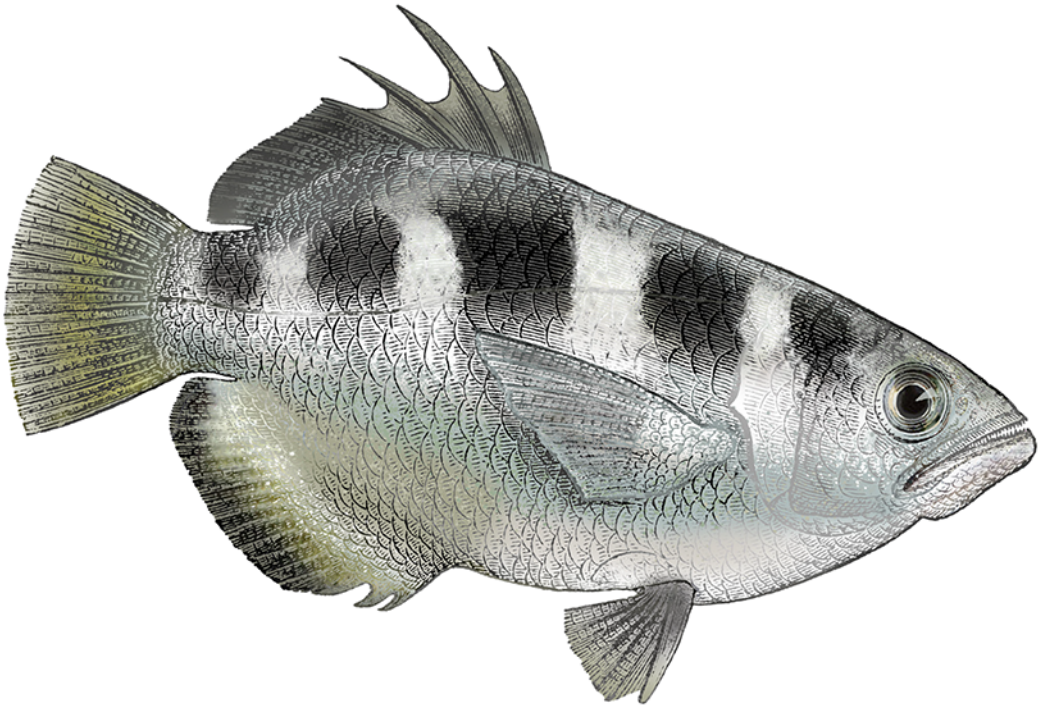
O'REILLY®

Helion

Myślenie statystyczne

Jak analizować dane i wydobywać z nich wiedzę

Wydanie III



Allen B. Downey

Tytuł oryginału: Think Stats: Exploratory Data Analysis, 3rd Edition

Tłumaczenie: Tomasz Walczak

ISBN: 978-83-289-3113-8

Authorized Polish translation of the English edition of *Think Stats, 3rd Edition*
ISBN 9781098190255 © 2025 Allen B. Downey

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
helion.pl/user/opinie/stamy3
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

ROZDZIAŁ 8.

Szacowanie

Wyobraź sobie, że mieszkasz w mieście liczącym 10 000 mieszkańców i chcesz przewidzieć, kto wygra zbliżające się wybory. Teoretycznie możesz zapytać każdego mieszkańca, na kogo zamierza głosować. Jeśli wszyscy odpowiedzą szczerze, możesz na tej podstawie dokonać wiarygodnej prognozy.

Jednak nawet w małym mieście przeprowadzenie ankiety wśród całej populacji może być niepraktyczne. Na szczęście nie jest to konieczne. Jeśli przepytasz losowo wybraną grupę osób, można na podstawie tej próby wnioskować o preferencjach wyborczych całej populacji. Ten proces, czyli wykorzystywanie próby do wyciągania wniosków o populacji, nazywany jest wnioskowaniem statystycznym.

Wnioskowanie statystyczne obejmuje szacowanie (inaczej estymację), któremu poświęcony jest ten rozdział, oraz testowanie hipotez, co omawiam w następnym rozdziale.

Ważenie pingwinów

Wyobraź sobie, że jesteś naukowcem prowadzącym badania na Antarktyce, studiującym lokalną populację pingwinów. Jednym z Twoich zadań jest monitorowanie średniej wagi pingwinów, która zmienia się w ciągu roku. Ważenie każdego pingwina żyjącego w tym środowisku byłoby niepraktyczne, więc Twój plan zakłada wybór każdego tygodnia losowej próby 10 pingwinów, ważenie ich i szacowanie na podstawie tej próby średniej dla całej populacji, czyli **średniej populacji**.

Istnieje wiele sposobów szacowania średniej populacji na podstawie próby, ale tu skupiam się na dwóch metodach: średniej z próby i medianie z próby. Obie te techniki są rozsądnym wyborem. Sprawdź, która okaże się lepsza. Zastanów się również, co właściwie oznacza określenie „lepsza” w tym kontekście.

Na potrzeby przykładu załóżmy, że wagi pingwinów pochodzą z rozkładu normalnego o znanej średniej i ustalonym odchyleniu standardowym. Oznacz średnią i odchylenie standardowe jako `mu` i `sigma` oraz przypisz im wartości w kilogramach:

```
mu = 3.7
sigma = 0.46
```

Te wartości są **parametrami** rozkładu normalnego, co oznacza, że określają one konkretny rozkład. Gdy znasz te parametry, możesz użyć biblioteki NumPy do zasymulowania procesu próbkowania i wygenerowania próby o dowolnej wielkości. Na przykład oto hipotetyczna próba 10 wag:

```
sample = np.random.normal(mu, sigma, size=10)
sample
```

```
array([4.44719887, 3.41859205, 3.45704099, 3.20643443, 4.09808751,
       2.6412922 , 4.50261341, 3.34984483, 3.84675798, 3.58528963])
```

A oto średnia i mediana dla próby:

```
np.mean(sample), np.median(sample)
```

```
(3.6553151902291945, 3.521165310619601)
```

Średnia i mediana różnią się na tyle, że warto się zastanowić, która z nich jest lepszym estymatorem. Aby to sprawdzić, użyj poniższej funkcji do wygenerowania hipotetycznych prób o rozmiarze `n`:

```
def make_sample(n):
    return np.random.normal(mu, sigma, size=n)
```

