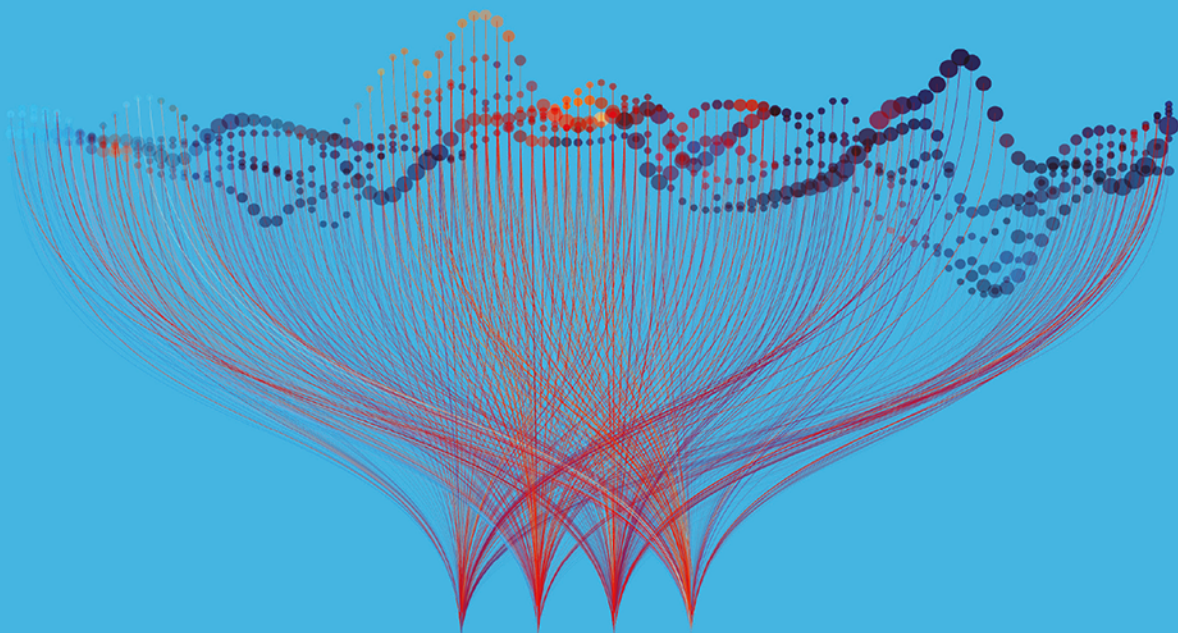


DATA SCIENCE_____

Programowanie, analiza i wizualizacja
danych z wykorzystaniem języka R



MICHAEL FREEMAN | JOEL ROSS

Tytuł oryginału: Programming Skills for Data Science: Start Writing Code to Wrangle, Analyze, and Visualize Data with R

Tłumaczenie: Tomasz Walczak

ISBN: 978-83-283-5782-2

Authorized translation from the English language edition, entitled PROGRAMMING SKILLS FOR DATA SCIENCE: START WRITING CODE TO WRANGLE, ANALYZE, AND VISUALIZE DATA WITH R, 1st Edition by FREEMAN, MICHAEL; ROSS, JOEL, published by Pearson Education, Inc, publishing as Addison-Wesley Professional, Copyright © 2019 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. POLISH language edition published by HELION SA, Copyright © 2020.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

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/datasi>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/datasi.zip>

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

Spis treści

Przedmowa	11
Wprowadzenie	13
O autorach	19
CZĘŚĆ I Wprowadzenie	21
Rozdział 1 Przygotowywanie komputera	23
1.1. Przygotowywanie narzędzi używanych w wierszu poleceń	24
1.1.1. Wiersz poleceń w systemie macOS	24
1.1.2. Wiersz poleceń w systemie Windows	25
1.1.3. Wiersz poleceń w systemie Linux	25
1.2. Instalowanie systemu git	25
1.3. Tworzenie konta w serwisie GitHub	26
1.4. Wybieranie edytora tekstu	26
1.4.1. Atom	26
1.4.2. Visual Studio Code	27
1.4.3. Sublime Text	27
1.5. Pobieranie języka R	28
1.6. Pobieranie środowiska RStudio	28
Rozdział 2 Używanie wiersza poleceń	31
2.1. Uruchamianie wiersza poleceń	31
2.2. Poruszanie się w systemie plików	32
2.2.1. Przechodzenie do innego katalogu	33
2.2.2. Wyświetlanie listy plików	35
2.2.3. Ścieżki	35
2.3. Zarządzanie plikami	37
2.3.1. Uczenie się nowych instrukcji	37
2.3.2. Symbole wieloznaczne	39
2.4. Radzenie sobie z błędami	40
2.5. Przekierowywanie danych wyjściowych	41
2.6. Polecenia związane z siecią	42

CZĘŚĆ II Zarządzanie projektami 45**Rozdział 3 Kontrola wersji z użyciem systemu git i serwisu GitHub 47**

3.1. Czym jest git?	47
3.1.1. Podstawowe zagadnienia związane z systemem git	48
3.1.2. Czym jest GitHub?	49
3.2. Konfigurowanie narzędzi i tworzenie projektu	50
3.2.1. Tworzenie repozytorium	51
3.2.2. Sprawdzanie stanu	51
3.3. Śledzenie zmian w projekcie	52
3.3.1. Dodawanie plików	53
3.3.2. Zatwierdzanie	54
3.3.3. Proces używania systemu git	55
3.4. Zapisywanie projektów w witrynie GitHub	56
3.4.1. Forki i klonowanie	57
3.4.2. Wysyłanie i pobieranie	59
3.5. Dostęp do historii projektu	60
3.5.1. Historia rewizji	60
3.5.2. Powrót do starszych wersji	61
3.6. Ignorowanie plików w projekcie	62

Rozdział 4 Tworzenie dokumentacji za pomocą języka Markdown 65

4.1. Pisanie kodu w języku Markdown	65
4.1.1. Formatowanie tekstu	66
4.1.2. Bloki tekstu	66
4.1.3. Hiperłącza	67
4.1.4. Rysunki	67
4.1.5. Tabele	68
4.2. Wyświetlanie dokumentów w języku Markdown	68

CZĘŚĆ III Podstawowe umiejętności z zakresu języka R 71**Rozdział 5 Wprowadzenie do języka R 73**

5.1. Programowanie z użyciem języka R	73
5.2. Uruchamianie kodu w języku R	74
5.2.1. Używanie środowiska RStudio	74
5.2.2. Używanie języka R w wierszu poleceń	76
5.3. Dodawanie komentarzy	78
5.4. Definiowanie zmiennych	78
5.4.1. Podstawowe typy danych	80
5.5. Szukanie pomocy	83
5.5.1. Nauka uczenia się języka R	84

Rozdział 6 Funkcje 89

6.1. Czym jest funkcja?	89
6.1.1. Składnia funkcji w języku R	90
6.2. Wbudowane funkcje języka R	91
6.2.1. Argumenty nazwane	92

6.3. Wczytywanie funkcji	93
6.4. Pisanie funkcji	95
6.4.1. Debugowanie funkcji	97
6.5. Instrukcje warunkowe	98
Rozdział 7 Wektory	101
7.1. Czym jest wektor?	101
7.1.1. Tworzenie wektorów	101
7.2. Operacje wektorowe	102
7.2.1. Ponowne używanie elementów	104
7.2.2. Prawie wszystko jest wektorem	105
7.2.3. Funkcje wektorowe	105
7.3. Indeksy w wektorach	107
7.3.1. Listy indeksów	108
7.4. Filtrowanie wektorów	109
7.5. Modyfikowanie wektorów	110
Rozdział 8 Listy	113
8.1. Czym jest lista?	113
8.2. Tworzenie list	114
8.3. Dostęp do elementów listy	115
8.4. Modyfikowanie list	117
8.4.1. Pojedyncze i podwójne nawiasy kwadratowe	118
8.5. Stosowanie funkcji do list za pomocą wywołania lapply()	119
CZĘŚĆ IV Przekształcanie danych	121
Rozdział 9 Jak zrozumieć dane?	123
9.1. Proces generowania danych	123
9.2. Wyszukiwanie danych	124
9.3. Rodzaje danych	126
9.3.1. Skale pomiarowe	126
9.3.2. Struktury danych	127
9.4. Interpretowanie danych	129
9.4.1. Zdobywanie wiedzy w danej dziedzinie	129
9.4.2. Jak zrozumieć schematy danych?	131
9.5. Odpowiadanie na pytania na podstawie danych	133
Rozdział 10 Ramki danych	135
10.1. Czym jest ramka danych?	135
10.2. Praca z ramkami danych	136
10.2.1. Tworzenie ramek danych	136
10.2.2. Opisywanie struktury ramek danych	137
10.2.3. Dostęp do ramek danych	138
10.3. Praca z danymi CSV	139
10.3.1. Katalog roboczy	140
10.3.2. Zmienne w postaci czynników	142

Rozdział 11 Operowanie danymi za pomocą pakietu dplyr	145
11.1. Gramatyka operowania danymi	145
11.2. Podstawowe funkcje pakietu dplyr	146
11.2.1. Pobieranie (funkcja select())	147
11.2.2. Filtrowanie (funkcja filter())	149
11.2.3. Dodawanie kolumn (funkcja mutate())	150
11.2.4. Sortowanie danych (funkcja arrange())	151
11.2.5. Tworzenie podsumowań (funkcja summarize())	152
11.3. Wykonywanie operacji sekwencyjnych	153
11.3.1. Operator potoku	154
11.4. Analizowanie ramek danych z wykorzystaniem grupowania	155
11.5. Złączanie ramek danych	157
11.6. Pakiet dplyr w praktyce — analizowanie danych na temat lotów ..	162
Rozdział 12 Porządkowanie danych za pomocą pakietu tidyr	169
12.1. Czym jest porządkowanie danych?	169
12.2. Od kolumn do wierszy — gather()	171
12.3. Z wierszy na kolumny — spread()	172
12.4. Pakiet tidyr w praktyce — eksplorowanie statystyki na temat edukacji	174
Rozdział 13 Dostęp do bazy danych	181
13.1. Przegląd relacyjnych baz danych	181
13.1.1. Czym jest relacyjna baza danych?	182
13.1.2. Tworzenie relacyjnej bazy danych	184
13.2. Wstęp do języka SQL	185
13.3. Dostęp do bazy danych w języku R	189
Rozdział 14 Używanie internetowych interfejsów API	193
14.1. Czym jest internetowy interfejs API?	193
14.2. Żądania REST	194
14.2.1. Identyfikatory URI	194
14.2.2. Operacje (czasowniki) z protokołu HTTP	201
14.3. Dostęp do internetowych interfejsów API w R	201
14.4. Przetwarzanie danych w formacie JSON	203
14.4.1. Przetwarzanie danych w formacie JSON	205
14.4.2. Spłaszczanie danych	207
14.5. Interfejsy API w praktyce — znajdowanie kubańskiego jedzenia w Seattle	209
CZĘŚĆ V Wizualizacje danych	215
Rozdział 15 Projektowanie wizualizacji danych	217
15.1. Cel wizualizacji	217
15.2. Wybieranie układu graficznego	219
15.2.1. Wizualizowanie jednej zmiennej	220
15.2.2. Wizualizowanie wielu zmiennych	223
15.2.3. Wizualizowanie danych hierarchicznych	227

