

---

# Wydajne programowanie w R

*Praktyczny przewodnik po lepszym  
programowaniu*

*Colin Gillespie i Robin Lovelace*

*przekład: Krzysztof Kapustka*

APN Promise  
Warszawa 2018

**O'REILLY®**

---

# Spis treści

Przedmowa .....	ix
<b>1. Wprowadzenie.....</b>	<b>1</b>
Wymagania wstępne .....	2
Dla kogo jest ta książka i jak z niej korzystać?.....	2
Czym jest wydajność? .....	4
Czym jest wydajne programowanie w R?.....	5
Dlaczego wydajność?.....	7
Umiejętności uniwersalne zapewniające wydajność .....	8
Pisanie bezwzrokowe .....	8
Spójny styl i konwencje kodowania .....	9
Testy porównawcze i profilowanie .....	10
Wykonywanie testów porównawczych .....	10
Przykład testu porównawczego .....	11
Profilowanie .....	12
Materiały do książki .....	15
Pakiet R .....	15
Wersja online.....	16
Lektura uzupełniająca.....	16
<b>2. Wydajna konfiguracja.....</b>	<b>17</b>
Wymagania wstępne .....	18
Pięć głównych wskazówek dla zapewnienia wydajnej konfiguracji R .....	18
System operacyjny.....	18
Monitorowanie systemu operacyjnego i zasobów .....	19
Wersje R .....	22
Instalowanie R.....	22
Aktualizowanie R .....	23
Instalowanie pakietów R .....	24
Instalowanie pakietów R z zależnościami .....	25
Aktualizowanie pakietów R.....	25
Uruchamianie R.....	26
Argumenty uruchamiania R.....	26
Przegląd plików startowych R.....	27

Lokalizacja plików startowych .....	28
Plik .Rprofile .....	30
Przykładowy plik .Rprofile .....	30
Plik .Renviron .....	35
RStudio .....	37
Instalowanie i aktualizowanie RStudio .....	37
Układ paneli .....	38
Opcje programu RStudio .....	40
Autouzupełnianie .....	41
Skróty klawiszowe .....	43
Wyświetlanie obiektów i tabel .....	44
Zarządzanie projektami .....	44
BLAS i alternatywne interpretery R .....	46
Testowanie zysków wydajności po użyciu BLAS .....	47
Inne interpretery .....	48
Przydatne materiały .....	49
Lektura uzupełniająca .....	49
<b>3. Wydajne programowanie .....</b>	<b>51</b>
Wymagania wstępne .....	51
Pięć głównych wskazówek dla zapewnienia wydajnego programowania .....	51
Porady ogólne .....	52
Przydział pamięci .....	53
Kod zwektoryzowany .....	54
Komunikacja z użytkownikiem .....	57
Błędy krytyczne: stop() .....	57
Ostrzeżenia: warning() .....	58
Wyjście informacyjne: message() i cat() .....	59
Niewidoczne wartości zwracane .....	59
Faktory .....	60
Porządkowanie .....	61
Stały zbiór kategorii .....	61
Rodzina funkcji apply .....	62
Przykład: zbiór danych o filmach .....	64
Spójność typu .....	65
Buforowanie zmiennych .....	66
Domknięcia funkcji .....	68
Kompilator kodu pośredniego .....	70
Przykład: funkcja do obliczania średniej .....	70
Kompilowanie kodu .....	71
Lektura uzupełniająca .....	72

<b>4. Wydajny przepływ pracy</b> .....	<b>73</b>
Wymagania wstępne .....	73
Pięć głównych wskazówek dla zapewnienia wydajnego przepływu pracy ...	74
Typologia planowania projektów .....	74
Planowanie i zarządzanie projektem. ....	76
Podział pracy. ....	78
Przepływ pracy a kryteria SMART. ....	79
Wizualizowanie planów w R. ....	80
Wybór pakietów. ....	81
Wyszukiwanie pakietów .....	83
Jak wybierać pakiety? .....	83
Publikacja .....	85
Tworzenie dynamicznych dokumentów przy użyciu R Markdown .....	86
Pakiety R .....	88
Lektura uzupełniająca. ....	90
<b>5. Wydajne wejście/wyjście</b> .....	<b>91</b>
Wymagania wstępne .....	92
Pięć głównych wskazówek dla zapewnienia wydajnego wejścia/wyjścia danych	92
Uniwersalne importowanie danych z użyciem rio .....	93
Formaty zwykłego tekstu .....	94
Różnice pomiędzy fread() i read_csv() .....	97
Wstępne przetwarzanie tekstu poza R. ....	100
Formaty plików binarnych .....	100
Natywne formaty binarne: Rdata czy Rds? .....	101
Format pliku Feather .....	102
Testy porównawcze formatów plików binarnych. ....	102
Protocol Buffers .....	104
Pozyskiwanie danych z Internetu .....	104
Uzyskiwanie dostępu do danych przechowywanych w pakietach .....	105
Lektura uzupełniająca. ....	106
<b>6. Wydajna stolarka danych</b> .....	<b>107</b>
Wymagania wstępne .....	108
Pięć głównych wskazówek dla zapewnienia wydajnej stolarki danych .....	108
Wydajne ramki danych z wykorzystaniem pakietu tibble. ....	109
Oczyszczanie danych za pomocą pakietu tidyr i wyrażeń regularnych. ....	110
Tworzenie wąskich tabel za pomocą funkcji gather() .....	112
Podział zmiennych za pomocą funkcji separate() .....	113
Pozostałe funkcje tidyr. ....	114
Wyrażenia regularne .....	114
Wydajne przetwarzanie danych za pomocą pakietu dplyr. ....	117