W ramach pierwszego eksperymentu sprawdź, co dzieje się ze średnią i medianą dla próby wraz ze wzrostem jej wielkości. Użyj funkcji `logspace` z biblioteki NumPy do wygenerowania zakresu wartości `ns` od 10 do 100 000 równomiernie rozłożonych na skali logarytmicznej:

```
ns = np.logspace(1, 5).astype(int)
```

Możesz użyć wyrażenia listowego do wygenerowania hipotetycznej próby dla każdej wartości `n`, obliczenia średniej i pobrania wyników:

```
means = [np.mean(make_sample(n)) for n in ns]
```

To samo zrób dla mediany:

```
medians = [np.median(make_sample(n)) for n in ns]
```

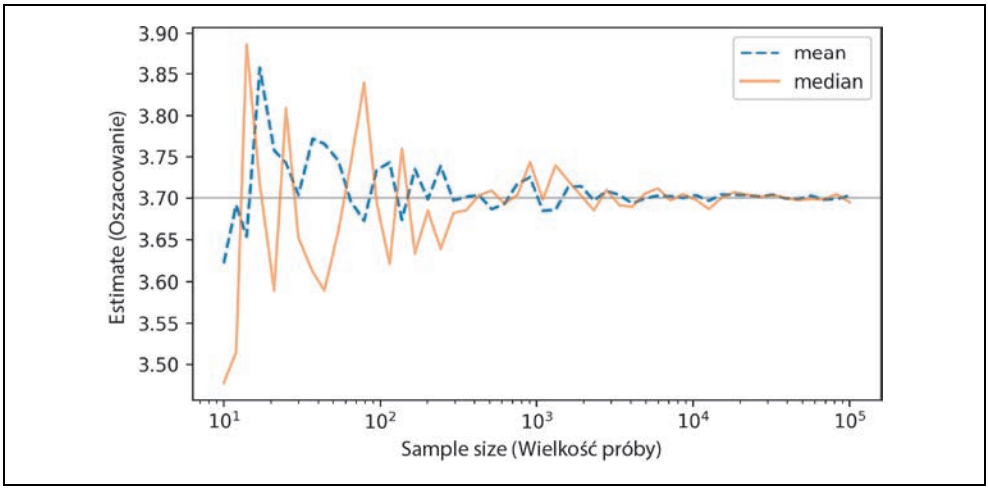
Statystyka używana do szacowania właściwości populacji (na przykład średnia lub mediana z próby) nazywana jest **estymatorem**.

Rysunek 8.1 przedstawia zachowanie tych estymatorów wraz ze wzrostem wielkości próby. Linia pozioma oznacza rzeczywistą średnią dla populacji:

```
plt.axhline(mu, color="gray", lw=1, alpha=0.5)
plt.plot(ns, means, "--", label="mean")
plt.plot(ns, medians, alpha=0.5, label="median")

decorate(xlabel="Sample size", xscale="log", ylabel="Estimate")
```

Dla obu estymatorów wraz ze wzrostem wielkości próby oszacowania zbiegają się do rzeczywistej wartości. Pokazuje to, że są one **zgodne**, co jest jedną z pożądanych cech dobrego estymatora. Jeśli więc uwzględnić tę własność, średnia i mediana wydają się równie dobre.



Rysunek 8.1. Zachowanie estymatorów dla prób o różnej wielkości

Na rysunku 8.1 można zauważyć, że oszacowania są czasami zbyt wysokie, a czasami zbyt niskie. Wygląda na to, że wahania są mniej więcej symetryczne względem prawdziwej wartości. Na tej podstawie można przeprowadzić kolejny eksperyment: jeśli zbierzesz wiele prób o tej samej wielkości i obliczysz wiele oszacowań, jaka będzie ich średnia?

Poniższa pętla symuluje ten scenariusz. Generuje 10 001 prób po 10 pingwinów każda i oblicza średnią dla każdej próby:

```
means = [np.mean(make_sample(n=10)) for i in range(10001)]
np.mean(means)
```

```
3.70034508492869
```

Średnia z uzyskanych wartości średnich jest zbliżona do rzeczywistej średniej użytej do wygenerowania prób: 3,7 kg.

Następna pętla symuluje ten sam scenariusz, ale tym razem oblicza medianę dla każdej próby:

```
medians = [np.median(make_sample(n=10)) for i in range(10001)]
np.mean(medians)
```

```
3.701214089907223
```

Średnia z tych hipotetycznych median jest również bardzo zbliżona do rzeczywistej średniej populacji.

Te wyniki pokazują, że średnia i mediana z próby są estymatorami **nieobciążonymi**. Oznacza to, że po uśrednieniu oszacowań są one poprawne. Słowo „obciążenie” w różnych kontekstach ma różne znaczenia, co może być źródłem nieporozumień. Tu „nieobciążony” oznacza, że średnia z oszacowań odpowiada rzeczywistej wartości parametru.

Do tej pory wykazałem, że oba estymatory są zgodne i nieobciążone. Wciąż jednak nie wiadomo, który z nich jest lepszy. Przeprowadź jeszcze jeden eksperyment, aby sprawdzić, który

estymator jest dokładniejszy. Słowo „dokładność” także może mieć różne znaczenia w zależności od kontekstu. Jednym ze sposobów jej ilościowej oceny jest **błąd średniokwadratowy** (ang. *mean squared error* — MSE). Następna funkcja oblicza różnice między oszacowaniami a rzeczywistą wartością i zwraca średnią z kwadratów tych błędów:

```
def mse(estimates, actual):
    """Błąd średniokwadratowy dla sekwencji oszacowań."""
    errors = np.asarray(estimates) - actual
    return np.mean(errors**2)
```

Warto zauważyć, że błąd średniokwadratowy można obliczyć tylko wtedy, gdy znana jest rzeczywista wartość. W praktyce zazwyczaj jest inaczej. W końcu gdyby rzeczywista wartość była znana, nie trzeba byłoby jej szacować. Jednak w omawianym eksperymencie wiadomo, że rzeczywista średnia populacji wynosi 3,7 kg, więc można użyć jej do obliczenia błędu średniokwadratowego dla średnich z próby:

```
mse(means, mu)
```

```
0.020871984891289382
```

Jeśli próby liczą po 10 elementów i używasz średnich z próby do szacowania średniej populacji, to błąd średniokwadratowy wynosi około 0,021 kg^2. Teraz przyjrzyj się błędowi średniokwadratowemu dla median z próby:

```
mse(medians, mu)
```

```
0.029022273128644173
```

Jeśli używasz median z próby do szacowania średniej populacji, błąd średniokwadratowy wynosi około 0,029 kg^2. W tym przykładzie średnia z próby jest lepszym estymatorem niż mediana z próby. Gdy dane pochodzą z rozkładu normalnego, średnia z próby jest zwykle *najlepszym* nieobciążonym estymatorem średniej populacji (w tym sensie, że minimalizuje błąd średniokwadratowy).