15.3. Wybieranie skutecznego kodowania graficznego	229
15.3.1. Skuteczne kolory	231
15.3.2. Wykorzystanie atrybutów przeduwagowych	234
15.4. Ekspresywne prezentacje danych	236
15.5. Zwiększanie estetyki	238
Rozdział 16 Tworzenie wizualizacji za pomocą pakietu ggplot2	241
16.1. Gramatyka grafiki	241
16.2. Tworzenie podstawowych wykresów za pomocą ggplot2	242
16.2.1. Określanie obiektów geometrycznych	245
16.2.2. Odwzorowania aspektów estetycznych	247
16.3. Złożone układy i dostosowywanie opcji	248
16.3.1. Dostosowywanie pozycji	248
16.3.2. Zmianianie stylu za pomocą skal	250
16.3.3. Układ współrzędnych	253
16.3.4. Fasety	254
16.3.5. Etykiety i uwagi	255
16.4. Tworzenie map	257
16.4.1. Kartogramy	258
16.4.2. Mapy punktowe	261
16.5. Pakiet ggplot2 w praktyce — mapa eksmisji w San Francisco	262
Rozdział 17 Interaktywne wizualizacje w języku R	267
17.1. Pakiet plotly	269
17.2. Pakiet rbokeh	271
17.3. Pakiet leaflet	273
17.4. Interaktywne wizualizacje w praktyce — analizowanie zmian w Seattle	276
CZĘŚĆ VI Tworzenie i udostępnianie aplikacji	281
Rozdział 18 Tworzenie dynamicznych raportów za pomocą platformy R Markdown	283
18.1. Konfigurowanie raportu	283
18.1.1. Tworzenie plików .rmd	284
18.1.2. Kompilowanie dokumentów	286
18.2. Integrowanie tekstu w formacie Markdown i kodu w języku R	287
18.2.1. Wykonywalne fragmenty kodu w języku R	287
18.2.2. Kod wewnątrzwierszowy	288
18.3. Wyświetlanie danych i wizualizacji w raportach	289
18.3.1. Wyświetlanie łańcuchów znaków	289
18.3.2. Wyświetlanie list w formacie Markdown	290
18.3.3. Wyświetlanie tabel	291
18.3.4. Wyświetlanie wykresów	292
18.4. Udostępnianie raportów jako stron internetowych	293
18.5. Platforma R Markdown w praktyce — raport na temat oczekiwanej długości życia	295

Rozdział 19 Tworzenie interaktywnych aplikacji internetowych za pomocą platformy Shiny	301
19.1. Platforma Shiny	301
19.1.1. Podstawowe zagadnienia dotyczące platformy Shiny	302
19.1.2. Struktura aplikacji	303
19.2. Projektowanie interfejsów użytkownika	307
19.2.1. Treści statyczne	308
19.2.2. Dynamiczne dane wejściowe	310
19.2.3. Dynamiczne dane wyjściowe	311
19.2.4. Układy	312
19.3. Tworzenie serwerów aplikacji	315
19.4. Publikowanie aplikacji na platformę Shiny	318
19.5. Platforma Shiny w praktyce — wizualizacja śmiertelnych postrzeżeń przez policję	320
Rozdział 20 Praca zespołowa	327
20.1. Śledzenie różnych wersji kodu za pomocą gałęzi	327
20.1.1. Praca z różnymi gałęziami	329
20.1.2. Scalanie gałęzi	332
20.1.3. Scalanie a konflikty	333
20.1.4. Scalanie w serwisie GitHub	335
20.2. Prowadzenie projektów z użyciem gałęzi funkcji	337
20.3. Współpraca w ramach scentralizowanego procesu pracy	338
20.3.1. Tworzenie centralnego repozytorium	339
20.3.2. Używanie gałęzi funkcji w scentralizowanym procesie pracy	341
20.4. Współpraca w procesie pracy z użyciem forków	342
Rozdział 21 Dalsza nauka	347
21.1. Uczenie statystyczne	347
21.1.1. Ocena zależności	347
21.1.2. Prognozowanie	348
21.2. Inne języki programowania	348
21.3. Odpowiedzialność etyczna	349
Skorowidz	351

Jak zrozumieć dane?

We wcześniejszych rozdziałach opisano podstawy programowania na potrzeby pracy z danymi. Szczegółowo wyjaśniono, jak nakazać komputerowi przetwarzanie danych. Aby posłużyć się komputerem do analizy danych, potrzeba *uzyskać dostęp* do zbioru danych i *zinterpretować* je, aby móc zadawać na ich temat sensowne pytania. Dzięki temu możesz przekształcić surowe dane w informacje umożliwiające podejmowanie działań (ang. *actionable information*).

Ten rozdział zawiera ogólny przegląd pokazujący, jak interpretować zbiory danych w kontekście nauki o danych. Szczegółowo opisano tu źródła danych, formaty danych i strategie określania zadawanych pytań. Opracowanie przejrzystego modelu umysłowego znaczenia wartości w zbiorze danych jest niezbędne do zaprogramowania komputera na potrzeby skutecznej analizy danych.

9.1. Proces generowania danych

Zanim zaczniesz pracę z danymi, ważne jest, abyś zrozumiał, *skąd one pochodzą*. Istnieje wiele procesów rejestrowania zdarzeń jako danych. Każdy z tych procesów ma określone ograniczenia i wymaga przyjęcia pewnych założeń. Oto podstawowe kategorie technik rejestrowania danych:

- **Czujniki.** W ostatniej dekadzie ilość danych rejestrowanych przez czujniki gwałtownie wzrosła. Czujniki automatycznie odbierające i rejestrujące informacje, np. czujniki zanieczyszczenia badające jakość powietrza, trafiają do obszaru przetwarzania prywatnych danych (pomyśl o opaskach FitBit lub innych krokomierzach). Jeśli takie urządzenia zostały poprawnie skalibrowane, stanowią niezawodny i spójny mechanizm zbierania danych.
- **Ankiety.** Dane, które trudniej jest mierzyć z zewnątrz, np. opinie lub przeżycia ludzi, można zbierać za pomocą *ankiet*. Ponieważ ankiety są oparte na samoopisie, jakość danych może być różna (w zależności od ankiety i poszczególnych ankietowanych). W niektórych dziedzinach ludzie mogą słabo pamiętać opisywane zagadnienia (np. to, co jedli w zeszłym tygodniu). Czasem mają też motywację do tego, by odpowiadać w konkretny sposób (np. przypisywać sobie więcej prozdrowotnych zachowań, niż rzeczywiście przejawiają). Należy wykrywać tego rodzaju błędy systematyczne i jeśli to możliwe, uwzględnić je w analizach.
- **Przechowywanie rejestrów.** W wielu dziedzinach organizacje stosują automatyczne i ręczne procesy rejestrowania działań. Na przykład szpital może rejestrować długość

i skutki każdej przeprowadzanej operacji (a jednostki rządowe mogą wymagać od szpitala raportów na ten temat). Rzetelność takich danych zależy od jakości systemów używanych do ich generowania. Także eksperymenty naukowe wymagają sumiennego rejestrowania wyników.

- **Wtórne analizy danych.** Dane można też kompilować na podstawie *istniejących zbiorów wiedzy* lub pomiarów. Można np. zliczać wystąpienia słów w tekstach historycznych (komputery mogą w tym pomóc!).

Wszystkie te metody zbierania danych mogą prowadzić do problemów i błędów systematycznych. Na przykład czujniki mogą być nieprecyzyjne, ludzie mogą prezentować się w określony sposób w ankietach, rejestry mogą uwzględniać tylko konkretne zadania, a istniejące zbiory wiedzy mogą nie zawierać niektórych punktów widzenia. Gdy pracujesz z jakimś zbiorem danych, koniecznie należy rozważyć, skąd dane pochodzą (*kto je zarejestrował, jak i dlaczego*), aby móc je skutecznie i sensownie analizować.

9.2. Wyszukiwanie danych

Możliwość rejestrowania i przechowywania danych w komputerach doprowadziła do gwałtownego wzrostu ilości danych, jakie można analizować — od osobistych pomiarów biologicznych (*ile kroków zrobiłem?*), przez struktury sieci społecznościowych (*kim są moi znajomi?*), po prywatne informacje, które wyciekły z niezabezpieczonych witryn i agencji rządowych (*jakie są numery PESEL użytkowników?*). W pracy zapewne będziesz korzystał z wewnętrznych danych zbieranych lub zarządzanych przez Twoją firmę. Mogą to być bardzo różnorodne dane: od zamówień w kawiarni fair trade po wyniki badań medycznych. Takie dane mogą być tak różne jak rodzaje firm, ponieważ obecnie *każdy* rejestruje dane i dostrzega wartość ich analizy.

Na szczęście istnieje jest też mnóstwo bezpłatnych, powszechnie dostępnych zbiorów danych, z którymi można pracować. Organizacje często udostępniają publicznie duże ilości danych, aby umożliwić powtarzanie eksperymentów, promować przejrzystość lub po prostu sprawdzać, jak inni potrafią wykorzystać określone dane. Takie zbiory danych doskonale nadają się do budowania umiejętności i portfolio w zakresie nauki o danych. Dane te są dostępne w różnych formatach, np. jako arkusze CSV (zobacz rozdział 10.), relacyjne bazy danych (zobacz rozdział 13.) i interfejsy API usług sieciowych (zobacz rozdział 14.).

Oto popularne źródła publicznie dostępnych zbiorów danych:

- **Publikacje rządowe.** Organizacje rządowe (i inne systemy biurokratyczne) w ramach codziennej działalności generują *mnóstwo* danych i często udostępniają je, aby uchodzić za przejrzyste i odpowiedzialne instytucje przed opinią publiczną. Obecnie dane są publicznie dostępne w wielu krajach, na przykład w Stanach Zjednoczonych¹, Kanadzie², Indiach³ itd. Także lokalne instytucje udostępniają dane. Na przykład miasto Seattle⁴ udostępnia duże ilości danych w łatwo dostępnym formacie. Dane rządowe dotyczą

¹ **Publiczne dane rządu Stanów Zjednoczonych:** <https://www.data.gov>.

² **Publiczne dane rządu Kanady:** <https://open.canada.ca/en/open-data>.

³ **Platforma publicznych danych rządu Indii:** <https://data.gov.in>.

