

Marek Gągolewski

WYDANIE II
POSZERZONE

Programowanie w języku R

Analiza danych,
obliczenia,
symulacje



Programowanie w języku R

Marek Gągolewski

Programowanie w języku R

Analiza danych,
obliczenia,
symulacje



[Kup książkę](#)

Projekt okładki **Hubert Zacharski**

Ilustracja na okładce **shutterstock/antishock**

Wydawca **Łukasz Łopuszański**

Redaktor prowadzący **Iwona Lewandowska**

Redaktor **Ewa Ławrynowicz**

Koordynator produkcji **Anna Bączkowska**

Skład i łamanie **FixPoint**, Warszawa

Zastrzeżonych nazw firm i produktów użyto w książce wyłącznie w celu identyfikacji.

Copyright © by Wydawnictwo Naukowe PWN SA
Warszawa 2014, 2016

ISBN 978-83-01-18939-6

Wydanie II
Warszawa 2016

Wydawnictwo Naukowe PWN SA
02-460 Warszawa, ul. Gottlieba Daimlera 2
tel. 22 69 54 321, faks 22 69 54 288
infolinia 801 33 33 88
e-mail: pwn@pwn.com.pl; reklama@pwn.pl
www.pwn.pl

Druk i oprawa: OSDW Azymut Sp. z o.o.

[Kup książkę](#)

SPIS TREŚCI

Przedmowa	XIII
---------------------	------

I Podstawy

1. Środowisko R i program RStudio	3
1.1. Cechy języka i środowiska R	3
1.2. Organizacja pracy w R i RStudio	4
1.2.1. Konsola R	5
1.2.2. Program RStudio	6
1.2.3. Pierwsze kroki w trybie interaktywnym	8
1.2.4. Edytor skryptów	10
1.2.5. System pomocy	11
2. Typy atomowe: wektory i NULL	13
2.1. Klasyfikacja typów obiektów w języku R	13
2.2. Wektory atomowe	14
2.2.1. Wektory wartości logicznych	14
2.2.2. Wektory liczbowe	17
2.2.3. Wektory napisów	18
2.2.4. Pozostałe typy wektorów atomowych i ich hierarchia	19
2.3. Tworzenie obiektów nazwanych	25
2.4. Braki danych, wartości nieskończone i nie-liczby	29
2.5. Typ pusty (NULL)	31
3. Operacje na wektorach	34
3.1. Podstawowe operatory	34
3.1.1. Operatory arytmetyczne	35
3.1.2. Operatory logiczne	38
3.1.3. Operatory relacyjne	39
3.1.4. Priorytety operatorów	41
3.2. Indeksowanie wektorów. Filtrowanie danych	42
3.2.1. Rodzaje indeksatorów	43
3.2.2. Modyfikowanie wybranych elementów	45
3.3. Przegląd funkcji wbudowanych	46
3.3.1. Zwektoryzowane funkcje matematyczne	46

3.3.2.	Agregacja danych	51
3.3.3.	Operacje na sąsiadujących elementach wektorów	54
3.3.4.	Wyszukiwanie indeksów i wybór elementów wektorów	55
3.3.5.	Operacje oparte na permutowaniu elementów wektorów	57
3.3.6.	Operacje na zbiorach	59
3.3.7.	Podstawowe operacje na napisach	60
4.	Listy	63
4.1.	Tworzenie list	63
4.2.	Indeksowanie list	66
4.2.1.	Operator „[”	66
4.2.2.	Operator „[”	66
4.2.3.	Modyfikowanie zawartości list	67
4.3.	Wybrane operacje na listach	70
4.3.1.	Łączenie, rozwijanie i powielanie list	70
4.3.2.	Wywoływanie funkcji na wszystkich elementach listy	73
5.	Funkcje	78
5.1.	Tworzenie obiektów typu funkcja	79
5.1.1.	Bloki wyrażeń	81
5.1.2.	Sprawdzanie poprawności argumentów	84
5.1.3.	Kilka uwag dla projektantów funkcji	87
5.1.4.	Biblioteki funkcji w plikach .R	88
5.1.5.	Odwoływanie się do funkcji z pakietów R	89
5.2.	Zasięg nazw w funkcjach	90
5.3.	Parametry i argumenty	92
5.3.1.	Przekazywanie argumentów przez wartość	92
5.3.2.	Parametry z argumentami domyślnymi	93
5.3.3.	Parametr specjalny „...”	94
6.	Atrybuty obiektów	97
6.1.	Nadawanie i odczytywanie atrybutów	97
6.2.	Atrybuty specjalne	100
6.2.1.	Atrybut <i>comment</i>	101
6.2.2.	Atrybut <i>names</i> . Wektory z etykietami	101
6.2.3.	Atrybut <i>class</i> . Wstęp do programowania obiektowego S3	106
6.3.	O zachowywaniu i gubieniu atrybutów przez funkcje	111
7.	Typy złożone	114
7.1.	Macierze i tablice	114
7.1.1.	Tworzenie macierzy	114
7.1.2.	Indeksowanie macierzy	118
7.1.3.	Tablice jako uogólnienie macierzy	120
7.1.4.	Atrybut <i>dimnames</i> . Etykietowanie wierszy i kolumn	121
7.1.5.	Reprezentacja macierzy i tablic	122
7.1.6.	Wybrane operacje na macierzach	126
7.2.	Szeregi czasowe	129
7.3.	Czynniki	131
7.3.1.	Tworzenie czynników	132