Minimalizacja błędu średniokwadratowego jest pożądaną cechą estymatora, jednak ten błąd nie zawsze jest najlepszym sposobem na podsumowanie odchyleń od rzeczywistej wartości. Po pierwsze, trudno go zinterpretować. W tym przykładzie jednostką błędu średniokwadratowego są kilogramy do kwadratu, przez co trudno jest zrozumieć jego znaczenie.

Jednym z rozwiązań jest użycie pierwiastka kwadratowego z błędu średniokwadratowego (ang. *root mean squared error* — RMSE). Inna możliwość to wykorzystanie średniej z wartości bezwzględnych błędów, nazywanej średnim błędem bezwzględnym (ang. *mean absolute error* — MAE). Poniższa funkcja oblicza średni błąd bezwzględny dla sekwencji oszacowań:

```
def mae(estimates, actual):
    """Średni błąd bezwzględny dla sekwencji oszacowań."""
    errors = np.asarray(estimates) - actual
    return np.mean(np.abs(errors))
```

Oto średni błąd bezwzględny dla średnich z próby:

```
mae(means, mu)
```

```
0.11540433749505272
```

A to średni błąd bezwzględny dla median z próby:

```
mae(medians, mu)
```

```
0.13654429774596036
```

Można oczekiwać, że średnia z próby będzie odbiegać od rzeczywistej średniej o około 0,115 kg, a mediana z próby o 0,137 kg. Dlatego w tym przykładzie lepszym wyborem będzie prawdopodobnie średnia z próby.

Odporność

Rozważ teraz inny scenariusz. Załóżmy, że w 2% prób zważenia pingwina zwierzę przypadkowo naciska przycisk zmiany jednostek i waga zostaje zapisana w funtach zamiast w kilogramach. Jeśli przyjąć, że taki błąd pozostaje niezauważony, wprowadza on do próby wartość odstającą.

Poniższa funkcja symuluje ten scenariusz — mnoży 2% wag przez współczynnik konwersji równy 2,2 funta na kilogram:

```
def make_sample_with_errors(n):
    sample = np.random.normal(mu, sigma, size=n)
    factor = np.random.choice([1, 2.2], p=[0.98, 0.02], size=n)
    return sample * factor
```

Aby zobaczyć, jaki ma to wpływ na rozkład, wygeneruj dużą próbę:

```
sample = make_sample_with_errors(n=1000)
```

Aby zwizualizować rozkład próby (rysunek 8.2), użyj jądrowego estymatora gęstości oraz obiektu `Pdf`, opisanych w podrozdziale „Estymacja jądrowa gęstości” w rozdziale 6.:

```
from scipy.stats import gaussian_kde
from thinkstats import Pdf

kde = gaussian_kde(sample)
domain = 0, 10
pdf = Pdf(kde, domain)
pdf.plot(label='estimated density')
decorate(xlabel="Penguin weight (kg)", ylabel="Density")
```

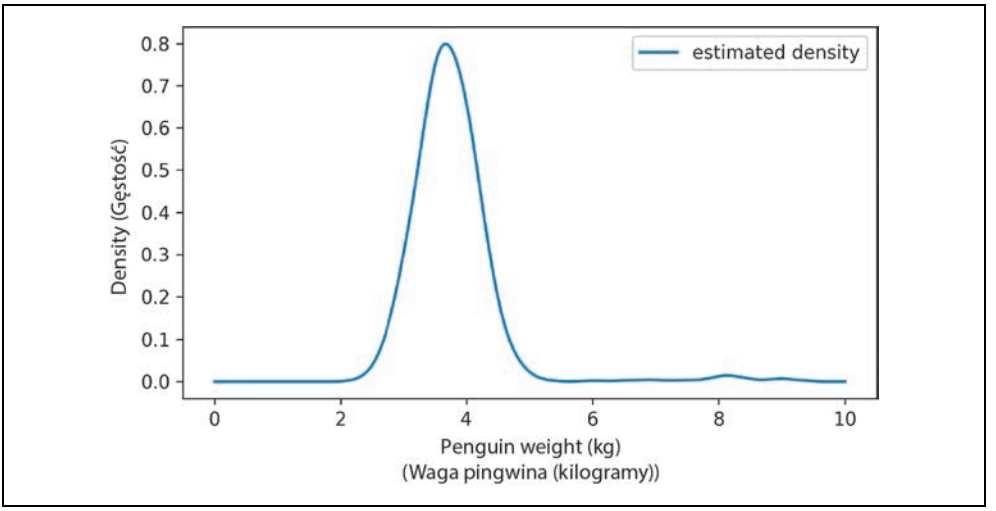
Błędy pomiaru sprawiają, że obok dominanty widocznej w okolicy 3,7 kg pojawia się druga, w pobliżu 8 kg.

Teraz powtórz ten eksperyment, ale tym razem zasymuluj wiele prób o wielkości 10 elementów, oblicz średnią dla każdej próby, a następnie wylicz średnią z tych średnich:

```
means = [np.mean(make_sample_with_errors(n=10)) for i in range(10001)]
np.mean(means)
```

```
3.786352945690677
```

Błędy pomiaru sprawiają, że średnia z próby jest zwykle wyższa niż 3,7 kg.