⁴ **Portal publicznych danych miasta Seattle:** <https://data.seattle.gov>.

różnych zagadnień, przy czym wpływ na treść danych może mieć sytuacja polityczna związana z ich zbieraniem i przechowywaniem.

- **Wiadomości i dzienniki.** Dziennikarstwo pozostaje jednym z najważniejszych kontekstów, w jakich dane są zbierane i analizowane. Dziennikarze wykonują dużo czarnej roboty w obszarze generowania danych — przeszukują istniejące źródła wiedzy, przepytują ludzi, przeprowadzają ankiety oraz w inny sposób ujawniają i łączą wcześniej ukryte lub ignorowane informacje. Media informacyjne zwykle publikują przeanalizowane, podsumowujące informacje do użytku odbiorców, jednak czasem udostępniają też źródła danych, aby inni mogli potwierdzać i rozwijać wnioski z analiz. Na przykład *New York Times*⁵ udostępnia dużą część danych historycznych w usługach sieciowych, a poświęcony danym politycznym blog *FiveThirtyEight*⁶ udostępnia w serwisie GitHub wszystkie dane (błędne modele i inne informacje), których dotyczą publikowane na tym blogu artykuły.
- **Badania naukowe.** Innym doskonałym źródłem danych są badania naukowe prowadzone na uczelniach lub przez instytucje komercyjne. Badania naukowe są (teoretycznie) dobrze uzasadnione i ustrukturyzowane oraz zapewniają sensowne dane, jeśli są one analizowane w odpowiednim dla nich kontekście. Ponieważ badania muszą być upowszechniane, a następnie sprawdzane przez inne osoby, aby były użyteczne, często są publicznie udostępniane na potrzeby analiz i krytyki. Niektóre czasopisma naukowe, np. poważany magazyn *Nature*, wymagają, aby autorzy udostępniali dane innym do sprawdzenia (zapoznaj się z listą repozytoriów danych naukowych tego magazynu⁷).
- **Sieci i media społecznościowe.** Internet jest jednym ze źródeł, gdzie generowane są największe ilości danych. Dane są tu rejestrowane automatycznie w wyniku użytkowania aplikacji społecznościowych takich jak Facebook, Twitter i Google oraz interakcji w tych narzędziach. Aby lepiej zintegrować swoje usługi z codziennym życiem użytkowników, firmy zarządzające mediami społecznościowymi udostępniają swoje dane programowo innym programistom. Możesz np. uzyskać dostęp na żywo do danych z Twittera⁸, które są używane do różnych ciekawych analiz. Także Google zapewnia programowy dostęp⁹ do wielu swoich usług (w tym do wyszukiwarki i serwisu YouTube).
- **Społeczności internetowe.** Wraz z szybkim wzrostem popularności nauki o danych rozwija się też społeczność specjalistów z tej dziedziny. Ta społeczność i używane przez nią serwisy internetowe są następnym doskonałym źródłem ciekawych i różnorodnych zbiorów danych oraz analiz. Na przykład serwis *Kaggle*¹⁰ udostępnia wiele zbiorów

⁵ Serwis dla programistów prowadzony przez dziennik *New York Times*: <https://developer.nytimes.com>.

⁶ *FiveThirtyEight* — Our Data: <https://data.fivethirtyeight.com>.

⁷ *Nature* — polecane repozytoria danych: <https://www.nature.com/sdata/policies/repositories>.

⁸ Platforma dla programistów w serwisie Twitter: <https://developer.twitter.com/en/docs>.

⁹ Przegląd interfejsów API Google'a: <https://developers.google.com/apis-explorer/>.

¹⁰ *Kaggle* — the home of data science and machine learning: <https://www.kaggle.com>.

danych, a także zadań związanych z ich analizą. Serwis *Socrata*¹¹ (gdzie znajduje się repozytorium danych miasta Seattle) także rejestruje różnorodne zbiory danych (często od jednostek komercyjnych i rządowych). Nieco podobny serwis *UCI Machine Learning Repository*¹² zawiera zbiory danych używanych do uczenia maszynowego. Pochodzą one głównie ze źródeł akademickich. Istnieje też wiele innych internetowych list źródeł danych, w tym specjalna lista dyskusyjna w serwisie Reddit — */r/Datasets*¹³.

Dostępnych jest więc mnóstwo pochodzących z rzeczywistego świata zbiorów danych, nad którymi możesz pracować — niezależnie od tego, czy szukasz odpowiedzi na konkretne pytania, czy chcesz tylko eksplorować dane lub szukasz pomysłów.

9.3. Rodzaje danych

Gdy masz już zbiór danych, musisz zrozumieć jego strukturę i zawartość, zanim zaczniesz go (programowo) badać. Zrozumienie *rodzajów* danych zależy od umiejętności rozróżniania skal pomiarowych, a także różnych struktur używanych do przechowywania danych.

9.3.1. Skale pomiarowe

Dane mogą obejmować różne rodzaje wartości (reprezentowane w R za pomocą typów danych). Na ogólnym poziomie wartości można omawiać w kategoriach **skal pomiarowych**¹⁴ — sposobu klasyfikowania wartości danych w kategoriach tego, jak można je mierzyć i jak porównywać je z innymi wartościami.

W dziedzinie statystyki wartości są zwykle dzielone na cztery kategorie opisane w tabeli 9.1.

Dane nominalne nie mają określonego uporządkowania. Nie można np. powiedzieć, że „jabłka są bardziej niż pomarańcze”, choć można stwierdzić, czy dany owoc jest jabłkiem lub pomarańczą. Dane nominalne często służą do określania, że dana obserwacja należy do konkretnej kategorii lub grupy. Na danych nominalnych zwykle nie przeprowadza się analiz matematycznych (np. nie da się ustalić „średniej” na podstawie owoców). Można jednak analizować liczby wystąpień lub rozkłady. Dane nominalne można reprezentować za pomocą łańcuchów znaków (np. nazw owoców), ale też liczb (np. „pierwszy rodzaj owoców”, „drugi rodzaj owoców” itd.). Jednak to, że wartość w zbiorze danych jest liczbą, nie oznacza, że możesz wykonywać operacje matematyczne. Zwróć też uwagę, że wartości logiczne (TRUE i FALSE) są wartościami nominalnymi.

W **danych porządkowych** określona jest *kolejność* wartości. Takie dane można stosować do kategoryzowania, ale też do określania, że niektóre grupy są *większe* lub *mniej* od innych. Na przykład hotele lub restauracje można sklasyfikować jako *pięciogwiazdkowe*, *czterogwiazdkowe* itd. W takich danych występuje uporządkowanie, jednak odległości między wartościami mogą być różne. Możliwe jest tu wyznaczenie wartości minimalnej, maksymalnej, a nawet mediany,

¹¹ *Socrata* — platforma typu „dane jako usługa”: <https://opendata.socrata.com>.

¹² *UCI Machine Learning Repository*: <https://archive.ics.uci.edu/ml/index.php>.

¹³ */r/DataSets*: <https://www.reddit.com/r/datasets/>.

¹⁴ Stevens, S.S. (1946). On the theory of scales of measurement. *Science*, 103(2684), 677 – 680. <https://doi.org/10.1126/science.103.2684.677>.

Tabela 9.1. Skale pomiarowe

Skala	Przykład	Operacje
Nominalna Bez kolejności, służy do klasyfikowania.	Owoce: jabłka, banany, pomarańcze itd.	=, != „takie same lub różne”
Porządkowa Kolejność, można sortować.	Oceny hoteli: pięciogwiazdkowe, czterogwiazdkowe itd.	=, !=, <, > „takie same lub różne”
Ilorazowa (stosunkowa) Kolejność, ustalony punkt zero.	Długości: 1 cal, 1,5 cala, 2 cale itd.	=, !=, <, >, +, -, *, / „dwa razy takie”
Interwałowa (przedziałowa) Kolejność, bez ustalonego punktu zero.	Daty: 15/05/2012, 17/04/2015 itd.	=, !=, <, >, +, - „trzy jednostki większe”

jednak nie da się obliczyć średniej statystycznej (ponieważ wartości porządkowe nie definiują, o ile jedna wartość jest większa od innej). Zauważ, że zmienne nominalne można traktować jak porządkowe, jeśli wyznaczona zostanie kolejność. Skutkuje to jednak zmianą skali pomiarowej danych. Na przykład kolory tworzą zwykle skalę *nominalną*. Nie można stwierdzić, że „czerwony jest większy od niebieskiego”. Jest tak mimo zwyczajowego porządkowania opartego na kolorach tęczy. Gdy mówisz, że „czerwony występuje przed niebieskim (w tęczy)”, zastępujesz nominalną skalę kolorów skalą porządkową, reprezentującą *pozycje kolorów w tęczy* (które same zależą od *stosunku* długości fal). Dane porządkowe są jednocześnie danymi nominalnymi.

Skala ilorazowa (czasem nazywana **stosunkową**) to najczęściej używana w praktyce skala pomiarowa. Dane dotyczące liczby wystąpień w populacji, wartości pieniężnych lub ilości aktywności są zwykle mierzone z użyciem skali ilorazowej. Dla takich danych można obliczać średnie, a także mierzyć odległości między różnymi wartościami (pomiar odległości jest możliwy także dla danych interwałowych). Zgodnie z oczekiwaniami dane ilorazowe umożliwiają określanie stosunku dwóch wartości (np. wartość x jest dwa razy większa od wartości y). Dane ilorazowe są uznawane za ciągłe.

Skala interwałowa (inaczej **przedziałowa**) jest podobna do ilorazowej, jednak nie występuje tu ustalony punkt zero. Na przykład dat nie można analizować w ujęciu *proporcjonalnym* (nie możesz np. stwierdzić, że *środa* jest dwa razy większa od *poniedziałku*). Możliwe jest obliczenie odległości (interwału) między dwoma wartościami (np. *dwa dni*), jednak nie da się ustalić *stosunku* tych wartości. Dane interwałowe także są uznawane za ciągłe.

Identyfikowanie i rozumienie skal pomiarowych dla określonych atrybutów danych jest ważne, gdy rozważasz, jak analizować zbiór danych. Przede wszystkim musisz wiedzieć, jakiego rodzaju analizy statystyczne będą poprawne dla określonych danych. Musisz też wiedzieć, jak interpretować zjawiska mierzone za pomocą tych danych.

9.3.2. Struktury danych

W praktyce będziesz musiał porządkować opisane w poprzednich rozdziałach liczby, łańcuchy znaków, wektory i listy wartości z wykorzystaniem bardziej złożonych formatów. Dane są porządkowane za pomocą pomocnych **struktur** (zwłaszcza gdy zbiór danych jest duży), aby lepiej

pokazać, co reprezentują określone liczby i łańcuchy znaków. Na potrzeby pracy z danymi z rzeczywistego świata musisz zrozumieć takie struktury i terminologię używaną do ich opisu.