7.3.2.	Reprezentacja czynników	132
7.3.3.	Czynniki a wektory napisów	134
7.3.4.	Wybrane operacje na czynnikach	135
7.4.	Ramki danych	138
7.4.1.	Reprezentacja ramek danych	139
7.4.2.	Operatory indeksowania. Filtrowanie danych	141
7.4.3.	Wybrane operacje na ramach danych	144
8.	Pielęgnowanie kodu	156
8.1.	Organizacja kodu	157
8.1.1.	Projekty i skrypty	157
8.1.2.	Tworzenie własnych pakietów R	158
8.2.	Obsługa wyjątków	159
8.2.1.	Rodzaje wyjątków	159
8.2.2.	Obsługa komunikatów diagnostycznych i ostrzeżeń	160
8.2.3.	Obsługa błędów	161
8.3.	Testowanie oprogramowania	162
8.4.	Debugowanie kodu	165
8.5.	Pomiar i poprawa wydajności kodu	167
8.5.1.	Badanie krótkich fragmentów kodu	167
8.5.2.	Profilowanie aplikacji	168
8.5.3.	Złożoność czasowa i pamięciowa algorytmów	171
9.	Modyfikacja przepływu sterowania	174
9.1.	Wyrażenia warunkowe <code>if</code> i <code>if...else</code>	175
9.1.1.	Określanie testowanego warunku	178
9.1.2.	Wartości zwracane przez wyrażenia warunkowe	181
9.1.3.	Funkcja <code>return()</code> . Rekurencja	182
9.2.	Pętle	184
9.2.1.	Pętla <code>while</code>	184
9.2.2.	Pętla <code>repeat</code>	189
9.2.3.	Pętla <code>for</code>	190
9.3.	Uwagi na temat wydajności pętli	193
II	Przygotowanie danych	
10.	Przetwarzanie napisów	203
10.1.	Podstawowe operacje na napisach	203
10.1.1.	Wyznaczanie długości napisów	203
10.1.2.	Porównywanie napisów	204
10.1.3.	Łączenie i powielanie napisów	206
10.1.4.	Przycinanie i wypełnianie	207
10.1.5.	Formowanie napisów na podstawie innych obiektów	208
10.1.6.	Zmiana pojedynczych znaków	211
10.1.7.	Wyznaczanie podnapisów	211
10.1.8.	Pozostałe operacje	213
10.2.	Wyszukiwanie wzorca	214
10.2.1.	Wzorce ustalone	215