Rysunek 8.2. Rozkład próby z błędnymi wartościami

A teraz przeprowadź ten sam eksperyment, ale tym razem użyj median z próby:

```
medians = [np.median(make_sample_with_errors(n=10)) for i in range(10001)]
np.mean(medians)
```

```
3.7121869836715353
```

Średnia z median z prób również jest wyższa niż 3,7 kg, ale odchylenie jest mniejsze niż dla średniej ze średnich. Jeśli porównasz błędy średniokwadratowe oszacowań, zobaczysz, że mediany z prób są znacznie dokładniejsze:

```
mse(means, mu), mse(medians, mu)
```

```
(0.06853430354724438, 0.031164467796883758)
```

Jeżeli pomiary rzeczywiście pochodzą z rozkładu normalnego, średnia z próby minimalizuje błąd średniokwadratowy. Jednak w sytuacji, gdy to założenie jest naruszone, średnia z próby nie minimalizuje tego błędu. Mediana z próby jest mniej wrażliwa na wartości odstające, więc jest mniej obciążona i daje mniejszy błąd średniokwadratowy. Estymatory, które dobrze radzą sobie z wartościami odstającymi i podobnymi naruszeniami założeń, nazywane są **odpornymi**.

Szacowanie wariancji

Rozważ teraz inny przykład. Załóżmy, że chcemy oszacować wariancję wagi pingwinów. W podrozdziale „Statystyki podsumowujące” w rozdziale 1. przedstawiłem dwa sposoby obliczania wariancji próby. Obiecałem wtedy, że różnicę między nimi wyjaśnię później — i ten moment właśnie nadszedł.

Stosuje się dwa sposoby obliczania wariancji próby, ponieważ jeden z nich jest estymatorem obciążonym wariancji populacji, a drugi jest estymatorem nieobciążonym. Poniższa funkcja oblicza estymator obciążony, który jest sumą kwadratów odchyleń podzieloną przez `n`:

```
def biased_var(xs):
    # Oblicza wariancję z n w mianowniku.
    n = len(xs)
    deviations = xs - np.mean(xs)
    return np.sum(deviations**2) / n
```

Aby przetestować tę funkcję, wygeneruj za pomocą symulacji wiele prób o liczebności 10, oblicz obciążoną wariancję dla każdej próby, a następnie wylicz średnią z tych wariancji:

```
biased_vars = [biased_var(make_sample(n=10)) for i in range(10001)]
np.mean(biased_vars)
```

```
0.19049277659404473
```

Wynik wynosi około 0,19, ale w tym przykładzie wiadomo, że rzeczywista wariancja populacji to około 0,21. Oznacza to, że ten sposób obliczania wariancji próby daje średnio zaniżone wyniki, co potwierdza, że jest to estymator obciążony.

Następna funkcja oblicza estymator nieobciążony, który jest sumą kwadratów odchyleń podzieloną przez `n-1`:

```
def unbiased_var(xs):
    # Oblicza wariancję z n-1 w mianowniku.
    n = len(xs)
    deviations = xs - np.mean(xs)
    return np.sum(deviations**2) / (n - 1)
```

Aby przetestować tę funkcję, wygeneruj wiele prób i oblicz dla każdej z nich nieobciążoną wariancję:

```
unbiased_vars = [unbiased_var(make_sample(n=10)) for i in range(10001)]
np.mean(unbiased_vars)
```

```
0.21159109492300626
```

Średnia nieobciążonych wariancji prób jest bardzo zbliżona do rzeczywistej wartości, co jest zgodne z oczekiwaniami wobec estymatorów nieobciążonych.

Przy wielkości próby równej 10 różnica między estymatorami obciążonymi a nieobciążonymi wynosi około 10%, co może być znaczącą wartością. Przy próbach o wielkości 100 różnica ta wynosi już tylko 1%, co w praktyce zwykle nie ma znaczenia.

Rozkłady próbkowania

Do tej pory używane były dane z symulacji, uzyskane przy założeniu, że wagi pingwinów mają rozkład normalny o znanych parametrach. Teraz zobacz, co się stanie, gdy użyjesz rzeczywistych danych.

W latach 2007 – 2010 naukowcy ze stacji badawczej Palmer na Antarktyce zmierzyli i zważyli 342 pingwiny z lokalnej populacji. Zebrane przez nich dane są dostępne za darmo. Instrukcje dotyczące ich pobierania znajdują się w notatniku do tego rozdziału.

Do wczytania danych możesz użyć biblioteki Pandas:

```
penguins = pd.read_csv("penguins_raw.csv").dropna(subset=["Body Mass (g)"])
penguins.shape
```

```
(342, 17)
```

Zbiór danych obejmuje trzy gatunki pingwinów:

```
penguins["Species"].value_counts()
```

```
Species
Adelie Penguin (Pygoscelis adeliae)        151
Gentoo penguin (Pygoscelis papua)          123
Chinstrap penguin (Pygoscelis antarctica)   68
Name: count, dtype: int64
```

W pierwszym przykładzie pobierz tylko pingwiny maskowe (`Chinstrap`):

```
chinstrap = penguins.query('Species.str.startswith("Chinstrap")')
```

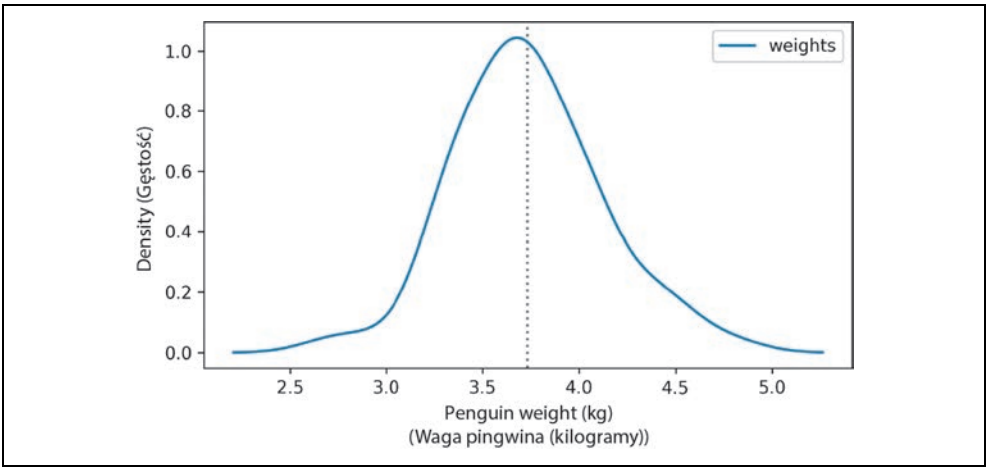
Użyj następującej funkcji do wygenerowania wykresów szacowanych funkcji gęstości prawdopodobieństwa:

```
def plot_kde(sample, name="estimated density", **options):
    kde = gaussian_kde(sample)
    m, s = np.mean(sample), np.std(sample)
    plt.axvline(m, color="gray", ls=":")

    domain = m - 4 * s, m + 4 * s
    pdf = Pdf(kde, domain, name)
    pdf.plot(**options)
```

