

Marek Gągolewski
Maciej Bartoszek
Anna Cena

Przetwarzanie i analiza danych w języku Python



Przetwarzanie i analiza danych w języku Python

Marek Gągolewski
Maciej Bartoszek
Anna Cena

Przetwarzanie i analiza danych w języku Python



Projekt okładki **Hubert Zacharski**

Ilustracja na okładce **shutterstock/hipatbig**

Wydawca **Łukasz Łopuszański**

Redaktor prowadzący **Iwona Lewandowska**

Redaktor **Ewa Ławrynowicz**

Koordynator produkcji **Anna Bączkowska**

Skład i łamanie **FixPoint**

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

Copyright © by Wydawnictwo Naukowe PWN SA
Warszawa 2016

ISBN: 978-83-01-18940-2

Wydanie I
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.

SPIS TREŚCI

Przedmowa	XI
------------------------	----

I Podstawy języka Python

1. Wprowadzenie	3
1.1. Język i środowisko Python	3
1.1.1. Instalacja dystrybucji środowiska Python	3
1.1.2. Instalacja pakietów	5
1.2. Notatniki Jupyter	7
1.2.1. Tryby pracy	7
1.2.2. Najważniejsze skróty klawiszowe	10
1.2.3. Podstawy języka Markdown	10
1.3. Pierwsze kroki w języku Python	12
2. Typy skalarne	16
2.1. Liczby	16
2.1.1. Operatory arytmetyczne	18
2.1.2. Konwersja typów	21
2.1.3. Tworzenie obiektów nazwanych	22
2.1.4. Funkcje wbudowane	23
2.1.5. Pola i metody	24
2.1.6. Arytmetyka zmiennopozycyjna	25
2.2. Wartości logiczne	26
2.2.1. Operatory relacyjne	27
2.2.2. Operatory logiczne	28
2.3. Napisy	28
2.3.1. Tworzenie napisów	28
2.3.2. Podstawowe operacje na napisach	30
3. Typy sekwencyjne i iterowalne	32
3.1. Podstawowe rodziny obiektów typu sekwencyjnego	33
3.1.1. Listy i krotki	33
3.1.2. Zakresy	35
3.1.3. Napisy	35

3.2.	Zarządzanie elementami	35
3.2.1.	Wybieranie elementów	35
3.2.2.	Modyfikacja elementów	38
3.2.3.	Dodawanie i usuwanie elementów	39
3.2.4.	Kopiowanie referencji, kopiowanie płytkie a głębokie	41
3.3.	Obiekty iterowalne	45
3.4.	Działania na obiektach iterowalnych i typu sekwencyjnego	47
3.4.1.	Podstawowe metody i funkcje	47
3.4.2.	Krotki identyfikatorów po lewej stronie operatora przypisania	50
3.4.3.	Wyrażenia listotwórcze i generatory	51
3.4.4.	Formatowanie napisów	54
4.	Słowniki i zbiory	56
4.1.	Słowniki	56
4.1.1.	Tworzenie słowników	56
4.1.2.	Podstawowe metody i funkcje	58
4.2.	Zbiory	61
4.2.1.	Tworzenie zbiorów	61
4.2.2.	Podstawowe metody i funkcje	62
5.	Instrukcje sterujące	64
5.1.	Instrukcja warunkowa	64
5.2.	Pętle	66
5.2.1.	Pętla <code>while</code>	66
5.2.2.	Pętla <code>for</code>	67
5.2.3.	Instrukcje <code>break</code> i <code>continue</code> oraz blok <code>else</code> w pętlach	69
5.3.	Obsługa wyjątków	73
5.3.1.	Zgłaszanie wyjątków	74
5.3.2.	Rodzaje wyjątków	74
5.3.3.	Wychwytywanie wyjątków	75
6.	Funkcje	77
6.1.	Definiowanie funkcji	77
6.1.1.	Dokumentowanie funkcji	78
6.1.2.	Wartość zwracana	79
6.1.3.	Wyrażenia <code>lambda</code>	80
6.2.	Parametry i argumenty	81
6.2.1.	Sposób przekazywania argumentów	81
6.2.2.	Sprawdzanie poprawności argumentów	82
6.2.3.	Dopasowywanie argumentów	84
6.2.4.	Parametry z argumentami domyślnymi	84
6.2.5.	Rozpakowywanie argumentów	85
6.2.6.	Parametry specjalne <code>*args</code> i <code>**kwargs</code>	86
6.3.	Zasięg zmiennych	88
6.3.1.	Zmienne lokalne	88
6.3.2.	Zmienne globalne	88
6.3.3.	Zmienne nielokalne, fabryki funkcji i domknięcia	90
6.4.	Pakiety	92