10.2.2. Wyrażenia regularne	218
10.2.3. Wzorce rozmyte	229
10.3. Data i czas	230
10.3.1. Reprezentacja dat	230
10.3.2. Reprezentacja czasu	231
10.3.3. Operacje arytmetyczne	233
10.3.4. Konwersja daty i czasu	234
10.4. Reprezentacja napisów	235
10.4.1. Kodowanie ASCII	235
10.4.2. 8-bitowe kodowania polskich liter diakrytyzowanych	237
10.4.3. Kodowanie UTF-8	238
10.4.4. Konwersja kodowań	239
11. Przetwarzanie plików	241
11.1. Podstawowe operacje na plikach i katalogach	241
11.1.1. Ścieżki dostępu do plików i katalogów	241
11.1.2. Bieżący katalog roboczy. Ścieżki względne	243
11.1.3. Informacje o plikach i katalogach	244
11.1.4. Wybrane działania na plikach i katalogach	245
11.1.5. Wyszukiwanie plików i katalogów	246
11.2. Serializacja i deserializacja obiektów	248
11.3. Popularne formaty plików	249
11.3.1. Pliki CSV	250
11.3.2. Pliki JSON	254
11.3.3. Pliki XML	255
11.4. Dostęp do baz danych SQL	256
11.5. Dowlone pliki tekstowe	257
11.5.1. Odczyt plików tekstowych	258
11.5.2. Zapis plików tekstowych	258
11.6. Połączenia	259
11.6.1. URL, czyli ujednolicony lokalizator zasobów	259
11.6.2. Tworzenie połączeń	260
11.6.3. Otwieranie i zamykanie połączeń	262
11.6.4. Odczyt danych z połączeń	262
11.6.5. Zapis danych do połączeń	265
11.6.6. Zarządzanie otwartymi połączeniami	266
11.6.7. Nota o plikach binarnych	267

III Prezentacja wyników

12. Tworzenie wykresów	271
12.1. Schemat systemów graficznych w środowisku R	271
12.2. Podstawy użycia pakietu graphics	273
12.2.1. Strona i rysunki	274
12.2.2. Parametry graficzne	275
12.2.3. Rysowanie punktów i odcinków (łamanych)	279
12.2.4. Barwy	282
12.2.5. Rysowanie wielokątów	284
12.2.6. Wypisywanie tekstu	286

12.2.7. Układ współrzędnych	287
12.2.8. Tworzenie wielu rysunków na jednej stronie	291
12.3. Wybrane wysokopoziomowe operacje graficzne	292
12.3.1. Rysowanie układu współrzędnych	292
12.3.2. Adnotacje i legenda	293
12.3.3. Wizualizacja danych jednowymiarowych	295
12.3.4. Wizualizacja danych dwuwymiarowych	298
12.3.5. Wizualizacja danych wielowymiarowych	302
12.4. Urządzenia graficzne	304
12.4.1. Urządzenia <code>pdf()</code> , <code>svg()</code> i <code>postscript()</code>	307
12.4.2. Urządzenia <code>png()</code> i <code>jpeg()</code>	307
13. Generowanie raportów przy użyciu pakietu knitr	309
13.1. Język Markdown	309
13.2. Podstawy użycia pakietu knitr	316
13.3. Ustawienia wstawek	320
13.3.1. Identyfikatory wstawek i zależności między nimi	320
13.3.2. Pamięć podręczna	321
13.3.3. Wyświetlanie kodu i wyników tekstowych	322
13.3.4. Rysunki	323
13.3.5. Ustawienia globalne	324
IV Zastosowania	
14. Obliczenia numeryczne	337
14.1. Wprowadzenie	337
14.2. Algebra wektorów i macierzy	340
14.2.1. Podstawowe operacje	341
14.2.2. Normy	342
14.2.3. Metryki i inne odległości	344
14.2.4. Wektory i wartości własne	348
14.2.5. Rozkład Choleskiego	350
14.2.6. Rozkład QR	351
14.2.7. Rozkład SVD	354
14.3. Różniczkowanie i całkowanie	356
14.3.1. Różniczkowanie numeryczne	356
14.3.2. Całkowanie numeryczne	359
14.4. Optymalizacja	360
14.4.1. Algorytmy programowania matematycznego	362
14.4.2. Algorytmy optymalizacji ciągłej ogólnego zastosowania	365
14.5. Interpolacja i wygładzanie	368
14.5.1. Interpolacja jednowymiarowa	368
14.5.2. Interpolacja dwuwymiarowa	369
14.5.3. Wygładzanie	370
14.6. Rozwiązywanie (układów) równań (nie)liniowych	372
14.6.1. Wyznaczanie miejsc zerowych funkcji jednej zmiennej	372
14.6.2. Rozwiązywanie układów równań liniowych	374
14.6.3. Rozwiązywanie układów równań nieliniowych	374