Modyfikowanie nazw kolumn .....	119
Modyfikowanie klas kolumn. ....	120
Filtrowanie wierszy. ....	121
Łączenie operacji .....	122
Agregowanie danych .....	124
Niestandardowa ewaluacja .....	127
Łączenie zbiorów danych .....	128
Praca z bazami danych .....	130
Bazy danych i dplyr .....	132
Przetwarzanie danych przy użyciu data.table .....	134
Lektura uzupełniająca .....	137
<b>7. Wydajna optymalizacja .....</b>	<b>139</b>
Wymagania wstępne .....	140
Pięć głównych wskazówek dla zapewnienia wydajnej optymalizacji. ....	140
Profilowanie kodu .....	141
Rozpoczynanie pracy z pakietem profvis .....	141
Przykład: symulacja gry Monopol .....	142
Wydajny język R .....	144
Funkcja if() kontra ifelse(). ....	144
Sortowanie i porządkowanie. ....	145
Odwracanie kolejności elementów. ....	146
Które indeksy mają wartość TRUE?. ....	146
Konwertowanie faktorów na wartości numeryczne. ....	147
Operatory logiczne AND i OR. ....	147
Operacje na wierszach i kolumnach .....	147
Funkcje is.na() i anyNA() .....	148
Macierze. ....	148
Przykład: optymalizowanie funkcji move_square() .....	151
Przetwarzanie równoległe .....	153
Współbieżne wersje funkcji z rodziny Apply .....	153
Przykład: Węże i drabiny. ....	154
Ostrożne wychodzenie z funkcji .....	155
Równoległy kod w systemach Linux i OS X .....	155
Rcpp. ....	156
Prosta funkcja w C++ .....	157
Polecenie cppFunction() .....	158
Typy danych w C++ .....	159
Funkcja sourceCpp() .....	159
Wektory i pętle .....	161
Macierze. ....	164
C++ z dodatkiem lukru składniowego .....	164

Materiały dla Rcpp . . . . .	165
Lektura uzupełniająca . . . . .	165
<b>8. Wydajny sprzęt . . . . .</b>	<b>167</b>
Wymagania wstępne . . . . .	167
Pięć głównych porad dotyczących wydajnego sprzętu. . . . .	167
Informacje podstawowe: Czym jest bajt? . . . . .	168
Pamięć RAM . . . . .	169
Dyski twarde: HDD kontra SSD . . . . .	172
Systemy operacyjne: 32- i 64-bitowe. . . . .	173
Procesor . . . . .	174
Obliczenia w chmurze. . . . .	176
Amazon EC2 . . . . .	176
<b>9. Wydajna współpraca . . . . .</b>	<b>177</b>
Wymagania wstępne . . . . .	178
Pięć głównych wskazówek dla zapewnienia wydajnej współpracy. . . . .	178
Styl kodowania . . . . .	178
Formatowanie kodu w RStudio . . . . .	179
Nazwy plików . . . . .	180
Wczytywanie pakietów . . . . .	180
Komentowanie . . . . .	180
Nazwy obiektów . . . . .	181
Przykładowy pakiet . . . . .	182
Operacje przypisania . . . . .	183
Znaki odstępu . . . . .	183
Wcięcia . . . . .	184
Nawiasy klamrowe . . . . .	184
Kontrola wersji. . . . .	185
Zatwierdzanie kodu . . . . .	185
Integracja Git w RStudio . . . . .	186
GitHub . . . . .	187
Gałęzie, rozwidlenia, pobieranie i klonowanie. . . . .	187
Przegląd kodu. . . . .	189
Lektura uzupełniająca . . . . .	190
<b>10. Wydajne uczenie się . . . . .</b>	<b>191</b>
Wymagania wstępne . . . . .	191
Pięć głównych wskazówek dla wydajnego uczenia się . . . . .	191
Korzystanie z wewnętrznej pomocy R . . . . .	192
Wyszukiwanie tematów w R . . . . .	193
Wyszukiwanie i korzystanie z winiet . . . . .	195

Uzyskiwanie pomocy na temat funkcji .....	196
Czytanie kodu źródłowego R .....	198
swirl .....	199
Materiały online.....	199
Stack Overflow .....	201
Listy mailingowe i grupy .....	201
Zadawanie pytań .....	202
Minimalny zbiór danych .....	202
Minimalny przykład .....	203
Pogłębianie wiedzy .....	203
Szerzenie wiedzy .....	205
Lektura uzupełniająca.....	206
<b>Dodatek A. Wykorzystywane pakiety .....</b>	<b>207</b>
<b>Dodatek B. Lektura uzupełniająca .....</b>	<b>211</b>
<b>Indeks .....</b>	<b>215</b>
<b>O autorach .....</b>	<b>224</b>
<b>Kolofon .....</b>	<b>225</b>