W praktyce większość zbiorów danych jest przechowywana jako **tabela** z informacjami, a poszczególne wartości danych są uporządkowane w *wiersze* i *kolumny* (zobacz rysunek 9.1). Te tabele przypominają arkusze kalkulacyjne z programów takich jak Microsoft Excel. W tabeli każdy wiersz reprezentuje **rekord** lub **obserwację** — wystąpienie jednej mierzonej rzeczy, np. osoby lub spotkania sportowego. Każda kolumna reprezentuje **atrybut** — określoną właściwość lub aspekt mierzonej rzeczy, np. wagę lub wzrost człowieka albo wynik meczu. Każdą wartość danych można powiązać z **komórką** takiej tabeli.

	▲ name	height	weight
1	Ada	64	135
2	Bartek	74	156
3	Cyryl	69	139
4	Daria	69	144
5	Ela	71	152

	▲ name	height	weight
1	Ada	64	135
2	Bartek	74	156
3	Cyryl	69	139
4	Daria	69	144
5	Ela	71	152

Diagram illustrating the relationship between the table and its components:

- An arrow labeled "Atrybut" points to the "height" column header.
- An arrow labeled "Rekord lub obserwacja" points to the first row (index 1).

Rysunek 9.1. Tabela danych (wagi i wzrostu osób). Wiersze reprezentują obserwacje, natomiast kolumny odpowiadają atrybutom

W tym ujęciu tabela jest kolekcją mierzonych rzeczy. Dla każdej cechy danej rzeczy określona jest konkretna wartość. Ponieważ wszystkie obserwacje mają te same cechy (atrybuty), można analizować je porównawczo. Ponadto dzięki uporządkowaniu danych w tabeli każdej wartości (komórce) można automatycznie nadać dwa znaczenia: z której obserwacji pochodzi i jaki atrybut reprezentuje. Ta struktura umożliwi wydobycie znaczenia semantycznego na podstawie liczb. Liczba 64 na rysunku 9.1 nie jest przypadkową wartością — to „wzrost Ady”.

Tabela z rysunku 9.1 reprezentuje *niewielki* (a nawet *malutki*) zbiór danych, obejmujący tylko pięć obserwacji (wierszy). Wielkość zbioru danych jest zwykle mierzona na podstawie liczby obserwacji. Niewielki zbiór danych obejmuje kilkadziesiąt obserwacji, natomiast w dużym zbiorze mogą znajdować się tysiące lub setki tysięcy rekordów. Pojęcie „big data” oznacza zbiory danych, które są tak duże, że nie można ich wczytać do pamięci komputera bez specjalnych zabiegów. Takie zbiory mogą obejmować miliardy, a nawet tryliony wierszy. Jednak nawet

zbiory danych o stosunkowo niewielkiej liczbie obserwacji mogą zawierać dużą liczbę komórek, jeśli dla każdej obserwacji rejestrowanych jest wiele atrybutów (przy czym takie tabele często można „obrócić”, aby wierszy było więcej, a kolumn mniej; zobacz rozdział 12.). Liczba obserwacji i atrybutów (wierszy i kolumn) wyznacza **wymiary** zbioru danych. Nie należy mylić tych wymiarów z „dwuwymiarową” strukturą danych tabeli (struktura jest dwuwymiarowa, ponieważ każda wartość oznacza *dwie* rzeczy: obserwację i atrybut).

Choć dane często są strukturyzowane w przedstawiony sposób, nie muszą być reprezentowane za pomocą jednej tabeli. Złożone zbiory danych mogą obejmować dane zapisane w wielu tabelach (np. w bazie danych; zobacz rozdział 13.). W innych złożonych strukturach danych każda komórka tabeli może obejmować wektor, a nawet całą tabelę. Wtedy tabela nie jest już dwuwymiarowa, ale trójwymiarowa lub o większej liczbie wymiarów. Wiele zbiorów danych udostępnianych przez usługi sieciowe ma strukturę zagnieżdżonych tabel. Informacje na ten temat zawiera rozdział 14.

9.4. Interpretowanie danych

Pierwszą rzeczą, jaką należy zrozumieć po pobraniu zbioru danych (który znalazłeś w internecie lub otrzymałeś w firmie), jest zrozumienie znaczenia danych. Wymaga to zrozumienia dziedziny, w jakiej pracujesz, a także konkretnego użytego schematu danych.

9.4.1. Zdobycie wiedzy w danej dziedzinie

Pierwszy krok w kierunku zrozumienia zbioru danych polega na zbadaniu i poznaniu dziedziny problemu. **Dziedzina problemu** to zestaw zagadnień powiązanych z danym problemem, czyli kontekst danych. Praca z danymi wymaga **wiedzy z zakresu dziedziny**. Musisz choć na podstawowym poziomie rozumieć dziedzinę problemu, aby móc przeprowadzić jakiegokolwiek sensowne analizy danych. Musisz też opracować model umysłowy dotyczący znaczenia danych. Należy zrozumieć znaczenie i funkcje wszystkich atrybutów (aby nie wykonywać obliczeń na pozbawionych kontekstu liczbach), zakres oczekiwanych wartości poszczególnych atrybutów (na potrzeby wykrywania wartości odstających i innych błędów), a także subtelności, które mogą nie być od razu widoczne w zbiorze danych (np. błędy systematyczne lub agregacje, które mogą ukrywać ważne związki przyczynowo-skutkowe).

Oto konkretny przykład: jeśli chcesz przeanalizować tabelę z rysunku 9.1, musisz najpierw zrozumieć, co oznacza „wzrost” i „waga” osoby, używane jednostki (cale, centymetry, a może coś innego?), oczekiwany zakres wartości (czy wzrost 64 Ady oznacza, że jest ona niska?) i inne zewnętrzne czynniki, które mogą wpływać na dane (np. wiek osób).

Zapamiętaj: Nie musisz być ekspertem w dziedzinie problemu (choć z pewnością nie zaszkodzi nim być). Musisz jedynie posiadać *wystarczającą* wiedzę, aby pracować z danymi z tej dziedziny.

Choć dane o wzroście ludzi i inne zbiory danych z tej książki powinny być dla większości czytelników zrozumiałe, w praktyce z dużym prawdopodobieństwem natrafisz na dziedziny problemu, w których nie jesteś ekspertem. Jeszcze bardziej problematyczne jest to, że może

Ci się *wydawać*, że rozumiesz daną dziedzinę problemu, jednak w rzeczywistości posiadasz jej błędny model umysłowy (jest to błąd w obszarze *metapoznania*).

Rozważ np. zbiór danych z rysunku 9.2 (jest to zrzut tabeli z repozytorium danych miasta Seattle). Ten zbiór danych zawiera informacje na temat zezwoleń na użytkowanie gruntu, związanych z dość skomplikowaną procedurą biurokratyczną, z którą wiele osób może nie być zaznajomionych. Oto pytanie: jak zdobyć wystarczającą wiedzę z dziedziny problemu, aby móc zrozumieć i przeanalizować ten zbiór danych?

Application/Permit Num	Permit Type	Address	Description
3022652	DESIGN REVIEW WITH EDG, SEPA THRESHOLD DETERMINATION	911 WESTERN AVE	Land Use Application to
3009478	ADMINISTRATIVE CONDITIONAL USE, SEPA THRESHOLD DETERMINATION	1745 24TH AVE S	Land Use Application to
3008870	SEPA THRESHOLD DETERMINATION	2701 S CHARLESTOWN ST	REVISED BY 3013602 Lar
3007778	DESIGN REVIEW WITH EDG, SEPA THRESHOLD DETERMINATION	1605 BELLEVUE AVE	Land Use Application to
3008235	SHORT PLAT	927 29TH AVE S	Canceled for failure to re
3008234	SHORT PLAT	911 29TH AVE S	Canceled for failure to re
3003274	DESIGN REVIEW WITH EDG, SEPA THRESHOLD DETERMINATION	8512 20TH AVE NE	Land Use Application to
3003225	DESIGN REVIEW WITH EDG, SEPA THRESHOLD DETERMINATION	3025 NE 130TH ST	Land use application to z
3004392	SEPA THRESHOLD DETERMINATION, SHORELINE DEVELOPMENT, SPECIAL EXCEPTION	1201 AMGEN CT W	Shoreline substantial de
3003328	SHORT PLAT	4426 44TH AVE SW	Land use permit to subd
3003127	ADMIN DESIGN REVIEW WITH EDG	619 13TH AVE E	CANCELLED - DECISION I
3003226	ADMINISTRATIVE DESIGN REVIEW, SEPA THRESHOLD DETERMINATION	6400 30TH AVE SW	Land Use Permit to appr
3026712			
3026542			

Rysunek 9.2. Podgląd danych na temat zezwoleń na użytkowanie gruntu w Seattle¹⁵. Zawartość została zmodyfikowana na potrzeby tej książki

Zdobywanie wiedzy z dziedziny prawie zawsze wymaga nauki z zewnętrznych źródeł. Rzadko uda Ci się zrozumieć dziedzinę na podstawie samych analiz liczb z arkusza kalkulacyjnego. Aby zdobyć ogólną wiedzę z dziedziny, warto zacząć od ogólnego źródła wiedzy, *Wikipedii*, gdzie łatwo znajdziesz podstawowe opisy. Koniecznie przeczytaj powiązane artykuły lub materiały, aby lepiej zrozumieć zagadnienie. Aby przejrzeć obszernie źródła informacji w internecie, należy zapoznać się z różnymi materiałami, do których prowadzą odsyłacze, i przełożyć znalezione informacje na używany zbiór danych.

¹⁵ **Miasto Seattle — zezwolenia na użytkowanie gruntu** (aby uzyskać dostęp, trzeba założyć bezpłatne konto): <https://data.seattle.gov/Permitting/Land-Use-Permits/uuyd-8gak>.

Jednak najlepszym sposobem na poznanie dziedziny problemu jest znalezienie *eksperta z dziedziny*, który objaśni Ci dane zagadnienie. Jeśli chcesz dowiedzieć się czegoś o pozwoleniach na użytkowanie gruntu, spróbuj znaleźć kogoś, kto korzystał z takiego pozwolenia. Drugie w kolejności rozwiązanie polega na zapytaniu się bibliotekarza. Bibliotekarze uczą się pomagać ludziom w znajdowaniu i zdobywaniu podstawowej wiedzy z dziedziny. W bibliotece możesz też znaleźć wyspecjalizowane źródła informacji.

9.4.2. Jak zrozumieć schematy danych?

Gdy już na ogólnym poziomie rozumiesz kontekst dla zbioru danych, możesz zacząć je interpretować. Musisz skupić się na zrozumieniu **schematu danych** (czyli tego, co reprezentują wiersze i kolumny), a także kontekstu wartości. Zachęcamy, aby w analizach zadawać sobie następujące pytania:

Jakie metadane są dostępne dla zbioru danych?