15. Symulacje	376
15.1. Generowanie liczb (pseudo)losowych	376
15.1.1. Źródła (pseudo)losowości	377
15.1.2. Określanie ziarna generatora	378
15.1.3. Szczegóły działania generatora	379
15.2. Rozkłady prawdopodobieństwa	381
15.2.1. Nazwy funkcji związanych z rozkładami	381
15.2.2. Wybrane jednowymiarowe rozkłady prawdopodobieństwa	382
15.2.3. Zmienne losowe wielowymiarowe	386
15.3. Przykładowe eksperymenty symulacyjne	390
15.3.1. Badanie mocy testu Shapiro–Wilka	391
15.3.2. Własności estymatorów parametrów rozkładu Gamma	392
15.3.3. Całkowanie Monte Carlo	396
15.3.4. Krosvalidacja klasyfikatora	398

V Zagadnienia zaawansowane

16. Zarządzanie środowiskiem R	403
16.1. Podstawowe informacje	403
16.1.1. Informacje o środowisku R	403
16.1.2. Informacje o systemie	406
16.1.3. Uruchamianie i zamykanie środowiska R	407
16.1.4. Historia poleceń	408
16.2. Opcje globalne	408
16.3. Ustawienia lokalizacyjne	412
16.4. Rozszerzanie możliwości środowiska R	415
16.4.1. Instalacja i aktualizacja pakietów	416
16.4.2. Wywoływanie innych programów	421
16.5. Zarządzanie pamięcią	422
16.5.1. Informacja o rozmiarze obiektów	422
16.5.2. Kopiowanie na żądanie	424
16.5.3. Automatyczne odśmiecanie pamięci	425
16.6. Typ podstawowy, tryb a klasa obiektów	425
17. Środowiska	428
17.1. Środowiska jako zbiory obiektów	428
17.1.1. Podstawowe operacje na obiektach w środowisku	429
17.1.2. Środowiska a listy	431
17.1.3. Przekazywanie środowisk funkcjom	433
17.2. Wskaźniki na środowiska otaczające	435
17.2.1. Przypadek ręcznie tworzonych środowisk	435
17.2.2. Ścieżka wyszukiwania	436
17.2.3. Przestrzenie nazw i środowiska eksportowane przez pakiety	441
18. Syntaktyka i semantyka języka R	442
18.1. Obiekty reprezentujące wyrażenia języka R	442
18.1.1. Parser	443
18.1.2. Cytowanie	446

18.1.3. Wywołania, czyli wyrażenia złożone	446
18.2. Środowiskowy model obliczeń	451
18.2.1. Ewaluacja wyrażen	452
18.2.2. Bieżące środowisko ewaluacyjne	454
18.3. Ewaluacja wyrażen w obrębie funkcji	457
18.3.1. Lokalne środowiska ewaluacyjne	459
18.3.2. Środowiska otaczające lokalne środowiska ewaluacyjne	460
18.3.3. Środowiska wywołujące	464
18.3.4. Ewaluacja argumentów	465
18.4. Formuły	471
18.4.1. Przykłady funkcji stosujących argumenty typu formuła	471
18.4.2. Formuły jako wywołania	473
18.4.3. Przetwarzanie formuł	474
19. Programowanie zorientowane obiektowo	476
19.1. Paradygmaty programowania obiektowego	476
19.2. Klasy S3	478
19.2.1. Określanie klasy obiektu	479
19.2.2. Ekspediowanie metod	479
19.2.3. Przeciążanie metod	482
19.3. Klasy S4	483
19.3.1. Definiowanie klas i tworzenie obiektów	484
19.3.2. Definiowanie funkcji generycznych i metod	487
19.4. Klasy referencyjne (RC)	491
19.5. Specjalne rodzaje funkcji	492
19.5.1. Funkcje podstawieniowe	492
19.5.2. Przeciążanie operatorów	494
19.5.3. Wbudowane grupy funkcji generycznych	495
20. Integracja R i C++ przy użyciu pakietu Rcpp	498
20.1. Wprowadzenie	499
20.1.1. Tryby pracy z Rcpp	499
20.1.2. Podstawy składni języka C++	503
20.2. Operacje na wektorach atomowych	509
20.2.1. Dostęp do wektorów	509
20.2.2. Tworzenie wektorów	512
20.2.3. Kopiowanie płytkie i głębokie	513
20.2.4. Braki danych	515
20.2.5. Przegląd funkcji z R/C API	517
20.2.6. Przegląd funkcji i metod z pakietu Rcpp	520
20.3. Operacje na pozostałych typach obiektów	521
20.3.1. Listy	521
20.3.2. Funkcje	523
20.3.3. Atrybuty obiektów	524
20.3.4. Obiekty typów złożonych	525
20.3.5. Wskaźniki	527
Bibliografia	532
Skorowidz	537