Rysunek 8.3 przedstawia rozkład wag pingwinów maskowych w kilogramach. Pionowa przerywana linia oznacza średnią z próby:

```
weights = chinstrap["Body Mass (g)"] / 1000
plot_kde(weights, "weights")
decorate(xlabel="Penguin weight (kg)", ylabel="Density")
```



Rysunek 8.3. Rozkład wag pingwinów maskowych

Średnia z próby wynosi około 3,7 kg.

```
sample_mean = np.mean(weights)
sample_mean
```

```
3.733088235294118
```

Jeśli celem jest oszacowanie średniej populacji, 3,7 kg to rozsądna wartość. Ale na ile dokładne jest to oszacowanie?

Jednym ze sposobów na udzielenie odpowiedzi na to pytanie jest obliczenie **rozkładu próbkowania** dla średniej, który pokazuje, jak bardzo oszacowana średnia zmienia się między próbami. Gdyby rzeczywista średnia i rzeczywiste odchylenie standardowe dla populacji były znane, można byłoby zamodelować proces pobierania prób i obliczyć rozkład próbkowania. Jednak gdy rzeczywista średnia populacji jest znana, nie trzeba jej szacować!

Na szczęście istnieje prosty sposób przybliżenia rozkładu próbkowania, zwany **wielokrotnym próbkowaniem**. Podstawowa idea polega na wykorzystaniu próby do stworzenia modelu populacji, a następnie użyciu tego modelu do symulacji procesu pobierania prób.

Zastosujesz **sparametryzowane wielokrotne próbkowanie**, co oznacza, że użyjesz próby do oszacowania parametrów populacji, a następnie wykorzystasz rozkład teoretyczny do wygenerowania nowych prób.

Poniższa funkcja wykonuje ten proces z wykorzystaniem rozkładu normalnego. Zwróć uwagę, że nowe próby mają taki sam rozmiar jak pierwotna próba:

```
def resample(sample):
    m, s = np.mean(sample), np.std(sample)
    return np.random.normal(m, s, len(sample))
```

Następna pętla wykorzystuje funkcję `resample` do wygenerowania wielu prób i obliczenia średniej dla każdej z nich:

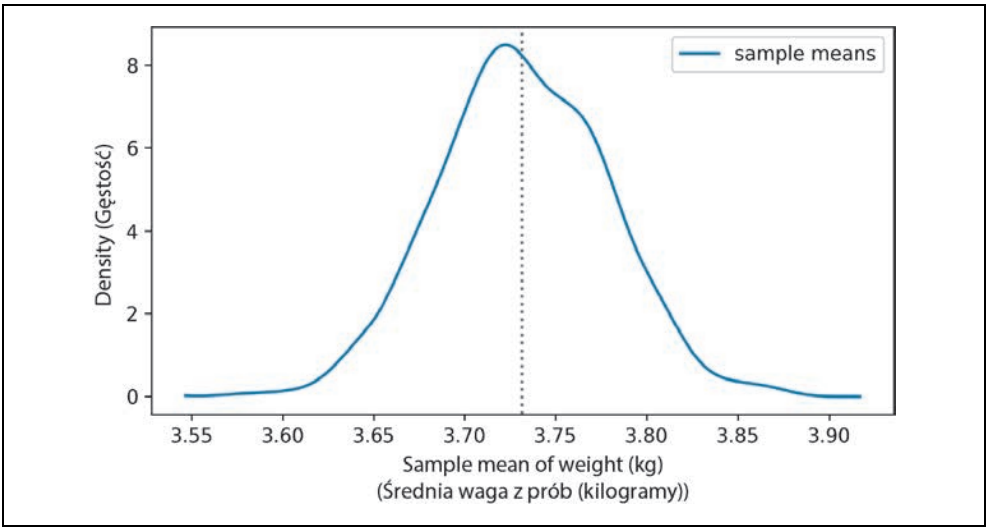
```
sample_means = [np.mean(resample(weights)) for i in range(1001)]
```

Rysunek 8.4 przedstawia rozkład średnich z prób:

```
plot_kde(sample_means, "sample means")
decorate(xlabel="Sample mean of weight (kg)", ylabel="Density")
```

Ten wynik przybliża rozkład próbkowania średnich z prób. Pokazuje to oczekiwaną zmienność średniej z próby dla wielu prób o tej samej wielkości (przy założeniu, że model populacji jest dokładny).

Nieformalnie można powiedzieć, że średnia z próby mogłaby wynosić tylko 3,55 lub aż 3,9, gdyby pobrać inną próbę o tej samej wielkości.



Rysunek 8.4. Rozkład próbkowania średnich z prób

Błąd standardowy

Jedną z metod określania szerokości rozkładu próbkowania jest obliczenie jego odchylenia standardowego. Ta wartość nazywana jest **błędem standardowym**:

```
standard_error = np.std(sample_means)
standard_error
```

```
0.04626531069684985
```

W tym przykładzie błąd standardowy wynosi około 0,045 kg. Oznacza to, że jeśli zbierzesz wiele prób, można oczekiwać, że średnie z tych prób będą się różnić średnio o około 0,045 kg.

Ludzie często mylą błąd standardowy z odchyleniem standardowym. Warto zapamiętać, że:

- odchylenie standardowe określa zmienność pomiarów,
- błąd standardowy określa precyzję oszacowania.