Wiele publicznie dostępnych zbiorów danych jest powiązanych z podsumowującymi objaśnieniami, instrukcjami dostępu i użytkowania, a nawet opisami poszczególnych atrybutów. Tego rodzaju **metadane** (czyli dane na temat danych) pochodzą bezpośrednio od źródła danych, dlatego są najlepszym sposobem na zrozumienie, jakie wartości są reprezentowane w poszczególnych komórkach tabeli.

Na przykład na stronie dotyczącej zezwoleń na użytkowanie gruntu w Seattle znajdują się: krótkie podsumowanie (choć warto sprawdzić, co oznacza punkt „over-the-counter review application”), liczba kategorii i oznaczeń, wymiary zbioru danych (w czasie powstawania tej książki dostępnych jest 14 200 wierszy) i krótkie opisy wszystkich kolumn.

Wyjątkowo ważnym aspektem metadanych, który należy sprawdzić, jest następująca kwestia:

Kto utworzył ten zbiór danych? Skąd pochodzą dane?

Zrozumienie tego, kto wygenerował zbiór danych (i jak to zrobił) pozwoli Ci ustalić, gdzie znaleźć więcej informacji na temat tych danych. Dowiesz się też, kim są eksperci z określonej dziedziny. Znajomość źródła danych i metod ich zbierania może też pomóc Ci odkryć ukryte błędy systematyczne lub inne subtelnosci, które nie są oczywiste na podstawie samych danych. Na przykład na stronie dotyczącej zezwoleń na użytkowanie gruntu napisano, że dane zostały udostępnione przez dział planowania i rozwoju miasta Seattle (obecnie dział budownictwa i inspekcji — Department of Construction & Inspections). Jeśli poszukasz tej jednostki, może znajdziesz jej witrynę¹⁶. Jest ona dobrym miejscem do tego, by znaleźć więcej informacji na temat danych z używanego zbioru.

Gdy już zrozumiesz metadane, możesz zacząć analizować same dane:

Jakie atrybuty znajdują się w zbiorze danych?

¹⁶ **Dział budownictwa i inspekcji miasta Seattle** (dostęp wymaga założenia bezpłatnego konta): <http://www.seattle.gov/dpd/>.

Niezależnie od dostępności metadanych musisz zrozumieć kolumny tabeli. Przyjrzyj się każdej kolumnie i sprawdź, czy znasz odpowiedzi na następujące pytania:

1. Jaki atrybut z rzeczywistego świata dana kolumna ma reprezentować?
2. W jakich jednostkach podawane są wartości (dotyczy to danych ciągłych)?
3. Jakie kategorie są reprezentowane i co one oznaczają (dotyczy to danych nominalnych)?
4. Jaki jest możliwy zakres wartości?

Jeśli metadane zawierają *legendę* tabeli, zadanie jest łatwe. W przeciwnym razie do zrozumienia atrybutów konieczne może być przeanalizowanie źródła danych, co wymaga dodatkowych badań nad dziedziną problemu.

Wskazówka: Gdy przeglądasz zbiór danych — a tak naprawdę niemal dowolne materiały — powinieneś *zapisywać* pojęcia i wyrażenia, których nie znasz, aby móc je później sprawdzić. Dzięki temu nie będziesz (nieprawidłowo) domyślał się znaczenia pojęć i łatwiej rozdzielił nazwy, które już znasz, od tych jeszcze nieznanach.

Na przykład w metadanych w zbiorze danych dotyczącym zezwoleń na użytkowanie gruntu znajdują się jednoznaczne opisy kolumn, jednak gdy przyjrzyysz się przykładowym danym, zobaczysz, że niektóre wartości mogą wymagać dodatkowych analiz. Na przykład czym są różne rodzaje zezwoleń (Permit Types) i rodzaje decyzji (Decision Types)? Gdy wrócisz do źródła danych (na stronę główną działu budownictwa), możesz przejść do strony zezwoleń (Permits), a następnie kliknąć Permits We Issue (A-Z), aby zobaczyć pełną listę rodzajów zezwoleń. Dzięki temu dowiesz się np., że zezwolenie PLAT dotyczy „tworzenia lub modyfikowania poszczególnych działek w ramach nieruchomości”, czyli zmiany granicy działek.

Aby zrozumieć atrybuty, musisz przyjrzeć się przykładowym obserwacjom. Otwórz arkusz kalkulacyjny lub tabelę i przejrzyj kilka pierwszych wierszy, aby zobaczyć, jakiego rodzaju wartości występują i co może z nich wynikać.

W całym procesie nieustannie powinieneś zadawać sobie pytanie:

Jakich pojęć nie znam lub nie rozumiem?

W zależności od dziedziny problemu zbiór danych może obejmować dużą ilość *żargonu* — zarówno w samych danych, jak i w ich objaśnieniach. Upewnij się, że rozumiesz wszystkie używane pojęcia techniczne. Jest to bardzo ważna kwestia, jeśli chcesz skutecznie omawiać i analizować dane.

Ostrzeżenie: Zwracaj uwagę na akronimy, których nie znasz, i koniecznie sprawdzaj ich znaczenie.

Na przykład w podglądzie tabeli (Table Preview) wiele wartości atrybutu Permit Type to SEPA. Gdy poszukasz tego akronimu, znajdziesz stronę z opisem ustawy *State Policy Environmental Act* (wymaga ona uwzględnienia wpływu sposobu użytkowania gruntu na środowisko), a także informacje na temat procesu „określenia wartości progowej” (Threshold Determination).

Interpretowanie zbioru danych wymaga więc analiz i pracy, która *nie* jest programowaniem. Choć może się wydawać, że taka praca utrudnia Ci postępy w przetwarzaniu danych, poprawny model umysłowy danych jest przydatny i niezbędny do przeprowadzania analiz.

9.5. Odpowiadanie na pytania na podstawie danych

Prawdopodobnie najtrudniejszym aspektem analizy danych jest skuteczne zadawanie pytań na podstawie danych, aby uzyskać pożądane informacje. Specjalista od nauki o danych często odpowiada za przekładanie różnych pytań z dziedziny na konkretne obserwacje i atrybuty w zbiorze danych. Przyjrzyj się np. następującemu pytaniu:

Jaka jest najgorsza choroba w Stanach Zjednoczonych?

Aby odpowiedzieć na to pytanie, trzeba zrozumieć dziedzinę problemu — pomiary obciążenia chorobami — i znaleźć zbiór danych, który nadaje się do udzielenia odpowiedzi. Odpowiednim zbiorem danych może być tu badanie *Global Burden of Disease*¹⁷ przeprowadzone przez jednostkę Institute for Health Metrics and Evaluation. Szczegółowo opisano tam *obciążenie chorobami* w Stanach Zjednoczonych i na świecie.

Po pobraniu tego zbioru danych musisz **zoperacjonalizować** podstawowe pytanie. Przeanalizujmy każde kluczowe słowo — musisz zidentyfikować zbiór *chorób*, a następnie zdefiniować, co oznacza, że choroba jest „najgorsza”. Pytanie można ująć bardziej konkretnie, co ilustrują poniższe interpretacje:

- Która choroba powoduje *największą liczbę zgonów* w Stanach Zjednoczonych?
- Która choroba powoduje *największą liczbę przedwczesnych zgonów* w Stanach Zjednoczonych?
- Która choroba powoduje *największą liczbę niepełnosprawności* w Stanach Zjednoczonych?

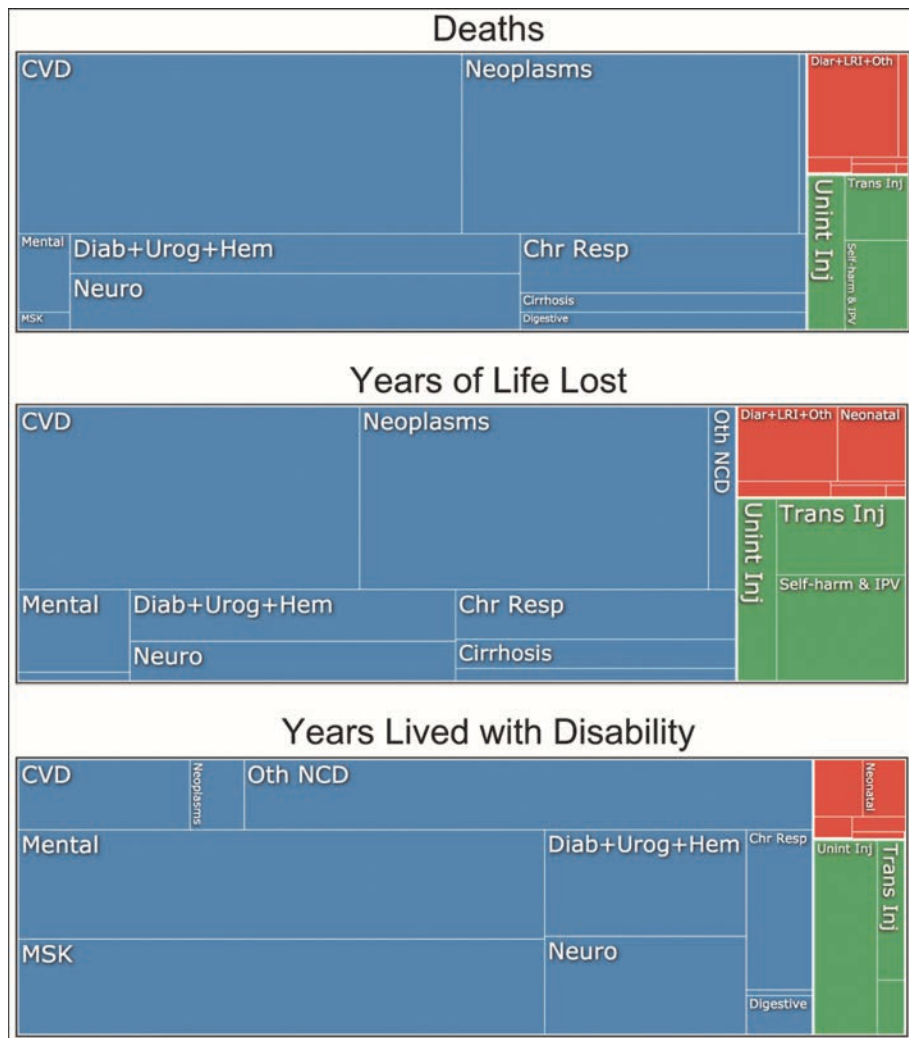
W zależności od definicji „najgorszej” choroby będziesz wykonywał zupełnie inne obliczenia i analizy. Zapewne uzyskasz też inne odpowiedzi. Dlatego musisz umieć zdecydować, *co dokładnie oznacza dane pytanie*. Zadanie to wymaga zrozumienia niuansów z dziedziny problemu związanej z pytaniem.