PRZEDMOWA

Korporacje, rządy, instytuty naukowe, a nawet zwykli użytkownicy internetu generują informacje różnego rodzaju: dotyczące rozrywki, kultury, komunikacji międzyludzkiej, ekonomii, handlu, przemysłu, transportu, zdrowia, środowiska itd. Postęp technologiczny, jaki dokonał się w naukach informacyjnych na przestrzeni ostatnich dekad, skutkuje tym, że dzisiejszy świat wytwarza więcej danych, niż jest ich w stanie efektywnie przetworzyć. Owo „wąskie gardło” często wcale nie jest spowodowane brakiem dostępności odpowiednich algorytmów i narzędzi, ale raczej trudnością w znalezieniu – tak przecież rozbuchowywanych przez pracodawców – dobrze przygotowanych profesjonalistów.

Przekształcanie surowych danych (ilościowych, jakościowych, tekstu itp.) na przyswajalną *wiedzę* jest celem działania m.in. *analitików danych*, specjalistów *business intelligence*, *badaczy opinii i rynku* czy wreszcie *data scientists*. Dziedziny te wymagają nie tylko obycia w zakresie szeroko pojętej probabilistyki, szeregów czasowych, metod statystycznego i maszynowego uczenia się, modelowania, przetwarzania obrazów, językoznawstwa itp., co wystarczająco pilny student może poznać lepiej lub gorzej z różnego rodzaju kursów lub lektury dostępnych podręczników, lecz także wyobraźni, kreatywności, komunikatywności i znajomości szczególnych obszarów zastosowań (np. procesów zachodzących w firmie, ekonomii, medycyny, nauk społecznych). I wreszcie, za wręcz niezbędną uważa się *biegłą umiejętność obsługi narzędzi informatycznych, które służą do przechowywania i przetwarzania informacji, wydobywania z nich wiedzy i prezentacji uzyskanych wyników* – i to na poziomie, który nie tyle implikuje sprawne wykorzystanie istniejących rozwiązań, ile przede wszystkim umożliwia *analizę, projektowanie, implementację, testowanie i wdrażanie własnych pomysłów*, a także dzielenie się efektami pracy z innymi.

R a inne programy. Na rynku znajdziemy szeroki wybór oprogramowania, które wspiera pracę analityków danych. Przede wszystkim mamy dostęp do narzędzi skrojonych na miarę potrzeb tej grupy osób, przez co praca w nich jest bardzo wygodna i podstawowe czynności można wykonać przy ich użyciu naprawdę bardzo efektywnie. Należą do nich m.in. SAS, Stata, Statistica, SPSS i Weka.

Jednakże w praktyce to, co jest ich zaletą, może szybko obrócić się w przypadku konieczności zmierzenia się z bardziej ambitnymi, nieszablonowymi zadaniami w wadę

– a takich wyzwań jest coraz więcej. Pomimo dość stromej, w każdym razie początkowo, „krzywej uczenia się” środowiska programistyczne, których trzonem są interpretowane języki programowania ogólnego zastosowania, oferują tu znacznie większy stopień swobody, przez co możliwości ich użycia są praktycznie nieograniczone. Wśród najbardziej popularnych narzędzi tego typu znajdziemy m.in. chętnie wybierany przez „rasowych” programistów język Python z rodziną pakietów SciPy, SciKits i Pandas, zob. [29], oraz szczególnie cenione przez statystyków, osoby zajmujące się wizualizacją danych i analityków danych z mniejszym informatycznym przygotowaniem, środowisko R [73].