W analizowanym zbiorze danych odchylenie standardowe wagi pingwinów maskowych wynosi około 0,38 kg:

```
np.std(weights)
```

```
0.3814986213564681
```

Z kolei błąd standardowy średniej wagi wynosi około 0,046 kg:

```
np.std(sample_means)
```

```
0.04626531069684985
```

Odchylenie standardowe określa, jak bardzo pingwiny różnią się wagą. Błąd standardowy informuje o precyzji oszacowania. Te statystyki odpowiadają na różne pytania.

Istnieje jednak pewna zależność między tymi wartościami. Gdy znasz odchylenie standardowe i wielkość próby, możesz w przybliżeniu oszacować błąd standardowy średniej w następujący sposób:

```
def approximate_standard_error(sample):
    n = len(sample)
    return np.std(sample) / np.sqrt(n)
```

```
approximate_standard_error(weights)
```

```
0.046263503290595163
```

Ten wynik jest zbliżony do wartości uzyskanej metodą wielokrotnego próbkowania.

Przedziały ufności

Innym sposobem podsumowania rozkładu próbkowania jest obliczenie **przedziału ufności**. Na przykład 90-procentowy przedział ufności zawiera 90% wartości z rozkładu próbkowania. Możemy go określić na podstawie percentyli 5% i 95%. Oto 90-procentowy przedział ufności dla średniej wagi pingwinów maskowych:

```
ci90 = np.percentile(sample_means, [5, 95])
ci90
```

```
array([3.6576334 , 3.80737506])
```

Przy interpretowaniu przedziału ufności można pochopnie stwierdzić, że istnieje 90% prawdopodobieństwa, iż prawdziwa wartość parametru populacji mieści się w 90-procentowym przedziale ufności. W tym przykładzie w tym ujęciu oznaczałoby to, że z 90-procentowym prawdopodobieństwem średnia waga w populacji pingwinów maskowych wynosi od 3,66 do 3,81 kg.

W ścisłym podejściu do prawdopodobieństwa, tak zwanym ujęciu **częstościowym**, taka interpretacja nie jest dopuszczalna. W wielu podręcznikach statystyki można natrafić na informację, że jest ona niepoprawna.

Jednak moim zdaniem to podejście jest niepotrzebnie rygorystyczne. Zgodnie z rozsądnym ujęciem teorii prawdopodobieństwa przedział ufności można interpretować w intuicyjny sposób: istnieje 90% szans, że prawdziwa wartość mieści się w 90-procentowy przedziale ufności.

Przedziały ufności pozwalają jednak określić jedynie zmienność wynikającą z próbkowania, czyli pomiaru tylko części populacji. Rozkład próbkowania nie uwzględnia innych źródeł błędów, w szczególności błędu systematycznego próbkowania i błędu pomiaru. Te zagadnienia omawiam w następnym podrozdziale.

Źródła błędów

Załóżmy, że zamiast średniej wagi pingwinów z Antarktyki chcesz poznać średnią wagę kobiet w mieście, w którym mieszkasz. Nie możesz jednak losowo wybrać reprezentatywnej grupy kobiet i ich zważyć.

Prostszym rozwiązaniem jest „próbkowanie telefoniczne", czyli losowy wybór numerów z książki telefonicznej, wykonanie połączenia, poproszenie o rozmowę z dorosłą kobietą, a następnie zapytanie jej o wagę. Jednak takie podejście ma oczywiste wady.

Na przykład próba ogranicza się do osób posiadających numery w książce telefonicznej, co eliminuje ludzi bez telefonów (którzy mogą być biedniejsi od przeciętnej) lub z zastrzeżonymi numerami (którzy mogą być bogatsi). Ponadto jeśli będziesz dzwonić na numery domowe w ciągu dnia, z mniejszym prawdopodobieństwem dotrzesz do osób pracujących. Ponadto jeśli zapytasz o wagę tylko osobę, która odbierze telefon, zmniejszamy prawdopodobieństwo uwzględnienia innych ludzi korzystających z danej linii telefonicznej.

Jeśli czynniki takie jak dochód, zatrudnienie i wielkość gospodarstwa domowego mają związek z wagą, a jest to prawdopodobne, wyniki badania będą wtedy zniekształcone. Problem ten nazywany jest **błędem doboru próby**, ponieważ wynika z samego procesu pobierania prób.

Proces ten jest również wrażliwy na samoselekcję, która też jest rodzajem błędu systematycznego próby. Niektóre osoby odmówią udzielenia odpowiedzi, a jeśli tendencja do odmowy jest związana z wagą, może to wpłynąć na wyniki badania.

Ponadto jeśli tylko zapytasz ludzi o ich wagę, zamiast rzeczywiście ich ważyć, wyniki mogą być niedokładne. Nawet chętni do pomocy respondenci mogą zaokrąglać wartości w górę lub w dół, jeśli nie czują się komfortowo ze swoją rzeczywistą wagą. A nie wszyscy respondenci są skłonni do współpracy. Te niedokładności są przykładami **błędów pomiaru**.

Przy podawaniu szacunkowych wartości warto określić zmienność wynikającą z próbkowania przez podanie błędu standardowego lub przedziału ufności. Należy jednak pamiętać, że ta zmienność jest tylko jednym (i często nie najistotniejszym) ze źródeł błędów.

Słowniczek

Średnia populacji

Prawdziwa średnia wartość danej wielkości w całej populacji, w odróżnieniu od średniej z próby, obliczanej na podstawie podzbioru danych.

Parametr

Jedna z wartości określających konkretny rozkład w zbiorze rozkładów. Na przykład parametrami rozkładu normalnego są średnia i odchylenie standardowe.

Estymator

Obliczana na podstawie próby miara statystyczna służąca do oszacowania parametru populacji.

Zgodność

Estymator jest zgodny, jeśli wraz ze wzrostem wielkości próby jego wartość dąży do rzeczywistej wartości szacowanego parametru.

Nieobciążony

Estymator jest nieobciążony, jeśli dla danej wielkości próby średnia z oszacowań dla próby jest równa rzeczywistej wartości szacowanego parametru.

Błąd średniokwadratowy (MSE)

Miara dokładności estymatora. Jest to średnia kwadratów różnic między szacowanymi a rzeczywistymi wartościami parametrów (przy założeniu, że prawdziwa wartość jest znana).

Odporność

Estymator jest odporny, jeśli pozostaje dokładny nawet wtedy, gdy zbiór danych zawiera wartości odstające lub błędy albo nie odpowiada idealnie rozkładowi teoretycznemu.