II Przetwarzanie danych

7. Wektory, macierze i inne tablice	97
7.1. Tworzenie i reprezentacja tablic	97
7.1.1. Funkcja <code>array()</code>	98
7.1.2. Reprezentacja tablic	100
7.1.3. Typ przechowywanych elementów	101
7.1.4. Tworzenie tablic specjalnego rodzaju	103
7.1.5. Łączenie tablic	106
7.2. Podstawowe metody i funkcje	108
7.2.1. Operatory arytmetyczne. Uzgadnianie kształtów	108
7.2.2. Operacje relacyjne i logiczne	113
7.2.3. Zwektoryzowane funkcje matematyczne	115
7.2.4. Agregacja danych	118
7.2.5. Inne operacje	121
7.3. Indeksowanie tablic	123
7.3.1. Indeksowanie wektorów	123
7.3.2. Indeksowanie macierzy	128
7.3.3. Indeksowanie tablic N -wymiarowych	132
7.3.4. Wyszukiwanie indeksów elementów spełniających zadane kryteria	134
8. Ramki danych	137
8.1. Tworzenie ramek danych	138
8.1.1. Konstruktor klasy <code>DataFrame</code>	138
8.1.2. Importowanie ramek danych z plików i innych źródeł	139
8.1.3. Odczytywanie podstawowych informacji o ramkach danych	140
8.2. Zmienne, czyli obiekty typu <code>Series</code>	143
8.2.1. Wydobywanie poszczególnych zmiennych	143
8.2.2. Tworzenie i reprezentacja zmiennych	144
8.2.3. Zmienne typu <code>data</code> i <code>czas</code>	145
8.2.4. Zmienne jakościowe i porządkowe	146
8.3. Etykiety, czyli obiekty typu <code>Index</code>	150
8.3.1. Etykietowanie wierszy i kolumn	151
8.3.2. Etykiety hierarchiczne	152
8.4. Indeksowanie zmiennych i ramek danych	154
8.4.1. Wybór elementów pojedynczej zmiennej	154
8.4.2. Wybór podzbioru wierszy i kolumn ramki danych	160
8.5. Wybrane operacje	164
8.5.1. Dodawanie oraz usuwanie kolumn i wierszy	164
8.5.2. Przekształcanie zmiennych	166
8.5.3. Podsumowania ramek danych i zmiennych	168
8.5.4. Sortowanie ramek danych	172
8.5.5. Zmiana kształtu ramek danych	173
8.5.6. Obserwacje brakujące	176
9. Przetwarzanie napisów	179
9.1. Operacje na pojedynczych napisach	179
9.1.1. Podstawowe stałe napisowe i operacje na pojedynczych znakach	180

