as_cmap=True)
    plt.hexbin(x, y, gridsize=25, cmap=cmap, **kwargs)

g = sns.FacetGrid(kc_tax_zip, col='ZipCode', col_wrap=2)    ❶
g.map(hexbin, 'SqFtTotLiving', 'TaxAssessedValue',
      extent=[0, 3500, 0, 700000])    ❷
g.set_axis_labels('Powierzchnia domu', 'Wartość podatku')
g.set_titles('Kod pocztowy {col_name:.0f}')
```



Rysunek 1.12. Wartość podatku a powierzchnia domu w zależności od kodu pocztowego

❶ Argumenty `col` i `row` służą do określania warunkowania zmiennych. W przypadku pojedynczej zmiennej można umieszczać wiele wykresów aspektowych na jednym rysunku za pomocą kombinacji argumentów `col` i `col_wrap`.

❷ Metoda `map` wywołuje funkcję `hexbin` wraz z podzbiorami pierwotnego zestawu danych dla różnych kodów pocztowych. Argument `extent` definiuje granice osi x i y.

Koncepcja warunkowania zmiennej w systemie graficznym została zapoczątkowana w *Trellis graphics* i rozwinięta przez Ricka Beckera, Billa Clevelanda i wiele innych osób z Bell Labs [Trellis Graphics]. Pomysł ten rozprzestrzenił się w postaci dodatków do tworzenia wykresów, takich jak pakiety R `lattice` [lattice] i `ggplot2` lub moduły Pythona `seaborn` [seaborn] i `Bokeh` [bokeh]. Warunkowanie zmiennych jest także zintegrowane z platformami Business Intelligence, takimi jak Tableau i Spotfire. Wraz z nastaniem ery superszybkich komputerów współczesne platformy do wizualizacji osiągnęły znacznie więcej, niż przewidywano na początku badań eksploracyjnych. Mimo to kluczowe założenia i narzędzia wymyślone pół wieku temu (np. proste wykresy pudełkowe) nadal stanowią podstawy tych systemów.

Główne zasady

- Wykresy hexagon binning i konturowe są pomocnymi narzędziami, które umożliwiają graficzne badanie dwóch zmiennych numerycznych naraz bez przytłaczania ogromną ilością danych.
- Tablica kontyngencji jest typowym narzędziem do badania liczności dwóch zmiennych skategoryzowanych.
- Boxploty i wykresy skrzypcowe umożliwiają zobrazowanie zmiennych numerycznych względem zmiennych skategoryzowanych.

Dla pogłębienia wiedzy

- Bardzo dobre omówienie *grammar for graphics* (gramatyki dla grafiki, w `ggplot` — „gg") to: Benjamin Baumer, Daniel Kaplan i Nicholas Horton, *Modern Data Science with R*, Chapman & Hall/CRC Press, 2017.
- Bardzo dobra dokumentacja to: Hadley Wickham, *ggplot2: Elegant Graphics for Data Analysis*, Springer, 2009.
- Internetowy przewodnik po ggplot2 autorstwa Josefa Fruehwalda znajduje się pod adresem: *http://www.ling.upenn.edu/~joseff/avml2012/*.

Podsumowanie

Eksploracyjna analiza danych (**EDA**, ang. *exploratory data analysis*), zapoczątkowana przez Johna Tukeya, stworzyła podstawy, które stały się początkami data science. EDA zakłada, że w każdym projekcie opartym na danych pierwszy i najważniejszy krok to **przyjrzenie się danym**. Opisując i wizualizując dane, możesz zyskać wartościowe informacje i zrozumieć projekt.

Niniejszy rozdział omawia wiele rozmaitych zagadnień, począwszy od prostych metryk położenia i rozproszenia, skończywszy na graficznym przedstawieniu zależności pomiędzy wieloma zmiennymi, jak pokazano na rysunku 1.12. Społeczność wolnego oprogramowania stworzyła zróżnicowany zestaw narzędzi i technik. W połączeniu z możliwościami języków R i Python dało to ogromny potencjał w zakresie badania i analizowania danych. Badania eksploracyjne powinny być podstawą każdego dowolnego projektu data science.

Statystyka: klasyczne narzędzia w najnowszych technologiach!

Metody statystyczne są kluczowym narzędziem w data science, mimo to niewielu analityków danych zdobyło wykształcenie w ich zakresie. Może im to utrudniać uzyskiwanie dobrych efektów. Zrozumienie praktycznych zasad statystyki okazuje się ważne również dla programistów R i Pythona, którzy tworzą rozwiązania dla data science. Kursy podstaw statystyki rzadko jednak uwzględniają tę perspektywę, a większość podręczników do statystyki w ogóle nie zajmuje się narzędziami wywodzącymi się z informatyki.

To drugie wydanie popularnego podręcznika statystyki przeznaczonego dla analityków danych. Uzupełniono je o obszerne przykłady w Pythonie oraz wyjaśnienie, jak stosować poszczególne metody statystyczne w problemach data science, a także jak ich nie używać. Skoncentrowano się też na tych zagadnieniach statystyki, które odgrywają istotną rolę w data science. Wyjaśniono, które koncepcje są ważne i przydatne z tej perspektywy, a które mniej istotne i dlaczego. Co ważne, poszczególne koncepcje i zagadnienia praktyczne przedstawiono w sposób przyswajalny i zrozumiały również dla osób nienawykłych do posługiwania się statystyką na co dzień.

W książce między innymi:

- analiza eksploracyjna we wstępnym badaniu danych
- próby losowe a jakość dużych zbiorów danych
- podstawy planowania eksperymentów
- regresja w szacowaniu wyników i wykrywaniu anomalii
- statystyczne uczenie maszynowe
- uczenie nienadzorowane a znaczenie danych niesklasyfikowanych

Peter Bruce jest ekspertem w dziedzinie nauczania statystyki. Prowadzi Institute for Statistics Education, gdzie oferuje setki kursów skierowanych między innymi do naukowców.
Dr Andrew Bruce jest głównym analitykiem w Amazonie. Od trzydziestu lat zajmuje się statystyką i nauką o danych — opracowuje rozwiązania problemów z wielu branż.
Dr Peter Gedeck jest badaczem w Collaborative Drug Discovery. Tworzy algorytmy uczenia maszynowego do przewidywania właściwości substancji stanowiących potencjalne leki.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-283-7427-0
9 788328 374270
INFORMATYKA W NAJLEPSZYM WYDANIU
Cena: 69,00 zł