Na rysunku 9.3 pokazano wizualizacje, które pomagają odpowiedzieć na zadane pytanie. Na rysunku znajdują się rzuty *map drzewa* (ang. *treemap*) z internetowego narzędzia *GBD Compare*¹⁸. Mapa drzewa przypomina wykres kołowy, ale jest zbudowana z prostokątów. Powierzchnia każdego segmentu jest proporcjonalna do wartości danych. Dodatkową zaletą map drzew jest to, że pozwalają pokazywać *hierarchie* informacji dzięki *zagnieżdżaniu* różnych poziomów prostokątów jedne w drugich. Na przykład na rysunku 9.3 obciążenie chorobami zakaźnymi (wyróżnionymi na czerwono) jest przedstawione w jednym segmencie wykresu.

W zależności od tego, jak zoperacjonalizujesz „najgorszą chorobę”, wybrana zostanie inna z nich. Na rysunku 9.3 widać, że prawie 90% wszystkich zgonów jest spowodowanych chorobami niezakaźnymi (kolor niebieski), takimi jak choroby układu krążenia (*CVD*) i nowotwory (*Neoplasms*). Gdy uwzględnić wiek pacjentów (za pomocą wskaźnika *utracone lata życia*), odsetek zgonów spowodowanych przez te choroby spada do 80%. Ten wskaźnik pozwala też zidentyfikować przyczyny zgonów, które w nieproporcjonalny sposób wpływają na młodych ludzi. Te przyczyny to np. wypadki drogowe (*Trans Inj*) i samookaleczenia. Te czynniki są

¹⁷ IHME — **globalne obciążenie chorobami**: <http://www.healthdata.org/node/835>.

¹⁸ GBD Compare — wizualizacja globalnego obciążenia chorobami: <https://vizhub.healthdata.org/gbd-compare/>.



Rysunek 9.3. Mapy drzewa z narzędzia GBD Compare. Widoczne są tu proporcje zgonów (u góry), utraconych lat życia (środek) i lat przeżytych z niepełnosprawnością (u dołu) powodowanych przez różne choroby w Stanach Zjednoczonych

wyróżnione kolorem zielonym (zobacz środkowy wykres na rysunku 9.3). Z kolei jeśli za „najgorsze” choroby uznać te, które powodują najwięcej fizycznych niepełnosprawności w populacji (dolny wykres na rysunku 9.3), ujawnia się wpływ chorób układu mięśniowo-kostnego (MSK) i zaburzeń umysłowych (*Mental*).

Ponieważ analizy danych służą do znajdowania odpowiedzi na pytania, pierwszy krok polega na upewnieniu się, że dobrze rozumiesz samo pytanie i sposoby pomiarów. Dopiero po przełożeniu pytań na konkretne atrybuty (kolumny) danych możesz przeprowadzić skuteczną i sensowną analizę.

Skorowidz

A

adres URL, 194
 znaki zabronione, 198
algorytm, 93
 obiektywizm, 349
analiza danych, 133, 217
 eksploracyjna, 347
 jednostka, 157
 siła zależności, 347
 statystyczna, 218
 wtórna, 124
ankieta, 123
aplikacja
 dynamiczna, 310
 interfejs użytkownika, *Patrz:* interfejs
 użytkownika
 internetowa, 301
 reaktywna, 303
 serwer, 302
 serwer, 304, 305, 306, 307
 tworzenie, 315
architektura REST, 201
argument, 89, 90, 92, 93, 96
 anonimowy, 309
 kolejność, 98
arkusz
 CSV, 124
 kalkulacyjny, 140
Atom, 23, 26, 27, 50
 paleta poleceń, 27
 podgląd plików Markdown, 69

B

baza danych
 hasło, 190
 klucz, *Patrz:* klucz
 kolumna, 182

rekord, *Patrz:* baza danych wiersz
relacyjna, 158, 181, 182
 tworzenie, 184
 wiersz, 182

biblioteka, *Patrz:* pakiet
big data, 128
BitBucket, 50
błąd, 40
 metapoznania, 130
Bokeh, 267
Boole George, 81
Brewer Cynthia, 232
Bryan Jenny, 161

C

chmura, 42
code review, *Patrz:* kod ocena
ColorBrewer, 232, 252
commit, *Patrz:* rewizja
CRAN, 28
czujnik, 123

D

dane, 126
 agregacja, 145, 152, 153, 237
 analiza, *Patrz:* analiza danych
 atrybut, 128, 132
 ciągłe, 220
 dziedzina, 129, 131, 133
 filtrowanie, 145, 149, 187, 210
 gramatyka operowania, 145, 163, 167, 241, 242
 aspekty estetyczne, 241, 244, 250
 dostosowanie pozycji, 242, 248
 faseta, 242, 254
 obiekt geometryczny, 241
 skala, 242, 250
 skala kolorów, 252
 system współrzędnych, 242, 253

dane

- hierarchiczne, 227
- histogram, *Patrz:* histogram
- interpretowanie, 129, 130, 131, 132
- jednostka analiz, 157
- kodowanie graficzne, 229, 230, 231, 247
 - ekspresywność, 236, 237
 - estetyka, 238, 244
 - kolor, 231, 232, 233, 247
 - legenda, 239
 - przetwarzanie przeduwagowe, 235
 - stosunek danych do tuszu, 238
- łączenie, 145
- modyfikowanie, 145
- nominalne, 126, 142, 220, 222
 - współwystępowanie, 225
- obserwacje odstające, 220
- operacjonalizacja, 133
- pobieranie, 145, 147, 194
 - JSON, 205, 206, 207
- pochodzenie, 131
- porządkowanie, 145, 151
- porządkowe, 126
- przetwarzanie
 - JSON, 203, 204
 - niestandardowe, *Patrz:* NSE
- rejestr, 123
- schemat, 131
- sortowanie, 151
- splaszczanie, 207, 209
- struktura, 127, 128, 129
 - zagnieżdżona, 227
- typ, *Patrz:* typ
- ustrukturyzowane, 203
- wartościowanie leniwe, 192
- wejściowe, 301
- wizualizacja, *Patrz:* wizualizacja
- wyjściowe, 283
 - dynamiczne, 303, 311, 312, 315, 316
 - jako tabela, 312
 - jako tekst, 311
 - przekierowanie, 41
 - R Markdown, 289
 - statyczne, 317
- wzorzec, 217
- zagregowanie, 222
- zbiór, 140
 - anscombe, 217
 - iris, 267
 - midwest, 242

- źródło, 123, 124, 125
 - zdalne, 191

- debugowanie, 155, 286
 - z użyciem gumowej kaczki, 83
- dokumentacja, 37, 65, 86
 - grafika, 67
 - hiperłącze, 67
 - tabela, 68
 - wbudowana, 83
- domena, 195
- domknięcie, 317

E

- edytor tekstu, 26
 - Atom, *Patrz:* Atom
 - RStudio, *Patrz:* RStudio
 - Sublime Text, *Patrz:* Sublime Text
 - vim, 54
 - Visual Studio Code, *Patrz:* Visual Studio Code

F

- facet_, *Patrz:* funkcja:
- faktor, 136, 140, 142
 - tworzenie, 143
- faseta, 224, 254
- feature branch, *Patrz:* gałąź funkcji
- fork, 342, 344
- format
 - .Rmd, 284
 - CSV, 124, 139, 203
 - HTML, 293
 - JSON, 115, 197, 201, 203
 - para klucz-wartość, 203
 - pobieranie danych, 205, 206
 - przetwarzanie danych, 206, 207
 - splaszczanie danych, 207, 209
 - tablica, 204
 - SVG, 275
- forum StackOverflow, 83
- funkcja, 89
 - a, 309
 - addCircles, 275, 277
 - addProviderTiles, 274
 - addTiles, 273, 274
 - aes, 244, 245, 247
 - aes_string, 245
 - argument, *Patrz:* argument
 - arrange, 145, 151
 - as.data.frame, 157

- as.factor, 143
- askForPassword, 190
- c, 91, 101, 148
- cat, 290
- checkboxGroupInput, 310
- checkboxInput, 310
- ciało, 96
- collect, 192
- colnames, 137, 138
- colorFactor, 279
- content, 203, 209, 211
- dane
 - wejściowe, *Patrz:* argument
 - wyjściowe, 90
- data, 140
- data.frame, 136
- dataTableOutput, 312
- dbConnect, 190
- dbListTables, 191
- debugowanie, 97
- dim, 137
- em, 309
- facet_wrap, 254
- figure, 272
- filter, 145, 149, 156, 163
- flatten, 208, 209, 211
- fluidPage, 304
- fromJSON, 206, 209
- full_join, 161
- generująca dane, 316, 317
- generująca dane, 303, 304
- geom_
 - odzworowanie aspektów estetycznych, 244, 247, 248
 - transformacja statystyczna, 247
- geom_col, 245, 248
- geom_hex, 245
- geom_label_repel, 256
- geom_line, 245
- geom_point, 244, 245
- geom_polygon, 245, 258, 263
- GET, 202
- ggplot, 243, 244, 249, 261
- ggplotly, 269
- group_by, 155, 156, 247
- h1, 308, 309
- h2, 309
- head, 137
- help, 84
- hermetyzowanie, 317
- htmlOutput, 311
- img, 309
- inner_join, 161
- install.packages, 94
- is.data.frame, 206
- kable, 291, 296
- labs, 255, 270
- lapply, 119, 120
- layout, 270
- leaflet, 273
- left_join, 158, 159, 160
- length, 102, 105
- library, 94, 319
- map_data, 258
- mean, 156
- mutate, 145, 150, 151
- my_df, 138
- names, 114, 206, 211
- navbarPage, 313
- nazwa, 95, 96
- nchar, 91, 106
- ncol, 137
- nrow, 137
- p, 309
- palette_fn, 279
- parametr, *Patrz:* argument
- paste, 91, 92, 104, 106, 291
- plot_ly, 270
- plotlyOutput, 312
- plotOutput, 312
- print, 79, 206, 289
- przestrzeń nazw, *Patrz:* przestrzeń nazw
- qplot, 245
- radioButtons, 311
- read.csv, 140
- rename, 151
- renderDataTable, 316
- renderLeaflet, 316
- renderPlot, 316
- renderPlotly, 316
- renderPrint, 316
- renderTable, 316
- renderText, 306, 316
- return, 96
- right_join, 161
- round, 91, 92, 93, 106
- rownames, 137
- sapply, 120
- scale_color_brewer, 252
- scale_color_manual, 252

funkcja

select, 145, 147, 149, 186
 selectInput, 310
 seq, 91, 102
 setView, 274
 setwd, 141
 shinyApp, 305, 323
 show_query, 191
 showLogs, 319
 sidebarLayout, 313
 składnia, 90, 95
 sliderInput, 310
 source, 200, 286, 292, 307
 str, 206
 str_count, 94
 strong, 309
 sum, 91
 summarise, *Patrz:* funkcja summarize
 summarize, 145, 152, 156, 247
 tableOutput, 312
 tail, 137
 tbl, 191
 textInput, 306, 308, 310
 textOutput, 308, 311
 theme, 260, 270
 toupper, 91
 tworzenie, 95
 układu, 304
 verbatimTextOutput, 312
 View, 137
 wartość, 96
 wbudowana, 91
 wczytywanie, 93
 wektorowa, 106
 zagnieżdżanie, 154
 znaczników, 308