9.1.2.	Wyszukiwanie ustalonego wzorca	182
9.1.3.	Translacja znaków	183
9.1.4.	Sprawdzanie, czy wszystkie znaki należą do podanej kategorii	184
9.1.5.	Dzielenie i sklejanie tekstu	184
9.2.	Wyszukiwanie wzorca przy użyciu wyrażeń regularnych	185
9.2.1.	Definiowanie wyrażeń regularnych	186
9.2.2.	Przegląd funkcji	188
9.2.3.	Wydzielone podwyrażenia i odwołania do nich	189
9.3.	Zwektoryzowane operacje na obiektach <code>Index</code> i <code>Series</code>	190
10.	Przetwarzanie plików i zasobów w internecie	196
10.1.	Operacje na drzewie katalogów	196
10.1.1.	Ścieżki dostępu	196
10.1.2.	Wyszukiwanie plików na dysku	198
10.2.	Przetwarzanie plików	200
10.2.1.	Otwieranie pliku w różnych trybach	200
10.2.2.	Odczytywanie zawartości pliku	202
10.2.3.	Zapisywanie danych do pliku	203
10.2.4.	Serializacja obiektów	204
10.2.5.	Popularne formaty plików	205
10.3.	Pozyskiwanie danych ze stron internetowych	208
10.3.1.	Wydobywanie tabel w postaci ramek danych	209
10.3.2.	Ręczne przetwarzanie kodu źródłowego strony	209
10.3.3.	Parsowanie kodu HTML i wydobywanie pojedynczych elementów	211
11.	Dostęp do baz danych	215
11.1.	Przykładowa baza danych: <code>nycflights13</code>	215
11.2.	Obsługa baz danych	218
11.2.1.	Połączenie z bazą danych	218
11.2.2.	Eksportowanie danych do bazy	218
11.2.3.	Odczytywanie danych z bazy	219
11.2.4.	Funkcje z pakietu <code>pandas</code>	220
11.3.	Ćwiczenia	221
11.3.1.	Wybór unikatowych podzbiorów kolumn	222
11.3.2.	Agregacja danych w podgrupach	223
11.3.3.	Filtrowanie danych wejściowych i wyników	226
11.3.4.	Sortowanie wyników	230
11.3.5.	Operacje teoriomnościowe	232
11.3.6.	Złączenia	234

III Analiza danych

12.	Wizualizacja danych	239
12.1.	Rysowanie podstawowych obiektów	240
12.1.1.	Łamane	240
12.1.2.	Punkty i różne symbole	241
12.1.3.	Wielokąty	242
12.1.4.	Adnotacje tekstowe	243

12.2.	Parametry graficzne	244
12.2.1.	Sposoby kreślenia punktów i odcinków	244
12.2.2.	Sposoby określania barw	244
12.2.3.	Napisy formatujące	246
12.2.4.	Ustawienia osi	247
12.3.	Rysunki jako kombinacje obiektów podstawowych	248
12.3.1.	Wiele obiektów na jednym wykresie	248
12.3.2.	Legenda	250
12.3.3.	Wiele wykresów na jednej stronie	251
12.4.	Graficzna prezentacja danych	255
12.4.1.	Wybrane wykresy dla danych jakościowych	255
12.4.2.	Wybrane wykresy dla danych ilościowych	258
12.4.3.	Wybrane wykresy dla funkcji dwuwymiarowych	262
13.	Wnioskowanie statystyczne	265
13.1.	Wybrane rozkłady prawdopodobieństwa	265
13.1.1.	Podstawowe rodziny rozkładów	265
13.1.2.	Generowanie liczb pseudolosowych	273
13.2.	Estymacja parametrów i charakterystyk rozkładów	275
13.2.1.	Estymacja punktowa	276
13.2.2.	Estymacja przedziałowa	278
13.3.	Wykorzystanie testów statystycznych w analizie danych	280
13.3.1.	Testy zgodności	281
13.3.2.	Testy parametryczne	290
13.3.3.	Testy nieparametryczne	295
14.	Wybrane algorytmy uczenia maszynowego	298
14.1.	Przykładowy zbiór danych: winequality	298
14.2.	Analiza regresji	300
14.2.1.	Regresja liniowa	301
14.2.2.	Ocena jakości dopasowania modelu	304
14.2.3.	Model wielomianowy	306
14.2.4.	Wybór zmiennych do modelu	307
14.3.	Klasyfikacja	310
14.3.1.	Metoda k -najbliższych sąsiadów	312
14.3.2.	Ocena jakości klasyfikatora	312
14.3.3.	Drzewa decyzyjne i lasy losowe	315
14.3.4.	Porównanie krzyżowe	318
14.4.	Analiza skupień	320
14.4.1.	Algorytm k -średnich	320
14.4.2.	Hierarchiczna analiza skupień	326