Wielokrotne próbkowanie

Metoda przybliżania rozkładu próbkowania oszacowań przez symulację procesu próbkowania.

Rozkład próbkowania

Rozkład statystyki dla różnych możliwych prób pobranych z tej samej populacji.

Sparametryzowane wielokrotne próbkowanie

Metoda wielokrotnego próbkowania, która polega na szacowaniu parametrów populacji na podstawie danych z próby, a następnie wykorzystaniu rozkładu teoretycznego do symulacji procesu próbkowania.

Błąd standardowy

Odchylenie standardowe rozkładu próbkowania, które określa zmienność oszacowań wynikającą z próbkowania losowego (ale nie uwzględnia błędów pomiaru ani nierepreprezentatywnego doboru próby).

Przedział ufności

Zakres, który zawiera najbardziej prawdopodobne wyniki w rozkładzie próbkowania.

Błąd doboru próby

Wada w sposobie doboru próby, która sprawia, że próba nie jest reprezentatywna dla całej populacji.

Błąd pomiaru

Niedokładność w obserwowaniu, pomiarze lub rejestrowaniu danych.

Ćwiczenia

Ćwiczenie 8.1

Jedną z zalet metod wielokrotnego próbkowania jest łatwość ich rozszerzania na inne statystyki. W tym rozdziale pokazałem, jak obliczyć średnią wagę pingwinów dla próby, a następnie za pomocą wielokrotnego próbkowania przybliżyłem rozkład próbkowania dla średniej. Teraz zrobisz to samo dla odchylenia standardowego.

Oblicz odchylenie standardowe z próby dla wag pingwinów maskowych. Następnie użyj metody `resample` do przybliżenia rozkładu próbkowania dla odchylenia standardowego. Wykorzystaj ten rozkład do obliczenia błędu standardowego oszacowań i 90-procentowego przedziału ufności.

Ćwiczenie 8.2

Zbiór danych z badań BRFSS zawiera samodzielnie podawane wartości wzrostu i wagi dla próby dorosłych osób w Stanach Zjednoczonych. Wykorzystaj te dane do oszacowania średniego wzrostu dorosłych mężczyzn. Użyj metody wielokrotnego próbkowania, aby przybliżyć rozkład próbkowania i obliczyć 90-procentowy przedział ufności.

Ponieważ próba jest bardzo duża, przedział ufności jest bardzo wąski, co oznacza, że zmienność wynikająca z losowego próbkowania jest niewielka. Jednak inne źródła błędów mogą mieć większe znaczenie. Jakie inne źródła błędów mogą Twoim zdaniem wpływać na wyniki w tym przykładzie?

Ćwiczenie 8.3

W grach takich jak piłka nożna czy hokej czas między bramkami zazwyczaj podlega rozkładowi wykładniczemu (co pokazałem w podrozdziale „Gęstość prawdopodobieństwa” w rozdziale 6.). Załóżmy, że obserwujesz próbę czasów między bramkami. Jeśli przyjmiesz, że próba pochodzi z rozkładu wykładniczego, jak możesz oszacować rzeczywistą średnią tego rozkładu? Możesz rozważyć użycie średniej z próby lub mediany z próby. Sprawdź, czy któryś z tych wskaźników jest zgodnym i nieobciążonym estymatorem. W eksperymentach przyjmij, że rzeczywisty średni czas między bramkami wynosi 10 minut:

```
actual_mean = 10
```

Poniższa funkcja generuje próbę z rozkładu wykładniczego o określonej wartości średniej i liczebności:

```
def make_exponential(n):
    return np.random.exponential(actual_mean, size=n)
```

Wykorzystaj tę funkcję do wygenerowania prób o różnych rozmiarach. Oblicz średnią dla każdej z nich. Czy wraz ze wzrostem `n` średnie z prób zbiegają się do rzeczywistej średniej?

Następnie wygeneruj próby o różnych rozmiarach i oblicz medianę dla każdej z nich. Sprawdź, czy mediany prób zbiegają się do rzeczywistej mediany populacji.

W kolejnym kroku wygeneruj wiele prób o rozmiarze 10 i sprawdź, czy średnia z próby jest nieobciążonym estymatorem średniej populacji.

Na koniec sprawdź, czy mediana z próby jest nieobciążonym estymatorem mediany populacji.

Ćwiczenie 8.4

W tym rozdziale zbadałem obciążony estymator wariancji i wykazałem, że faktycznie jest on obciążony. Pokazałem również, że nieobciążony estymator rzeczywiście jest nieobciążony. Teraz sprawdzisz odchylenie standardowe.

Aby oszacować odchylenie standardowe populacji, można obliczyć pierwiastek kwadratowy z obciążonego lub nieobciążonego estymatora wariancji. Wygląda to następująco:

```
def biased_std(sample):
    # Pierwiastek kwadratowy z obciążonego estymatora wariancji.
    var = biased_var(sample)
    return np.sqrt(var)
```

```
def unbiased_std(sample):
    # Pierwiastek kwadratowy z nieobciążonego estymatora wariancji.
    var = unbiased_var(sample)
    return np.sqrt(var)
```

Użyj funkcji `make_sample` do wygenerowania wielu prób po 10 elementów z rozkładu normalnego o średniej 3,7 i odchyleniu standardowym 0,46. Sprawdź, czy którakolwiek z tych prób jest nieobciążonym estymatorem odchylenia standardowego.

Ćwiczenie 8.5

To ćwiczenie bazuje na problemie niemieckich czołgów i jest uproszczoną wersją rzeczywistej analizy przeprowadzonej przez Wydział Wojny Ekonomicznej amerykańskiej ambasady w Londynie podczas II wojny światowej.

Wyobraź sobie, że jesteś szpiegiem aliantów, a Twoim zadaniem jest oszacowanie liczby czołgów zbudowanych przez Niemców. Jako dane masz do dyspozycji numery seryjne odczytane z `k` zdobytych czołgów.