Środowisko R, będące głównym bohaterem niniejszej książki, jest aktywnie rozwijane przez społeczność *open source*, której każdy z nas może stać się czynnym członkiem. Dzięki wytężonej pracy licznych pasjonatów i systemowi ogólnie dostępnych pakietów¹ (ang. *packages*) niezmiernie łatwe jest rozbudowywanie zdolności tego narzędzia o tak istotne możliwości jak komunikacja z bazami danych, przetwarzanie wielkich zbiorów danych, obliczenia równoległe, zbieranie danych ze stron internetowych itd. Mamy tutaj do wyboru mnóstwo implementacji znanych technik i metod statystycznej analizy danych oraz algorytmów maszynowego uczenia się, m.in. analizy przeżycia, algorytmów *bootstrapowych*, sieci neuronowych, metod analizy regresji i klasyfikacji, analizy skupień, szeregów czasowych, danych finansowych, redukcji wymiarowości itd. Co więcej, przeglądając publikacje naukowe, możemy zaobserwować, że wiele nowych metod analizy danych jest najpierw implementowanych właśnie w R.

CIEKAWOSTKA

Dużym niedopatrzaniem jest określanie R mianem „pakietu do obliczeń statystycznych”. Taka etykieta działa wręcz na jego szkodę, kreuje bowiem fałszywą opinię, że jest to narzędzie o wąskich możliwościach. Po pierwsze, jest to całe *środowisko* zbudowane wokół funkcyjnego, interpretowanego języka programowania ogólnego zastosowania o naprawdę ciekawych i godnych podziwu cechach – obliczenia statystyczne i wizualizacja danych są tylko jednym z wielu możliwych obszarów jego użycia, aczkolwiek oczywiście najbardziej docenianym na całym świecie. Po drugie, to właśnie dzięki *pakietom* (nakładkom rozszerzającym jego możliwości) obliczenia takowe są możliwe. Pakietów służących do przeprowadzania „obliczeń statystycznych” (np. stats) nie trzeba wcale łądować – zamiast nich można korzystać z narzędzi zupełnie innego rodzaju (por. też tematykę najczęściej pobieranych pakietów na s. 420).

Cel książki. Mówi się, że komputer jest tak mądry, jak jego programista. Dobry programista to taki, który potrafi zaprogramować wszystko, o czym jest w stanie pomyśleć. Nabywszy solidnych umiejętności, dostrzega, że często nie warto jest „wywierać otwartych drzwi” i można skorzystać z wyników pracy innych osób. Jest przygotowany jednak do tego, by dostrzec alternatywne podejścia do interesującego go zagadnienia, potrafi docenić ich mocne strony i dostrzec ograniczenia. Wreszcie wie, kiedy należy „wziąć

¹W repozytorium CRAN znajdziemy aktualnie (czerwiec 2016 r.) prawie 10 000 pakietów. Co więcej, obserwujemy, że liczba pakietów rośnie wykładniczo.

sprawy w swoje ręce” i pewne mechanizmy dostosować samodzielnie do własnych, konkretnych potrzeb.

Znakomita część publikacji dostępnych nie tylko na polskim (por. np. [8, 9, 18, 34, 36, 77, 90, 99]), ale i zagranicznym rynku wydawniczym dotyczy sposobów wykorzystywania środowiska R w różnych zastosowaniach praktycznych – czyli uczy tak naprawdę *obcowania* z konkretnymi *pakietami*. Monografii na temat samego programowania w języku R jest niewiele. Oprócz napisanych przez jego twórców kilku pozycji, w szczególności [5, 14, 15, 89], w których są przede wszystkim omówione wybrane aspekty tworzenia oprogramowania i które są raczej przeznaczone dla tych, którzy znają już ten język dość dobrze, mamy jeszcze np. dość przystępnie napisane angielskojęzyczne książki [50, 56, 93]. Żadna z nich nie omawia jednak zarówno *fundamentów* języka, jak i zagadnień zaawansowanych w sposób wystarczająco wyczerpujący.

WAŻNE

Niniejsza publikacja stanowi kompletny i wyczerpujący kurs programowania w języku R – od omówienia *podstaw języka*, aż po przedstawienie *zaawansowanych zagadnień*. Towarzyszące jej ćwiczenia pomogą rozwijać umiejętności biegłego implementowania nowych algorytmów, dostosowywania do własnych potrzeb już istniejących oraz automatyzowania żmudnych – gdyby je wykonywać ręcznie – zadań w dosłownie każdym obszarze zastosowań. Co więcej, lektura jej pozwoli zrozumieć, dlaczego obliczenia w R są realizowane w taki, a nie inny sposób.