IV Tworzenie własnego oprogramowania

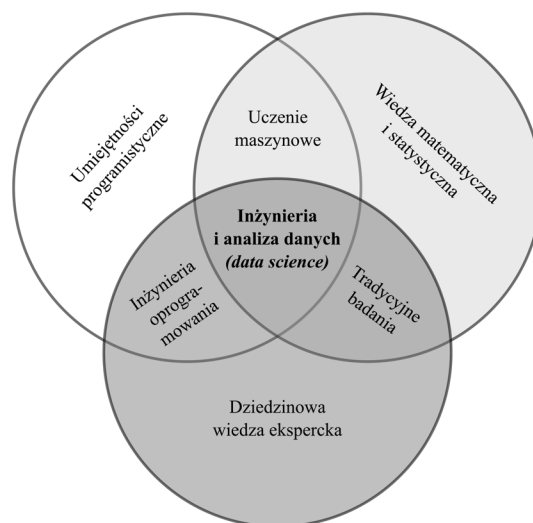
15.	Moduły, pakiety i skrypty	331
15.1.	Projekty wielomodułowe	331
15.1.1.	Środowisko programistyczne Spyder	331
15.1.2.	Tworzenie i ładowanie modułów	332

15.1.3. Tworzenie i ładowanie pakietów	335
15.1.4. Ścieżki wyszukiwania modułów i pakietów	336
15.2. Skrypty	336
15.2.1. Uruchomienie skryptu z poziomu powłoki	337
15.2.2. Przekazywanie argumentów	338
15.2.3. Skrypty a moduły. Testy jednostkowe	339
16. Programowanie obiektowe	343
16.1. Klasy i relacje między nimi	344
16.1.1. Definiowanie klasy	344
16.1.2. Dziedziczenie	346
16.2. Metody	348
16.2.1. Przeciążanie metod. Polimorfizm	348
16.2.2. Metody i pola statyczne	350
16.2.3. Metody specjalne	351
16.3. Pola	357
16.3.1. Definiowanie z góry ustalonych pól w klasie	357
16.3.2. Pola prywatne, chronione i publiczne	358
Bibliografia	361
Skorowidz	363

PRZEDMOWA

Żyjemy w fascynującej dobie przeciążenia informacyjnego: dla korporacji, instytucji i zainteresowanych jednostek zarówno samo zdobycie różnorodnych danych, jak i późniejsze ich przechowywanie nie stanowi już właściwie żadnego problemu – jeśli tylko dysponują odpowiednimi środkami. Wyzwaniem pozostaje jednak przetworzenie tego oceanu nic niemówiących ciągów bitów na użyteczną wiedzę, np. plan marketingowy firmy, zoptymalizowany pod różnymi względami sposób rekomendowania treści użytkownikom portalu internetowego, taktykę inwestowania na giełdzie, dalszy kierunek badań R&D. Z tego rodzaju wyzwaniami mierzy się m.in. względnie nowa, ale prężnie rozwijająca się dziedzina *inżynierii i analizy danych* (ang. *data science*), która wymaga od zajmującej się nią profesjonalistów nie tylko dziedzinowej wiedzy eksperckiej oraz szeroko pojętej kreatywności i ciekawości świata, ale także kompetencji matematycznych (w szczególności statystycznych) oraz umiejętności programistycznych; por. rys. 0.1.