G

gałąź, 327

funkcji, 337, 338, 341
 master, 329
 przechodzenie, 330, 331
 scalanie, 332, 335, 336
 tworzenie, 329
 usuwanie, 332

GBD Compare, 133

Gentleman Robert, 73

git model gałęzi, 327

GitHub, 23, 26, 47, 49, 56

fork, *Patrz:* fork

interfejs API, *Patrz:* interfejs API GitHub

klucz SSH, 50

konfiguracja, 56

plik .html, 293

pull request, *Patrz:* pull request

scalanie rewizji, 336

zmiany, 49

GitHub Desktop, 50

GitHub Pages, 294

konfiguracja, 294

GitLab, 50

grafika tworzenie, 93

H

hasło, 43, 200

histogram, 220

I

IDE, 23, 74

RStudio, *Patrz:* RStudio

identyfikator URI, 194, 195, 196, 210

bazowy, 195

Ihaka Ross, 73

instrukcja

apt-get, 26

przekierowania, 41

SELECT, 185, 186, 187

warunkowa, 98, 99

integrated development environment, *Patrz:* IDE

interfejs

API, 193

autoryzacja, 209

Bokeh, 271

GitHub, 194, 198, 201

Plotly, 269

punkt końcowy, 196

typu REST, 201

Yelp Fusion, 209

użytkownika, 301, 302, 304, 305, 306, 307, 312

projektowanie, 308

treść statyczna, 308, 309

tworzenie, 322

interfejs użytkownika, 304, 305, 306, 307

interpreter, 28

J

język

otwarty, 73

programowania

składnia, 79
 do obliczeń statystycznych, 73
 HTML, 309
 interpretowany, 73
 JavaScript, 269
 kompilowany, 73
 Markdown, *Patrz:* Markdown
 Matlab, 269
 Python, 269, 348
 R, 24, *Patrz:* R
 S, 73
 SQL, *Patrz:* SQL
 z typowaniem dynamicznym, 80
 z typowaniem statycznym, 80
 znaczników, 65

K

kartogram, 258
 katalog
 nadrzędny, 36
 roboczy, 33, 141
 klauzula
 GROUP_BY, 188
 JOIN, 188
 ON, 188
 ORDER_BY, 188
 WHERE, 187, 188
 klucz
 API, 209, *Patrz:* token dostępu
 główny, 182
 ssh, 43
 zewnątrzny, 182
 knitting, 286
 kod
 ocena, 343
 styl, 79, *Patrz też:* wytyczne na temat stylu
 wykonywanie, 75
 w wierszu poleceń, 77
 kodowanie graficzne estetyka, 241
 kolekcja jednowymiarowa, 113
 komentarz, 32
 dodawanie, 78
 komputer zdalny, 42, 43
 komunikat, 40, 83
 kryteria ekspresywności Mackinlaya, 236
 kwartet Anscombe'a, 217, 218

L

liczba, 82

lista, 113, 135
 element, 115
 etykieta, 113, 114, 116
 indeks, 115, 116
 modyfikowanie, 117
 klucz = wartość, 136, 198
 nazwana, 136
 nienumerowana, 290
 tworzenie, 114
 z formatu JSON, 206
 zagnieżdżona, 116, 204

Ł

łańcuch znaków, 81, 94
 liczba znaków, 106
 w formacie Markdown, 289

M

macierz, 139
 wykresów punktowych, *Patrz:* wykres
 punktowy macierz
 Mackinlaya kryteria ekspresywności, *Patrz:*
 kryteria ekspresywności Mackinlaya
 manual, *Patrz:* dokumentacja
 mapa, 113, 209, 257
 ciepła, 226
 drzewa, 133, 222, 227
 fragment, 262
 interaktywna, 273, 275
 kafelek, 273, 274, *Patrz też:* mapa fragment
 Carto, 274
 OpenStreetMap, 274
 kropkowa, 258
 legenda, 279
 punktowa, 261
 warstwa, 275
 Markdown, 27, 65, 66, 67, *Patrz też:* R Markdown,
 pakiet rmarkdown, plik .Rmd
 konwersja na pdf, 70
 wyświetlanie dokumentu, 68, 69
 menu rozwijane, 310
 metadane, 48, 131, 132, 285
 metoda
 add_headers, 210
 statystyczna, 347
 Microsoft Azure, 42
 migawka, 48, 56
 model REST, *Patrz:* żądanie REST
 MySQL, 185

N

nagłówek HTTP, 209
 narzędzie GBD Compare, *Patrz:* GBD Compare
 nauka o danych, 13
 moralność, 349
 non-standard evaluation, *Patrz:* NSE
 notacja
 dolara, 115, 117, 137, 138
 snake_case, 79, 95
 z nawiasem kwadratowym, 107, 118, 148
 z podwójnym nawiasem kwadratowym, 116,
 118, 137, 138, 322
 z pojedynczym nawiasem kwadratowym
 ramka danych, 138, 139
 NSE, 147

O

OAuth, 200
 obiekt geometryczny, 242, 243, 245, 248
 obserwacja, 128, *Patrz też:* rekord
 odpowiedź, 202
 OpenStreetMap, 274
 operator
 %>%, *Patrz:* operator potoku
 dwukropka, 102, 108
 logiczny, 82
 matematyczny, 79, 80, 93
 porównania, *Patrz:* operator relacji
 potoku, 154, 155
 przypisania, 79
 relacji, 81

P

pakiet, 93
 DBI, 190
 dbplyr, 190, 191, 192
 dplyr, 93, 145, 147, 154, 162, 191, 193, 247,
 249, 296
 instalowanie, 146
 join, 145
 operacje na grupach, 155
 DT, 312
 ggmap, 262
 ggplot2, 93, 146, 241, 242, 244, 257, 267, 312
 rozszerzenie, 262
 httr, 202, 205
 instalowanie, 94
 jsonlite, 205, 206, 208

kableExtra, 291
 knitr, 283, 286, 291
 leaflet, 267, 268, 273, 279, 322
 lintr, 95
 magrittr, 155
 plotly, 267, 268, 269
 interfejs API, 269
 pscl, 146
 randomForest, 93
 rbokeh, 267, 268, 271, 272, 273
 interfejs API, 271
 RColorBrewer, 233
 rmarkdown, 283
 rsconnect, 318, 319
 rstudioapi, 190
 rworldmap, 296
 shiny, 303
 stringr, 94
 styler, 95
 tidyr, 146, 249
 tidyverse, 146
 wczytywanie, 94
 zależność, 155
 panel mainPanel, 322
 pętla, 107
 platforma
 R Markdown, *Patrz:* R Markdown, pakiet
 Markdown, plik .Rmd, Markdown
 Shiny, *Patrz:* Shiny
 shinyapps.io, 318
 plik, 37
 .csv, 140, 181, 203, 319
 .DS_Store, 62, 63
 .gitignore, 62, 63, 200
 .html, 293
 .Rmd, 42, 284
 dane, 285, 286
 kod, 285, 286, 287, 288
 kompilowanie, 286, 287, 289
 nagłówek, 285
 .sqlite, 184
 analysis.R, 296
 app.R, 305, 307, 319
 format, *Patrz:* format
 index.html, 294
 index.Rmd, 296
 kopiowanie, 37, 43
 README.md, 69
 rozszerzenie, 26
 server.R, 307

- server.R, 307
 - shapefile, 258, 259
 - ścieżka, *Patrz:* ścieżka
 - ui.R, 307
 - usuwanie, 37
 - podręcznik, *Patrz:* dokumentacja
 - pole tekstowe, 310
 - polecenie
 - !!, 37
 - argument, 34, 38
 - cat, 37
 - cd, 34, 35, 43
 - cp, 37, 44
 - curl, 39
 - cut, 39
 - echo, 40
 - exit, 43
 - git
 - add, 52, 53, 63
 - branch, 329, 330
 - checkout, 61, 62, 330, 331, 332
 - checkout master, 62
 - clone, 57, 58, 63
 - commit, 54, 56, 63, 331
 - config, 50
 - init, 51
 - log, 60, 61
 - merge, 332, 333, 335
 - pull, 59, 63
 - push, 59, 63
 - remote, 59
 - reset, 56
 - revert, 62
 - status, 51, 52, 54, 63, 331, 333, 335
 - grep, 39, 42
 - head, 39
 - history, 37
 - ls, 35, 43, 51
 - man, 38
 - mkdir, 37
 - opcja, 38
 - open, 37
 - pwd, 32, 43, 141
 - q, 38
 - rm, 37
 - say, 39
 - scp, 44
 - sed, 39
 - sort, 39
 - ssh, 42
 - start, 37
 - uniq, 39
 - uzupełnianie za pomocą tabulacji, 36
 - wc, 39, 42
 - zatrzymanie, 40
 - Postgres, 184, 185, 190
 - PostgreSQL, *Patrz:* Postgres
 - powłoka
 - Bash, 24
 - Git Bash, 25
 - Terminal, 24, 25
 - systemowa, 24, 31
 - wiersz polecenia, 25
 - PR, *Patrz:* pull request
 - prognozowanie, 348
 - program
 - Atom, *Patrz:* Atom
 - gałąź, *Patrz:* gałąź
 - git, 23, 25, 47, 48, 49
 - gałąź, 49, 51
 - konfiguracja, 50
 - Linux, 26
 - macOS, 25
 - Windows, 25
 - GitHub, *Patrz:* GitHub
 - Powershell, 25
 - RGui, 77
 - programowanie zespołowe, 48, 49, 59, 327, 336
 - fork, 342
 - scentralizowany proces pracy, 338, 339, 340, 341
 - protokół
 - HTTP, 194, 201
 - HTTPS, 195
 - SSH, 42, 44
 - przeglądarka less, 38
 - przestrzeń nazw, 94
 - przycisk opcji, 311
 - pull request, 343, 345
 - punkt kontrolny, 48
- ## R
- R
 - instalowanie, 28
 - nauka, 84, 85
 - sesja interaktywna, 76, 77
 - R Markdown, 283, *Patrz też:* pakiet Markdown,
 - plik .Rmd, Markdown
 - format wyjściowy, 293
 - HTML, 293