Szczególne uwaga została poświęcona sposobom ładowania i zapisywania zbiorów danych oraz ich generowaniu na podstawie różnych źródeł, a także ich wstępnej obróbce, czyszczeniu, aż po samą analizę i prezentację wyników. Ma ona za zadanie wspomagać nas w drodze ku programistycznej samodzielności – abyśmy mogli wyjść poza gotowe schematy i śmiało mierzyć się z nowymi wyzwaniami, przed którymi stawia nas tzw. era informacji. Odpowiedni nacisk został też położony na tworzenie kodu wysokiej jakości, nie tylko w sensie jego poprawności, czytelności i łatwości rozbudowy, ale i szybkości działania oraz ekonomicznego zużycia pamięci.

Omawiając zagadnienia programowania w języku R na słusznym poziomie ogólności, zrezygnowaliśmy ze „słownikowego” przeglądu wielu ciekawych funkcji rozproszonych po różnych pakietach (algorytmów weryfikacji hipotez, analizy wariancji i przeżycia, drzew klasyfikacyjnych, maszyn wektorów podpierających, szczególnych typów wykresów – lista ta mogłaby się ciągnąć w nieskończoność). Obiecujemy jednak, że po lekturze tej książki będziemy w stanie znaleźć potrzebne nam narzędzie i samodzielnie je poznać, a w razie potrzeby napisać własne. Dzięki temu zmniejszyliśmy też ryzyko tego, że dzieło to będzie po prostu nie tylko nudne, ale i szybko się zdezaktualizuje.

Adresaci książki. Idealnego Czytelnika tej książki najłatwiej możemy określić przez... zaprzeczenie. Otóż nie polecamy jej osobom, które:

- wcale nie są zainteresowane tym, by nauczyć się programować w języku R, ani tym, by udoskonalić swoje umiejętności;

- nie mają potrzeby w swojej pracy zawodowej lub naukowej wykonywać żadnych obliczeń, automatyzować procesów przetwarzania danych ani tworzyć grafiki użytkowej (np. wykresów);
- nie mają czasu przeczytać jej od początku do końca;
- znają już język R tak (nie)dobrze, że wydaje im się, iż niczego nowego się nie dowiedzą z lektury rozdz. 2–9;
- nie zamierzają twórczo poeksperymentować z przedstawionymi przykładami oraz nie mają chęci rozwiązywania ćwiczeń;
- boją się „zarazić” radością z tworzenia nowego oprogramowania.

W przeciwnym przypadku niniejsza pozycja jest po prostu idealna. Innymi słowy:

WAŻNE

Pozycję tę możemy polecić każdemu, kto jest zainteresowany dokładnym poznaniem wszystkich możliwości języka i środowiska R, a także efektywnym wykorzystaniem go w swojej pracy zawodowej lub naukowej. Dotyczy to zarówno osób, które nie programowały jeszcze w R, jak i tych, które z poziomu podstawowego, średniozaawansowanego lub zaawansowanego chcą wspiąć się jeszcze wyżej.

Układ książki. Przedstawiony w książce materiał jest podzielony na pięć następujących bloków tematycznych:

- 1) W pierwszej części przedstawiamy elementarz programowania w R oraz najważniejsze operacje na podstawowych typach danych, w tym na wektorach atomowych, funkcjach i listach. Tłumaczymy, dlaczego w R należy unikać stosowania m.in. pętli i jak się bez nich obyć. Mówimy też o atrybutach obiektów i ich roli w programowaniu zorientowanym obiektowo w stylu S3 oraz o tym, że typy złożone, takie jak macierz, czynnik czy ramka danych, są prostym rozszerzeniem typów podstawowych. Przedstawimy najważniejsze techniki pielęgnowania kodu i zapewniania, że będzie on dobrej jakości, po czym wprowadzimy wyrażenia służące do samodzielnej kontroli przepływu sterowania, które przydają się w sytuacjach, gdy jakiegoś algorytmu naprawdę nie da się zaimplementować wyłącznie przy użyciu funkcji wbudowanych.
- 2) W drugiej części omawiamy sposoby przetwarzania danych tekstowych, odczytywania i zapisywania najważniejszych formatów plików, zbierania danych ze stron internetowych oraz dostępu do baz danych SQL.
- 3) Trzecią część poświęcamy prezentacji wyników obliczeń, w tym tworzeniu dowolnie skomplikowanych wykresów i raportów.
- 4) W czwartej części doskonalimy nasze umiejętności w konkretnych obszarach zastosowań: analizie danych, obliczeniach numerycznych i naukowych oraz symulacjach.
- 5) W piątej części poznajemy zaawansowane aspekty programowania w języku R, takie jak środowiskowy model obliczeń, parsowanie i niestandardową ewaluację wyrażeń oraz mechanizmy programowania zorientowanego obiektowo przy

użyciu klas typu S3, S4 i RC. Mówimy także, jak tworzyć rozszerzenia dla środowiska R w języku C++, która to umiejętność przydaje się szczególnie w sytuacjach, gdy pewne funkcje okazują się być „wąskim gardłem” pod względem wydajności lub zużycia pamięci.