Na rynku dostępnych jest wiele narzędzi do obliczeń analizodanowych, jednak w ostatnim czasie obserwujemy, że wielu specjalistów coraz chętniej sięga po otwarte



Rysunek 0.1. Trzy główne obszary kompetencji specjalistów inżynierii i analizy danych

i wolnodostępne rozwiązania. Aby oprogramowanie spełniało swoją funkcję w wyróżnionym przez nas obszarze zastosowań, musi zawierać pokaźny zestaw najbardziej pożytecznych, sprawdzonych, gotowych do użycia metod, służących m.in. do sporządzania wykresów, wyznaczania modeli regresji i klasyfikacji czy weryfikowania hipotez statystycznych. Co ważne, musi być ono oparte także na „pełnowymiarowym” języku programowania, tak by korzystający z niego nie czuł się w żaden sposób ograniczony – prawdziwe wyzwania *data science* bardzo rzadko wpisują się w proste szablony. Dzięki temu możliwe jest także automatyzowanie procesów przetwarzania danych i sprawianie, by przeprowadzane analizy były odtwarzalne.

O jednym z takich niezmiernie popularnych narzędzi – środowisku R (zob. np. [16]) – mówi się, że zostało stworzone przez statystyków dla statystyków. Mimo że R oferuje coraz lepszą łączność ze wszystkimi popularnymi systemami baz danych, wsparcie dla systemów *big data* (HDFS, Spark itd.), możliwość tworzenia rozszerzeń w językach Java i C++, bardzo rzadko jest on doceniany przez osoby niemające wykształcenia okołooanalizodanowego. W szczególności — z racji tego, że semantyka języka R została zainspirowana różnymi pochodnymi języka Lisp, który współcześnie nie jest już zbyt często używany — obserwujemy, że trudno jest do niego przekonać informatyków: ich opinia jest o tyle ważna, że np. implementowane przez specjalistów *data science* modele predykcyjne stają się potem częścią większych rozwiązań projektowych.

Środowisko Python jest z kolei oparte na języku o znacznie szerszych zastosowaniach. Stworzone przez programistów dla programistów pozwala na tworzenie wszelkiej maści projektów informatycznych – od szybkich prototypów rozwiązań (zgodnie z ideą *rapid development*) po duże, wielomodułowe aplikacje składające się z kilkuset klas. Jego wysoka przydatność jest powszechnie znana od wielu lat w takich obszarach, jak programowanie aplikacji sieciowych, internetowych i mobilnych, przetwarzanie obrazów i sygnałów audio, tworzenie gier komputerowych itp. Co ważne, od jakiegoś czasu w szerokiej gamie powiązanych ze sobą pakietów i bibliotek możemy odnaleźć wiele nowoczesnych i wydajnie zaimplementowanych algorytmów uczenia maszynowego (ang. *machine learning*), statystycznych systemów uczących się, wizualizacji danych itd. Wszystko to sprawia, że to właśnie Python staje coraz częściej narzędziem z wyboru dla specjalistów *data science* – ma ono bowiem ogromne możliwości.

Wśród dostępnych na polskim rynku wydawniczym pozycji poświęconych wprowadzeniu do programowania w „bazowym” języku Python możemy wymienić m.in. [1, 8, 18, 27, 34, 35, 46]. Zainteresowani szeroko pojętą analizą danych do tej pory zmuszeni byli już jednak odwoływać się do lepszych lub gorszych angielskojęzycznych tytułów [4, 9, 20, 26, 40, 41, 43, 48]. Niniejsza książka jest pierwszą polską pozycją poświęconą w całości przetwarzaniu i analizie danych z wykorzystaniem środowiska Python.

Cel książki i jej adresaci. Książka stawia sobie za cel przygotować Czytelnika do samodzielnego przeprowadzenia procesu analizy danych, od pobrania i załadowania zbioru danych, poprzez jego wstępne przetworzenie i wyczyszczenie, aż po samą analizę.

Czytelnik będzie potrafił dokonać wizualizacji danych oraz przedstawić wyniki analizy w formie raportów. Ponieważ wierzymy, że lepiej dać przysłowiową wędkę niż złowioną rybę, kładziemy nacisk na dokładne omówienie samego języka Python 3 i najważniejszych pakietów towarzyszących, tak by Czytelnik był przygotowany na twórcze mierzenie się z nowymi problemami oraz dalsze samodzielne zgłębianie literatury przedmiotu. Wiemy, że pewne rozwiązania, które stworzy Czytelnik, będą przeznaczone do wielokrotnego użytku i tym samym będą zasługiwać na wdrożenie w ramach większych projektów informatycznych. Z tego powodu omawiamy także zestaw dobrych praktyk z zakresu inżynierii oprogramowania.