Przy założeniu, że Niemcy mają `N` czołgów ponumerowanych od 1 do `N`, a wszystkie czołgi z tego zakresu mogły zostać zdobyte z równym prawdopodobieństwem, można oszacować `N` w następujący sposób:

```
def estimate_tanks(sample):
    m = np.max(sample)
    k = len(sample)
    return m + (m - k) / k
```

Przyjmij na przykład, że `N` wynosi 122:

```
N = 122
tanks = np.arange(1, N + 1)
```

Do wygenerowania losowej próby `k` czołgów możesz użyć następującej funkcji:

```
def sample_tanks(k):
    return np.random.choice(tanks, replace=False, size=k)
```

Oto przykład:

```
sample = sample_tanks(5)
sample
```

```
array([74, 71, 95, 10, 17])
```

A oto oszacowanie przeprowadzone na podstawie tej próby:

```
estimate_tanks(sample)
```

```
113.0
```

Sprawdź, czy ten estymator jest obciążony.

Ćwiczenie 8.6

W wielu dyscyplinach sportowych, szczególnie w koszykówce, zawodnicy i kibice wierzą w zjawisko zwane „gorącą ręką" (ang. *hot hand*). Oznacza ono, że gracz, który trafił kilka kolejnych rzutów, ma większe szanse na trafienie następnego, a zawodnik, który kilkakrotnie spudłował, prawdopodobnie ponownie nie trafi.

Autorzy znanego artykułu naukowego zaproponowali sposób na sprawdzenie, czy tak zwana gorąca ręka jest rzeczywistym zjawiskiem, czy tylko iluzją. Ich pomysł polegał na analizie sekwencji trafień i chybień w profesjonalnych meczach koszykówki[1]. Dla każdego zawodnika autorzy obliczyli ogólne prawdopodobieństwo trafienia rzutu oraz warunkowe prawdopodobieństwo trafienia po trzech kolejnych celnych rzutach. Dla ośmiu z dziewięciu graczy okazało się, że prawdopodobieństwo trafienia było *niższe* po serii trzech celnych rzutów. Na podstawie tego i innych wyników badacze doszli do wniosku, że „nie ma dowodów na pozytywną korelację między wynikami kolejnych rzutów". Przez kilka dekad wielu ludzi wierzyło, że mit gorącej ręki został obalony.

Jednakże przyjęty wniosek został oparty, przynajmniej częściowo, na błędzie statystycznym. W artykule z 2018 roku wykazano, że statystyka użyta w pierwszym badaniu — prawdopodobieństwo trafienia po trzech kolejnych udanych rzutach — jest obciążona błędem systematycznym[2]. Nawet jeśli szansa na trafienie w każdej próbie wynosi dokładnie 0,5 i nie ma żadnej

[1] T. Gilovich, R. Vallone i A. Tversky, *The Hot Hand in Basketball: On the Misperception of Random Sequences*, „Cognitive Psychology" 17(3), s. 295 – 314.

[2] J.B. Miller i A. Sanjurjo, *Surprised by the Hot Hand Fallacy? A Truth in the Law of Small Numbers*, „Econometrica" 86(6), s. 2019 – 2047.

korelacji między wynikami, prawdopodobieństwo trafienia po trzech sukcesach nadal jest *niższe niż 0,5*.

Nie jest oczywiste, dlaczego tak się dzieje, co tłumaczy, dlaczego ten błąd pozostawał niewykryty przez tak długi czas. Nie będę próbował wyjaśniać w tym miejscu tego zjawiska, ale możesz wykorzystać metody opisane w tym rozdziale, aby to sprawdzić. Użyj następującej funkcji do wygenerowania ciągu zer i jedynek z prawdopodobieństwem 0,5 i bez korelacji:

```
def make_hits_and_misses(n):
    # Generuje losową sekwencję zer i jedynek.
    return np.random.choice([0, 1], size=n)
```

W notatniku do tego rozdziału udostępniam funkcję, która znajduje wszystkie podsekwencje trzech trafień (jedynek) i zwraca element sekwencji następujący po nich.

Wygeneruj dużą liczbę sekwencji o długości 100 elementów, a następnie w każdej z nich znajdź każdy rzut następujący po trzech trafieniach. Oblicz procent tych rzutów, które są trafieniami. Wskazówka: jeśli dana sekwencja nie zawiera trzech kolejnych trafień, funkcja zwróci pustą sekwencję. Musisz uwzględnić w kodzie tę możliwość.

Jaki jest średni odsetek trafień przy wielokrotnym uruchomieniu tej symulacji? Jak zmienia się ten wynik, gdy zwiększasz lub zmniejszasz długość sekwencji?

Jeśli chcesz się szybko nauczyć statystyki i stosowania jej w praktyce, to ta książka jest dla Ciebie!

Zachary del Rosario, adiunkt w Olin College of Engineering

Dla większości z nas statystyka jest poddziedziną matematyki związaną z opracowywaniem teoretycznych podstaw prawdopodobieństwa i wnioskowania statystycznego. Analitycy danych podchodzą do tego inaczej: dla nich statystyka jest niezbędnym zestawem narzędzi i praktyk, które służą do pracy z danymi, odpowiadania na pytania i ułatwiają podejmowanie najlepszych decyzji.

To trzecie wydanie przewodnika cenionego przez analityków danych, inżynierów oprogramowania i pasjonatów danologii. Dzięki niemu szybko nauczysz się korzystać z bibliotek NumPy, SciPy i Pandas. Poznasz różne metody eksploracji i wizualizacji danych, odkrywania zależności i trendów, a także prezentowania wyników. Struktura książki odpowiada rzeczywistemu procesowi pracy ze zbiorem danych: od importowania i oczyszczenia, przez analizę wieloczynnikową, aż po wizualizację uzyskanych wyników.

W książce znajdziesz również takie zagadnienia jak:

- analiza rozkładów danych i wizualizacja wzorców za pomocą bibliotek Pythona
- korzystanie z modeli regresji
- analiza szeregów czasowych i analiza przeżycia
- tworzenie zrozumiałych wizualizacji danych
- rozwiązywanie typowych problemów związanych z analizą danych

Allen B. Downey jest emerytowanym profesorem Olin College of Engineering. Obecnie pełni funkcję głównego danologa w PyMC Labs. Jest znany jako autor serii podręczników z zakresu informatyki i analizy danych.

Helion
helion.pl
HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-3113-8
9 788328 931138
Cena: 89,00 zł