R Markdown

- kod, 287, 288
- pomoc, 289
- zastosowania, 295
- ramka danych, 113, 120, 135, 189
 - filtrowanie, 138, 139
 - jako tabela Markdown, 291
 - struktura, 137
 - tibble, 156
 - tworzenie, 136
 - z formatu JSON, 206
 - z nazwami wierszy, 150
 - złączenie, *Patrz:* złączenie
- Raymond Eric, 47
- referencja, 191
- rekord, 128
- relacja wiele do wielu, 182
- repozytorium
 - ignorowanie plików, 62
 - lista sugestii, 63
 - scalanie wersji, 59
- repozytorium, 48
 - centralne, 49, 338, 339
 - dodawanie plików, 53, 54
 - fork, 57, 58
 - lokalne, 56
 - poczekalnia, 53, 54
 - podrzędne, 51
 - stan, 51, 52
 - tworzenie, 51, 58, 339
 - w chmurze, 56
 - zdalne, 49, 56
- rewizja, 48, 60
 - HEAD, 329
 - historia, 60
 - opis, 55
 - scalanie, 335, 336
 - sekwencja, 327, 328
 - skrót, 327
- RStudio, 23, 26, 28, 50, 74, 189, 284, 307
 - dokumentacja, 75
 - katalog roboczy, 141
 - konsola, 75
 - panel, 74, 75
 - podgląd kodu przed kompilacją, 292
 - skrót klawiaturowy, 155
 - społeczność, 84
 - środowisko, 75
 - zainstalowanie, 28

S

- serwer
 - WWW, 194
 - zdalny, 43, 193
- serwis shinyapps.io, 318
- Shiny, 301, 302, 306, 309, 310, 312, 315, 317, 318, 320
 - hosting aplikacji, 318
- sieć społecznościowa, 124, 125
- skala pomiarowa, 126, 127
- skrypt, 73, 75, *Patrz też:* kod app.R, 303
- słownik, 113
- słowo kluczowe
 - AS, 187
 - else, 98
 - function, 95
 - if, 98
 - INNER JOIN, 188
 - LEFT JOIN, 188
 - LIKE, 187
 - OUTER JOIN, 188
 - RIGHT JOIN, 188
- snapshot, *Patrz:* migawka
- Sourcetree, 50
- SQL
 - nauka, 185
 - złączenie, 188
- SQLite, 184, 185, 190
- statystyka, 126
- Sublime Text, 27
- suwak, 310, 311
- symbol wieloznaczny, 39
- system
 - kontroli wersji, 25, 47, 48, 327
 - odwracalność, 61
 - RDMS, 184, 185, 189
 - uwierzytelniania OAuth, *Patrz:* OAuth

Ś

- ścieżka, 35, 44
 - bezwzględna, 36, 67, 141
 - względna, 36, 67, 141, 142
- środowisko programistyczne zintegrowane, *Patrz:* IDE

T

- tabela, 128, 135
 - interaktywna, 312

tablica, 204

- asocjacyjna, 113
- zagnieżdżona, 204

 tekst, 40

- blok, 66
- formatowanie, 65, 66
- hiperłącze, 67
- tabela, 68

 terminal, 24, 31

 token dostępu, 199, 200

 Tufte Edward, 238

 typ

- całkowitoliczbowy, 82
- liczb zespolonych, 82
- liczbowy, 80
- logiczny, 81
- podstawowy, 80
- znakowy, 81

U

uczenie statystyczne, 347

 układ

- fluidPage, 313, 322
- mainPanel, 313
- sidebarPanel, 313
- tabPanel, 313
- tworzenie, 312, 315
- zagnieżdżanie, 313

 USA wybory prezydenckie, 146

 usługa

- sieciowa, 193, 194
 - dokumentacja, 196
 - rejestracja, 199
 - zasoby podrzędne, 196
- typu REST, 201
- uwierzytelniania
 - OAuth, *Patrz:* OAuth

V

Visual Studio Code, 27

W

warstwa, 242, 243

 wartościowanie leniwe, 192

 wartość

- NA, 108, 111, 118, 140
- NULL, 117, 118
- skalarna, 105

wektor, 136, 143

- działania element po elemencie, 102

 element

- indeks, 107, 108
- indeks ujemny, 108
- liczba, 102
 - używanie ponowne, 104, 105

 filtrowanie, 109, 111, 118

 indeksów, 108

 łączenie, 106

 modyfikowanie, 110, 111

 podzbiór, 107, 108

 tworzenie, 101, 102

 wartości logicznych, 109, 139

 wektoryzacja, 107

 Wickham Hadley, 145, 241

 budżet, 302, 303, 310, 312

- argument, 311
- galeria, 311
- interaktywny, 310
- interaktywny, 307

 wiersz

- poleczeń, 24, 31
 - Linux, 25
 - macOS, 24
 - uruchamianie, 31
 - Windows, 25

 Wilkinson Leland, 241

 witryna, 302

 wizualizacja, 133, 217

- cel, 217
- ekspresywność, 236, 237
- estetyka, 238
- interaktywna, 267, 276
- jednej zmiennej, 220
- kolor, 231, 232, 233
- legenda, 239
- ograniczenia, 219
- przetwarzanie przeduwagowe, 235
- stosunek danych do tuszu, 238
- wielu zmiennych, 223

 wykres, 242, 292, 312

- etykieta osi, 255, 256
- interaktywny, 312
- kodowanie na podstawie powierzchni, 227
- kołowy, 222
 - zagnieżdżony, 227
- podrzędny, 254
- promieni słonecznych, 229
- pudełkowy, 220

wykres

- punktowy, 223
 - macierz, 223
 - skrzypcowy, 220, 224
 - słonecznikowy, 227
 - słupkowy, 220, 277
 - kolor, 249
 - skumulowany, 222
 - tytuł, 255
 - wielofasetowy, 254
- wyrażenie, 80
- dynamiczne, 316, 317
- wytyczne na temat stylu, 79
- tidyverse, 79, 81, 95
 - naruszanie, 95

Z

- zapytanie, 188, 189, 191, 192, 195
- parametr, 196, 197, 198, 200
 - tworzenie, 186
- złączenie, 157, 158
- filtrujące, 161
 - kolejność argumentów, 160
 - lewostronne, 158, 159, 160, 188
 - pełne, 188
 - prawostronne, 161, 188
 - SQL, 188
 - wektor z nazwami, 159
 - wewnętrzne, 161
 - zewewnętrzne, 161
- zmienna
- anonimowa, 91, 154
 - nazwa, 78
 - pośrednia, 153
 - przechowująca funkcję, 90
 - przypisywanie wartości, 79
 - systemowa PATH, 78
 - zasięg, 96
- znak
- ' , *Patrz:* znak apostrofu
 - !! , 37
 - # , *Patrz:* znak kratki
 - \$, 32, *Patrz:* znak dolara
 - %>% , *Patrz:* operator potoku
 - & , *Patrz:* znak ampersand
 - * , *Patrz:* znak gwiazdki

- ., *Patrz:* znak kropki
- ., 36
- / , *Patrz:* znak ukośnik
- ; , *Patrz:* znak dwukropka
- ? , *Patrz:* znak zapytania, *Patrz:* znak zapytania
- ?? , 84
- @ , 43
- [[]] , 116
- [] , 38, 107
- ` , *Patrz:* znak grawis
- {} , 96, 203
- | , 42, 68
- ~ , *Patrz:* znak tyldy
- <- , *Patrz:* operator przypisania
- > , 41
- >> , 42
- ampersand, 198
- apostrofu, 81
- cudzysłowu, 81
- dolara, 115
- dwukropka, 102, 108
- grawis, 287, 288
- gwiazdki, 39, 186
- kratki, 32, 78, 308
- kropki, 36
- łańcuch, *Patrz:* łańcuch znaków
- plus, 81
- potoku, *Patrz:* znak |
- tyldy, 36
- ukośnik, 33, 35
- zachęty, 32
- zapytania, 84, 197

Ż

żądanie

- DELETE, 201
- GET, 201, 202, 209, 210
- HTTP, 194, 201
- odpowiedź, 194, *Patrz:* odpowiedź
- OPTIONS, 201
- PATCH, 201
- POST, 201
- PR, *Patrz:* pull request
- PUT, 201
- REST, 193, 194

PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Po prostu R i dane. Wyciśniesz każdą kroplę wiedzy!

Aby surowe dane przekuć w gotową do wykorzystania wiedzę, potrzebna jest umiejętność ich analizy, przekształcania i niekiedy również wizualizacji. Nagrodą za włożony w to wysiłek jest lepsze rozumienie różnych złożonych zagadnień z wielu dziedzin wiedzy. Co więcej, znajomość procesów programowego przetwarzania danych pozwala na szybkie wykrywanie i opisywanie wzorców danych, praktycznie niemożliwych do dostrzeżenia innymi technikami. Dla wielu badaczy jednak barierą na drodze do skorzystania z tych atrakcyjnych możliwości jest konieczność pisania kodu.

Oto podręcznik programowania w języku R dla analityków danych, szczególnie przydatny dla osób, które nie mają doświadczenia w tej dziedzinie. Dokładnie opisano tu potrzebne narzędzia i technologie. Zamieszczono wskazówki dotyczące instalacji i konfiguracji oprogramowania do pisania kodu, wykonywania go i zarządzania nim, a także śledzenia wersji projektów i wprowadzanych w nich zmian oraz korzystania z innych podstawowych mechanizmów. Poszczególne kroki tworzenia kodu w języku R wyjaśniono dokładnie i przystępnie. Dzięki tej książce można płynnie przejść do konkretnych zadań i budować potrzebne aplikacje. Zrozumienie prezentowanych w niej treści ułatwiają liczne przykłady i ćwiczenia, co pozwala szybko przystąpić do skutecznego analizowania własnych zbiorów danych.

W książce:

- przygotowanie środowiska pracy i rozpoczęcie programowania w R
- podstawy zarządzania projektami, kontrola wersji i generowanie dokumentacji
- ramki danych, pakiety *dplyr* i *tidyr*
- kod do ich wizualizacji i pakiet *ggplot2*
- tworzenie aplikacji i techniki współpracy w zespołach specjalistów

MICHAEL FREEMAN – jest wykładowcą akademickim specjalizującym się w nauce o danych i ich wizualizacji danych. Wcześniej prowadził ogólnoświatowe badania dotyczące zdrowia publicznego. Interesuje się wykorzystaniem nauki o danych w obszarze sprawiedliwości społecznej.

JOEL ROSS – jest wykładowcą akademickim. Specjalizuje się w nauczaniu programowania. Interesuje się badaniami z zakresu gier i grywalizacji oraz systemami „przetwarzania bez granic”. Prowadził też badania nad systemami finansowania społecznościowego i wspomaganie ekorozwoju.

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

Sprawdź nasze szkolenia
SZKOLENIA
AKADEMIA IT & BUSINESS
WWW.SZKOLENIA.HELION.PL

KOD KORZYŚCI
Sięgnij po więcej!



ISBN 978-83-283-5782-2