Książka jest przeznaczona dla szerokiego grona odbiorców, m.in. (obecnych i przyszłych) analityków danych, *data scientists*, specjalistów *business intelligence*, naukowców, badaczy i programistów. Szczególnie polecamy ją:

- Osobom, które już mają wiedzę teoretyczną i umiejętności w zakresie wizualizacji danych, statystyki lub uczenia maszynowego, które wykorzystywały oprogramowanie typu R, SAS czy SPSS w analizie danych i które pragną wreszcie poznać, jak wykonać podobne czynności w środowisku Python.
- Programistom języka Python zainteresowanym inżynierią i analizą danych w całej swej okazałości. Proponowany kurs zapoznaje Czytelnika z najistotniejszymi technikami, dając przy tym inspirację i podstawy do dalszego – samodzielnego już – zgłębiania wiedzy teoretycznej i rozwoju umiejętności praktycznych. Pozwala też zrozumieć wyzwania stojące przed analitykami danych i znaleźć z nimi „wspólny język”.
- Osobom, które znają podstawy programowania w jakimś innym języku, a które chcą poznać język Python w kontekście, którego nie oferują inne podręczniki. Książka bowiem zawiera całościowy kurs języka – problemy analizy danych dla niektórych mogą być po prostu niewinnym pretekstem do wykonania całej serii przyjemnych ćwiczeń. Ciekawe zbiory danych prowokują do stawiania pytań i poszukiwania niebanalnych odpowiedzi, do których droga prowadzi przez programowanie. Po zapoznaniu się z przedstawionym tutaj materiałem Czytelnika nie zdziwi żadna zawilość ani cecha języka, będzie on w stanie swobodnie posługiwać się dokumentacją różnych pakietów, a także samodzielnie utworzyć większy fragment oprogramowania.

Pisząc ten podręcznik, mieliśmy również na uwadze potrzeby studentów i wykładowców kierunków takich jak matematyka, statystyka, informatyka, fizyka czy ekonomia – jako podręcznik wspomagający zajęcia ze statystyki lub uczenia maszynowego bądź będący podstawą kursu wprowadzającego do szeroko rozumianej inżynierii i analizy danych. Czerpaliśmy z doświadczenia zdobytego w trakcie prowadzenia zajęć na Wydziale Matematyki i Nauk Informatycznych Politechniki Warszawskiej oraz szkoleń z cyklu *Data Science Retreat* w Berlinie.

Struktura książki. Materiał podzieliliśmy na cztery następujące części. Ufamy, że ten przejrzysty układ pozwoli nie tylko lepiej uporządkować przyswajaną wiedzę, ale i później – każdorazowo w razie potrzeby – łatwo wyszukiwać potrzebne informacje.

Podstawy języka Python 3. Zaczynamy od omówienia instalacji dystrybucji Anaconda i sposobów pracy z notatnikami Jupyter w rozdz. 1. Następnie omawiamy w wyczerpujący sposób podstawy „bazowego” języka Python – w każdym razie w zakresie potrzebnym do rozpoczęcia prawdziwej przygody z *data science*. W szczególności interesują nas najważniejsze typy danych: wartości skalarne (rozdz. 2), listy, krotki i inne typy sekwencyjne oraz iterowalne (rozdz. 3), słowniki i zbiory (rozdz. 4), a także instrukcje sterujące (rozdz. 5) i sposoby definiowania własnych funkcji (rozdz. 6).