Zmiany w drugim wydaniu. Pierwsze wydanie *Programowania w języku R* zostało przygotowane w 2013 r. W ukazującym się po 3 latach drugim wydaniu zaszedł szereg istotnych zmian – wszystkie partie tekstu zostały zrewidowane, a wiele z nich nawet zostało napisanych ponownie. Do najbardziej zauważalnych różnic możemy zaliczyć:

- uproszczenie materiału przedstawianego na początku każdego rozdziału – trudniejsze partie zostały przesunięte zgodnie z duchem „od ogółu do szczegółu”;
- dodanie ćwiczeń (wraz z rozwiązaniami) do tekstu głównego;
- zamieszczenie większej liczby przykładów z analizy danych (opis implementacji wielu ważnych metod maszynowego uczenia się, więcej uwagi poświęconej ramkom danych);
- nowy rozdział na temat integracji C++ z R (rozdz. 20; coraz więcej pakietów jest tworzona przy użyciu Rcpp, co bardzo pozytywnie wpłynęło w ostatnich latach na szeroko pojętą jakość tego rodzaju dodatków);
- omówienie pakietu stringi w rozdziale na temat przetwarzania napisów (rozdz. 10);
- przesunięcie rozdziału na temat pętli najdalej, jak to było możliwe;
- opis sposobów dostępu do baz danych SQL, przetwarzania plików JSON i XML oraz technik wydobywania informacji ze stron internetowych (*web scraping*) w rozdz. 11;
- złączenie dwóch rozdziałów na temat generowania grafiki w jeden (rozdz. 12);
- opis języka Markdown w rozdz. 13 (jego popularność i dostępność z poziomu R w ostatnim czasie znacząco wzrosła);
- dokładniejsze omówienie sposobów tworzenia pakietów i poprawy jakości kodu w rozdz. 8;
- bardziej przystępne omówienie mechanizmów niestandardowej ewaluacji i jej znaczenia w rozdz. 17 i 18.

Uwagi o użytym oprogramowaniu. Wszystkie obliczenia zostały wykonane przy użyciu środowiska R w wersji deweloperskiej 3.4.0 (SVN rev. 70486) oraz pakietu knitr (por. [97] i rozdz. 13), a sama książka została złożona w systemie L^AT_EX [47, 68].

Podziękowania. Dziękuję moim przyjaciółom i bliskim współpracownikom za liczne uwagi i komentarze na różnych etapach przygotowywania pierwszego i/lub drugiego wydania, w szczególności Maciejowi Bartoszkowi, Annie Cenie, Janowi Laskowi, dr Annie Olwert, Agacie Sakowicz, Bartłomiejowi Tartanusowi, Barbarze Żogale-Siudem oraz p. red. Izabeli Mika. Składam także wyrazy podziękowania dla moich studentów z Wydziału Matematyki i Nauk Informacyjnych Politechniki Warszawskiej, uczestnikom i organizatorom kursów *Data Science Retreat* w Berlinie i *International PhD Studies Program* w Instytucie Podstaw Informatyki PAN, a także wszystkim użytkownikom

pakietów, których jestem autorem lub współautorem. Inspirujecie mnie do ciągłego doskonalenia moich umiejętności.

Zdaję sobie sprawę, że dzieło takiej objętości nie jest niestety pozbawione błędów, za które całkowitą odpowiedzialność ponoszę oczywiście ja sam. Proszę więc o zgłaszanie wszelkich uwag na mój adres e-mail: marek@gagolewski.com. Na stronie internetowej github.com/gagolews/Programowanie_w_jezyku_R/, do której odwiedzenia zachęcam, zamieściłem przykładowe kody źródłowe i aktualną wersję erraty.

Marek Gagolewski

Warszawa, czerwiec 2016 r.