Przetwarzanie danych. W drugiej części zajmujemy się zagadnieniami związanymi ze wstępnym przetwarzaniem danych i przygotowaniem ich do analizy. Omawiamy szczegółowo pakiet *numpy*, który udostępnia wektory, macierze i inne n -wymiarowe tablice a także szeroką gamę metod i funkcji operujących na nich (rozdz. 7). Dalej skupiamy się na opartym na *numpy* pakiecie *pandas*, przy którego użyciu możemy reprezentować i przekształcać rekordy zapisane w postaci tabelarycznej (rozdz. 8). Nie zapominamy przy tym o innych ważnych zagadnieniach: przetwarzaniu napisów i wydobyciu wiedzy z informacji tekstowych (rozdz. 9), obsłudze plików i automatycznym zbieraniu informacji z internetu (rozdz. 10), a także łączeniu się z bazami danych SQL (rozdz. 11). Co więcej, na zakończenie tej partii materiału przedstawiamy zestaw kilkudziesięciu ćwiczeń, które są poświęcone najczęściej wykonywanym w praktyce operacjom na ramkach danych, m.in. wyszukiwaniu informacji, przekształceniu zmiennych, filtrowaniu wierszy i kolumn, agregacji zmiennych w podgrupach utworzonych przez kombinacje wielu czynników oraz złączaniu tabel.

Analiza danych. W trzeciej części nasza uwaga jest skupiona na szeroko pojętej analizie danych, czyli na różnorodnych metodach, które pozwalają przekuć surowe informacje na użyteczną wiedzę. Najpierw poznajemy pakiety *matplotlib* i *seaborn*, na których podstawie będziemy dokonywać wizualizacji różnych aspektów udostępnionych nam danych oraz wyników przeprowadzanych analiz (rozdz. 12). Następnie przechodzimy do opisu dostępnych w środowisku Python metod statystycznych (rozdz. 13) – w szczególności problemów estymacji nieznanymi parametrów i charakterystyk rozkładów oraz weryfikacji hipotez. Dzięki nim będziemy potrafili odpowiadać na pewne istotne pytania w sytuacji, gdy mamy do czynienia z *nie-wielkimi* próbkami, np. czy wpływ określonego czynnika na zachowanie się pewnej zmiennej jest rzeczywiście istotny. Z kolei w rozdz. 14 omawiamy trzy najważniejsze grupy algorytmów maszynowego uczenia się: regresji, klasyfikacji i analizy skupień. Przy ich użyciu możemy modelować różne rodzaje zależności między zmiennymi, przewidywać wartości kluczowych charakterystyk dla jeszcze niezaobserwowanych próbek oraz dokonywać automatycznej segmentacji (podziału) zbioru danych na ciekawe podgrupy.

Tworzenie własnego oprogramowania. Ostatnią część książki poświęcamy zagadnieniom z dziedziny inżynierii oprogramowania – dobrze działający proces przetwarzania czy modelowania danych nierzadko należy wdrożyć jako część większego projektu informatycznego. I tak w rozdz. 15 poznajemy sposoby tworzenia własnych modułów, pakietów i skryptów, a w 16 – własnych klas, czyli nowych typów danych.

Dzięki nim możemy efektywniej panować nad złożonością pisanych przez nas programów i wygodniej dzielić się efektami naszej pracy z innymi.

WAŻNE

Materiały uzupełniające do książki, m.in. zbiory danych, przykładowe skrypty i erraty, udostępniliśmy na stronie github.com/gagolews/Analiza_danych_w_jezyku_Python/.

Podziękowania. Chcielibyśmy serdecznie podziękować naszym współpracownikom i przyjaciołom: Barbarze Żogale-Siudem (Instytut Badań Systemowych PAN) i Grzegorzowi Siudemowi (Wydział Fizyki Politechniki Warszawskiej) za liczne uwagi, komentarze i erraty do wstępnej wersji niniejszej książki. Dziękujemy także naszym wyjątkowym studentom na Wydziale Matematyki i Nauk Informacyjnych PW – w szczególności Natalii Potockiej, Martynie Śpiewak oraz Małgorzacie Dobkowskiej – za uczestnictwo w jej „beta testach” oraz p. Izabeli Mika (Instytut Podstawowych Problemów Techniki PAN) za wiele cennych porad redakcyjnych i językowych.

*Marek Gągolewski
Maciej Bartoszuł
Anna Cena*

Warszawa, sierpień 2016 r.